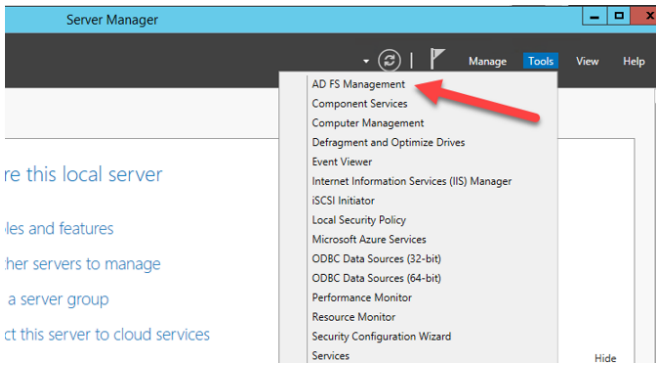
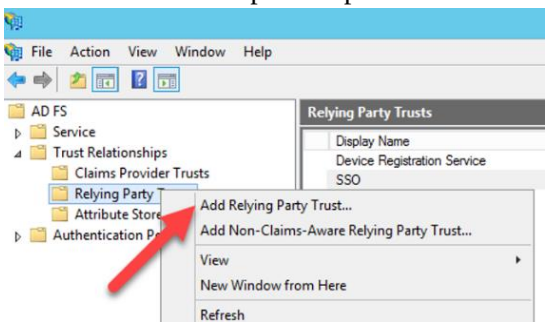
Anexos

Anexo 1

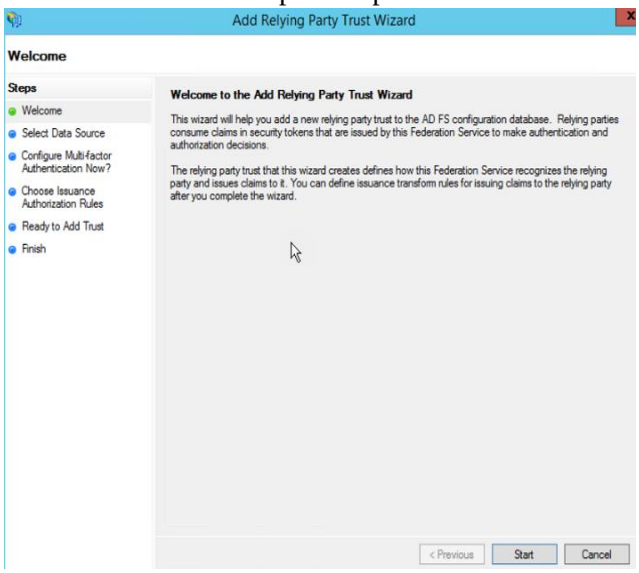
1. Inicie la herramienta de administración de AD FS



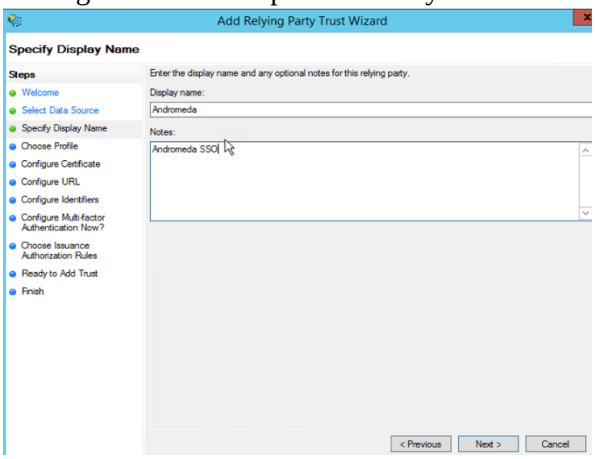
2. Expanda Relación de confianza, haga clic con el botón derecho en Confianza de parte dependiente, luego haga clic en Agregar confianza de parte dependiente



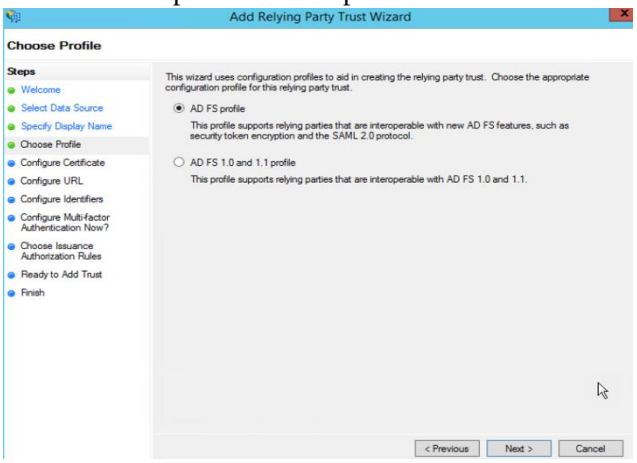
3. Haga clic en Inicio en la pantalla del asistente para ingresar manualmente los datos sobre la confianza de la parte dependiente.



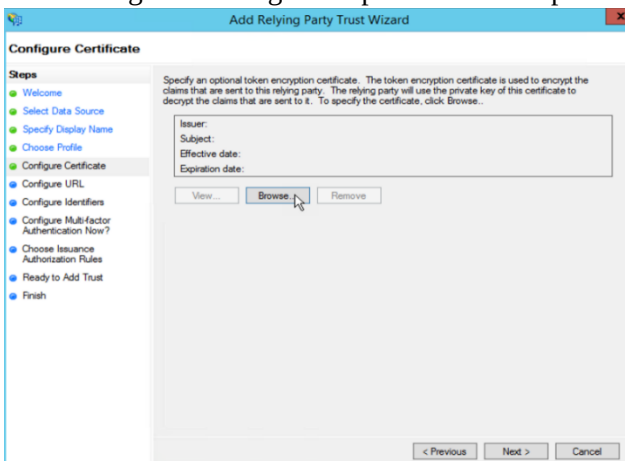
4. Ingrese el nombre para mostrar y las notas



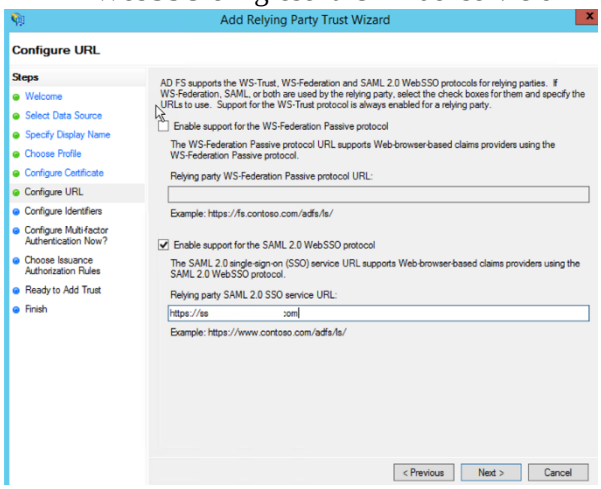
5. Seleccione el perfil de AD FS, en este ejemplo se selecciona la opción uno dado que se la aplicación en el punto final (EndPoint) se comunica por medio del protocolo SAML



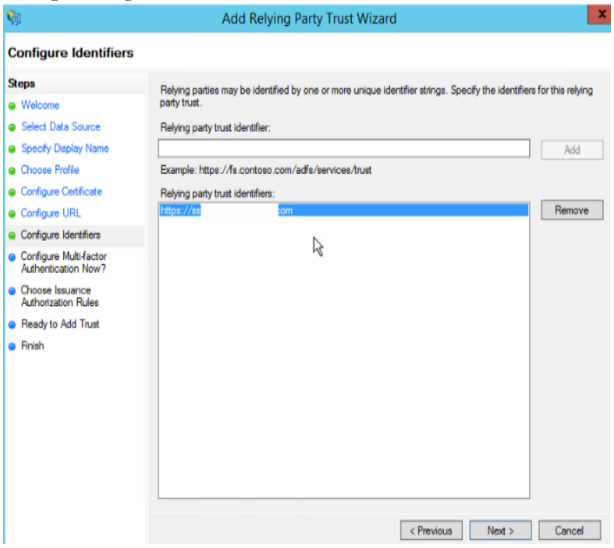
6. Haga clic en siguiente para omitir este paso.



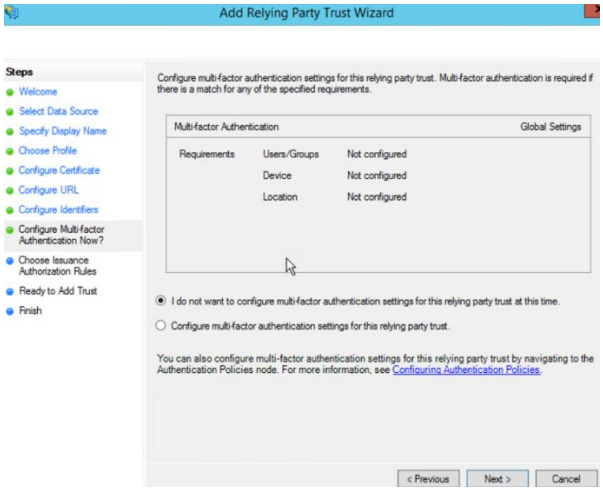
7. Habilite el soporte para el protocolo SAML 2.0 WebSSO e ingrese la URL del servicio



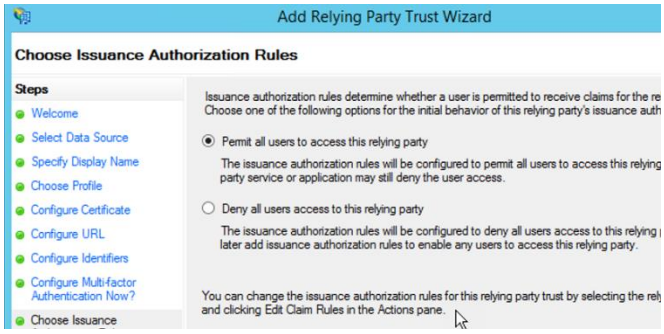
8. Introduzca los identificadores de confianza de la parte que confía



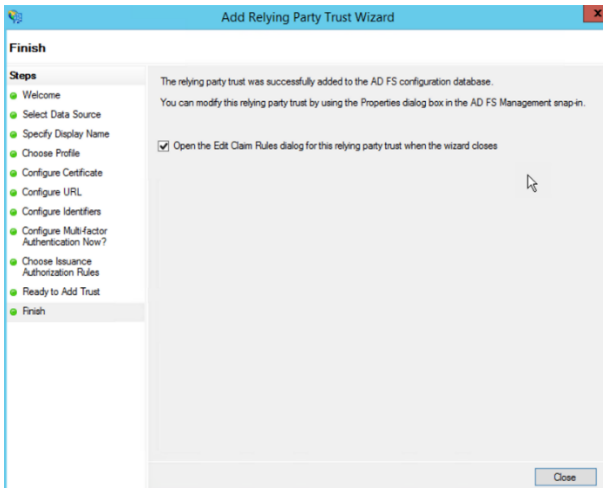
9. En esta configuración, no se habilita la autenticación multifactor



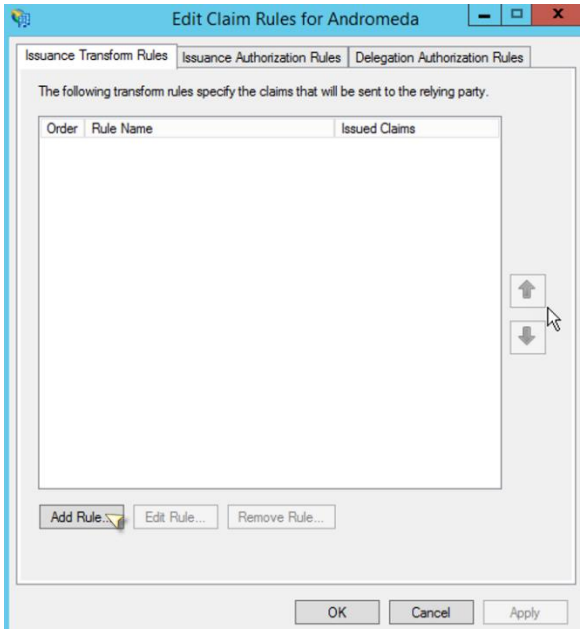
10. Seleccione Permitir a todos los usuarios acceder a la parte de confianza.



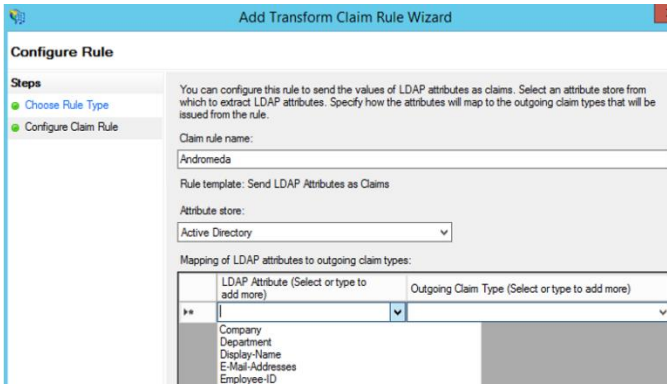
11. Seleccione la opción de “Open the Edit Claims Rules Dialog” y de clic a salir.



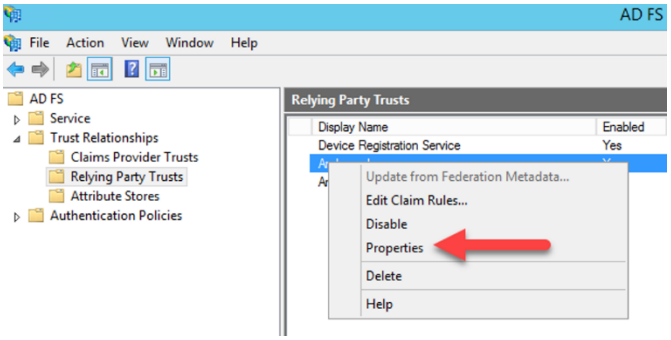
12. Clic a agregar regla



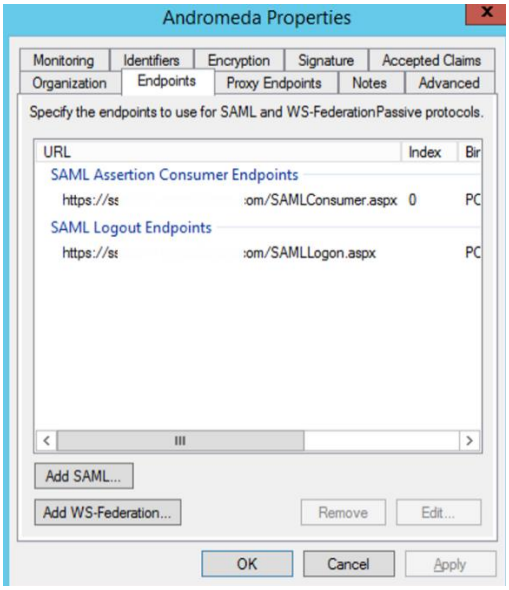
13. Ingrese el nombre de la regla y luego seleccione "Active Directory" para el almacén de atributos. Esta parte es necesaria solo si se desean habilitar atributos en la configuración SSO. Los valores en la columna de Outgoing Claim corresponden a cómo se envían desde la aplicación que consume este servicio.



14. Después de ingresar las reclamaciones (claim), haga clic con el botón derecho en la confianza de la parte que confía, luego haga clic en propiedades



15. Seleccione los puntos finales y actualice según la aplicación consume este recurso, ejemplo:



16. Clic a OK para completar
17. Fin configuración ADFS

Anexo 2

Formato y estructura estándar de los POP3 en el estado de Transaction

Possible Responses:
+OK scan listing follows
-ERR no such message
Examples:
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
...
C: LIST 2
S: +OK 2 200
...
C: LIST 3
S: -ERR no such message, only 2 messages in maildrop
RETR msg
Arguments:
a message-number (required) which may NOT refer to a message marked as deleted
Restrictions:
may only be given in the TRANSACTION state

Ahora si el servidor POP3 envía una respuesta positiva, es tomada como multilínea, después del inicio +OK, el servidor POP3 genera el mensaje correspondiente al número del mensaje.

Possible Responses:
+OK message follows
-ERR no such message
Examples:
C: RETR 1
S: +OK 120 octets
S: <the POP3 server sends the entire message here>
S: .
DELE msg
Arguments:
a message-number (required) which may NOT refer to a message marked as deleted
Restrictions:
may only be given in the TRANSACTION state

Los servidores POP3 fija el mensaje como eliminado, las referencias futuras al número de mensajes asociados con el mensaje en un comando POP3 generan un error, El servidor POP3 no elimina realmente el mensaje hasta que la sesión POP3 entra en el estado UPDATE.

Possible Responses:
+OK message deleted
-ERR no such message
Examples:
C: DELE 1
S: +OK message 1 deleted
...
C: DELE 2
S: -ERR message 2 already deleted
NOOP
Arguments: none
Restrictions:
may only be given in the TRANSACTION state

El servidor POP3 no toma ninguna acción, simplemente da la respuesta positiva.

Possible Responses:
+OK
Examples:
C: NOOP
S: +OK
RSET
Arguments: none
Restrictions:
may only be given in the TRANSACTION state

Si los mensajes se marcan como eliminados no están marcados. Así como se muestra, el servidor POP3 responde con una respuesta positiva.

Possible Responses:
+OK
Examples:
C: RSET
S: +OK maildrop has 2 messages (320 octets)

Anexo 3

Módulos Base y Multiprocesos.

Módulos	Descripción
core	Funciones básicas del Apache que están siempre disponibles.
mpm_common	Colección de directivas que se implementan en más de un módulo multiproceso.
beos	Módulo de multiproceso optimizado para BeOS.
leader	Variable experimental de MPM.
mpm_netware	Módulo de multiproceso que implementa un servidor web optimizado para Novell NetWare.
mpmt_os2	MPM híbrido, multiproceso y multihilo para OS/2.
perchild	Módulo multiproceso que permite a los procesos demonio servir las peticiones que se asignan a distintos id de usuario.
prefork	Implementa un servidor sin hilos.
threadpool	Variante experimental del módulo estándar de MPM.
mpm_winnt	Módulo multiproceso optimizado para Windows NT
worker	Módulo multiproceso que implementa un híbrido multihilos y multiprocesos de servidor Web.

Anexo 4

Módulos Adicionales.

Módulos	Descripción
mod_access	proporciona control de acceso basándose en el nombre del host del cliente, su dirección IP u otras características de la petición del cliente.
mod_actions	este módulo se utiliza para ejecutar Scripts CGI, basándose en el tipo de medio o el método de petición.
mod_alias	proporcionado para mapear diferentes partes del sistema de ficheros del servidor en el árbol de documentos del servidor, y para redirección de URL's.
mod_asis	envío de ficheros que tienen sus propias cabeceras http.
mod_auth	autentificación de usuario utilizando ficheros de texto.
mod_auth_anon	permite a usuarios anónimos acceder a áreas autentificadas.
mod_auth_dbm	proporciona autentificación utilizando ficheros DBM.
mod_auth_digest	autentificación de usuario utilizando MD5.
mod_auth_ldap	permite la utilización un directorio LDAP para almacenar la base de datos de autentificación.
mod_autoindex	muestra los contenidos de un directorio automáticamente, parecido al comando la de Unix.
mod_cache	Cache de contenidos indexados por URI's.
mod_cern_meta	Semántica de etiquetas meta del CERN.
mod_cgi	Ejecución de Scritps CGI.
mod_cgid	ejecución de Scripts CGI utilizando un demonio CGI externo.
mod_charset_lite	para la especificación del juego de caracteres de las traducciones.
mod_deflate	comprime el contenido antes de ser enviado al cliente.
mod_dir	Proporcionado para redirecciones y para servir los ficheros de listado de directorios.
mod_disk_cache	Cache para almacenar contenidos identificados por URI.
mod_echo	Un servidor simple de echo para ilustrar los módulos del protocolo.
mod_env	modificación del entorno que se envía a los scripts CGI y las páginas SSI.
mod_expires	Generación de las cabeceras http Expires, de acuerdo de los criterios especificados por el usuario.
mod_ext_filter	pasa el cuerpo de la respuesta a través de un programa antes de enviársela al cliente.
mod_file_cache	cachea una lista estática de ficheros en memoria.
mod_headers	personalización de las peticiones HTTP y las cabeceras de las respuestas.
mod_imap	proceso de imágenes en el lado del servidor.

mod_include	Documentos HTML generados por el servidor (Server Side Includes).
mod_info	proporciona una visión comprensiva de la configuración del servidor.
mod_isapi	Extensiones ISAPI en Apache para Windows.
mod_ldap	pool de conexiones LDAP y cacheo de resultados para la utilización de otros módulos LDAP.
mod_log_config	registro de las peticiones hechas al servidor.
mod_logio	registro del número de bytes recibidos y enviados en cada respuesta.
mod_mem_cache	Cache de contenidos identificados por URI.
mod_mime	asocia las extensiones de peticiones de los ficheros con el comportamiento del fichero (manejadores y filtros) y contenido (tipos mime, idioma, juego de caracteres y codificación).
mod_mime_magic	determina el tipo MIME de un fichero mirando unos pocos bytes del contenido.
mod_negotiation	se proporciona para la negociación del contenido.
mod_proxy	servidor HTTP/1.1 proxy/gateway.
mod_proxy_connect	extensión de mod_proxy para la gestión de las peticiones CONNECT.
mod_proxy_ftp	soporte FTP para mod_proxy.
mod_proxy_http	soporte HTTP para el módulo mod_proxy.
mod_rewrite	proporciona un motor de reescritura basado en reglas que rescribe las peticiones de URL's al vuelo.
mod_setenvif	permite la configuración de las variables de entorno basándose en las características de la petición.
mod_so	carga del código ejecutable y los módulos en al iniciar o reiniciar el servidor.
mod_speling	intenta corregir las URL mal puestas por los usuarios, ignorando las mayúsculas y permitiendo hasta una falta.
mod_ssl	criptografía avanzada utilizando los protocolos Secure Sockets Layer y Transport Layer Security.
mod_status	proporciona información en la actividad y rendimiento del servidor.
mod_suexec	permite a los scripts CGI ejecutarse con un nombre y grupo específico.
mod_unique_id	proporciona variables de entorno y un identificador único para cada petición.
mod_userdir	directorios específicos para usuarios.
mod_usertrack	registro de actividad de un usuario en el sitio.

Anexo 5

Configuraciones mínimas de seguridad en los servidores Web.

Como se debe manejar las modificaciones para poder configurar la seguridad mínima dentro del servidor Apache hay una seria de pasos a seguir:

1. Deshabilitar la firma digital: La Firma digital en el servidor de Apache se refiere al nombre de la aplicación además de su versión las cuales se muestran al momento de hacer un requerimiento Web. La información básicamente no es necesaria la cual podría ser utilizada para quebrantar la seguridad del servidor.

El proceso para deshabilitarlo se realiza por medio del archivo de configuración de Apache donde podrá realizar las siguientes modificamos a los valores:

ServerSignature off

ServerTokens ProductOnly

2. Deshabilitar el HTTP TRACE: Se utiliza para la devolución la información que se recibe, y de este modo no podrá ser utilizado para modificarlo y que devuelva cookies HTTP para el hurto de sesiones HTTP. Además, se puede utilizar para ataques de Cross Site Scripting o XSS, por este punto se debe deshabilitar para que no sea utilizado como medio para quebrantar la seguridad del servidor y solamente se modifica el archivo de configuración.

TraceEnable off

3. Configuraciones de usuario y grupo Apache: Es muy importante que se asegure que el usuario y grupo, en algunas ocasiones se define en el Root y esto lleva una serie de grieta de seguridad, para poder mitigar esta grieta de seguridad se verifica que este de manera correcta de otro modo debemos modificarlas directivas.

User apache

Group apache

4. Deshabilitar los módulos inutilizados: el servidor Web Apache cuenta con una serie de módulos. Para evaluar cuáles son los módulos de nuestros servidores se están ejecutando se pueden visualizar mandando el siguiente comando bajo el perfil root.

 # grep –n LoadModule /etc/httpd/conf/httpd/conf

Se visualizan todos los módulos que Apache carga. Al realizar el Análisis de la lista y se confirman los que se utilizan en el servidor, los que no se utilizaran, se comente la línea en la configuración.

5. Limitar el tamaño de las solicitudes: cuando se permiten solicitudes demasiado extensas o de mayor tamaño en nuestros servidores Apache se establece la posibilidad de los ataques de Denial of Service (DOS). Apache cuenta con una directiva para limitarlo, la cual siempre esta ilimitada de este modo debemos modificarlos según la necesidad.

LimitRequestBody

6. Limitar el acceso a directorios fuera de su raíz: en algunas ocasiones es necesario que el servidor acceda algún directorio fuera de la raíz, pero si no es este el caso entonces deberemos evitar dar acceso de este modo no se deja vulnerable este punto. Para deshabilitar esto se debe modificar la entrada Directory de Document Root:

<Directory />

Order Deny, Allow

Deny from all

Options None

AllowOverride None

</Directory>

7. Implemente Mod_Security: De los Módulos más importante dentro del servidor web Apache es Mod_Security. Es el Firewall de aplicaciones web que realiza la acción de varias tareas conteniendo también el filtrado simple, filtrado de expresiones regulares, validaciones en las codificaciones de URL. Mod_security es sin duda el módulo más importante dentro de los módulos de Apache.

8. Restricciones de acceso por IP: Se debe restringir el acceso al portal y las sesiones puedan ser vistas desde una IP o en los segmentos de la red se agrega la siguiente línea en los ingresos a la sesión del directorio.

<Directory /sitioweb o carpeta>

Options None

AllowOverride None

Order deny,allow

Deny from all

Allow from 192.168.1.5

Allow from 192.200.0.0/24

De este modo podemos configurar para que nuestro servidor web Apache para que solo permitan los accesos a los directorios indicados en la dirección como el segmento de red.

9. Protéjase de Ataques DDOS: Los ataques DDOS se pueden tratar de mitigar de mejor manera, pero si bien es difícil la protección del cien por ciento, podemos aplicar las diferentes políticas o directrices para mitigar más el riesgo.

Timeout: Esta política ayuda a medir el tiempo de respuesta de un evento antes que finalice de parte del servidor se determina si falla. Su valor por defecto es de 300 segundos. Es recomendable que los tiempos de respuesta se mantenga los más bajos posibles ya que el sitio es constantemente blanco de este tipo de ataque. Además, este valor dependerá también de qué tipo de request recibe el sitio.

MaxClients: Esta directiva nos permite configurar el número de conexiones que vamos a permitir

simultáneamente. Las conexiones nuevas serán puestas en cola a partir del límite que configuremos.

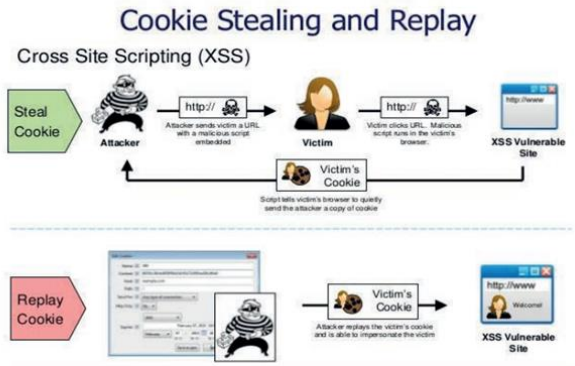
KeepAliveTimeout: Se establece el Máximo de tiempo que el servidor espera para un requerimiento que se solicite después antes que la conexión sea cerrada.

LimitRequestFields: Limita el número de solicitudes de encabezados HTTP que se aceptan del cliente, contiene un valor por defecto de 100, se puede reducir el número según sea conveniente para evitar estar en el radar de ataques DDOS.

LimitRequestFieldSize: Limita el tamaño de las solicitudes de encabezado HTTP.

Anexo 6

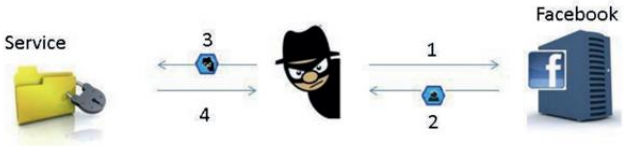
Robo de cookies y tokens



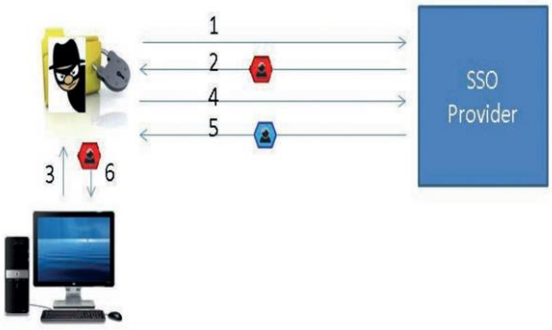
Los ataques en los robos de cookies se pueden prevenir enviando un parámetro oculto en las peticiones HTTP, de este modo no son utilizados los tokens del portador sino los JWT de este modo pueden ser firmados y cifrados de este modo están protegidas por multicapas de seguridad y confianza.

Anexo 7

Suplantar los tokens



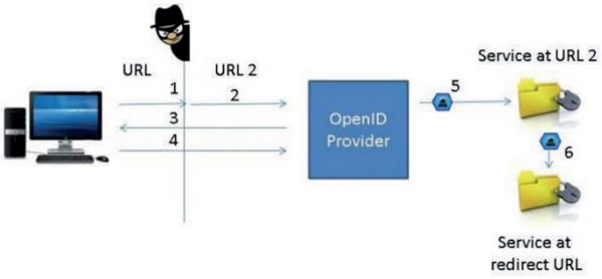
Cuando se realizan los cambios de sesiones es cuando se interceptan las credenciales de un SSO en los agentes de usuarios luego realizan la inserción de las credenciales SSO que interceptan en una construcción HTML, lo cual genera que el navegador envié las credenciales SSO al OP cuando el Exploit es vista.



Lo ideal para poder prevenir de manera efectiva a estos ataques pude ser la criptografía NONCE dentro de la comunicación, de modo se detectará el token que se ha devuelto por el OP no coincide con el que se generó y envió en el inicio de sesión del SSO.

Anexo 8

Redirección abierta

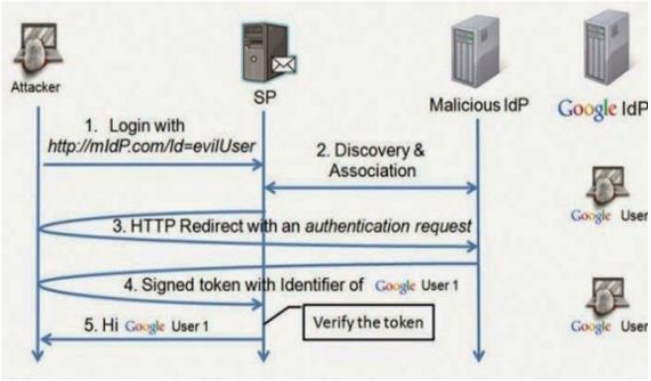


Para mitigar este riesgo de seguridad se debe realizar la validación que el código que de acceso que se recibe sea desde el dominio donde se genera el token para el acceso puede utilizarse una lista blanca que contenga todas las URLs de los redireccionamientos permitidos de este modo lograr evitar que se envié un código de acceso para una dirección no valida.

Anexo 9

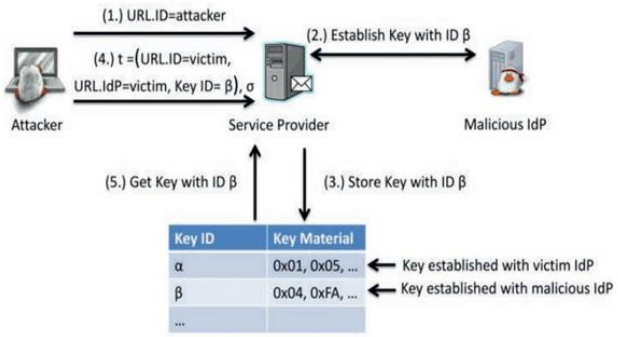
Suplantación de OP

Para ello se establece un relying party se deberá verificar la información descubierta. donde se debe verificar si el identificador introducido coincide con el identificador en el token de autenticación.



Anexo 10

Confusión de llave.



Anexo 11

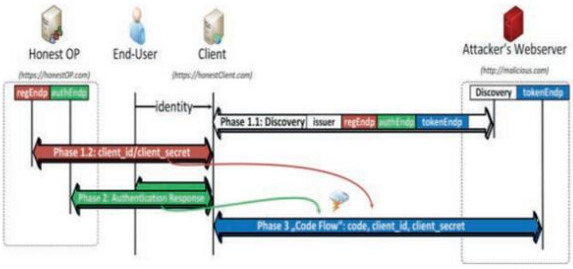
Ataque endpoint malicioso.

Estos ataques se realizan en tres fases:

Fase 1.1: inyección de endpoints malicioso: se establece obligar a un cliente válido a usar el Discovery Service malicioso del atacante. Para ello, construye un vínculo malicioso y lo envía al usuario.

Fase 1.2: registro dinámico: el cliente accede a regEndp para el registro. Se envía una solicitud de registro a https://honestOP. com/register y recibe un ID de cliente y un cliente secreto en la respuesta. Fase 2: autenticación de usuario y autorización: el cliente redirige al usuario al authEndp, https://login.honestOP.com/, donde el usuario tiene que autenticarse a sí mismo y autorizar al cliente.

Fase 3: el robo dependiendo del protocolo de flujo (código o implícito): los diferentes mensajes son enviados al atacante. Este endpoint es un recurso protegido OAuth 2.0 que devuelve peticiones del usuario autenticado. El token de acceso obtenida se envía como un token de portador por el cliente. Por lo tanto, el atacante puede obtener acceso a un token de acceso válido.



Anexo 12

Estructura de un mensaje XML en SOAP

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
Soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Header>
…
</soap:Header>
<soap:Body>
…
<soap:Fault>
…
</soap:Fault>
</soap:Body>
</soap:Envelope>
```

- Específica que es un documento XML y la versión
  - <?xml version="1.0″?>
- Establece el comienza el envelope (sobre) del mensaje)
  - <soap:Envelope
- Elemento envelope asociado con el namespace
  - xmlns:soap = "http://www.w3.org/2001/12/soap-envelope"
- Indicar la ubicación de los tipos de datos.
  - envelope Soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

- Inicio del Header.
    - <soap:Header>
- Finalización del Header.
    - </soap:Header>
- Inicia el Body
    - <soap:Body>
- Notificaciones de Fallo
    - <soap:Fault>
    -
- Cierre de la sección Fault.
    - </soap:Fault>
- Indica el final del cuerpo del mensaje.
    - </soap:Body>
- Fin del mensaje SOAP.
    - </soap:Envelope>

Anexo 13

LISTA DE CHEQUEO PARA LA INVESTIGACIÓN DE SOLUCIONES ACTUALES DE SINGLE SIGN ON Y SOLUCIONES POR SUSCRICIÓN

El siguiente instrumento será utilizado para poder verificar las características de las soluciones single sign on actuales, Soluciones por suscripción: Okta, OneLogin, Auth0.

Solución	Okta	Lanzamiento	Enero 2009
Fabricante	NASDAQ, OKTA(Todd Mckinnon, Frederic Kerrest).		

Nº	Variable	Cumple	Valoración
1	Robustez	✓	Bueno
2	Escalabilidad	✓	Excelente
3	Compatibilidad	✓	Muy bueno
4	Adaptable	✓	Muy bueno
5	Integra con más de un servidor web	×	N/A
6	Servidores de autenticación	✓	Excelente
7	Interfaces	✓	Excelente

Solución	Auth0	Lanzamiento	2006
Fabricante			

Nº	Variable	Cumple	Valoración
1	Robustez	✓	Muy Bueno
2	Escalabilidad	×	N/A
3	Compatibilidad	✓	Excelente
4	Adaptable	✓	Muy Bueno
5	Integra con más de un servidor web	✓	Excelente
6	Servidores de autenticación	✓	Excelente
7	Interfaces	✓	Muy Bueno

Solución	OneLogin	Lanzamiento	2009
Fabricante	Thomas Pedersen, Christian Pedersen		

Nº	Variable	Cumple	Valoración

1	Robustez	×	N/A
2	Escalabilidad	×	N/A
3	Compatibilidad	✓	Muy Bueno
4	Adaptable	✓	Excelente
5	Integra con más de un servidor web	×	N/A
6	Servidores de autenticación	✓	Muy Bueno
7	Interfaces	✓	Excelente

Anexo 14

Este instrumento será utilizado para analizar la información necesaria sobre: Tecnologías para implementación, Servidores de autenticación, Interfaces, Middleware, Protocolos de comunicación (SAML, OpenID, OAuth2.0), Servicios web: SOAP, REST

CUADRO RESUMEN SOBRE TECNOLOGÍAS PARA LA IMPLEMENTACIÓN DE LA SOLUCIÓN SINGLE SIGN ON, SERVIDORES DE AUTENTICACION, INTERFACES, MIDDLEWARE, PROTOCOLOS DE COMUNICACIÓN Y SERVICIOS WEB.

Tecnologías	Unidad evaluada				
	Manejo de mensajes	Rapidez de procesamiento	Manejo de cargas	Arquitectura	Autenticación
Tecnología para la implementación *(SP y seguridad)*	*Certificado de seguridad:* X.509 es utilizado como medida de seguridad en la comunicación de un SSO. Se basa en un par de claves pública y privada para cifrar y descifrar el contenido, están relacionadas matemáticamente y solo se puede descifrar utilizando el otro. *Tokens:* Son una cadena de texto que contiene un significado, una validación o un identificador.	*IIS:* Hilos de procesos, configuración por medio de archivos. *Apache:* El servidor web acepta solicitudes del cliente y envía una respuesta a su petición solicitada. Manejo de procesamientos múltiples y MOD_SSL que habilita SSL v3 y TLS.		*Apache:* Se establece por módulos, se configuran a través de las directivas en cada uno de módulos; estos se clasifican en módulos base, multiproceso y adicionales. *IIS:* Incluye Servicio de Activación de Procesos de Windows (WAS), módulos y tuberías integradas de procesamiento.	*Token*: Se generan del lado de un servidor de autenticaciones para la obtención de un Identificador (ID Token). El token toma un valor para ser utilizado como llaves de acceso.
Servidores de autenticación *(IdP)*			*IdP POP3:* Se manejan tres estados diferentes: Authorization, Transaction, Update (cierra conexión TCP). Provee la comunicación por medio de textos planos. *IdP DB:* Separación física y lógica de datos, permite distribuir	*IdP ADFS:* Windows server, configuración de ROLES. *IdP OpenLDAP*: OpenLDAP tiene dos niveles, uno es el "frontend" que maneja el procesamiento del protocolo y las conexiones de redes. El segundo nivel es el "backend" que hace el trabajo real de almacenar o recuperar datos en respuesta a las solicitudes LDAP. Un overlay es un componente de software que puede ser insertado entre el frontend y el backend e interceptar peticiones y lanzar otras acciones en ellas.	*IdP ADFS*: Producto Microsoft. 1-Usuario navega a una URL proporcionada por el servicio ADFS. 2-ADFS autentica a través del servicio AD. 3-ADFS proporciona al usuario un reclamo de autenticación. 4-la aplicación de destino otorga o niega el acceso en función del servicio de confianza federada creado. *IdP OpenLDAP:* Implementación de servidor

			la concurrencia de peticiones. Replicación de datos.	*IdP POP3:* El método de acceso al servidor de correos se realiza por medio de TCP/IP de manera encriptada utilizando el puerto 995. Comandos de POP3 de 3 a 4 caracteres. *IdP DB:* Datos organizados y relacionados entre sí, los cuales son almacenados por medio de sistemas de información y de manera correcta para luego poder realizar la manipulación de estos. Se puede configurar replicación de datos para seguridad e integridad.	de código abierto de LDAP, junto con Kerberos, *IdP POP3:* Es manejada por el protocolo SMTP. *IdP DB:* Se debe comprender cuál es la estructura en la que se establece la autentificación en las bases de datos: se establece el servidor, estando en el servidor de base de datos se debe asignar un usuario para tener acceso a las bases de datos que desea acceder y por ultimo tiene acceso a sus objetos donde obtendrá la información
Interfaces *(Integración, APIS, Toolkits)*	Las soluciones SSO analizadas integran y documentan el uso de los diferentes protocolos de comunicación. Así mismo proveen amplia documentación sobre las APIs o servicios que exponen para que su consumo.			Las colusiones SSO están desarrolladas en diferentes tecnologías, y dado que se integra de varios componentes tecnológicos, es necesaria la correcta orquestación de todos estos y definir el alcance requerido. Para adaptaciones con estas soluciones se debe seguir la arquitectura definida por el fabricante.	Las soluciones SSO utilizan diferentes IdP, se debe establecer una relación de "confianza" entre el SP e IdP para validar las peticiones sean todas de un origen confiable.
Middleware *(SSO como middleware)*	La comunicación en el SSO es mediante el envío de paquetes de datos que se denominan "mensajes. •Integridad en el envío del mensaje. •Uso de encabezados en		Permite dos modos básicos de comunicación: síncronos y asíncronos	Construye una especie de puente entre los diferentes sistemas al habilitar comunicaciones y transferencia de datos, administra aplicaciones dispares tanto dentro	--

	el mensaje. •Uso de propiedades, atributos o parámetros. •Manejo de estructuras de datos, como bien lo son JSON, XML. Habilita la comunicación maquina a máquina (API)			de una sola organización o entre organizaciones independientes. Provee un conjunto de APIs que se exponen a las aplicaciones para que estas las consuman	
Protocolos de comunicación	*SAML:* Hay tres tipos diferentes de afirmaciones (assertion): Autenticación, atributo, decisión de autorización. Integra certificados de seguridad para validación de los mensajes. *OAuth2.0*: Se debe registrar la o las aplicaciones con el servicio para establecer el envío de mensajes. Integra tokens de acceso. *OpenID Connect:* La comunicación se realiza por medio de los Token ID es un JSON web Token (JWT) el cual consiste en manejar las notificaciones sobre la autenticación de los usuarios.	--		*OpenID Connect*: Es un "perfil" de OAuth 2.0 diseñado específicamente para la liberación y autenticación de atributos. Es descentralizado donde se puede identificar en una página web por medio de la URL, la cual además es verificada por medio de un IdP. *SAML*: Proporciona más control a las empresas para mantener sus inicios de sesión SSO más seguros. Utiliza XML. *OAuth 2.0*: Es mejor en dispositivos móviles y usa JSON. Define cuatro roles: propietario del recurso, cliente, servidor de recursos (API), servidor de autorización (API).	*OAuth 2.0* : Es un marco de autorización y puede usarse para muchas tareas interesantes, una de las cuales es la autenticación de personas. Genera concesiones (grants) *SAML*: Transfiere la identidad del usuario de un lugar (el proveedor de identidad) a otro (el proveedor de servicios).
Servicios web	*SOAP:* Es un protocolo estándar que es definido por dos objetos en diferentes procesos para la comunicación por el intercambio de datos XML. Facilita la comunicación al estilo RPC entre el cliente y servidor de manera remota. *REST:* No se considera un estándar y como se ha mencionado es un estilo de arquitectura. Sin embargo, está	--	*SOAP:* No se asocia a: lenguajes, protocolo de transporte. Aprovecha estándares existentes, interoperabilidad entre múltiples entornos.	*REST:* La arquitectura se refiere a una colección de principios, que definen como los recursos son definidos y diseccionados. REST describe la interfaz de transmisión de datos específicos de un domino sobre HTTP sin una capa adicional, como hace SOAP.	

	basado en los siguientes estándares: HTTP, URL, MymeType, XML/HTML/JSON. Cada mensaje contiene toda la información necesaria para comprender y completar la petición, lo que significa que la comunicación entre las partes no guarda estado alguno.		*REST:* Escalabilidad de la interacción con los componentes, puesta en funcionamiento independiente, generalidad de interfaces, compatibilidad con componentes intermedios. Ambos manejan códigos de estado.		

Anexo 15

LISTA DE CHEQUEO PARA LA INVESTIGACIÓN DE PORTAFOLIO DE SOLUCIONES WEB, COSTOS DE ADQUIISICION, DESARROLLO E INFRAESTRUCTURA.

El siguiente instrumento será utilizado para poder verificar las características del: Portafolio de soluciones web, Costos de adquisición, Costos de desarrollo, Costos en infraestructura.

Solución	Soluciones Web			
Nº	Variables	Aplica	Cumple	No cumple
1	Gestiona los inicios de sesión	✓	Excelente	--
2	Políticas de seguridad	✓	Muy Bueno	--
3	Autenticación multifactor	✓	Bueno	--
4	Mantenimiento.	✓	--	Regular
5	Tipos de suscripciones.	×	N/A	N/A
6	Recursos humanos: cambios, tiempo.	×	N/A	N/A
7	Equipo para desarrollo.	×	N/A	N/A
8	Mitigación de riesgos.	×	N/A	N/A
9	Infraestructura.	×	N/A	N/A

Costos	Adquisición			
Nº	Variables	Aplica	Cumple	No cumple
1	Gestiona los inicios de sesión	×	N/A	N/A
2	Políticas de seguridad	×	N/A	N/A
3	Autenticación multifactor	✓	Bajo	--
4	Mantenimiento.	✓	Medio	--
5	Tipos de suscripciones.	✓	Medio	--
6	Recursos humanos: cambios, tiempo.	×	N/A	N/A
7	Equipo para desarrollo.	×	N/A	N/A
8	Mitigación de riesgos.	×	N/A	N/A
9	Infraestructura.	✓	Bajo	--

Costos	Desarrollo			
Nº	Variables	Aplica	Cumple	No cumple
1	Gestiona los inicios de sesión	×	N/A	N/A
2	Políticas de seguridad	✓	Medio	--
3	Autenticación multifactor	×	N/A	N/A
4	Mantenimiento.	✓	Bajo	--
5	Tipos de suscripciones.	×	N/A	N/A
6	Recursos humanos: cambios, tiempo.	✓	Medio	--
7	Equipo para desarrollo.	✓	Medio	--
8	Mitigación de riesgos.	×	N/A	N/A
9	Infraestructura.	✓	--	Muy Bajo

Costos	Infraestructura			
Nº	Variables	Aplica	Cumple	No cumple
1	Gestiona los inicios de sesión	×	N/A	N/A
2	Políticas de seguridad	×	N/A	N/A
3	Autenticación multifactor	×	N/A	N/A
4	Mantenimiento.	✓	Medio	--
5	Tipos de suscripciones.	×	N/A	N/A
6	Recursos humanos: cambios, tiempo.	✓	Bajo	--
7	Equipo para desarrollo.	×	N/A	N/A
8	Mitigación de riesgos.	✓	Medio	--
9	Infraestructura.	✓	Medio	--