UNIVERSIDAD DON BOSCO

VICERRECTORÍA DE ESTUDIOS DE POSTGRADO

TRABAJO DE GRADUACIÓN

MECANISMO DE COMUNICACIÓN Y PROTOCOLO DE CREACIÓN DE CÓDIGOS DE VERIFICACIÓN ANTIPHISHING

PARA OPTAR AL GRADO DE

MAESTRO EN SEGURIDAD Y GESTIÓN DE RIESGOS INFORMÁTICOS

ASESOR:

M. EN C. LIL MARÍA RODRÍGUEZ HENRÍQUEZ

PRESENTADO POR:

JOSUÉ MARIO GUILLÉN ROSALES

Antiguo Cuscatlán, La Libertad, El Salvador, Centroamérica

Marzo 2015

AGRADECIMIENTOS

Agradezco primeramente a Dios, por haberme bendecido con las capacidades de razonamiento y análisis, las cuales son herramientas en mi diario vivir, en mi trabajo, en mis estudios, en mis proyectos y en mis relaciones interpersonales.

Agradezco a mis padres, por haberme inculcado los valores morales que rigen mi comportamiento para con las personas, la sociedad y el medio ambiente, los cuales me obligan a ser una mejor persona, respetuosa de Dios, de mis prójimos y de mi entorno.

Agradezco a mi alma máter la Universidad Don Bosco y a mis docentes, por haberme dado las herramientas y el conocimiento científico necesarios que me permitieron culminar una etapa más en mi desarrollo académico.

Agradezco a la M. en C. Lil María Rodríguez, por asesorarme en mi proceso de graduación, dándome valiosos aportes a mi investigación y ayudarme a expresar de forma clara y profesional los conceptos y razonamientos que mi trabajo de graduación produjo.

Agradezco a mis compañeros y compañeras, de los cuales pude aprender de su experiencia en cada una de sus áreas profesionales; por su compañerismo y amistad a lo largo de la maestría y que durará más allá de la universidad.

Mario Guillén

ÍNDICE

RESUMEN

En la actualidad el *phishing* es un problema de suplantación de identidad muy explotado. Por lo general, en este tipo de ataque el adversario envía un correo electrónico suplantando la identidad de alguna otra entidad, por ejemplo, un correo electrónico solicitando un restablecimiento de contraseña de una cuenta de banca en línea. La aplicación de filtros de correo o el cifrado de mensajes se vuelven soluciones efectivas pero poco viables en cuanto a costos o complejidad del lado del usuario final. Por ello, en este artículo proponemos un mecanismo de comunicación y un protocolo para la creación y envío de códigos de verificación antiphishing, con los cuales damos una solución alternativa de bajo coste para el remitente y fácil verificación para el destinatario.

Palabras clave*:* Phishing, Antiphishing, Correo electrónico, Códigos de verificación.

INTRODUCCIÓN

El spam ha sido una de las prácticas clásicas desde que se inventó el correo electrónico, y es que es una forma fácil, efectiva y directa de llegar a millones de usuarios en un solo intento. Según el Informe de Seguridad de Cisco de 2013, el volumen de spam había disminuido un 18% en los últimos años [1]; sin embargo, esto no es suficiente, ya que el spam malicioso sigue siendo una amenaza. En el Informe de Seguridad de Cisco de 2014, los temas para mensajes de spam siguen manteniéndose en depósitos bancarios, notificaciones de pago, compras por internet, fotos adjuntas, notificaciones de envío, citas por internet, impuestos, Facebook, tarjetas o vales de regalos y PayPal [2].

Los atacantes de spam han tenido que reinventarse ante las medidas defensivas que los proveedores de servicio de correo electrónico han tomado. El spam malicioso lo que busca es explotar la confianza del usuario a través de contenidos de correo electrónico muy similar o idéntico al de un remitente conocido. El problema con los contenidos, es el *phishing*, ya que estos correos pueden llevar a los usuarios a sitios web maliciosos, donde se les solicite información confidencial que está siendo recabada por el adversario, por ejemplo, números de tarjeta de crédito y su código de seguridad, o un usuario y contraseña.

El spam y los spammers están evolucionando, y en muchos escenarios la incorporación de algoritmos de seguridad y soluciones tecnológicas no es suficiente, por lo que también es necesario culturizar a los usuarios y tomar medidas defensivas directas, como la denuncia de ciertos servidores de correo electrónico o IP de la cual se están recibiendo constantemente correos no deseados.

Dentro de este entorno donde la seguridad de los usuarios se ve ampliamente comprometida, surgen una serie de preguntas como saber ¿si se es víctima de spam?, ¿si el correo que he recibido no es *phishing*?, ¿si quien me dice enviar el correo, es legítimo? y ¿cómo saber todo esto sin ser un experto en informática? Todas ellas sin duda difíciles de contestar, sin embargo, en este trabajo se propone un sistema de verificación de correos electrónicos de fácil implementación para un proveedor de envío de correos, de fácil comprensión y utilización por parte de un usuario final. Dicho sistema debe agregar una validación extra a las ya existentes, en el sentido de evitar ser víctima de *phishing* y ser capaz de identificar un correo no legítimo, para su posterior denuncia.

I. ESTADO DEL ARTE

Las soluciones actuales que buscan combatir o al menos reducir los ataques de *phishing* están desarrolladas bajo los criterios de las mejores prácticas antiphishing para Proveedores de Servicio de Internet (ISPs por sus siglas en inglés) y proveedores de email del APWG (Anti-Phishing Working Group). Se presentan esquemas de mensajes de entrada para servidores de correo entrante o usuarios destinatarios y esquemas de mensajes de salida para servidores de correo saliente o usuarios remitentes. En los de entrada, existen diversas soluciones comerciales y gratuitas de filtrado, basados en filtros bayesianos, listas negras de direcciones IP, esquemas heurísticos y de huellas dactilares, filtros basados en URL, filtros del lado del cliente, entre otros [3]. En los de salida, existen filtros antiphishing que son instalados en los proveedores de servicio de internet que buscan analizar el tráfico de mensajes salientes de los sitios web que son alojados en ellos. Si se detecta contenido o mensajes phishing, el ISP puede bloquear o denegar el acceso estos sitios [3].

Existen dos tecnologías de autenticación de correo electrónico ampliamente aceptadas, la primera es un conglomerado de políticas para enviadores o SPF por sus siglas en inglés (Sender Policy Framework) y la segunda son llaves de dominio de un correo identificado o DKIM por sus siglas en inglés (DomainKeys Identified Mail). SPF y DKIM proveen métodos básicos de autenticación de nombres de dominio para correos electrónicos.

Uno de los precursores de estas técnicas fue PayPal, quien junto con Yahoo! Mail y más tarde Gmail, colaboraron para trabajar en un sistema para reducir ataques de spam y *phishing*. El problema, es que esta es una tecnología de hace más de una década y ahora no resulta del todo viable [4].

En un esfuerzo reciente de la industria se diseñó una capa adicional basada en autenticación, llamada Autenticación de mensaje basado en dominio, reporte y conformidad o DMARC por sus siglas en inglés (Domain-based Message Authentication, Reporting & Conformance). Permite a un remitente indicar que sus correos electrónicos están protegidos por SPF y/o DKIM, y le dice a un destinatario qué hacer si ninguno de estos métodos de autenticación verificó; también proporciona un mecanismo de reporte, por parte de los remitentes o enviadores a los dueños o proveedores de dominios [5, 6].

Dos grandes firmas que han implementado esta tecnología son Gmail y Twitter. Para Gmail como servidor de correo electrónico, esto ha significado reducir altamente la tasa de correo spam recibido y por ende no permitir que correos de servidores no autenticados caigan en las bandejas de los usuarios [7, 8, 9, 10]. Para Twitter, como enviador o remitente de correos, ha significado minimizar el riesgo de mal uso de su marca, enviando mensajes altamente verificados con los servidores de correo electrónicos [11].

Ambas empresas han implementado dichos cambios hace casi ya un año, y a pesar de que la industria ha estado trabajando en estándares de autenticación de correos electrónicos, todavía no se cuenta con una solución completa y funcional. Por lo que aún existen varios retos que solucionar, tales como:

- Muchos servidores de correo electrónico y servidores de nombres de dominio no han implementado dicha tecnología, por lo que phishers aún pueden fácilmente atacar dominios sin protección.
- Aunque se implementen estándares antiphishing, los atacantes pueden atentar o quebrar llaves criptográficas débiles.
- Las políticas son publicadas únicamente vía DNS, lo que representa una potencial amenaza, ya que una mala configuración podría significar múltiples consultas DNS por mensaje, convirtiéndose en un ataque por denegación de servicio por DNS [5].
- Si un mensaje autenticado o no, logra pasar las políticas o filtros del servidor destinatario, y llega a la bandeja del usuario destinatario común (sin conocimientos técnicos), éste no tendrá una forma de verificar la autenticidad del mensaje.
- Las evaluaciones están basadas únicamente en el RFC5322.From y no se realiza análisis de contenido; es decir, que un servidor autenticado puede enviar contenido similar en forma y fondo al contenido de otro servidor; convirtiéndose en un “phishing con servidores autenticados”.

El objetivo a lograr con esta propuesta es crear una capa de verificación (opcional) del origen y autenticidad de un mensaje de correo electrónico del lado del cliente, de fácil implementación del proveedor remitente o enviador y fácil manejo del usuario destinatario, proveyendo un mecanismo de verificación posterior a los implementados en la capa de autenticación basada en dominios con DMARC. Dicho mecanismo brindará una forma visual y práctica para determinar si un correo electrónico recibido (autenticado o no en la capa DMARC) es de la persona o institución que dice ser.

II. DEFINICIONES Y NOMENCLATURA

Phishing

Es el acto de suplantar la identidad de un remitente *r1* por parte de un remitente malintencionado *r2*, mediante el envío de un mensaje *m* de longitud *n*, con contenido similar al utilizado por el remitente original *r1*, dirigido a un destinatario *d*.

Código

Es una cadena de caracteres alfanuméricos *s* de longitud definida *l* la cual calculamos como $l \leftarrow |s|$. Dicha cadena es distinguible tanto por una computadora como por un humano.

Captcha

Es una imagen o representación gráfica *g* de un código *s*, generado a partir de una función G, el cual definimos $g \leftarrow G(s)$. Dicha imagen según el estado del arte de la tecnología de reconocimiento de imágenes, es únicamente distinguible por un humano [12, 13].

Credenciales de conexión

Son identificadores entregados por una entidad registradora y validadora de una entidad usuaria que consume un servicio de la primer entidad. Dichos identificadores son utilizados para verificar la validez e identidad de quien solicita la apertura de un canal de comunicación.

Entidad registradora y validadora

Se encarga de registrar y validar los datos de un usuario. Mediante un sitio web ofrece un formulario para que los usuarios ingresen sus datos y posteriormente valida su identidad. Para la propuesta, esta entidad será representada por el emisor de códigos, estudiado más adelante.

Token

Es una cadena de caracteres alfanuméricos k de longitud definida n la cual calculamos como $n \leftarrow |k|$, con tiempo de duración t, la cual representa un identificador entre dos participantes en una comunicación protocolar.

III. MECANISMO DE COMUNICACIÓN

La solución propuesta en este artículo requiere la participación de tres entidades: remitente, emisor de códigos, destinatario. Las cuales se definen a continuación:

Remitente

Es el proveedor de envío de correos electrónicos, ya sea de forma masiva o individual. Este es el encargado de armar el mensaje[1], solicitar el código de verificación al emisor de códigos y enviarlo al destinatario.

Emisor de Códigos

El emisor de códigos o EC, es el proveedor del servicio de validación. Esta entidad se considera confiable y no corruptible. Este recibe peticiones de generación de códigos de verificación, genera los códigos a partir de las peticiones, y los devuelve a la entidad solicitante. También le permite a la entidad destinatario verificar la validez, origen y contenido de un mensaje ligado a un código previamente provisto.

Destinatario

Es quien recibe el mensaje proveniente del remitente, el cual deberá contener el código de verificación previamente generado por el EC. Este podrá verificar la validez, el origen y el contenido del mensaje que recibió, a partir del código incluido en el mensaje mismo. Dicha validación la realizará directamente con el EC.

Las líneas de comunicación son bidireccionales entre el remitente y el EC, y el destinatario y el EC, a diferencia de entre el remitente y el destinatario, la cual es unidireccional. El destinatario, en concepto, no puede interactuar con el remitente, ya que no tiene la necesidad de intercambiar mensajes, al menos al nivel de la capa que se trata en este artículo.

La comunicación ocurre en tres fases, entre tres pares de actores. En la primera fase, a la cual denominamos negociación de código, participan el remitente y el EC[2]. En la segunda

[1] Cabe destacar que el remitente no es necesariamente el creador del contenido del mensaje, ya que puede ser un usuario final el encargado para el caso de un mensaje de correo electrónico entre dos usuarios finales. En caso de ser una comunicación directa del remitente sin intervención de un usuario o con representación de la entidad remitente, el contenido sí puede ser creado por él mismo.

fase, a la cual denominamos envío de mensaje, participan el remitente y el destinatario, y en la última y tercera fase, a la cual denominamos verificación del mensaje, participan el destinatario y el EC.

Fase 1 – Negociación de código

El objetivo de esta fase es obtener un código de validación ligado a un mensaje a ser enviado. Quien inicia esta fase es el remitente con solicitud al EC. Esta fase comprende dos interacciones entre las partes; la primera es la validación de identidad y la segunda es la solicitud de código.

Validación de identidad

Consiste en el envío de sus credenciales de conexión del remitente al EC, las cuales deben ser verificadas para entablar una conexión. Dicha conexión se realiza satisfactoriamente, con la devolución de un token[3] de conexión por parte del EC.

Solicitud de código

Consiste en el envío por parte del remitente al EC, del contenido que llevará el mensaje que será enviado al destinatario posteriormente. El EC mediante un *Protocolo de Creación de Códigos* (que se estudia más adelante), genera un código de validación y lo devuelve al remitente, para que este sea añadido como parte del mensaje al destinatario.

Fase 2 – Envío de mensaje

El objetivo de esta fase es enviar el mensaje al destinatario, con su contenido y añadiendo el código de verificación previamente negociado con el EC. El añadir el código al mensaje no debe representar un costo alto agregado al ya estimado del envío de un mensaje ordinario. Del lado del destinatario, el mensaje es recibido y almacenado. La estructura del mensaje contempla el mensaje original más el código.

---

[2] Para poder entablar una comunicación válida entre el remitente y el EC, es necesario que exista un registro y validación previos por parte del remitente en el sistema del EC. Dicho registro y validación debe proveer de credenciales de conexión al remitente con las que se identificará y que el EC verificará en la negociación de código.

[3] La generación del token de conexión no se contempla como parte de la solución, pero puede tomarse como modelo el mismo protocolo C estudiado más adelante sin reducción de hash, o puede implementarse cualquier protocolo de generación de tokens.

Fase 3 – Verificación del mensaje

El objetivo de esta fase es verificar a través de un mecanismo asíncrono al envío del mensaje, la validez, el origen y la integridad del contenido del mismo, a partir de su código. La forma en que el destinatario realiza la verificación se da en tres pasos:

Reconocimiento y Extracción del Código

El destinatario debe reconocer y extraer fácilmente el código recibido en el mensaje y luego proporcionarlo al EC, para que este haga la búsqueda del mismo en su base de códigos previamente generados.

Validación del Código

Si el EC puede corroborar la existencia del código, este le muestra la información del origen y el contenido del mensaje ligado a ese código. En este punto, lo único de lo que se puede estar seguro, es de la existencia y validez del código, mas no de la fiabilidad del mensaje; es decir, el hecho de que un código sea válido, no significa que el mensaje sea válido.

Validación del mensaje

Esta segunda validación es realizada con ayuda de la entidad EC, al regenerar el código a partir del mensaje que el destinatario recibió. Finalmente el destinatario, puede comparar ambos códigos, en caso de que sean exactamente iguales, se puede decir que el correo verifica y es legítimo; en caso contrario, es decir, si el mensaje genera un código diferente, se encuentra ante un ataque de phishing. Como validación extra, el destinatario puede verificar la información del remitente mediante la información registrada con el EC, y que éste último le proporciona.

Las tres fases están representadas en el diagrama de interacciones de la figura 1.

IV. ESTRUCTURA DE BASE DE DATOS

El EC debe almacenar en su base de datos, los datos del remitente que obtuvo en el registro. Los datos de las credenciales de conexión asignadas al remitente. Todas las conexiones establecidas entre el remitente y el EC. Los datos básicos de todos los mensajes procesados por el remitente, y por último los códigos generados por mensaje,

junto con el hash del mismo y la cadena utilizada en el protocolo para generar cada código. El mensaje del remitente no es necesario almacenarlo. El diagrama de la base de datos puede observarse en la figura 2.

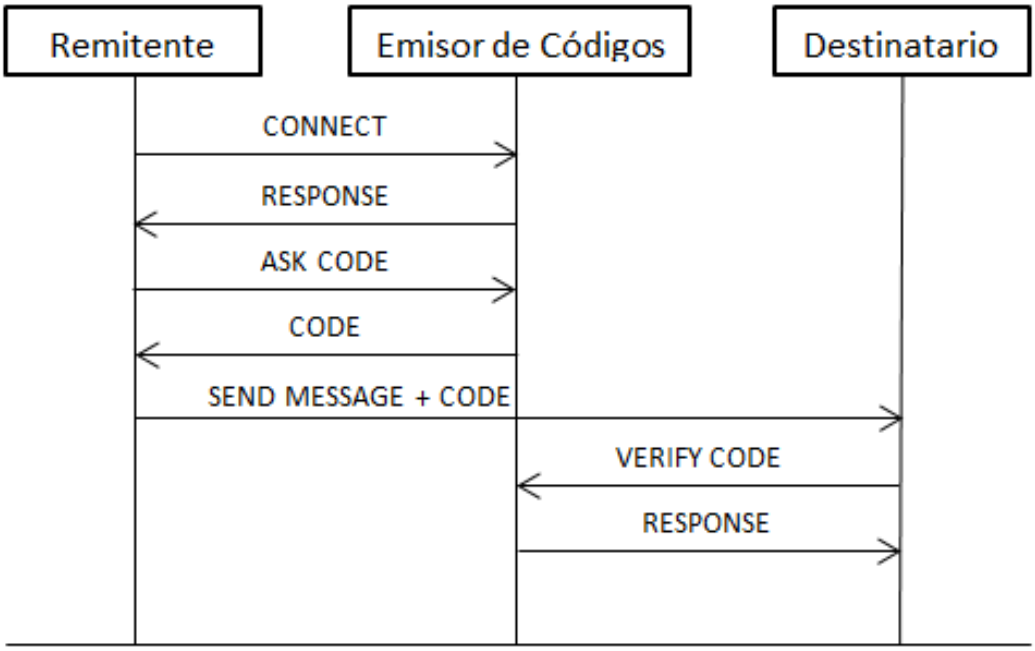

Figura 1 – Mecanismo de comunicación

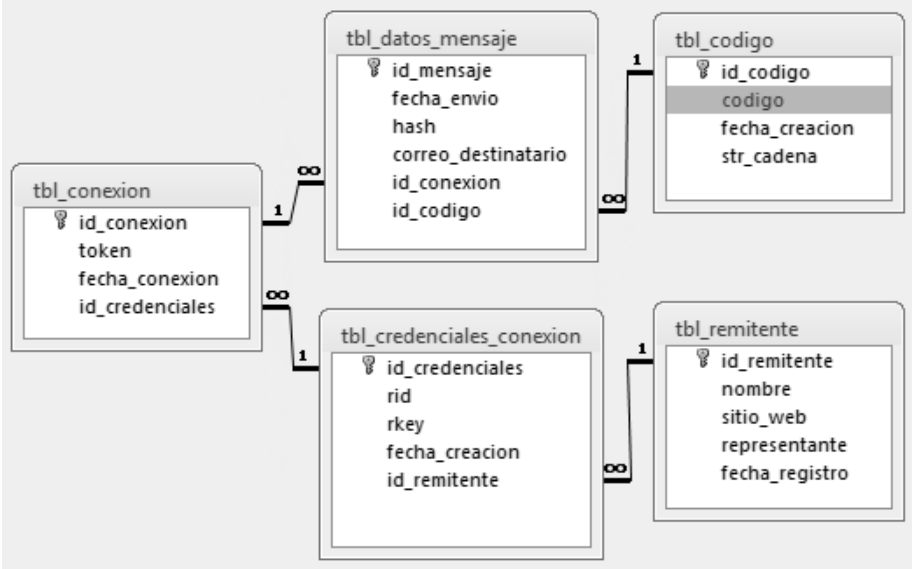

Figura 2 – Estructura de base de datos

V. PROTOCOLO DE CREACIÓN DE CÓDIGOS

Un Código de Verificación es una cadena binaria generada por el EC, a petición del remitente, el cual está directamente relacionado a un mensaje a ser enviado a un destinatario. Dicho código es un identificador del mensaje, el cual puede ser verificado posteriormente, mientras tenga vigencia. El código es la parte más importante del protocolo y del proceso de comunicación entre el remitente, el EC y el destinatario; de hecho es lo que los relaciona. Este en sí, es la solución propuesta en este artículo al problema del *phishing*, ya que un código generado por el EC y añadido a un mensaje por parte del remitente y recibido en el mensaje por parte del destinatario, le brindará la validación extra que necesita éste último para verificar la validez del origen y del mensaje mismo. A este código le llamamos *Código de Verificación Antiphishing* o *CVA*.

Para la generación del código a partir de una petición de un remitente pre-registrado y validado por el EC, se deberá seguir un protocolo *C* con las siguientes primitivas:

- El espacio de mensajes *M*.
- El conjunto de códigos *CVA*.
- Una función time $T : 0 \rightarrow t$ asociada al tiempo UNIX.
- Un conjunto *STR* el cual consiste de cadenas cortas de un alfabeto especificado.
- Una función de concatenación $D : \{ M', STR', t \} \rightarrow d$
  Esta función concatena el mensaje original, la cadena del alfabeto y el tiempo UNIX.
- Una función hash $H : d \rightarrow h$
- Una función $MAC : M' \rightarrow mac$
- Una llave *k* que solo conoce el EC.

Dado un mensaje $x \in M$, el protocolo *C* produce un código como se muestra en la figura 3.

Ya tenemos definido el protocolo para la generación de códigos de verificación a partir de un mensaje recibido. Para cada mensaje, será un código diferente, el cual se ve afectado principalmente por el mensaje original, la cadena del alfabeto y el tiempo del momento diferencial en el que se compute la función.

Protocolo C(x)

1. t ← T( );
2. s ← STR;
3. d ← D(x, s, t);
4. h ← H(d);
5. c ← MAC_k(h);
6. return c;

Figura 3 – El protocolo C

VI. SERVICIOS DE SEGURIDAD A BRINDAR

Integridad

Mediante el digesto del mensaje junto con el CVA, se está garantizando al momento de realizar el proceso de verificación con el EC, que el mensaje es el original generado con ese CVA y que no ha sido modificado.

Autenticación

Este servicio es prácticamente el objetivo del mecanismo y del protocolo, ya que lo que se busca es garantizar que un mensaje recibido sea de una entidad validada y es quien dice ser; así como se busca poder identificar un correo no válido o un remitente falso.

El éxito de la propuesta se basa en cumplir este servicio en específico, y gracias a la combinación de un mecanismo que permite la comunicación entre partes debidamente identificadas y validadas, y un protocolo que permite la creación de códigos de verificación válidos, se logra ofrecer un sistema de entidades debidamente autenticadas en más de una capa.

Una de ellas es la capa de seguridad ofrecida por *SSL v3.0*, otra es la capa de inicio de sesión al entablar la comunicación por webservice brindada por *SOAP*, otra es la capa de conexión brindada por la verificación en base de datos al momento de intentar una función *connect* por parte del remitente con el EC.

A esto le aunamos el servicio de verificación brindado por el EC, para que el destinatario pueda verificar la autenticidad del correo electrónico recibido mediante la validación del código recibido. Esta última fase es manual y totalmente opcional, pero es en sí donde se logra constatar el funcionamiento y validez de la propuesta presentada en este artículo, ya que el usuario destinatario podrá verificar 3 cosas. Primero, la validez del código recibido; segundo, la validez del remitente del correo electrónico; y tercero, la validez del contenido recibido.

No repudio

Este servicio está únicamente del lado del remitente, es decir, no repudio en el origen, ya que la propuesta está orientada a garantizar el origen del correo electrónico recibido, más no la recepción satisfactoria del mismo. Al recibir un correo electrónico que contiene un código de verificación válido, este tuvo que haber sido solicitado por un remitente validado y autenticado, por lo que el mensaje lleva el factor de no repudio hacia el remitente, el cual debe hacerse cargo de dicho mensaje enviado.

VII. CASOS DE PRUEBA DEL DESTINATARIO

Caso 1 – Correo sin CVA o caso nulo

El destinatario recibe un correo electrónico sin CVA y la verificación no es posible. Puede solicitarse al remitente hacer uso de CVAs para verificación.

Caso 2 – Correo con CVA válido

1. El destinatario recibe un correo electrónico con CVA.
2. Identifica el CVA.
3. Entra al sitio web del EC.
4. Ingresa el CVA.
5. El CVA es válido.
6. Verifica la información del remitente.
7. La información del remitente es correcta.
8. El correo electrónico es correctamente validado.

Caso 3 – Correo con CVA válido pero no único

1. El destinatario recibe un correo electrónico con CVA.
2. Identifica el CVA.
3. Entra al sitio web del EC.
4. Ingresa el CVA.
5. El CVA es válido pero ha sido utilizado en más de un correo.
6. El destinatario ingresa su correo electrónico y fecha de recibido.
7. Los datos ingresados son correctos.
8. El destinatario verifica la información del remitente.
9. La información del remitente es correcta.
10. El correo electrónico es correctamente validado.

Caso 4 – Correo con CVA no válido

1. El destinatario recibe un correo electrónico con CVA.
2. Identifica el CVA.
3. Entra al sitio web del EC.
4. Ingresa el CVA.
5. El código no existe o no es válido.
6. El correo electrónico no se da por validado.
7. El destinatario reporta el remitente.

Caso 5 – Phishing de CVA

1. El destinatario recibe un correo electrónico con CVA.
2. Identifica el CVA.
3. Entra al sitio web del EC.
4. Ingresa el CVA.
5. El CVA es válido.
6. El destinatario verifica la información del remitente.
7. La información del remitente no es correcta, pertenece a otro remitente.
8. El correo electrónico no se da por validado.
9. El destinatario reporta el CVA y el remitente.

Caso 6 – Reutilización o uso incorrecto de CVA

1. El destinatario recibe un correo electrónico con CVA.
2. Identifica el CVA.
3. Entra al sitio web del EC.
4. Ingresa el CVA.
5. El CVA es válido.
6. El destinatario verifica la información del remitente.
7. La información del remitente es correcta.
8. Se verifica el contenido mediante el cálculo del digesto y regeneración de código.
9. El contenido es diferente al recibido.
10. El correo electrónico no se da por validado.
11. El destinatario reporta el CVA y el remitente.

Caso 7 – Correo con múltiples CVAs

1. El destinatario recibe un correo electrónico con más de un CVA.
2. Identifica los CVA.
3. Entra al sitio web del EC.
4. Realiza el proceso de validación individual para cada CVA.
5. Si uno de los CVA no es válido
   5.1 El correo electrónico no se da por validado.
   5.2 El destinatario reporta el CVA y al remitente.
6. Si el CVA seleccionado o todos los CVA son válidos, el correo electrónico es correctamente validado.

VIII. ILEGIBILIDAD DEL CÓDIGO DE VERIFICACIÓN

Uno de los propósitos en temas de seguridad, es la ilegibilidad que puede llegar a tener un mensaje entre dos humanos, para una máquina o robot. El estado del arte en cuanto a reconocimiento de imágenes se refiere, no le permite a una máquina diferenciar a partir de una representación gráfica o imagen el texto que dibuja.

Nuestro código de verificación, sin una conexión segura o sin ninguna función de cifrado aplicada, puede ser interceptado, difundido e incluso falsificado. Para evitar esto y en

buscas de contrarrestar el *phishing* producto de métodos automatizados o manuales, nos valemos de la ilegibilidad aplicada a nuestro código. Para ello modificaremos nuestro protocolo y le agregaremos una función captcha, la cual generará una imagen o representación gráfica de nuestro código, permitiendo su identificación y lectura únicamente por un humano. Para ello agregaremos la siguiente entidad:

- Una función generadora de CAPTCHA G que tome como entrada nuestro código c.

Es decir, que dado un mensaje $x \in M$, el protocolo *C* con ilegibilidad aplicada produce un código como se muestra en la figura 4.

Vale aclarar que la robustez del CAPTCHA generado por nuestra función G le dará mayor complejidad al problema de obtención del código de verificación, es decir, entre más ofuscado esté el código en la imagen generada, más complicada será su lectura.

Protocolo C(x)

1. $t \leftarrow T(\ )$;
2. $s \leftarrow STR$;
3. $d \leftarrow D(x, s, t)$;
4. $h \leftarrow H(d)$;
5. $c \leftarrow MAC_k(h)$;
6. $c' \leftarrow G(c)$;
7. return c';

Figura 4 – El protocolo C con ilegibilidad aplicada

Este nivel de robustez puede ser regulado de acuerdo a las configuraciones establecidas en la misma función G o a solicitud del remitente.

IX. INDEPENDENCIA DEL MECANISMO DE COMUNICACIÓN CON EL PROTOCOLO DE CREACIÓN DE CÓDIGOS

La solución propuesta consta de estos dos elementos, el mecanismo de comunicación entre las partes ya estudiadas; y el protocolo de creación de códigos. Para ambos casos,

se proponen soluciones totalmente independientes, es decir, que una no se vea afectada por la otra, no obstante, se requieren de las dos para resultar en una solución funcional.

La independencia que muestran estos elementos se debe entender como la posibilidad de mejorarlos o inclusive reemplazarlos por otras propuestas, particularmente en el protocolo de creación de códigos de verificación; es decir, la solución utilizando el mecanismo de comunicación propuesto es capaz de soportar la integración de diferentes protocolos de códigos de verificación. Fácilmente puede adaptarse un protocolo diferente que responda al problema de generar un identificador del mensaje del remitente.

De necesitarse o para fines académicos, puede implementarse nuevos o ya existentes protocolos de creación de códigos y probarse en la infraestructura definida para el mecanismo de comunicación. Independientemente los algoritmos aplicados, la solución será la misma.

X. ÚLTIMAS OBSERVACIONES

A pesar del nivel de seguridad que puede representar un código de verificación en forma de CAPTCHA, esto puede generar un nivel de complejidad del lado del lector [14] (por ejemplo, personas con discapacidades visuales o navegadores con fallas de interpretación de imágenes), por lo que puede considerarse una solución de captchas o códigos de verificación compartidos. Esta forma de solución sugiere el envío de múltiples códigos a un mismo destino o a diferentes destinos, como parte de una validación múltiple obligatoria u opcional.

XI. CONCLUSIONES

En este artículo se ha planteado una solución que permite verificar de la identidad de un remitente de correo electrónico y la integridad del contenido recibido, mediante la creación, emisión y envío de Códigos de Verificación Antiphishing. Para ello, es necesario contar con un mecanismo de comunicación entre tres entidades, el remitente, el emisor de códigos y el destinatario. Además, se requiere de la implementación de un protocolo de creación de códigos con ilegibilidad aplicada. Este código debe añadirse como parte del

mensaje del correo electrónico. El usuario destinatario de forma opcional puede tomar este código y el mensaje recibidos e ir al sistema del emisor de códigos y verificar de forma sencilla la procedencia del mismo, el remitente y el contenido. Con esto se logra tener una capa extra de verificación antiphishing de bajo coste para el remitente y fácil verificación para el destinatario.

REFERENCIAS

[1] Cisco, «2013 Cisco Annual Security Report,» 2013.

[2] Cisco, «2014 Cisco Annual Security Report,» 2014.

[3] Messaging Anti-Abuse Working Group, Anti-Phishing Working Group, «Anti-Phishing Best Practices for ISPs and Mailbox Providers,» 07 2006. [En línea]. Available: http://docs.apwg.org/reports/bestpracticesforisps.pdf. [Último acceso: 30 12 2014].

[4] DMARC, «DMARC - Overview,» [En línea]. Available: http://www.dmarc.org/overview.html. [Último acceso: 30 12 2014].

[5] D. Crocker, «Using DMARC draft-crocker-dmarc-bcp-03,» 09 05 2014. [En línea]. Available: http://www.ietf.org/archive/id/draft-crocker-dmarc-bcp-03.txt. [Último acceso: 30 12 2014].

[6] E. Z. M. Kucherawy, «Domain-based Message Authentication, Reporting and Conformance (DMARC) draft-kucherawy-dmarc-base-09,» 26 12 2014. [En línea]. Available: https://datatracker.ietf.org/doc/draft-kucherawy-dmarc-base/?include_text=1. [Último acceso: 30 12 2014].

[7] E. Protalinski, «Google says 91.4% of non-spam emails sent to Gmail users are now authenticated using antiphishing standards,» 06 12 2013. [En línea]. Available: http://thenextweb.com/google/2013/12/06/google-says-91-4-authenticated-non-spam-emails-sent-gmail-users-now-using-antiphishing-standards/. [Último acceso: 30 12 2014].

[8] Google, «Add digital signatures with DKIM,» [En línea]. Available:

https://support.google.com/a/answer/174124?hl=en. [Último acceso: 30 12 2014].

[9] Google, «Authorize senders with SPF,» [En línea]. Available: https://support.google.com/a/answer/33786. [Último acceso: 30 12 2014].

[10] Google, «Prevent outgoing spam with DMARC,» [En línea]. Available: https://support.google.com/a/answer/2466580. [Último acceso: 30 12 2014].

[11] J. A, «Introducing DMARC for Twitter.com emails,» 21 02 2013. [En línea]. Available: https://blog.twitter.com/2013/introducing-dmarc-for-twittercom-emails. [Último acceso: 30 12 2014].

[12] L. v. Ahn, «Using Hard AI Problems For Security,» 2003. [En línea]. Available: http://www.captcha.net/captcha_crypt.pdf. [Último acceso: 25 10 2014].

[13] D. C. Sandra Díaz-Santiago, «On Securing Communication From Profilers,» [En línea]. Available: https://eprint.iacr.org/2012/124.pdf. [Último acceso: 25 10 2014].

[14] E. Bursztein, «How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation,» 2010. [En línea]. Available: http://www.computer.org/cms/Computer.org/ComputingNow/homepage/2013/0113/captchas.pdf. [Último acceso: 25 10 2014].