Informe Laboratorio 1
Fundamentos de Seguridad Informática

Marco Centurión, Juan Andrés Friss

4 de abril de 2016

Resumen

1. Práctica 1

Esta práctica apunta a mostrar la importancia de la encriptación de los datos sensibles en las comunicaciones por red. Para esto se realizó un ataque Man in the Middle, demostrando lo fácil que puede ser para un atacante tener acceso a la información que se transmite por un canal no seguro.

Realización de la Práctica

La herramienta que se utilizó en esta práctica es Ettercap ('a comprehensive suite for man in the middle attacks'); la misma provee de muchas funcionalidades y variados ataques, de los cuales utilizamos la variante ARP poisoning.

Este ataque infecta la caché ARP de las víctimas con el fin de que los paquetes sean enviados al atacante en lugar de a sus destinos originales. Luego de recibir los paquetes el atacante puede modificar y forwardear los mismos, aunque para nuestros fines la modificación no es necesaria.

Luego de ingresar al laboratorio virtual y observar que la VM estaba conectada a dos redes, teniendo la conexión a la red de ataque en su interfaz eth1, se confeccionó y ejecutó el siguiente comando:

```
sudo ettercap -T -i eth1 -M arp:remote /10.0.20.8/ /10.0.20.4/
```

El mismo comenzó a capturar los paquetes enviados entre los hosts indicados y a desplegarlos en pantalla. Se reconocieron varios de los mensajes intercambiados entre las víctimas (se observó un handshake TCP, por ejemplo) de los cuales el que resaltaba era el siguiente, enviado al puerto 7 del servidor (donde habitualmente corre el *echo protocol*):

```
Fri Mar 25 15:47:06 2016
TCP  10.0.20.4:56378 --> 10.0.20.8:7 | AP

<begin message>
QmVob2xkLCB0aGUgZm9vbCBzYWl0aCwgIlB1dCBub3QgYWxsIHRoaW5lIGVnZ3Mg
aW4gdGhlIG9uZSBiYXNrZXQiLS13aGljaCBpcwpidXQgYSBtYW5uZXIgb2Ygc2F5
aW5nLCAiU2NhdHRlciB5b3VyIG1vbmV5IGFuZCB5b3VyIGF0dGVudGlvbjsiIGJ1
dCB0aGUgd2lzZQptYW4gc2FpdGgsICJQdXQgYWxsIHlvdXIgZWdncyBpbiB0aGUg
b25lIGJhc2tldCBhbmQtLVdBVENIIFRIQVQgQkFTS0VULiIKCQktLSBNYXJrIFR3
YWluLCAiUHVkZCduaGVhZCBXaWxzb24ncyBDYWxlbmRhciIK
<end message>
```

El mensaje parecía no estar encriptado, sino codificado en base64, por lo cual se utilizó *base64* para decodificar el mensaje y obtener en un formato ASCII el mensaje enviado por la víctima:

```
Behold, the fool saith, "Put not all thine eggs in the one
   basket"--which is
but a manner of saying, "Scatter your money and your attention;"
   but the wise
man saith, "Put all your eggs in the one basket and--WATCH THAT
   BASKET."
     -- Mark Twain, "Pudd'nhead Wilson's Calendar"
```

Conclusiones

En base a la experiencia de realizar esta práctica se llegó a las siguientes conclusiones:

Realizar un ataque MitM resulta relativamente sencillo gracias a las herramientas que se encuentran disponibles al público, sin costo alguno y con muy poco o nulo expertise. Esto hace importante que se tomen medidas para mitigar este tipo de ataques ya que al ser tan sencilla la realización de los mismos las probabilidades de ser víctima de uno aumenta considerablemente.

Es importante comprender la distinción entre cifrar y codificar: una confusión de este estilo en la programación de una aplicación que maneje datos sensibles podría tener resultados catastróficos. Esto no significa que la codificación no tenga sus usos; en particular base64 se puede utilizar para representar datos binarios en ASCII y así poder transmitirlos de una manera sencilla por canales que no se podrían utilizar de otra forma.

Para mitigar los riesgos de un ataque MitM no alcanza tampoco con simplemente encriptar ya que el atacante podría estar interceptando los mensajes por un largo tiempo y comprometer todo el proceso de intercambio de claves. Es por esto que la implementación de sistemas como PKI resultan muy importantes a la hora de mantener la privacidad y seguridad en la red.

2. Práctica 2

Realización de la Práctica

Esta práctica apunta a aplicar los conocimientos adquiridos en las clases teóricas sobre criptografía asimétrica. En esta práctica se utilizaron dos herramientas: GPG y OpenSSL. GPG es un programa de criptografía de clave pública que sirve para firmar y cifrar datos. Se genera un par de claves (pública/privada) para el usuario. OpenSSL es un conjunto de herramientas criptográficas que posee una librería criptográfica de propósito general. En primer lugar se trabajó con la herramienta GPG. Se comenzó por generar un par de claves para el grupo

```
gpg --gen-key
```

Luego se cifró el archivo paracifrar.txt utilizando la clave pública de los docentes contenida en el archivo fsi-pub.asc

```
gpg --import fsi-pub.asc
gpg --compress-algo none --output paracifrar.gpg --encrypt
   --recipient fsi@fing.edu.uy
```

Se recurrió al flag "–compress-algo none" ya que el proceso de compresión causaba que el cifrado llevara un tiempo excesivamente largo y, al ser el archivo relativamente pequeño, se consideró que la compresión no era una prioridad.

A continuación se exportó la clave pública generada para el grupo, utili-

zando la opción armor para obtener un formato más amigable para el intercambio:

```
gpg --armor --export marco.centurion@fing.edu.uy --output
   fsi20-pub.asc
```

Por último se firmó la clave pública de los docentes y se le asignó un nivel de confianza.

```
gpg --edit-key fsi@fing.edu.uy
trust
3
gpg --sign-key fsi@fing.edu.uy
gpg --armor --export fsi@fing.edu.uy --output fsi-pub-signed.asc
```

Se decidió utilizar un nivel marginal de confianza ya que si bien se sabe que los docentes comprenden las consecuencias de firmar una clave, no tenemos experiencia previa con firmas de los mismos, por lo que no podemos darles confianza completa.

En segundo lugar se trabajó con la herramienta OpenSSL. Se comenzó por generar y exportar un par de claves RSA para el grupo

```
openssl genrsa -out key.pem 2048
openssl rsa -in key.pem -pubout -out fsi20-pub.pem
```

Por último se generó un Certificate Signing Request

```
openssl req -new -key key.pem -out fsi20.csr
```

Este archivo sería lo que enviaríamos a una CA si quisieramos que la misma nos certificara. El mismo, además de estar firmado con nuestra clave privada y contener la clave pública, contiene información acerca del solicitante como puede ser nombre, título, entidad, página web, etc.

Conclusiones

Uno de los objetivos de aprendizaje de esta práctica consiste en entender las diferencias entre criptografía asimétrica y simétrica. Cuando se utiliza criptografía simétrica, se tienen algoritmos que requieren de solo una clave que se utiliza tanto para encriptar y desencriptar. El problema principal que se tiene es cómo compartir esta clave de manera segura entre las dos partes que se comunican.

Para resolver esta problemática se tiene la criptografía asimétrica. En ella se tienen dos claves distintas para cifrar y descifrar, "fáciles"de generar, pero es computacionalmente imposible obtener la de descifrado a partir de la de cifrado. Por lo tanto, la clave de cifrado se puede hacer pública (y por esto es que este tipo de criptografía se conoce como criptografía de clave pública). Un problema que presentan este tipo de algoritmos es que son más pesados y lentos y suelen requerir de más recursos del sistema, y entonces se tarda más tanto en encriptar como en desencriptar.

Es por esto que en la práctica generalmente se utilizan procedimientos híbridos, donde se utiliza criptografía de clave pública para poder realizar un intercambio del cual surge una clave simétrica que luego se utiliza para encriptar el resto de la comunicación. De este modo se soluciona el problema de intercambiar una clave simétrica pero mantenemos la simplicidad computacional en el resto de la comunicación.

La criptografía asimétrica tiene varios modelos de confianza mediante los cuales se busca poder afirmar la autenticidad de las claves obtenidas. De estos los más utilizados hoy en día son PKI (Public Key Infrastructure) y el modelo Web of Trust (el utilizado por GPG). PKI se basa en la existencia de entidades que se encargan de verificar la identidad de los usuarios y luego utilizar su propia firma para certificar que una clave pública efectivamente pertenece a dichos usuarios. Este modelo es susceptible a ataques DoS, ya que si cae la entidad certificadora se puede dificultar o imposibilitar la verificación de los certificados.

GPG, por otro lado, utiliza el modelo Web of Trust que se asemeja bastante a lo que son las relaciones interpersonales humanas: yo confío en Alice, Alice tiene cierto grado de confianza en Bob y en base a eso yo decido si confiar en Bob y en las personas en las que confíe Bob. Basado en estos principios, en GPG un usuario puede firmar la clave de otro usuario y de esta forma asignarle un cierto grado de confianza:

- unknown: el estado inicial de las claves añadidas al keyring. No se puede afirmar nada acerca del dueño de la clave.
- none: se sabe que el usuario ha firmado incorrectamente claves.
- marginal: se sabe que el usuario comprende las implicaciones que conlleva firmar una clave y que valida las mismas antes de hacerlo.
- full: se sabe que el usuario tiene una excelente comprensión del proceso de firma, tanto así que la firma del mismo es tan válida como la propia.

En base a estos niveles se decide en quién confiar, considerando una clave válida si cumple las siguientes condiciones:

1. La clave fue firmada por una cantidad suficiente de personas:
    - Fue firmada por una persona en la que tengo confianza plena.
    - Fue firmada por al menos tres personas en las cuales tengo confianza marginal.
2. El camino de claves firmadas entre el usuario y yo es de largo 5 o menor

Otro aspecto importante a aprender es la diferencia entre cifrar y firmar. Firmar digitalmente se puede entender como la firma escrita: se utiliza para garantizar nuestra identidad como autores del envío. Firmar un mensaje digitalmente no aporta seguridad en cuanto a que el contenido no sea visto por terceros; aporta la seguridad para quien lo recibe de que ha sido enviado por quien se dice que lo envió. Por el otro lado, cifrar significa ocultar o convertir la información a una forma "disfrazada", para ser transmitida a través de un canal inseguro y de esta manera que sólo el destinatario real del mensaje pueda leerlo.

Ejemplos prácticos del uso de esta herramienta pueden ser todos los vistos en la práctica, desde generar pares de claves para encriptar o desencriptar mensajes o archivos hasta firmar documentos. Igualmente es importante notar que la herramienta GPG es en general más utilizada para comunicaciones entre personas. Por el otro lado, la herramienta OpenSSL es generalmente utilizada para garantizar la seguridad en servidores o sitios web incluyendo los certificados que se usan.

3. Práctica 3

Esta práctica apunta a concientizar acerca de las vulnerabilidades asociadas al algoritmo de hashing MD5 y las consecuencias que estas vulnerabilidades ocasionan.

Realización de la Práctica

En primer lugar se calcularon los valores del hash MD5 y SHA-1 para cada uno de los archivos con el objetivo de obtener una rápida comparación entre

los mismos. Para esto se ejecutaron los siguientes comandos y se obtuvieron las salidas indicadas:

```
md5sum carta.ps -- A25F7F0B29EE0B3968C860738533A4B9
md5sum hack.ps -- A25F7F0B29EE0B3968C860738533A4B9
sha1sum carta.ps -- 07835FDD04C9AFD283046BD30A362A6516B7E216
sha1sum hack.ps -- 3548DB4D0AF8FD2F1DBE02288575E8F9F539BFA6
```

Claramente si utilizaramos como prueba de integridad el hash MD5 estaríamos en problemas ya que los dos archivos aparentan ser distintos, si nos guiamos por el SHA-1.

A continuación se utilizó un visor de postscript, y se pudo ver claramente que los archivos contienen textos distintos: uno elogiando a un empleado y proponiéndolo para cargos de mayor importancia y el otro indicando que dicho empleado puede tener acceso a cierta información confidencial y secreta.

Luego se utilizó un editor de texto, donde se ve el código PS, y se obervaron ciertas diferencias en algunos carácteres cercanos al cabezal, aunque el resto de ambos archivos parecen ser idénticos, incluyendo ambos archivos los dos textos que se pueden visualizar con el visor. El lugar donde se observaron diferencias parece ser un string escrito dos veces, con una leve diferencia en el caso del archivo 'hack.ps'.

Por último se ejecutó un diff para comprobar si ambos archivos son efectivamente diferentes:

```
diff carta.ps hack.ps
```

y se obtuvo como resultado "Binary files carta.ps and hack.ps difer", por lo que podemos afirmar con total seguridad (si es que la diferencias observadas previamente no alcanzaran) que los archivos son diferentes.

Conclusiones

Con la realización de esta práctica se lograron sacar varias conclusiones:

Este ataque fue posible debido a que postscript es un lenguaje que permite incluir elementos que pueden desplegarse o no dependiendo de condiciones booleanas. De este modo se comparan al comienzo dos strings y dependiendo se su igualdad se muestra un mensaje o el otro; esto, sumado a una vulnerabilidad de MD5 que se describirá luego, permite tener dos documentos que varían sólo en los string que se comparan y generan el mismo hash MD5 pese a generar distintos documentos cuando se visualizan con un visor de PS.

Investigando se llegó al origen de este ataque, denominado "Blind Passenger Attack"por Lucks y Daum, y se encontraron las debilidades que se utilizaron para el mismo:

- En primer lugar se utilizó la debilidad de MD5 frente a colisiones para encontrar dos string distintos que, combinados con el preámbulo del archivo, dieran el mismo hash: $MD5(preambulo.R_1) = MD5(preambulo.R_2)$.
- Luego se utilizó otra debilidad conocida que ocasiona que, dados los $X_i = preambulo.R_i$ del item anterior, la concatenación de los mismos generen el mismo hash: $MD5(X_1||S) = MD5(X_2||S)$

De esta forma se generó el preámbulo y el string S de modo que se compare el X_i colocado con X_1 y dependiendo del resultado de esa operación se muestre un mensaje o el otro pese a que los hash MD5 de ambos archivos son iguales.

Luego de analizar el ataque presentado se reflexionó acerca de la importancia de la elección de los algorimos de hashing a utilizar en las aplicaciones de firma digital. Las firmas que utilicen algoritmos para los cuales se puedan encontrar colisiones fácilmente o para los cuales se encuentran debilidades como la segunda indicada previamente son susceptibles a ataques como el descripto y no pueden considerarse seguras.

Finalmente se resalta la importancia de mantenerse informado acerca de los avances en los campos relacionads con la seguridad informatica de modo de poder realizar decisiones informadas al momento de elegir algoritmos y procedimientos a utilizar cuando se estén llevando a cabo actividades relacionadas con información sensible o confidencial.