



ANÁLISIS FORENSE INFORMÁTICO

Tarea 1

ESTHER CARRILLO GÁLVEZ

Contenido

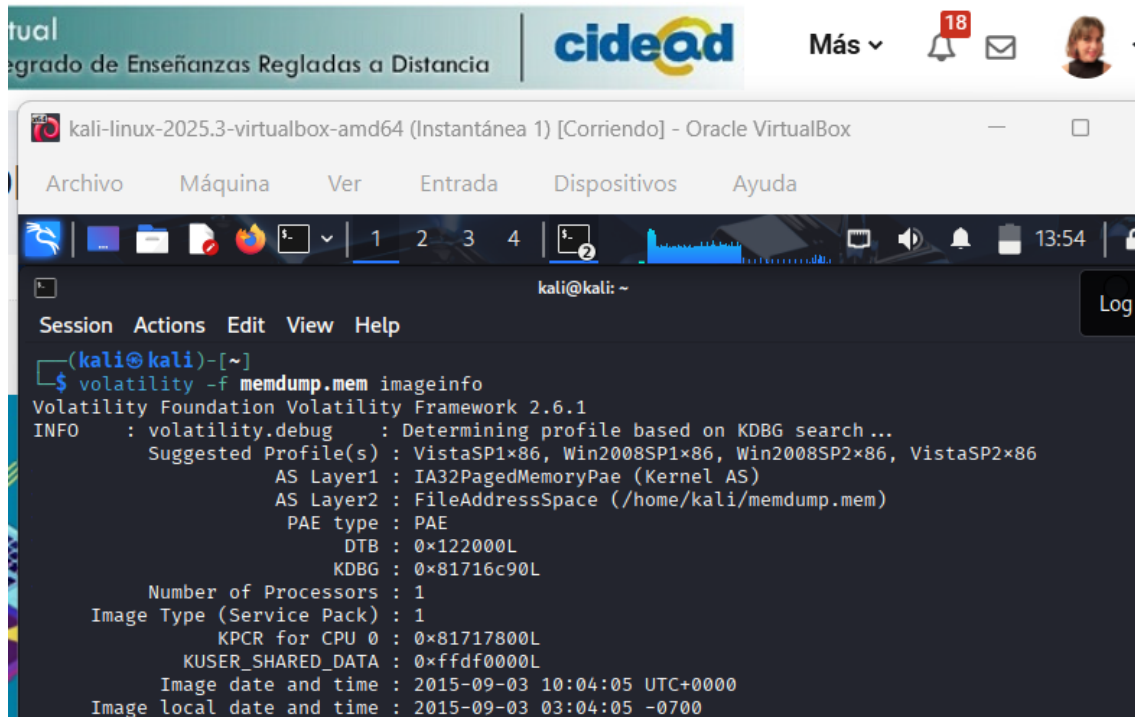
¿QUÉ PASARÍA SI SE HUBIERA APAGADO ESTE SERVIDOR?.....	3
¿QUÉ TIPO DE COMANDOS HA EJECUTADO EL CIBERCRIMINAL? ¿QUÉ SUGIERE?	3
¿CÓMO SE HAN EJECUTADO LOS COMANDOS?	8
¿QUÉ ACTIVIDAD MALICIOSA HAS VISTO?.....	9
¿PUEDES IDENTIFICAR DESDE QUE IP VINO EL ATAQUE?	14
¿QUÉ TIPO DE ATAQUE PUDO SER? ¿QUÉ TIPO DE MALWARE SE HA ENCONTRADO?	15
BIBLIOGRAFÍA	16

¿QUÉ PASARÍA SI SE HUBIERA APAGADO ESTE SERVIDOR?

La memoria RAM es volátil esto significa que necesita energía eléctrica para retener los datos. Si se hubiera apagado el servidor, toda la información de *memdump.mem* se hubiera perdido, así como las conexiones de red, las listas de procesos y cualquier información contenida en ella. Además de haber alterado los archivos temporales o haber alterado los *timestamps*. Por tanto, si se hubiera apagado no habría datos en la memoria RAM y, por consiguiente, no se habría conseguido su imagen.

¿QUÉ TIPO DE COMANDOS HA EJECUTADO EL CIBERCRIMINAL? ¿QUÉ SUGIERE?

Para determinar el tipo de comandos que ha ejecutado el cibercriminal, se deberá “interpretar” los datos binarios de la imagen de la memoria RAM mediante la herramienta Volatility. A continuación, mediante el plugin *imageinfo*, se determinará el sistema operativo que usó el cibercriminal desde el equipo del que se extrajo la memoria RAM, en este caso *Windows VistaSP1x86*.



```
kali@kali: ~  
$ volatility -f memdump.mem imageinfo  
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/kali/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 : 0xffdf0000L  
Image date and time : 2015-09-03 10:04:05 UTC+0000  
Image local date and time : 2015-09-03 03:04:05 -0700
```

Un plugin importante de Volatility es *pstree* que muestra una lista jerárquica de los procesos del sistema operativo, indicando los procesos padres y los procesos hijos.

ula Virtual

ntro Integrado de Enseñanzas Regladas a Distancia

cidead

Más ▾

18

kali-linux-2025.3-virtualbox-amd64 (Instantánea 1) [Corriendo] - Oracle VirtualBox

Archivo

Máquina

Ver

Entrada

Dispositivos

Ayuda

1

2

3

4

5

14:00

kali@kali: ~

Session

Actions

Edit

View

Help

ArmAddressSpace: No valid DTB found

(kali@kali)-[~]

\$ volatility -f memdump.mem --profile=VistaSP1x86 pstree

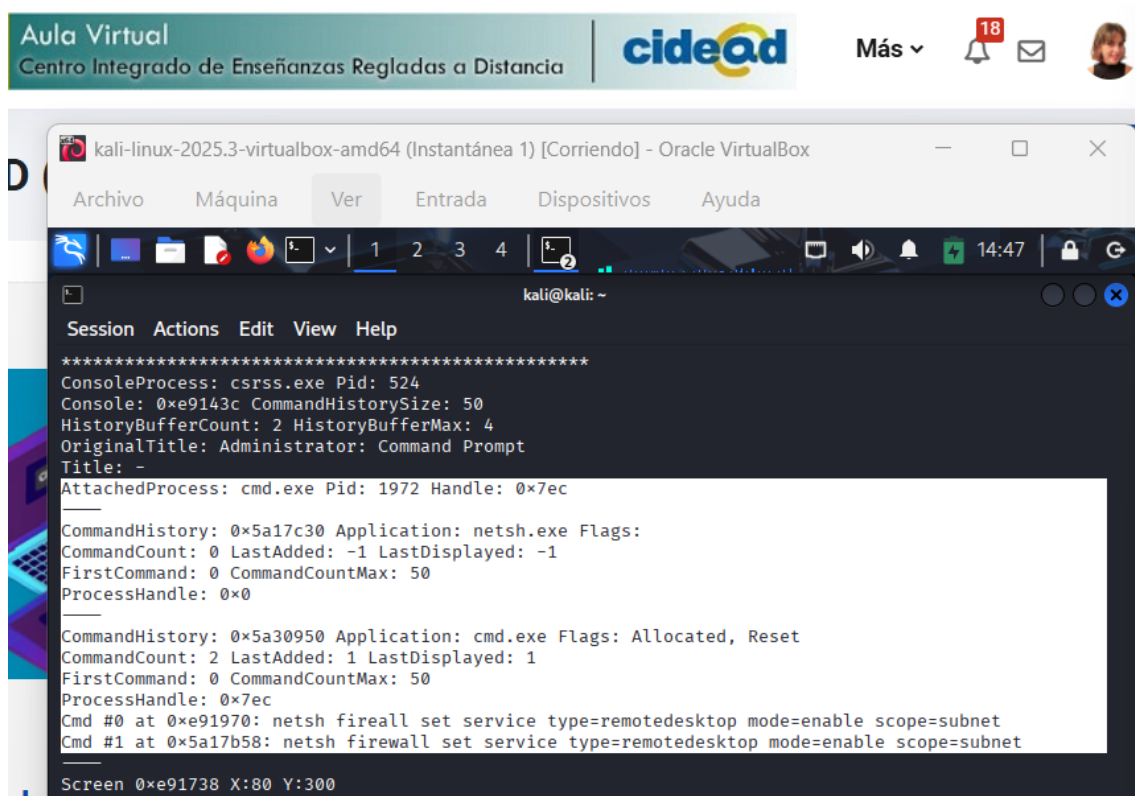
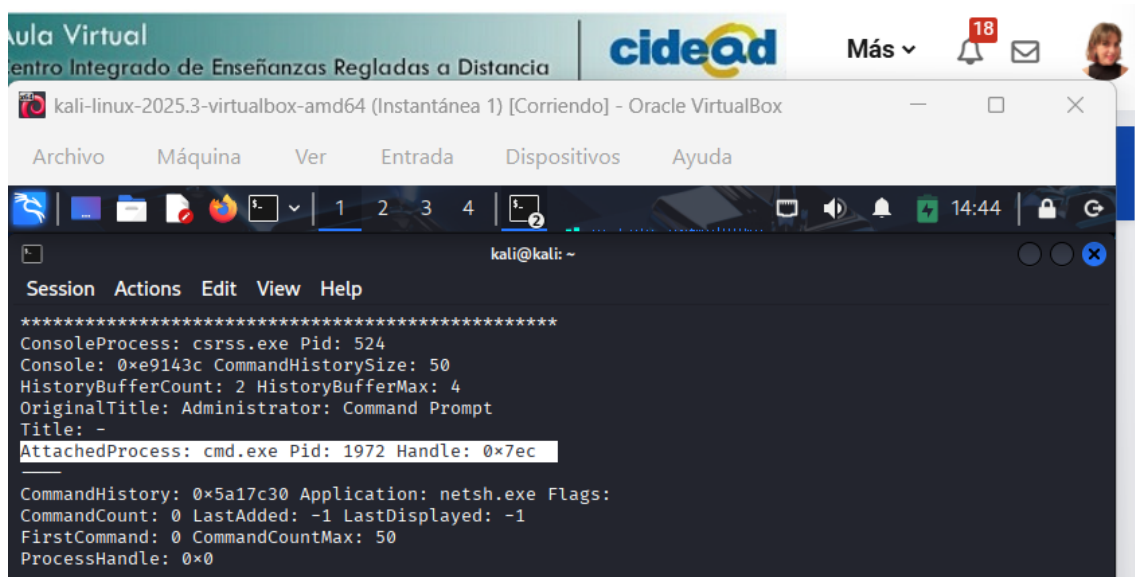
Volatility Foundation Volatility Framework 2.6.1

Name	Pid	PPid	Thds	Hnds	Time
0x8392c9f8:wininit.exe	532	472	3	102	2015-08-23 20:27:28
UTC+0000					
. 0x8393bd90:services.exe	608	532	7	238	2015-08-23 20:29:06
UTC+0000					
.. 0x83a0eb88:svchost.exe	1024	608	37	913	2015-08-23 10:29:53
UTC+0000					
... 0x8427c730:wuauc.lt.exe	2516	1024	2	140	2015-09-02 09:01:13
UTC+0000					
... 0x83dca020:taskeng.exe	1984	1024	5	135	2015-08-23 10:30:08
UTC+0000					
... 0x83b2b020:taskeng.exe	1444	1024	10	245	2015-08-23 10:30:34
UTC+0000					
.. 0x8324cb70:TrustedInstalle	3848	608	5	110	2015-09-03 10:03:06
UTC+0000					
.. 0x83a1e020:SLsvc.exe	1040	608	4	75	2015-08-23 10:29:53
UTC+0000					
.. 0x83a365d0:svchost.exe	1176	608	22	257	2015-08-23 10:29:56
UTC+0000					
... 0x83e2f168:dwm.exe	1688	1176	3	77	2015-08-23 10:30:34
UTC+0000					
.. 0x839d4020:svchost.exe	792	608	8	305	2015-08-23 20:29:45
UTC+0000					
.. 0x839ded90:VBxService.exe	836	608	8	115	2015-08-23 20:29:46
UTC+0000					
.. 0x83ae6c28:svchost.exe	1568	608	3	73	2015-08-23 10:30:05
UTC+0000					
.. 0x83a3e020:svchost.exe	1204	608	18	518	2015-08-23 10:29:56
UTC+0000					
.. 0x83a18020:svchost.exe	1012	608	6	147	2015-08-23 10:29:53
UTC+0000					
.. 0x83f8e5d0:msdtc.exe	2620	608	11	165	2015-08-23 10:32:10
UTC+0000					

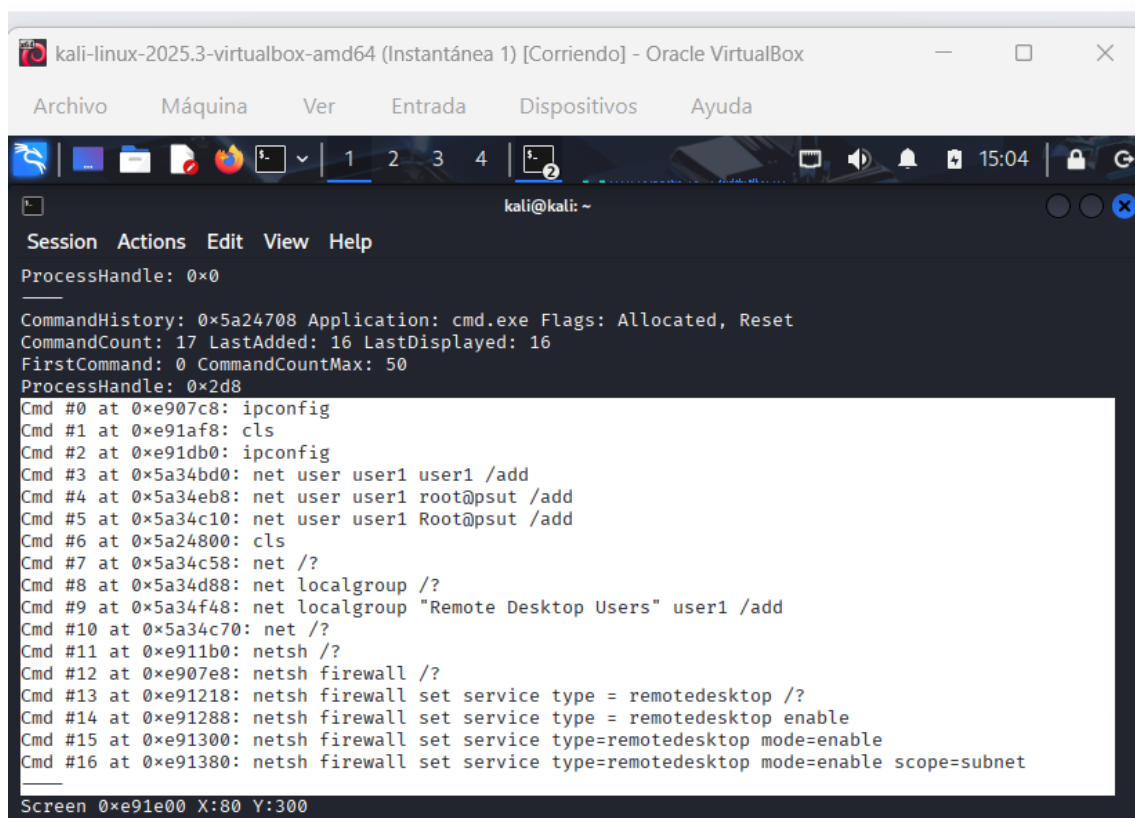
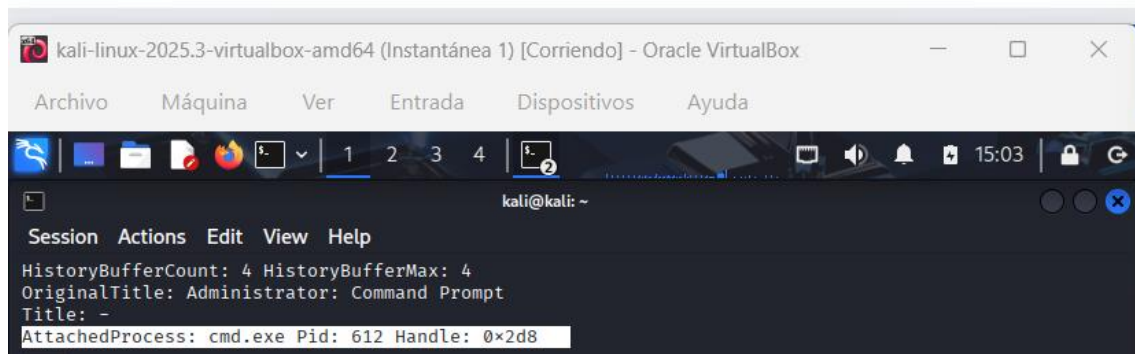
Se observan procesos sospechosos en la lista recogida en la imagen. Entre ellos se encuentran dos procesos *cmd.exe* cuyos PID son 612 y 1972, respectivamente.

Process Name	PID	PPID	Session	Architecture	Start Time	End Time
UTC+0000						
.. 0x83a06020:svchost.exe	984	608	15	306	2015-08-23 10:29:52	
UTC+0000						
.. 0x83af2d90:svchost.exe	1680	608	5	44	2015-08-23 10:30:05	
UTC+0000						
.. 0x83adfd90:svchost.exe	1512	608	9	117	2015-08-23 10:30:04	
UTC+0000						
.. 0x83f84d90:svchost.exe	2424	608	9	227	2015-08-23 10:31:51	
UTC+0000						
.. 0x83ae4af0:svchost.exe	1556	608	5	123	2015-08-23 10:30:05	
UTC+0000						
.. 0x839f0020:svchost.exe	892	608	7	262	2015-08-23 10:29:52	
UTC+0000						
.. 0x83942020:lsass.exe	620	532	19	628	2015-08-23 20:29:18	
UTC+0000						
.. 0x83945d90:lsm.exe	628	532	10	166	2015-08-23 20:29:19	
UTC+0000						
.. 0x83912208:csrss.exe	484	472	11	400	2015-08-23 20:27:22	
UTC+0000						
.. 0x83e368e0:explorer.exe	816	676	22	756	2015-08-23 10:30:34	
UTC+0000						
.. 0x83e652a0:VBoxTray.exe	1816	816	8	114	2015-08-23 10:30:38	
UTC+0000						
.. 0x83f68300:FTK Imager.exe	2120	816	13	382	2015-09-03 10:03:37	
UTC+0000						
.. 0x83faa020:xampp-control.e	2768	816	2	119	2015-08-23 10:32:17	
UTC+0000						
.. 0x83e4d7c0:httpd.exe	2796	2768	1	92	2015-08-23 10:32:21	
UTC+0000						
... 0x83fd77a8:httpd.exe	2880	2796	155	483	2015-08-23 10:32:26	
UTC+0000						
.. 0x83fd5200:FileZillaServer	2856	2768	5	35	2015-08-23 10:32:25	
UTC+0000						
.. 0x83f9ec70:mysqld.exe	2804	2768	23	570	2015-08-23 10:32:23	
UTC+0000						
.. 0x83e7b7f8:cmd.exe	612	816	1	72	2015-08-23 10:30:44	
UTC+0000						
.. 0x84259100:cmd.exe	1972	816	1	19	2015-09-02 09:28:30	
UTC+0000						

El plugin *consoles* combina la funcionalidad de *cmdscan* con información de los procesos de la consola. Es decir, extrae el historial de comandos ejecutados dentro de las consolas activas al momento del volcado. Se revela que dentro del proceso 1972 se ejecutaron comandos para abrir el firewall y permitir conexiones *remotedesktop*.



Así como también se ha encontrado evidencia de actividad maliciosa con la creación de un nuevo usuario *user1* con contraseña *user1* y es añadido al grupo *remotedesktop* para darle permisos en el firewall y acceder al escritorio remoto.



Con el plugin *hashdump* se extraen los hashes de las contraseñas de usuarios, entre los cuales se incluye el usuario hacker. Este usuario activa todas las alarmas ante actividad maliciosa.

```

kali@kali: ~
Session Actions Edit View Help

(kali@kali)-[~]
$ volatility --profile=VistaSP1x86 hashdump
Volatility Foundation Volatility Framework 2.6.1
Administrator:500:aad3b435b51404eeaad3b435b51404ee:63d6a39b8467b94ae92ab1931d4079dd:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
user1:1005:aad3b435b51404eeaad3b435b51404ee:817875ce4794a9262159186413772644:::
hacker:1006:aad3b435b51404eeaad3b435b51404ee:817875ce4794a9262159186413772644:::
  
```

¿CÓMO SE HAN EJECUTADO LOS COMANDOS?

El atacante ejecutó los comandos mediante el *cmd.exe* con privilegios de administrador, de modo que debió tener acceso al equipo víctima. Con el plugin *timeliner* se puede observar un ejemplo claro de en qué momento exacto se ejecutaron los procesos, como es en el caso del primer *cmd.exe* que se ejecutó el 23 de agosto de 2015 a las 10:30:44.

```

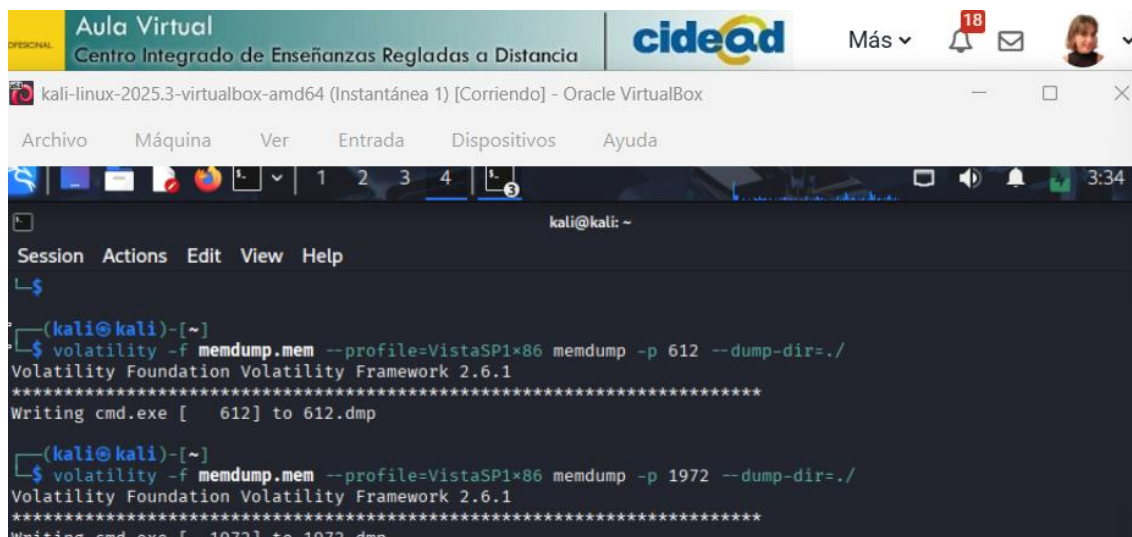
kali@kali: ~
Session Actions Edit View Help

2015-09-03 06:08:37 UTC+0000 [THREAD] | svchost.exe | PID: 1204/TID: 2116
2015-08-23 10:30:36 UTC+0000 [THREAD] | explorer.exe | PID: 816/TID: 788
2015-08-23 10:30:36 UTC+0000 [THREAD] | explorer.exe | PID: 816/TID: 216
2015-08-23 10:30:38 UTC+0000 [THREAD] | VBoxTray.exe | PID: 1816/TID: 272
2015-08-23 10:30:37 UTC+0000 [THREAD] | explorer.exe | PID: 816/TID: 732
2015-08-23 10:30:39 UTC+0000 [THREAD] | VBoxTray.exe | PID: 1816/TID: 1672
2015-09-03 10:02:57 UTC+0000 [THREAD] | explorer.exe | PID: 816/TID: 1332
2015-08-23 10:30:42 UTC+0000 [THREAD] | explorer.exe | PID: 816/TID: 1340
2015-08-23 10:30:39 UTC+0000 [THREAD] | VBoxTray.exe | PID: 1816/TID: 488
2015-08-23 10:30:39 UTC+0000 [THREAD] | VBoxTray.exe | PID: 1816/TID: 764
2015-08-23 10:30:39 UTC+0000 [THREAD] | VBoxTray.exe | PID: 1816/TID: 300
2015-08-23 10:30:39 UTC+0000 [THREAD] | VBoxTray.exe | PID: 1816/TID: 972
2015-08-23 10:30:39 UTC+0000 [THREAD] | svchost.exe | PID: 1176/TID: 1324
2015-08-23 10:30:43 UTC+0000 [THREAD] | svchost.exe | PID: 1024/TID: 1120
2015-08-23 10:30:43 UTC+0000 [THREAD] | explorer.exe | PID: 816/TID: 1196
2015-08-23 10:30:44 UTC+0000 [THREAD] | cmd.exe | PID: 612/TID: 348
2015-09-03 10:03:14 UTC+0000 [THREAD] | explorer.exe | PID: 816/TID: 4060
2015-08-23 10:30:43 UTC+0000 [THREAD] | explorer.exe | PID: 816/TID: 960
2015-08-23 10:30:47 UTC+0000 [THREAD] | explorer.exe | PID: 816/TID: 1072
2015-08-23 10:30:48 UTC+0000 [THREAD] | svchost.exe | PID: 1176/TID: 1100
  
```


¿QUÉ ACTIVIDAD MALICIOSA HAS VISTO?

Hasta ahora la actividad maliciosa que se ha observado es que el atacante mediante el *cmd.exe* y con privilegios de administrador, creó el usuario *hacker* con permisos y para abrir el firewall con conexiones remotas.

Pero para ir más allá, se analizarán los dos procesos *cmd* sospechosos vistos anteriormente con el plugin *memdump*.

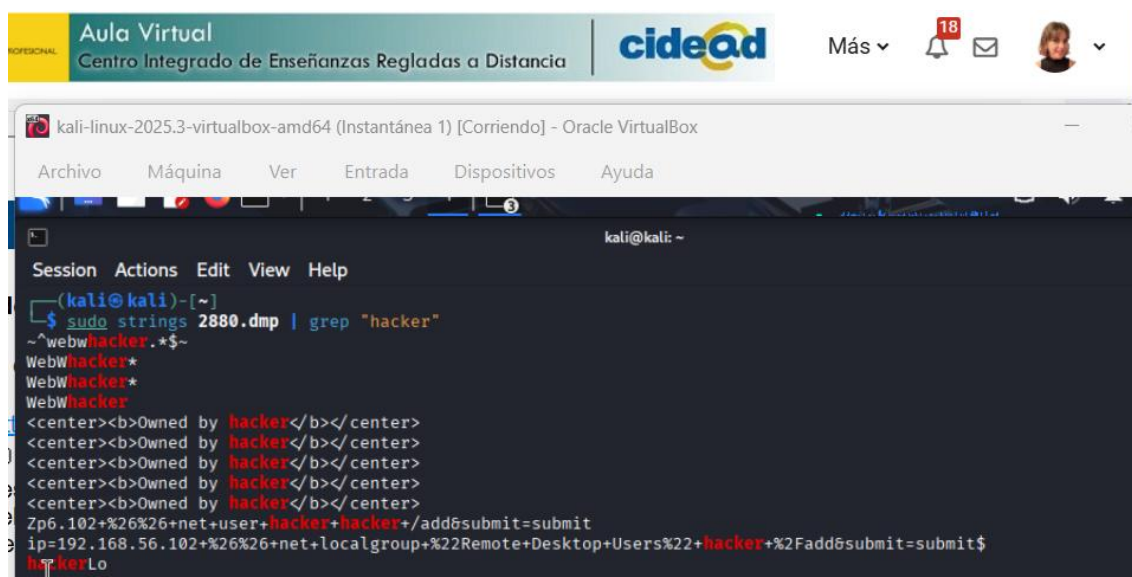


```
kali@kali: ~  
Session Actions Edit View Help  
~$ volatility -f memdump.mem --profile=VistaSP1x86 memdump -p 612 --dump-dir=.  
Volatility Foundation Volatility Framework 2.6.1  
*****  
Writing cmd.exe [ 612] to 612.dmp  
  
~$ volatility -f memdump.mem --profile=VistaSP1x86 memdump -p 1972 --dump-dir=.  
Volatility Foundation Volatility Framework 2.6.1  
*****  
Writing cmd.exe [ 1972] to 1972.dmp
```

A continuación, se ejecuta el plugin *strings* para extraer todas las cadenas de texto legible del proceso 2880 y que se identificó como *httpd.exe*. Con *grep hacker*, el programa filtra por las líneas que contienen la palabra *hacker*.

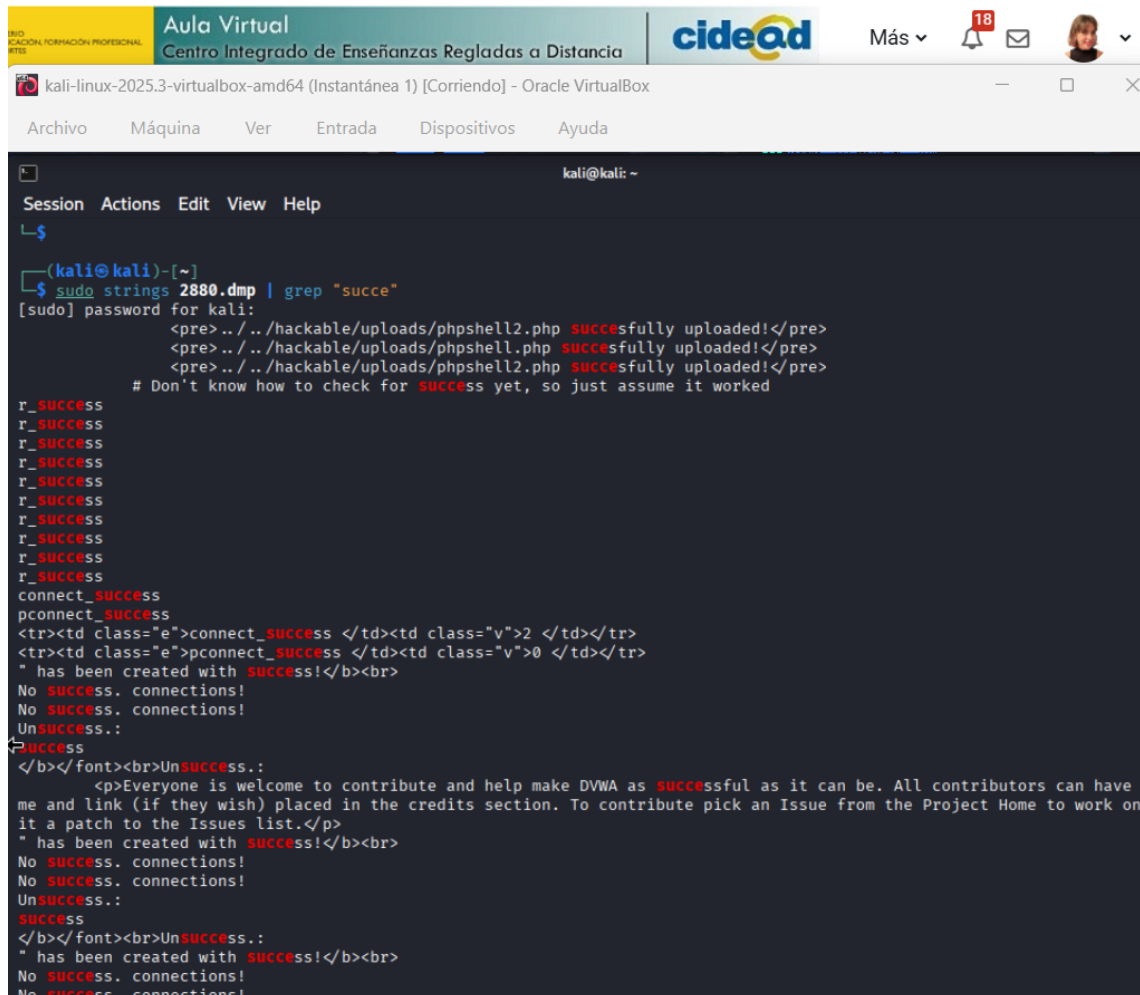
Nota: para los procesos 612 y 1972 no se encontró ninguna cadena.

Este resultado indica que el atacante utilizó el servidor web *httpd.exe* para el ataque desde una IP privada 192.168.56.102.



```
kali@kali: ~  
Session Actions Edit View Help  
~$ sudo strings 2880.dmp | grep "hacker"  
~^webWhacker.*$~  
WebWhacker*  
WebWhacker*  
WebWhacker*  
<center><b>Owned by hacker</b></center>  
<center><b>Owned by hacker</b></center>  
<center><b>Owned by hacker</b></center>  
<center><b>Owned by hacker</b></center>  
<center><b>Owned by hacker</b></center>  
Zp6.102+%26%26+net+user+hacker+hacker+add6submit=submit  
ip=192.168.56.102+%26%26+net+localgroup+%22Remote+Desktop+Users%22+hacker+%2Fadd6submit=submit$  
hackerLo
```

Por consiguiente, para comprobar si el servidor web devolvió algún mensaje de éxito después de que el atacante ejecutara sus comandos, se busca la cadena *succe*. Definitivamente el atacante subió un webshell *phpshell2.php* al servidor web. De modo que con este script, consigue abrir una consola de comandos remota.



```

Aula Virtual
Centro Integrado de Enseñanzas Regladas a Distancia
cideon
Más 18
kali-linux-2025.3-virtualbox-amd64 (Instantánea 1) [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
kali@kali: ~
Session Actions Edit View Help
$ sudo strings 2880.dmp | grep "succe"
[sudo] password for kali:


```

..../hackable/uploads/phpshell2.php successfully uploaded!


```

..../hackable/uploads/phpshell2.php successfully uploaded!


```

..../hackable/uploads/phpshell2.php successfully uploaded!
Don't know how to check for success yet, so just assume it worked
r_success
r_success
r_success
r_success
r_success
r_success
r_success
r_success
r_success
r_success
connect_success
pconnect_success
connect_success	2
pconnect_success	0

" has been created with success!
No success. connections!
No success. connections!
Unsuccess.:
success

Unsuccess.:
<p>Everyone is welcome to contribute and help make DVWA as successful as it can be. All contributors can have
me and link (if they wish) placed in the credits section. To contribute pick an Issue from the Project Home to work on
it a patch to the Issues list.</p>
" has been created with success!
No success. connections!
No success. connections!
Unsuccess.:
success

Unsuccess.:
" has been created with success!
No success. connections!
No success. connections!

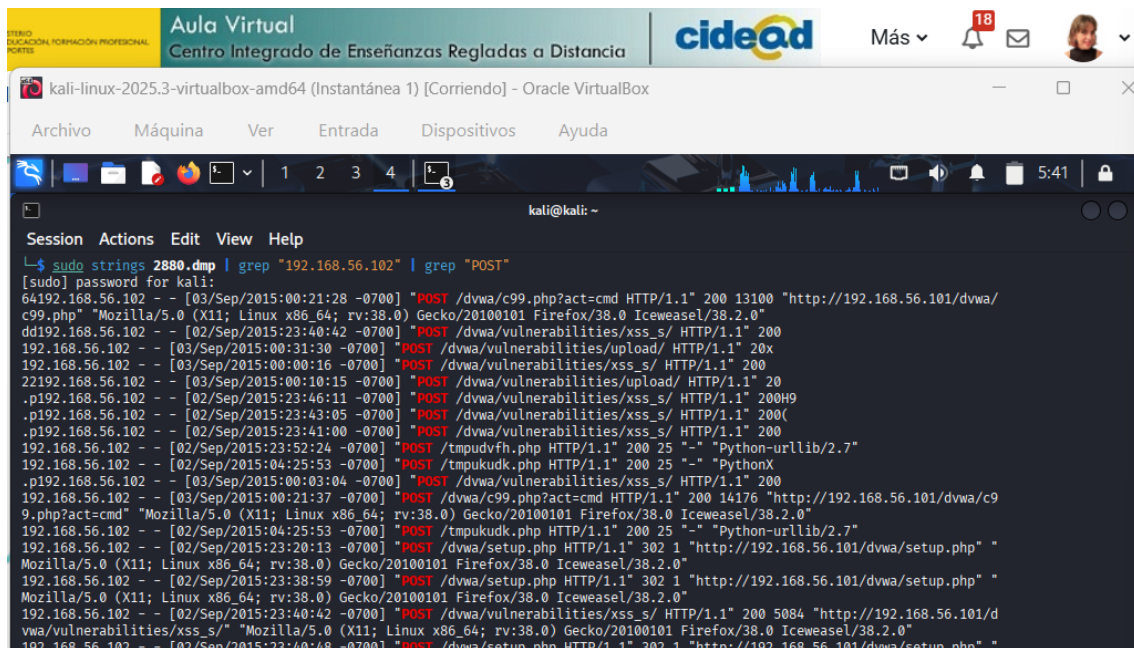
```


```


```

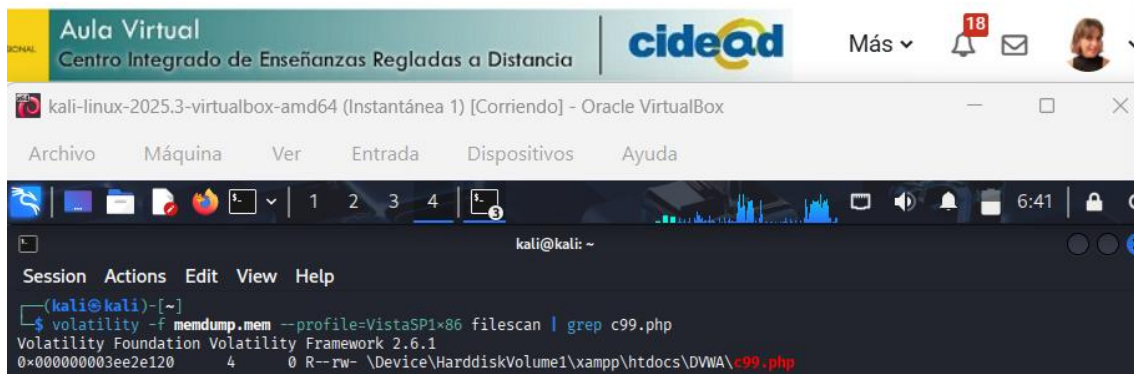

```

Una vez localizada la IP desde la que atacó el cibercriminal, se filtra por la cadena *192.168.56.102* para rastrear su actividad en el servidor web. Además, se utiliza el método *POST* para enviar datos desde el cliente al servidor y comprobar que el atacante utilizó su dirección IP para enviar al servidor *httpd.exe* comandos y exploits.



```
Session Actions Edit View Help
$ sudo strings 2880.dmp | grep "192.168.56.102" | grep "POST"
[sudo] password for kali:
64192.168.56.102 -- [03/Sep/2015:00:21:28 -0700] "POST /dvwa/c99.php?act=cmd HTTP/1.1" 200 13100 "http://192.168.56.101/dvwa/c99.php" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedove/38.2.0"
dd192.168.56.102 -- [02/Sep/2015:23:40:42 -0700] "POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1" 200
192.168.56.102 -- [03/Sep/2015:00:31:30 -0700] "POST /dvwa/vulnerabilities/upload/ HTTP/1.1" 20x
192.168.56.102 -- [03/Sep/2015:00:00:16 -0700] "POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1" 200
22192.168.56.102 -- [03/Sep/2015:00:10:15 -0700] "POST /dvwa/vulnerabilities/upload/ HTTP/1.1" 20
.p192.168.56.102 -- [02/Sep/2015:23:46:11 -0700] "POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1" 200H9
.p192.168.56.102 -- [02/Sep/2015:23:43:05 -0700] "POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1" 200(
.p192.168.56.102 -- [02/Sep/2015:23:41:00 -0700] "POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1" 200
192.168.56.102 -- [02/Sep/2015:23:52:24 -0700] "POST /tmpudvfh.php HTTP/1.1" 200 25 "-" Python-urllib/2.7"
192.168.56.102 -- [02/Sep/2015:04:25:53 -0700] "POST /tmpukudk.php HTTP/1.1" 200 25 "-" PythonX
.p192.168.56.102 -- [03/Sep/2015:00:03:04 -0700] "POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1" 200
192.168.56.102 -- [03/Sep/2015:00:21:37 -0700] "POST /dvwa/c99.php?act=cmd HTTP/1.1" 200 14176 "http://192.168.56.101/dvwa/c99.php?act=cmd" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedove/38.2.0"
192.168.56.102 -- [02/Sep/2015:04:25:53 -0700] "POST /tmpukudk.php HTTP/1.1" 200 25 "-" Python-urllib/2.7"
192.168.56.102 -- [02/Sep/2015:23:20:13 -0700] "POST /dvwa/setup.php HTTP/1.1" 302 1 "http://192.168.56.101/dvwa/setup.php" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedove/38.2.0"
192.168.56.102 -- [02/Sep/2015:23:38:59 -0700] "POST /dvwa/setup.php HTTP/1.1" 302 1 "http://192.168.56.101/dvwa/setup.php" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedove/38.2.0"
192.168.56.102 -- [02/Sep/2015:23:40:42 -0700] "POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1" 200 5084 "http://192.168.56.101/dvwa/vulnerabilities/xss_s/" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedove/38.2.0"
192.168.56.102 -- [02/Sep/2015:23:40:48 -0700] "POST /dvwa/setup.php HTTP/1.1" 302 1 "http://192.168.56.101/dvwa/setup.php" "
```

Se observa que el atacante ha interactuado con la puerta trasera haciendo una petición *POST* contra el fichero *c99.php*. Mediante un *filescan | grep c99.php* se confirma que el archivo *c99.php* se subió a la aplicación vulnerable DVWA dentro de XAMPP.



```
Session Actions Edit View Help
(kali@kali)-[~]
$ volatility -f memdump.mem --profile=VistaSP1x86 filescan | grep c99.php
Volatility Foundation Volatility Framework 2.6.1
0x000000003ee2e120 4 0 R--rw- \\Device\\HarddiskVolume1\\xampp\\htdocs\\DVWA\\c99.php
```

Se vuelve a ejecutar el plugin *strings* filtrando por *?cmd=* para revelar la interacción entre el atacante y la web shell.

Aula Virtual
Centro Integrado de Enseñanzas Regladas a Distancia

cidead

Más 18

kali-linux-2025.3-virtualbox-amd64 (Instantánea 1) [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

6:58

```
kali@kali: ~  
Session Actions Edit View Help  
$ volatility -f memdump.mem --profile=VistaSP1x86 filescan | grep c99.php  
Volatility Foundation Volatility Framework 2.6.1  
0x000000003ee2e120 4 0 R--rw- \Device\HarddiskVolume1\xampp\htdocs\DVWA\c99.php  
  
(kali@kali)-[~]  
$ sudo strings 2880.dmp | grep "192.168.56.102" | grep "?cmd="  
[sudo] password for kali:  
/p192.168.56.102 - - [02/Sep/2015:04:26:23 -0700] "GET /tmpbiwuc.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpkudk.php HTTP/1.1" 200 11 "-" "sqlmap/1.0-dev-non  
192.168.56.102 - - [03/Sep/2015:00:17:49 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=dir%20C:\\ HTTP/1.1" 200 1934 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedove/38.2.0"  
192.168.56.102 - - [02/Sep/2015:23:59:38 -0700] "GET /tmpbrjvl.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpudvfh.php HTTP/1.1" 200 11 "-" "sqlmap/1.0-dev-non(I  
192.168.56.102 - - [03/Sep/2015:00:18:02 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=dir HTTP/1.1" 200 463 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedove/38.2.0"  
192.168.56.102 - - [03/Sep/2015:00:17:58 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=mkdir%20abc HTTP/1.1" 200 - "-" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedove/38.2.0"  
192.168.56.102 - - [02/Sep/2015:04:25:53 -0700] "GET /tmpbiwuc.php?cmd=echo%20command%20execution%20test HTTP/1.1" 200 36 "-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"  
192.168.56.102 - - [02/Sep/2015:23:59:38 -0700] "GET /tmpbrjvl.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpbrjvl.php HTTP/1.1" 200 11 "-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"  
192.168.56.102 - - [02/Sep/2015:04:25:53 -0700] "GET /tmpbiwuc.php?cmd=echo%20command%20execution%20test HTTP/1.1" 200 36 "-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"  
192.168.56.102 - - [02/Sep/2015:04:26:04 -0700] "GET /tmpbiwuc.php?cmd=dir HTTP/1.1" 200 853 "-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"  
192.168.56.102 - - [02/Sep/2015:04:26:23 -0700] "GET /tmpbiwuc.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpkudk.php HTTP/1.1" 200 11 "-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"  
192.168.56.102 - - [02/Sep/2015:04:26:23 -0700] "GET /tmpbiwuc.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpbiwuc.php HTTP/1.1" 200 11 "-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"  
192.168.56.102 - - [02/Sep/2015:23:52:24 -0700] "GET /tmpbrjvl.php?cmd=echo%20command%20execution%20test HTTP/1.1" 200 36 "-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"  
192.168.56.102 - - [02/Sep/2015:23:59:38 -0700] "GET /tmpbrjvl.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpudvfh.php HTTP/1.1" 200 11 "-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"  
192.168.56.102 - - [02/Sep/2015:23:59:38 -0700] "GET /tmpbrjvl.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpbrjvl.php HTTP/1.1" 200 11 "-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"  
192.168.56.102 - - [03/Sep/2015:00:16:13 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=dir HTTP/1.1" 200 419 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedove/38.2.0"  
192.168.56.102 - - [03/Sep/2015:00:17:49 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=dir%20C:\\ HTTP/1.1" 200 1934 "-" "
```

La salida muestra peticiones *GET* y *POST* provenientes de la IP del atacante. En las respuestas *GET* aparece la url *sqlmap.org* confirmando que el atacante utilizó la herramienta *sqlmap* para detectar y explotar vulnerabilidades de inyección SQL.

Aula Virtual
Centro Integrado de Enseñanzas Regladas a Distancia

cidead

Más ▾ 18

kali-linux-2025.3-virtualbox-amd64 (Instantánea 1) [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

```
kali@kali: ~  
Session Actions Edit View Help  
--$ volatility -f memdump.mem --profile=VistaSP1x86 filescan | grep c99.php  
Volatility Foundation Volatility Framework 2.6.1  
0x000000003ee2e120      4      0 R--rw- \Device\HarddiskVolume1\xampp\htdocs\DVWA\c99.php  
  
--(kali@kali)-[~]  
--$ sudo strings 2880.dmp | grep "192.168.56.102" | grep "?cmd="  
[sudo] password for kali:  
/p192.168.56.102 - - [02/Sep/2015:04:26:23 -0700] "GET /tmpbiwuc.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpukudk.ph  
b HTTP/1.1" 200 11 "-" "sqlmap/1.0-dev-non  
192.168.56.102 - - [03/Sep/2015:00:17:49 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=dir%20C:\\ HTTP/1.1" 200 1934 "-" "  
Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedevil/38.2.0"  
192.168.56.102 - - [02/Sep/2015:23:59:38 -0700] "GET /tmpbrjvl.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpudvfh  
.php HTTP/1.1" 200 11 "-" "sqlmap/1.0-dev-non(I  
192.168.56.102 - - [03/Sep/2015:00:18:02 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=dir HTTP/1.1" 200 463 "-" "Mozilla/  
5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedevil/38.2.0"  
192.168.56.102 - - [03/Sep/2015:00:17:58 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=mkdir%20abc HTTP/1.1" 200 - "-" "  
Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Icedevil/38.2.0"  
ne-Co192.168.56.102 - - [02/Sep/2015:04:25:53 -0700] "GET /tmpbiwuc.php?cmd=echo%20command%20execution%20test HTTP/1.1" 200 36 "  
-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"  
:0192.168.56.102 - - [02/Sep/2015:23:59:38 -0700] "GET /tmpbrjvl.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpbrjvl.php  
TP/1.1" 200 11 "-" "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"
```

Finalmente se vuelve a filtrar por la cadena *sqlmap* para revelar las peticiones *http* y se observan sentencias SQL de tipo *SELECT*, confirmando que el atacante realizó consultas contra el servidor *mysql*.

¿QUÉ TIPO DE ATAQUE PUDO SER? ¿QUÉ TIPO DE MALWARE SE HA ENCONTRADO?

El tipo de ataque se denomina SQL Injection. Se identificó un malware *phpshell2.php* y *c99.php* lo que permitió al atacante ejecutar comandos a través de un navegador web a través de un servidor vulnerable o comprometido.

BIBLIOGRAFÍA

Volatility Foundation. (s.f.). *Command Reference*. GitHub: Volatility Foundation.

Recuperado el 14 de diciembre de 2025, de

<https://github.com/volatilityfoundation/volatility/wiki/Command-Reference#pslist>

Ergo, Hackers. (2021, 8 de marzo). *VOLATILITY // Cómo hacer un análisis de memoria y determinar si una maquina esta infectada* [Video]. YouTube.

<http://www.youtube.com/watch?v=RFYbevW6hxl>

Behackerpro. (2021, 3 de septiembre). *Encontrando Secretos en la memoria del computador con Volatility Instalación No Standalone en Win10* [Video]. YouTube.

<http://www.youtube.com/watch?v=iU9mqB4h3Tg>

Hackavis. (2025, 6 de abril). *Como Usar Volatility para Analizar Memoria RAM en Windows. [Guía Completa]* [Video]. YouTube.

<http://www.youtube.com/watch?v=QeE6asCnD6E>

<https://github.com/volatilityfoundation/volatility/wiki/Command-Reference#pslist>