

Práctica 5. Pen Testing

Capture The Flag

Introducción

Antes de comenzar con el propio informe sobre el troyano Zeus, voy a realizar una breve explicación sobre la herramienta volatility, su utilidad, funcionamiento y como lo he instalado.

¿Qué es volatility?

Es una herramienta de código abierto que se enfoca principalmente en el análisis forense de memoria, y se utiliza para la respuesta a incidentes y análisis de malware. Está escrita en Python y es compatible en Windows, Mac OS X y Linux.

Y como ya he dicho sirve para la extracción de artefactos digitales de una memoria volátil como es en el caso de esta práctica “zeus.vmem”.

¿Sobre qué sistemas funciona?

Al inicio de esta práctica he intentado instalar la herramienta en Windows 11, y he tenido problemas para cambiar la version de Python que estaba utilizando puesto que con Python 3.6.9 los prints del código de la herramienta me daban error. Como no fuí capaz de arreglarlo al final he utilizado mi máquina Ubuntu, donde no he tenido ningún problema ni con la instalación ni la ejecución de volatility.

Instalación

Con apenas dos comandos he conseguido instalarlo. En la terminal introduciendo:

- 1- git clone <https://github.com/volatilityfoundation/volatility.git>
- 2- sudo python setup.py install

Ahora que ya he comentado de forma breve la herramienta voy a pasar con la elaboración del informe.

INFORME – PENTESTING

Para empezar, disponemos de una imagen de la memoria Ram afectada por Zeus. Primero, ejecuto una de las opciones más sencillas que proporciona volatility, con este comando vamos a poder saber el sistema operativo sobre el que se ha realizado el dump sobre el que luego vamos a realizar todo el análisis. Para ello ejecuto **“sudo python2.7 vol.py -f trojan.vmem/zeus.vmem imageinfo”** desde el directorio donde se encuentra vol.py.

```
carlosmartin@carlosmartin-TUF-Gaming-FX5050T-FX5050T:~/volatility$ sudo python2.7 vol.py -f trojan.vmem/zeus.vmem imageinfo
```

Tras la ejecución del comando obtenemos información:

```
INFO      : volatility.debug      : Determining profile based on KDBG search...
           Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
           AS Layer1            : IA32PagedMemoryPae (Kernel AS)
           AS Layer2            : FileAddressSpace (/home/carlosmartin/volatility/trojan.vmem/zeus.vmem)
           PAE type              : PAE
           DTB                  : 0x319000L
           KDBG                  : 0x80544ce0L
           Number of Processors : 1
           Image Type (Service Pack) : 2
           KPCR for CPU 0       : 0xffdf000L
           KUSER_SHARED_DATA    : 0xffdf000L
           Image date and time  : 2010-08-15 19:17:56 UTC+0000
           Image local date and time : 2010-08-15 15:17:56 -0400
```

En la información podemos ver que nos sugieren dos posibles perfiles a los que puede pertenecer el dump, que puede ser Windows XP SP2 o SP3 cuyas arquitecturas son x86, así como el huso horario -0400.

Ahora que ya conozco los dos potenciales perfiles a los que puede pertenecer, voy a analizar con el comando **“sudo python2.7 vol.py -f trojan.vmem/zeus.vmem -profile=WinXPSP2x86 pslist”**. Antes de continuar voy hablar sobre el comando anterior.

Para la ejecución del comando utilizo el plugin pslist proporcionado por volatility, el cuál nos va a proporcionar una lista de todos los procesos de un sistema, en este caso del perfil que le hemos mandado analizar WinXPSPx86. Este plugin nos va a proporcionar mucha información sobre los procesos entre ellos el Offset Virtual (para obtener el Offset Físico tendríamos que añadir -P), nombre e ID del proceso, también el ID del proceso del padre, número de subprocesos, número de identificadores y también la fecha y hora en la que el proceso empezó y terminó.

Ejecutando el comando comentado obtenemos:

Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start	Exit
0x810b1660	System	4	0	58	379	-----	0		
0xff2ab020	smss.exe	544	4	3	21	-----	0	2010-08-11 06:06:21 UTC+0000	
0xff1ecda0	csrss.exe	608	544	10	410	0	0	2010-08-11 06:06:23 UTC+0000	
0xff1ec978	winlogon.exe	632	544	24	536	0	0	2010-08-11 06:06:23 UTC+0000	
0xff247020	services.exe	676	632	16	288	0	0	2010-08-11 06:06:24 UTC+0000	
0xff255020	lsass.exe	688	632	21	405	0	0	2010-08-11 06:06:24 UTC+0000	
0xff218230	vmacthlp.exe	844	676	1	37	0	0	2010-08-11 06:06:24 UTC+0000	
0x80ff88d8	svchost.exe	856	676	29	336	0	0	2010-08-11 06:06:24 UTC+0000	
0xff217560	svchost.exe	936	676	11	288	0	0	2010-08-11 06:06:24 UTC+0000	
0x80fbf910	svchost.exe	1028	676	88	1424	0	0	2010-08-11 06:06:24 UTC+0000	
0xff22d558	svchost.exe	1088	676	7	93	0	0	2010-08-11 06:06:25 UTC+0000	
0xff203b80	svchost.exe	1148	676	15	217	0	0	2010-08-11 06:06:26 UTC+0000	
0xff1d7da0	spoolsv.exe	1432	676	14	145	0	0	2010-08-11 06:06:26 UTC+0000	
0xff1b8b28	vmtoolsd.exe	1668	676	5	225	0	0	2010-08-11 06:06:35 UTC+0000	
0xff1fdc88	VMUpgradeHelper	1788	676	5	112	0	0	2010-08-11 06:06:38 UTC+0000	
0xff143b28	TPAutoConnSvc.e	1968	676	5	106	0	0	2010-08-11 06:06:39 UTC+0000	
0xff25a7e0	alg.exe	216	676	8	120	0	0	2010-08-11 06:06:39 UTC+0000	
0xff364310	wscntfy.exe	888	1028	1	40	0	0	2010-08-11 06:06:49 UTC+0000	
0xff38b5f8	TPAutoConnect.e	1084	1968	1	68	0	0	2010-08-11 06:06:52 UTC+0000	
0x80f60da0	wuauclt.exe	1732	1028	7	189	0	0	2010-08-11 06:07:44 UTC+0000	
0xff3865d0	explorer.exe	1724	1708	13	326	0	0	2010-08-11 06:09:29 UTC+0000	
0xff3667e8	VMwareTray.exe	432	1724	1	60	0	0	2010-08-11 06:09:31 UTC+0000	
0xff374980	VMwareUser.exe	452	1724	8	207	0	0	2010-08-11 06:09:32 UTC+0000	
0x80f94588	wuauclt.exe	468	1028	4	142	0	0	2010-08-11 06:09:37 UTC+0000	
0xff224020	cmd.exe	124	1668	0	-----	0	0	2010-08-15 19:17:55 UTC+0000	2010-08-15 19:17:56 UTC+0000

Aquí podemos ver todos los procesos y de qué procesos "dependen", es decir, el PID del proceso padre. Como hacer encadenamiento hacia atrás de esta forma no es intuitivo y lioso, volatility nos ofrece otro plugin llamado pstree, que nos proporciona el árbol de procesos. Con el comando: **"sudo python2.7 vol.py -f trojan.vmem/zeus.vmem --profile=WinXPSP2x86 pstree"**.

Name	Pid	PPid	Thds	Hnds	Time
0x810b1660:System	4	0	58	379	1970-01-01 00:00:00 UTC+0000
. 0xff2ab020:smss.exe	544	4	3	21	2010-08-11 06:06:21 UTC+0000
.. 0xff1ec978:winlogon.exe	632	544	24	536	2010-08-11 06:06:23 UTC+0000
... 0xff255020:lsass.exe	688	632	21	405	2010-08-11 06:06:24 UTC+0000
... 0xff247020:services.exe	676	632	16	288	2010-08-11 06:06:24 UTC+0000
.... 0xff1b8b28:vmtoolsd.exe	1668	676	5	225	2010-08-11 06:06:35 UTC+0000
..... 0xff224020:cmd.exe	124	1668	0	----	2010-08-15 19:17:55 UTC+0000
.... 0x80ff88d8:svchost.exe	856	676	29	336	2010-08-11 06:06:24 UTC+0000
.... 0xff1d7da0:spoolsv.exe	1432	676	14	145	2010-08-11 06:06:26 UTC+0000
.... 0x80fbf910:svchost.exe	1028	676	88	1424	2010-08-11 06:06:24 UTC+0000
..... 0x80f60da0:wuauclt.exe	1732	1028	7	189	2010-08-11 06:07:44 UTC+0000
..... 0x80f94588:wuauclt.exe	468	1028	4	142	2010-08-11 06:09:37 UTC+0000
..... 0xff364310:wscntfy.exe	888	1028	1	40	2010-08-11 06:06:49 UTC+0000
.... 0xff217560:svchost.exe	936	676	11	288	2010-08-11 06:06:24 UTC+0000
.... 0xff143b28:TPAutoConnSvc.e	1968	676	5	106	2010-08-11 06:06:39 UTC+0000
..... 0xff38b5f8:TPAutoConnect.e	1084	1968	1	68	2010-08-11 06:06:52 UTC+0000
.... 0xff22d558:svchost.exe	1088	676	7	93	2010-08-11 06:06:25 UTC+0000
.... 0xff218230:vmacthlp.exe	844	676	1	37	2010-08-11 06:06:24 UTC+0000
.... 0xff25a7e0:alg.exe	216	676	8	120	2010-08-11 06:06:39 UTC+0000
.... 0xff203b80:svchost.exe	1148	676	15	217	2010-08-11 06:06:26 UTC+0000
.... 0xff1fdc88:VMUpgradeHelper	1788	676	5	112	2010-08-11 06:06:38 UTC+0000
.. 0xff1ecdad0:csrss.exe	608	544	10	410	2010-08-11 06:06:23 UTC+0000
0xff3865d0:explorer.exe	1724	1708	13	326	2010-08-11 06:09:29 UTC+0000
. 0xff374980:VMwareUser.exe	452	1724	8	207	2010-08-11 06:09:32 UTC+0000
. 0xff3667e8:VMwareTray.exe	432	1724	1	60	2010-08-11 06:09:31 UTC+0000

Debido a que los comandos anteriores nos ocultan algunos procesos que pueden ser de utilidad, voy a utilizar otro plugin llamado psxview, que nos va a mostrar también los procesos ocultos. Con el comando: **"sudo python2.7 vol.py -f trojan.vmem/zeus.vmem --profile=WinXPSP2x86 psxview"**.

Offset(P)	Name	PID	pslist	psscan	thrdproc	pspcid	csrss	session	deskthrd	ExitTime
0x06015020	services.exe	676	True	True	True	True	True	True	True	
0x063c5560	svchost.exe	936	True	True	True	True	True	True	True	
0x06499b80	svchost.exe	1148	True	True	True	True	True	True	True	
0x04c2b310	wscntfy.exe	888	True	True	True	True	True	True	True	
0x049c15f8	TPAutoConnect.e	1084	True	True	True	True	True	True	True	
0x05f027e0	alg.exe	216	True	True	True	True	True	True	True	
0x05f47020	lsass.exe	688	True	True	True	True	True	True	True	
0x010f7588	wuauclt.exe	468	True	True	True	True	True	True	True	
0x01122910	svchost.exe	1028	True	True	True	True	True	True	True	
0x069d5b28	vmtoolsd.exe	1668	True	True	True	True	True	True	True	
0x06384230	vmacthlp.exe	844	True	True	True	True	True	True	True	
0x0115b8d8	svchost.exe	856	True	True	True	True	True	True	True	
0x04b5a980	VMwareUser.exe	452	True	True	True	True	True	True	True	
0x010c3da0	wuauclt.exe	1732	True	True	True	True	True	True	True	
0x04a065d0	explorer.exe	1724	True	True	True	True	True	True	True	
0x04be97e8	VMwareTray.exe	432	True	True	True	True	True	True	True	
0x0211ab28	TPAutoConnSvc.e	1968	True	True	True	True	True	True	True	
0x06945da0	spoolsv.exe	1432	True	True	True	True	True	True	True	
0x066f0978	winlogon.exe	632	True	True	True	True	True	True	True	
0x0655fc88	VMUpgradeHelper	1788	True	True	True	True	True	True	True	
0x061ef558	svchost.exe	1088	True	True	True	True	True	True	True	
0x06238020	cmd.exe	124	True	True	False	True	False	False	False	2010-08-15 19:17:56 UTC+0000
0x066f0da0	csrss.exe	608	True	True	True	True	False	True	True	
0x05471020	smss.exe	544	True	True	True	True	False	False	False	
0x01214660	System	4	True	True	True	True	False	False	False	
0x069a7328	VMip.exe	1944	False	True	False	False	False	False	False	2010-08-15 19:17:56 UTC+0000

Esto nos sirve para poder identificar ciertos procesos donde podría estar ejecutándose el malware, en este caso el troyano zeus. Para conocer de donde puede provenir dicho malware, debemos conocer como funciona este. De forma habitual estos malwares, utilizan conexiones remotas para poder enviar información desde el equipo infectado, hasta el equipo de quién quiere obtener dicha información.

Para ello, voy a buscar conexiones que hayan sido establecidas. El pugling connscan, nos va a permitir detectar todas las conexiones (incluidas ocultas) TCP IP que se hayan realizado en el sistema. Para ello

ejecuto el comando: `“sudo python2.7 vol.py -f trojan.vmem/zeus.vmem --profile=WinXPSP2x86 connscan”`.

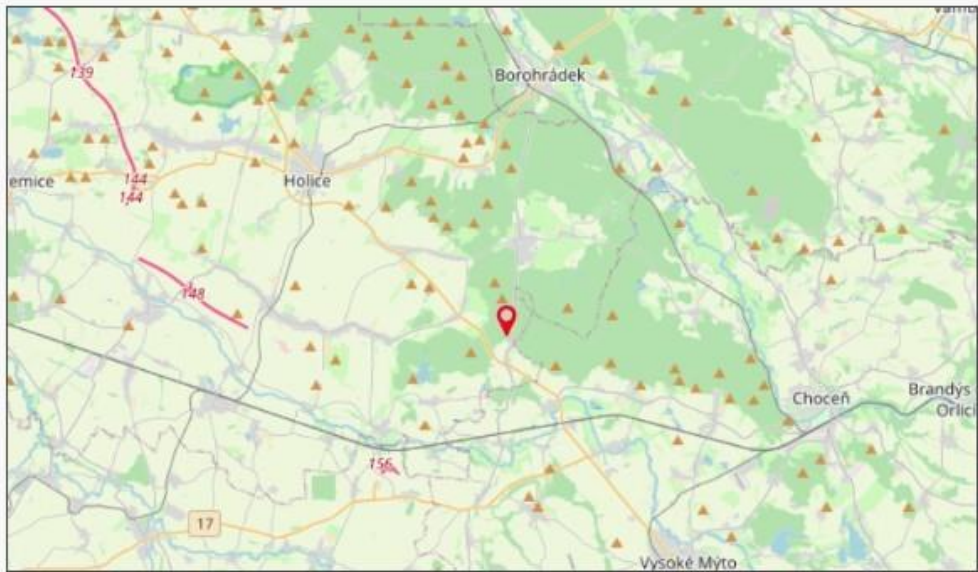
Offset(P)	Local Address	Remote Address	Pid
-----	-----	-----	---
0x02214988	172.16.176.143:1054	193.104.41.75:80	856
0x06015ab0	0.0.0.0:1056	193.104.41.75:80	856

Podemos ver que existen dos conexiones, una desde nuestra máquina local, y otra desde una máquina remota con la ip “193.104.41.75” a través del puerto 80 (Puerto HTTP). También sabemos el PID del proceso que genera esta conexión, el PID 856, que consultando el listado de procesos de capturas anteriores vemos que este PID se asocia con el proceso “svchost.exe”, el cuál analizaremos en profundidad luego.


Ya disponemos de la IP de la máquina a la que se ha conectado el troyano Zeus, una vez ya entró en la máquina de entrenamiento de los drones militares ucranianos, la cuál que quería comprometer. Cuya IP es la que he comentado en el párrafo anterior. Para conocer la zona geográfica aproximada de esta máquina, voy a utilizar varias herramientas web y contrastar si la localización que estas nos proporcionan coincide.

Para ello busco en Google “geolocalización ip” y obtengo numerosas páginas web.

1. Geolocation.com



API de geolocalización del W3C demo

País	Región	Ciudad
Czechia 	Pardubický kraj	Jaroslav
Código Postal	Latitud	Longitud
533 74	50.01224	16.07788

2. Maxmind.com

IP Addresses

193.104.41.75

Enter up to 25 IP addresses separated by spaces or commas. You can also [test your own IP address](#).

Submit

GeoIP2 City Plus Web Service Results

IP Address	Country Code	Location	Network	Postal Code	Approximate Coordinates*	Accuracy Radius (km)	ISP	Organization	Do
193.104.41.75	CZ	Chrudim, Chrudim District, Pardubický kraj, Czechia, Europe	193.104.41.0/24	537 01	49.9487, 15.7933	10	ECOMP spol s r.o.	ECOMP spol s r.o.	cp

Y por último una tercera.

3. Nordvpn.com

Búsqueda de dirección IP

Introduce la dirección IP que te interesa:

193.104.41.75

Obtener detalles de IP

Proveedor de servicios de internet: ECOMP spol s r.o.

País: Czechia, Chrudim District, Chrudim

Ciudad: Chrudim

Nombre del servidor: cpsnet.cz

Región/Estado: Chrudim District

Código de zona: 537 01

Las 3 páginas web coinciden con que dicha IP proviene de República Checa, y aún más en concreto de la región de Chrudim. Con certeza no podemos concretar más puesto que las dos primeras páginas nos ofrecen coordenadas, que podemos ver que son bastante similares, pero no exactamente iguales, puesto que tienen un ligero rango de “fallo”.

De forma resumida, conocemos tanto la IP → **193.104.41.75**, así como la zona geográfica de la máquina a la que se conecta el troyano zeus → **República Checa (Región: Chrudim)**.

Ahora que ya conocemos la IP así como su zona, voy a buscar si esta IP se encuentra en algunas listas negras de IPs potencialmente peligrosas. Toolbox es una herramienta que nos permite consultarlo y como vemos en la siguiente captura, hoy en día esta IP está limpia y no pertenece a ninguna de las 82 listas negras, en las que la aplicación web nos permite consultar.

SuperTool Beta7

193.104.41.75 [Blacklist Check](#)

blacklist:193.104.41.75 [Monitor This](#) [Solve Email Delivery Problems](#) [blacklist](#)

Are your email senders blacklisted? [LEARN MORE](#)

Checking **193.104.41.75** against **82** known blacklists...
Listed **0** times with **2** timeouts

	Blacklist	Reason	TTL	ResponseTime	
✓ OK	0SPAM			47	
✓ OK	Abuse.ro			114	
✓ OK	Abusix Mail Intelligence Blacklist			3	
✓ OK	Abusix Mail Intelligence Domain Blacklist			2	
✓ OK	Abusix Mail Intelligence Exploit list			3	
✓ OK	Anonmails DNSBL			106	

Para tener más de un resultado, consultamos otra página web que detecta la presencia de IPs en listas negras, en este caso ipvoid. De nuevo obtenemos el mismo resultado, la IP está limpia:

IP Blacklist Check

Scan an IP address through multiple DNS-based blackhole list (DNSBL) and IP reputation services, to facilitate the detection of IP addresses involved in malware incidents and spamming activities. This service checks in real-time an IP address through more than 80 IP reputation and DNSBL services. This service is built with the IP Reputation API by APIVoid.

193.104.41.75 [Check IP Address](#)

IP Address Information

Analysis Date	2022-11-25 13:13:47
Elapsed Time	5 seconds
Detections Count	0/106

Ahora necesitaríamos indicar evidencias de la presencia del troyano Zeus en la evidencia digital proporcionada. Sabemos el PID del proceso que genera la conexión con la IP que hemos hablado en párrafos anteriores.

Por tanto, ahora voy a utilizar el plugin handles que proporciona volatility, puesto que nos permite consultar los ficheros relacionados con un proceso. Este plugin nos permite añadir el parámetro -t, para especificar qué tipo de handle es el que queremos que nos muestre.

Con el comando **“sudo python2.7 vol.py -f trojan.vmem/zeus.vmem --profile=WinXPSP2x86 handles -t Event -p 856”** vamos a observar los eventos que ha generado el proceso.

Offset(V)	Pid	Handle	Access	Type	Details
0xff267ea0	856	0x28	0x21f0003	Event	
0xff2569f0	856	0x38	0x1f0003	Event	
0x80efd4a0	856	0x48	0x1f0003	Event	DINPUTWINMM
0xff268768	856	0x50	0x1f0003	Event	
0xff268738	856	0x54	0x1f0003	Event	
0x80f58f60	856	0x68	0x1f0003	Event	userenv: User Profile setup event
0xff25bfc0	856	0x6c	0x1f0003	Event	
0xff25be90	856	0xa0	0x1f0003	Event	
0xff25bf90	856	0xa4	0x1f0003	Event	
0xff267ab0	856	0xa8	0x1f0003	Event	
0xff267a80	856	0xac	0x1f0003	Event	
0xff269768	856	0xb4	0x1f0003	Event	
0xff2a3b60	856	0xc4	0x1f0003	Event	
0xff25d9f8	856	0xd0	0x1f0003	Event	
0xff25dd10	856	0xd4	0x1f0003	Event	
0xff280308	856	0x1f0	0x1f0003	Event	
0x80f618e8	856	0x1f4	0x1f0003	Event	crypt32LogoffEvent
0xff2342b8	856	0x1fc	0x1f0003	Event	
0xff22f0b0	856	0x204	0x1f0003	Event	
0xff2232b8	856	0x208	0x1f0003	Event	
0x80f331d8	856	0x20c	0x1f0003	Event	TermSrvReadyEvent
0x80f78ef0	856	0x230	0x1f0003	Event	
0xff1269b0	856	0x23c	0x1f0003	Event	
0xff268c18	856	0x240	0x100003	Event	
0xff22bed8	856	0x250	0x1f0003	Event	
0xff3b02c8	856	0x254	0x1f0003	Event	
0xff3b0298	856	0x258	0x1f0003	Event	
0xff3b91c8	856	0x25c	0x1f0003	Event	
0x80f762a8	856	0x260	0x1f0003	Event	
0xff378268	856	0x268	0x1f0003	Event	
0xff3960b8	856	0x270	0x1f0003	Event	
0x81001358	856	0x278	0x1f0003	Event	
0x80fca0b0	856	0x290	0x1f0003	Event	WinMMConsoleAudioEvent
0xff395118	856	0x2a0	0x1f0003	Event	ReconEvent
0xff3b6ea0	856	0x2a4	0x1f0003	Event	TermSrv: machine GP event
0xff2762f8	856	0x2a8	0x1f0003	Event	
0x80fcb068	856	0x2ac	0x1f0003	Event	
0x80fb05c8	856	0x2b0	0x100000	Event	userenv: Machine Group Policy has been applied

Con el comando **“sudo python2.7 vol.py -f trojan.vmem/zeus.vmem --profile=WinXPSP2x86 handles -t File -p 856”** vamos a observar los ficheros que están relacionados con el proceso.

Offset(V)	Pid	Handle	Access	Type	Details
0xff2495f8	856	0xc	0x100020	File	\Device\HarddiskVolume1\WINDOWS\system32
0xff26beb8	856	0x4c	0x100001	File	\Device\KsecDD
0xff26bbf8	856	0x64	0x100020	File	\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
0xff269668	856	0xb8	0x12019f	File	\Device\NamedPipe\nt\NtControlPipe2
0xff25d4b8	856	0x104	0x100000	File	\Device\Idfs
0xff27a280	856	0x28c	0x12019f	File	\Device\Termdd
0xff27e028	856	0x294	0x12019f	File	\Device\Termdd
0xff260028	856	0x2d0	0x12019f	File	\Device\NamedPipe\Ctx_WinStation_API_service
0xff284028	856	0x2d4	0x12019f	File	\Device\NamedPipe\Ctx_WinStation_API_service
0xff262330	856	0x2f4	0x12019f	File	\Device\Termdd
0xff220330	856	0x2f8	0x12019f	File	\Device\Termdd
0xff22b990	856	0x338	0x12019f	File	\Device\NamedPipe\lsarpc
0xff216c88	856	0x3d8	0x12019f	File	\Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Local Settings\Temporary Internet Files\Content.IE5\index.dat
0xff1fd028	856	0x3e8	0x12019f	File	\Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Cookies\index.dat
0xff298258	856	0x3f0	0x12019f	File	\Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Local Settings\History\History.IE5\index.dat
0xff298b28	856	0x40c	0x100020	File	\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
0xff1398c0	856	0x434	0x12019f	File	\Device\NamedPipe_AVIRA_2108
0xff1db540	856	0x454	0x100020	File	\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
0xff1d8258	856	0x498	0x12019f	File	\Device\NamedPipe\ROUTER
0x80fcb068	856	0x49c	0x120116	File	\Device\Tcp
0x80fbc060	856	0x4a0	0x1200a0	File	\Device\Tcp
0x80f06828	856	0x4a4	0x1200a0	File	\Device\Ip
0x80f2a028	856	0x4a8	0x100003	File	\Device\Ip
0x80fd1960	856	0x4ac	0x1200a0	File	\Device\Ip
0x80ef9990	856	0x4dc	0x12019f	File	\Device\NamedPipe\ROUTER
0x80fd12c0	856	0x528	0x1200a0	File	\Device\Tcp
0x80f2ae80	856	0x544	0x1f01ff	File	\Device\Afd\Endpoint
0xff26c028	856	0x548	0x1f01ff	File	\Device\Tcp
0x80ffc6a0	856	0x554	0x1200a0	File	\Device\NetBT_Tcpip_{050E22C4-A428-4EF8-A24C-45BFC93B64B7}

En nuestro caso, nos interesan ver los objetos de tipo Mutant (de exclusión mutua), que están asociados con el proceso que genera la comunicación que hemos comentado anteriormente, el proceso cuyo PID es 856. Por tanto, ejecutando el comando **“sudo python2.7 vol.py -f trojan.vmem/zeus.vmem -profile=WinXPSP2x86 handles -t Mutant -p 856”**.

Buscando información en Internet he encontrado, que el nombre del mutex que emplea el troyano Zeus es “_AVIRA_2108” y por tanto si el proceso (PID=856) que interviene en la conexión contiene este mutex, será una evidencia clara de que dicho proceso estaba infectado con el troyano.

Offset(V)	Pid	Handle	Access	Type	Details
0xff257148	856	0x24	0x1f0001	Mutant	SHIMLIB_LOG_MUTEX
0xff149878	856	0x158	0x1f0001	Mutant	
0xff2342e8	856	0x1d8	0x1f0001	Mutant	
0xff3864f8	856	0x1e4	0x120001	Mutant	ShimCacheMutex
0xff21e0e0	856	0x1ec	0x1f0001	Mutant	
0xff22f0e0	856	0x1f8	0x1f0001	Mutant	
0xff2232e8	856	0x200	0x1f0001	Mutant	
0xff2741f0	856	0x218	0x1f0001	Mutant	746bbf3569adEncrypt
0xff15a2c0	856	0x238	0x1f0001	Mutant	
0x80fca0e0	856	0x288	0x1f0001	Mutant	
0x80ef7a38	856	0x3d4	0x100000	Mutant	_!MSFTHISTORY!
0x80fdc1b8	856	0x3dc	0x1f0001	Mutant	c:\windows\system32\config\systemprofile\local settings\temporary internet files\content.ie5!
0x80f18290	856	0x3e0	0x1f0001	Mutant	c:\windows\system32\config\systemprofile\cookies!
0x80fbb40	856	0x3ec	0x1f0001	Mutant	c:\windows\system32\config\systemprofile\local settings\history\history.ie5!
0x80f8e1a8	856	0x3f8	0x1f0001	Mutant	ZonesCacheCounterMutex
0x80f66898	856	0x3fc	0x1f0001	Mutant	ZonesCounterMutex
0x80f30c90	856	0x404	0x1f0001	Mutant	ZonesLockedCacheCounterMutex
0xff2071d0	856	0x418	0x100000	Mutant	WininetStartupMutex
0xff1e3d48	856	0x420	0x1f0001	Mutant	
0x80f27f60	856	0x424	0x1f0001	Mutant	
0x80f0cb60	856	0x428	0x100000	Mutant	WininetProxyRegistryMutex
0xff27b7e8	856	0x43c	0x1f0001	Mutant	_AVIRA_2108
0x80f19200	856	0x450	0x1f0001	Mutant	
0xff1e68b0	856	0x460	0x100000	Mutant	RasPbFile

Como se puede ver en la captura, aparece el Mutant “_AVIRA_2108” y “RasPbFile”, mutantes que se asocian con la presencia del troyano Zeus.

Para consultar la persistencia de este malware, voy a buscar los registros de “WinLogon” ya que consultando en Internet, he encontrado que este troyano inyecta su código en winlogon.exe. Con el comando **“python2.7 vol.py -f trojan.vmem/zeus.vmem printkey -K "Microsoft\Windows NT\CurrentVersion\Winlogon"”**.

```
Legend: (S) = Stable (V) = Volatile

-----
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\software
Key name: Winlogon (S)
Last updated: 2010-08-15 19:17:23 UTC+0000

Subkeys:
(S) GPExtensions
(S) Notify
(S) SpecialAccounts
(V) Credentials

Values:
REG_DWORD AutoRestartShell : (S) 1
REG_SZ DefaultDomainName : (S) BILLY-DB5B96DD3
REG_SZ DefaultUserName : (S) Administrator
REG_SZ LegalNoticeCaption : (S)
REG_SZ LegalNoticeText : (S)
REG_SZ PowerdownAfterShutdown : (S) 0
REG_SZ ReportBootOk : (S) 1
REG_SZ Shell : (S) Explorer.exe
REG_SZ ShutdownWithoutLogon : (S) 0
REG_SZ System : (S)
REG_SZ Userinit : (S) C:\WINDOWS\system32\userinit.exe,C:\WINDOWS\system32\sdra64.exe,
REG_SZ VmApplet : (S) rundll32 shell32,Control_RunDLL "sysdm.cpl"
REG_DWORD SfcQuota : (S) 4294967295
REG_SZ allocatecdroms : (S) 0
REG_SZ allocatedasd : (S) 0
REG_SZ allocatefloppies : (S) 0
REG_SZ cachedlogonscount : (S) 10
REG_DWORD forceunlocklogon : (S) 0
REG_DWORD passwordexpirywarning : (S) 14
REG_SZ scremoveoption : (S) 0
REG_DWORD AllowMultipleTSSessions : (S) 1
REG_EXPAND_SZ UIHost : (S) logonui.exe
REG_DWORD LogonType : (S) 1
REG_SZ Background : (S) 0 0 0
REG_SZ AutoAdminLogon : (S) 0
REG_SZ DebugServerCommand : (S) no
REG_DWORD SFCDisable : (S) 0
REG_SZ WinStationsDisabled : (S) 0
REG_DWORD HibernationPreviouslyEnabled : (S) 1
REG_DWORD ShowLogonOptions : (S) 0
REG_SZ AltDefaultUserName : (S) Administrator
REG_SZ AltDefaultDomainName : (S) BILLY-DB5B96DD3
```


Carlos Martín Sanz
Grado en Ingeniería Informática – Mención Computación

También sabemos que cada vez que se ejecuta el ejecutable del troyano se escribe en el disco duro. La ruta para acceder a ello es “C: \WINDOWS\system32\sdra64.exe” como podemos ver en la captura anterior. Se encuentra al lado de la ruta de inicio de usuario, ya que el troyano se activa o ejecuta cuando encendemos o ponemos en funcionamiento la máquina infectada.

Ahora buscamos mas información sobre este ejecutable sdra64.exe con un filescan. Grep nos permite que nos muestre solo aquellos ficheros donde encuentra “sdra64.exe”:

```
(base) carlosmartin@carlosmartin-TUF-Gaming-FX5050T-FX5050T:~/volatility$ sudo python2.7 vol.py -f trojan.vmem/zeus.vmem --profile=WinXPSP2x86 filescan | grep sdra64.exe
Volatility Foundation Volatility Framework 2.6.1
0x00000000029d9b40 1 1 R----- \Device\HarddiskVolume1\WINDOWS\system32\sdra64.exe
0x00000000029d9cf0 1 0 -WD--- \Device\HarddiskVolume1\WINDOWS\system32\sdra64.exe
```

Nuestra intención ahora es detectar si ese ejecutable contiene virus, primero tenemos que hacer un volcado del fichero. Una vez lo tengamos , llevarlo a una página web que lo escaneé y nos diga los posibles virus que este contiene.

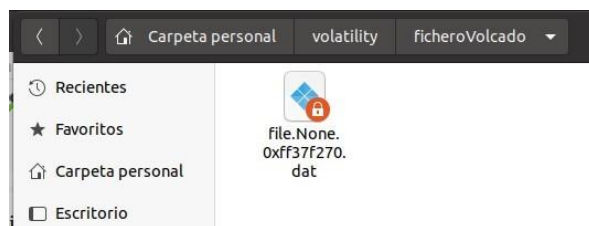
Primero el volcado del fichero, mediante el comando “python2.7 vol.py -f trojan.vmem/zeus.vmem --profile=WinXPSP2x86 dumpfiles -Q 0x00000000029d9b40”. Con ese comando no podía ejecutarlo puesto que me daba error, ya que no proporcionaba un directorio dump “Please specify a dump directory”. Para solucionarlo creo un directorio y lo añado al comando poniendo “-D ficheroVolcado”. Esto lo hago para ambos ficheros que contienen sdra64.exe:

```
~/volatility$ mkdir ficheroVolcado
~/volatility$ sudo python2.7 vol.py -f trojan.vmem/zeus.vmem --profile=WinXPSP2x86 dumpfiles -Q 0x00000000029d9b40 -D ficheroVolcado/
sudo python2.7 vol.py -f trojan.vmem/zeus.vmem --profile=WinXPSP2x86 dumpfiles -Q 0x00000000029d9cf0 -D ficheroVolcado/
```

Y esto es la salida que obtenemos:

```
DataSectionObject 0x029d9b40 None \Device\HarddiskVolume1\WINDOWS\system32\sdra64.exe
DataSectionObject 0x029d9cf0 None \Device\HarddiskVolume1\WINDOWS\system32\sdra64.exe
```

Ahora que ya hemos creado el fichero .dat, en la carpeta /volatility/ficheroVolcado, vamos a llevarlo a la página web (<https://www.virustotal.com/gui/home/upload>) donde subimos el fichero .dat que hemos creado.



Subiendo el fichero en VirusTotal obtenemos:

File Information:

- File Name: file.None.0xff37f270.dat
- Size: 128.00 KB
- Uploaded: 2022-11-05 16:03:47 UTC (20 days ago)
- File Type: EXE

Detection Summary: 59 security vendors and 1 sandbox flagged this file as malicious.

Security Vendors' Analysis:

Vendor	Detection	Engine	Detection
Ad-Aware	Trojan.Generic.3255903	AhnLab-V3	Win-Trojan/Zbot2.Gen
Alibaba	Trojan:Win32/Starter.ali2000005	ALYac	Trojan.Generic.3255903
Antiy-AVL	Trojan.Generic.ASMalwS.31	Arcabit	Trojan.Generic.D31AE5F
Avast	Win32:Evo-gen [Trj]	AVG	Win32:Evo-gen [Trj]
Avira (no cloud)	TR/Crypt.XPACK.Gen	BitDefender	Trojan.Generic.3255903

Carlos Martín Sanz

Grado en Ingeniería Informática – Mención Computación

La aplicación web nos muestra que 59 de las 71 herramientas que emplea, lo reconocen como un malware, como se puede ver el rojo la mayoría de ellas lo clasifican como un Trojan (Troyano), Zbot, nombre con el que el troyano Zeus también es conocido.

Conclusiones

Con esto finalizo el análisis forense de zeus.vmem, de la cuál mediante la herramienta volatility hemos conseguido obtener tanto la IP como la zona geográfica de la máquina a la cual se conectó de vuelta, una vez robados los datos necesarios de los drones ucranianos. Y también he demostrado la existencia del virus troyano (Zeus) en el sistema, así como la persistencia de este.

FIN