

# U3W3L5 – Progetto Settimanale

Progetto finale

**Autore:** Elena Pagnacco

## Sintesi

Il presente report documenta un'attività di laboratorio strutturata in quattro fasi, focalizzata sull'**amministrazione di sistema**, l'**analisi delle minacce** e l'**auditing di rete**. Le attività comprendono l'uso di **Windows PowerShell** per l'automazione e il **troubleshooting** di rete, l'analisi di Indicatori di Compromissione (IoC) tramite il sandbox **Any.run**, la ricognizione di rete e l'identificazione di servizi tramite **Nmap**, e infine la decostruzione tattica di un attacco **SQL Injection** analizzando un file di cattura (**PCAP**) in **Wireshark**.

## Obiettivi

- Esplorare le differenze tra **prompt** dei comandi e cmdlet di **PowerShell**, analizzando tabelle di routing (netstat) e processi di sistema.
- Studiare un report di minacce generato dalla piattaforma **Any.run**, identificando gli Indicatori di Compromissione (**IoC**).
- Comprendere e utilizzare **Nmap** per esplorare il localhost, la rete locale (LAN) e il server remoto scanme.nmap.org, identificando porte, servizi e sistemi operativi
- Visualizzare e analizzare un attacco di **SQL Injection** esaminando un file **PCAP** per estrarre informazioni sensibili, come gli hash delle password e la versione del database compromesso.

## Strumenti Utilizzati

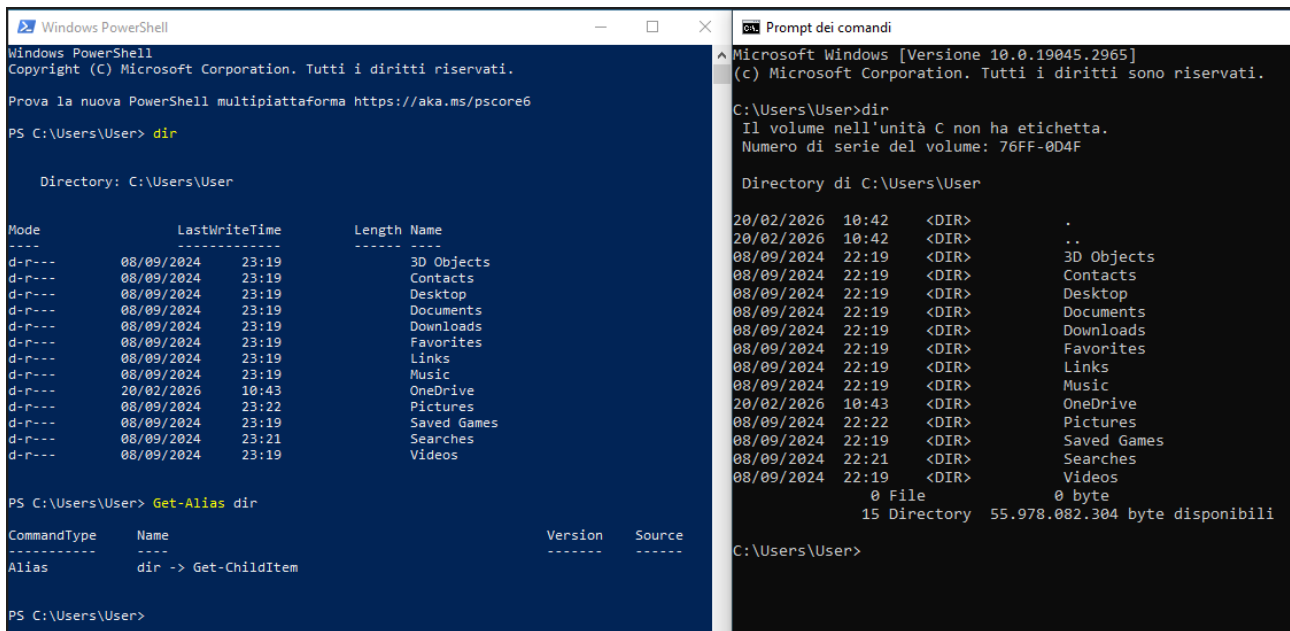
- **Windows PowerShell**: Potente strumento di automazione Microsoft che funge da console e linguaggio di scripting. A differenza del CMD che restituisce stringhe di testo, PowerShell opera tramite **Cmdlet** che restituiscono oggetti .NET.
  - **Any.run**: Sandbox interattiva per l'analisi dinamica dei **malware**. Permette di osservare il comportamento di un file in un ambiente sicuro e isolato, registrando traffico di rete, processi generati e modifiche al file system.
  - **Nmap (Network Mapper)**: Utility open-source standard di settore utilizzata per la **Network Discovery** e i **Security Audit**. Funziona inviando pacchetti IP grezzi in modi specifici per determinare gli host disponibili e i servizi in esecuzione.
  - **Wireshark**: Analizzatore di protocolli di rete impiegato per catturare e ispezionare il traffico a livello di pacchetto. Utile per analizzare file PCAP e seguire flussi di comunicazione TCP/http.
-

# Svolgimento

## Parte 1: Esercizio 1- Usare Windows PowerShell

### 1. Comandi base, Alias e Cmdlet

In questa fase è stato confrontato l'output del comando `$ dir` nel Prompt dei Comandi (CMD) e in PowerShell. In PowerShell, i comandi nativi, chiamati **Cmdlet**, seguono una sintassi *Verbo-Nome* (es. `Get-ChildItem`). Eseguendo il comando `$ Get-Alias dir` in PowerShell, viene confermato che `dir` funge semplicemente da alias per il cmdlet nativo `$ Get-ChildItem`.



The image shows two side-by-side terminal windows. The left window is 'Windows PowerShell' and the right is 'Prompt dei comandi' (Windows Command Prompt).

**Windows PowerShell Output:**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tutti i diritti riservati.

Prova la nuova PowerShell multiplatforma https://aka.ms/pscore6

PS C:\Users\User> dir

Directory: C:\Users\User

Mode                LastWriteTime         Length Name
----                -
d-r-----      08/09/2024   23:19             30 3D Objects
d-r-----      08/09/2024   23:19             10  Contacts
d-r-----      08/09/2024   23:19             10  Desktop
d-r-----      08/09/2024   23:19             10  Documents
d-r-----      08/09/2024   23:19             10  Downloads
d-r-----      08/09/2024   23:19             10  Favorites
d-r-----      08/09/2024   23:19             10  Links
d-r-----      08/09/2024   23:19             10  Music
d-r-----      20/02/2026   10:43             10  OneDrive
d-r-----      08/09/2024   23:22             10  Pictures
d-r-----      08/09/2024   23:19             10  Saved Games
d-r-----      08/09/2024   23:21             10  Searches
d-r-----      08/09/2024   23:19             10  Videos

PS C:\Users\User> Get-Alias dir

CommandType      Name
-----
Alias            dir -> Get-ChildItem

PS C:\Users\User>
```

**Prompt dei comandi Output:**

```
Microsoft Windows [Versione 10.0.19045.2965]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\User>dir
Il volume nell'unità C non ha etichetta.
Numero di serie del volume: 76FF-0D4F

Directory di C:\Users\User

20/02/2026  10:42  <DIR>      .
20/02/2026  10:42  <DIR>      ..
08/09/2024  22:19  <DIR>      3D Objects
08/09/2024  22:19  <DIR>      Contacts
08/09/2024  22:19  <DIR>      Desktop
08/09/2024  22:19  <DIR>      Documents
08/09/2024  22:19  <DIR>      Downloads
08/09/2024  22:19  <DIR>      Favorites
08/09/2024  22:19  <DIR>      Links
08/09/2024  22:19  <DIR>      Music
08/09/2024  22:19  <DIR>      OneDrive
20/02/2026  10:43  <DIR>      Pictures
08/09/2024  22:22  <DIR>      Saved Games
08/09/2024  22:19  <DIR>      Searches
08/09/2024  22:21  <DIR>      Videos
08/09/2024  22:19  <DIR>

0 File      0 byte
15 Directory 55.978.082.304 byte disponibili

C:\Users\User>
```

Figura 1 Powershell vs prompt dei comandi

Testando ulteriori comandi di base richiesti, come **ping**, **cd** e **ipconfig**, i risultati ottenuti nella console **PowerShell** sono identici a quelli restituiti dal classico **Prompt dei Comandi**. Questo conferma che **PowerShell** supporta e interpreta correttamente le utility di rete e di navigazione tradizionali di Windows, oltre ai propri **Cmdlet** nativi.

### 2. Esplorazione della Rete (netstat)

L'utility `netstat` mostra le statistiche del protocollo e le connessioni TCP/IP attuali. Utilizzando il comando `$ netstat -r` è stata visualizzata la "IPv4 Route Table" (Tabella di Routing). Analizzando l'output, il **gateway IPv4** responsabile dell'instradamento verso reti esterne (associato alla destinazione di rete 0.0.0.0) corrisponde all'indirizzo IP 10.0.2.2.

```

PS C:\Users\User> netstat -r
=====
Elenco interfacce
 10...08 00 27 96 c2 10 .....Intel(R) PRO/1000 MT Desktop Adapter
 1.....Software Loopback Interface 1
=====

IPv4 Tabella route
=====
Route attive:

```

Indirizzo rete	Mask	Gateway	Interfaccia	Metrica
0.0.0.0	0.0.0.0	10.0.2.2	10.0.2.15	25
10.0.2.0	255.255.255.0	On-link	10.0.2.15	281
10.0.2.15	255.255.255.255	On-link	10.0.2.15	281
10.0.2.255	255.255.255.255	On-link	10.0.2.15	281
127.0.0.0	255.0.0.0	On-link	127.0.0.1	331
127.0.0.1	255.255.255.255	On-link	127.0.0.1	331
127.255.255.255	255.255.255.255	On-link	127.0.0.1	331
224.0.0.0	240.0.0.0	On-link	127.0.0.1	331
224.0.0.0	240.0.0.0	On-link	10.0.2.15	281
255.255.255.255	255.255.255.255	On-link	127.0.0.1	331
255.255.255.255	255.255.255.255	On-link	10.0.2.15	281

```

=====
Route permanenti:
 Nessuna
=====

```

Figura 2 Tabella di routing (output di netstat -r)

### 3. Associazione Processi e Connessioni TCP

Avviando **PowerShell** con privilegi di Amministratore (Run as Administrator), è stato lanciato il comando \$ netstat -abno.

Questo comando visualizza tutte le connessioni (-a), gli eseguibili (-b), in formato numerico (-n), associati al relativo Process ID (-o). Selezionando il PID 900 e cercandolo nella scheda "Dettagli" di Gestione Attività (Task Manager), è stato possibile identificare il processo **svchost.exe**, eseguito dall'utente SERVIZIO DI RETE (NETWORK SERVICE), e ispezionarne le proprietà.

The image shows two side-by-side screenshots. The left screenshot is a PowerShell window displaying the output of the command `netstat -abno`. It lists various network connections and the processes associated with them. The right screenshot shows the Windows Task Manager's 'Gestione attività' window, specifically the 'Dettagli' (Details) tab. It shows a list of processes, with **svchost.exe** (PID 900) highlighted. A 'Proprietà - svchost' dialog box is open, showing the 'Descrizione' (Description) tab, which provides details about the process, including its type, version, and file path.

Figura 3 output di "netstat -abno" e task manager e dettagli del processo con PID 900

Dalla finestra "Dettagli" del Task Manager e da quella "Proprietà" del processo, si possono ottenere alcune informazioni interessanti come il nome dell'eseguibile (es. svchost.exe), lo stato di esecuzione (Running), il

nome utente proprietario (NETWORK SERVICE), la versione file, il Copyright, l'utilizzo di CPU/Memoria (4,788 K) e una descrizione del processo (Host Process for Windows Services).

## 4. Automazione dell'OS: Svuotare il Cestino

I **cmdlet** di **PowerShell** semplificano la gestione del sistema riducendo i passaggi grafici. Utilizzando il comando `$ clear-recyclebin`, il sistema ha richiesto un prompt di conferma di sicurezza. Digitando "Y" (Yes) o "S" (Sì), tutti i file sono stati eliminati **permanentemente** dal PC.

```
PS C:\Windows\system32> clear-recyclebin

Conferma
Eeguire l'operazione?
Esecuzione dell'operazione "Clear-RecycleBin" sulla destinazione "Tutto il contenuto del Cestino".
[S] Sì [T] Sì a tutti [N] No [U] No a tutti [O] Sospendi [?] Guida (il valore predefinito è "S"): s
PS C:\Windows\system32>
```

Figura 4 eliminazione file dal cestino tramite PowerShell

Altri comandi potenzialmente utili sono:

- **Get-EventLog**: per analizzare rapidamente i log di sistema (es. accessi falliti).
- **Get-Process**: per individuare processi sospetti in memoria.
- **Test-NetConnection**: per testare la connettività e le porte aperte senza strumenti esterni.

---

## Parte 2: Esercizio 2- Studio IoC (Any.run)

Dall'analisi del sandbox generata sulla piattaforma **Any.run**, emerge l'esecuzione di una complessa catena di infezione derivante dalla navigazione web verso un URL compromesso.

### Minaccia Rilevata (Payload Delivery)

L'analisi inizia con il processo **firefox.exe** (PID 6596) che visita un repository **GitHub** utilizzato impropriamente per l'hosting di malware. Questa azione innesca il download silente (**Drop**) e la successiva esecuzione non di uno, ma di **due payload malevoli paralleli**: il primo denominato **Jvczfhe.exe** (PID 7492) e il secondo denominato **Muadnrd.exe** (PID 7824).

### Identificazione Forense (Hashes della Kill Chain)

L'analisi forense permette di mappare due fasi distinte dell'infezione, identificando l'impronta digitale univoca sia del file di innesco che dei payload finali:

**1. File di Innesco (Submitted Object):** L'artefatto iniziale che scatena l'esecuzione di Firefox verso l'URL compromesso presenta i seguenti hash:

- **MD5:** 00B5E91B42712471CDFBDB37B715670C
- **SHA256:** 0307EE805DF8B94733598D5C3D62B28678EAEADBF1CA3689FA678A3780DD3DF0

**2. Payload Finali (Dropped Executables):** Una volta completato il download su disco, gli hash univoci dei veri e propri eseguibili malevoli risultano essere:

- **Jvczfhe.exe:**
  - **MD5:** 5EC4256E6A2367502A8058F4BC8F4ECC
  - **SHA256:** E6A7AAFF54EB6D06ACFC6F1DFA21A85B767DBF7FF3E9BFDF2DDBDECED86AA9B2
- **Muadnrd.exe:**

- **MD5:** 9773175646F2942573BB40551B142A99
- **SHA256:** B662E7213F4985684439E655BD92EA4B9A1566E76712BB86E1238113A35B90A0

14623 ms	6596	firefox.exe	C:\Users\admin\Downloads\OOD5yt-b.exe.part	105 Kb	executable
14936 ms	6596	firefox.exe	C:\Users\admin\Downloads\Jvczfhe.exe	105 Kb	executable
14951 ms	6596	firefox.exe	C:\Users\admin\Downloads\Jvczfhe.exe:Zone.Identifier	50 b	text
57199 ms	1356	WerFault.exe	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\AppCrash_Jvczfhe.exe_e8f4a47beb21929faaf5bbbc7cb947adda294c9_7cb78550_5641ad1f-ff35-4b34-8a69-19f8a8163939\Report.wer	–	Oversized
57719 ms	1356	WerFault.exe	C:\Users\admin\AppData\Local\CrashDumps\Jvczfhe.exe.7492.dmp	9 Mb	Oversized
105.12 s	6596	firefox.exe	C:\Users\admin\Downloads\xtorOyHX.exe.part	106 Kb	executable
105.37 s	6596	firefox.exe	C:\Users\admin\Downloads\Muadnrd.exe	106 Kb	executable

Figura 5 Dettaglio delle attività di file system (Dropped Executables) generate dal processo *firefox.exe*, con evidenza dei percorsi di salvataggio dei payload malevoli sul disco.

## Artefatti Forensi e Verifica OSINT

L'analisi profonda dei file generati dal processo *firefox.exe* ha permesso di tracciare con assoluta precisione la genesi di entrambi i payload, rivelando dettagli forensi cruciali che collegano l'attività di rete alla scrittura su disco:

- **Tracciamento di Jvczfhe.exe:** L'analisi degli hash conferma che il file è stato inizialmente scaricato come **OOD5yt-b.exe.part** (SHA256: E6A7A AFF54EB6D06ACFC6F1DFA21A85B767DBF7FF3E9BDF2DD8DECED86AA9B2), estensione tipica dei download incompleti di Firefox, per poi essere rinominato in *Jvczfhe.exe*. È stata inoltre rilevata la creazione dell'Alternate Data Stream (ADS) *Jvczfhe.exe:Zone.Identifier* ("Mark of the Web"), prova inconfutabile della provenienza esterna del file. La verifica su VirusTotal ha confermato la pericolosità del payload, identificato come **Trojan da 46 motori su 72**.

6596	firefox.exe	C:\Users\admin\Downloads\OOD5yt-b.exe.part	SHA256: E6A7A AFF54EB6D06ACFC6F1DFA21A85B767DBF7FF3E9BDF2DD8DECED86AA9B2	executable
6596	firefox.exe	C:\Users\admin\Downloads\Jvczfhe.exe	SHA256: E6A7A AFF54EB6D06ACFC6F1DFA21A85B767DBF7FF3E9BDF2DD8DECED86AA9B2	executable

Figura 6 Dettaglio delle attività di file system che mostra la correlazione tra il file temporaneo (.part) e il payload finale tramite hash MD5/SHA256 per l'eseguibile *Jvczfhe.exe*.

- **Tracciamento di Muadnrd.exe:** Parallelamente, è stato tracciato il download del secondo payload, inizialmente apparso sul file system come **xtorOyHX.exe.part** (SHA256: B662E721...) e successivamente rinominato in *Muadnrd.exe*. Nonostante questo hash risulti attualmente non rilevato (Zero-Day) su VirusTotal, l'analisi dinamica del sandbox ne ha confermato la natura malevola attraverso le attività di Process Injection documentate nel grafo di esecuzione.

6596	firefox.exe	C:\Users\admin\Downloads\xtorOyHX.exe.part	SHA256: B662E7213F4985684439E655BD92EA4B9A1566E76712BB86E1238113A35B90A0	executable
6596	firefox.exe	C:\Users\admin\Downloads\Muadnrd.exe	SHA256: B662E7213F4985684439E655BD92EA4B9A1566E76712BB86E1238113A35B90A0	executable

Figura 7 Dettaglio delle attività di file system che mostra la correlazione tra il file temporaneo (.part) e il payload finale tramite hash MD5/SHA256 per l'eseguibile *Muadnrd.exe*.

Questo doppio tracciamento dimostra una strategia di attacco combinata, dove un malware noto (*Jvczfhe*) viene accompagnato da un secondo eseguibile meno visibile ai controlli statici, entrambi veicolati tramite tecniche di download silente.

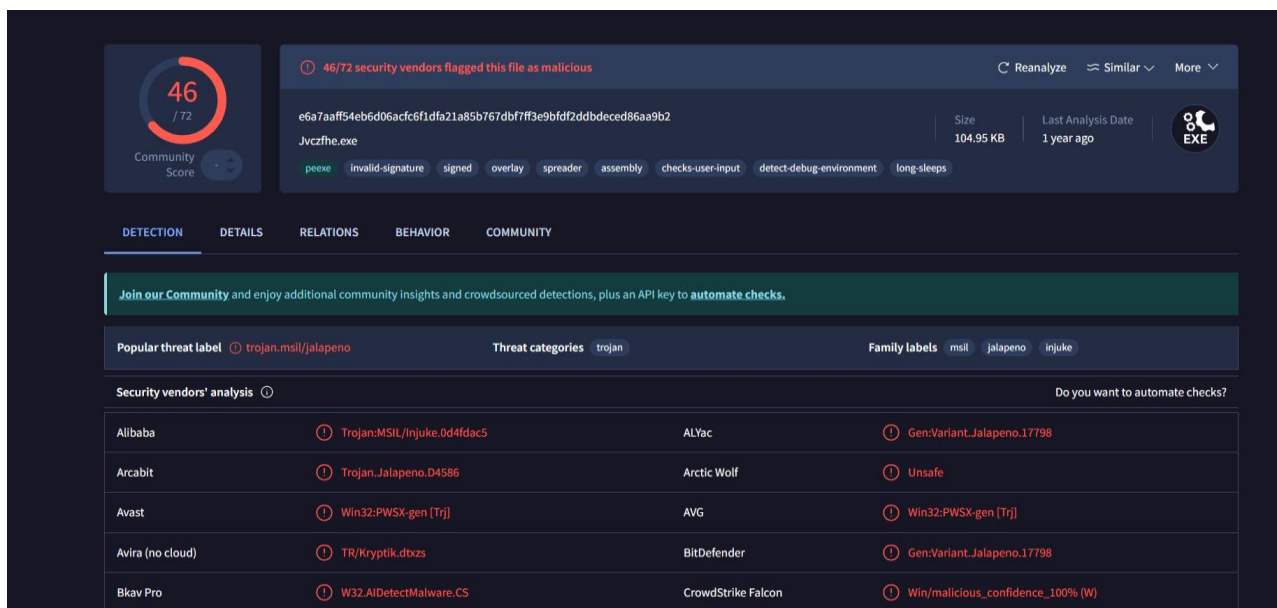


Figura 8 Analisi OSINT su VirusTotal per l'hash del payload Jvczfhe.exe. Il report evidenzia una detection rate di 46/72 e la classificazione della minaccia come "Trojan", confermando la pericolosità del file nonostante le tecniche evasive iniziali.

## Analisi Comportamentale ed Evasione

Il malware esibisce tecniche di **Living off the Land** (LOLBins) per eludere il rilevamento degli antivirus. Nello specifico, la minaccia esegue tecniche di **Process Injection**: come visibile dal grafo di esecuzione, l'eleggibile "sequestra" processi di sistema legittimi di Windows, come **InstallUtil.exe** e il sistema di segnalazione errori **WerFault.exe**, per eseguire comandi o nascondere le proprie tracce (flaggati chiaramente come minacce critiche "THREAT" dal sandbox). Entrambi i rami di esecuzione avviano inoltre script tramite cmd.exe usando timeout.exe per ritardare l'operatività e aggirare i controlli dinamici degli ambienti di analisi.

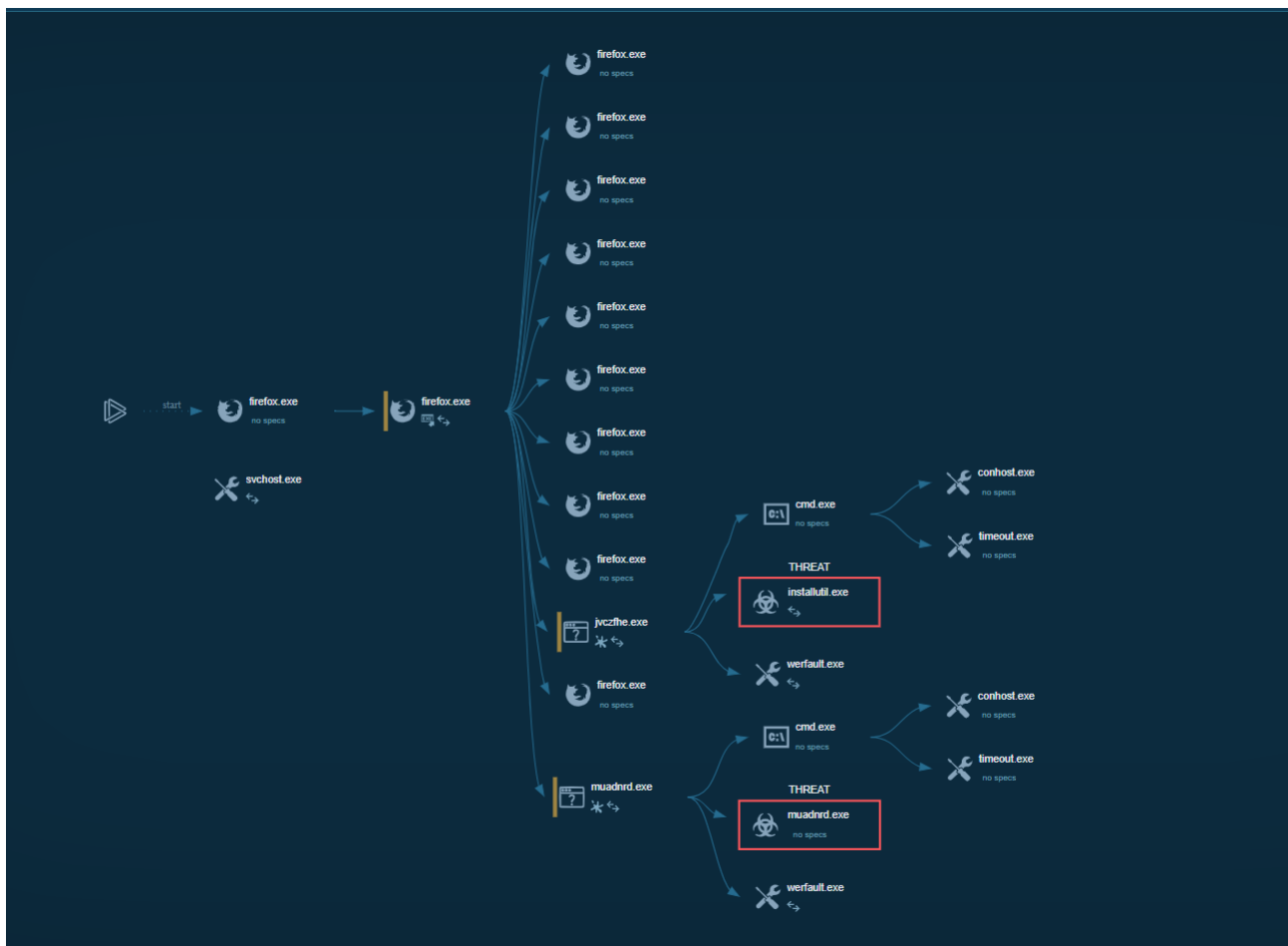


Figura 9 Grafo di esecuzione generato da Any.run che evidenzia l'infezione parallela, la compromissione tramite Process Injection (flaggata come THREAT) e l'utilizzo dei processi di sistema (LOLBins).

## Network IoC e Comando e Controllo (C2)

L'ispezione delle firme di rete (**Threats**) rileva diverse richieste DNS dinamiche verso domini associati a **\*.duckdns.org**. L'utilizzo di servizi Dynamic DNS è un forte **Indicatore di Compromissione**, poiché è una tecnica standard utilizzata dagli attaccanti per comunicare con la propria infrastruttura di Comando e Controllo (C2) mutando rapidamente indirizzo IP.

HTTP Requests		31	Connections		99	DNS Requests		8	Network Threats		19	<input type="checkbox"/> Hide whitelisted	duck	X	PCAP	▼
Timeshift	Status		Rep	Domain									IP			
55658 ms	Requested		?	egehgdhejbjhtre.duckdns.org									IP Addresses not found			
55659 ms	Requested		?	egehgdhejbjhtre.duckdns.org									IP Addresses not found			
55659 ms	Responded		?	egehgdhejbjhtre.duckdns.org									91.92.253.47			
134.01 s	Responded		?	egehgdhejbjhtre.duckdns.org									91.92.253.47			
186.26 s	Responded		?	egehgdhejbjhtre.duckdns.org									91.92.253.47			
186.26 s	Requested		?	egehgdhejbjhtre.duckdns.org									IP Addresses not found			
212.88 s	Responded		?	egehgdhejbjhtre.duckdns.org									91.92.253.47			
264.12 s	Responded		?	egehgdhejbjhtre.duckdns.org									91.92.253.47			

Figura 10 Il pannello delle minacce di rete (Threats/DNS) che evidenzia le query verso domini associati a servizi di Dynamic DNS (\*.duckdns.org).

## Parte 3: Bonus 1- Esplorazione di Nmap

**Avviso di Sicurezza:** Le scansioni Nmap sono state eseguite esclusivamente verso il localhost, la rete LAN di laboratorio e il server remoto 'scanme.nmap.org', esplicitamente autorizzato per questi test.



## 1. Analisi del Manuale (Man Pages)

In ambiente Linux (VM CyberOps), il comando `$ man nmap` ha aperto il manuale operativo del tool.

Dalla lettura del manuale si evince che Nmap ("Network Mapper") è uno strumento open source. Viene utilizzato principalmente per l'esplorazione della rete e l'auditing di sicurezza, permettendo di determinare rapidamente quali host sono disponibili, quali servizi offrono, i sistemi operativi in esecuzione e il tipo di filtri o firewall in uso.

Usando la funzione di ricerca `/example`, è stato individuato l'esempio `$ nmap -A -T4 scanme.nmap.org`.

- **L'opzione -A:** Abilita il rilevamento del sistema operativo e della versione, la scansione tramite script e il traceroute.
- **L'opzione -T4:** Imposta un template di temporizzazione più aggressivo per un'esecuzione più rapida.

**Example 1. A representative Nmap scan**

```
# nmap -A -T4 scanme.nmap.org
```

Figura 11 Example 1

## 2. Scansione Localhost e Subnet LAN

Per determinare i parametri di rete, è stato eseguito `$ ip address` ottenendo l'IP della VM (10.0.2.15/24) appartenente alla rete 10.0.2.0.

- **Localhost (nmap -A -T4 localhost):** La scansione dell'indirizzo di loopback ha rivelato la presenza di due porte principali in ascolto.  
La prima è la porta **21/tcp (FTP)**, fornita dal demone vsftpd 2.0.8 or later, la quale risulta vulnerabile in quanto permette l'accesso *Anonymous FTP* (login consentito senza credenziali).  
La seconda è la porta **22/tcp (SSH)**, gestita dal servizio **OpenSSH 10.0 (protocol 2.0)**. Questo servizio indica che la macchina è configurata per accettare connessioni e amministrazione remota sicura tramite riga di comando, esponendo però un potenziale punto di attacco se le credenziali sono deboli (brute-force).



```
[analyst@secOps ~]$ nmap -A -T4 localhost
Starting Nmap 7.97 ( https://nmap.org ) at 2026-02-20 09:21 -0500
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0016s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--  1 0      0          0 Mar 26  2018 ftp_test
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to 127.0.0.1
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 2
|     vsFTPD 3.0.5 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 10.0 (protocol 2.0)
Service Info: Host: Welcome

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.39 seconds
```

Figura 12 Localhost scan che rileva le porte 21 e 22 aperte

- **Rete Locale (nmap -A -T4 10.0.2.0/24):** Eseguendo la scansione dell'intera subnet con privilegi di amministratore, Nmap è riuscito a bypassare i filtri di base e ha individuato **3 host attivi** sulla LAN.
  - L'host 10.0.2.15 (la macchina locale) espone le porte 21 (FTP, vsFTPD 3.0.5) e 22 (SSH, OpenSSH 10.0).
  - L'host 10.0.2.2 presenta le porte 135/tcp (msrpc) e 445/tcp (microsoft-ds) aperte. L'impronta del sistema operativo (OS Fingerprint) e la tabella di routing confermano che si tratta del Gateway della rete NAT fornito dall'hypervisor VirtualBox.
  - L'host 10.0.2.3 espone unicamente la porta 53/tcp (DNS), operando come server di risoluzione dei nomi interno alla rete virtuale.

```
Nmap done: 256 IP addresses (3 hosts up) scanned in 103.48 seconds
```

Figura 13 Esecuzione della scansione Nmap con privilegi di amministratore sulla rete locale (10.0.2.0/24) e rilevamento degli host attivi

### 3. Scansione Server Remoto Autorizzato

Lo scopo del server [scanme.nmap.org](https://scanme.nmap.org) è quello di consentire agli utenti di testare il tool Nmap legalmente su un target autorizzato, senza scansionare infrastrutture senza permesso.

La scansione verso **scanme.nmap.org** ha rivelato le seguenti informazioni:

- **IP e OS:** 45.33.32.156, sistema operativo identificato come Linux.
- **Porte e Servizi Aperti:** 22/tcp (SSH - OpenSSH 6.6.1p1), 80/tcp (HTTP - Apache httpd 2.4.7), 9929/tcp (nping-echo) e 31337/tcp (tcpwrapped).

- **Porte Filtrate:** Diverse porte (996 filtered tcp ports) risultano filtrate, suggerendo la presenza di un firewall o di regole di packet filtering che bloccano le sonde di Nmap.

```
[analyst@secOps ~]$ nmap -A -T4 scanme.nmap.org

Starting Nmap 7.97 ( https://nmap.org ) at 2026-02-20 09:51 -0500
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.18s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256  96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256  33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
|_ _http-title: Go ahead and ScanMe!
|_ _http-favicon: Nmap Project
|_ _http-server-header: Apache/2.4.7 (Ubuntu)
9929/tcp  open  nping-echo   Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.83 seconds
```

Figura 14 L'output della scansione verso scanme.nmap.org

## Parte 4: Bonus 2- Attacco a un database MySQL (Analisi PCAP)

In questa fase è stato utilizzato **Wireshark** per analizzare il file **SQL\_Lab.pcap**, contenente il traffico di un attacco SQL Injection della durata di circa *8 minuti* contro un'applicazione web vulnerabile. Dalla cattura emerge che l'**IP sorgente (attaccante)** è 10.0.2.4 e l'**IP destinazione (vittima)** è 10.0.2.15.

L'analisi è stata condotta seguendo i flussi **TCP/HTTP** (Clic destro sul pacchetto > Segui > Flusso TCP/HTTP) per leggere le richieste GET in chiaro scambiate tra client e server.

### 1. Verifica della Vulnerabilità (Il payload "1=1")

L'attaccante ha manipolato il campo **ID** dell'URL inserendo il **payload 1=1**. Poiché in logica booleana "1=1" è un'affermazione sempre vera, il database invece di negare l'accesso ha restituito l'intero record dell'utente, dimostrando che l'input non era filtrato e il database eseguiva ciecamente il codice inserito nell'URL.

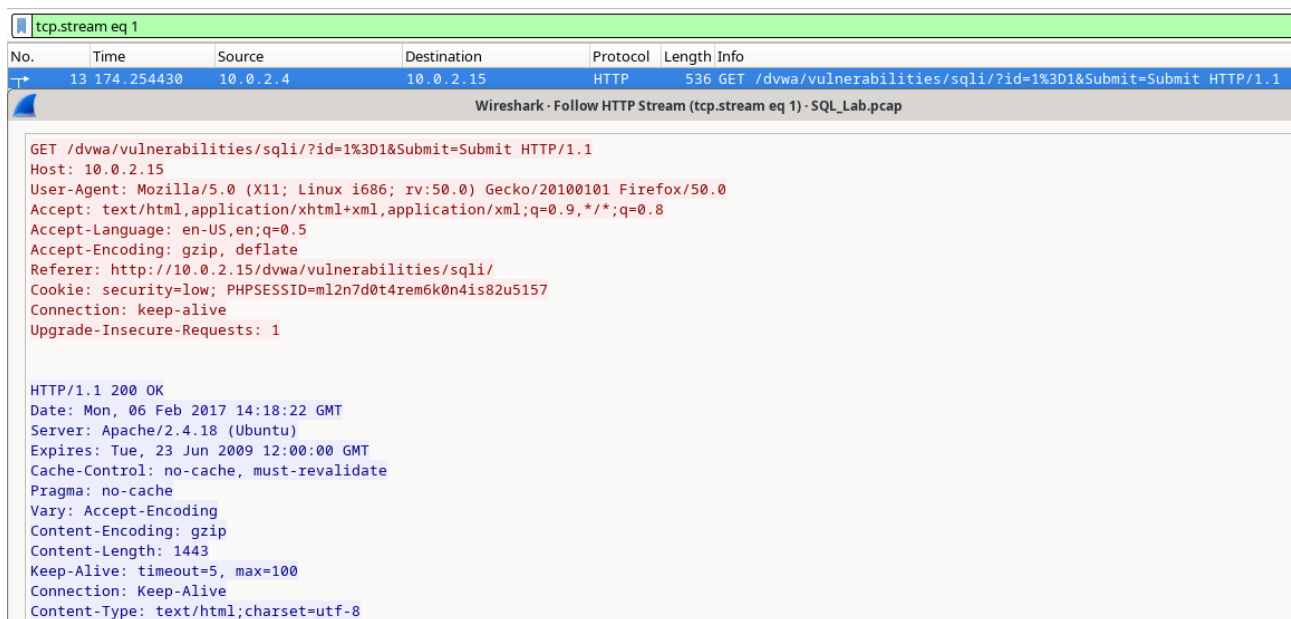


Figura 15 Wireshark: Flusso HTTP con la query 1=1 visibile (tcp.stream eq 1)

## 2. Enumerazione del Database e della Versione

Successivamente, l'attaccante ha iniettato la query **1' or 1=1 union select database(), user()#**. Il server ha risposto svelando il nome del database (**dvwa**) e l'utente corrente (**root@localhost**). Aumentando la complessità, l'attaccante ha usato **1' or 1=1 union select null, version ()#** ottenendo la versione esatta del sistema in uso: **5.7.12-0ubuntu1.1**. Questa informazione è critica perché permette a un attaccante di cercare exploit specifici per quella esatta release.

## 3. Esfiltrazione delle Tabelle e delle Password

L'aggressore ha poi mirato allo schema del database usando il costrutto **1' or 1=1 union select null, table\_name from information\_schema.tables#**, costringendo il server a stampare un lungo elenco di tabelle, tra cui la tabella **users**.

- **Nota analitica:** Se l'attaccante avesse usato una clausola WHERE (es. **WHERE table\_name='users'**), il database avrebbe restituito un output filtrato e molto più corto, limitando il rumore generato nei log di rete.

L'attacco si è concluso con l'estrazione degli hash delle password tramite la query **1' or 1=1 union select user, password from users#**.

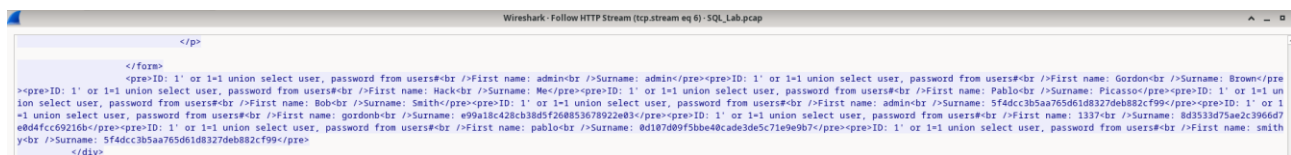


Figura 16 Finestra "Follow HTTP Stream" in Wireshark che mostra gli hash delle password esfiltrati

Leggendo l'output, l'account **1337** risulta associato all'hash **8d3533d75ae2c3966d7e0d4fcc69216b**. Utilizzando uno strumento di cracking online (**CrackStation**), è stato possibile decifrare facilmente l'hash, rivelando la password in chiaro: **charley**.

Hash	Type	Result
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley

Figura 17 Crack della password d'interesse tramite CrackStation.

## Domande di Riflessione Finali

1. **Il Ruolo di PowerShell in Sicurezza:** PowerShell è essenziale per un analista di sicurezza in quanto permette di automatizzare compiti ripetitivi (come l'estrazione massiva di log eventi con Get-EventLog), monitorare configurazioni di rete, gestire servizi sospetti ed eseguire script di "threat hunting" su molteplici endpoint contemporaneamente.
2. **Nmap: Difesa e Offesa:** Nmap è uno strumento a doppio taglio. Aiuta gli amministratori di rete a verificare le configurazioni del firewall, chiudere porte non necessarie e fare audit di sicurezza. Tuttavia, un attore malevolo lo utilizza nella fase iniziale di *reconnaissance* (ricognizione) per mappare la topologia della rete, scoprire servizi vulnerabili e pianificare attacchi mirati.
3. **Rischi e Prevenzione della SQL Injection:** Il rischio per i siti web basati su SQL è gravissimo: un attaccante può aggirare l'autenticazione, rubare o alterare i dati, fino a compromettere l'intero server. Per prevenire questi attacchi, è fondamentale implementare **due contromisure principali**:
  - Validare, sanificare e filtrare rigidamente tutto l'input fornito dagli utenti.
  - Utilizzare **Query Parametrizzate** (Stored Procedures o Prepared Statements), che separano logicamente il codice SQL dai dati inseriti dall'utente, impedendo che l'input venga interpretato come comando eseguibile. Inoltre, l'implementazione di un Web Application Firewall (WAF) fornisce un ulteriore livello vitale di mitigazione e monitoraggio.