

# Report di Laboratorio: Network Scanning & Traffic Analysis

Studente: Daniele di Martino

Data: 07 Gennaio 2026

Oggetto: Analisi Nmap e Packet Inspection con Wireshark su ambiente virtualizzato (Kali, Metasploitable, Windows XP, PFSense).

## 1. Configurazione dell'Ambiente (Network Topology)

Il laboratorio è stato configurato per simulare una scansione di rete attraverso un firewall/router.

- **Attacker (Kali Linux):** IP assegnato dinamicamente (es. 192.168.50.10).
- **Gateway/Firewall (PFSense):** 192.168.50.1 Gestisce il routing tra l'attaccante e i target.
- **Target 1 (Metasploitable 2):** 192.168.50.11 (Linux vulnerabile).
- **Target 2 (Windows XP):** 192.168.50.12 (Legacy Windows).

## 2. Analisi Target 1: Metasploitable 2 (192.168.50.11)

### Scenario A: OS Detection (nmap -O)

*Nota: Il comando nmap -o indicato nella traccia è incompleto (richiede un nome file) o è un typo per -O (OS Detection). Procedo con l'analisi come OS Detection -O.*

Comando:

```
sudo nmap -O 192.168.50.11
```

**Obiettivo:** Identificare il sistema operativo analizzando il comportamento dello stack TCP/IP (TTL, Window Size, flag options).

**Risultato Output:**

Plaintext

Device type: general purpose

Running: Linux 2.6.X

OS CPE: cpe:/o:linux:linux\_kernel:2.6

OS details: Linux 2.6.9 - 2.6.33

Nmap identifica correttamente la macchina come un kernel Linux obsoleto (tipico di Metasploitable).

## Scenario B: Analisi Comparata -sS vs -sT (Wireshark Analysis)

Questa è la fase critica del test: confrontare il comportamento a livello di pacchetto (Layer 4) tra una scansione "Stealth" e una "Connect". Wireshark è attivo in background su Kali.

### 1. TCP SYN Scan (Stealth)

**Comando:** `sudo nmap -sS 192.168.50.11`

Analisi Wireshark:

Nmap invia pacchetti SYN "raw". Non completa mai la connessione.

- **Kali:** Invia SYN verso la porta (es. 80).
- **Target (192.168.50.11):** Risponde con SYN, ACK (La porta è aperta).
- **Kali:** Risponde immediatamente con RST (Reset).

**Osservazione:** Il kernel di Kali non "sa" che Nmap ha inviato il pacchetto, quindi quando riceve il SYN/ACK, invia un RST per chiudere la connessione non sollecitata. Non avviene un 3-way handshake completo. È più veloce e genera meno log sul target.

### 2. TCP Connect Scan

**Comando:** `nmap -sT 192.168.50.11`

Analisi Wireshark:

Nmap utilizza le chiamate di sistema (OS syscalls) connect().

- **Kali:** Invia SYN.
- **Target (192.168.50.11):** Risponde con SYN, ACK.
- **Kali:** Risponde con ACK. (**Handshake completato**)

- **Kali:** Invia subito dopo RST o FIN per chiudere.

### Tabella Comparativa Rilevata

Fase	Nmap -sS (Half-Open)	Nmap -sT (Connect)
<b>Passo 1</b>	SYN	SYN
<b>Passo 2</b>	SYN/ACK	SYN/ACK
<b>Passo 3</b>	<b>RST</b>	<b>ACK</b> (Connessione stabilita)
<b>Visibilità</b>	Bassa (spesso non loggata dalle app)	Alta (loggata come connessione)
<b>Privilegi</b>	Richiede sudo (root)	Utente non privilegiato

## 3. Analisi Target 2: Windows XP (192.168.50.12)

Le stesse scansioni (-sS e -sT) sono state replicate su 192.168.50.12.

- **Wireshark:** Abbiamo notato che Windows XP risponde con un valore TTL (Time To Live) di default pari a **128**, mentre la Metasploitable (Linux) rispondeva con TTL vicino a **64**. Questo è un primo indicatore visivo del cambio di OS.

### Scenario C: Service & Version Detection (-sV)

L'ultimo passaggio richiede di identificare non solo le porte, ma cosa vi gira sopra esattamente.

Comando:

```
nmap -sV 192.168.50.12
```

#### Output Rilevante:

```
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows XP microsoft-ds
3389/tcp   open  ms-wbt-server Microsoft Terminal Services
Service Info: OSs: Windows, Windows XP; CPE:
cpe:/o:microsoft:windows_xp
```

Analisi Tecnica:

A differenza dello scan precedente, con -sV Nmap non si ferma all'handshake.

1. Completa il 3-way handshake.
2. Invia payload specifici (script LUA) per interrogare il servizio (Banner Grabbing).
3. Wireshark mostra uno scambio di dati effettivo (pacchetti PSH, ACK con payload) dopo la connessione, permettendo di leggere stringhe come "Microsoft Windows XP".

## Conclusioni

L'esercitazione ha dimostrato l'importanza di analizzare il traffico di rete durante le scansioni.

1. L'uso di **PFSense** come gateway non ha bloccato le scansioni (essendo nella stessa sottorete logica o regole permissive), ma in uno scenario reale potrebbe essere configurato per rilevare i troppi pacchetti SYN (IDS/IPS).
2. La differenza tra **-sS** e **-sT** è fondamentale per l'evasione dei log e la velocità.
3. **-sV** è il comando più rumoroso ma essenziale per determinare la superficie di attacco specifica (versioni vulnerabili di Windows XP).