

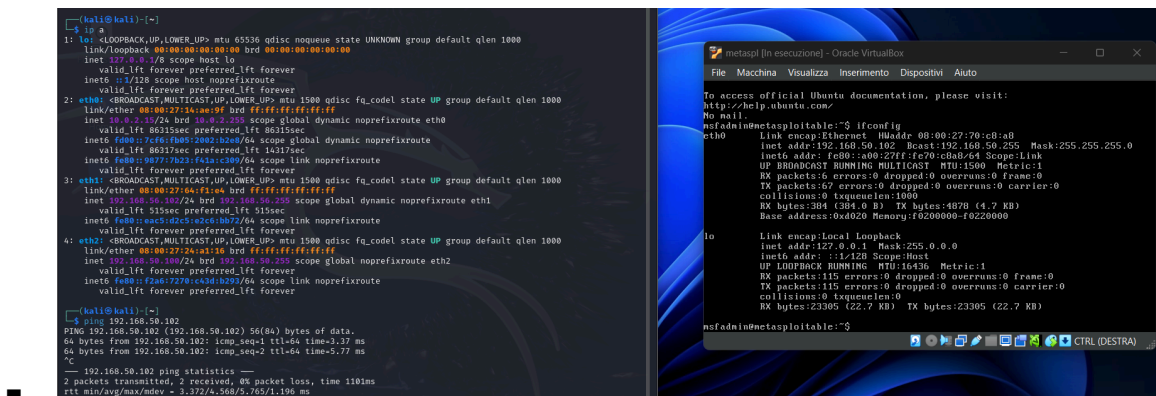
S11L3

La seguente relazione documenta i passaggi svolti per osservare e analizzare la stretta di mano TCP a 3 vie utilizzando gli strumenti **tcpdump** e Wireshark. Le macchine virtuali utilizzate sono Kali Linux (client) e Metasploitable (server), entrambe configurate per comunicare tramite rete Host-Only.

Preparazione dell'Ambiente

1. Configurazione della Rete:

- Kali Linux e Metasploitable sono state configurate sulla stessa rete Host-Only.
- Verifica della connessione con il comando **ping** da Kali verso Metasploitable.



Screenshot 1: Output del comando **ping** da Kali verso Metasploitable che conferma la connessione.

2. Installazione degli Strumenti:

- Entrambi gli strumenti, **tcpdump** e Wireshark, sono stati preinstallati su Kali Linux.

Parte 1: Analisi della Stretta di Mano TCP con Wireshark

1. Avvio di Wireshark:

- Wireshark è stato avviato su Kali Linux selezionando l'interfaccia di rete attiva
- La cattura del traffico è stata avviata generando traffico TCP con una connessione tra Kali e Metasploitable.

2. Filtraggio e Identificazione della Stretta di Mano TCP:

- Applicato il filtro **tcp** per isolare i pacchetti TCP e visualizzare chiaramente la sequenza della stretta di mano.
- Identificati i tre pacchetti chiave:
 - **SYN**: Pacchetto inviato dal client al server per avviare la connessione.
 - **SYN-ACK**: Pacchetto di risposta del server.
 - **ACK**: Pacchetto finale del client per completare la connessione.

18	227.952702674	192.168.50.100	192.168.50.102	TCP	74 50416 → 4444 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
19	227.954751835	192.168.50.102	192.168.50.100	TCP	74 4444 → 50416 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1
20	227.954808771	192.168.50.100	192.168.50.102	TCP	66 50416 → 4444 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=835380

Screenshot : Sequenza dei pacchetti SYN, SYN-ACK, e ACK visualizzata in Wireshark.

3. Analisi Dettagliata:

- Per ciascun pacchetto (SYN, SYN-ACK, ACK), sono stati analizzati i dettagli nel pannello inferiore di Wireshark:
 - Header TCP, flag (SYN, ACK), numeri di sequenza e porte.

19	181.312531531	192.168.50.102	192.168.50.100	TCP	74 4444 → 38212 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1
20	181.312562034	192.168.50.100	192.168.50.102	TCP	66 38212 → 4444 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=835745
21	181.321856895	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
22	182.324616034	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
23	183.326272332	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
24	186.325121528	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
25	186.474456538	PCSSystemtec_24:a1:...	PCSSystemtec_70:c8:...	ARP	42 Who has 192.168.50.102? Tell 192.168.50.100
26	186.476864819	PCSSystemtec_70:c8:...	PCSSystemtec_24:a1:...	ARP	60 192.168.50.102 is at 08:00:27:70:c8:a8
27	187.325926981	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
28	188.329693207	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
29	191.330947110	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
30	192.329911965	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
31	193.331536800	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
32	196.331196993	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
33	197.348879788	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
34	198.343022312	PCSSystemtec_70:c8:...	Broadcast	ARP	60 Who has 192.168.50.1? Tell 192.168.50.102
35	213.306628021	fe80::f2a6:7270:c43:ff02::2	ICMPv6	62 Router Solicitation	

Screenshot : Dettaglio di un pacchetto (es. SYN) con i campi chiave evidenziati.

Parte 2: Utilizzo di netcat per la Generazione del Traffico TCP

1. Avvio di un Listener su Metasploitable:

- Su Metasploitable è stato avviato un listener TCP utilizzando il comando:
nc -lvp 4444
 - Questo comando configura Metasploitable per ascoltare sulla porta 4444.

```

RX bytes:23305 (22.7 KB) TX bytes:0
msfadmin@metasploitable:~$ nc -lvp 4444
listening on [any] 4444 ...

```

Screenshot : Terminale di Metasploitable con il listener attivo.

2. Connessione da Kali Linux:

- Su Kali Linux, è stata stabilita una connessione verso Metasploitable utilizzando il comando:
nc [IP di Metasploitable] 4444

```

(kali@kali)-[~]
$ nc 192.168.50.102 4444

```

Screenshot : Terminale di Kali Linux che mostra la connessione stabilita con Metasploitable.

Parte 3: Cattura del Traffico con tcpdump

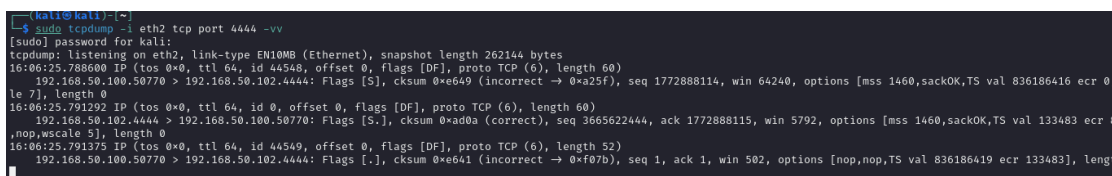
1. Avvio di tcpdump:

- Su Kali Linux, il comando seguente è stato utilizzato per catturare i pacchetti TCP sulla porta 4444:

`sudo tcpdump -i eth2 tcp port 4444 -vv`

- Questo comando ha permesso di catturare il traffico generato durante la connessione tra Kali e Metasploitable.

■ Screenshot



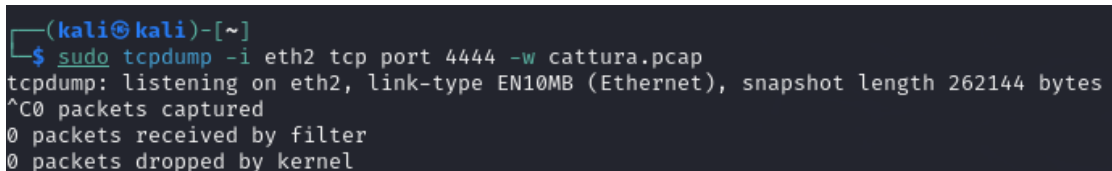
```
(kali@kali)-[~]
└─$ sudo tcpdump -i eth2 tcp port 4444 -vv
[sudo] password for kali:
tcpdump: listening on eth2, link-type EN10MB (Ethernet), snapshot length 262144 bytes
16:06:25.788600 IP (tos 0x0, ttl 64, id 44548, offset 0, flags [DF], proto TCP (6), length 60)
  192.168.50.100.50770 > 192.168.50.102.4444: Flags [S], cksum 0xe649 (incorrect -> 0xa25f), seq 1772888114, win 64240, options [mss 1460,sackOK,TS val 836186416 ecr 0], length 0
16:06:25.791292 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
  192.168.50.102.4444 > 192.168.50.100.50770: Flags [S.], cksum 0xad0a (correct), seq 3665622444, ack 1772888115, win 5792, options [mss 1460,sackOK,TS val 133483 ecr 0], length 0
16:06:25.791375 IP (tos 0x0, ttl 64, id 44549, offset 0, flags [DF], proto TCP (6), length 52)
  192.168.50.100.50770 > 192.168.50.102.4444: Flags [.], cksum 0xe641 (incorrect -> 0xf07b), seq 1, ack 1, win 502, options [nop,nop,TS val 836186419 ecr 133483], length 0
```

: Terminale di Kali con tcpdump in esecuzione e traffico TCP visibile in tempo reale.

2. Salvataggio dei Pacchetti:

- I pacchetti catturati sono stati salvati in un file `.pcap` utilizzando il comando:

`sudo tcpdump -i eth2 tcp port 4444 -w cattura.pcap`

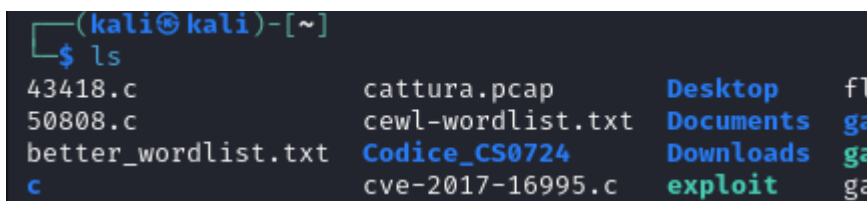


```
(kali@kali)-[~]
└─$ sudo tcpdump -i eth2 tcp port 4444 -w cattura.pcap
tcpdump: listening on eth2, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C0 packets captured
0 packets received by filter
0 packets dropped by kernel
```

■ **Screenshot** : Terminale di Kali che mostra il comando di salvataggio del file `.pcap`.

- Verificata l'esistenza del file con il comando `ls`:

`ls`



```
(kali@kali)-[~]
└─$ ls
43418.c          cattura.pcap      Desktop          fl
50808.c          cewl-wordlist.txt Documents        ga
better_wordlist.txt Codice_CS0724     Downloads       ga
c                cve-2017-16995.c exploit          ga
```

■ **Screenshot** : Directory di Kali che mostra il file `cattura.pcap` salvato.

Risultati

- **Wireshark** ha mostrato la sequenza completa della stretta di mano TCP (SYN, SYN-ACK, ACK) con un'analisi dettagliata di ciascun pacchetto.
- **tcpdump** ha catturato correttamente il traffico TCP, dimostrando la capacità di utilizzare strumenti da riga di comando per analisi di rete.
- Il file `.pcap` è stato salvato con successo, ma non è stato ulteriormente analizzato.

Conclusioni

Questo laboratorio ha mostrato come catturare e analizzare il traffico TCP utilizzando due strumenti fondamentali:

- **Wireshark** per un'analisi approfondita e visuale dei dettagli del protocollo TCP.
- **tcpdump** per catturare pacchetti direttamente da riga di comando e salvarli per analisi successive.

Il processo di osservazione della stretta di mano TCP è stato completato con successo, dimostrando la capacità di combinare strumenti grafici e da riga di comando per analisi pratiche di rete.