

# TRAFFIC DNS

Usare Wireshark per Osservare Traffico DNS

**Obiettivi:**

- **Parte 1: Preparare gli Host per Catturare il Traffico**
- **Parte 2: Analizzare i Pacchetti usando Wireshark**
- **Parte 3: Visualizzare i Pacchetti usando tcpdump**

## SCENARIO

Wireshark è uno strumento open source per la cattura e l'analisi dei pacchetti. Wireshark fornisce una scomposizione dettagliata dello stack dei protocolli di rete. Wireshark permette di filtrare il traffico per la risoluzione dei problemi di rete, investigare problemi di sicurezza e analizzare i protocolli di rete. Poiché Wireshark permette di visualizzare i dettagli dei pacchetti, può essere usato come strumento di riconoscimento da un attaccante. In questo laboratorio, installerai Wireshark e lo userai per filtrare i pacchetti DNS e visualizzare i dettagli sia dei pacchetti di query DNS che di quelli di risposta.

# PARTE 1: CATTURARE IL TRAFFICO DNS

## 1. Pulizia della Cache DNS su Kali Linux

A differenza di Windows, Kali (e Linux in generale) non ha sempre un servizio di caching DNS attivo di default, a meno che non siano installati servizi come `systemd-resolved`, `nscd`, o

- Installa: `sudo apt install systemd-resolved`
- Controlla: `sudo systemctl status systemd-resolved`
- Pulisci: `sudo resolvectl flush-caches`

```
(kali㉿kali)-[~]
└─$ sudo systemctl status systemd-resolved
● systemd-resolved.service - Network Name Resolution
    Loaded: loaded (/usr/lib/systemd/system/systemd-resolved.service; enabled; preset: enabled)
      Active: active (running) since Wed 2026-02-18 17:01:23 CET; 1min 4s ago
        Docs: man:systemd-resolved.service(8)
               man:org.freedesktop.resolve1(5)
               https://systemd.io/Writing_NETWORK_CONFIGURATION_MANAGERS
               https://systemd.io/Writing_RESOLVER_CLIENTS
      Main PID: 476 (systemd-resolve)
        Status: "Processing requests..."
         Tasks: 1 (limit: 6957)
        Memory: 9.9M (peak: 10.4M)
          CPU: 124ms
        CGroup: /system.slice/systemd-resolved.service
                └─476 /usr/lib/systemd/systemd-resolved

Feb 18 17:01:23 kali systemd[1]: Starting systemd-resolved.service - Network Name Resolution ...
Feb 18 17:01:23 kali systemd-resolved[476]: Positive Trust Anchors:
Feb 18 17:01:23 kali systemd-resolved[476]: . IN DS 20326 8 2 e06d4b80b0f1d39a95c0b0d7c65d08458e88040
Feb 18 17:01:23 kali systemd-resolved[476]: . IN DS 38696 8 2 683d2d0acb8c9b712a1948b27f741219298d0a45
Feb 18 17:01:23 kali systemd-resolved[476]: Negative trust anchors: home.arpa 10.in-addr.arpa 16.172.1.
Feb 18 17:01:23 kali systemd-resolved[476]: Using system hostname 'kali'.
Feb 18 17:01:23 kali systemd[1]: Started systemd-resolved.service - Network Name Resolution.
Feb 18 17:01:24 kali systemd-resolved[476]: eth0: Bus client set default route setting: yes
Feb 18 17:01:24 kali systemd-resolved[476]: eth0: Bus client set DNS server list to: 8.8.8.8
Feb 18 17:01:24 kali systemd-resolved[476]: eth0: Bus client set DNS server list to: 8.8.8.8, 10.0.2.3

(kali㉿kali)-[~]
└─$ sudo resolvectl flush-caches
```

Inserire il nome di dominio di un sito web. Il nome di dominio www.cisco.com è usato in questo esempio.

```
> www.cisco.com
Server:          8.8.8.8
Address:         8.8.8.8#53

Non-authoritative answer:
www.cisco.com canonical name = www.cisco.com.akadns.net.
www.cisco.com.akadns.net canonical name = wwwds.cisco.com.edgekey.net.
wwwds.cisco.com.edgekey.net canonical name = wwwds.cisco.com.edgekey.net.g
lobalredir.akadns.net.
wwwds.cisco.com.edgekey.net.globalredir.akadns.net      canonical name = e2867
.dsca.akamaiedge.net.
Name:           e2867.dsca.akamaiedge.net
Address:        23.60.188.118
Name:           e2867.dsca.akamaiedge.net
Address:        2a02:26f0:2d80:699::b33
Name:           e2867.dsca.akamaiedge.net
Address:        2a02:26f0:2d80:691::b33
> exit
```

Fare clic su Stop capturing packets (Interrompi cattura pacchetti) per fermare la cattura di Wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	8.8.8.8	DNS	84	Standard query 0x85ac A www.cisco.com OPT
2	0.019141686	8.8.8.8	10.0.2.15	DNS	266	Standard query response 0x85ac A www.cisco.com CNAME www.cisco.com.akadns.net CNAME wwwds.cisco.com.edgekey.net CNAME wwwds.globalredir.akadns.net
3	0.021346640	10.0.2.15	8.8.8.8	DNS	96	Standard query 0xe684 AAAA e2867.dsca.akamaiedge.net OPT
4	0.041230231	8.8.8.8	10.0.2.15	DNS	152	Standard query response 0xe684 AAAA e2867.dsca.akamaiedge.net AAAA 2a02:26f0:2d80:699::b33 AAAA 2a02:26f0:2d80:691::b33 OPT

## PARTE 2: ESPLORARE IL TRAFFICO DELLE QUERY DNS

Analizzare i Pacchetti usando Wireshark

Applicare un filtro alla cattura salvata.

- Osservare il traffico catturato nel riquadro Elenco Pacchetti  Packet List) di Wireshark. Inserire udp.port == 53 nella casella del filtro e fare clic sulla freccia (o premere invio) per visualizzare solo i pacchetti DNS.
- Selezionare il pacchetto DNS che contiene Standard query e A www.cisco.com nella colonna Info.
- Nel riquadro Dettagli Pacchetto  Packet Details), notare che questo pacchetto ha Ethernet II, Internet Protocol Version 4, User Datagram Protocol e Domain Name System (query).
- Espandere Ethernet II per visualizzare i dettagli. Osservare i campi di origine e destinazione.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	10.0.2.15	8.8.8.8	DNS	84	Standard query 0x85ac A www.cisco.com OPT
2	0.019141686	8.8.8.8	10.0.2.15	DNS	266	Standard query response 0x85ac A www.cisco.com CNAME
3	0.021346640	10.0.2.15	8.8.8.8	DNS	96	Standard query 0xe684 AAAA e2867.dsca.akamaiedge.net
4	0.041230231	8.8.8.8	10.0.2.15	DNS	152	Standard query response 0xe684 AAAA e2867.dsca.akamaiedge.net

```
Frame 1: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_2f:3e:2f (08:00:27:2f:3e:2f), Dst: 52:55:0a:00:02:02 (52:55:0a:00:02:02)
  Destination: 52:55:0a:00:02:02 (52:55:0a:00:02:02)
    .... ..1. .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
    .... ..0. .... .... .... = IG bit: Individual address (unicast)
  Source: PCSSystemtec_2f:3e:2f (08:00:27:2f:3e:2f)
    .... ..0. .... .... .... = LG bit: Globally unique address (factory default)
    .... ..0. .... .... .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
  IP Stream index: 0
```

1. Quali sono gli indirizzi MAC di origine e destinazione?

**MAC di origine (Source): 08:00:27:2f:3e:2f (identificato come PCSSystemtec).**

**MAC di destinazione (Destination): 52:55:0a:00:02:02.**

2. A quali interfacce di rete sono associati questi indirizzi MAC?

**MAC di origine (08:00:27:2f:3e:2f): È associato all'interfaccia eth0 del PC Kali Linux.**

**MAC di destinazione (52:55:0a:00:02:02): È associato all'interfaccia del Gateway/Router locale, che riceve il pacchetto per inoltrarlo verso Internet.**

Espandere Internet Protocol Version 4. Osservare gli indirizzi IPv4 di origine e destinazione.

```
▼ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 8.8.8.8
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  0000 00.. = Differentiated Services Codepoint: Default (0)
  .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 70
  Identification: 0xef3f (61247)
▼ 000. .... = Flags: 0x0
  0... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 64
  Protocol: UDP (17)
  Header Checksum: 0x6f49 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.0.2.15
  Destination Address: 8.8.8.8
  [Stream index: 0]
```

### 3. Quali sono gli indirizzi IP di origine e destinazione?

Espandendo la sezione **Internet Protocol Version 4** dell'immagine, si identificano i seguenti indirizzi:

**Indirizzo IP di origine (Source Address):** 8.8.8.8.

**Indirizzo IP di destinazione (Destination Address):** 10.0.2.15.

### 4. A quali interfacce di rete sono associati questi indirizzi IP?

Dato che questo pacchetto è una risposta DNS ("Standard query response") che arriva dall'esterno verso la nostra macchina, le interfacce sono così suddivise:

**IP di origine (8.8.8.8):** Questo indirizzo è associato a un **server DNS pubblico di Google** su Internet. Nel contesto della cattura, rappresenta l'endpoint remoto che ha risolto la query per www.cisco.com.

**IP di destinazione (10.0.2.15):** Questo indirizzo è associato all'interfaccia di rete locale della **macchina virtuale Kali**. Nello specifico, come visto nel livello Ethernet precedente, l'interfaccia è identificata come eth0.

**Note:**

il campo **Protocol** è impostato su **UDP (17)**. A differenza dei pacchetti TCP visti nei test precedenti, UDP è un protocollo "connectionless", il che lo rende ideale per query rapide come quelle DNS, ma anche più suscettibile a tecniche di spoofing se non protetto correttamente.

f. Espandere User Datagram Protocol ✓ UDP★ . Osservare le porte di origine e destinazione.

```
User Datagram Protocol, Src Port: 53, Dst Port: 39657
  Source Port: 53
  Destination Port: 39657
  Length: 232
  Checksum: 0xd45a [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
  [Stream Packet Number: 2]
  ▾ [Timestamps]
    [Time since first frame: 0.019141686 seconds]
    [Time since previous frame: 0.019141686 seconds]
  UDP payload (224 bytes)
```

5. Quali sono le porte di origine e destinazione?

**Porta di origine (Source Port):** 53.

**Porta di destinazione (Destination Port):** 39657.

6. Qual è il numero di porta DNS predefinito?

**Numero di porta DNS predefinito:** La porta predefinita per il protocollo DNS è la **53**.

Poiché si tratta di una risposta inviata dal server DNS (8.8.8.8) verso il nostro computer (10.0.2.15), la porta di origine è la 53 (il servizio DNS stesso) e la porta di destinazione è la porta effimera generata dal nostro client per questa specifica richiesta.

Determinare l'indirizzo IP e MAC del PC.

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
          inet6 fe80::9ecc:d189:8258:4ad0 prefixlen 64 scopeid 0x20<link>
            inet6 fd17:625c:f037:2:ebab:5:5710:dc87:371e prefixlen 64 scopeid 0x0<global>
l0:
      ether 08:00:27:2f:3e:2f txqueuelen 1000 (Ethernet)
        RX packets 9339 bytes 11522221 (10.9 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 2867 bytes 592522 (578.6 KiB)
        TX errors 0 dropped 21 overruns 0 carrier 0 collisions 0

eth1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether 08:00:27:3c:91:8c txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
          inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 237 bytes 27687 (27.0 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 237 bytes 27687 (27.0 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## 7. Confrontare gli indirizzi MAC e IP nei risultati di Wireshark con gli indirizzi IP e MAC.

Qual è la tua osservazione?

**Indirizzo IP Locale:** L'interfaccia eth0 mostra un indirizzo inet 10.0.2.15. Questo valore corrisponde esattamente all'**Indirizzo IP di destinazione** (Destination Address) visualizzato nei dettagli del protocollo IPv4 di Wireshark per il pacchetto di risposta DNS.

**Indirizzo MAC Locale:** L'interfaccia eth0 riporta un indirizzo fisico ether 08:00:27:2f:3e:2f. Questo valore coincide con l'**Indirizzo MAC di destinazione** visualizzato nei dettagli Ethernet II di Wireshark (identificato anche come PCSSystemtec\_2f:3e:2f).

L'osservazione principale è che i dati catturati da Wireshark confermano l'identità della macchina locale come destinataria della comunicazione.

## PARTE 3: ESPLORARE IL TRAFFICO DELLE RISPOSTE DNS

a. Selezionare il corrispondente pacchetto DNS di risposta che ha Standard query response e A www.cisco.com nella colonna Info

8. Quali sono gli indirizzi MAC e IP e i numeri di porta di origine e destinazione?

Analizzando le sezioni Ethernet, IP e UDP del Frame 2:

**Indirizzi MAC:**

**Origine (Source):** 52:55:0a:00:02:02.

**Destinazione (Destination):** 08:00:27:2f:3e:2f (PCSSystemtec).

**Indirizzi IP:**

**Origine (Source Address):** 8.8.8.8.

**Destinazione (Destination Address):** 10.0.2.15.

**Numeri di Porta:**

**Origine (Source Port):** 53.

**Destinazione (Destination Port):** 39657.

9. Come si confrontano con gli indirizzi nei pacchetti di query DNS?

Confrontando questi dati con quelli del pacchetto di query (Frame 1), si osserva quanto segue:

**Inversione dei Ruoli:** Gli indirizzi e le porte di origine nel pacchetto di query diventano i parametri di destinazione nel pacchetto di risposta, e viceversa.

**MAC Address:** Nella query, il MAC del PC era l'origine; nella risposta, è diventato la destinazione. L'indirizzo 08:00:27:2f:3e:2f corrisponde all'interfaccia eth0 del PC.

**Indirizzi IP:** L'indirizzo IP del PC (10.0.2.15) era l'origine della query verso il server di Google (8.8.8.8). Nella risposta, 8.8.8.8 invia le informazioni richieste a 10.0.2.15.

**Porte UDP:** La query è stata inviata alla porta predefinita 53 partendo dalla porta effimera 39657. La risposta parte dalla porta 53 per tornare alla porta 39657 del PC, permettendo all'applicazione (nslookup) di identificare la sessione corretta.

b. Espandere Domain Name System (response). Quindi espandere Flags, Queries, e Answers. c. Osservare i risultati.

```
udp.port == 53
No. Time Source Destination Protocol Length Info
1 0.000000000 10.0.2.15 8.8.8.8 DNS 84 Standard query 0x85ac A www.cisco.com OPT
2 0.0191411686 8.8.8.8 10.0.2.15 DNS 266 Standard query response 0x85ac A www.cisco.com CN
3 0.021346640 10.0.2.15 8.8.8.8 DNS 96 Standard query 0xe684 AAAA e2867.dsca.akamaiedge.l
4 0.041230231 8.8.8.8 10.0.2.15 DNS 152 Standard query response 0xe684 AAAA e2867.dsca.ak

.... .0 .... .... .... = IG bit: Individual address (unicast)
+ Source: 52:55:0a:00:02:02 (52:55:0a:00:02:02)
.... .1 .... .... .... = LG bit: Locally administered address (this is NOT the factory def...
.... .0 .... .... .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
[Stream index: 0]
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 10.0.2.15
User Datagram Protocol, Src Port: 53, Dst Port: 39657
Domain Name System (response)
Transaction ID: 0x85ac
Flags: 0x8180 Standard query response, No error
1.... .... .... = Response: Message is a response
.000 0.... .... = Opcode: Standard query (0)
.... 0.... .... = Authoritative: Server is not an authority for domain
.... .1 .... .... = Recursion desired: Do query recursively
.... .1.... .... = Recursion available: Server can do recursive queries
.... 0.... .... = Z: reserved (0)

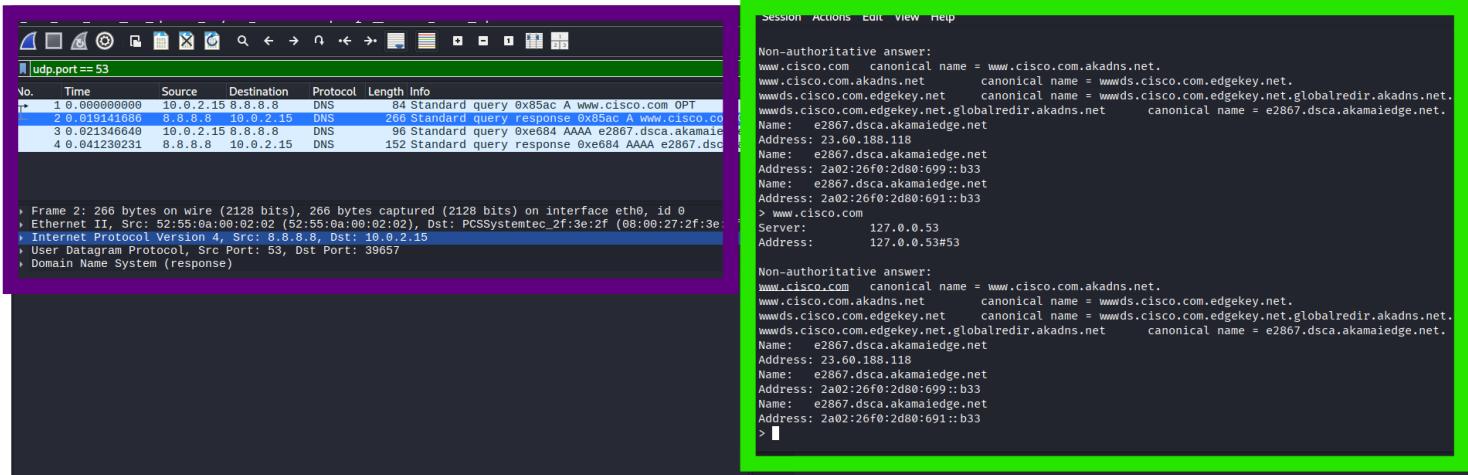
Questions: 1
Answer RRs: 5
Authority RRs: 0
Additional RRs: 1
Queries
www.cisco.com: type A, class IN
Name: www.cisco.com
[Name Length: 13]
[Label Count: 3]
Type: A (1) (Host Address)
Class: IN (0x0001)
Answers
www.cisco.com: type CNAME, class IN, cname www.cisco.com.akadns.net
Name: www.cisco.com
Type: CNAME (5) (Canonical NAME for an alias)
Class: IN (0x0001)
Time to live: 2399 (39 minutes, 59 seconds)
Data length: 26
CNAME: www.cisco.com.akadns.net
www.cisco.com.akadns.net: type CNAME, class IN, cname wwwds.cisco.com.edgekey.net
Name: www.cisco.com.akadns.net
Type: CNAME (5) (Canonical NAME for an alias)
Class: IN (0x0001)
Time to live: 2399 (39 minutes, 59 seconds)
Data length: 26
CNAME: wwwds.cisco.com.edgekey.net
```

## 10. Il server DNS può fare query ricorsive?

Sì, il server DNS può eseguire query ricorsive.

Questo è confermato dal campo **Flags** nella sezione Domain Name System (response), dove il bit relativo a **Recursion available** è impostato a 1 (.... .1.... .... = Recursion available: Server can do recursive queries). Inoltre, si nota che il client ha esplicitamente richiesto la ricorsione tramite il flag **Recursion desired** impostato a 1.

d. Osservare i record CNAME e A nei dettagli delle Risposte (✓ Answers).



### 11. Come si confrontano i risultati con quelli di nslookup?

In base all'analisi dei dati forniti, i record **CNAME** e **A** visualizzati nei dettagli della risposta DNS di Wireshark corrispondono esattamente ai risultati ottenuti tramite il comando nslookup nel terminale.

**Corrispondenza degli Alias (CNAME):** Entrambi gli strumenti mostrano la medesima catena di reindirizzamenti per risolvere www.cisco.com.  
www.cisco.com punta a www.cisco.com.akadns.net.

Quest'ultimo punta a wwwds.cisco.com.edgekey.net.  
Infine, viene risolto l'alias verso e2867.dsca.akamaiedge.net.

**Corrispondenza dell'Indirizzo IP (Record A):** Sia Wireshark che nslookup identificano l'indirizzo IPv4 finale come **23.60.188.118**.

**Osservazione sulla Visualizzazione:** Mentre nslookup presenta una sintesi testuale pulita (identificata come "Non-authoritative answer"), Wireshark fornisce dettagli aggiuntivi per ogni salto della catena, come il valore **Time to Live (TTL)** e la lunghezza dei dati, permettendo un'analisi tecnica più profonda.

## Riflessione:

### 12. Dai risultati di Wireshark, cos'altro puoi imparare sulla rete quando rimuovi il filtro?

Rimuovendo il filtro (come `udp.port == 53`), Wireshark mostra l'intero flusso di traffico che attraversa l'interfaccia di rete. Da questa visione d'insieme puoi apprendere diversi dettagli critici:

- **Dispositivi Attivi e Topologia:** Puoi identificare altri indirizzi IP e MAC presenti nel segmento di rete, come l'indirizzo 172.16.0.40 che interagisce con la tua macchina.
- **Protocolli di Servizio:** Vedresti pacchetti **ARP** (per la risoluzione degli indirizzi MAC), **ICMP** (ping), o traffico di broadcast come **DHCP** o **MDNS** che rivelano la presenza di stampanti, server o altri host.
- **Dettagli delle Sessioni Applicative:** Mentre prima vedevi solo la risoluzione del nome, senza filtri puoi osservare l'intera sequenza di connessione, come il **Three-way Handshake TCP** (SYN, SYN-ACK, ACK) e le successive richieste **HTTP GET**.
- **Identificazione dei Sistemi Operativi:** Analizzando parametri come il **Time to Live (TTL)** (che nel nostro caso è 64) o la dimensione della finestra TCP, un analista può dedurre se il sistema remoto è Linux, Windows o un dispositivo IoT.

### 13. Come può un attaccante usare Wireshark per compromettere la sicurezza della tua rete?

Per un attaccante, Wireshark è uno strumento di **Reconnaissance** (ricognizione) e **Sniffing** estremamente potente:

- **Intercettazione di Credenziali in Chiaro:** Se sulla rete transita traffico non crittografato (come HTTP, FTP o Telnet), l'attaccante può leggere direttamente username e password all'interno dei pacchetti.
- **Network Mapping Silenzioso:** Essendo uno strumento passivo, l'attaccante può mappare tutti i server e i servizi attivi sulla rete senza inviare un singolo pacchetto, rendendo l'attività invisibile ai comuni sistemi di rilevamento intrusioni (IDS).

- **Session Hijacking:** Monitorando i numeri di sequenza (**Sequence Numbers**) dei pacchetti TCP, un attaccante potrebbe tentare di iniettare pacchetti malevoli in una sessione esistente per dirottarla.
- **Analisi delle Vulnerabilità:** Vedendo le intestazioni dei pacchetti, l'attaccante può identificare versioni specifiche di software o server web (tramite i banner HTTP) e cercare exploit mirati per quelle versioni.