

Università degli Studi di Salerno



Dipartimento di Informatica

# Penetration Testing & Ethical Hacking

---

## Enumerating Target e Port Scanning

### Parte 1

Arcangelo Castiglione  
arcastiglione@unisa.it

# Outline

---

- Concetti Introduttivi
- Suite Protocollore TCP/IP
- Formato dei Messaggi TCP e UDP
- Active Enumeration
  - Network Scanner Nmap
  - Zenmap
  - Unicornscan
- Passive Enumeration
  - Shodan
  - ZoomEye
  - Censys
  - FOFA

# Outline

---

- **Concetti Introduttivi**
- Suite Protocollore TCP/IP
- Formato dei Messaggi TCP e UDP
- Active Enumeration
  - Network Scanner Nmap
  - Zenmap
  - Unicornscan
- Passive Enumeration
  - Shodan
  - ZoomEye
  - Censys
  - FOFA

# Enumerating Target

## Obiettivi e Motivazioni

---

- Fase tipicamente eseguita dopo aver individuato le macchine target attive (e raggiungibili) appartenenti all'asset
- Permette di acquisire ulteriori informazioni sulle macchine target
  - Stato delle porte
  - Protocolli e servizi di rete
  - Applicativi dei servizi
  - Sistemi Operativi
  - Etc

# Enumerating Target

## Obiettivi e Motivazioni

---

- Acquisire (**enumerare**) quante più informazioni possibili sui servizi di rete erogati dalle macchine target attive (***Target Enumeration***)
  - Informazioni che potranno successivamente essere utilizzate per individuare le vulnerabilità relative a questi servizi
- Ciascun **servizio** disponibile sulla macchina target è **erogato** tramite una determinata **porta**

# Enumerating Target

## Active vs. Passive Enumeration

---

### ➤ Due forme di **Target Enumeration**

#### ➤ **Active Enumeration**

- I metodi di enumerazione attiva richiedono un'interazione diretta con la macchina target
  - Mediante *Port Scanning*

#### ➤ **Passive Enumeration**

- I metodi di enumerazione passiva permettono di ottenere informazioni sulla macchina target senza interagire direttamente con essa
  - Utilizzando *Servizi di Terze Parti*

# Active Enumerating Target

## Port Scanning – Stato di una Porta

---

- Il *Port Scanning* è il metodo tramite cui è possibile determinare lo stato delle porte appartenenti ai seguenti protocolli di rete
  - **T**ransmission **C**ontrol **P**rotocol (**TCP**)
  - **U**ser **D**atagram **P**rotocol (**UDP**)

Una **porta** associata ad un certo servizio di rete **può essere**

---

# Active Enumerating Target

## Port Scanning – Stato di una Porta

---

- Il *Port Scanning* è il metodo tramite cui è possibile determinare lo stato delle porte appartenenti ai seguenti protocolli di rete
  - **T**ransmission **C**ontrol **P**rotocol (**TCP**)
  - **U**ser **D**atagram **P**rotocol (**UDP**)

Una **porta** associata ad un certo servizio di rete **può essere**



**APERTA**

Indica che il servizio è accessibile ed è in modalità di *Listening*



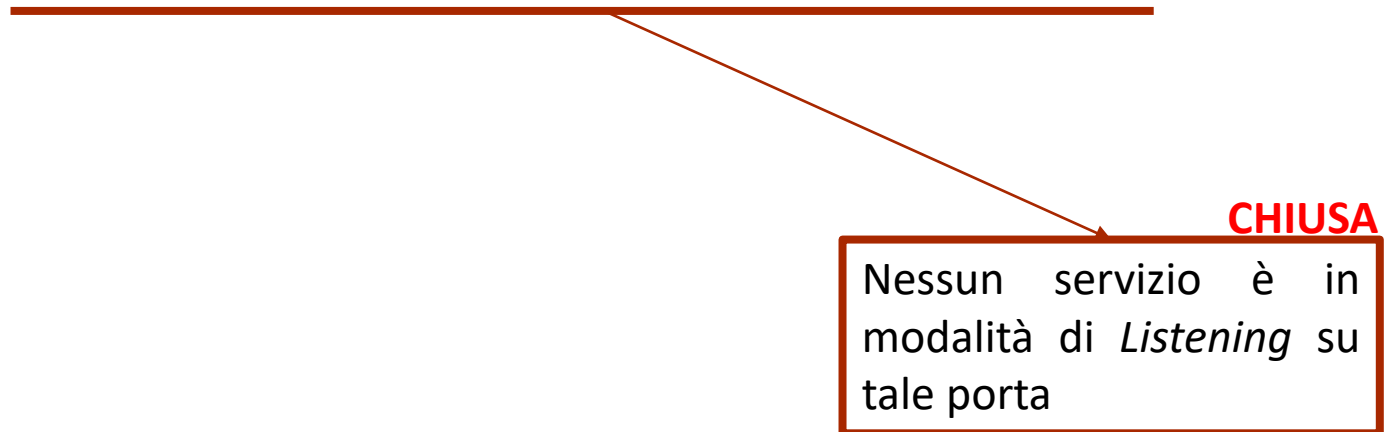
# Active Enumerating Target

## Port Scanning – Stato di una Porta

---

- Il *Port Scanning* è il metodo tramite cui è possibile determinare lo stato delle porte appartenenti ai seguenti protocolli di rete
  - **T**ransmission **C**ontrol **P**rotocol (**TCP**)
  - **U**ser **D**atagram **P**rotocol (**UDP**)

Una **porta** associata ad un certo servizio di rete **può essere**



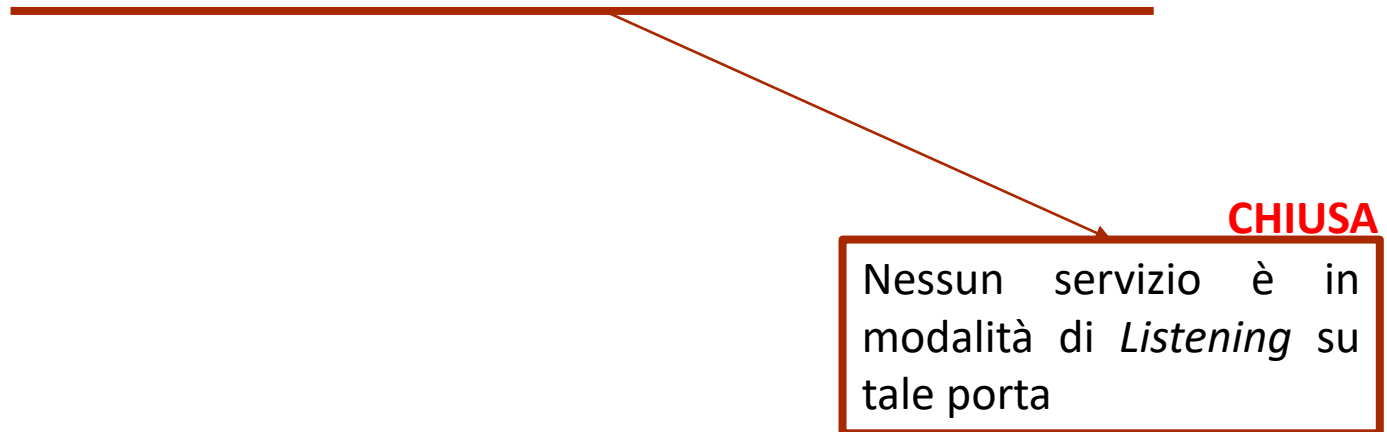
# Active Enumerating Target

## Port Scanning – Stato di una Porta

---

- Il *Port Scanning* è il metodo tramite cui è possibile determinare lo stato delle porte appartenenti ai seguenti protocolli di rete
  - **T**ransmission **C**ontrol **P**rotocol (**TCP**)
  - **U**ser **D**atagram **P**rotocol (**UDP**)

Una **porta** associata ad un certo servizio di rete **può essere**



Tuttavia, una porta potrebbe anche essere «**FILTRATA**»

# Active Enumerating Target

## Port Scanning – Porte e Vulnerabilità

---

- Dopo aver individuato lo **stato** di una porta il pentester potrebbe anche **controllare la versione del software** utilizzato dal **servizio di rete** erogato da tale porta
- Al fine di individuare eventuali vulnerabilità per tale servizio



# Active Enumerating Target

## Port Scanning – Porte e Vulnerabilità

---

### ➤ Esempio

- Una macchina target dispone di un Web Server il cui software è nella versione 1.0
- Sono presenti vulnerabilità note in tale versione del software
- Un utente malintenzionato potrebbe sfruttare tali vulnerabilità per attaccare il Web Server



# Outline

---

- Concetti Introduttivi
- **Suite Protocollore TCP/IP**
- Formato dei Messaggi TCP e UDP
- Active Enumeration
  - Network Scanner Nmap
  - Zenmap
  - Unicornscan
- Passive Enumeration
  - Shodan
  - ZoomEye
  - Censys
  - FOFA

# Suite Protocollore TCP/IP

## Caratteristiche

---

- Suite che include diversi protocolli (*suite protocollore*), i più importanti dei quali sono il protocollo **TCP** (Transmission Control Protocol) ed il protocollo **IP** (Internet Protocol)
  - **IP** si occupa principalmente dell'indirizzamento e del routing dei datagram
  - **TCP** è responsabile della gestione delle connessioni e dell'affidabilità del trasporto tra due endpoint
- **IP** è localizzato nel *Livello di Rete (Layer 3)* del modello *ISO/OSI*
- **TCP** è localizzato nel *Livello di Trasporto (Layer 4)* del modello *ISO/OSI*

# Suite Protocollore TCP/IP

## Caratteristiche

---

- Suite che include diversi protocolli (*suite protocollore*), i più importanti dei quali sono il protocollo **TCP** (Transmission Control Protocol) ed il protocollo **IP** (Internet Protocol)
  - **IP** si occupa principalmente dell'indirizzamento e del routing dei datagram
  - **TCP** è responsabile della gestione delle connessioni e dell'affidabilità del trasporto tra due endpoint
- **IP** è localizzato nel *Livello di Rete (Layer 3)* del modello *ISO/OSI*
- **TCP** è localizzato nel *Livello di Trasporto (Layer 4)* del modello *ISO/OSI*

**A livello di trasporto esiste anche il protocollo *UDP*, che analizzeremo successivamente**

# Suite Protocollore TCP/IP

## Caratteristiche TCP

---

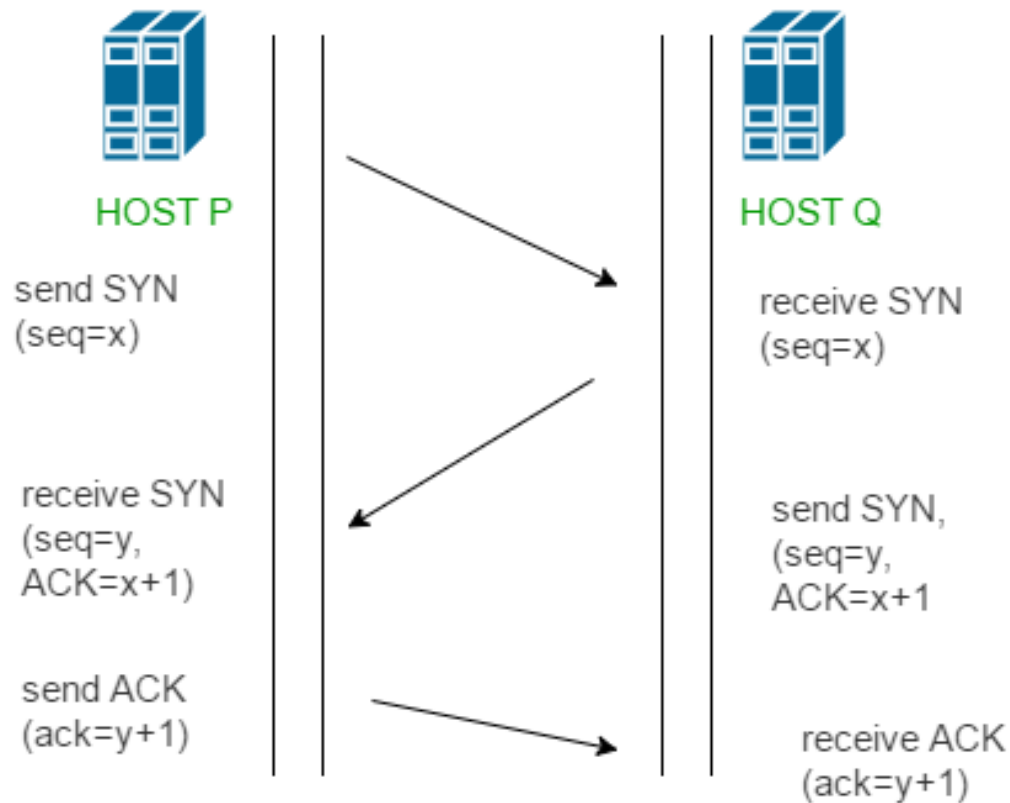
- Le caratteristiche principali del protocollo *TCP* sono le seguenti
  - **Orientato alla connessione – 1/2**
    - Prima che Client e Server possano comunicare devono **stabilire una connessione** utilizzando un protocollo chiamato ***three-way handshake***
    - Il Client inizializza la connessione inviando al Server
      - Un pacchetto contenente un **SYN** (*SYNchronize*) flag
      - Un numero iniziale di sequenza (***Initial Sequence Number – ISN***) scelto a caso
    - Il Server risponde al Client inviando
      - Un **SYN** contenente un nuovo **ISN**
      - Un **ACK** (*ACKnowledgment*) relativo al pacchetto **SYN** che ha ricevuto dal Client, il cui contenuto è dato da **ISN (del client) + 1**
    - Il Client risponde al Server con un **ACK** contenente **ISN (del Server) + 1**
  - A questo punto, la connessione è stabilita



# Suite Protocollore TCP/IP

## Caratteristiche TCP

---



# Suite Protocollore TCP/IP

## Caratteristiche TCP

---

- Le caratteristiche principali del protocollo *TCP* sono le seguenti
  - **Orientato alla connessione – 2/2**
    - Per **terminare la connessione**, *TCP* utilizza il seguente meccanismo
      - Il Client invia al Server un pacchetto con un **FIN** (*FINish*) flag
      - Il Server invia un pacchetto di **ACK** al Client così da informarlo della ricezione del pacchetto **FIN**
      - Quando il Server è pronto a chiudere la connessione invia al Client un pacchetto **FIN**
      - Il Client invia un **ACK** al Server per indicargli che ha ricevuto il suo pacchetto **FIN**

**N.B.** Generalmente, sia Client che Server possono terminare la connessione, mediante l'invio del pacchetto **FIN**

# Suite Protocollore TCP/IP

## Caratteristiche TCP

---

- Le caratteristiche principali del protocollo *TCP* sono le seguenti
  - **Protocollo Affidabile**
    - *TCP* utilizza numeri di sequenza ed **ACK** per identificare i pacchetti
    - Il ricevente invia un **ACK** per indicare che ha ricevuto il pacchetto
    - Quando un pacchetto va perso, *TCP* lo re-invierà automaticamente se non avrà ricevuto un **ACK** dal ricevente
    - Se i pacchetti non dovessero arrivare in ordine, *TCP* provvederà a riordinarli prima di inoltrarli al livello applicativo
  - I protocolli che trasmettono file o dati importanti tipicamente usano *TCP*

# Suite Protocollore TCP/IP

## Caratteristiche UDP

---

- Le caratteristiche principali del protocollo *UDP* sono le seguenti
  - **Protocollo senza connessione**
    - Per scambiarsi dati, Client e Server non devono prima stabilire una connessione
    - *UDP* «farà del suo meglio» per inviare i dati a destinazione, ma nel caso di perdite di pacchetti non provvederà a ritrasmetterli
  - **Utilizzato**
    - Nello streaming video ed in applicazioni multimediali, dove è tollerata una certa perdita di dati
    - Ma anche da protocolli quali *Domain Name System (DNS)*, *Dynamic Host Configuration Protocol (DHCP)* e *Simple Network Management Protocol (SNMP)*

# Suite Protocollore TCP/IP

## Le Porte – Caratteristiche

---

- Affinché le applicazioni siano in grado di comunicare, il livello di trasporto utilizza un **indirizzamento basato su porte**
- Un processo software (tipicamente) lato Server si mette in «*ascolto*» (*listening*) su uno specifico numero di porta ed eroga i suoi servizi tramite tale porta
  - Il Client invia dati al Server su tale porta in modo che vengano processati dall'applicazione attiva sul Server
- Sono utilizzati 16 bit per l'indirizzamento delle porte
  - Esistono quindi  $2^{16} = 65536$  porte
  - Il numero di porte varia da 0 a 65535

# Suite Protocollore TCP/IP

## Le Porte – Caratteristiche

---

- Gli intervalli di utilizzo dei numeri di porta sono regolamentati da convenzioni o accordi internazionali
- Le porte sono generalmente classificate in base a tre categorie
  - *Well-known Port*
  - *User o Registered Port*
  - *Private/Dynamic/Ephemeral Port*

# Suite Protocollore TCP/IP

## Le Porte – Caratteristiche

---

- **Well-known Port:** Vanno da **0** a **1023** e sono porte riservate
  - Usate da processi Server che devono essere eseguiti da amministratori o da utenti con privilegi specifici
- **User o Registered Port:** Vanno da **1024** a **49151** e sono porte per le quali un utente può chiedere la registrazione all'*Internet Assigned Number Authority (IANA)*
  - Così da riservare una di queste porte ad una specifica applicazione Client-Server
- **Private/Dynamic/Ephemeral Port:** Vanno da **49152** a **65535** ed ognuno può utilizzarle senza necessità di registrazione presso lo *IANA*

# Suite Protocollore TCP/IP

## Le Porte – Caratteristiche

---

- **Well-known Port:** Vanno da 0 a 1023 e sono porte riservate
  - Usate da processi Server che devono essere eseguiti da amministratori o da utenti con privilegi specifici
- **User o Registered Port:** Vanno da 1024 a 49151 e sono porte per le quali un utente può chiedere la registrazione all'*Internet Assigned Number Authority (IANA)*
  - Così da riservare una di queste porte ad una specifica applicazione Client-Server
- **Private/Dynamic/Ephemeral Port:** Vanno da 49152 a 65535 ed

**N.B. La classificazione delle porte in tali categorie è una «convenzione» e nulla vieta di utilizzare arbitrariamente qualsiasi numero di porta ammesso**



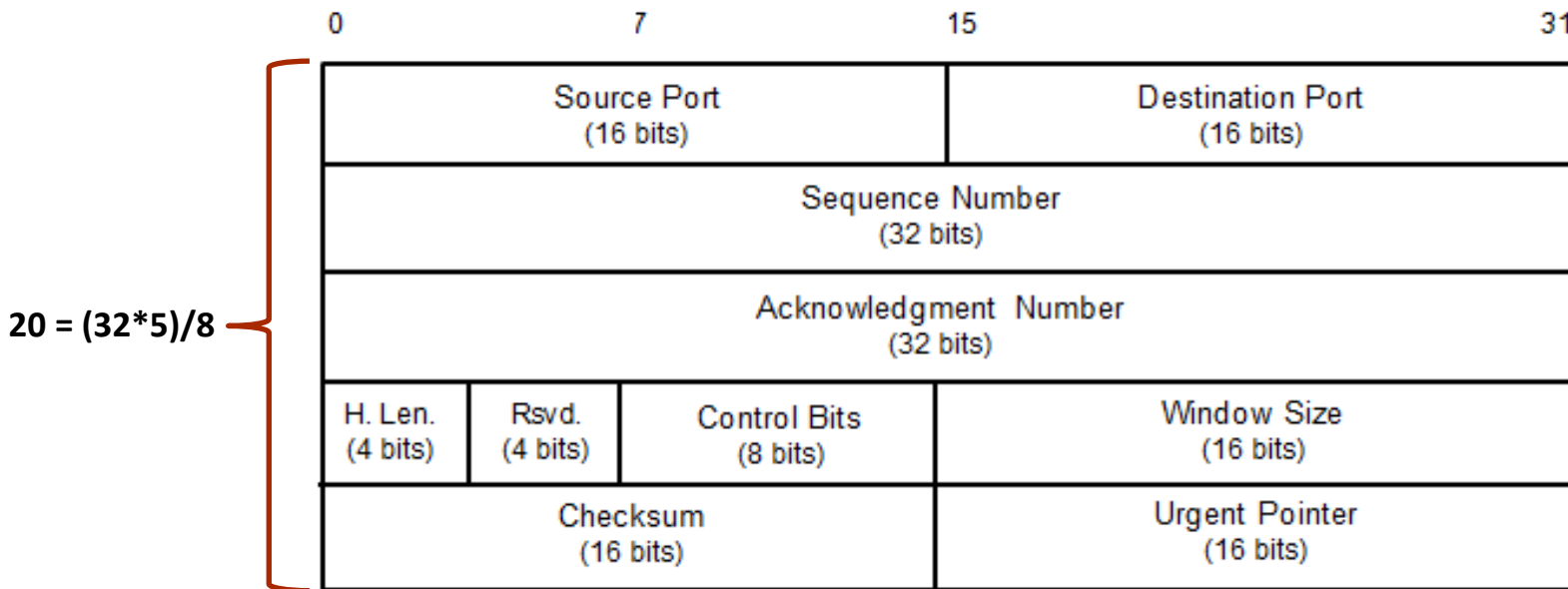
# Outline

---

- Concetti Introduttivi
- Suite Protocollo TCP/IP
- **Formato dei Messaggi TCP e UDP**
- Active Enumeration
  - Network Scanner Nmap
  - Zenmap
  - Unicornscan
- Passive Enumeration
  - Shodan
  - ZoomEye
  - Censys
  - FOFA

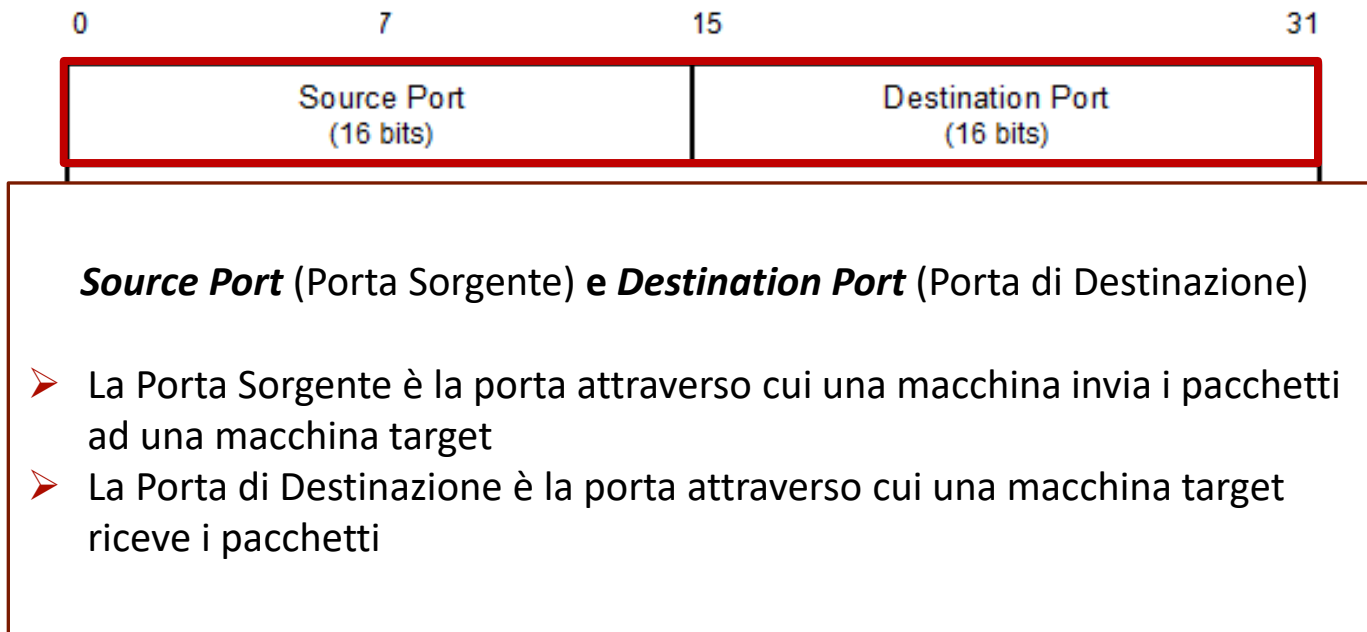
# Formato dei Messaggi TCP

- Un messaggio *TCP* è chiamato **segmento** ed è costituito da un **header** e da una **sezione dati**
- L'**header** è di 20 byte (senza opzioni *TCP*)



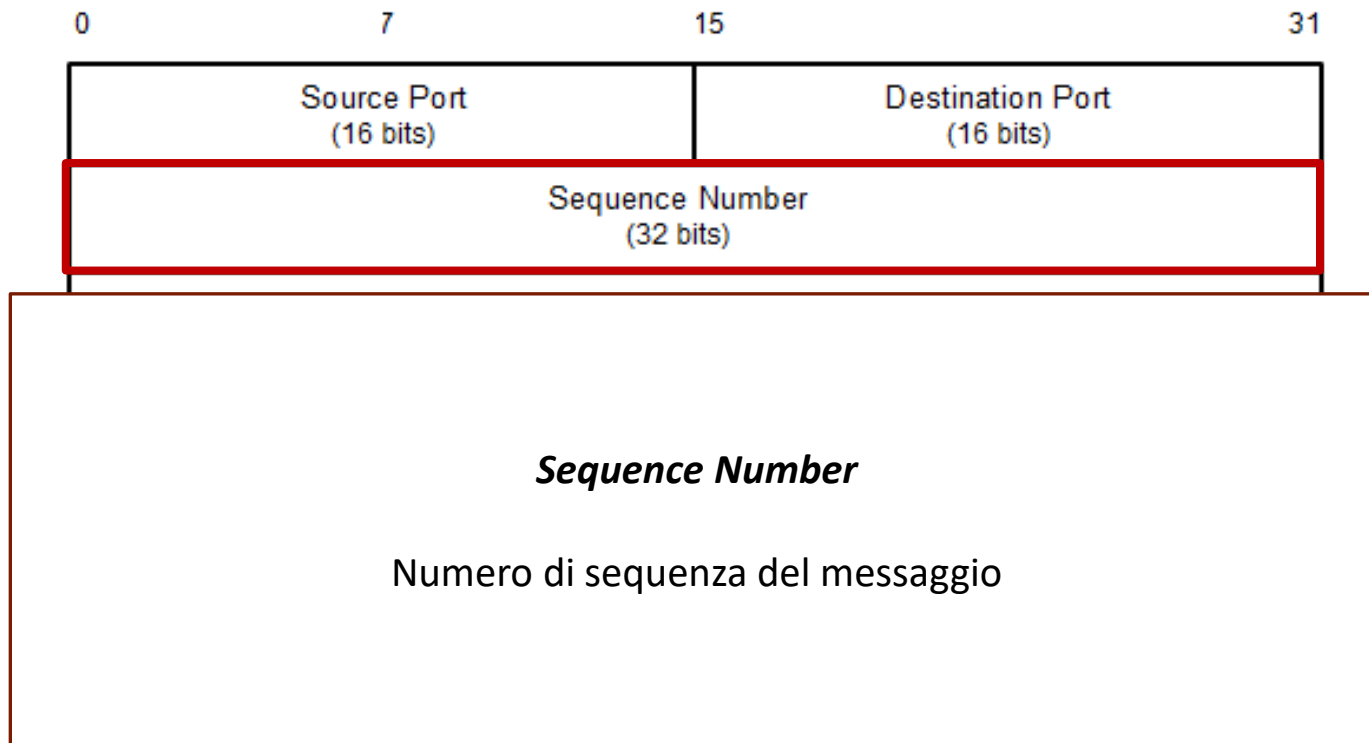
# Formato dei Messaggi TCP

---



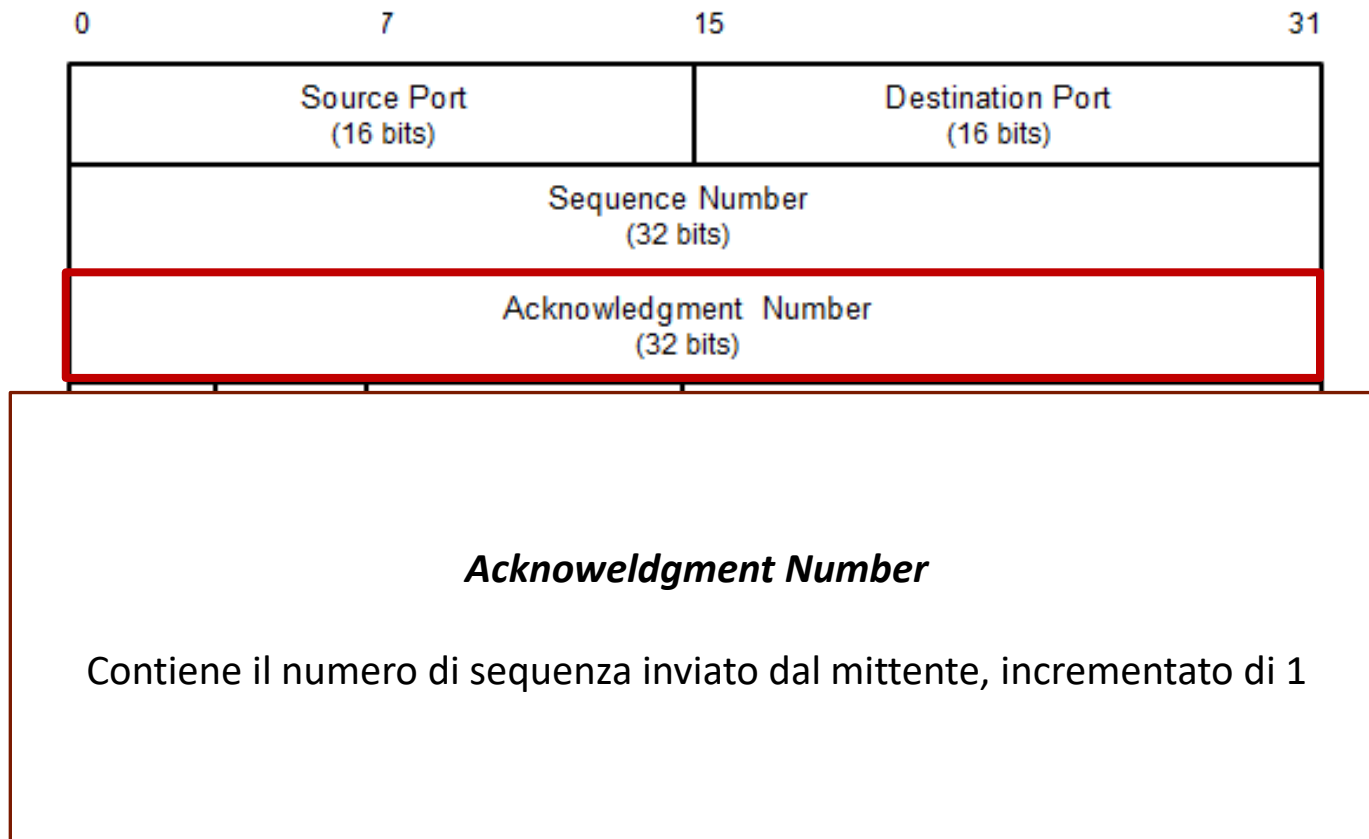
# Formato dei Messaggi TCP

---



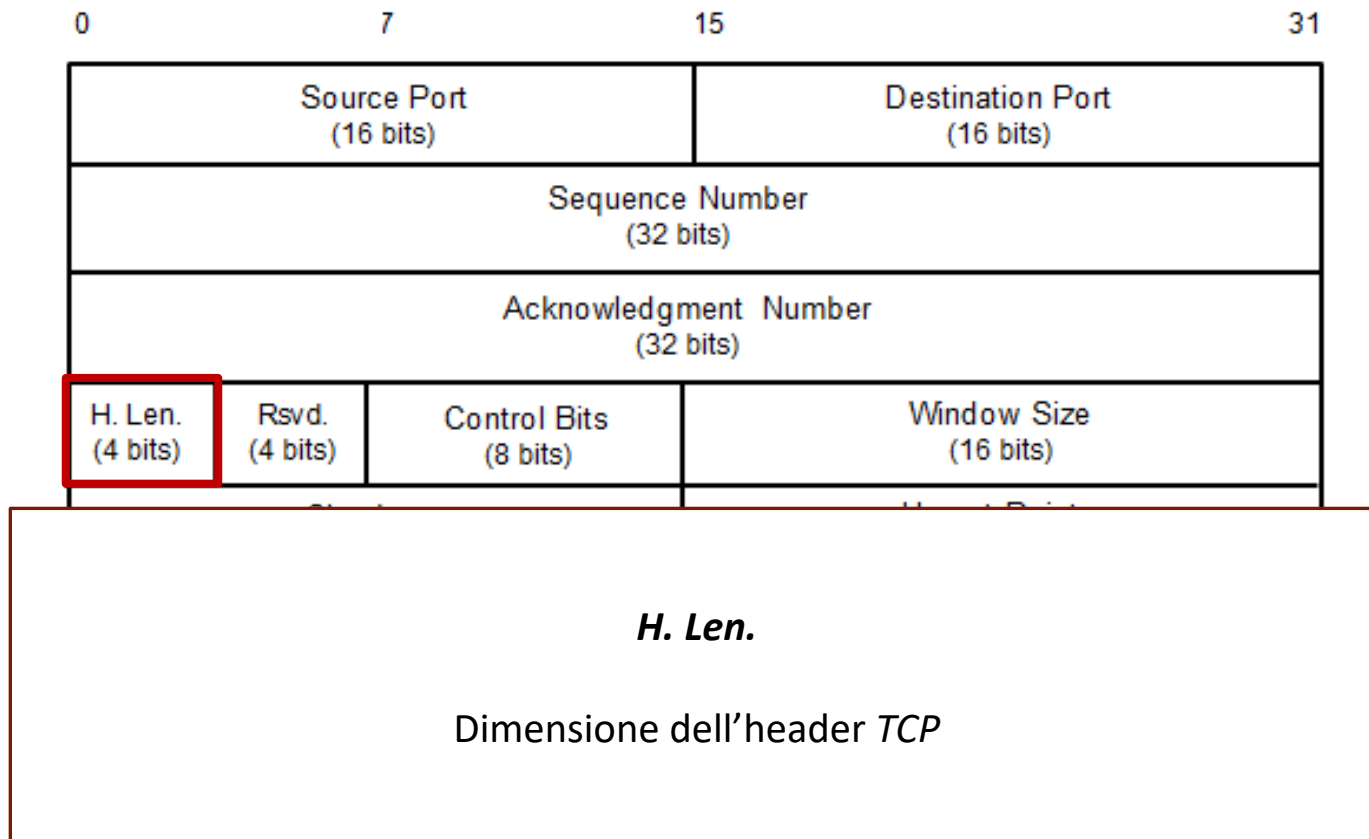
# Formato dei Messaggi TCP

---



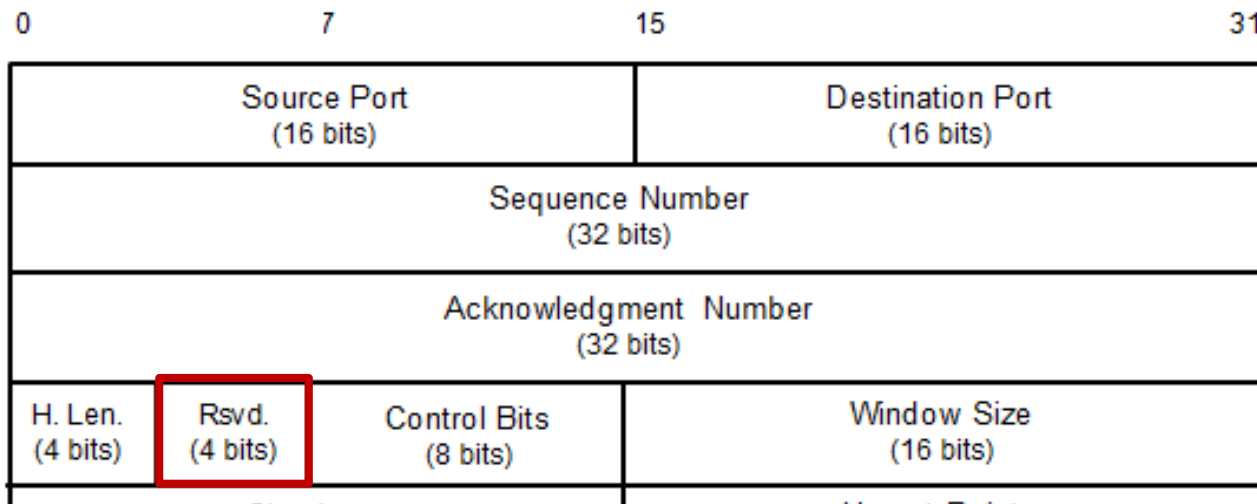
# Formato dei Messaggi TCP

---



# Formato dei Messaggi TCP

---

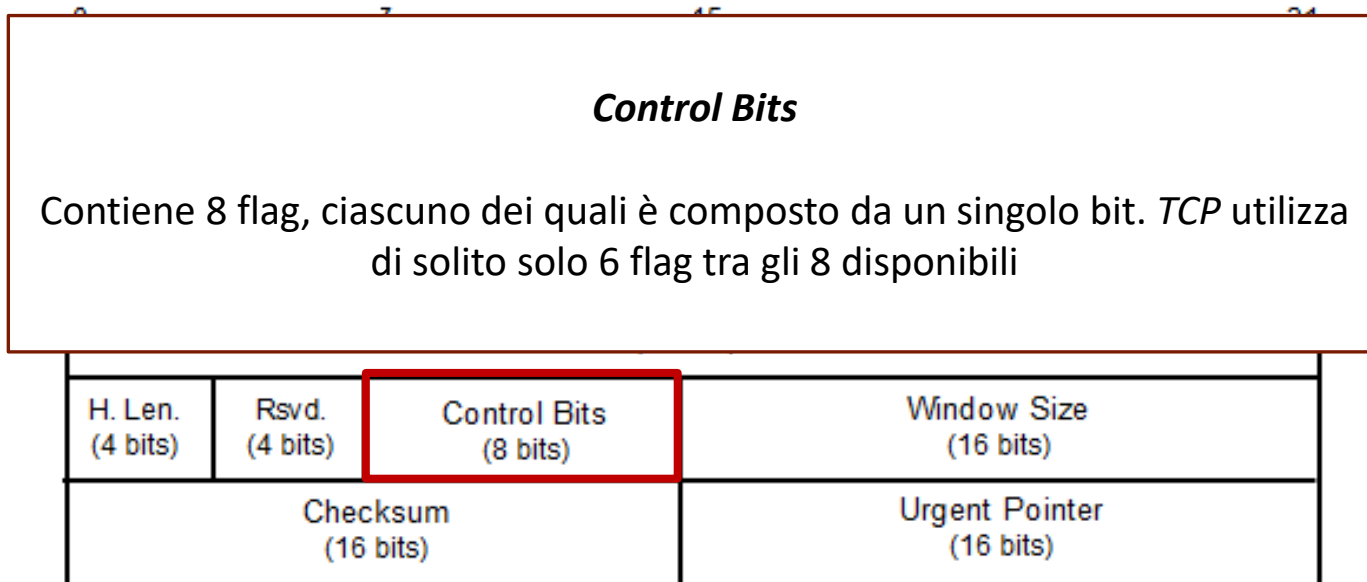


***Rsvd.***

Riservato per usi futuri, composto da 4 bit e deve avere valore 0

# Formato dei Messaggi TCP

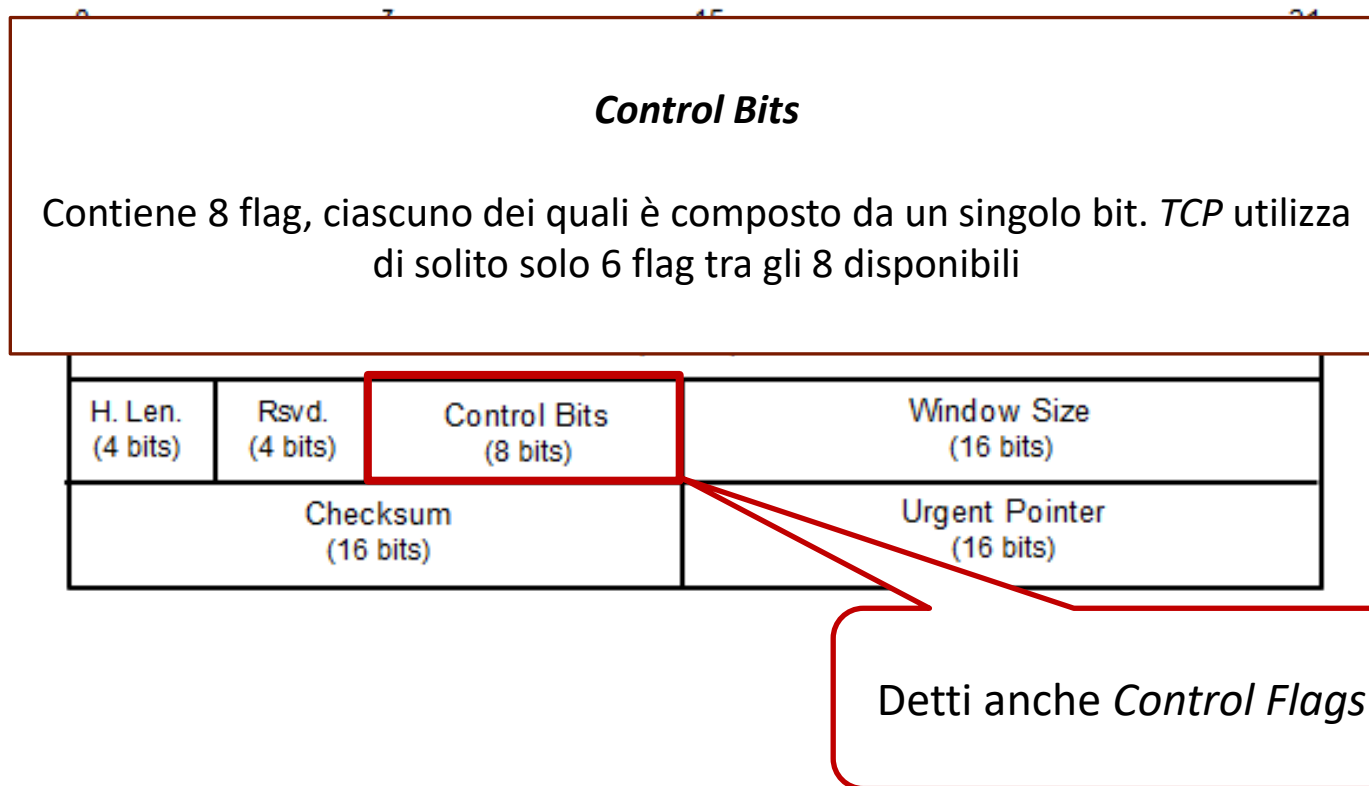
---



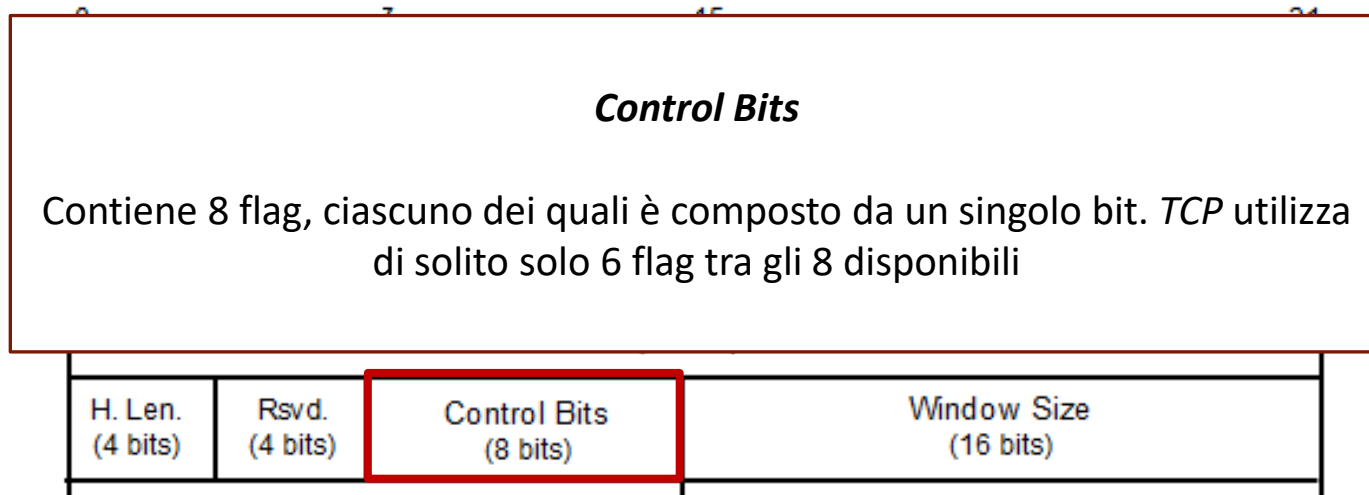


# Formato dei Messaggi TCP

---



# Formato dei Messaggi TCP



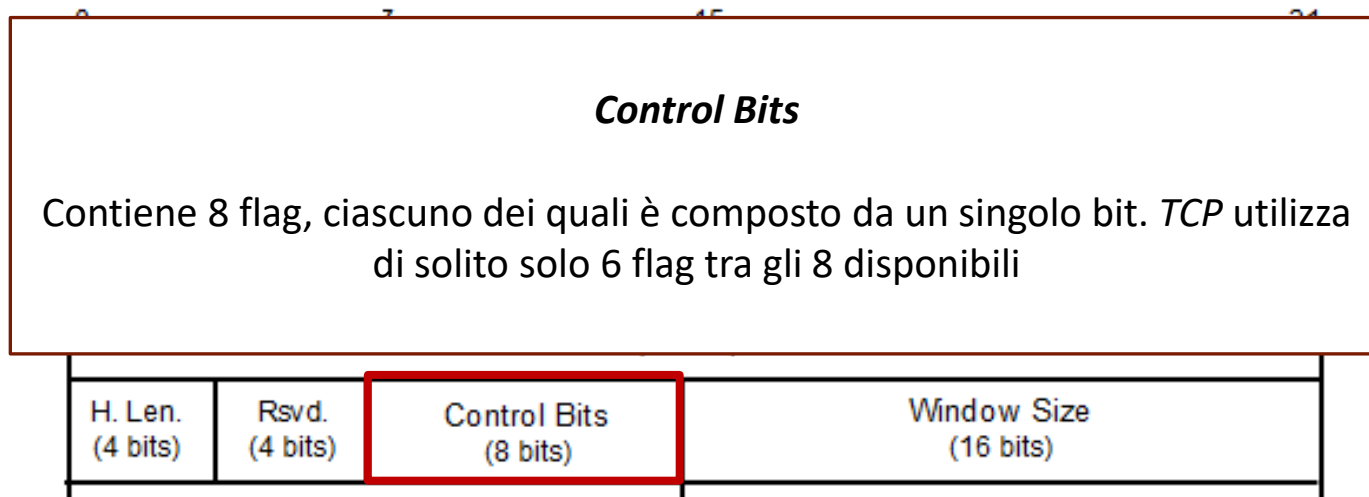
## RFC 793

### I sei bit flag di solito utilizzati da TCP

- **SYN:** Sincronizza i numeri di sequenza (utilizzato di solito per stabilire la connessione)
- **ACK:** Indica che il campo Acknowledgement è significativo; se un pacchetto ha questo flag attivo, esso è un ACK in risposta ad un pacchetto precedentemente ricevuto
- **RST:** Resetta la connessione
- **FIN:** Indica che non ci sono altri dati da inviare (utilizzato di solito per chiudere una connessione)
- **PSH:** Indica che i dati devono essere trasmessi immediatamente, invece di aspettare altri dati
- **URG:** Indica che il campo *Urgent Pointer* del messaggio è significativo

# Formato dei Messaggi TCP

---



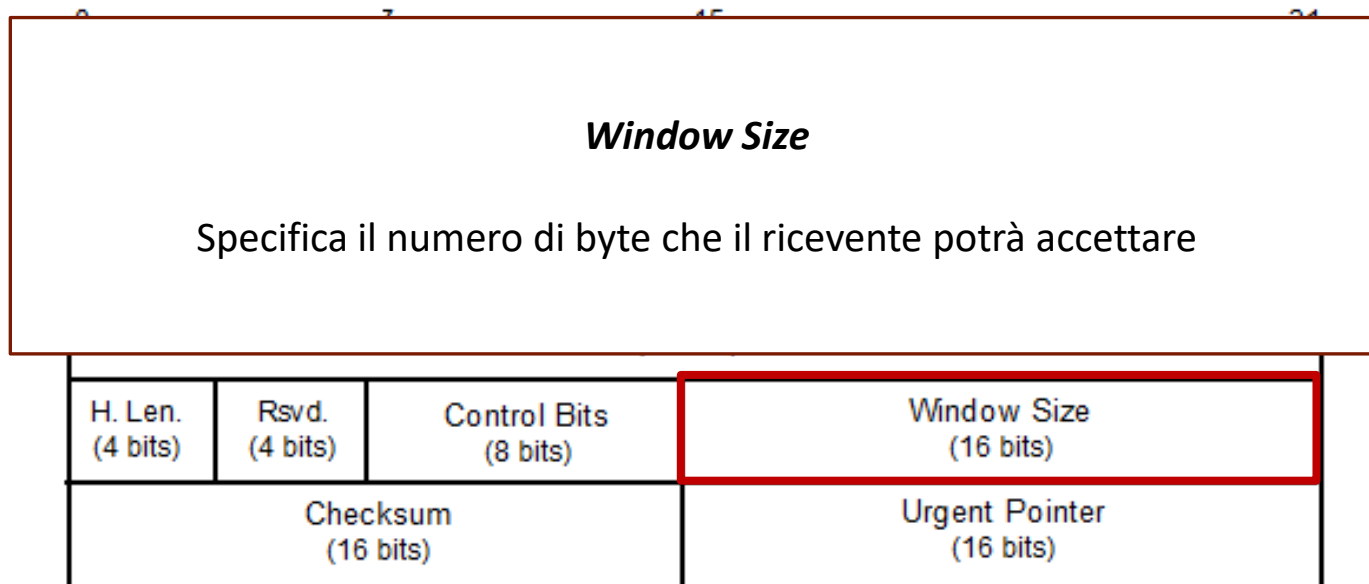
## RFC 3168

### Ulteriori bit flag utilizzati da TCP

- **CWR** (*Congestion Window Reduced*): Indica che il buffer di trasmissione del mittente si sta riempiendo a causa di una congestione. Sarà quindi necessario abbassare la velocità di trasmissione
- **ECN-Echo** (*Explicit Connection Notification-Echo*): indica che la connessione di rete sta riscontrando una congestione

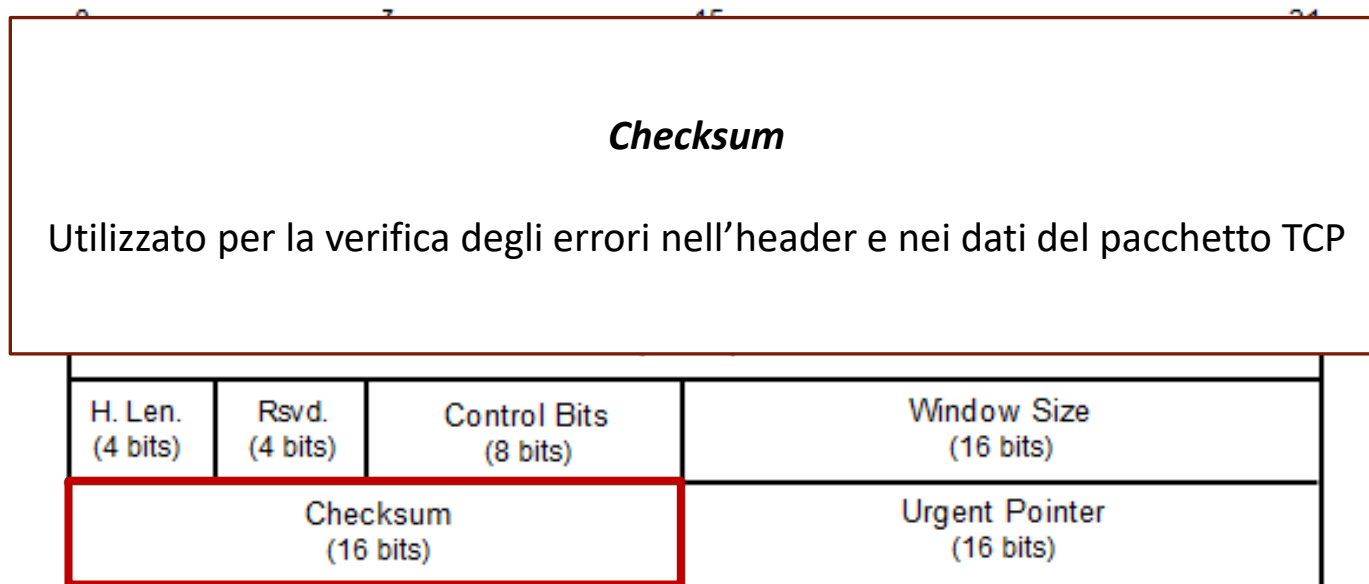
# Formato dei Messaggi TCP

---



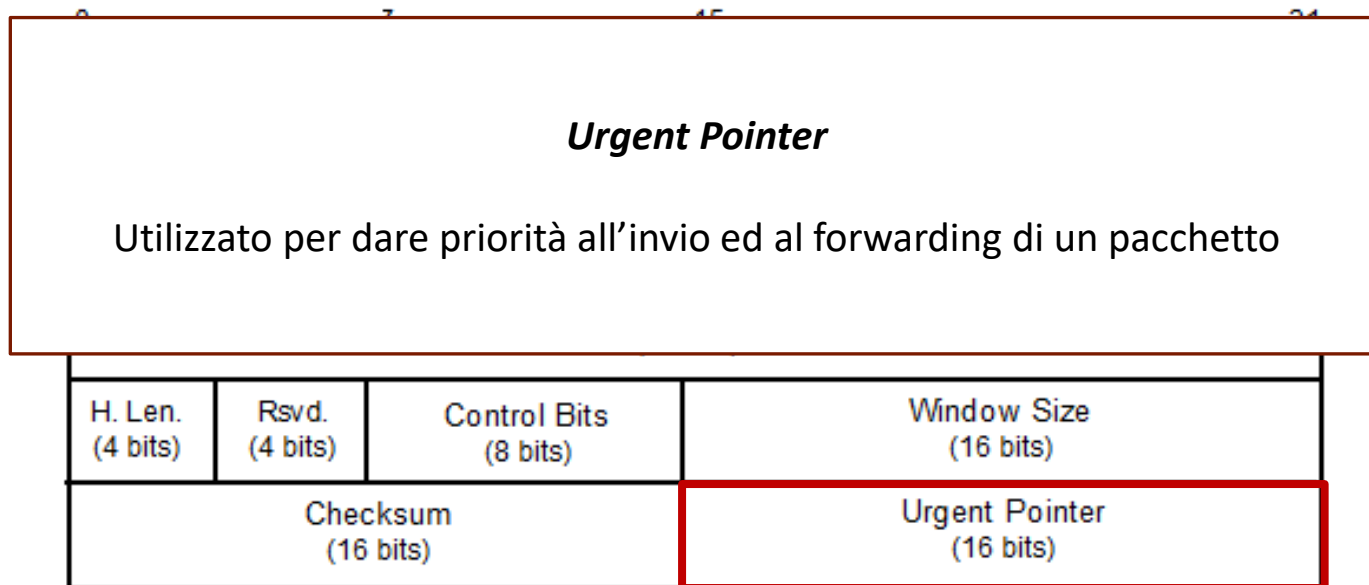
# Formato dei Messaggi TCP

---



# Formato dei Messaggi TCP

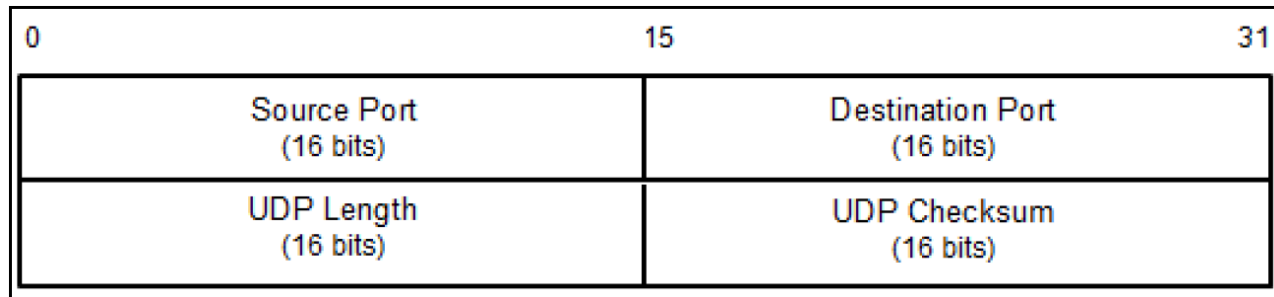
---



# Formato dei Messaggi UDP

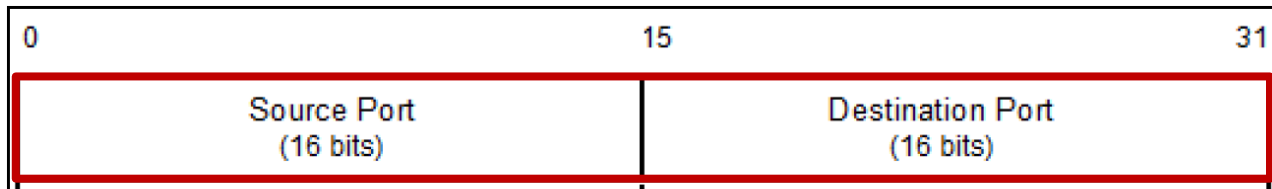
---

- Un messaggio UDP è costituito da un **header** e da una **sezione dati**
  - L'**header** è di 8 byte (senza opzioni UDP)



# Formato dei Messaggi UDP

---



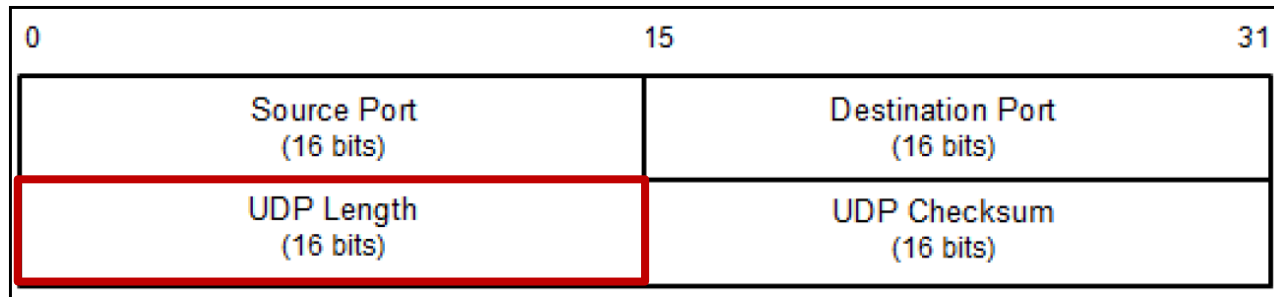
## ***Source Port (Porta Sorgente) e Destination Port (Porta di Destinazione)***

- La Porta Sorgente è la porta attraverso cui una macchina invia i pacchetti ad una macchina target
- La Porta di Destinazione è la porta attraverso cui una macchina target riceve i pacchetti



# Formato dei Messaggi UDP

---

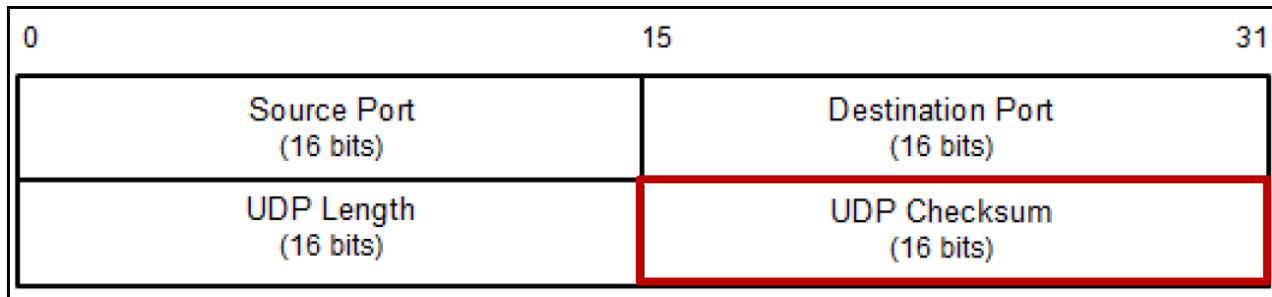


***UDP Length***

Dimensione dell'header UDP

# Formato dei Messaggi UDP

---



## ***UDP Checksum***

Utilizzato per la verifica degli errori nell'header e nei dati

# Outline

---

- Concetti Introduttivi
- Suite Protocollore TCP/IP
- Formato dei Messaggi TCP e UDP
- **Active Enumeration**
  - Network Scanner Nmap
  - Zenmap
  - Unicornscan
- Passive Enumeration
  - Shodan
  - ZoomEye
  - Censys
  - FOFA

# Active Enumeration

## Network Scanning – Service Enumeration

---

- **Service Enumeration:** consente di scoprire la versione del servizio erogato da una porta aperta sulla macchina target
- **N.B.** Le informazioni sulla versione di un determinato servizio sono di fondamentale importanza
  - Il pentester potrebbe cercare le vulnerabilità di sicurezza esistenti per tale versione del servizio

# Active Enumeration

## Network Scanning – Service Enumeration

---

- **Osservazione 1:** Talvolta gli amministratori di rete/sistema cambiano le porte predefinite per alcuni servizi
  - Ad esempio, un servizio *SSH* potrebbe non essere associato alla porta **22** (come da convenzione)
  - Un amministratore potrebbe associarlo alla porta **22222**
  - Se un pentester eseguisse solo una scansione sulla porta convenzionale del servizio *SSH* potrebbe non trovare tale servizio attivo
- **Osservazione 2:** Il pentester potrebbe avere difficoltà quando si tratta di analizzare *servizi proprietari*
  - In esecuzione su porte non standard



# Active Enumeration

## Network Scanning – Service Enumeration

---

- Utilizzando strumenti per l'enumerazione automatica dei servizi, i problemi caratterizzati in precedenza dalle Osservazioni 1 e 2 possono essere mitigati
  - Un servizio potrebbe essere individuato indipendentemente dalla porta che utilizza



# Outline

---

- Concetti Introduttivi
- Suite Protocollore TCP/IP
- Formato dei Messaggi TCP e UDP
- Active Enumeration
  - **Network Scanner Nmap**
  - Zenmap
  - Unicornscan
- Passive Enumeration
  - Shodan
  - ZoomEye
  - Censys
  - FOFA

# Nmap

## Caratteristiche

---

- Nmap («**N**etwork **M**apper»)
  - Strumento open source per esplorazioni della rete e controlli di sicurezza
  - Consente di scansionare rapidamente sia reti di grandi dimensioni che singoli host
  - Comunemente utilizzato da amministratori di sistema e di rete per
    - Controlli di Sicurezza
    - Attività di routine riguardanti la rete
      - Inventario di rete
      - Gestione dei programmi di aggiornamento dei servizi
      - Monitoraggio del tempo di attività degli host e dei servizi
      - Etc





# Nmap

## Principali Funzionalità

---

- *Port Scanner* estremamente potente e flessibile
- Oltre ad essere un port scanner Nmap fornisce ulteriori funzionalità
  - **Host Discovery**: Rileva gli host attivi all'interno dell'asset analizzato
    - Di default, per effettuare l'*Host Discovery*, Nmap invia
      - una *ICMP Echo Request*
      - un pacchetto *TCP SYN* alla porta 443
      - un pacchetto *TCP ACK* alla porta 80
      - una *ICMP Timestamp Request*



# Nmap

## Principali Funzionalità

---

- *Port Scanner* estremamente potente e flessibile
- Oltre ad essere un port scanner Nmap fornisce ulteriori funzionalità
  - **Service/Version Detection:** Oltre ad individuare le porte «aperte» sulla macchina target, Nmap permette di ricavare ulteriori informazioni su tali porte
    - Protocolli e servizi utilizzati
    - Nomi delle applicazioni
    - Versioni delle applicazioni utilizzate
    - Etc



# Nmap

## Principali Funzionalità

---

- *Port Scanner* estremamente potente e flessibile
- Oltre ad essere un port scanner Nmap fornisce ulteriori funzionalità
  - **Operating System (OS) Detection**
    - Nmap invia una serie di pacchetti alla macchina target ed esamina le risposte
    - Confronta queste risposte con un proprio database e mostra i dettagli se viene trovata una corrispondenza
  - **Network Traceroute**: Un traceroute Nmap inizia con un certo valore del *Time to Live (TTL)*
    - Il valore del TTL viene decrementato fino a quando non si raggiunge il valore 0
  - **Nmap Scripting Engine (NSE)**: Permette di aggiungere nuove funzionalità ad Nmap
    - Maggiori dettagli successivamente...



# Nmap

## Aggiornamento ed Utilizzo

---

- Prima di utilizzare Nmap è importante aggiornarlo
  - Ci potrebbero essere informazioni (*fingerprint*) relative a nuovi Sistemi Operativi, a nuovi servizi, etc
    - `apt-get update`
    - `apt-get upgrade`
- Nmap può essere avviato in due modi
  - Dal menu «01 – Information Gathering» di Kali
  - Digitando il comando **nmap** da terminale



# Nmap

## Tecniche di Scansione

---

### ➤ Nmap

- Fornisce varie tipologie di scansione, ciascuna con le proprie caratteristiche
- Permette anche di creare scansioni ad hoc
- Maggiori dettagli successivamente

Flag	Role	Command
-sS	TCP syn scan	nmap -sS <target>
-sT	TCP connect() scan	nmap -sT <target>
-sU	UDP scan	nmap -sU <target>
-sA	TCP ack scan	nmap -sA <target>
-sY	SCTP INIT scan	nmap -sY <target>
-sF	FIN Scan	nmap -sF <target>
-sP	Ping Scan	nmap -sP <target>
-sV	Version Detection	nmap -sV <target>
-sI	Idle Scan	nmap -sI <target>
-sW	TCP Window scan	nmap -sW <target>
-sM	TCP maimon scan	nmap -sM <target>

.....

.....

.....



# Nmap

## Primo Esempio di Utilizzo

- Utilizziamo Nmap per scansionare Metasploitable 2 (Indirizzo IP: 10.0.2.6)

➤ `nmap 10.0.2.6`

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-24 16:05 CET
Nmap scan report for 10.0.2.6
Host is up (0.00051s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:AE:29:E1 (Oracle VirtualBox virtual NIC)
```



# Nmap

## Primo Esempio di Utilizzo

- Utilizziamo Nmap per scansionare Metasploitable 2 (Indirizzo IP: 10.0.2.6)

➤ `nmap 10.0.2.6`

Numero di Porta / Protocollo

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-24 16:05 CET
Nmap scan report for 10.0.2.6
Host is up (0.00051s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
```

MAC Address: 08:00:27:AE:29:E1 (Oracle VirtualBox virtual NIC)



# Nmap

## Primo Esempio di Utilizzo

- Utilizziamo Nmap per scansionare Metasploitable 2 (Indirizzo IP: 10.0.2.6)

➤ `nmap 10.0.2.6`

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-24 16:05 CET
Nmap scan report for 10.0.2.6
Host is up (0.00051s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
```

Stato della Porta

MAC Address: 08:00:27:AE:29:E1 (Oracle VirtualBox virtual NIC)

Enumerate





# Nmap

## Primo Esempio di Utilizzo

- Utilizziamo Nmap per scansionare Metasploitable 2 (Indirizzo IP: 10.0.2.6)

➤ `nmap 10.0.2.6`

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-24 16:05 CET
Nmap scan report for 10.0.2.6
Host is up (0.00051s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
```

Nome del Servizio erogato dalla porta

MAC Address: 08:00:27:AE:29:E1 (Oracle VirtualBox virtual NIC)

Enumerate



# Nmap

## Stato delle Porte

---

- Nmap definisce **sei stati** per le **porte**
  - **Open**: esiste un'applicazione che accetta connessioni *TCP* o datagrammi *UDP*
  - **Closed**: sebbene la porta sia accessibile, non ci sono applicazioni in ascolto su tale porta
  - **Filtered**: Nmap non è in grado di determinare se la porta è «**aperta**» o meno
    - Probabilmente esiste un dispositivo di filtraggio dei pacchetti (ad esempio un *firewall*) che non permette di raggiungere la porta sulla macchina target

**N.B.** A seconda della tecnica di scansione utilizzata, verrà restituito un determinato insieme di stati per le porte



# Nmap

## Stato delle Porte

---

- Nmap definisce **sei stati** per le **porte**
  - **Unfiltered**: la porta è accessibile ma Nmap non può determinare se è «aperta» o «chiusa»
  - **Open | Filtered**: Nmap non è in grado di determinare se una porta è «aperta» o «filtrata»
  - **Closed | Filtered**: Nmap non è in grado di determinare se una porta è «chiusa» o «filtrata»

**N.B.** A seconda della tecnica di scansione utilizzata, verrà restituito un determinato insieme di stati per le porte



# Nmap

## Specificare il Target

---

➤ Nmap permette di specificare le macchine target in quattro modi

➤ **Singolo indirizzo IP (o singolo hostname)**

➤ Ad esempio, `nmap 10.0.2.6`

➤ **Un'intera rete di indirizzi IP adiacenti**, utilizzando la **notazione CIDR**

➤ Ad esempio, `nmap 10.0.2.0/24`

➤ 256 indirizzi IP: da `10.0.2.0` a `10.0.2.255`

➤ **Intervallo degli ottetti relativi agli indirizzi IP**

➤ Ad esempio, `nmap 10.0.2-4,6.1` (Indirizzi IP: `10.0.2.1`, `10.0.3.1`, `10.0.4.1`, `10.0.6.1`)

➤ **Indirizzi IP multipli**

➤ Ad esempio, `nmap 10.0.2.5 172.16.16-18,21.5` (Indirizzi IP: `10.0.2.5`, `172.16.16.5`, `172.16.17.5`, `172.16.18.5`, `172.16.21.5`)



# Nmap

## Specificare il Target

---

- Oltre a specificare il/i target da terminale, Nmap permette anche di farlo mediante un file testuale
  - Utilizzando l'opzione **-iL <inputfilename>**
- Questa opzione è utile se già si dispone degli indirizzi IP da analizzare
  - Ad esempio, ottenuti tramite la fase di *Information Gathering*
- Ciascuna entry del file deve essere separata da *spazi, tabulazioni o newline*

```
10.0.1.1-254  
10.0.2.1-254
```

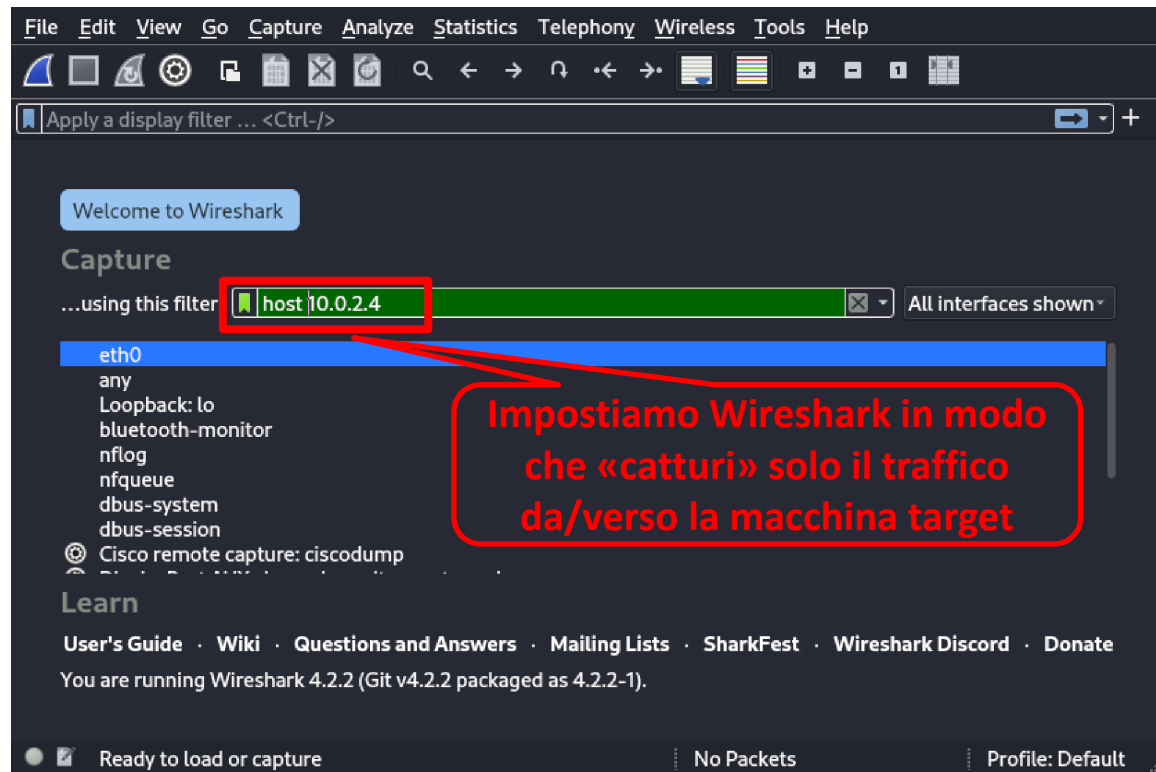
File di esempio: **lista.txt**



# Nmap

## Porte Scansionate di Default

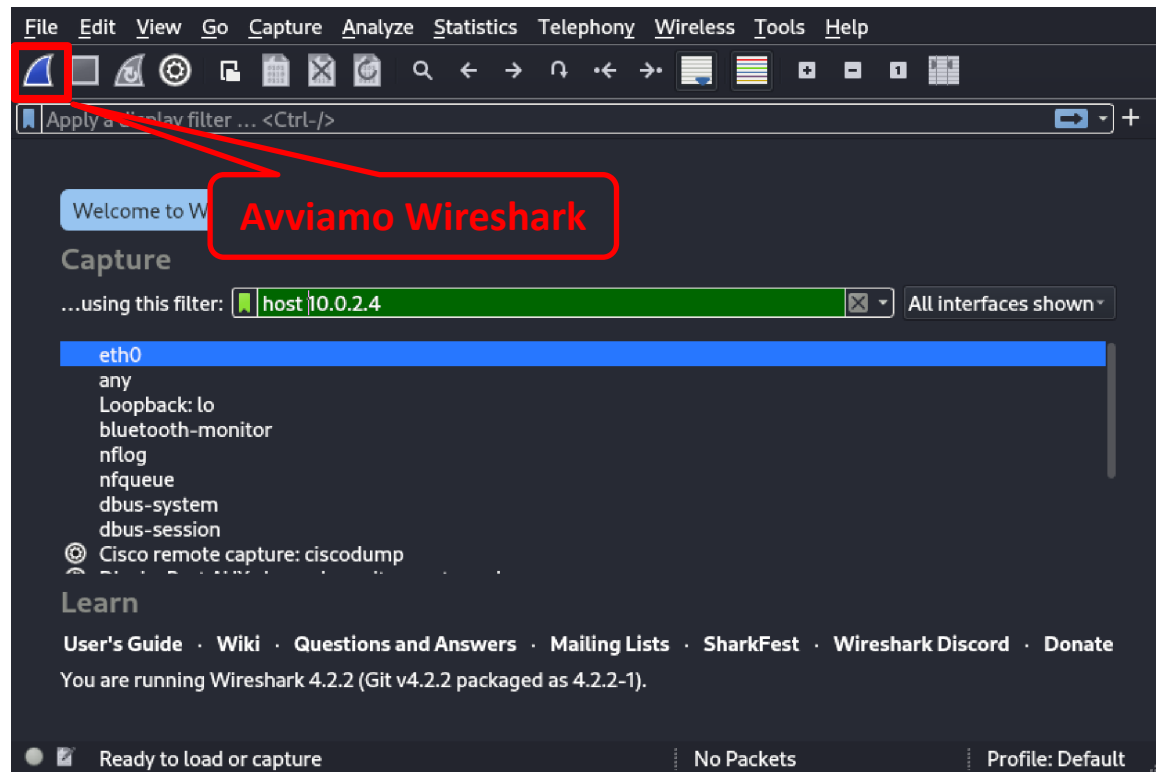
- Di default Nmap analizza (*scansiona*) **1002 porte TCP**
- Attraverso Wireshark vediamo quali sono tali porte



# Nmap

## Porte Scansionate di Default

- Di default Nmap analizza (*scansiona*) **1002 porte TCP**
- Attraverso Wireshark vediamo quali sono tali porte



# Nmap

## Porte Scansionate di Default

- Di **default Nmap analizza** (*scansiona*) **1002 porte TCP**
- Attraverso Wireshark vediamo quali sono tali porte

Output Parziale

```
$ nmap 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-17 09:17 EDT
Nmap scan report for 10.0.2.4 (10.0.2.4)
Host is up (0.014s latency)
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
```

Avviamo la scansione tramite nmap

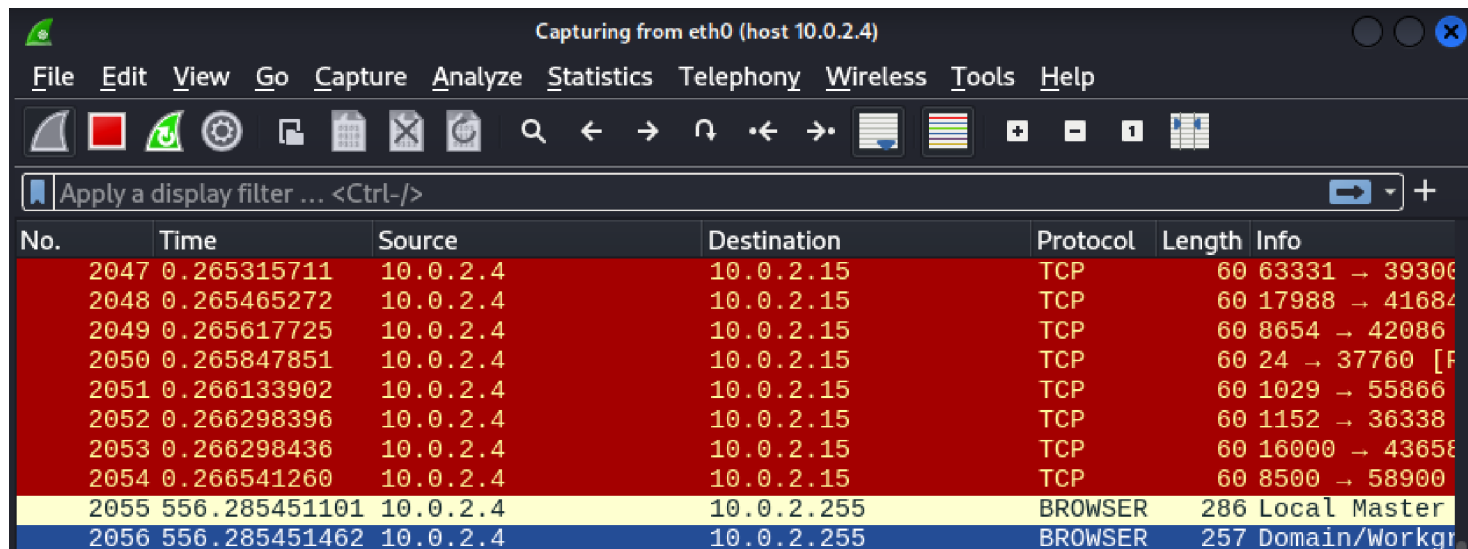




# Nmap

## Porte Scansionate di Default

- Di default Nmap analizza (*scansiona*) **1002 porte TCP**
- Attraverso Wireshark vediamo quali sono tali porte



Capturing from eth0 (host 10.0.2.4)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

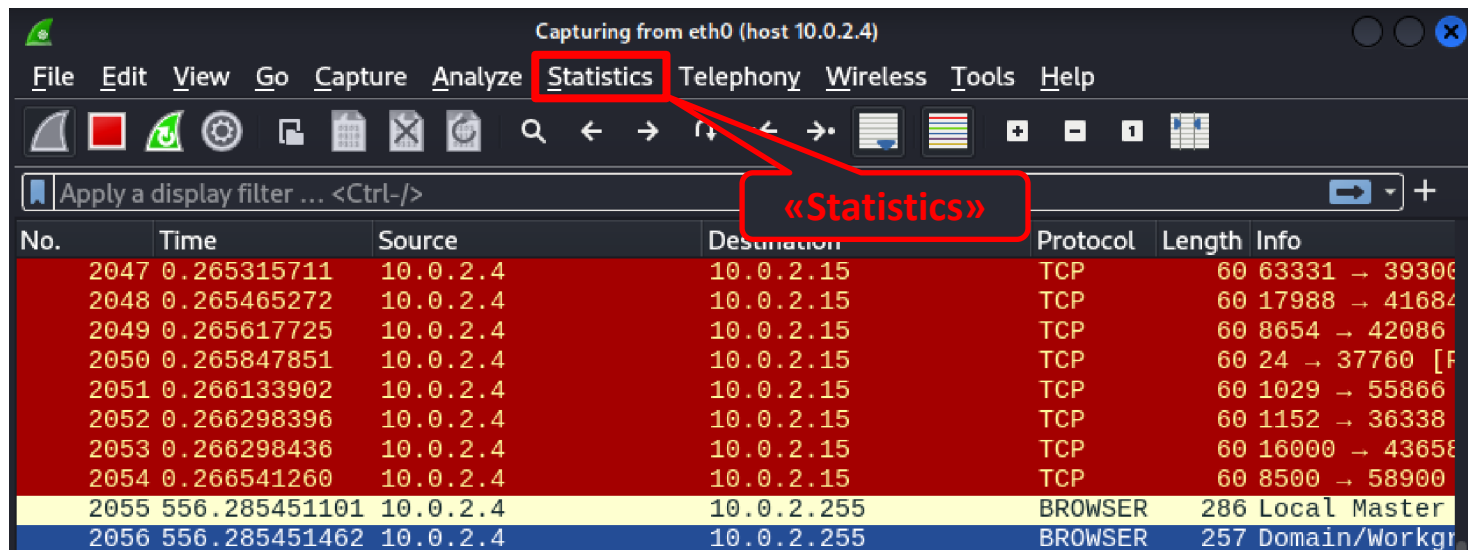
No.	Time	Source	Destination	Protocol	Length	Info
2047	0.265315711	10.0.2.4	10.0.2.15	TCP	60	63331 → 39306
2048	0.265465272	10.0.2.4	10.0.2.15	TCP	60	17988 → 41684
2049	0.265617725	10.0.2.4	10.0.2.15	TCP	60	8654 → 42086
2050	0.265847851	10.0.2.4	10.0.2.15	TCP	60	24 → 37760 [F
2051	0.266133902	10.0.2.4	10.0.2.15	TCP	60	1029 → 55866
2052	0.266298396	10.0.2.4	10.0.2.15	TCP	60	1152 → 36338
2053	0.266298436	10.0.2.4	10.0.2.15	TCP	60	16000 → 43658
2054	0.266541260	10.0.2.4	10.0.2.15	TCP	60	8500 → 58900
2055	556.285451101	10.0.2.4	10.0.2.255	BROWSER	286	Local Master
2056	556.285451462	10.0.2.4	10.0.2.255	BROWSER	257	Domain/Workgr



# Nmap

## Porte Scansionate di Default

- Di default Nmap analizza (*scansiona*) **1002 porte TCP**
- Attraverso Wireshark vediamo quali sono tali porte



Capturing from eth0 (host 10.0.2.4)

File Edit View Go Capture Analyze **Statistics** Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

«Statistics»

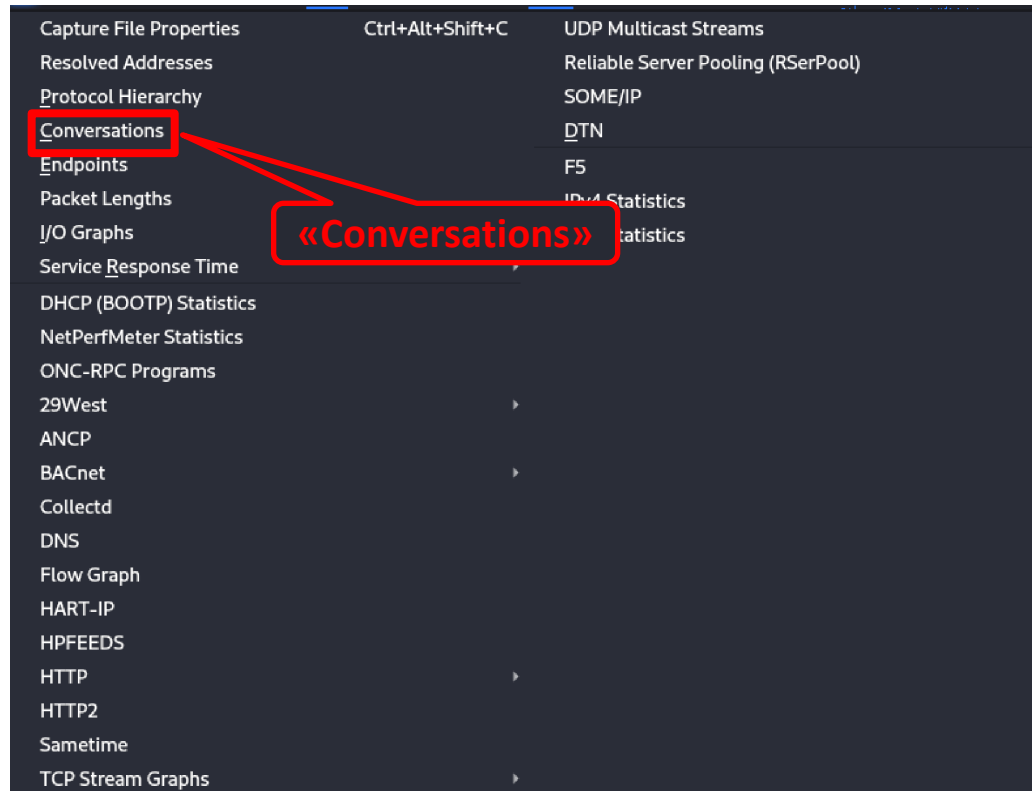
No.	Time	Source	Destination	Protocol	Length	Info
2047	0.265315711	10.0.2.4	10.0.2.15	TCP	60	63331 → 39306
2048	0.265465272	10.0.2.4	10.0.2.15	TCP	60	17988 → 41684
2049	0.265617725	10.0.2.4	10.0.2.15	TCP	60	8654 → 42086
2050	0.265847851	10.0.2.4	10.0.2.15	TCP	60	24 → 37760 [F
2051	0.266133902	10.0.2.4	10.0.2.15	TCP	60	1029 → 55866
2052	0.266298396	10.0.2.4	10.0.2.15	TCP	60	1152 → 36338
2053	0.266298436	10.0.2.4	10.0.2.15	TCP	60	16000 → 43658
2054	0.266541260	10.0.2.4	10.0.2.15	TCP	60	8500 → 58900
2055	556.285451101	10.0.2.4	10.0.2.255	BROWSER	286	Local Master
2056	556.285451462	10.0.2.4	10.0.2.255	BROWSER	257	Domain/Workgr



# Nmap

## Porte Scansionate di Default

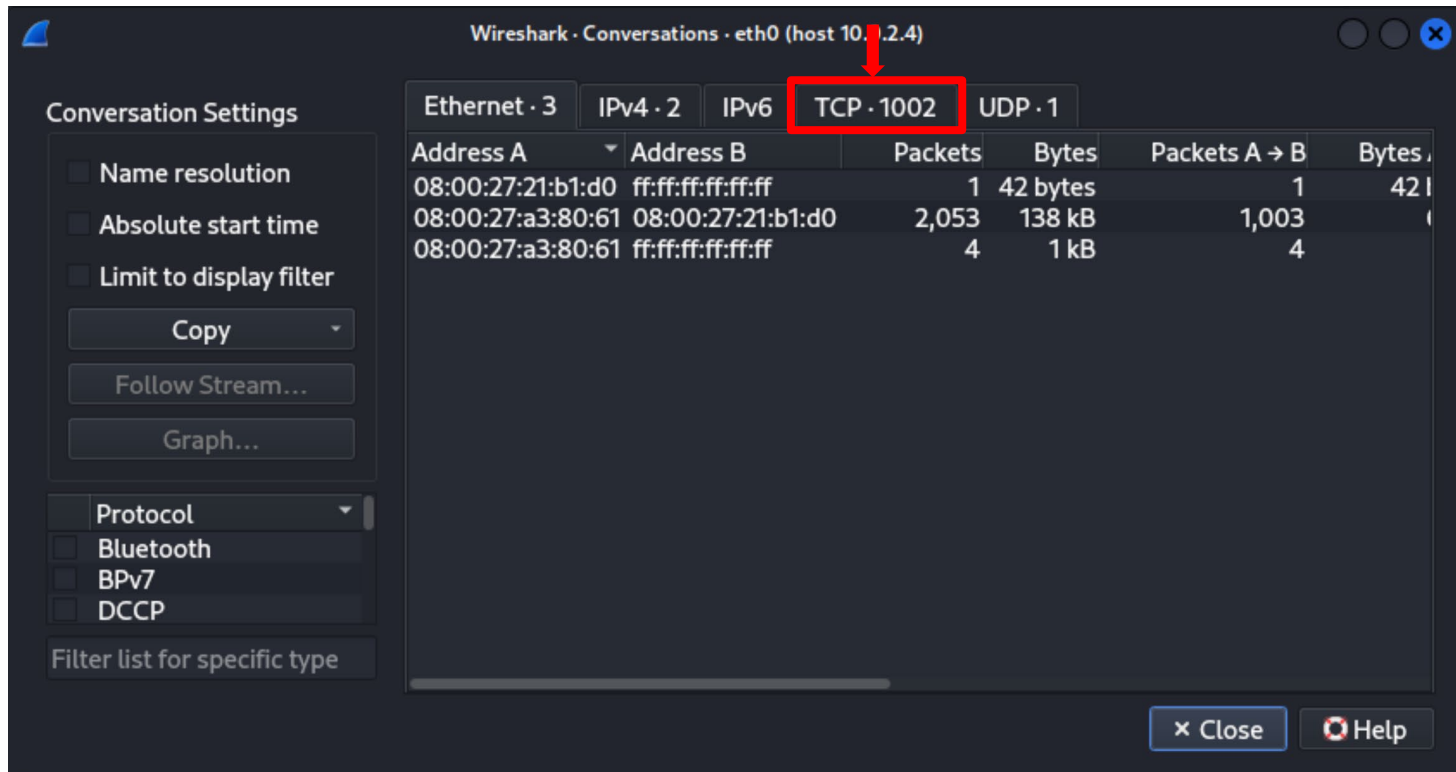
- Di default Nmap analizza (*scansiona*) **1002 porte TCP**
- Attraverso Wireshark vediamo quali sono tali porte



# Nmap

## Porte Scansionate di Default

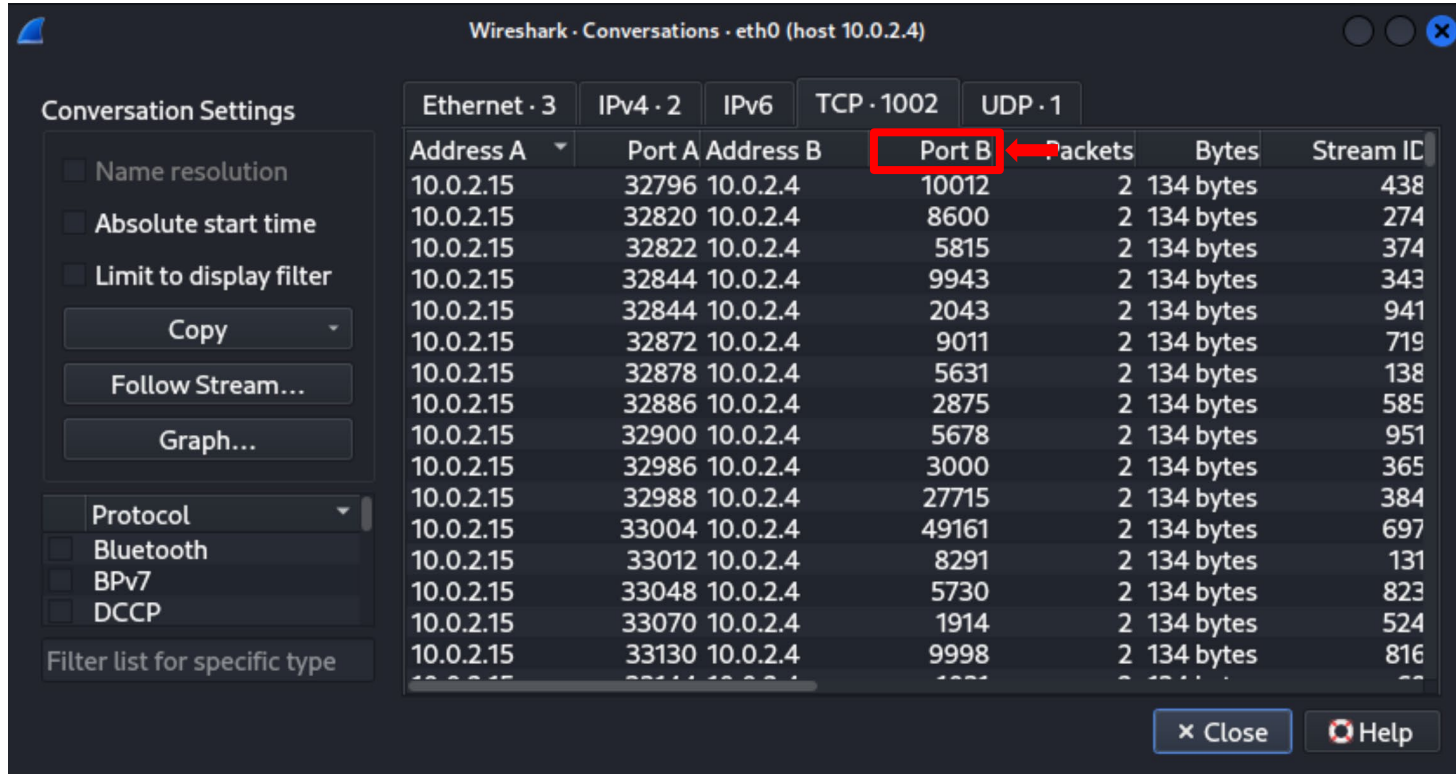
- Di **default Nmap analizza** (*scansiona*) **1002 porte TCP**
- Attraverso Wireshark vediamo quali sono tali porte



# Nmap

## Porte Scansionate di Default

- Di default Nmap analizza (*scansiona*) 1002 porte TCP
- Attraverso Wireshark vediamo quali sono tali porte



The screenshot shows the Wireshark interface with the 'Conversations' pane selected. The 'TCP · 1002' tab is active, displaying a list of scanned ports. A red box highlights the 'Port B' column header, and a red arrow points to it. The table lists 15 scanned ports, all of which are open (2 packets, 134 bytes each).

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID
10.0.2.15	32796	10.0.2.4	10012	2	134 bytes	438
10.0.2.15	32820	10.0.2.4	8600	2	134 bytes	274
10.0.2.15	32822	10.0.2.4	5815	2	134 bytes	374
10.0.2.15	32844	10.0.2.4	9943	2	134 bytes	343
10.0.2.15	32844	10.0.2.4	2043	2	134 bytes	941
10.0.2.15	32872	10.0.2.4	9011	2	134 bytes	719
10.0.2.15	32878	10.0.2.4	5631	2	134 bytes	138
10.0.2.15	32886	10.0.2.4	2875	2	134 bytes	585
10.0.2.15	32900	10.0.2.4	5678	2	134 bytes	951
10.0.2.15	32986	10.0.2.4	3000	2	134 bytes	365
10.0.2.15	32988	10.0.2.4	27715	2	134 bytes	384
10.0.2.15	33004	10.0.2.4	49161	2	134 bytes	697
10.0.2.15	33012	10.0.2.4	8291	2	134 bytes	131
10.0.2.15	33048	10.0.2.4	5730	2	134 bytes	823
10.0.2.15	33070	10.0.2.4	1914	2	134 bytes	524
10.0.2.15	33130	10.0.2.4	9998	2	134 bytes	816



# Nmap

## Porte Scansionate di Default

- Di default Nmap analizza (*scansiona*) **1002 porte TCP**
- Attraverso Wireshark vediamo quali sono tali porte

Ordiniamo le porte cliccando sull'intestazione di tale colonna

Port A	Address B	Port B	Packets	Bytes	Stream ID
10.0.2.15	10.0.2.4	57954	2	134 bytes	256
10.0.2.15	10.0.2.4	52820	2	134 bytes	189
10.0.2.15	10.0.2.4	49958	2	134 bytes	238
10.0.2.15	10.0.2.4	58938	2	134 bytes	915
10.0.2.15	10.0.2.4	47422	2	134 bytes	316
10.0.2.15	10.0.2.4	45474	2	134 bytes	947
10.0.2.15	10.0.2.4	39904	2	134 bytes	594
10.0.2.15	10.0.2.4	56622	2	134 bytes	228
10.0.2.15	10.0.2.4	42904	2	134 bytes	211
10.0.2.15	10.0.2.4	33332	2	134 bytes	266
10.0.2.15	10.0.2.4	36674	4	280 bytes	16
10.0.2.15	10.0.2.4	55464	4	280 bytes	17
10.0.2.15	10.0.2.4	56976	4	280 bytes	8
10.0.2.15	10.0.2.4	37760	2	134 bytes	997
10.0.2.15	10.0.2.4	44200	4	280 bytes	4
10.0.2.15	10.0.2.4	58818	2	134 bytes	503



# Nmap

## Specifica delle Porte

---

- Di default Nmap scansiona, secondo un ordine casuale, le **1002** porte «più comuni»
  - Tali porte sono selezionate in base al contenuto del file **nmap-services**
- Ciascuna entry del file **nmap-services** contiene
  - Nome del servizio e numero della porta, insieme al corrispondente protocollo
  - Valore che rappresenta la probabilità di trovare aperta tale porta
    - Probabilità ottenuta tramite euristiche ricavate da scansioni precedenti

<https://nmap.org/book/nmap-services.html>



# Nmap

## Specifica delle Porte (nmap-services)

---

- Ciascuna entry del file **nmap-services** contiene
  - Nome del servizio e numero della porta, insieme al corrispondente protocollo
  - Valore che rappresenta la probabilità di trovare aperta tale porta
    - Probabilità ottenuta tramite euristiche ricavate da scansioni precedenti

```
ssh      22/tcp  0.182286      # Secure Shell Login
ssh      22/udp  0.003905      # Secure Shell Login
telnet   23/tcp  0.221265      # Telnet
telnet   23/udp  0.006211      # Telnet
priv-mail 24/tcp  0.001154      # any private mail system
priv-mail 24/udp  0.000329      # any private mail system
smtp     25/tcp  0.131314      # Simple Mail Transfer
smtp     25/udp  0.001285      # Simple Mail Transfer
rsftp    26/tcp  0.007991      # RSFTP
nsw-fe   27/tcp  0.000138      # NSW User System FE
nsw-fe   27/udp  0.000395      # NSW User System FE
unknown  28/tcp  0.000050      #
msg-icp  29/tcp  0.000025      # MSG ICP
msg-icp  29/udp  0.000560      # MSG ICP
unknown  30/tcp  0.000527      #
msg-auth 31/tcp  0.000025      # MSG Authentication
```





# Nmap

## Specifica delle Porte (nmap-services)

- Ciascuna entry del file **nmap-services** contiene
  - **Nome del servizio e numero della porta, insieme al corrispondente protocollo**
  - Valore che rappresenta la probabilità di trovare aperta tale porta
    - Probabilità ottenuta tramite euristiche ricavate da scansioni precedenti

```
ssh      22/tcp    0.182286      # Secure Shell Login
ssh      22/udp    0.003905      # Secure Shell Login
telnet   23/tcp    0.221265
telnet   23/udp    0.006211
priv-mail 24/tcp    0.001154      # any private mail system
priv-mail 24/udp    0.000329      # any private mail system
smtp     25/tcp    0.131314      # Simple Mail Transfer
smtp     25/udp    0.001285      # Simple Mail Transfer
rsftp    26/tcp    0.007991      # RSFTP
nsw-fe   27/tcp    0.000138      # NSW User System FE
nsw-fe   27/udp    0.000395      # NSW User System FE
unknown  28/tcp    0.000050
msg-icp  29/tcp    0.000025      # MSG ICP
msg-icp  29/udp    0.000560      # MSG ICP
unknown  30/tcp    0.000527
msg-auth 31/tcp    0.000025      # MSG Authentication
```



# Nmap

## Specifica delle Porte (nmap-services)

- Ciascuna entry del file **nmap-services** contiene
  - Nome del servizio e numero della porta, insieme al corrispondente protocollo
  - **Valore che rappresenta la probabilità di trovare aperta tale porta**
    - **Probabilità ottenuta tramite euristiche ricavate da scansioni precedenti**

```
ssh      22/tcp    0.182286    # Secure Shell Login
ssh      22/udp    0.003905    # Secure Shell Login
telnet   23/tcp    0.221265
telnet   23/udp    0.006211
priv-mail 24/tcp    0.001154    # any private mail system
priv-mail 24/udp    0.000329    # any private mail system
smtp     25/tcp    0.131314    # Simple Mail Transfer
smtp     25/udp    0.001285    # Simple Mail Transfer
rsftp    26/tcp    0.007991    # RSFTP
nsw-fe   27/tcp    0.000138    # NSW User System FE
nsw-fe   27/udp    0.000395    # NSW User System FE
unknown  28/tcp    0.000050
msg-icp  29/tcp    0.000025    # MSG ICP
msg-icp  29/udp    0.000560    # MSG ICP
unknown  30/tcp    0.000527
msg-auth 31/tcp    0.000025    # MSG Authentication
```



# Nmap

## Specifica delle Porte (nmap-services)

- Ciascuna entry del file **nmap-services** contiene
  - Nome del servizio e numero della porta, insieme al corrispondente protocollo
  - Valore che rappresenta la probabilità di trovare aperta tale porta
    - Probabilità ottenuta tramite euristiche ricavate da scansioni precedenti

Commento relativo al servizio in esecuzione su una determinata porta

```
ssh 22/tcp 0.182286 # Secure Shell Login
ssh 22/tcp 0.03905 # Secure Shell Login
ssh 22/tcp 0.21265
ssh 22/tcp 0.06211
ssh 22/tcp 0.001154 # any private mail system
ssh 22/tcp 0.000329 # any private mail system
smtp 25/tcp 0.131314 # Simple Mail Transfer
smtp 25/udp 0.001285 # Simple Mail Transfer
rsftp 26/tcp 0.007991 # RSFTP
nsw-fe 27/tcp 0.000138 # NSW User System FE
nsw-fe 27/udp 0.000395 # NSW User System FE
unknown 28/tcp 0.000050
msg-icp 29/tcp 0.000025 # MSG ICP
msg-icp 29/udp 0.000560 # MSG ICP
unknown 30/tcp 0.000527
msg-auth 31/tcp 0.000025 # MSG Authentication
```



# Nmap

## Specifica delle Porte

---

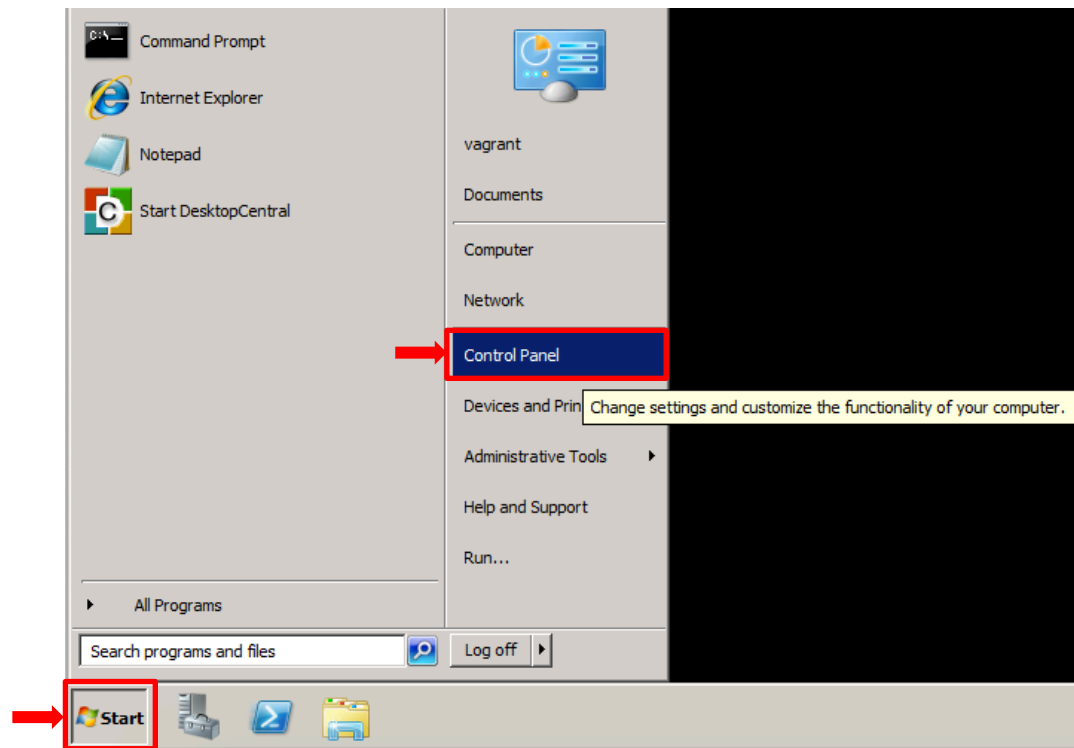
- Nmap consente di scegliere arbitrariamente le porte da scansionare
  - **-p *port\_range***: scansiona le porte definite tramite tale parametro
    - **Esempio 1**: per scansionare le porte da **1** a **1024** l'opzione è **-p 1-1024**
    - **Esempio 2**: per scansionare tutte le porte (da **1** a **65535**) l'opzione è **-p-**
    - **Esempio 3**: per scansionare le porte **21** e **23** l'opzione è **-p 21,23**
  - **-F (*fast*)**: scansiona solo le **100** porte più comuni
    - In base al contenuto del file **nmap-services**
  - **-x (*don't randomize port*)**: scansiona le porte sequenzialmente
    - Da quella con numero più piccolo a quella con numero più grande



# Nmap

## Specifica delle Porte – Esempio

- **Macchina target:** Metasploitable 3
- Disabilitiamo il firewall sulla macchina target



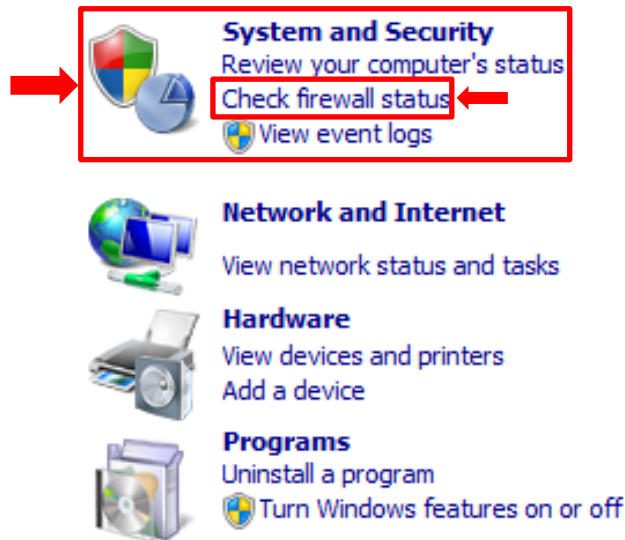
# Nmap

## Specifica delle Porte – Esempio

- **Macchina target:** Metasploitable 3
  - Disabilitiamo il firewall sulla macchina target

### Adjust your computer's settings

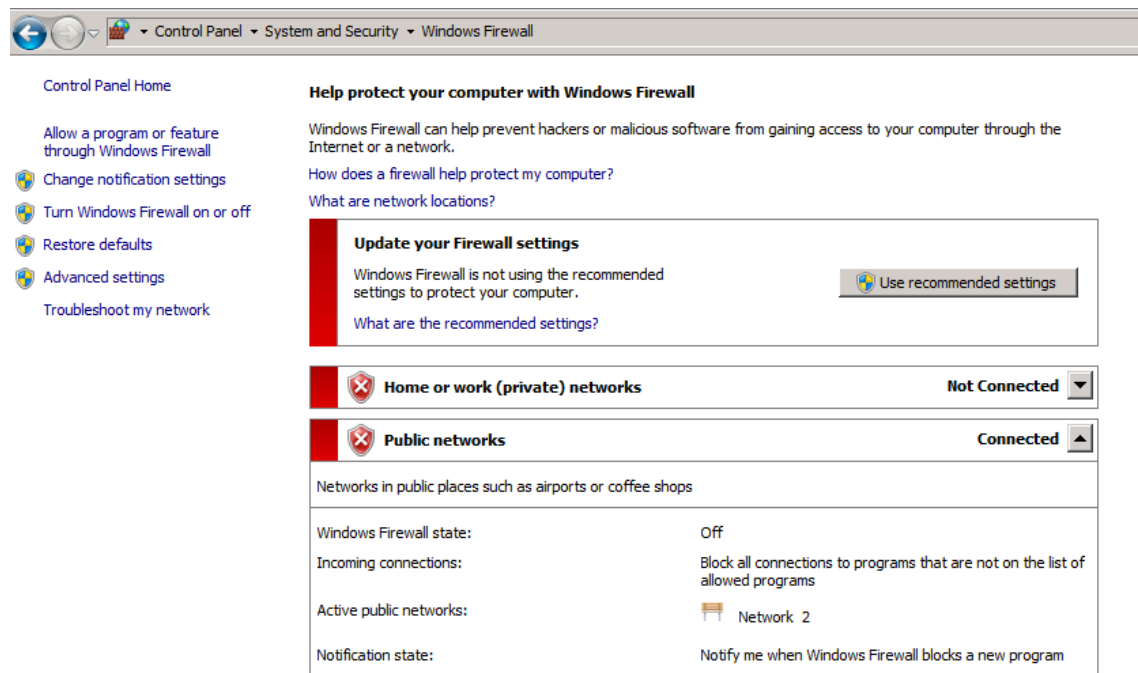
View by: Category ▼



# Nmap

## Specifica delle Porte – Esempio

- **Macchina target:** Metasploitable 3 (Indirizzo IP: 10.0.2.7)
- Disabilitiamo il firewall sulla macchina target





# Nmap

## Specifica delle Porte – Esempio

**nmap 10.0.2.7**

PORT	STATE	SERVICE
22/tcp	open	ssh
135/tcp	open	msrpc
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds
3000/tcp	open	ppp
3306/tcp	open	mysql
3389/tcp	open	ms-wbt-server
4848/tcp	open	appserv-http
7676/tcp	open	imqbrokerd
8009/tcp	open	ajp13
8022/tcp	open	oa-system
8031/tcp	open	unknown
8080/tcp	open	http-proxy
8181/tcp	open	intermapper
8383/tcp	open	m2mservices
8443/tcp	open	https-alt
9200/tcp	open	wap-wsp
49152/tcp	open	unknown
49153/tcp	open	unknown
49154/tcp	open	unknown
49157/tcp	open	unknown
49160/tcp	open	unknown

**Vs.**

**nmap -F 10.0.2.7**

PORT	STATE	SERVICE
22/tcp	open	ssh
135/tcp	open	msrpc
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds
3000/tcp	open	ppp
3306/tcp	open	mysql
3389/tcp	open	ms-wbt-server
8009/tcp	open	ajp13
8080/tcp	open	http-proxy
8443/tcp	open	https-alt
49152/tcp	open	unknown
49153/tcp	open	unknown
49154/tcp	open	unknown
49157/tcp	open	unknown





# Nmap

## Scansione di Default (utente root) – SYN Scan

---

### ➤ Opzione –sS

- Opzione di **scansione predefinita** se Nmap è eseguito da un **utente privilegiato** (*root o amministratore*)
  - Equivale ad invocare **nmap** senza alcuna opzione di scansione
  - **N.B.** Richiede i privilegi di root per poter funzionare
- 
- Nmap invia un pacchetto **SYN** ed attende una risposta da parte della macchina target
    - Se la risposta contiene **SYN/ACK**, allora la porta è «**aperta**»
    - Se la risposta contiene **RST/ACK**, allora la porta è «**chiusa**»
    - Se la risposta contiene un messaggio di errore «**ICMP Port Unreachable**» o se **non c'è alcuna risposta**, la porta è «**filtrata**»



# Nmap

## Scansione di Default (utente root) – SYN Scan

---

### ➤ **Opzione –sS**

- La scansione è eseguita rapidamente
- Scansione nota anche come *half-open* o *SYN stealth*
  - Essa non completa il *three-way handshake*
- Poiché il *three-way handshake* non viene completato, tipicamente tale scansione non viene memorizzata dagli IDS (Intrusion Detection System)



# Nmap

## Traffico Generato da una Scansione di Default (root)

- Per analizzare il traffico di rete generato da una scansione **nmap** utilizziamo **tcpdump**, un semplice ma potente *sniffer* di rete
- Per maggiori informazioni su **tcpdump**
  - **man tcpdump**

Output Parziale

```
TCPDUMP(8)                                System Manager's Manual                                TCPDUMP(8)

NAME
    tcpdump - dump traffic on a network

SYNOPSIS
    tcpdump [ -AbdDefhHIJKLLnNOpqStuUvxx# ] [ -B buffer_size ]
    [ -c count ]
    [ -C file_size ] [ -G rotate_seconds ] [ -F file ]
    [ -i interface ] [ -j tstamp_type ] [ -m module ] [ -M secret ]
    [ --number ] [ -Q in|out|inout ]
    [ -r file ] [ -V file ] [ -s snaplen ] [ -T type ] [ -w file ]
    [ -W filecount ]
    [ -E spi@ipaddr algo:secret.... ]
    [ -y datalinktype ] [ -z postrotate-command ] [ -Z user ]
    [ --time-stamp-precision=tstamp_precision ]
    [ --immediate-mode ] [ --version ]
    [ expression ]

DESCRIPTION
    Tcpdump prints out a description of the contents of packets on a net-
    work interface that match the boolean expression; the description is
```



# Nmap

## Traffico Generato da una Scansione di Default (root)

---

- Utilizzando **tcpdump** è possibile analizzare i seguenti flag impostati da nmap durante i vari tipi di scansione
  - [S] – SYN (*SYN packet*, richiesta per stabilire una nuova sessione)
  - [. ] – ACK (*ACK packet*, conferma di ricezione dei dati del mittente)
  - [P] – PSH (*PuSH*, push immediato dei dati da parte del sender)
  - [F] – FIN (*FINish*, sollecito di terminazione)
  - [U] – URG (*URGent*, ha precedenza sugli altri dati)
  - [R] – RST (*ReSeT*, indicazione di interruzione immediata della connessione)
  - [S . ] – SYN-ACK packet
  - [R . ] – RST-ACK packet

**N.B.** **tcpdump** richiede i privilegi di root per poter funzionare



# Nmap

Traffico Generato da una Scansione di Default (root)

---

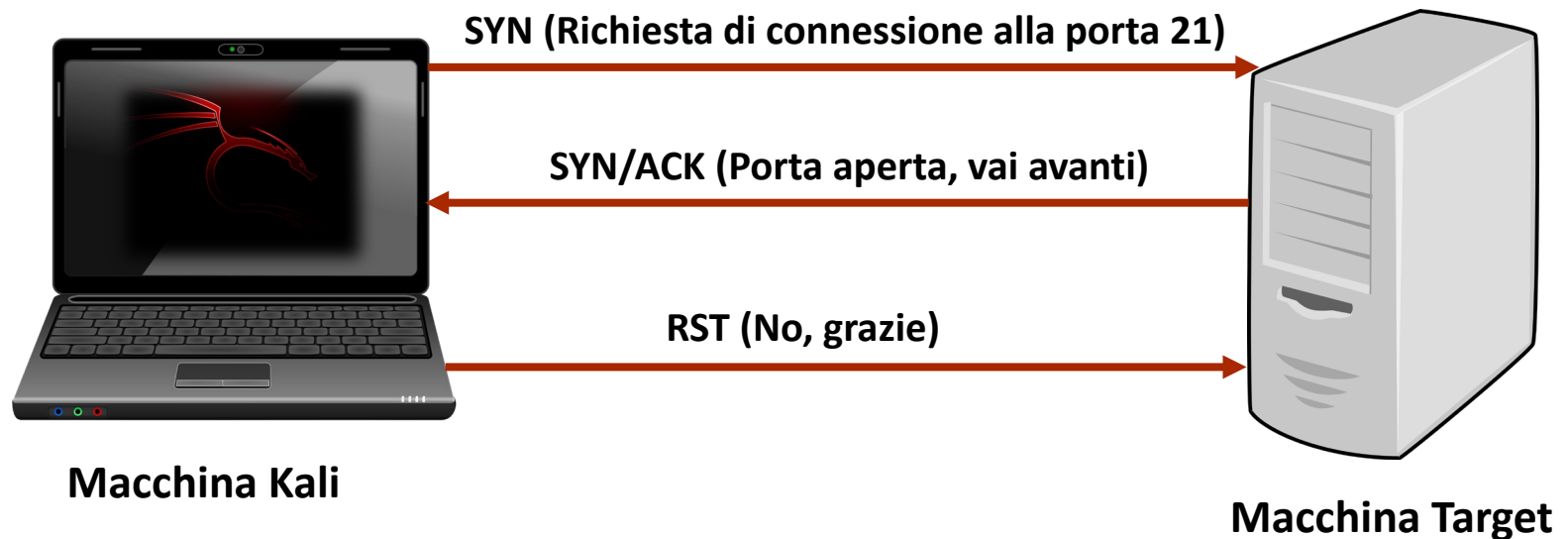
- **Esempio:** Supponiamo di avere il seguente scenario di rete
  - IP macchina Kali: **10.0.2.15**
  - IP macchina target (Metasploitable 2): **10.0.2.6**



# Nmap

Traffico Generato da una Scansione di Default (root)

## ➤ Caso 1: *Porta Aperta*



# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Avviamo **tcpdump** con gli opportuni parametri
  - `tcpdump -nnX tcp and host 10.0.2.15 | grep 10.0.2.6.21`



# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Avviamo `tcpdump` con gli opportuni parametri
  - `tcpdump -nnX tcp and host 10.0.2.15 | grep 10.0.2.6.21`

- **-nn:** utilizza un formato numerico di rappresentazione, sia per i nomi di dominio che per le porte
- **-X:** stampa l'header e i dati di ogni pacchetto, sia in formato ASCII che in formato esadecimale





# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Avviamo `tcpdump` con gli opportuni parametri
  - `tcpdump -nnX tcp and host 10.0.2.15 | grep 10.0.2.6.21`

Protocollo da analizzare



# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Avviamo `tcpdump` con gli opportuni parametri
  - `tcpdump -nnX tcp and host 10.0.2.15 | grep 10.0.2.6.21`

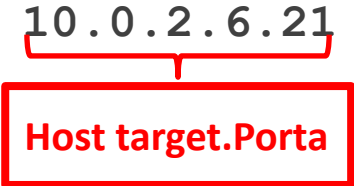
Host sorgente  
(Kali)



# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Avviamo **tcpdump** con gli opportuni parametri
  - `tcpdump -nnX tcp and host 10.0.2.15 | grep 10.0.2.6.21`  




# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Avviamo **tcpdump** con gli opportuni parametri
  - `tcpdump -nnX tcp and host 10.0.2.15 | grep 10.0.2.6.21`

```
root@kali:~# tcpdump -nn tcp and host 10.0.2.15 | grep 10.0.2.6.21
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```



# Nmap

Traffico Generato da una Scansione di Default (root)

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Avviamo **nmap** usando una nuova finestra (o un nuovo Tab) del Terminale ed attendiamo la fine della scansione
  - **nmap 10.0.2.6**

Output Parziale



```
root@kali:~# nmap 10.0.2.6
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-24 21:09 CET
Nmap scan report for 10.0.2.6
Host is up (0.00012s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
```



# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Analizzando l'output di `tcpdump` possiamo osservare quanto segue

```
21:09:31.694937 IP 10.0.2.15.45004 > 10.0.2.6.21: Flags [S], seq 1264759154, win 1024, options [mss 1460], length 0
21:09:31.695047 IP 10.0.2.6.21 > 10.0.2.15.45004: Flags [S.], seq 76123768, ack 1264759155, win 5840, options [mss 1460], length 0
21:09:31.695052 IP 10.0.2.15.45004 > 10.0.2.6.21: Flags [R], seq 1264759155, win 0, length 0
```



# Nmap

Traffico Generato da una Scansione di Default (root)

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Analizzando l'output di **tcpdump** possiamo osservare quanto segue

```
21:09:31.694937 IP 10.0.2.15.45004 > 10.0.2.6.21: Flags [S], seq 1264759154, win 1024, options [mss 1460], length 0
21:09:31.695047 IP 10.0.2.6.21 > 10.0.2.15.45004: Flags [S.], seq 76123768, ack 1264759155, win 5840, options [mss 1460], length 0
21:09:31.695052 IP 10.0.2.15.45004 > 10.0.2.6.21: Flags [R], seq 1264759155, win 0, length 0
```

- **La macchina Kali invia**
  - Un pacchetto contenente il flag SYN = [S] (Start Connection)
  - Il numero di sequenza (ISN) 1264759154



# Nmap

Traffico Generato da una Scansione di Default (root)

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Analizzando l'output di **tcpdump** possiamo osservare quanto segue

```
21:09:31.694937 IP 10.0.2.15.45004 > 10.0.2.6.21: Flags [S], seq 1264759154, win 1024, options [mss 1460], length 0
21:09:31.695047 IP 10.0.2.6.21 > 10.0.2.15.45004: Flags [S.], seq 76123768, ack 1264759155, win 5840, options [mss 1460], length 0
21:09:31.695052 IP 10.0.2.15.45004 > 10.0.2.6.21: Flags [R], seq 1264759155, win 0, length 0
```

- La macchina target risponde con
  - Un pacchetto contenente il flag SYN-ACK = [S.] (SynAck Packet)
  - Il numero di sequenza (ISN) 76123768
  - Un ACK al numero di sequenza ricevuto dalla macchina Kali
    - $1264759154 + 1 = 1264759155$





# Nmap

Traffico Generato da una Scansione di Default (root)

- **Caso 1:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Aperta*)
- Analizzando l'output di `tcpdump` possiamo osservare quanto segue

```
21:09:31.694937 IP 10.0.2.15.45004 > 10.0.2.6.21: Flags [S], seq 1264759154, win 1024, options [mss 1460], length 0
21:09:31.695047 IP 10.0.2.6.21 > 10.0.2.15.45004: Flags [S.], seq 76123768, ack 1264759155, win 5840, options [mss 1460], length 0
21:09:31.695052 IP 10.0.2.15.45004 > 10.0.2.6.21: Flags [R], seq 1264759155, win 0, length 0
```

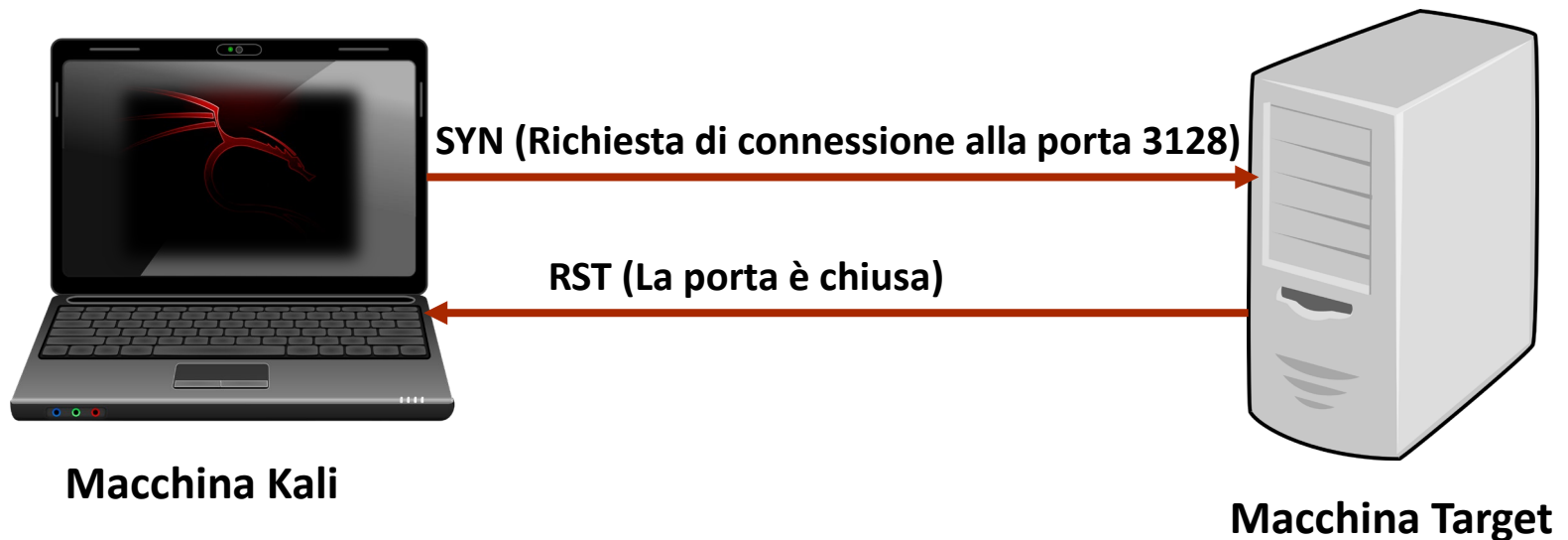
- **La macchina Kali invia**
  - Un pacchetto contenente il flag RST = [R] (Reset Connection)
  - Il numero di sequenza 1264759155 ricevuto dalla macchina target



# Nmap

Traffico Generato da una Scansione di Default (root)

## ➤ Caso 2: *Porta Chiusa*



# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 2:** Traffico generato tra la macchina Kali e la macchina target sulla porta **3128** (*Porta Chiusa*)
- Avviamo **tcpdump** con gli opportuni parametri
  - `tcpdump -nnX tcp and host 10.0.2.15 | grep 10.0.2.6.3128`



# Nmap

Traffico Generato da una Scansione di Default (root)

- **Caso 2:** Traffico generato tra la macchina Kali e la macchina target sulla porta **3128** (*Porta Chiusa*)
- Avviamo **nmap** usando una nuova finestra (o un nuovo Tab) del Terminale ed attendiamo la fine della scansione
  - **nmap 10.0.2.6**

Output Parziale

```
root@kali:~# nmap 10.0.2.6
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-24 21:09 CET
Nmap scan report for 10.0.2.6
Host is up (0.00012s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
```



# Nmap

Traffico Generato da una Scansione di Default (root)

- **Caso 2:** Traffico generato tra la macchina Kali e la macchina target sulla porta **3128** (*Porta Chiusa*)
- Analizzando l'output di `tcpdump` possiamo osservare quanto segue

```
21:32:37.038396 IP 10.0.2.15.47788 > 10.0.2.6.3128: Flags [S], seq 3202025346  
, win 1024, options [mss 1460], length 0  
21:32:37.038591 IP 10.0.2.6.3128 > 10.0.2.15.47788: Flags [R.], seq 0, ack 32  
02025347, win 0, length 0
```

- **La macchina Kali invia**
  - Un pacchetto contenente il flag SYN = [S] (Start Connection)
  - Il numero di sequenza (ISN) 3202025346



# Nmap

Traffico Generato da una Scansione di Default (root)

- **Caso 2:** Traffico generato tra la macchina Kali e la macchina target sulla porta **3128** (*Porta Chiusa*)
- Analizzando l'output di `tcpdump` possiamo osservare quanto segue

```
21:32:37.038396 IP 10.0.2.15.47788 > 10.0.2.6.3128: Flags [S], seq 3202025346  
, win 1024, options [mss 1460], length 0  
21:32:37.038591 IP 10.0.2.6.3128 > 10.0.2.15.47788: Flags [R.], seq 0, ack 32  
02025347, win 0, length 0
```

- La macchina target risponde con
  - Un pacchetto contenente il flag RST-ACK = [R.] (RstAck Packet)
  - Un ACK al numero di sequenza ricevuto dalla macchina Kali
    - $3202025346 + 1 = 3202025347$

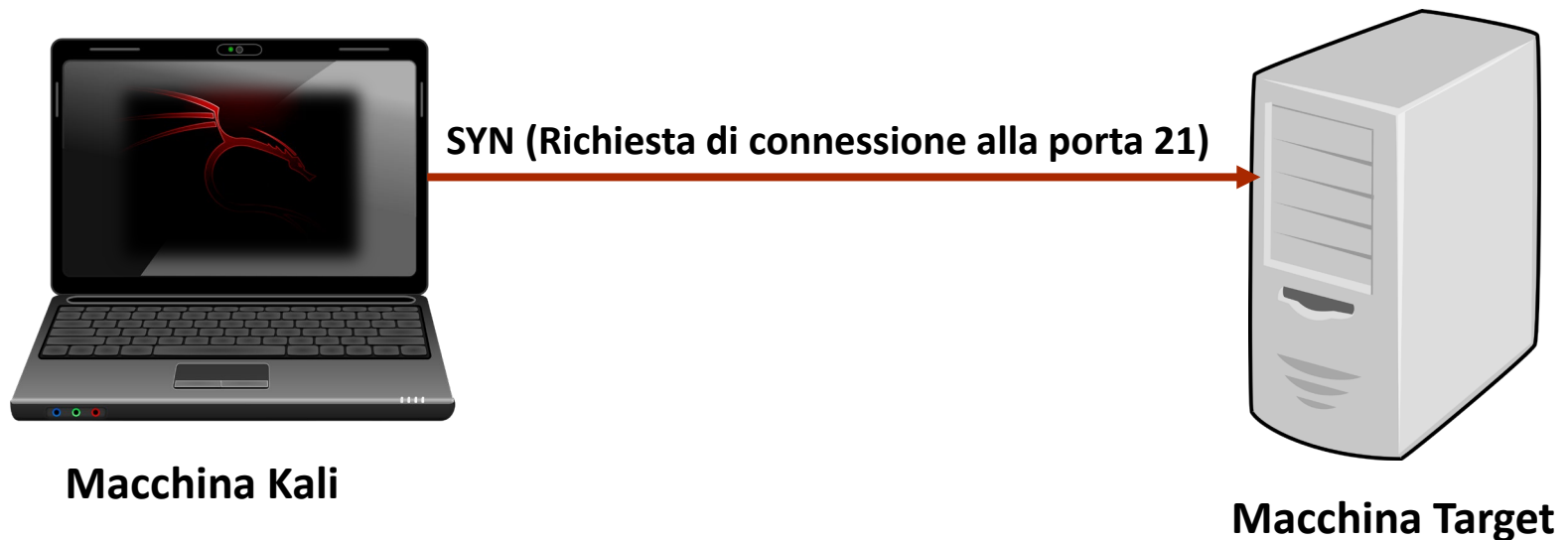


# Nmap

Traffico Generato da una Scansione di Default (root)

---

## ➤ Caso 3: *Porta Filtrata*

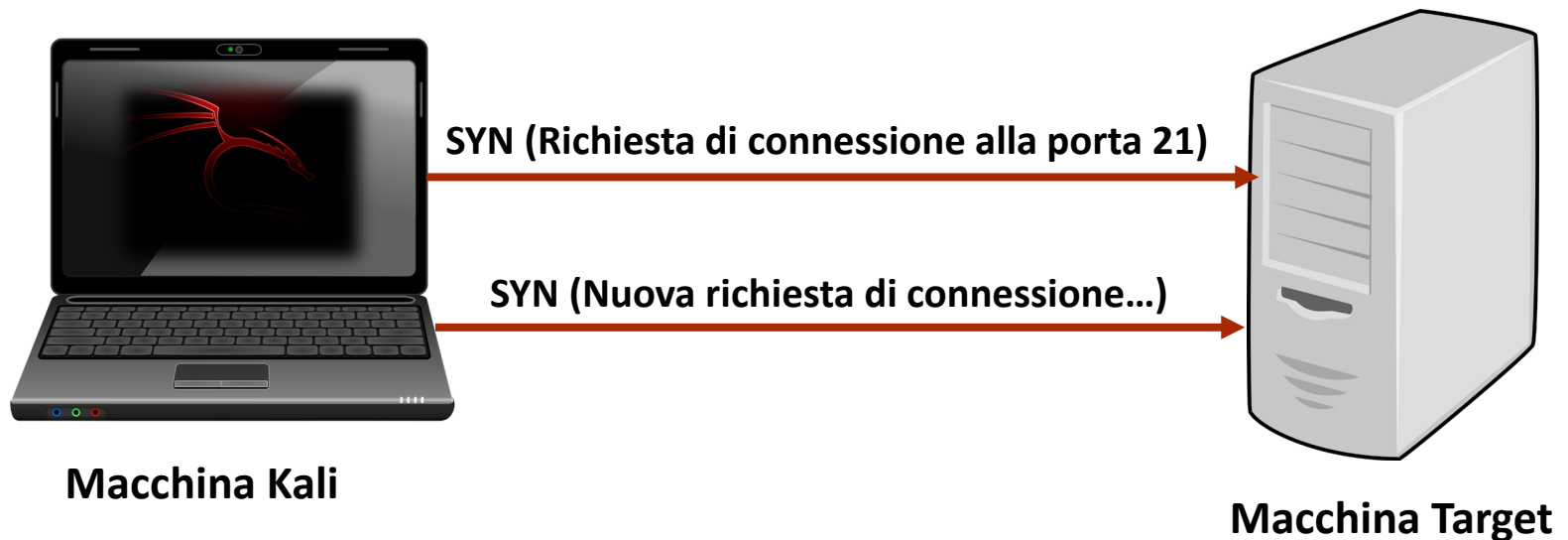


# Nmap

Traffico Generato da una Scansione di Default (root)

---

➤ Caso 3: *Porta Filtrata*

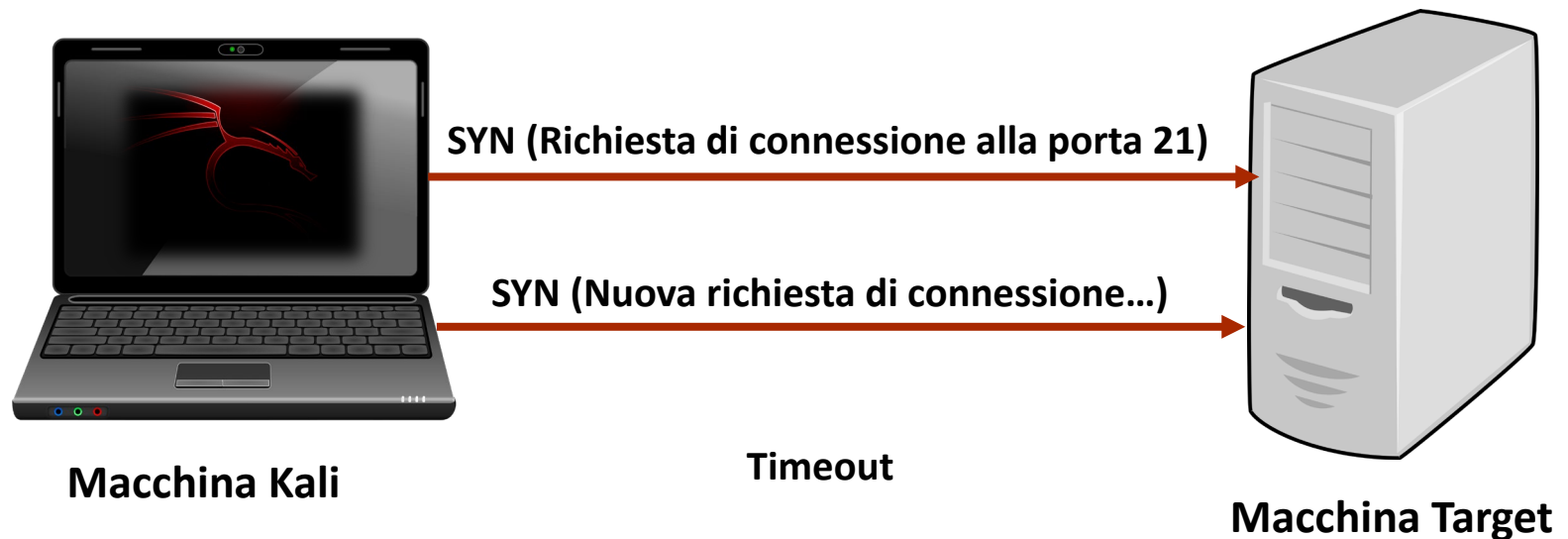




# Nmap

Traffico Generato da una Scansione di Default (root)

## ➤ Caso 3: *Porta Filtrata*



# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 3:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Filtrata*)
- Tramite il firewall **iptables** filtriamo tutte le porte, consentendo solo traffico *TCP* in ingresso verso la porta **22** della macchina target
  - Tutto il resto del traffico sarà bloccato dal firewall



# Nmap

## Traffico Generato da una Scansione di Default (root)

---

- **Macchina target:** Metasploitable 2 (Indirizzo IP: **10.0.2.10**)
- Configuriamo il **firewall** (comando **iptables**) sulla **macchina target** affinché esso
  - Cancelli eventuali politiche di filtro definite precedentemente
    - `iptables -F`
    - `iptables -t nat -F`
    - `iptables -X`
  - Accetti tutti i pacchetti relativi a connessioni sulla porta *TCP* 22 e scarti tutti gli altri
    - `iptables -P FORWARD DROP`
    - `iptables -P INPUT DROP`
    - `iptables -P OUTPUT ACCEPT`
    - `iptables -A INPUT -p tcp --dport 22 -j ACCEPT`



# Nmap

## Traffico Generato da una Scansione di Default (root)

---

- I comandi **iptables** possono essere inseriti in uno script
  - Ad esempio chiamato **iptables.sh**

```
iptables -F
iptables -t nat -F
iptables -X
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Contenuto dello script **iptables.sh**

- Impostiamo i permessi di esecuzione sullo script (**chmod 755 iptables.sh**) e poi lo eseguiamo (**./iptables.sh**)



# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 3:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Filtrata*)
- Avviamo **tcpdump** con gli opportuni parametri
- `tcpdump -nnX tcp and host 10.0.2.11 | grep 10.0.2.10.21`

Host sorgente  
(Kali)



# Nmap

Traffico Generato da una Scansione di Default (root)

---

- **Caso 3:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Filtrata*)
- Avviamo **nmap** usando una nuova finestra (o un nuovo Tab) del Terminale ed attendiamo la fine della scansione
  - **nmap 10.0.2.10**

```
root@kali:~# nmap 10.0.2.10
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-26 17:00 EDT
Nmap scan report for 10.0.2.10
Host is up (0.0025s latency).
➔ Not shown: 999 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:80:B2:70 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 4.41 seconds
```



# Nmap

Traffico Generato da una Scansione di Default (root)

- **Caso 3:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Filtrata*)
- Analizzando l'output di **tcpdump** possiamo osservare quanto segue

```
root@kali:~# tcpdump -nnX tcp and host 10.0.2.11 | grep 10.0.2.10.21
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:15:51.550573 IP 10.0.2.11.64372 > 10.0.2.10.21: Flags [S], seq 4024612871, win 1024, options [mss 1460], length 0
17:15:52.652945 IP 10.0.2.11.64373 > 10.0.2.10.21: Flags [S], seq 4024678406, win 1024, options [mss 1460], length 0
```

- La macchina Kali invia
  - Un pacchetto contenente il flag SYN = [S] (Start Connection)
  - Il numero di sequenza (ISN) 4024612871



# Nmap

Traffico Generato da una Scansione di Default (root)

- **Caso 3:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Filtrata*)
- Analizzando l'output di **tcpdump** possiamo osservare quanto segue

```
root@kali:~# tcpdump -nnX tcp and host 10.0.2.11 | grep 10.0.2.10.21
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:15:51.550573 IP 10.0.2.11.64372 > 10.0.2.10.21: Flags [S], seq 4024612871, win 1024, options [mss 1460], length 0
17:15:52.652945 IP 10.0.2.11.64373 > 10.0.2.10.21: Flags [S], seq 4024678406, win 1024, options [mss 1460], length 0
```

➤ La macchina Kali invia

- Un nuovo pacchetto contenente il flag SYN = [S] (Start Connection)
- Un nuovo numero di sequenza (ISN) 4024678406





# Nmap

Traffico Generato da una Scansione di Default (root)

- **Caso 3:** Traffico generato tra la macchina Kali e la macchina target sulla porta **21** (*Porta Filtrata*)
- Analizzando l'output di `tcpdump` possiamo osservare quanto segue

```
root@kali:~# tcpdump -nnX tcp and host 10.0.2.11 | grep 10.0.2.10.21
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:15:51.550573 IP 10.0.2.11.64372 > 10.0.2.10.21: Flags [S], seq 4024612871, win 1024, options [mss 1460], length 0
17:15:52.652945 IP 10.0.2.11.64373 > 10.0.2.10.21: Flags [S], seq 4024678406, win 1024, options [mss 1460], length 0
```

- Non avendo ricevuto alcuna risposta entro una certa soglia di timeout, `nmap` passa alla scansione della porta successiva

