Network Security
Introduzione alla network security

Alessandro PIZZORNI
Politecnico di Torino

OLICYBER
OLIMPIADI ITALIANE DI CYBERSICUREZZA

CYBERSECURITY
NATIONAL LAB

https://cybersecnatlab.it

License & Disclaimer

License Information

This presentation is licensed under the Creative Commons BY-NC License

To view a copy of the license, visit:

http://creativecommons.org/licenses/by-nc/3.0/legalcode

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Obiettivi

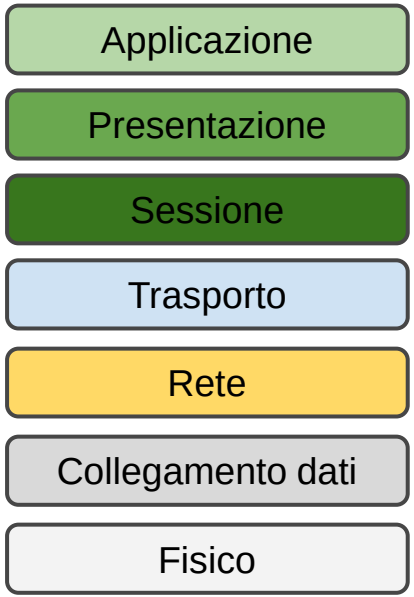
- Imparare le basi teoriche per affrontare challenge di network security
- Imparare a monitorare e salvare il traffico di rete
- Essere in grado di analizzare il traffico di rete attraverso Wireshark
- Imparare le basi di Pyshark per automatizzare il parsing di un file pcap

Indice

Modello ISO/OSI

- Sviluppato nel 1984 dall'International Organization for Standardization (ISO)
- Fornisce la logica per la comunicazione tra dispositivi all’interno di reti di calcolatori
- Non è lo standard implementato abitualmente su Internet

OSI Layers



Una guida per lo sviluppo di protocolli di rete.

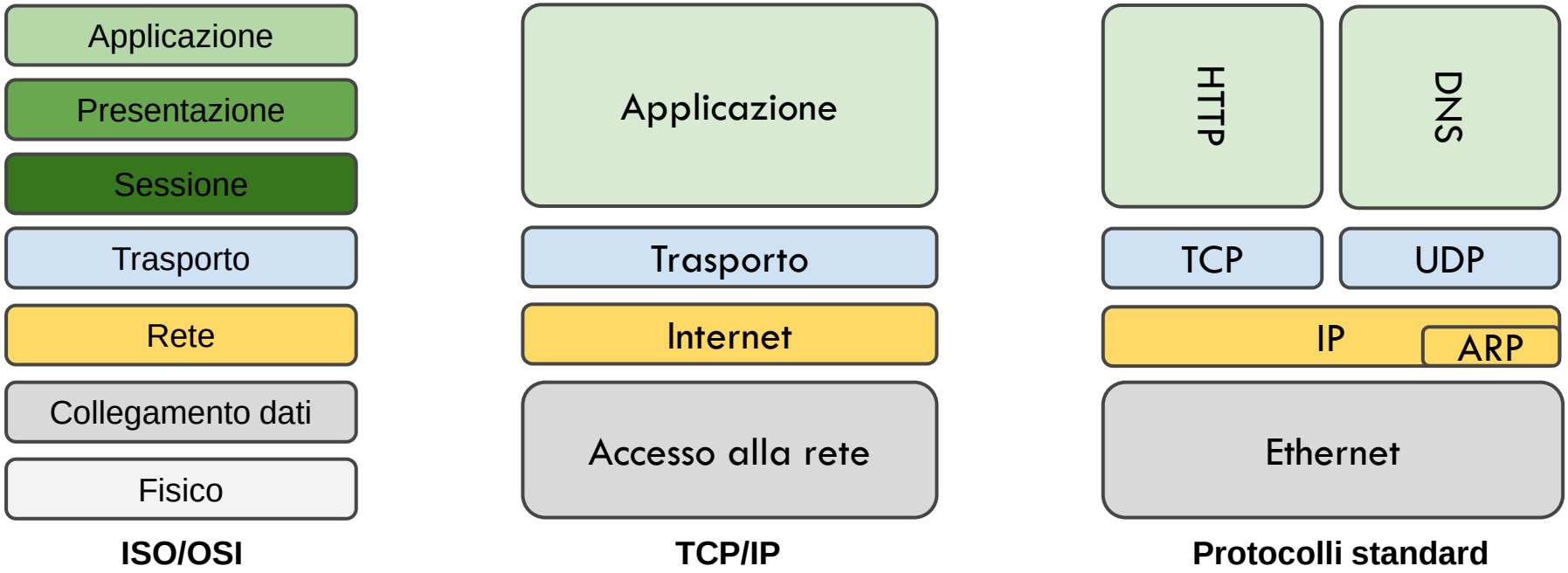
- Sette layers (strati)
- Basato su un approccio *divide et impera*: ogni problema viene risolto all'interno di un singolo layer
- Forte flessibilità: può adattarsi a nuovi protocolli e servizi di rete
- Non è necessario che le implementazioni si basino su tutti i livelli

Modello TCP/IP

Il modello TCP/IP è un insieme di protocolli utilizzati per la comunicazione all'interno di reti.

- È basato su quattro livelli
- È basato sui protocolli sviluppati e applicati su Internet (quindi è protocollo-dipendente)
- È lo standard utilizzato su Internet

Modello TCP/IP



Internet Protocol (IP)

L'Internet Protocol (IP) è il protocollo utilizzato nel layer 3 (rete).

- È lo standard per l'instradamento di pacchetti all'interno delle reti
- Incapsula i dati e li passa in forma di pacchetti
- Ogni pacchetto include:
  - l'indirizzo IP della sorgente
  - l'indirizzo IP del destinatario

Indirizzamento IP

- Gli indirizzi IP, nella versione 4, sono:
  - Formati da 32 bits
  - Raggruppati in quattro ottetti da 8 bit
  - Rappresentati da un numero decimale, ogni ottetto è separato dagli altri attraverso un punto

11000000 10101000 01100100 11001000

192 . 168 . 100 . 200

Indirizzi IP

192.168.100.200
(indirizzo dell'host)

192.168.100
(indirizzo di rete)

2^{32}

IPv4
indirizzi
disponibili

35
(civico)

via Dodecaneso
(via)

TCP vs UDP

- TCP e UDP sono I protocolli più utilizzati nel layer 4
  - TCP prima crea una connessione, poi invia il contenuto della comunicazione. UDP trasmette direttamente dati
  - Entrambi controllano errori attraverso checksum, ma UDP non fa nessuna azione a riguardo. TCP invece è in grado di correggere gli errori
  - TCP ordina I pacchetti, UDP invece non ha un ordine predefinito per I pacchetti
  - I pacchetti TCP sono più grandi in termini di spazio rispetto ai pacchetti UDP

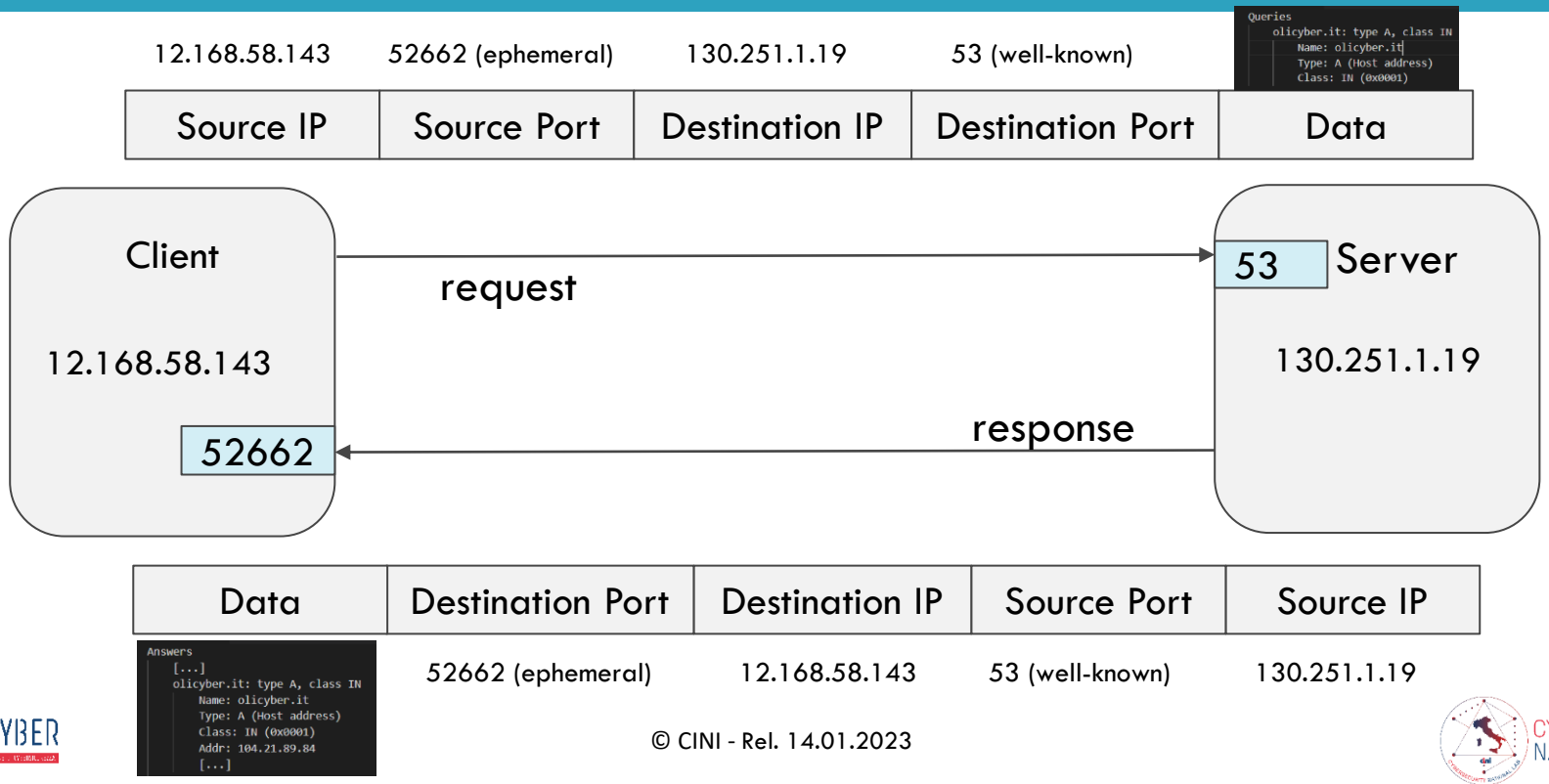
Porte

- Il layer 4 si occupa anche della comunicazione tra processi remoti
- Questi processi sono identificati attraverso porte così definite:
  - 16-bit unsigned integer (0-65535, 0 riservato)
  - Well-known ports (0-1023): usate da processi di Sistema particolarmente rilevanti
  - Registered ports (1024-49151): assegnate da IANA a particolari applicazioni previa registrazione
  - Ephemeral ports (49152–65535): porte dinamiche o per servizi private.
- Nonostante questa convenzione, qualsiasi servizio può ascoltare su qualsiasi porta!

Il modello client-server

- Il modello client-server è il paradigma più utilizzato per la comunicazione all'interno di reti
  - Si ha una relazione per cui un programma (client) richiede un servizio o una risorsa a un altro programma (server)
  - Il client deve conoscere l'indirizzo del server
  - Il server non per forza deve conoscere l'indirizzo (o l'esistenza) del client prima della connessione

Il modello client-server (esempio DNS)



Salvare il traffico di rete

- Per poter analizzare il traffico di rete passato è necessario prima salvarlo (dump del traffico)
- Tcpdump (https://github.com/the-tcpdump-group/tcpdump) è un esempio di tool che permette di visualizzare e salvare il traffico di rete
- Generalmente il traffico viene salvato all'interno di file con formato Packet Capture dall'estensione .pcap

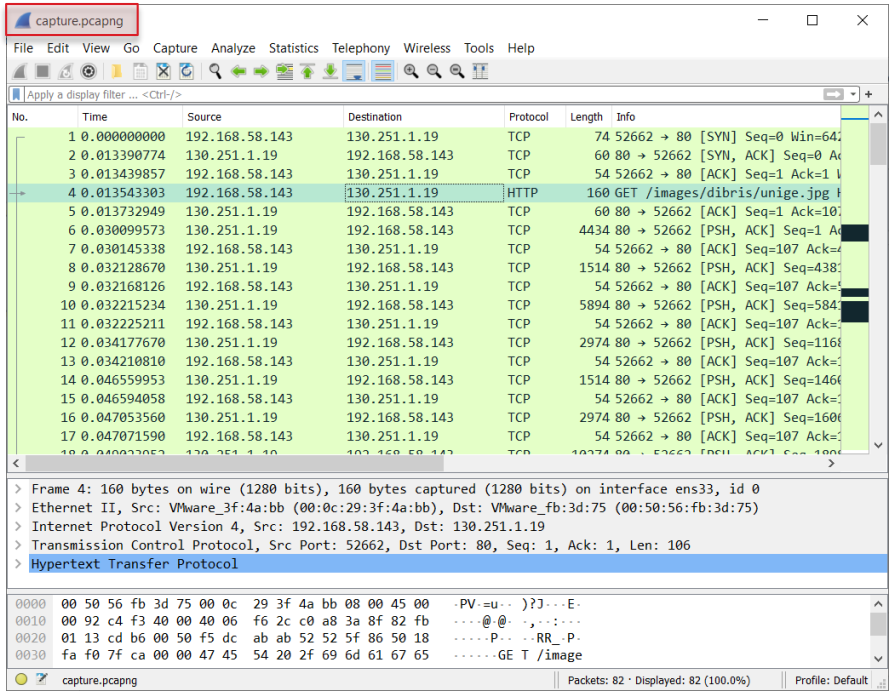
Wireshark

- Wireshark è un tool che permette di catturare traffico da una rete (sniffer) e analizzarlo
  - L'analisi può essere effettuata real-time oppure usando un file precedentemente salvato
  - I pacchetti sono composti da dati *generic*, essi andranno poi valutati livello per livello per estrapolarne informazioni
- Disponibile sia su Linux che Windows:
  - https://www.wireshark.org/

Challenges di network

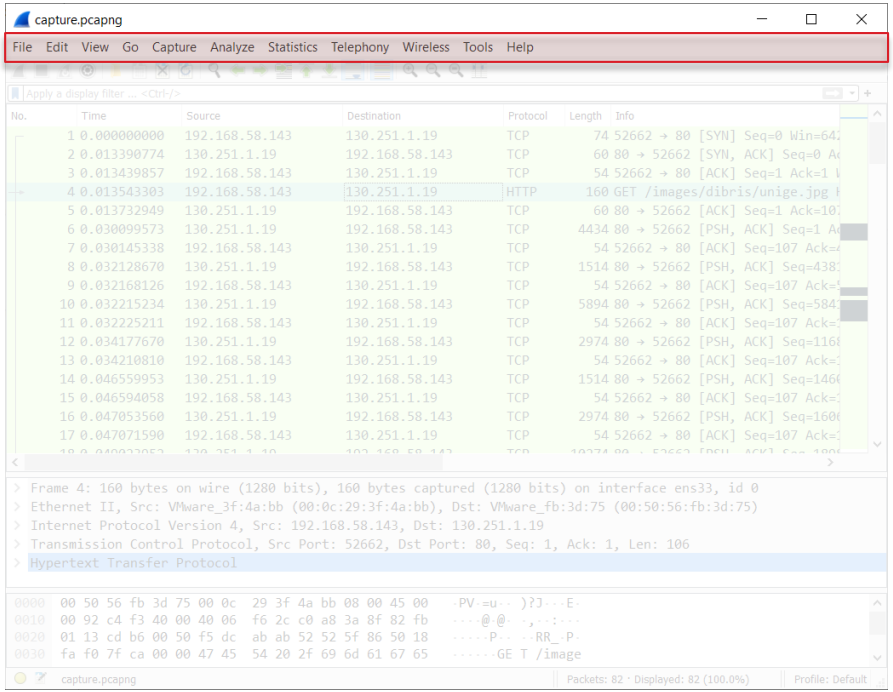
- Nella maggior parte delle sfide di network security vengono forniti ai giocatori dei file PCAP
- Per risolvere queste challenge i giocatori devono saper analizzare questi file per
  - Trovare la flag direttamente tra i byte
  - Rispondere a delle domande relative al traffico analizzato
- Wireshark è uno strumento utile per risolvere questo tipo di sfide

Wireshark GUI



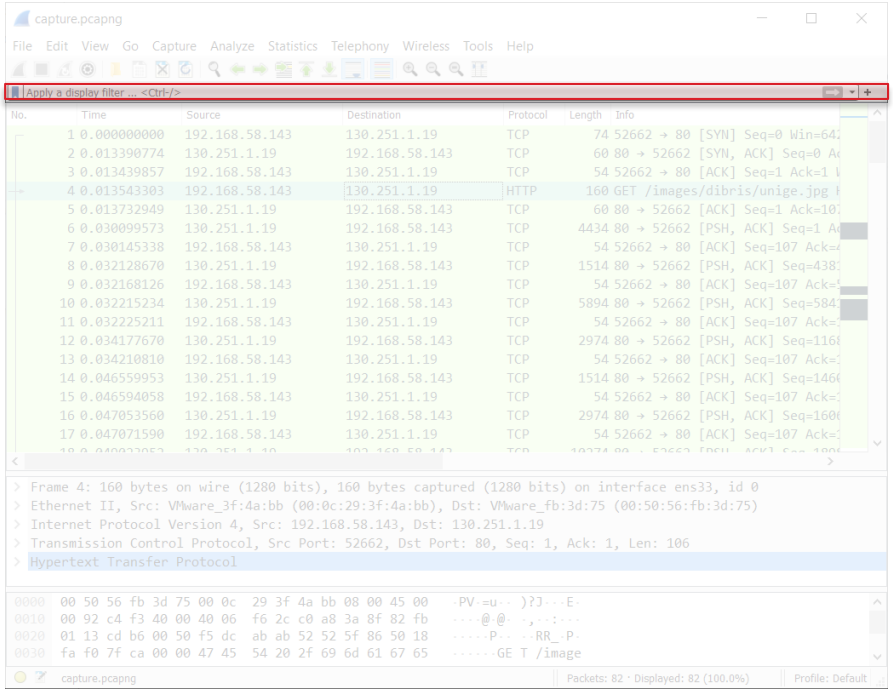
- Wireshark ha una interfaccia grafica (Graphical User Interface GUI)
- È possible aprire un file .pcap da analizzare dal menu *File* oppure utilizzando il comando open (CTRL+o)
  Esso apparirà nella schermata principale
- Utilizzando una versione di Wireshark in inglese è più facile cercare aiuto e informazioni su Internet

Wireshark GUI: menu



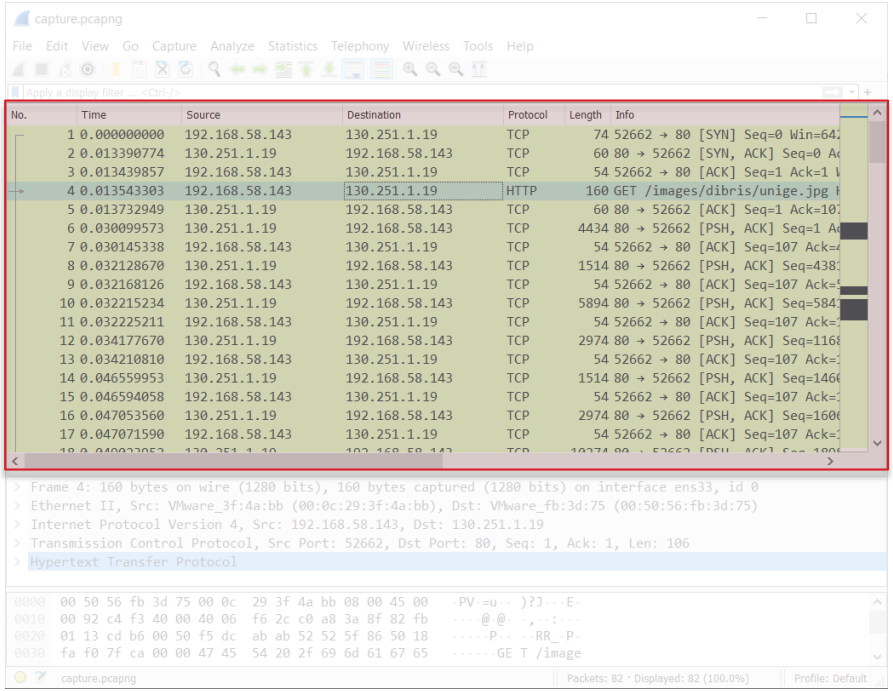
- Il menu è utilizzato per effettuare azioni
- Le azioni di maggiore interesse sono:
  - *File*: aprire o raggruppare file, salvare, stampare o esportare dati
  - *Edit*: trovare un pacchetto, evidenziare flussi, gestire le configurazioni
  - *View*: controllare come vengono visualizzati i pacchetti (colore, font …)
  - *Go*: andare ad un determinato pacchetto
  - *Analyze*: manipolare, filtrare, attivare o disattivare il focus su determinati protocolli, seguire i flussi (stream)
  - *Statistics*: visualizzare diverse statistiche quali gli indirizzi IP coinvolti nella comunicazione, numero di pacchetti scambiati etc. Utili per farsi un idea iniziale sul tipo di traffico contenuto in quel file aperto

Wireshark GUI: filter toolbar



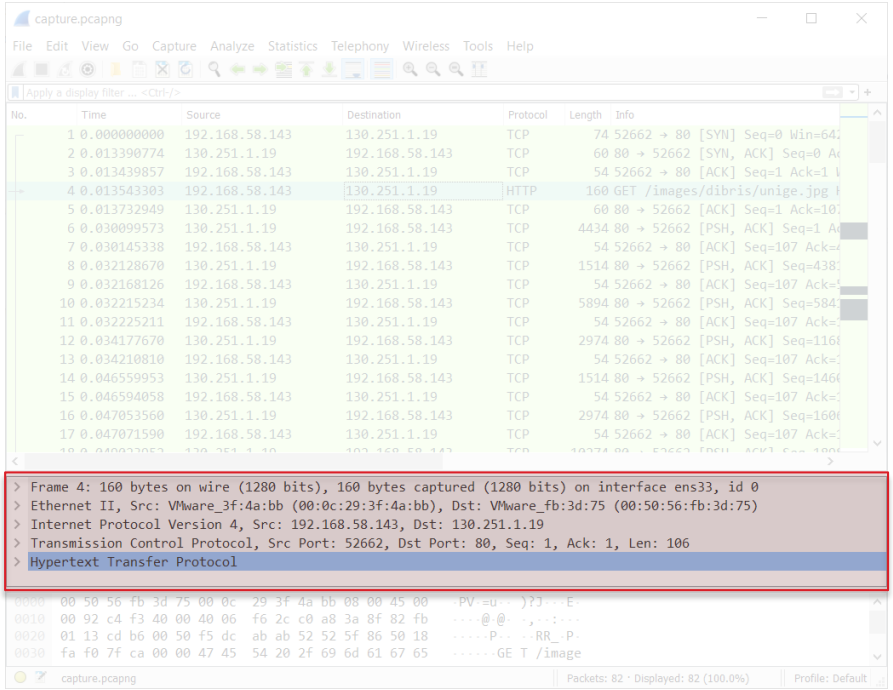
- Il menu dei filtri permette di modificare e applicare rapidamente dei filtri sui pacchetti
  - Gestire o salvare filtra salvati
  - Reset dei filtri
  - Applicare il filtro corrente
  - Selezionare un filtro d una lista di filtri usati di recente
  - Aggiungere un nuovo filter button (shortcup per applicare un determinato filtro)

Wireshark GUI: lista dei pacchetti



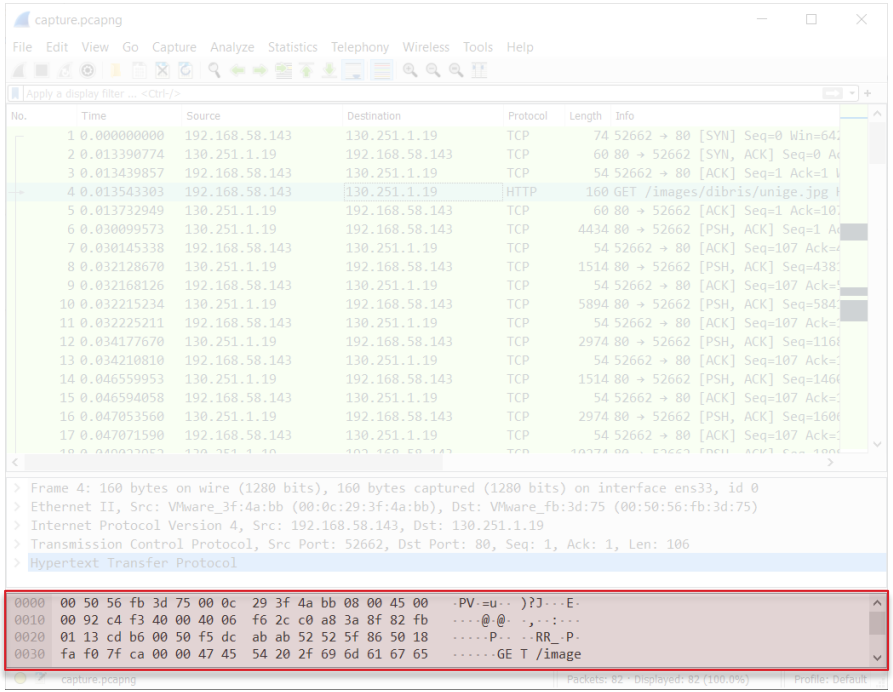
- Il pannello centrale mostra la lista di tutti i pacchetti catturati
- Ogni linea corrisponde a un pacchetto catturato
- Selezionando un pacchetto (singolo click) vengono visualizzati I dettagli del pacchetto all'interno della sottostante sezione *packet details* e *packet bytes*
- Si può cliccare sulle colonne per ordinare i pacchetti

Wireshark GUI: packet details



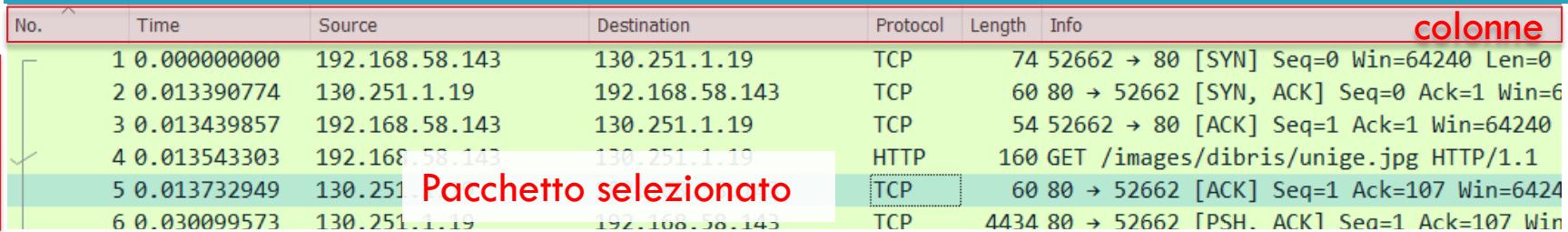
- Il pannello sottostante alla lista dei pacchetti mostra i dettagli del pacchetto selezionato
- In particolare mostra i protocolli e i campi del pacchetto con una struttura ad albero
  Ciascun ramo corrisponde ad un protocollo e può essere espanso per visualizzare i corrispondenti dati contenuti all'interno

Wireshark GUI: packet bytes



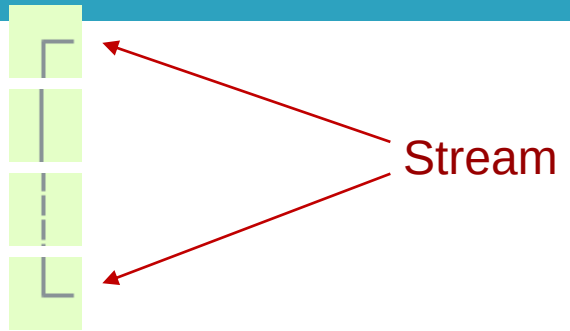
- Il pannello contenente i *packet bytes* mostra i dati del pacchetto selezionato in stile *hexdump*
- Ciascuna linea contiene:
  - L'offset dei dati
  - 16 byte rappresentati in base esadecimale
  - 16 caratteri ASCII (i caratteri non stampabili vengono rappresentati con un punto ".")

La lista dei pacchetti



1. **No.** Numero del pacchettto all'interno del file. Anche se vengono applicati dei filtri questo numero non cambia.
2. **Time** Timestamp del pacchetto (per cambiare formato andare su View → Time display format)
3. **Source** Indirizzo IP del mittente
4. **Destination** Indirizzo IP del destinatario
5. **Protocol** Nome del protocollo
6. **Length** Lunghezza del pacchetto
7. **Info** Informazioni riguardo il contenuto del pacchetto

Simboli per pacchetti collegati (stesso flusso/stream)

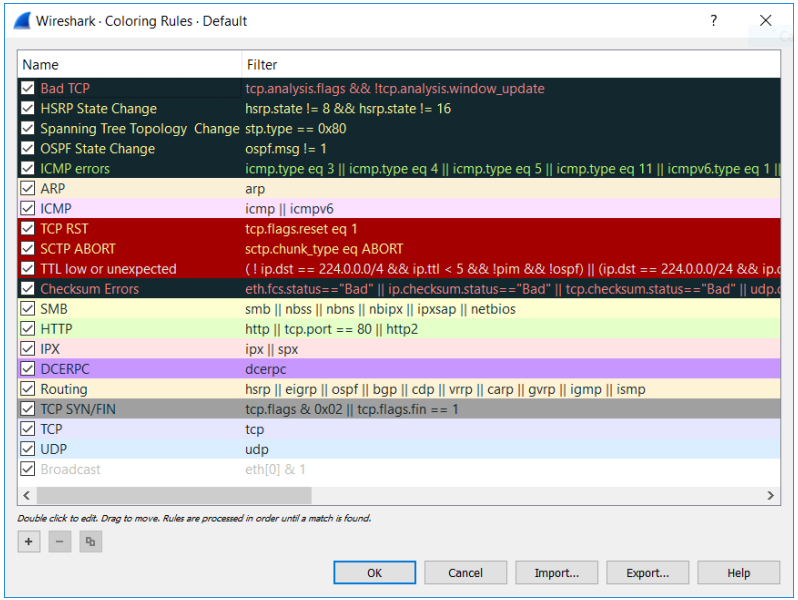


- Primo pacchetto del flusso
- Parte del flusso selezionato
- Non parte del flusso selezionato
- Ultimo pacchetto del flusso
- Richiesta
- Risposta

- Il pacchetto selezionato è una conferma di ricezione di questo pacchetto
- Il pacchetto selezionato è un duplicato di conferma di ricezione di questo pacchetto
- Il pacchetto selezionato ha a che fare con questo pacchetto (as esempio parte del contenuto)

Evidenziare pacchetti con colori

- Wireshark permette di evidenziare i pacchetti con colori diversi basandosi su regole (ad esempio protocolli diversi colori diversi)
- Per visualizzare la configurazione o modificarla
  - View → Coloring Rules...

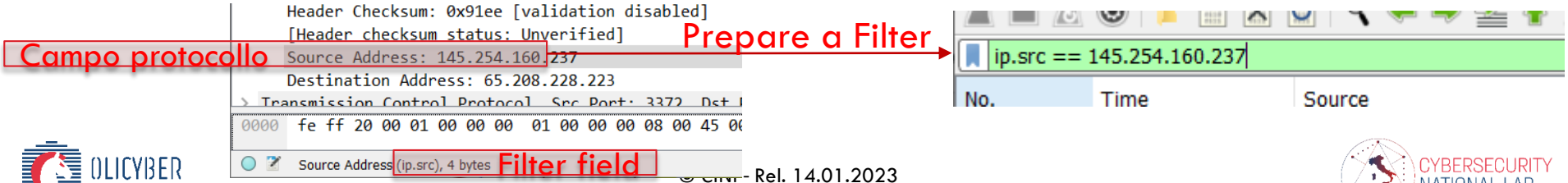


Filtri

- Wireshark fornisce un linguaggio per gestire i filtri
- Con i filtri è possibile controllare quali pacchetti andare a visualizzare
- I filtri possono essere utilizzati per:
  - Visualizzare solo pacchetti di un determinato protocollo
  - Cercare pacchetti con un campo con un determinato valore
  - […]
- I filtri possono essere combinati formando espressioni complesse, utilizzando operatori logici e parentesi

Creare un filtro

1. Help → Manual Pages → Wireshark Filters
2. Expression builder: click con il tasto destroy sulla toolbar → Display Filter Expression...
3. Selezionare il campo sul pannello *packet details:*
    1. Apply as filter: filtra la lista dei pacchetti con solo quelli che soddisfano l'espressione
    2. Prepare as filter: scrive l'espressione ma essa non viene ancora applicata alla lista dei pacchetti

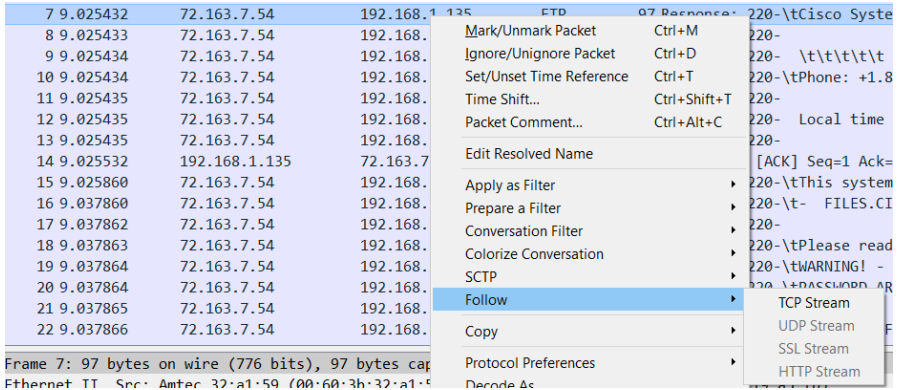


Filtri utili

- `ip.src` / `ip.dst` → filtra rispettivamente per l'indirizzo di origine e di destinazione
- Per filtrare per protocollo basta scrivere il nome che compare nella ripettiva colonna, tutto in lettere minuscule
- `protocol.port` → filtra per porta sul protocollo specificato, sostituire 'protocol' con il protocollo desiderato in letter minuscule
- `frame contains "stringa"` → filtra tutti I pacchetti che contengono "stringa"
- `frame.len` → filtra per la lunghezza (dimensione) del pacchetto, espressa in bytes
- Si riportano link della documentazione per approfondire:
  - https://wiki.wireshark.org/DisplayFilters
  - https://www.wireshark.org/docs/man-pages/wireshark-filter.html

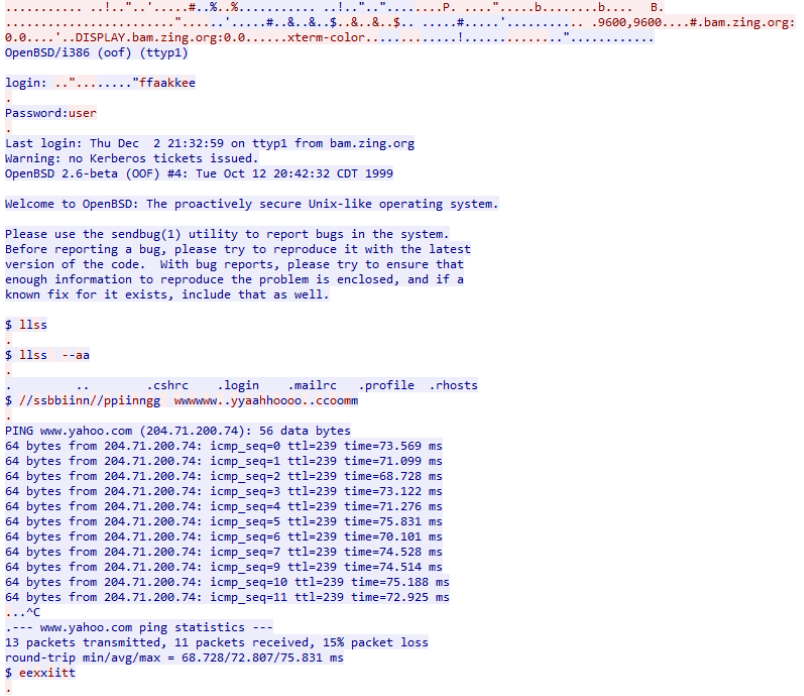
Seguire stream

- Seguire uno stream mostra una diversa visualizzazione del traffico di rete: anziché visualizzare un pacchetto singolo, vengono visualizzati I dati trasmessi tra mittente e destinatario
- Quando viene visualizzato uno stream, un filtro relativo allo stream corrente viene applicato. Solo i pacchetti di quello stream verranno visualizzati



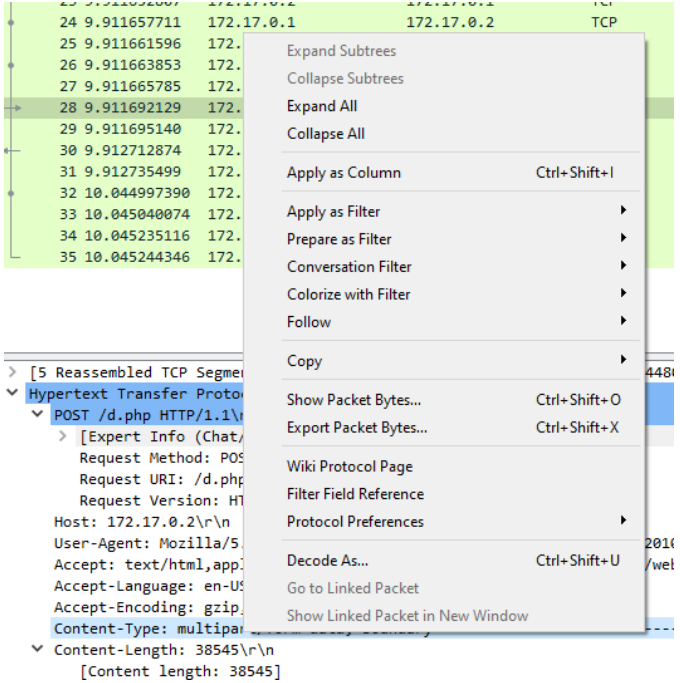
Seguire uno stream (esempio)

- Telnet è un protocollo di tipo client-server che può essere utilizzato per aprire la linea di commando su un host remoto
- In blu vengono visualizzati i dati dal server al client (ad esempio il prompt di login)
- In rosso vengono visualizzati i dati dal client al server (ad esempio il client che invia la password al server)
- I caratteri non stampabili vengono rappresentati con il punto “.”



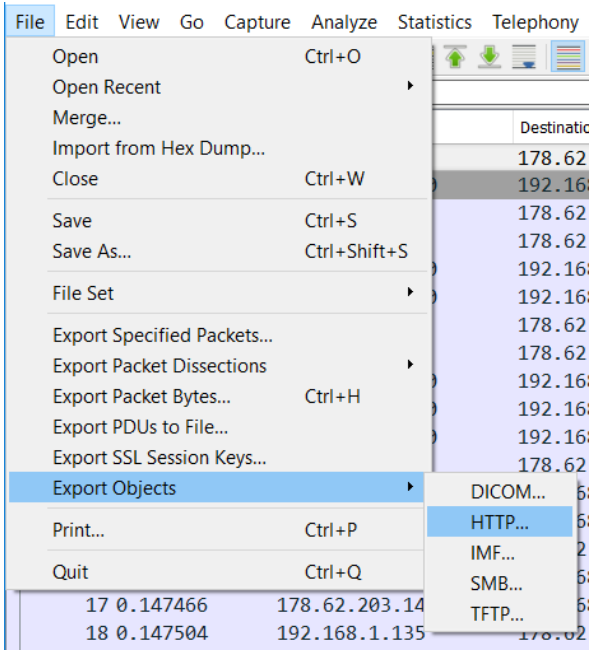
Estrarre artefatti dagli stream: esempio

- Estrarre e salvare un file JPEG scaricato usando HTTP
  1. Selezionare il pacchetto
  2. Andare sul packet bytes del pacchetto
  3. Cliccare con il tasto destro sul campo contenente l'artefatto
  4. Cliccare su *Export packet Bytes*
  5. Salvare il file



Estrarre artefatti: esempio 2

- File → Export Objects
- Questa features analizza gli stream di alcuni protocolli e ricostruisce alcuni oggetti come le pagine HTML, immagini etc…
- Questi file possono essere esportati e salvati su disco



Pyshark

- Pyshark è un wrapper di tshark (versione di Wireshark per riga di comando) per Python
- Utile per automatizzare operazioni sul molti pacchetti
- Installazione:

```
sudo apt install tshark #se non già presente
pip3 install pyshark
```

Pyshark: utilizzo (1)

- Per prima cosa è necessario importare la libreria `import pyshark` e caricare il file pcap:

```
cap = pyshark.FileCapture('/path/to/pcap/file.pcap')
```

- La variabile `cap` è una lista che contiene tutti i pacchetti presenti all'interno del file di cattura. Quindi, per accedere a un pacchetto basta accedere all'elemento della lista `cap[n]`. Attenzione: in Wireshark il primo pacchetto è il numero 1, mentre in Pyshark è lo 0, quindi bisogna sottrarre 1 al numero del pacchetto di Wireshark per ottenere il corrispettivo in Pyshark (e aggiungere 1 nel caso opposto)
- Con un semplice for-loop è possibile ciclare tra tutti i pacchetti della cattura: `for packet in cap:`

Pyshark: utilizzo (2)

- I pacchetti sono divisi in layer. È necessario accedere al layer appropriato prima di selezionare il campo desiderato:
  - `packet.ip.dst` → accede al layer IP del pacchetto e seleziona il campo dell'IP di destinazione
  - `packet.tcp.payload` → accede al layer TCP del pacchetto e seleziona il campo del payload
- Per verificare se un layer è presente nel pacchetto si può utilizzare il nome del layer: `if 'IP' in packet:`
- Per visualizzare tutti possibili campi, si può usare l'attributo `packet.layer.field_names` (es: `packet.ip.field_names`)

Pyshark: esempio

Semplice script per stampare a schermo (come bytestring) il payload di tutti i pacchetti tcp di un file:

```
import pyshark
cap = pyshark.FileCapture('/path/to/pcap/file.pcap')

for packet in cap:
    # Controllo se il pacchetto è TCP (controllo è case insensitive)
    if 'tcp' in packet:
        try:
            print(packet.tcp.payload.binary_value)
        except:
             continue
```

Network Security
Introduzione alla network security

Alessandro PIZZORNI
Politecnico di Torino

OLICYBER
OLIMPIADI ITALIANE DI CYBERSICUREZZA

CYBERSECURITY
NATIONAL LAB

https://cybersecnatlab.it