CYBERSECURITY NATIONAL LAB

CYBER HIGH SCHOOLS

SERICS
SECURITY AND RIGHTS IN THE CYBERSPACE

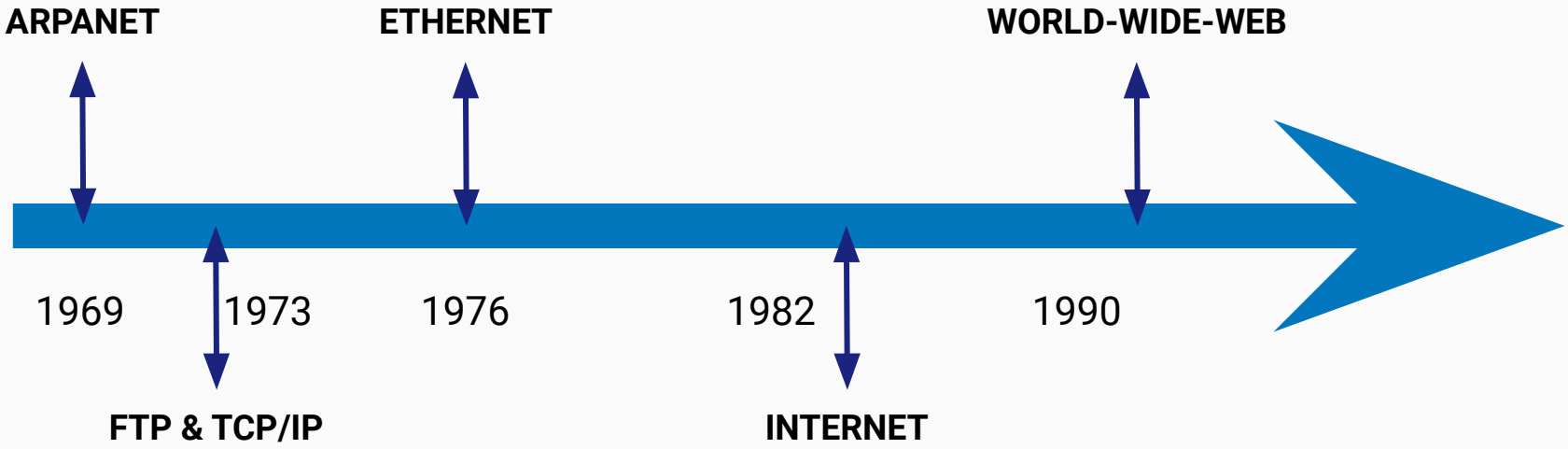
1st HighSchools CTF Workshop

Network Security

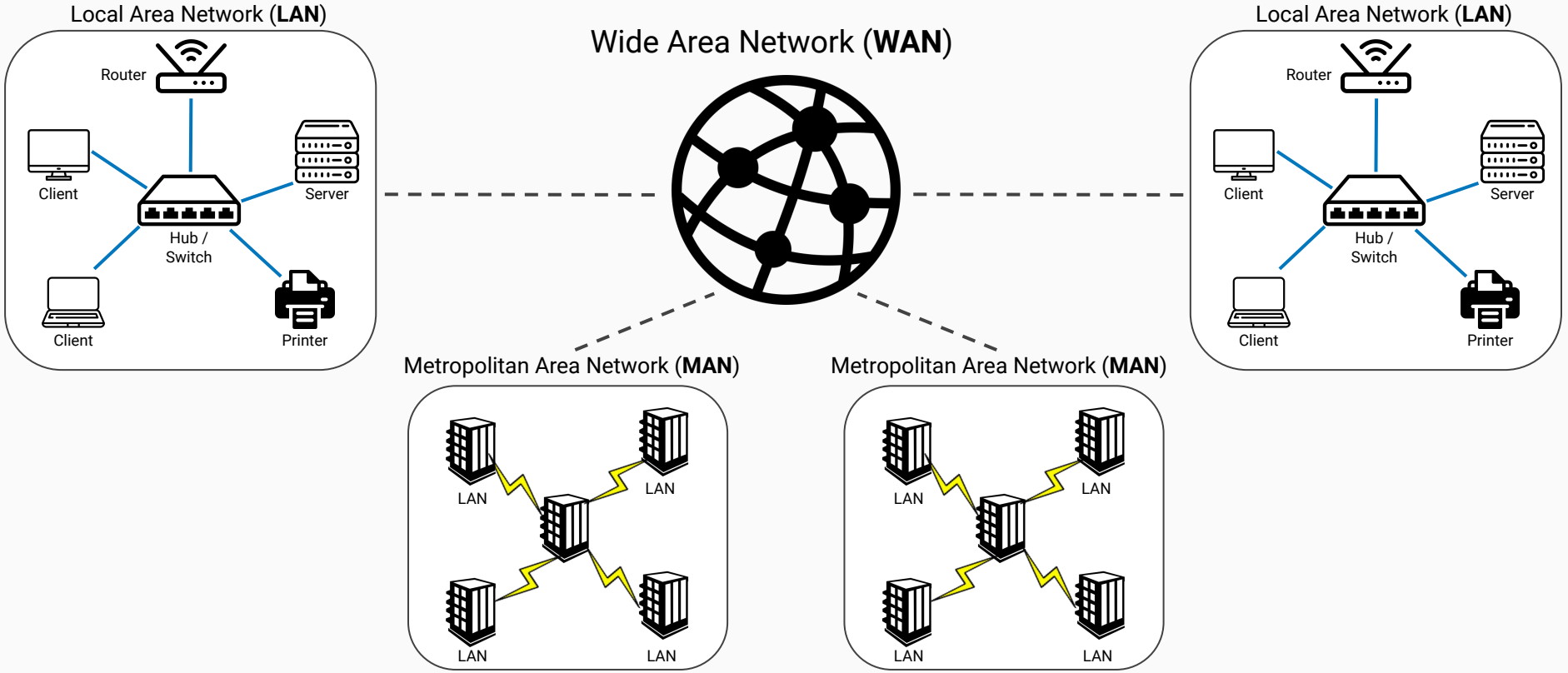
ITASEC23
ITALIAN CONFERENCE ON CYBERSECURITY
May 2nd-5th, 2023 - Bari

Claudio Ripa - Università di Bari
Nicola Guerrera - Politecnico di Bari

Un po’ di storia



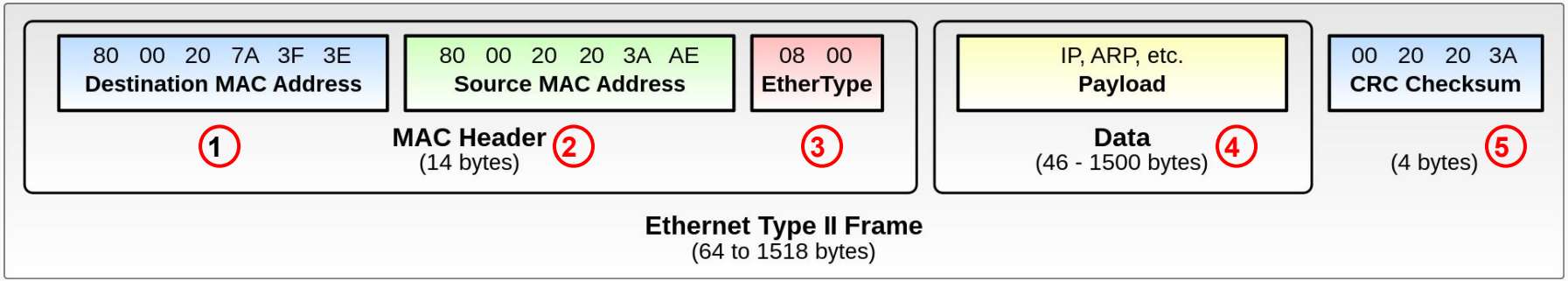
Internet: rete di reti



Modello ISO/OSI

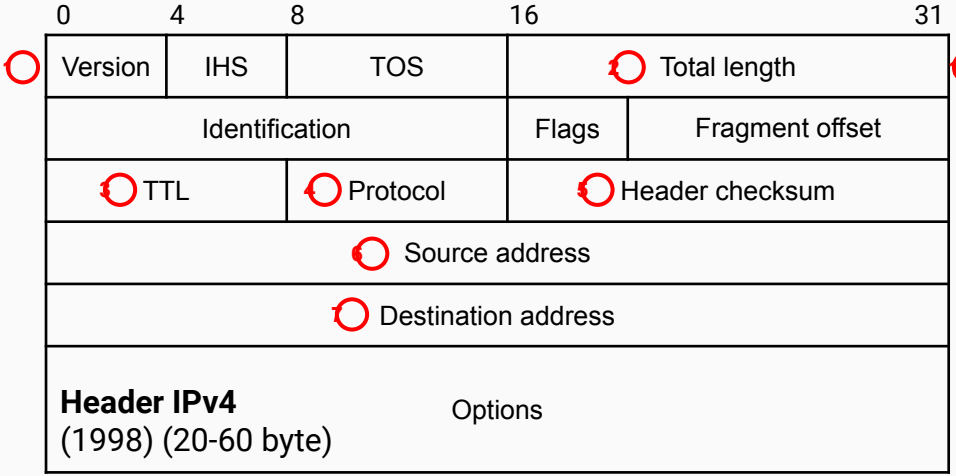
- Astrae la comunicazione tra sistemi attraverso un modello a 7 **livelli**
- Ogni livello **incapsula** il livello successivo
- Il livello successivo rappresenta il **payload** del livello precedente
- Ogni livello offre una serie di **servizi** ai livelli successivi
- Ogni livello utilizza i **servizi** offerti dai livelli precedenti

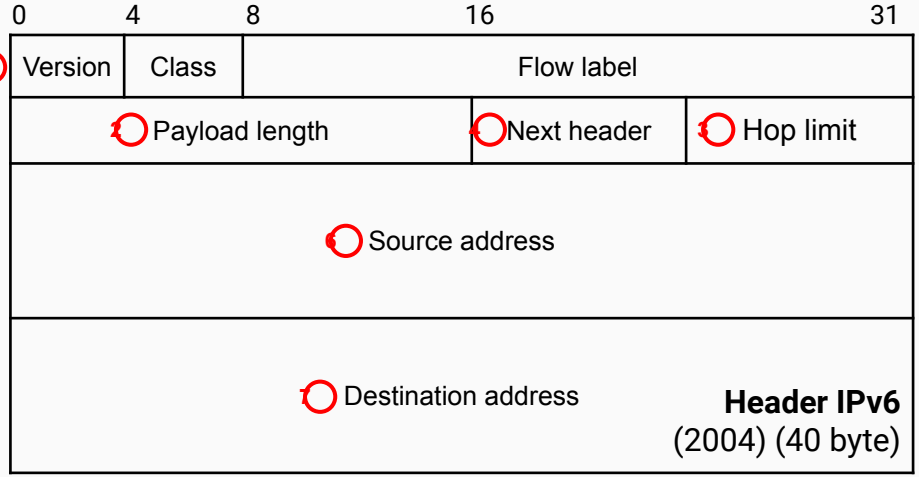
Livello 2: collegamento



1) Indirizzo fisico del **destinatario** del frame
2) Indirizzo fisico del **mittente** del frame
3) **Protocollo** dei dati incapsulati nel payload
4) **Dati** incapsulati
5) Codice per la verifica degli **errori**

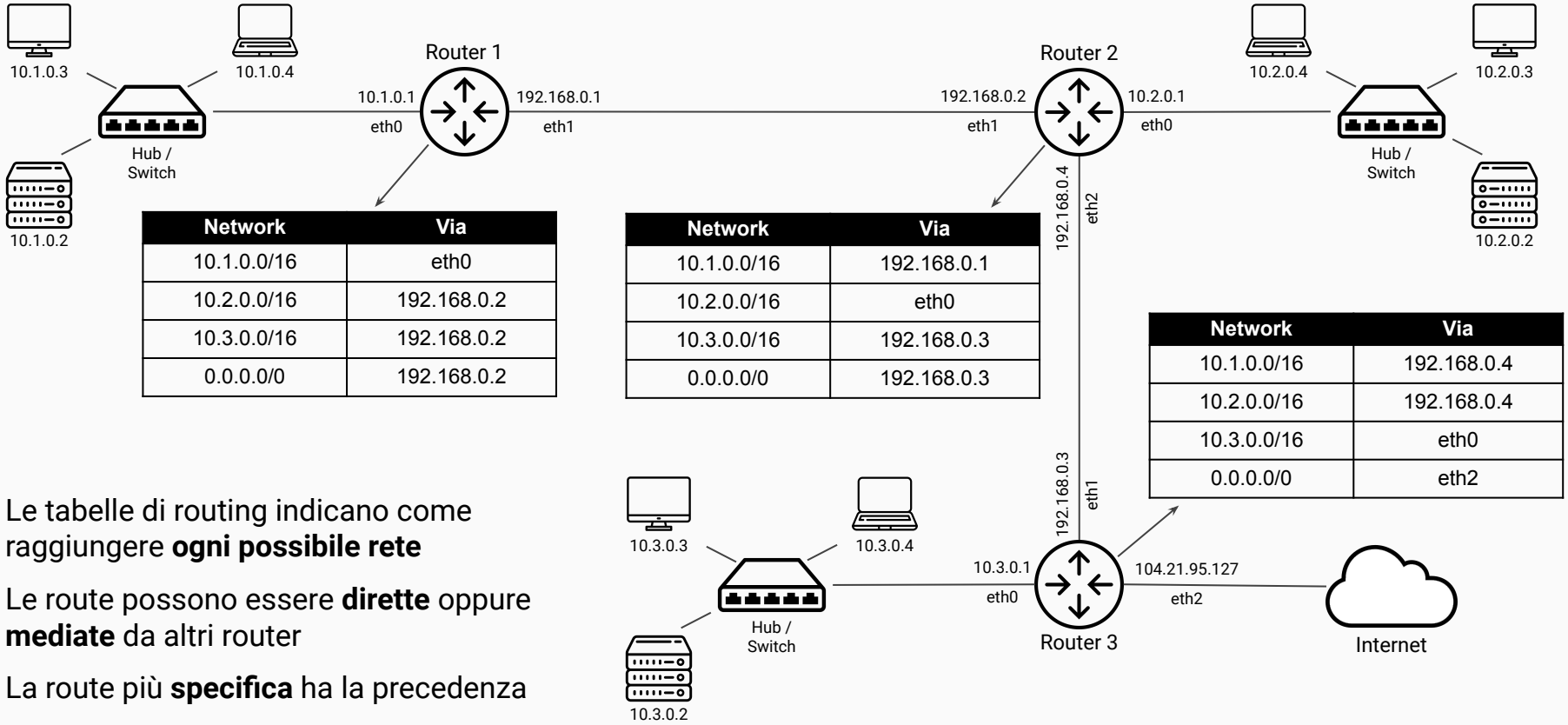
Livello 3: rete





1) **Versione** del protocollo IP
2) **Dimensione** del pacchetto IP / del payload
3) Numero di **hop rimanenti**
4) **Protocollo** dei dati incapsulati payload
5) Codice per la verifica degli **errori**
6) Indirizzo di rete del **mittente** del pacchetto
7) Indirizzo fisico del **destinatario** del pacchetto

Routing



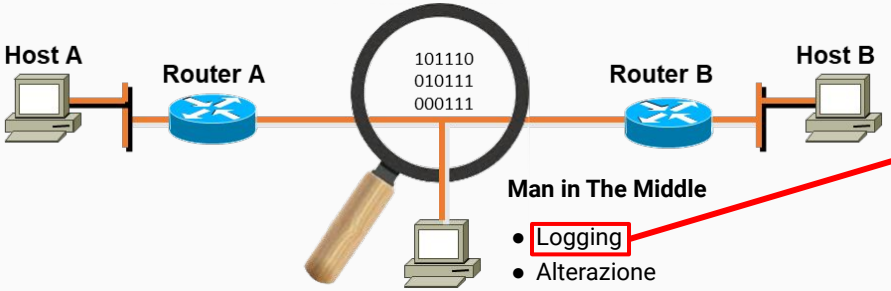
Network	Via
10.1.0.0/16	eth0
10.2.0.0/16	192.168.0.2
10.3.0.0/16	192.168.0.2
0.0.0.0/0	192.168.0.2

Network	Via
10.1.0.0/16	192.168.0.1
10.2.0.0/16	eth0
10.3.0.0/16	192.168.0.3
0.0.0.0/0	192.168.0.3

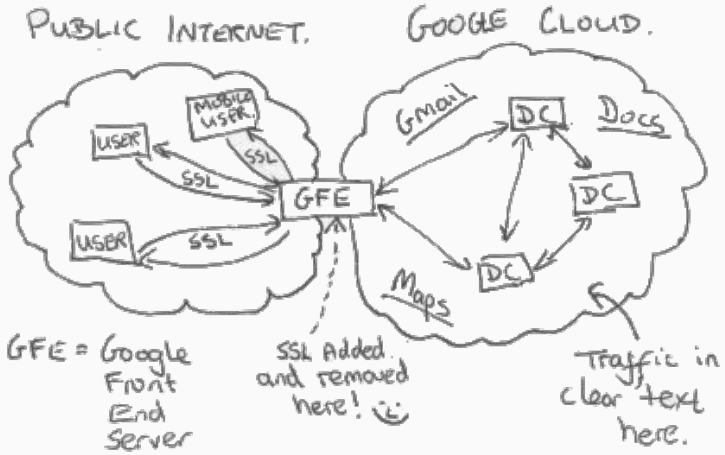
Network	Via
10.1.0.0/16	192.168.0.4
10.2.0.0/16	192.168.0.4
10.3.0.0/16	eth0
0.0.0.0/0	eth2

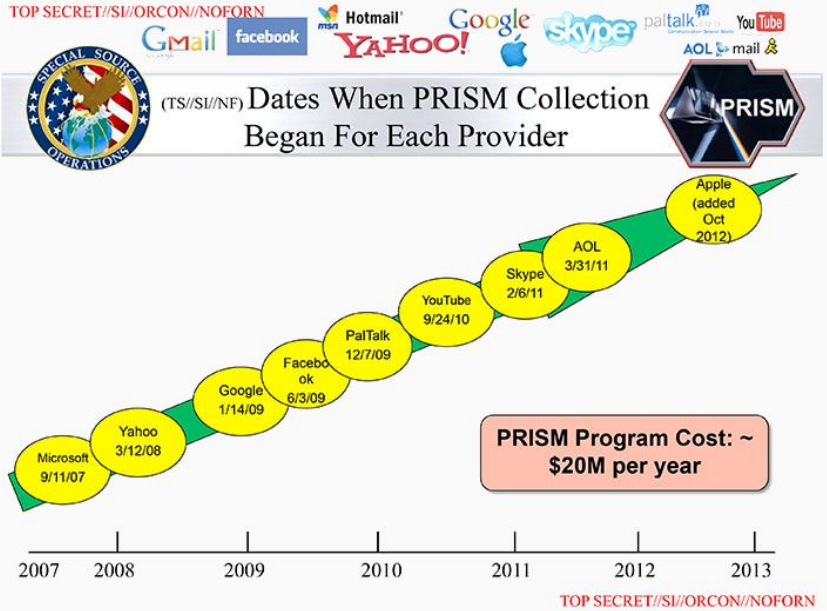
- Le tabelle di routing indicano come raggiungere **ogni possibile rete**
- Le route possono essere **dirette** oppure **mediate** da altri router
- La route più **specifica** ha la precedenza

Sniffing



- Logging
- Alterazione
- Interruzione
- Ripetizione





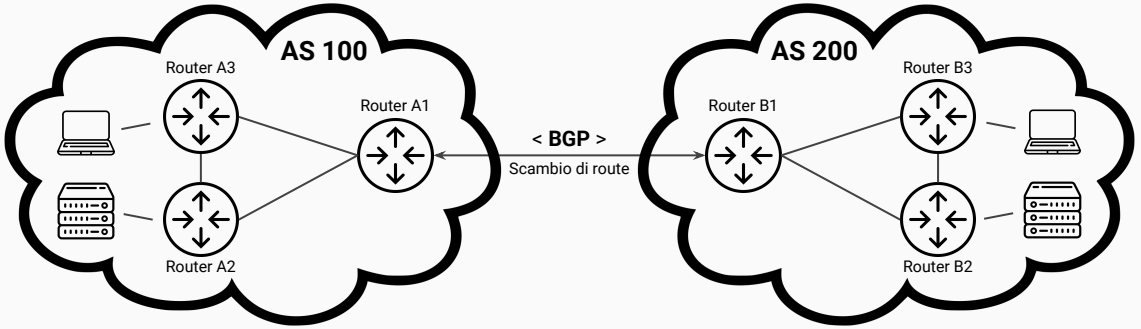
Progetto **PRISM** (National Security Agency)

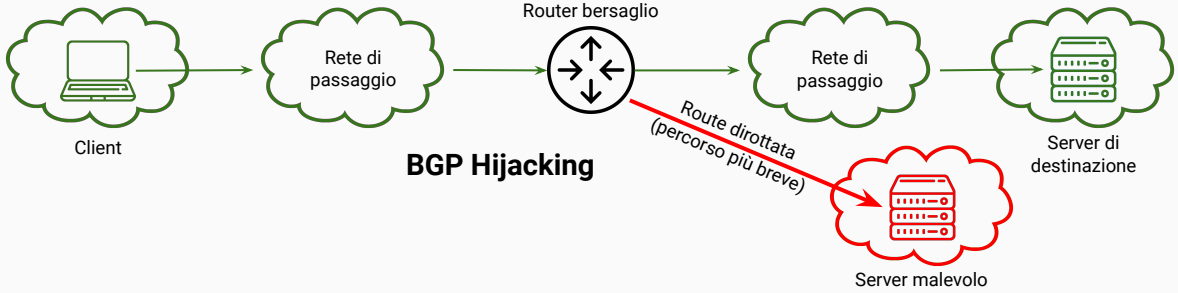
2007 - 2013

Funzionamento di PRISM

- **Traffico** tra i datacenter **non cifrato**
- **TAP** sui **cavi** in fibra ottica usati per l'interconnessione

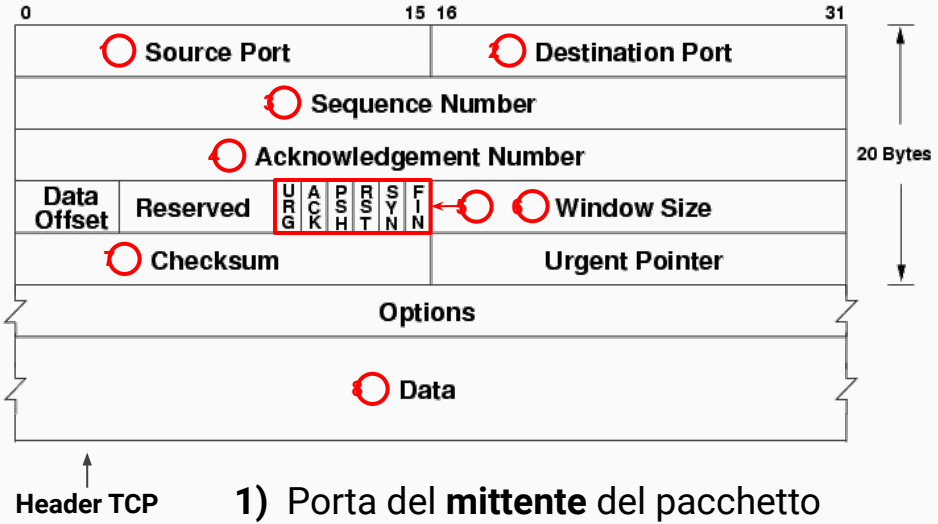
BGP: routing su scala globale





- I border gateway dei **sistemi autonomi** (AS) si scambiano informazioni sulle **route** conosciute attraverso appositi protocolli (come **BGP**)
- Le informazioni sulle route vengono **propagate** in maniera ricorsiva agli AS confinanti
- In questo modo ogni border gateway di ogni singola rete connessa ad Internet può conoscere le **rotte** e la **distanza** per raggiungere qualunque altra rete
- Protocolli come BGP hanno come prerequisito il ritenere **affidabili** le **informazioni** ricevute
- Se le informazioni non sono **corrette** o sono volutamente **alterate**, il traffico può prendere direzioni non previste o non volute

Livello 4: trasporto

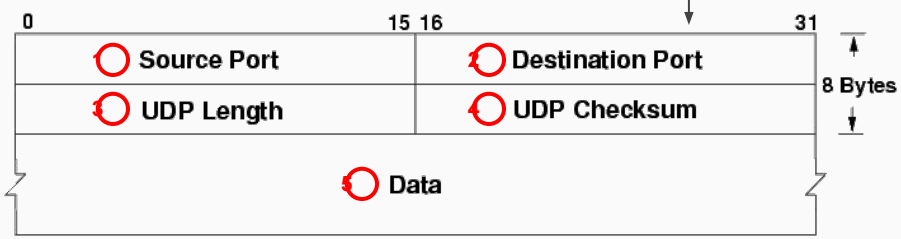


Header TCP

1) Porta del **mittente** del pacchetto
2) Porta del **destinatario** del pacchetto
3) **Dimensione** del pacchetto
4) Codice per la verifica degli **errori**
5) **Contenuto** del pacchetto

1) Porta del **mittente** del pacchetto
2) Porta del **destinatario** del pacchetto
3) Numero di **sequenza** del pacchetto
4) Ultimo numero di **sequenza ricevuto**
5) **Flag** del pacchetto
6) Buffer di ricezione **disponibile**
7) Codice per la verifica degli **errori**
8) **Contenuto** del pacchetto

Header UDP



Livelli 5 e 6: sessione e presentazione

Livelli ibridi, spesso **accorpati** tra di loro o con il livello 7 (applicativo)

Forniscono varie **funzioni**, tra cui:

- autenticazione
- autorizzazione
- codifica
- compressione
- crittografia

Sono generalmente gestiti da specifiche applicazioni

Alcuni esempi:

- **SOCKS**: protocollo utilizzato dall'omonima categoria di server proxy, serve a stabilire una connessione con una terza parte remota (indicata all'instaurazione della connessione)
- **TLS**: protocollo crittografico utilizzato per fornire caratteristiche quali confidenzialità, integrità dei dati ed autenticazione a servizi di livello applicativo
- **PGP**: suite di protocolli utilizzata per firmare o crittografare dati quali testi, file o email

Livello 7: applicativo

Livello dei **servizi** “finali”, il più evidente per gli utenti

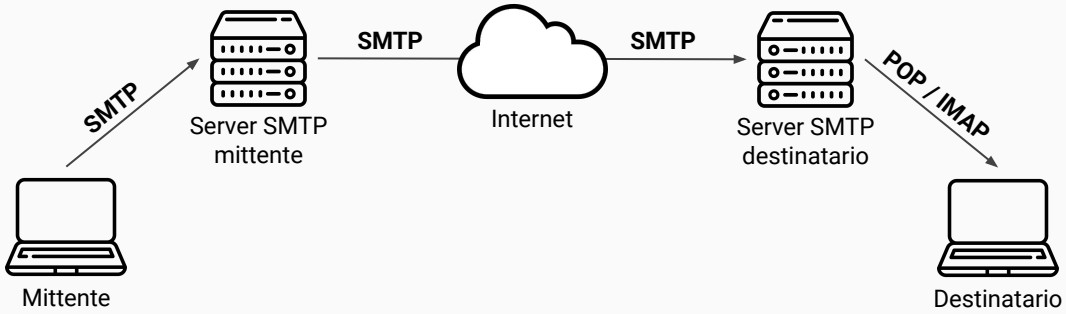
Comprende protocolli per le **funzionalità** più varie:

- trasferimento di file testuali e binari
- chat e VOIP
- streaming audio e video
- gaming
- accesso remoto a server
- trasferimento P2P

Alcuni esempi:

- **HTTP**: già menzionato in precedenza
- **FTP**, **TFTP**, **SFTP** e **SCP**: protocolli utilizzati per il trasferimento di file client-server
- **DHCP**: protocollo utilizzato per l’assegnazione automatica di indirizzi IP
- **DNS**: protocollo utilizzato per tradurre nomi simbolici in indirizzi di rete
- **SSH**, **RDP** e **VNC**: protocolli utilizzati per la connessione a host remoti
- **SMTP**, **POP** e **IMAP**: protocolli utilizzati per la posta elettronica

SMTP: Simple Mail Transfer Protocol



- Protocollo **testuale**
- Interazione di tipo **comando** - **risposta**
- **Contenuto** del messaggio nello stesso stile di HTTP

(prima gli **header**, separati da CR LF, poi il **corpo**)

- Tutti i **contenuti** sono codificati sotto forma di **testo**

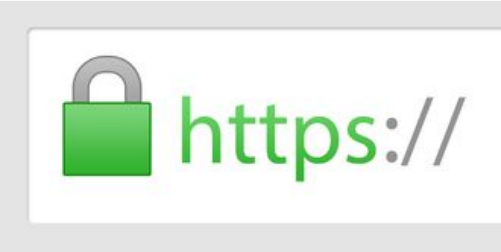
(anche le **immagini** e i file **binari** di qualunque genere)

```
S: 220 smtp2go.com ESMTP Exim
C: HELO mydomain.com
S: 250 Hello mydomain.com
C: MAIL FROM:<sender@mydomain.com>
S: 250 Ok
C: RCPT TO:<recipient@anotherdomain.com>
S: 250 Accepted
C: DATA
S: 354 Enter message, ending with "."
C: Subject: sample message
C: From: sender@mydomain.com
C: To: recipient@anotherdomain.com
C:
C: Greetings,
C: Typed message (content)
C: Goodbye.
C: .
S: 250 OK
C: QUIT
S: 221 www.sample.com closing connection
```

TLS/SSL

Transport Layer Security (TLS) e il suo predecessore **Secure Sockets Layer** (SSL) sono dei protocolli crittografici che permettono una comunicazione sicura dalla sorgente al destinatario (**end-to-end**) su reti TCP/IP fornendo *autenticazione*, *integrità dei dati* e *confidenzialità*.
Essi operano al di sopra del livello di trasporto.

L'esempio più comune di TLS/SSL è nel protocollo HTTPS, che consiste nella comunicazione tramite HTTP all'interno di una connessione criptata via TLS o SSL.

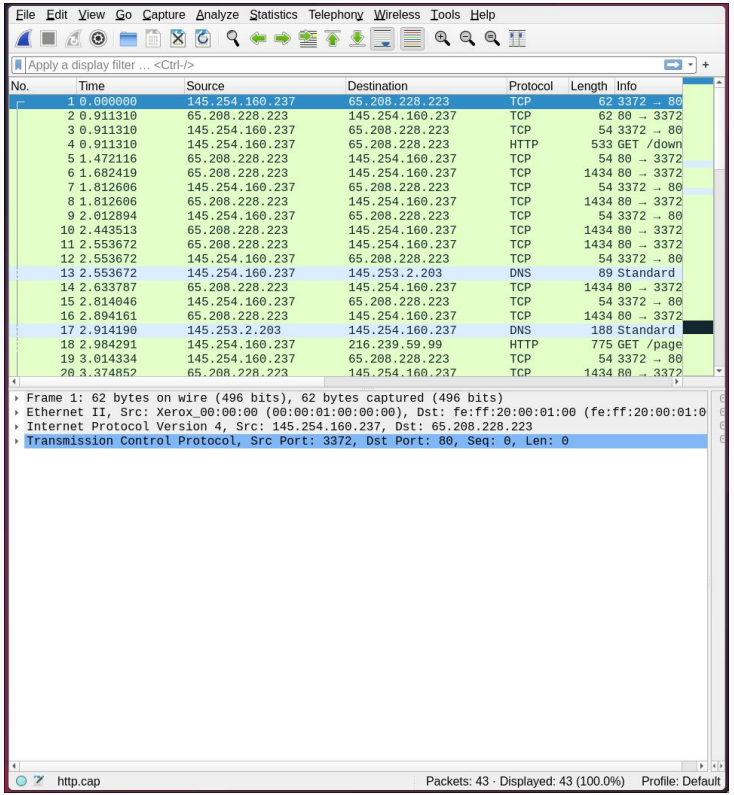


Ci sono **3 fasi principali:**

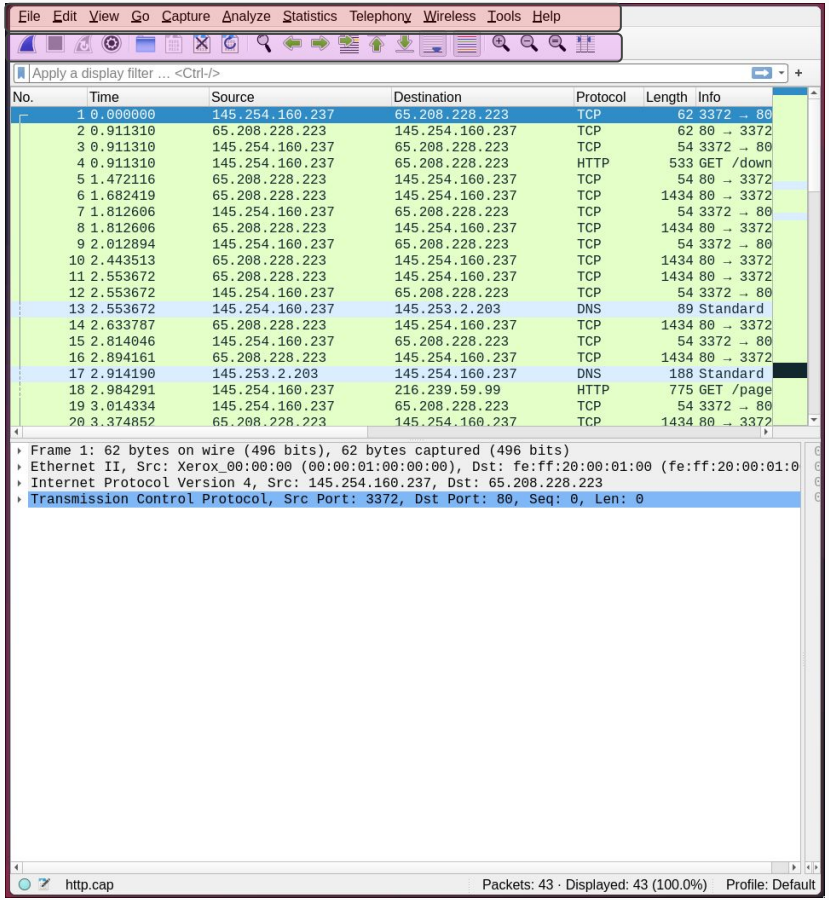
- **Negoziazione** fra le parti dell'algoritmo da utilizzare
- Scambio delle **chiavi** e autenticazione
- **Cifratura** simmetrica e autenticazione dei messaggi

Wireshark

- Software di analisi dei pacchetti di rete open source.
- Permette di catturare (in gergo “sniffare”) e analizzare i dati di rete in tempo reale o da file di cattura esistenti (pcap).
- Supporta una vasta gamma di protocolli di rete e può essere utilizzato per diagnosticare problemi di rete, individuare falle di sicurezza e analizzare il traffico.



Tour UI Wireshark

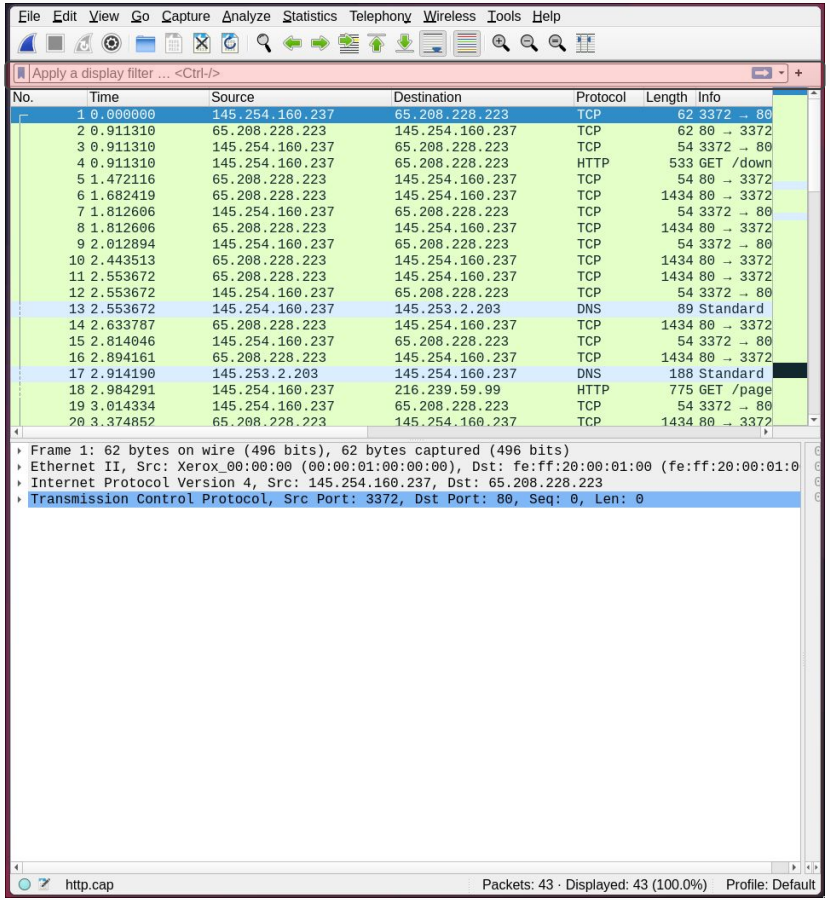


Il **menu** ci offre diverse funzionalità, tra le quali:

- **File**: aprire, salvare, ed esportare i file di cattura
- **Edit**: trovare pacchetti e marchiarli
- **View**: controlla le impostazioni del visualizzatore di pacchetti (permette di modificare il layout e il colore dei pacchetti)
- **Go**: saltare al pacchetto specificando il numero
- **Analyze**: seguire uno stream e modificare filtri dei pacchetti

La **toolbar** mostra alcune delle voci più utilizzate del menu

Tour UI Wireshark

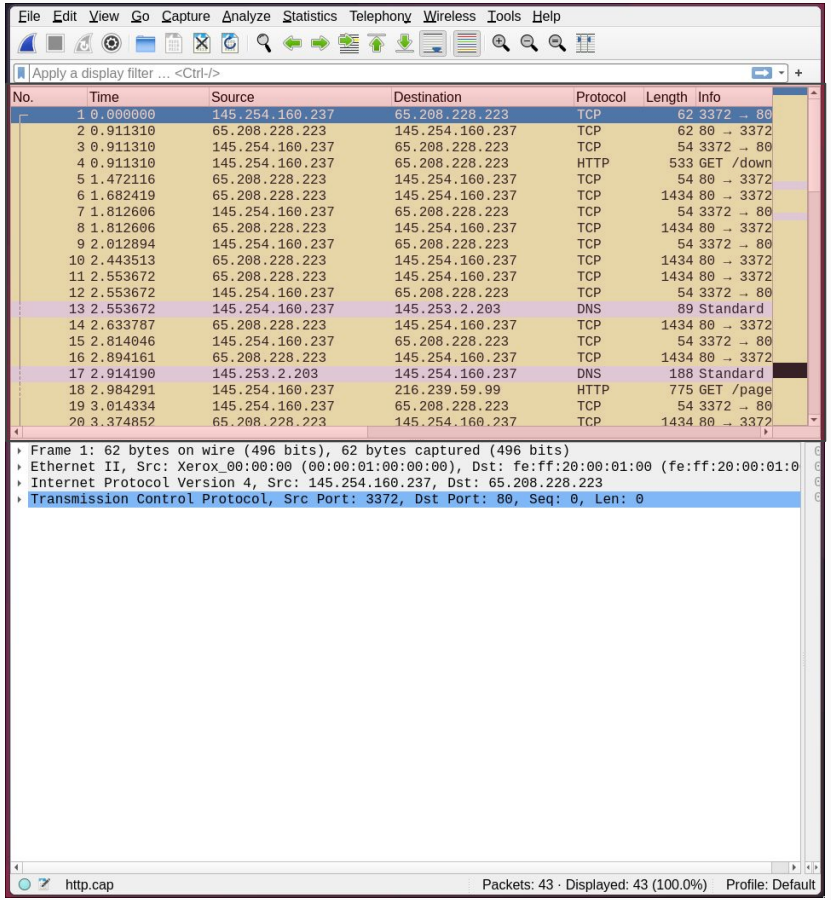


La **toolbar dei filtri** ci permette di applicare un filtro ai pacchetti, selezionandolo tra i filtri salvati, oppure scrivendone uno nuovo.

I filtri sono una delle funzioni più potenti di wireshark, in quanto ci permettono di analizzare solo i pacchetti di nostro interesse, ed eliminare il fastidioso “rumore di fondo”.

A breve faremo un tour sulla sintassi dei filtri, e vedremo i filtri più comuni.

Tour UI Wireshark

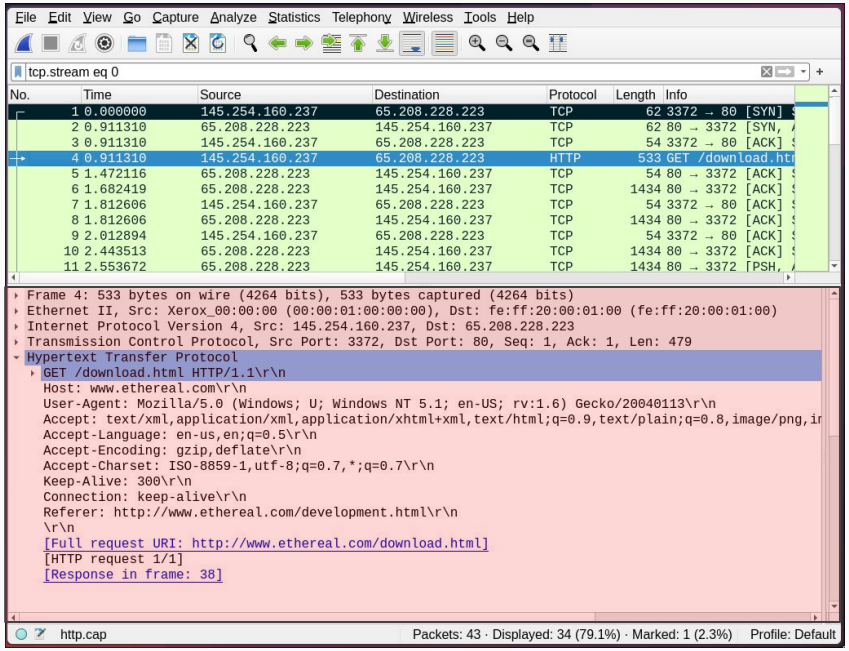


La **lista dei pacchetti** mostra tutti i pacchetti che rispettano il filtro specificato.

Di default mostrerà il numero del pacchetto, il tempo in secondi in cui quel pacchetto è stato catturato, il source e destination address, il protocollo utilizzato, la lunghezza del pacchetto, ed infine alcune informazioni.

E’ possibile cliccare sull’intestazione delle colonne per ordinare a seconda del valore scelto.

Tour UI Wireshark

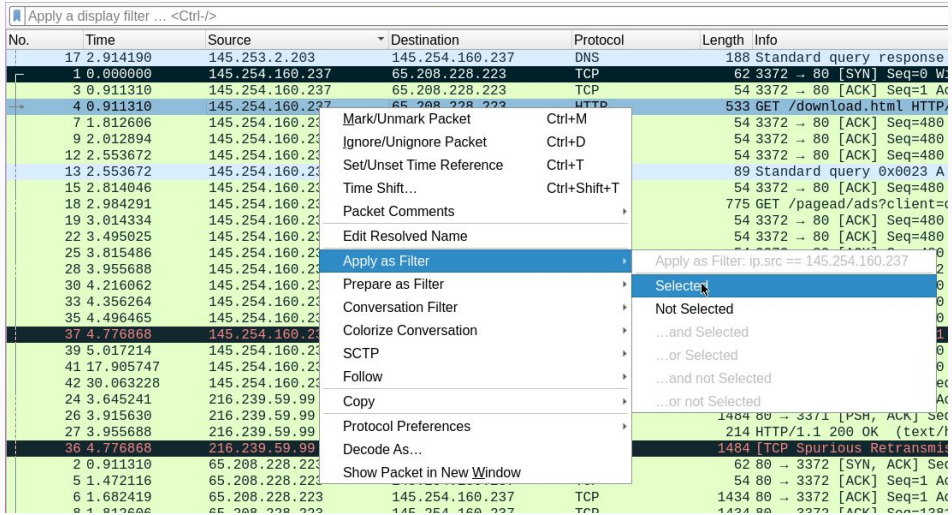


Il **visualizzatore dei dettagli del pacchetto** mostra la dissezione del pacchetto.

È divisa in base ai vari livelli e permette di analizzare ciascuno di essi.

Se, ad esempio, ci interessa analizzare gli Header di un pacchetto HTTP, basterà cliccare su Hypertext Transfer Protocol per analizzare nel dettaglio il pacchetto.

Filtri in Wireshark



I **filtri** semplificano notevolmente il lavoro di analisi.

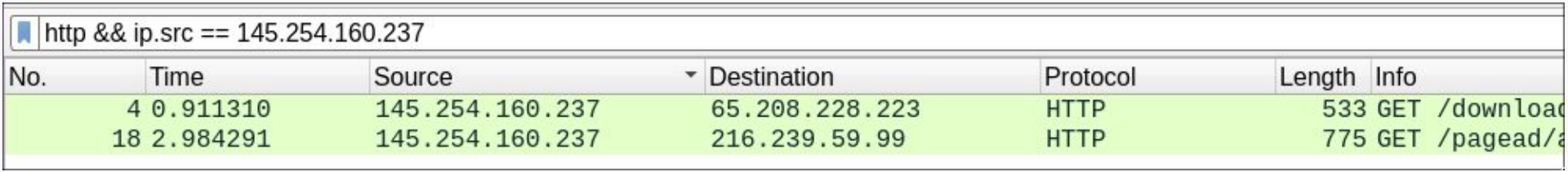
Possiamo filtrare per un determinato valore, facendo tasto destro -> *Apply as Filter*.

Selected farà in modo che solo i pacchetti con quel valore saranno visualizzati.

Not Selected eliminerà dalla visualizzazione i pacchetti con quel valore.

È possibile scrivere filtri più complessi utilizzando la toolbar dei filtri

Esempio: vogliamo visualizzare i pacchetti che contengono HTTP, da parte dell'ip 145.254.160.237



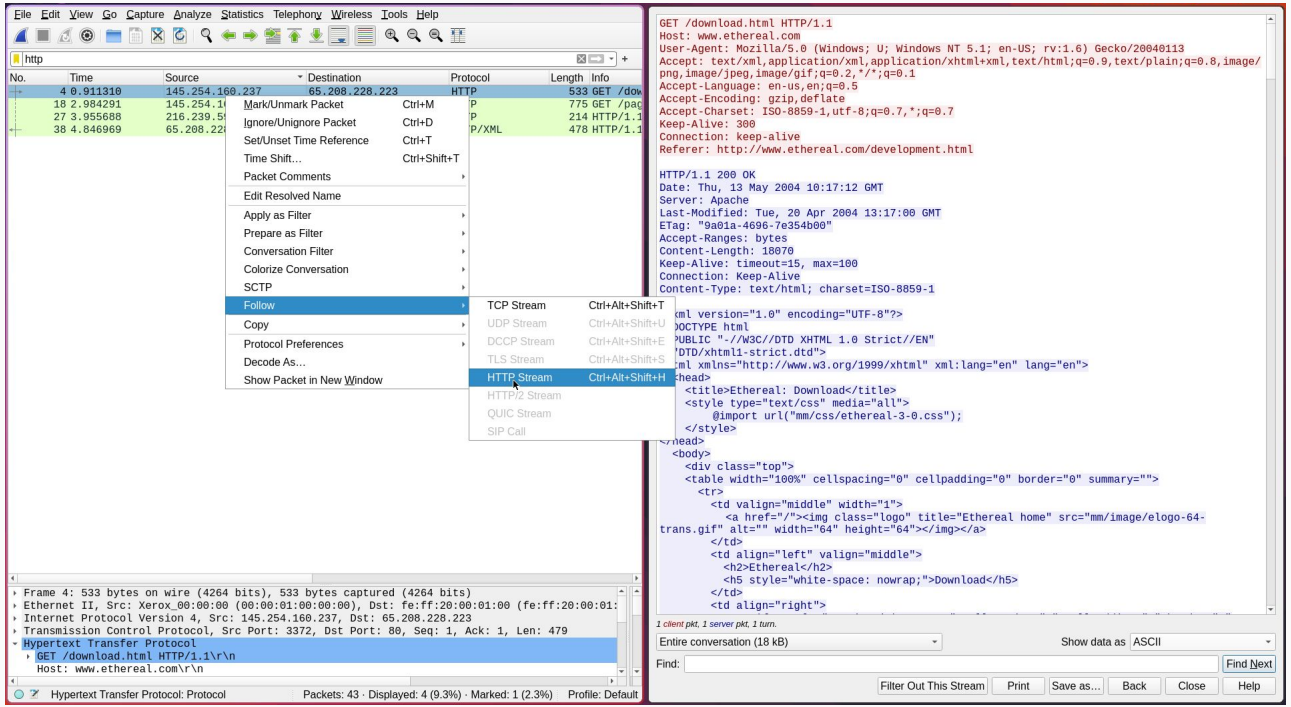
Cheat Sheet Filtri in Wireshark

SINTASSI OPERATORE	OPERATORE	DESCRIZIONE o ESEMPIO
and / **&&**	AND logico	Tutte le condizioni devono essere vere
or / **||**	OR logico	Almeno una delle condizioni deve essere vera
xor / **^^**	XOR logico	Solo una delle condizioni deve essere vera
not / !	negazione	Nega l'espressione che lo segue. **Esempio**: ***!(smtp)*** visualizzerà tutti i pacchetti tranne quelli SMTP
contains	contiene	Il protocollo o valore, deve contenere la stringa o il valore che segue. **Esempio**: ***http contains "ciao"*** visualizzerà tutti i pacchetti HTTP che contengono la stringa "ciao"

Cheat Sheet Filtri in Wireshark

SINTASSI OPERATORE	OPERATORE	ESEMPIO
eq / ==	uguale	ip.dest == 192.168.1.1
ne / !=	diverso	ip.dest != 192.168.1.1
gt / >	maggiore	frame.len > 10
lt / <	minore	frame.len < 10
ge / >=	maggiore o uguale	frame.len >= 10
le / <=	minore o uguale	frame.len <= 10

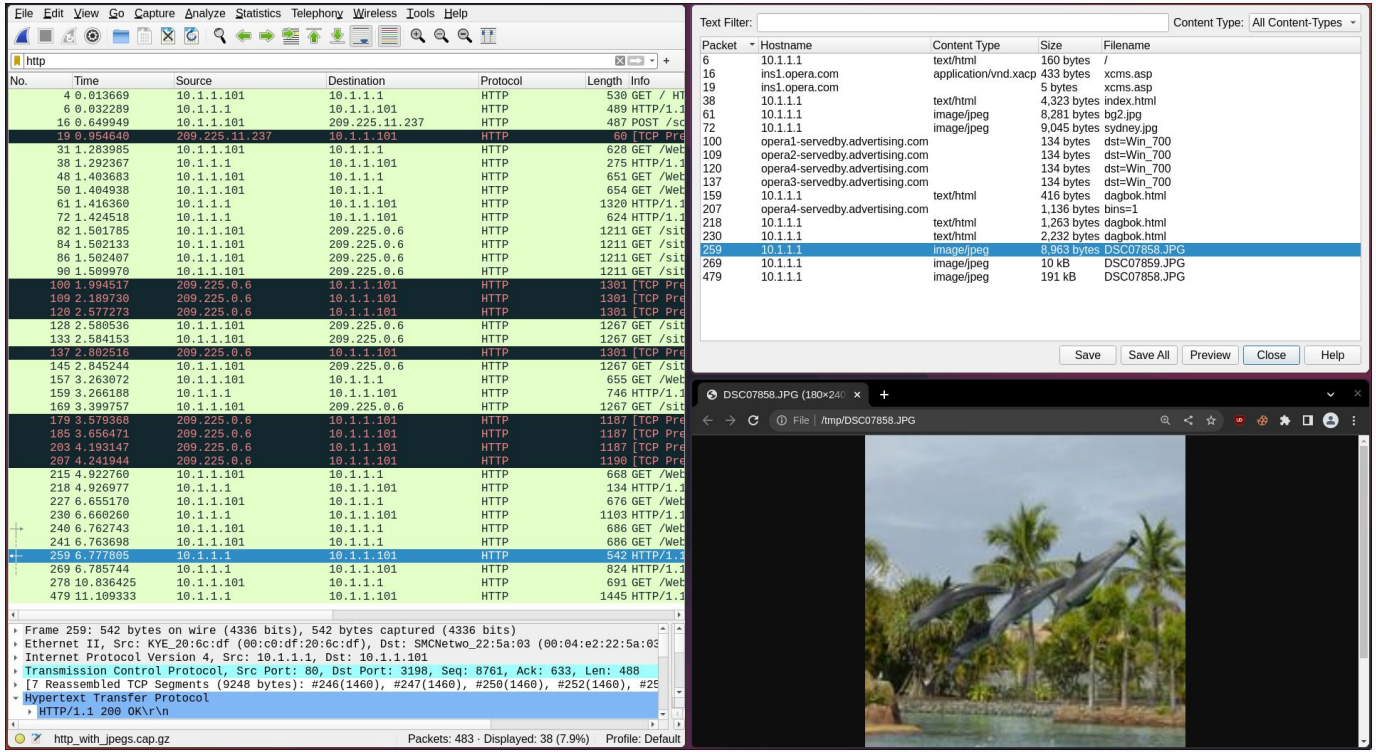
Seguire flussi in Wireshark



È possibile seguire i flussi di rete, facendo tasto destro sul pacchetto interessato, e cliccando su **Follow**.

È una funzione essenziale per poter visualizzare con facilità, in maniera ordinata, le query di un client con le relative risposte dal server.

Seguire flussi in Wireshark



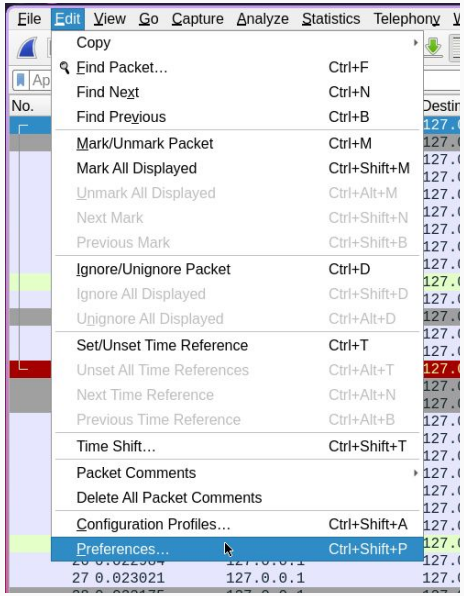
Cliccando su

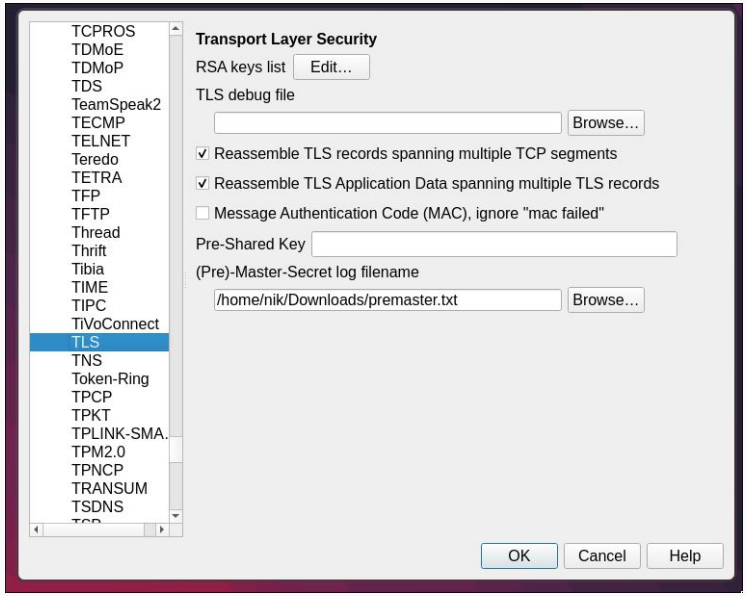
File -> Export Objects

e poi selezionando il protocollo da cui vogliamo estrarre oggetti, è possibile ottenere una lista di tutti gli oggetti presenti nei pacchetti.

Possiamo ad esempio salvare immagini scaricate via HTTP o vederne un' anteprima.

Decifrare TLS con il file premaster keylog





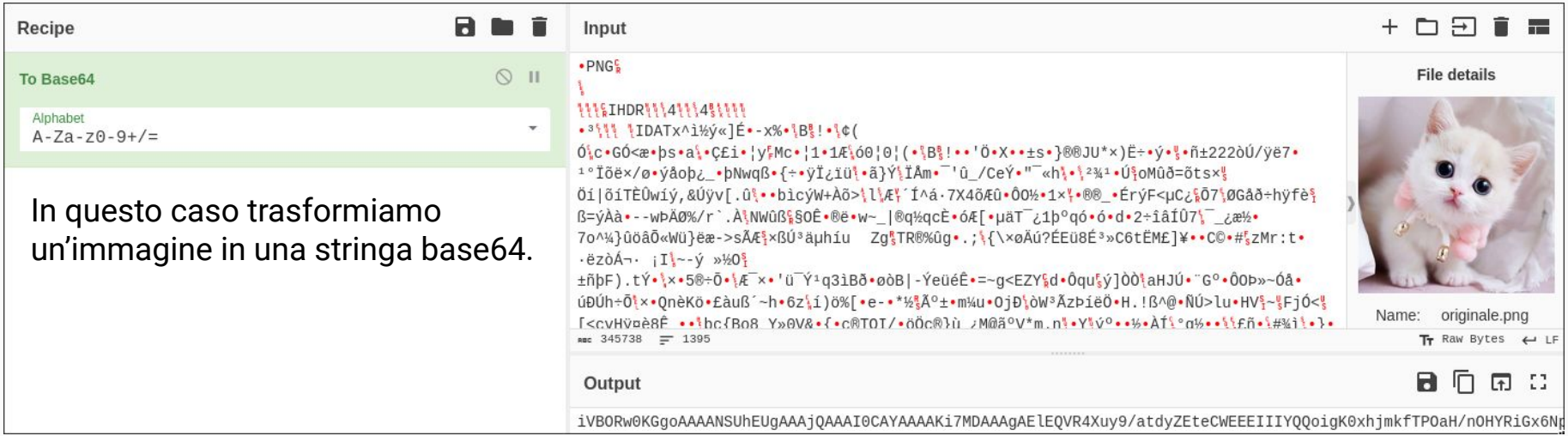
Cliccando su

Edit -> Preferences

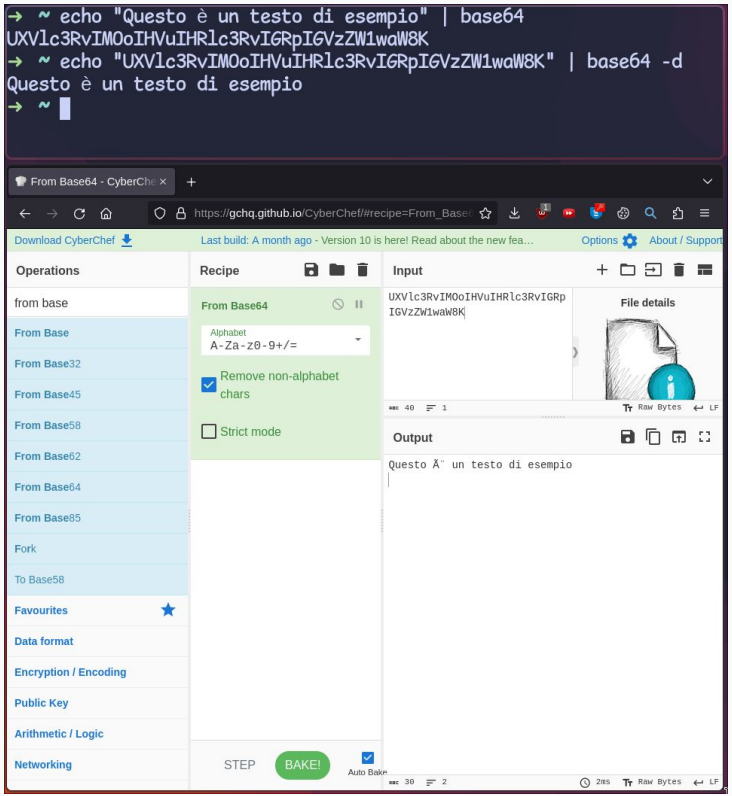
e poi selezionando **TLS** nella sezione *Protocols*, è possibile aggiungere la premaster key, utile a decifrare il traffico TLS.

Codifica dati in base64

Base64 è un sistema di codifica utilizzato per rappresentare informazioni binarie in formato testuale. I dati vengono convertiti in una sequenza di caratteri **ASCII**, in modo che possano essere trasferiti su reti in cui non è possibile trasmettere direttamente dei dati binari (ad esempio in email o HTTP).



Codifica dati in base64



Per codificare dati in base64, possiamo utilizzare un tool come **CyberChef**, con la funzione *To base64*, oppure possiamo scrivere i dati da codificare nello standard input dell'eseguibile Unix **base64**.

Per decodificare, invece, possiamo usare la funzione *From Base64* di CyberChef, oppure l'opzione *-d* in base64 delle GNU coreutils.

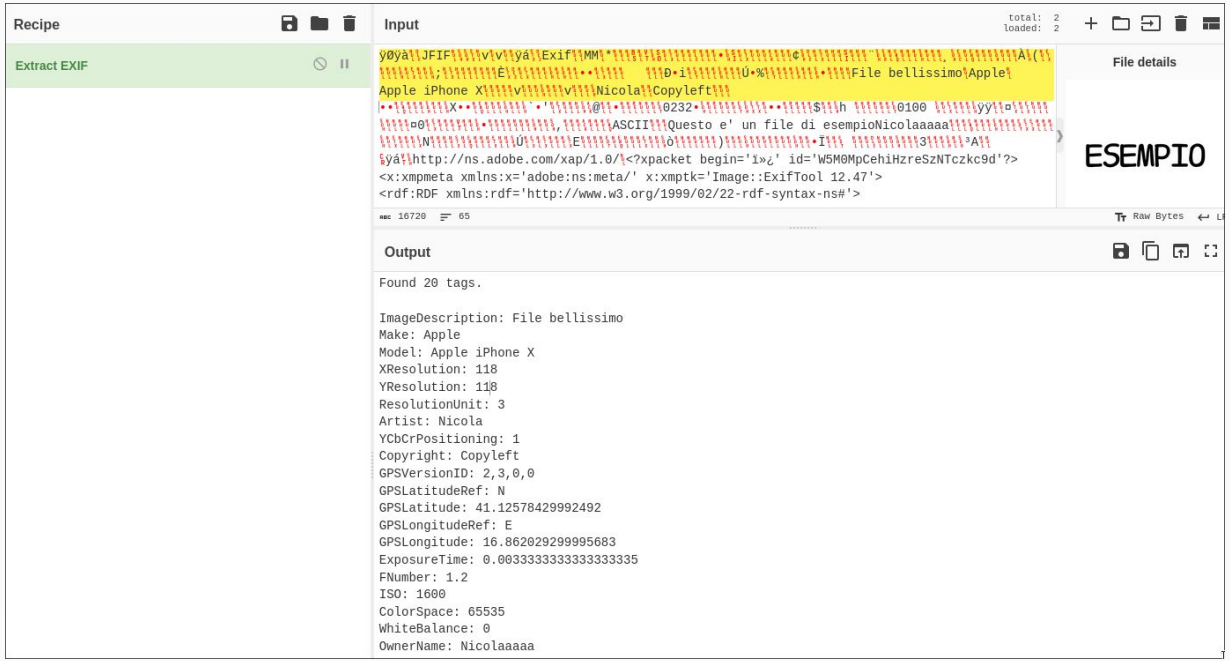
Metadati

- Presenti spesso nei file multimediali (es. foto, video e audio)
- Forniscono informazioni aggiuntive (data di creazione, codifica del file, lunghezza del video, dispositivo utilizzato per crearlo, coordinate GPS ecc…)

Exif (Exchangeable image file format)

- Aggiunge specifiche etichette (tag) di metadati, a formati esistenti tra i quali: JPEG, TIFF, RIFF e PNG.

```
$ exiftool -exif:All esempio.jpg
Image Description               : File bellissimo
Make                            : Apple
Camera Model Name               : Apple iPhone X
X Resolution                    : 118
Y Resolution                    : 118
Resolution Unit                 : cm
Artist                          : Nicola
Y Cb Cr Positioning             : Centered
Copyright                       : Copyleft
Exposure Time                   : 1/300
F Number                        : 1.2
ISO                             : 1600
Exif Version                    : 0232
Components Configuration        : Y, Cb, Cr, -
User Comment                    : Questo e' un file di
esempio
Flashpix Version                : 0100
Color Space                     : Uncalibrated
White Balance                   : Auto
Owner Name                      : Nicolaaaaa
GPS Version ID                  : 2.3.0.0
GPS Latitude Ref                : North
GPS Longitude Ref               : East
```



LSB (Lowest Significant Bit)

- Tecnica di steganografia utilizzata su immagini e file audio, che nasconde segreti modificando i dati del file
- Queste modifiche devono risultare impercettibili all’occhio o orecchio umano
- Come si nota dall'esempio, le due immagini hanno le stesse dimensioni, ma quella modificata nasconde un segreto…

originale.png

Filesize: 345739 byte

modificata.png

Filesize: 345739 byte

Vediamo più da vicino...

originale.png

modificata.png

Molto più da vicino…

- Un’immagine non è nient’altro che una matrice di pixel… (ogni pixel è rappresentato da 3 colori, R, B e G)
- Analizziamo nel dettaglio i primi pixel delle due immagini

originale.png

Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
202	200	210	202	200	210	204	202	212

modificata.png

Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
202	201	210	202	201	210	204	203	212

- Notiamo come nell’immagine modificata, alcuni colori di alcuni pixel variano di 1…
- Questo perché l’algoritmo di steganografia ha modificato il bit meno significativo del colore.
- Notiamo come quando l’ultimo bit e’ 0, il numero sara’ pari, mentre quando e’ 1, sara’ dispari

254

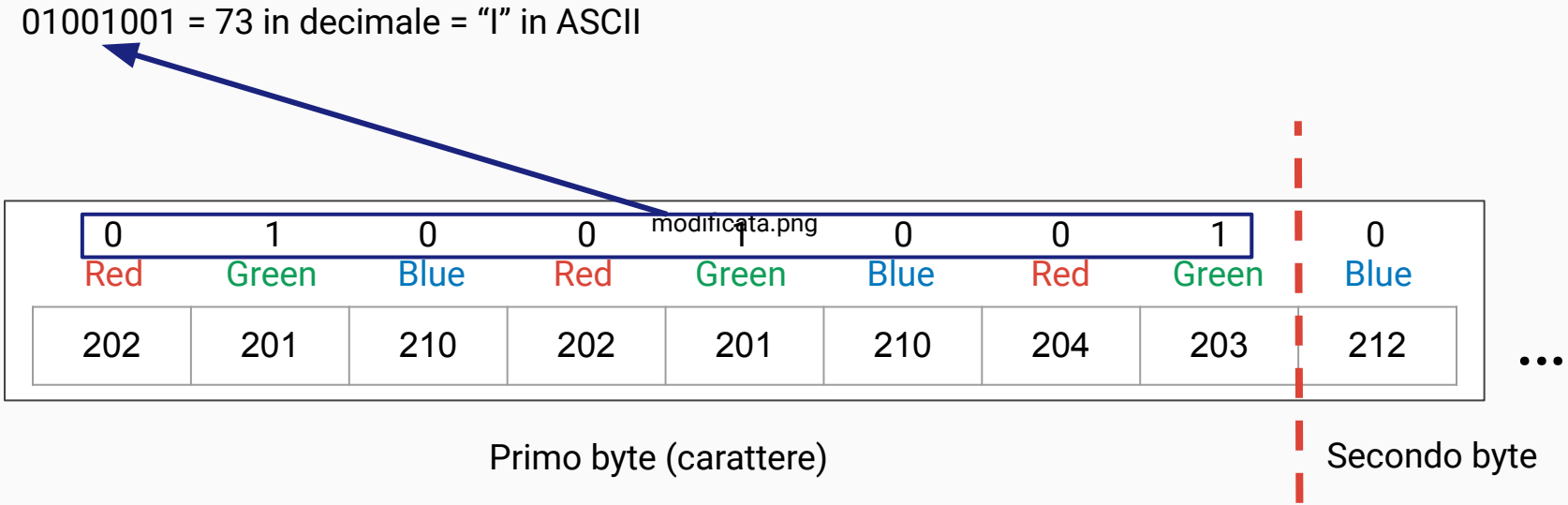
1	1	1	1	1	1	1	0

255

1	1	1	1	1	1	1	1

Molto più da vicino…

- Quindi ora ricaviamo i bit “nascosi” in questi pixel…



Molto più da vicino...

Recipe

Extract LSB

Colour Pattern #1	Colour Pattern #2	Colour Pattern #3	Colour Pattern #4
R	G	B	

Pixel Order	Bit
Row	0

Input

File details

Name: modificata.png
Size: 345,739 bytes
Type: image/png
Loaded: 100%

345739 1452 Raw Bytes LF

Output

I gatti sono molto carini(e altre informazioni segretissime)

Tools!

- steghide:
  - Permette di nascondere e decodificare dati in immagini e audio con algoritmi più complessi e più sicuri (con uso di crittografia)

- stegoveritas:
  - Tool “tutto in uno”, per scovare segreti nascosti in immagini, usando gli algoritmi più comuni

- Pillow:
  - Libreria Python che permette di manipolare immagini con facilità