Wireshark

Laboratorio di Reti AA 2023/2024

Ing. Filippo Poltronieri
filippo.poltronieri@unife.it

tcpdump

tcpdump è stato uno dei primi tool che consente di effettuare una analisi (dump) del traffico di rete. tcpdump è installato di default su parte delle distribuzioni Linux ed è disponibile in tutto il mondo UNIX.

tcpdump fornisce il supporto al filtraggio di pacchetti e consente di visualizzare informazioni specifiche di una vasta gamma di protocolli. Il manuale del comando ci fornisce una guida dettagliata ai vari parametri e al suo utilizzo.

Quindi risulta essere un tool veramente utile per il controllo del traffico, specialmente quando non abbiamo la possibilità di installare tool differenti.

Una valida evoluzione di tcpdump è rappresentata dal noto e sempre più utilizzato Wireshark.

Wireshark

Wireshark è il più famoso e usato network protocol analyzer esistente al mondo.

Un network protocol analyzer è un tool che cattura pacchetti di rete e prova a visualizzare il maggior numero di dettagli su di essi. Questo è utile per:

- Troubleshooting per la configurazione di apparati di rete
- Analisi di problemi di sicurezza
- Debugging di applicazioni distribuite
- Comprendere il funzionamento dei protocolli di rete
- Applicazioni per cybersecurity

Wireshark è disponibile per Linux, Windows, e Mac OS X, sia nella versione GUI che nella versione command-line (tshark).

http://www.wireshark.org/

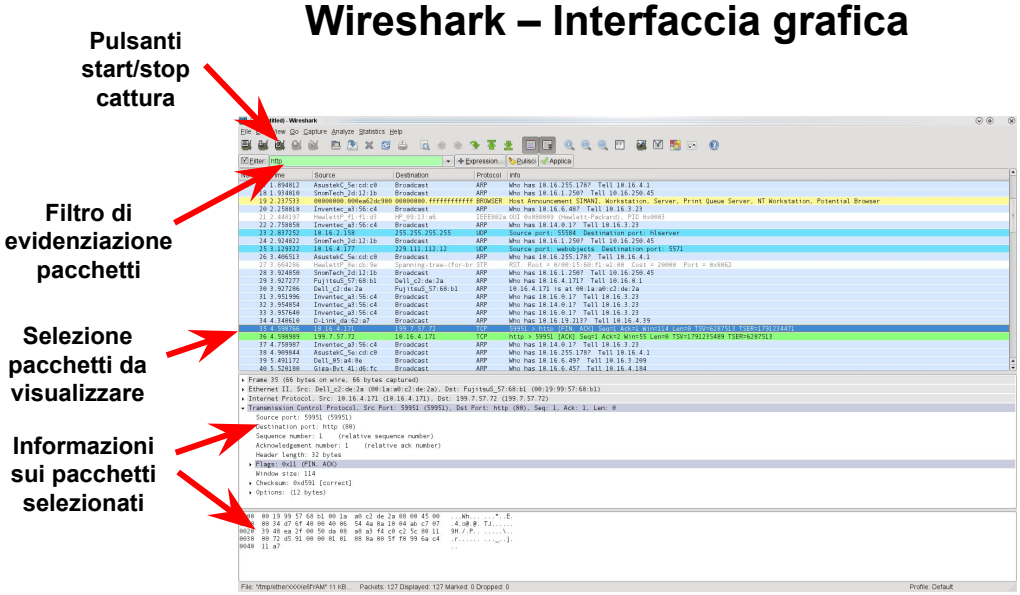
Wireshark – Funzioni supportate

Wireshark permette di:

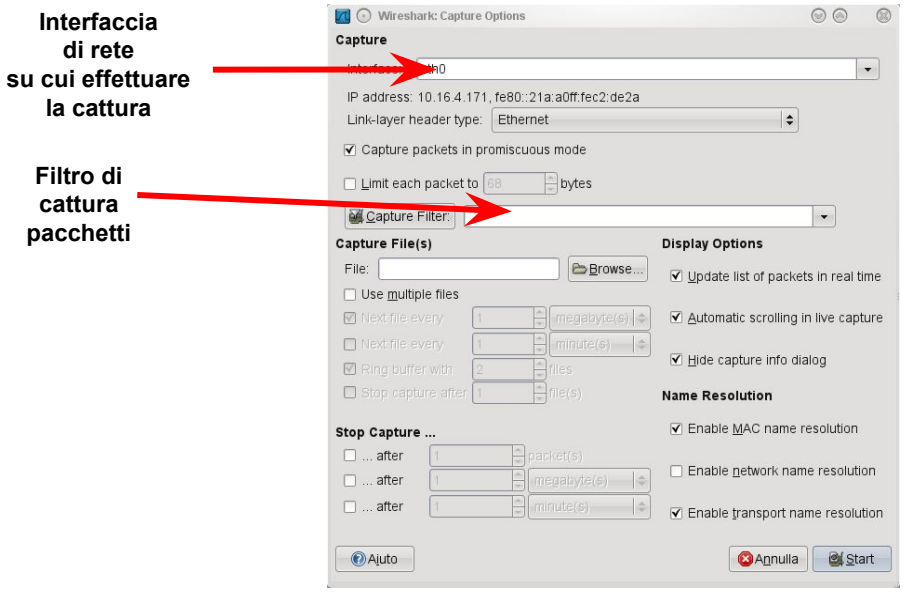
- Effettuare sessioni *live* di cattura dei pacchetti di rete
- Filtrare i pacchetti da catturare secondo determinate caratteristiche (es. solo pacchetti HTTP, solo pacchetti che provengono dalla porta 345, ecc.)
- Visualizzare informazioni dettagliate sui pacchetti, a seconda del rispettivo formato (Wireshark supporta migliaia di formati di pacchetto diversi)
- Salvare e successivamente ricaricare i dati di una sessione di cattura
- Cercare all'interno di una sessione i pacchetti che hanno specifiche caratteristiche
- Calcolare statistiche sui pacchetti catturati
- Importare/esportare dati da/verso altri software

Tutte le funzioni sono accessibili sia tramite una comoda interfaccia grafica (comando *wireshark*) che da un'interfaccia a riga di comando (comando *tshark*).
Le funzioni di Wireshark possono essere estese tramite script in linguaggio Lua.

Wireshark – Interfaccia grafica



Wireshark – Dialogo opzioni cattura



Wireshark – Filtraggio cattura pacchetti

Wireshark permette di filtrare i pacchetti da catturare secondo una sintassi derivata dallo standard BPF (Berkeley Packet Filter). Le due regole più usate (e utili) solo:

- `[src|dst] host <nome host o indirizzo IP>`
  Cattura solo i pacchetti inviati a o provenienti dall'host specificato.
- `[tcp|udp] [src|dst] port <nome servizio o numero porta>`
  Cattura solo i pacchetti inviati a o provenienti dalla porta specificata.

È possibile combinare le opzioni tra loro tramite gli operatori logici “and”, “or” e “not”.
Per ulteriori informazioni: http://wiki.wireshark.org/CaptureFilters (man pcap-filter)

Wireshark – Filtraggio cattura pacchetti

Esempi:

- `tcp port http` => solo traffico HTTP
- `port not 53 and not arp` => no traffico DNS o ARP
- `host www.example.com and not (port 80 or port 25)`
- `net 192.168.0.0/24` => cattura il traffico (to/from) di una sottorete
- `src net 192.168.0.0/24` => solo il traffico con sorgente in quella reta / destinazione con dst
- `ip` => catturare il solo traffico IPv4

Ci sono svariate opzioni che possono essere controllate, oltre che dal manuale online, anche dal manuale.

Wireshark – Filtraggio display pacchetti

All'interno del traffico catturato, Wireshark permette anche di evidenziare un particolare tipo di pacchetti (utilizzando i *display filter)*, in modo che essi siano più facili da individuare nel contesto di una lunga sessione di cattura.

I display filter supportano regole molto più complesse rispetto ai filtri di cattura (supporto regex, ecc.).
Esistono display filter per più di 1300 protocolli, con più di 110000 comandi.
Si vedano:
http://wiki.wireshark.org/DisplayFilters
http://www.wireshark.org/docs/dfref/
Esempi:
`tcp.port eq 25 or icmp` => evidenzia traffico SMTP o ICMP
`http.request.uri matches "gl=se$"` => pattern matching su URL HTTP
`ip.src == 10.43.54.65 or ip.dst == 10.43.54.65`

Wireshark – Filtraggio display pacchetti

La textbox di Display filter ci semplifica la scrittura attraverso una funzione di autocompletamento. La sintassi è differente rispetto alla BPF utililizzata nei filtri di cattura.

Ad esempio se vogliamo visualizzare il display dei soli pacchetti UDP con porta 53 (traffico DNS) dobbiamo utilizzare la seguente dicitura:
`udp.port eq 53`

.port fa matching sia con la porta sorgente che con la porta destinazione. Tuttavia è possibile filtrare specificatamente i pacchetti utilizzando porta sorgente e destinazione:

`udp.srcport` specifica la porta sorgente
`udp.dstport` specifica la porta destinazione

Esiste anche il display filter "dns" che ci consente di visualizzare direttamente il traffico DNS.
Cheat Sheet: https://packetlife.net/media/library/13/Wireshark_Display_Filters.pdf

Wireshark – Statische

La potenzialità di Wireshark è data dai numerosi plugin e tool di statistiche che ci permettono di analizzare con una comoda GUI il traffico catturato.

`Statistics -> TCPStreamGraph -> RoundTripTime` ci permette di visualizzare il RTT di una stream TCP specifico tra due endpoint durante il tempo di cattura.

`Statistics -> TCPStreamGraph -> Throughput` permette di visualizzare il throughput di uno specifico stream TCP.

`Statistics -> TCPStreamGraph -> WindowScaling` permette di visualizzare l'andamento della finestra mobile (Window) della connessione TCP.

Wireshark – Statische

Esistono poi tantissime altre funzionalità di statistiche, come analisi a livello di pacchetti IP. **Vediamone alcune…**

`Statistic → I/O Graphs`: è possibile settare diverse opzioni per analizzare il traffico a livello di interfaccia di rete. Possiamo specificare attraverso dei display filter diverse categorie di traffico in un plot. Utile per discriminare il traffico di diverse applicazioni/protocolli.

tshark – interfaccia testuale

Wireshark fornisce anche un'interfaccia testuale, tramite il comando *tshark*. Ad esempio, per catturare e stampare tutte le richieste HTTP di tipo GET effettuate (si notino le significative potenzialità di customizzazione dell'output offerte dall'opzione –T fields):

```
$ ~ sudo tshark -i any \
            -Y 'http.request.method == "GET"' \
            -T fields \
            -e http.request.method -e http.request.uri -e ip.dst

Capturing on 'any'
GET     /       216.58.198.3
GET     /       216.58.210.195
```

(tshark risulta quindi una valida alternativa a *tcpdump*, fornendo più funzionalità e presentando un'interfaccia significativamente meno ostica rispetto a tcpdump.)

Esercizi

Utilizzare wireshark sull'opportuna interfaccia di rete per:

1) Catturare il traffico DNS, analizzare i pacchetti e identificare diversi tipi di query. Si utilizzi il comando nslookup per generare query DNS.
2) Catturare a analizzare il traffico DHCP.
3) Catturare i dati di una sessione TCP realizzata attraverso **netcat** o **nc**. Utilizzando l'apposito tool di wireshark si devono visualizzare i messaggi testuali scambiati tra client e server.
4) Catturare il traffico HTTP e visualizzare richieste.
5) Visualizzare eventuale traffico ICMP per la segnalazione di errori.