Università di Verona
Dipartimento di Informatica

Architetture Multimediali

Analisi del traffico di rete

Davide Quaglia
Giuseppe Di Guglielmo

Scopo di questa esercitazione è imparare l'utilizzo di tool per l'analisi del traffico di una rete, in particolare vengono presentati per l'ambiente Linux

- ***tcpdump*** : è un *sniffer*, cioè uno strumento che permette di monitorare il traffico di rete con la possibilità di utilizzare filtri per limitare la quantità di dati, ricavati dall'osservazione della rete secondo varie regole di matching per i pacchetti.
- ***wireshark*** (ex ***ethereal***) : è anch'esso un software di analisi di protocollo, o *sniffe*r, utilizzato per la risoluzione di problemi di rete, per l'analisi e per lo sviluppo di software e di protocolli di comunicazione e per la didattica. Le funzionalità che offre sono molto simili a quelle di tcpdump, ma in più è dotato di un'interfaccia grafica e di più funzionalità di ordinamento e filtraggio.
- ***tethereal*** : è, come gli altri due, un software di analisi di protocollo ed è la versione utilizzabile da terminale di wireshark.

Al fine di una completa comprensione dei contenuti è prerequisito importante la conoscenza dei fondamenti di rete soprattutto nel contesto TCP/IP.

Per approfondimenti sui comandi e sulle opzioni si faccia riferimento ai manuali dei tool, scaricabili da

- **tcpdump** http://www.tcpdump.org/tcpdump_man.html
- **ethereal** (**wireshark**) http://www.wireshark.org/docs/
- **thethereal** http://www.ethereal.com/docs/man-pages/tethereal.1.html

Questi tool di analisi si basano tutti sulla libreria C ***libpcap***. La libreria in questione è nata intorno al 1993 all'Università della California. È supportata pienamente dalla comunità opensource ed è reperibile al sito http://www.tcpdump.org (tcpdump è lo sniffer per eccellenza che sfrutta la libpcap).
Le principali funzioni di questa libreria sono la possibilità di cercare e trovare device di rete (intesi come network adapter), gestire potenti filtri di cattura, analizzare pacchetto per pacchetto. Permette la gestione degli errori, quindi un buon livello di debug, ed infine ottimi strumenti per statistiche sull'analisi effettuata.

1. Scaricamento e installazione

I tool si possono scaricare liberamente dalle rispettive pagine web o probabilmente sono presenti già nell'installazione della propria distribuzione Linux.

Esistono tool di analisi di protocolli di rete per l'ambiente Windows, ad esempio ***Analyzer*** del Politecnico di Torino (http://analyzer.polito.it/) oppure ***WinDump*** prelevabile al sito http://www.winpcap.org/windump.

ATTENZIONE: per poter utilizzare le funzionalità di cattura di questi tool in ambiente Linux bisogna essere autenticati come utente *root* o aver installato il tool con *setuid* a root. In ogni caso questi tool possono essere utilizzati per analizzare catture precedentemente effettuate da utente root.

2. Alcuni concetti sullo *sniffing*

Sniffing in **reti ethernet non-switched**: In questo tipo di reti ethernet il mezzo trasmissivo è condiviso tramite un *hub* centrale, quindi tutte le schede di rete dei computer nella rete locale ricevono tutti i pacchetti, anche quelli destinati ad altri, selezionando i propri a seconda dell'indirizzo ***MAC*** (indirizzo hardware specifico della scheda di rete). Lo sniffing in questo caso consiste nell'impostare sull'interfaccia di rete la cosiddetta **modalità promiscua**, che disattiva il "filtro hardware" basato sul MAC permettendo al sistema l'ascolto di tutto il traffico passante sul cavo.

Sniffing in **reti ethernet switched**: In questo caso l'apparato centrale della rete, definito switch, si preoccupa, dopo un breve transitorio, di inoltrare su ciascuna porta solo il traffico destinato al dispositivo collegato a quella porta; ciascuna interfaccia di rete riceve, quindi solo i pacchetti destinati al proprio indirizzo, i pacchetti multicast e quelli broadcast. L'impostazione della modalità promiscua è quindi insufficiente per poter intercettare il traffico in una rete gestita da switch. Un metodo per poter ricevere tutto il traffico dallo switch da una porta qualunque è il **MAC flooding**. Tale tecnica consiste nell'inviare ad uno switch pacchetti appositamente costruiti per riempire la *CAM table* dello switch di indirizzi MAC fittizi. Questo attacco costringe lo switch ad entrare in una condizione detta di *fail open* che lo fa comportare come un hub, inviando così gli stessi dati a tutti gli apparati ad esso collegati.

3. Utilizzo di *tcpdump*

Questo applicativo è un tool di cattura, attraverso il quale è possibile monitorare il traffico in una rete. Il tool permette, tra l'altro, di limitare la cattura dei pacchetti impostando dei filtri basati, ad esempio, sull'interfaccia di ascolto, sul protocollo o sulla porta utilizzata. Sono inoltre disponibili una serie di opzioni, in particolare vi è la possibilità di limitare il numero di pacchetti catturati o quanti byte acquisire per ciascun pacchetto.

Si riporta un esempio di comando (da eseguire come *root* !)

```
$ tcpdump -i eth0 -w eth0.log -s0 tcp port 25
```

Analizziamo il comando con le varie opzioni utilizzate:

-i eth0 : rappresenta l'interfaccia dalla quale si intende catturare

-w eth0.log : rappresenta il nome del file che conterrà i pacchetti catturati

-s0 : la flag *-s* permette di specificare quanti byte acquisire da ogni singolo pacchetto (default 68); con *-s0* si richiede di acquisire l'intero pacchetto

tcp : le catture da effettuarsi saranno relative a questo protocollo

port 25 : questo campo limita le catture ai soli pacchetti che utilizzano come source e destination ID la porta 25, che, nel caso TCP, è la porta utilizzata dal protocollo *SMTP* (*Simple Mail Transport Protocol*).

Per interrompere il monitoraggio occorre utilizzare la combinazione *CTRL + C*.

Altri esempi di utilizzo di tcpdump:

Viene selezionata l'interfaccia *eth0* e catturato il flusso da e verso *edalab-srv01.sci.univr.it*, scrivendo sul file *eth0.log*

```
$ tcpdump -i eth0 -w eth0.log host edalab-srv01.sci.univr.it
```

Come il precedente esempio ma in questo caso vengono letti solo i pacchetti IP del protocollo TCP

```
$ tcpdump -i eth0 -w eth0.log \
  ip proto tcp host edalab-srv01.sci.univr.it
```

Come il precedente esempio ma in questo caso vengono letti solo i pacchetti da e verso la porta *23* (*telnet*)

```
$ tcpdump -i eth0 -w eth0.log \
  ip proto tcp host edalab-srv01.sci.univr.it \
  and port 23
```

Come il precedente esempio ma in questo caso vengono letti solo i pacchetti verso l'host *edalab-srv01.sci.univr.it (dst* sta per *destination* mentre *src* è *source)*

```
$ tcpdump -i eth0 -w eth0.log \
  ip proto tcp \
  dst host edalab-srv01.sci.univr.it \
  and port 23
```

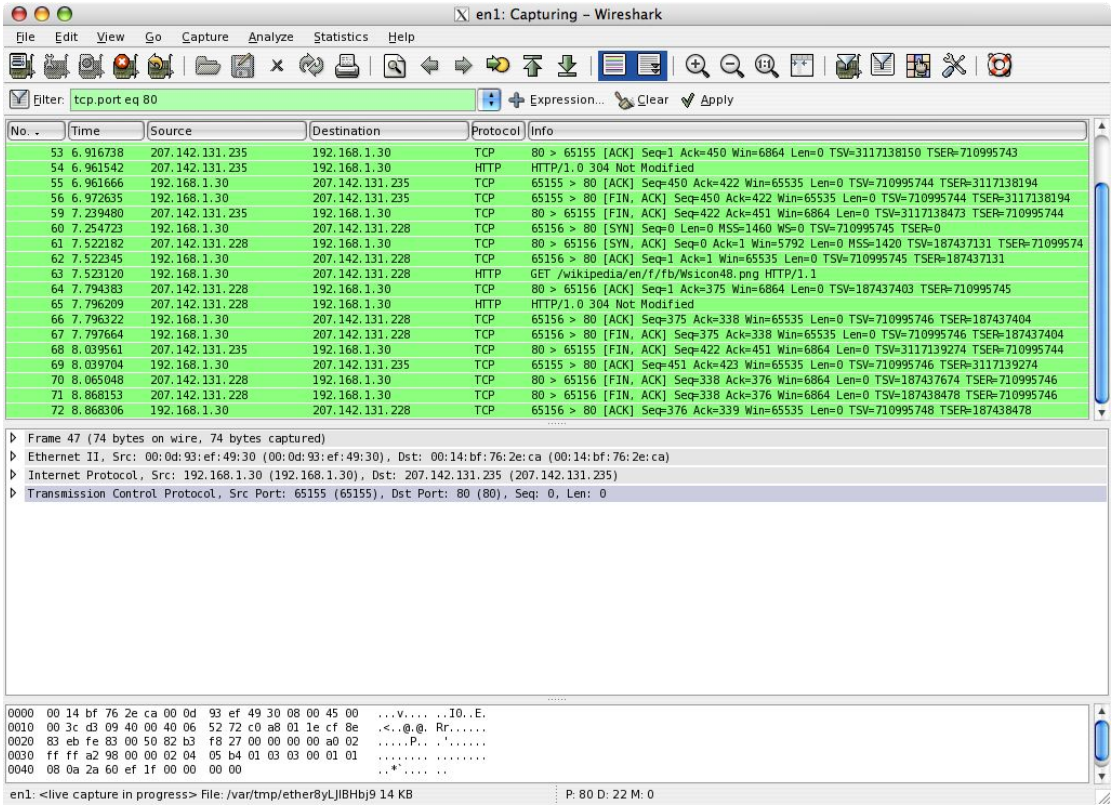
Le capacità di analisi e filtraggio vanno ben oltre questi esempi introduttivi, si consiglia di far riferimento alle pagine del manuale per verificarne le potenzialità.

3. Utilizzo di *wireshark*

Per lanciare l'applicazione:

```
$ wireshark
```

si ricordi che è necessario essere l'utente *root* se si intende fare una cattura diretta dalle interfacce di rete.



Alcune caratteristiche:

- i dati possono essere acquisiti “from the wire” (direttamente dal cavo) oppure possono essere letti su un file di cattura precedente
- i dati possono essere catturati dal vivo da reti Ethernet, IEEE 802.11, Token Ring, ecc.
- i dati di rete catturati possono essere esplorati tramite un'interfaccia grafica
- i filtri di visualizzazione possono essere usati per colorare o evidenziare selettivamente le informazioni sommarie sui pacchetti
- la visualizzazione dei dati può essere filtrata
- i protocolli di comunicazione possono essere scomposti, in quanto riesce a “comprendere” la struttura dei diversi protocolli di rete, quindi è in grado di visualizzare incapsulamenti e campi singoli, ed di interpretare il loro significato
- è possibile estrarre il contenuto del livello applicazione di una sessione TCP.

Il modo più immediato per avviare l'ascolto sull'interfaccia di rete di default è dal menu in alto di wireshark `Capture/Start`

E' possibile selezionare su quale interfaccia porsi in ascolto con la voce `Capture/Interface`

Per raffinare il processo di analisi e cominciare ad applicare dei filtri usare la voce `Capture/Options/Capture Filter [Start]`
In questo modo a priori si impone uno o più filtri rendendo più piccolo il file di log.

Dopo aver acquisito lo stream di dati è possibile analizzarlo a posteriori sempre mediante filtri con la voce `Analyze/Filter`

E' possibile seguire l'intero flusso di dati di una “conversazione” TCP mediante la voce `Analyze/Follow TCP Stream`

Infine è possibile effettuare delle statistiche generali con la voce `Statistics/Summary`

oppure osservare come si ridistribuisce il traffico sulla pila dei protocolli con la voce `Statistics/Protocol Hierarchy`

oppure individuare le conversazioni avvenute tra host con la voce `Statistics/Conversations`

4. Esercizi

Esercizio 1. utilizzare tcpdump o wireshark in modo da catturare sull’interfaccia principale di ricezione del traffico della macchina, tutti i pacchetti destinati al proprio indirizzo (niente modalità promiscua). Quindi “pingare” l’indirizzo della macchina (dove sti sta eseguendo tcpdump) e controllare il traffico ricevuto.

Esercizio 2. Usare tcpdump o wireshark in modo da catturare su file i primi 100 pacchetti provenienti o destinati a google.com sulla porta http.

Esercizio 3. Usare wireshark per “sniffare” la password dell'utente *pippo* che si connette all'host *edalab-res02.sci.univr.it* tramite *TELNET* (il file di log da utilizzare si trova nel pacchetto degli esercizi relativi alla lezione).