

Lezione 4: Introduzione a L3, il protocollo ARP

Claudio Ardagna, Patrizio Tufarolo – Università degli Studi di Milano

Insegnamento di Laboratorio di Reti di Calcolatori



Introduzione – 1

- ▶ Il livello 3, anche detto livello di rete, è quello che permette la trasmissione logica tra due host arbitrari, generalmente non direttamente connessi
- ▶ Esso riceve segmenti dal livello di trasporto (L4), e compone i pacchetti che verranno poi incapsulati dal livello datalink (L2)
- ▶ Le sue funzionalità sono
 - ▶ Possibilità di comunicazione senza stabilire una connessione
 - ▶ Indirizzamento degli host
 - ▶ Inoltro e instradamento dei pacchetti
 - ▶ Frammentazione e riassemblaggio dei pacchetti

Introduzione – 2

- ▶ Il protocollo L3 che andremo a studiare è il protocollo IP (Inter-networking Protocol)
- ▶ IP è nato per inter-connettere reti eterogenee, garantendo interlavoro e interoperabilità
- ▶ Ne esistono due versioni
 - ▶ IPv4: Consente di assegnare a ogni dispositivo di rete un indirizzo univoco a 32 bit, composto da 4 ottetti; prevede l'esistenza di 2^{32} host
 - ▶ IPv6: Nuova versione del protocollo, adottata ancora da pochi, con indirizzamento a 128 bit (2^{128} host)
- ▶ L'associazione tra indirizzo IP e indirizzo fisico è gestita tramite il protocollo ARP



Terminologia - 1

- ▶ ISO/OSI (Open System Interconnection)
- ▶ Ethernet
- ▶ MAC Address

Terminologia - 1

- ▶ ISO/OSI (Open System Interconnection)
 - ▶ Standard de iure che organizza l'architettura di una rete di calcolatori in una struttura composta da 7 livelli (stack di rete)
- ▶ Ethernet
 - ▶ Famiglia di tecnologie standardizzate per le reti che definisce specifiche tecniche per i livelli 1 e 2 (fisico e MAC) dello stack ISO/OSI
- ▶ MAC Address
 - ▶ Media Access Control Address, o indirizzo fisico, indirizzo a 48 bit che identifica univocamente un'interfaccia di rete

Terminologia – 2

- ▶ IP Protocol
 - ▶ Protocollo di livello 3 che permette l'interconnessione di reti eterogenee, consentendo l'interazione tra dispositivi posti in due reti di tipologie diverse, e/o non collegate tra di loro.
- ▶ ARP (Address Resolution Protocol)
 - ▶ Protocollo che si colloca tra livello 2 e livello 3, permettendo l'associazione di un indirizzo IPv4 al corrispondente indirizzo fisico (RFC 826)
 - ▶ Il protocollo analogo ad ARP per IPv6 è il Neighbor Discovery Protocol (RFC 4861)
- ▶ RARP (Reverse Address Resolution Protocol)
 - ▶ Protocollo usato per tradurre gli indirizzi ethernet in indirizzo IP (inverso di ARP) (RFC 903)
 - ▶ Consente, ad esempio, ad un host di scoprire il proprio indirizzo IP all'accensione, chiedendolo in broadcast agli altri host connessi alla rete
 - ▶ La sua funzionalità è stata resa obsoleta da BOOTP e da DHCP

Il protocollo ARP

- ▶ Collega il protocollo IP al protocollo implementato a livello datalink
- ▶ Lavora in broadcast: se il protocollo a livello datalink non implementa il broadcast (come ATM) non può essere usato!
- ▶ Se l'indirizzo IP sorgente e l'indirizzo IP destinazione appartengono alla stessa sottorete
 - ▶ Viene utilizzato ARP per ricavare l'indirizzo MAC di destinazione e compilare il frame ethernet (L2)
- ▶ Se l'indirizzo IP sorgente e l'indirizzo IP destinazione appartengono a due sottoreti diverse
 - ▶ Il frame ethernet viene compilato con l'indirizzo MAC del router di default

La tabella ARP

- ▶ Ogni host ha una tabella locale (ARP Table), che utilizza come cache
- ▶ Se un host vuole contattare un altro host, conoscendo l'indirizzo IP, andrà a leggere il corrispondente indirizzo nella ARP Table
- ▶ Se l'indirizzo non è presente nella ARP Table, l'host procederà inviando una richiesta ARP

Manipolare la tabella ARP con Linux - 1

- ▶ Il comando che consente di manipolare la tabella ARP su Linux è
 - ▶ `ip neighbor`
- ▶ Tramite questo comando è possibile aggiungere, rimuovere, aggiornare una entry nella tabella ARP
- ▶ L'operazione più comune eseguita manualmente sulla tabella ARP è quella di rimozione di una entry
- ▶ Infatti il popolamento della tabella avviene in automatico; alla disconnessione di un host invece la tabella rimane popolata e può essere di interesse dell'amministratore di sistema pulire la cache ARP

Manipolare la tabella ARP con Linux - 2

- ▶ Un comando alternativo è il comando
 - ▶ `arp`
- ▶ Tramite il comando `arp`:
 - ▶ `arp -d <indirizzo_ip>`
 - ▶ Cancella un indirizzo dalla tabella arp
 - ▶ `arp -s <indirizzo_ip> <indirizzo_mac>`
 - ▶ Aggiunge una coppia di indirizzi ip-mac alla tabella ARP
 - ▶ `arp -f`
 - ▶ Legge la tabella ARP da un file (default: `/etc/ethers`)
- ▶ Le entry ARP statiche possono essere rese permanenti agendo sul file `/etc/ethers` dei singoli host (Manuale: `man ethers`)
 - ▶ Ogni linea di questo file è una coppia
 - ▶ `<indirizzo_mac> <indirizzo_ip>`

Il protocollo ARP – Funzionamento

- ▶ L'host che vuole ottenere l'indirizzo MAC invia un pacchetto di **arp request** in **broadcast**
- ▶ La arp request contiene **l'indirizzo MAC del mittente** e **l'indirizzo IP** del quale si vuole scoprire l'indirizzo fisico
- ▶ Essendo una richiesta broadcast, sarà ricevuta da tutti gli host
- ▶ L'host che riconoscerà il proprio indirizzo IP all'interno della arp request, invierà una **arp reply** in **broadcast**
- ▶ Tutti gli host della rete riceveranno anche la arp reply ed aggiorneranno la propria tabella ARP, in modo da non dover fare ulteriori richieste

Problema di sicurezza!

- ▶ ARP non è un protocollo autenticato! Chiunque può rispondere, anche in modo illegittimo, alle richieste ARP
- ▶ È basato quindi sulla fiducia che la risposta avvenga da un host legittimo
- ▶ Possono verificarsi i seguenti scenari
 - ▶ **Conflitto di indirizzi IP**
 - ▶ Due host della rete, con MAC address differenti, hanno impostato il medesimo indirizzo IP. Questo provoca due ARP Reply, provenienti da host differenti, causando un malfunzionamento
 - ▶ **ARP Spoofing (ARP cache poisoning)**
 - ▶ Un host della rete invia in modo ripetitivo e forzato delle ARP Reply contenente dati falsati, alterando («avvelenando») la tabella ARP degli host presenti sulla rete
 - ▶ Viene utilizzato in attacchi MITM
 - ▶ Contromisura: tabella ARP statica (ad es., impostata tramite *ip neighbor*)
 - ▶ Può essere anche legittimo: ad esempio captive portal

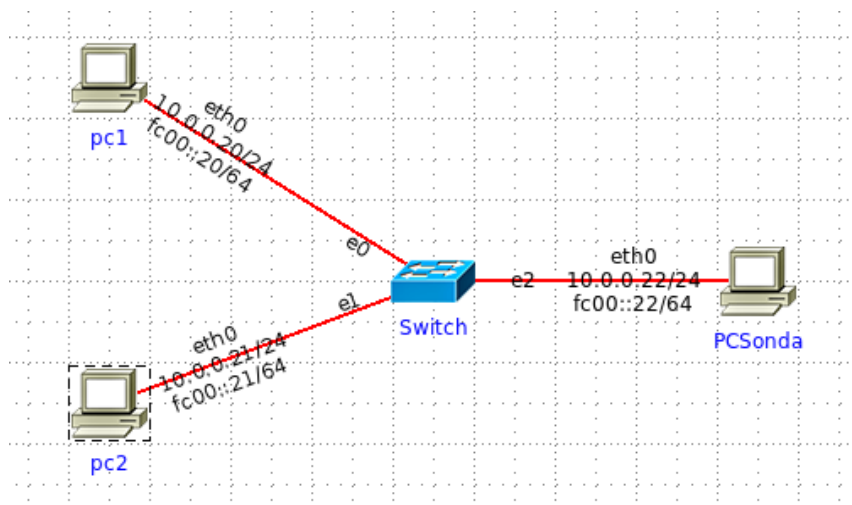


Esercizio 1 – Osservare ARP

- ▶ Creare una topologia composta da:
 - ▶ 1 Switch
 - ▶ 2 PC, connessi allo switch
 - ▶ Un PC sonda, connesso allo switch, senza impostare la porta di mirroring (in grado di vedere soltanto il traffico in broadcast)
- ▶ Avviare tcpdump sul pc sonda
- ▶ Effettuare un PING da PC1 a PC2 ed osservare lo scambio di informazioni effettuato con il protocollo ARP
- ▶ Osservare le tabelle ARP degli host con uno dei due comandi mostrati

Esercizio 1 - Soluzione

Topologia IMUNES



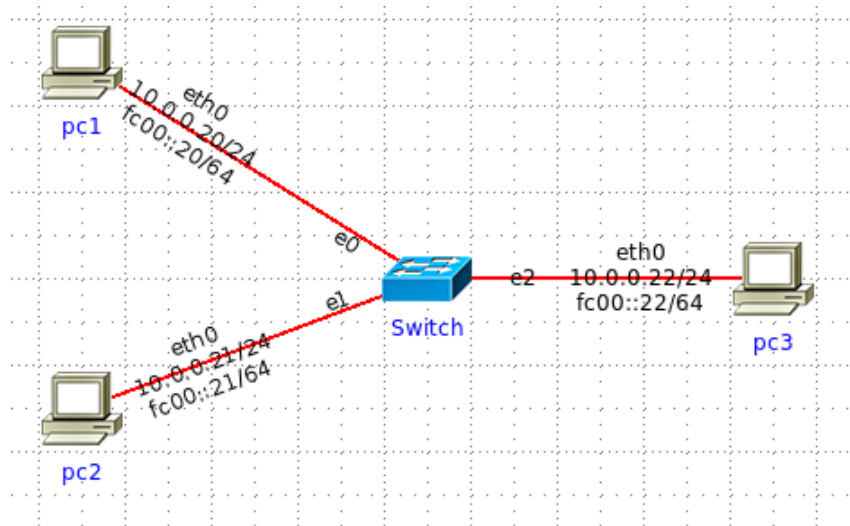
- ▶ Step 1
 - ▶ PCSonda
 - # tcpdump
- ▶ Step 2
 - ▶ pc1
 - # ping 10.0.0.21
- ▶ Output di tcpdump
 - ▶ Arp, request, who-has 10.0.0.21 ? Tell 10.0.0.20, length 28
- ▶ Risultato – tabella ARP di pc1 (ip neigh)
 - ▶ 10.0.0.21 dev eth0 lladdr 42:00:aa:00:00:01 STALE

Esercizio 2 – Arp spoofing and cache poisoning

- ▶ Creare una topologia a stella composta da:
 - ▶ 3 PC
 - ▶ 1 Switch (centro stella)
- ▶ Su PC3 (attaccante) avviare due attacchi di ARP Spoofing contro PC1 e PC2
 - ▶ Avvelenando la cache ARP di PC1 al fine di mandare tutto il traffico IP PC1 → PC2 verso PC3
 - ▶ Avvelenando la cache ARP di PC2 al fine di mandare tutto il traffico IP PC2 → PC1 verso PC3
- ▶ Il comando per lanciare un attacco di ARP Spoofing è:
 - ▶ `arp spoof -i <interfaccia> -t <target> <ip_da_falsificare>`
- ▶ Osservare le ARP table compromesse

Esercizio 2 - Soluzione

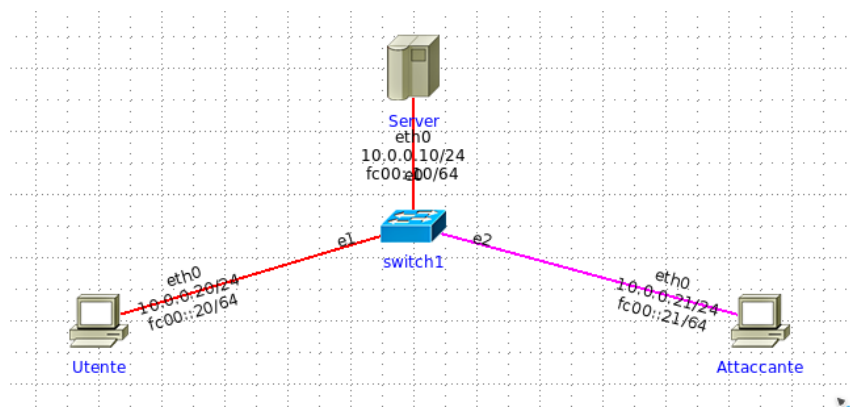
Topologia IMUNES



- ▶ Step 1
 - ▶ pc3
 - ▶ `# arpspoof -i eth0 -t 10.0.0.21 10.0.0.20`
- ▶ Step 2 (in parallelo)
 - ▶ pc3
 - ▶ `# arpspoof -i eth0 -t 10.0.0.20 10.0.0.21`
- ▶ Output di arpspoof (1)
 - ▶ `42:0:aa:0:0:2 42:0:aa:0:0:0 0806 42: arp reply 10.0.0.21 is at 42:0:aa:0:0:2`
- ▶ Output di arpspoof (2)
 - ▶ `42:0:aa:0:0:2 42:0:aa:0:0:1 0806 42: arp reply 10.0.0.20 is at 42:0:aa:0:0:2`
- ▶ Risultato – tabella ARP di pc1 dopo l' attacco:
 - ▶ `10.0.0.21 dev eth0 lladdr 42:00:aa:00:00:02 REACHABLE`
 - ▶ `10.0.0.22 dev eth0 lladdr 42:00:aa:00:00:02 REACHABLE`
- ▶ **L'IP di pc2 è associato al MAC di PC3 sulla tabella ARP di PC1! (analogamente per la tabella ARP di PC2)**

Esempio pratico: scenario di attacco 1

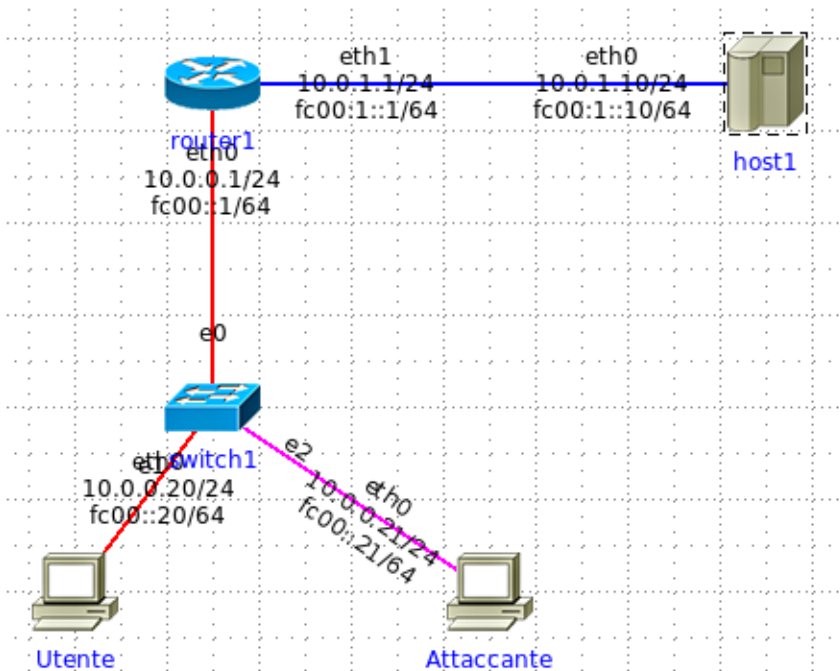
IMUNES



- ▶ L'utente e un webserver possono comunicare tramite uno switch
- ▶ Tramite ARP Cache poisoning l'attaccante avvelena la cache del client, effettuando lo spoofing dell'IP del server, e server
- ▶ L'utente, invece di contattare il server, contatterà l'attaccante
- ▶ Morale:
 - ▶ attenzione alle tabelle ARP nel setup di reti locali
 - ▶ mai effettuare autenticazione tramite IP

Esempio pratico: scenario di attacco 2

IMUNES



- ▶ Il router e lo switch sono il vostro router e il vostro switch di casa
- ▶ Siete tranquillamente connessi ad internet, e state navigando verso host1 (10.0.1.10).
- ▶ Un utente attacca la vostra Wi-Fi (link fucsia), tramite un attacco di ARP Poisoning si intromette tra voi e il server, intercettando e leggendo il vostro traffico di rete (es. tramite TCP Dump)

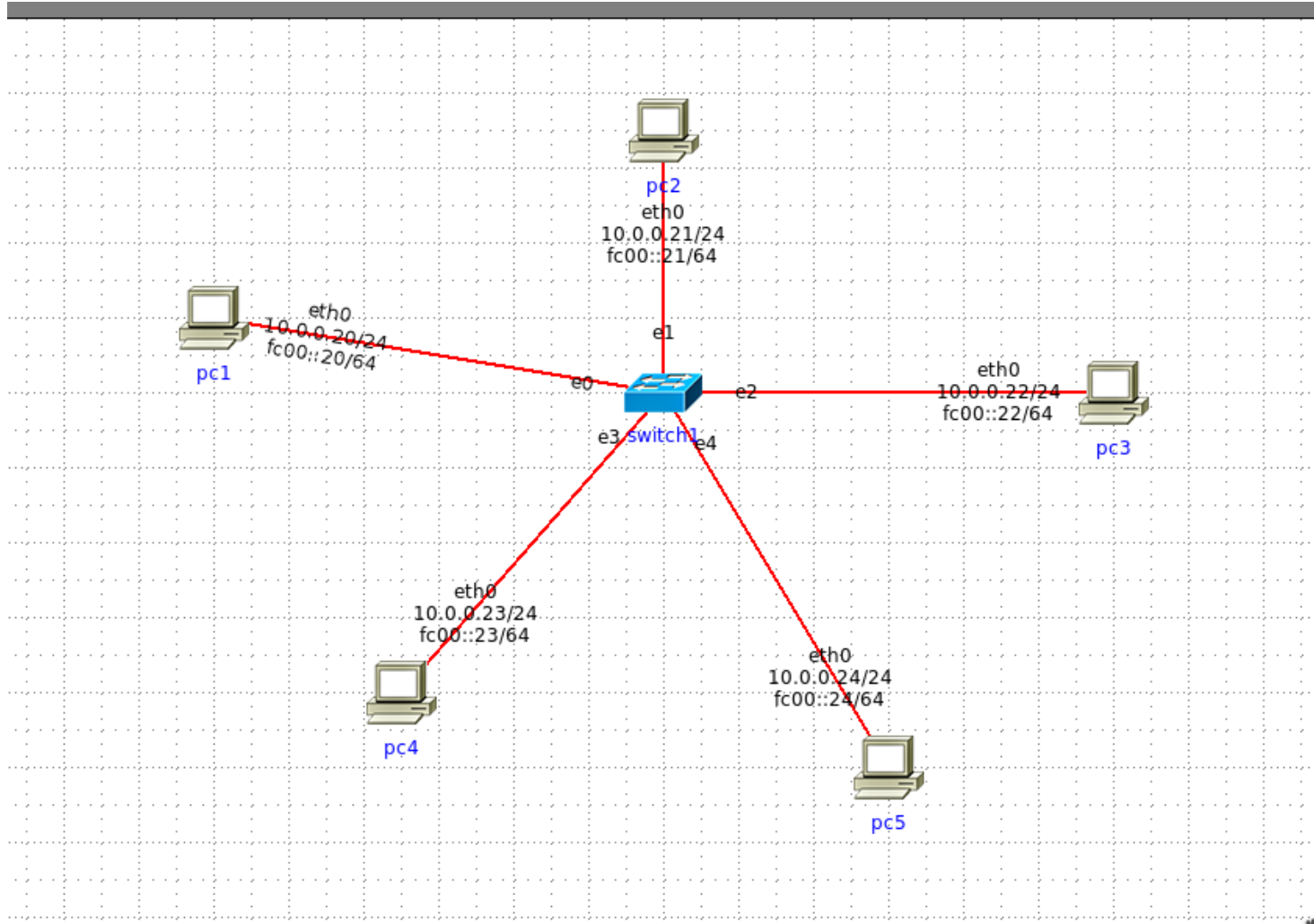
Tabella ARP Statica

- ▶ Per proteggerci da questi scenari possiamo utilizzare una tabella ARP statica, impostando delle entry almeno per gli host critici
- ▶ Ciò comporta due benefici:
 - ▶ Miglioramento delle performance (quando si tenterà di contattare l'host con la entry statica, non sarà necessario effettuarne la risoluzione)
 - ▶ Evitare gli attacchi di ARP spoofing e i conflitti di indirizzi ip sulla rete

Esercizio 3 – Tabella ARP Statica

- ▶ Creare una topologia a stella composta da:
 - ▶ 5 PC
 - ▶ 1 Switch (centro stella)
- ▶ Popolare il file `/etc/ethers` costituendo una ARP table statica, in modo da prevenire eventuali attacchi di IP spoofing

Esercizio 3 - Soluzione



Esercizio 3 – Soluzione - /etc/ethers

```
[utente@macchina ~]$ sudo cat << EOF >  
/etc/ethers
```

```
42:00:aa:00:00:00 10.0.0.20  
42:00:aa:00:00:01 10.0.0.21  
42:00:aa:00:00:02 10.0.0.22  
42:00:aa:00:00:03 10.0.0.23  
42:00:aa:00:00:04 10.0.0.24
```

```
EOF
```

```
[utente@macchina ~]$ arp -f
```

Contenuto effettivo
del file (potete
usare anche **nano** o
vim per fare la stessa
cosa)



Conclusioni

- ▶ Abbiamo visto il protocollo ARP
- ▶ Abbiamo imparato a manipolare la tabella ARP su Linux
- ▶ Abbiamo visto una vulnerabilità del protocollo ARP e approfondito due scenari di attacco
- ▶ Abbiamo configurato una tabella ARP statica analizzandone i benefici



