
Tesina di ”Sicurezza dei sistemi informatici”

Politecnico di Torino

# Manipolazione dei protocolli di rete con ettercap

31/01/2008

**Alberto Realis-Luc**

Matr. 142119

< Indice

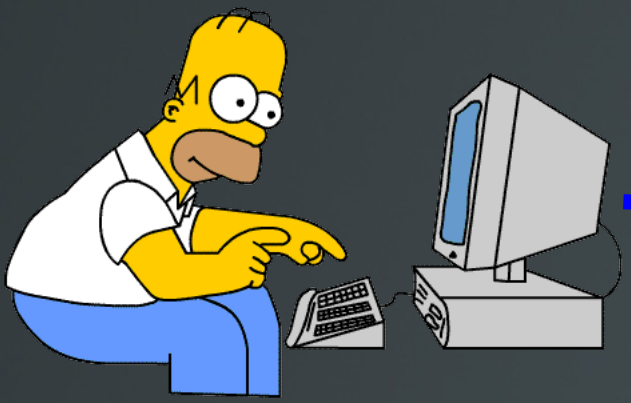
## Indice

## Introduzione


Ettercap è un tool per fare attacchi "*Man in the middle*" in reti LAN Ethernet basate sull'uso di switch. Si basa sulla manipolazione dei protocolli di rete di basso livello, in particolar modo opera principalmente sui protocolli ARP e DHCP. Permette di perpetrare attacchi MITM anche se non ci si trova collegati fisicamente nel mezzo tra le due vittime, è invece sufficiente essere collegati su una porta dello stesso switch di rete. In pratica ettercap permette di far passare tutto il traffico tra due host, o tra due gruppi di host per la macchina sulla quale viene esguito. L'attaccante avrà quindi a disposizione tutto il traffico che potrà quindi intercettare ma anche modificare per i più svariati scopi.

# Introduzione

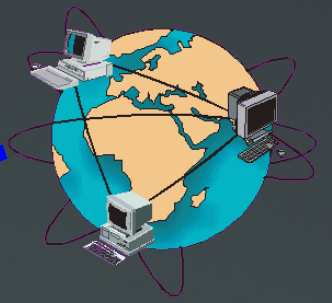
Classico scenario per attacchi con ettercap



Client

Switch

Gateway



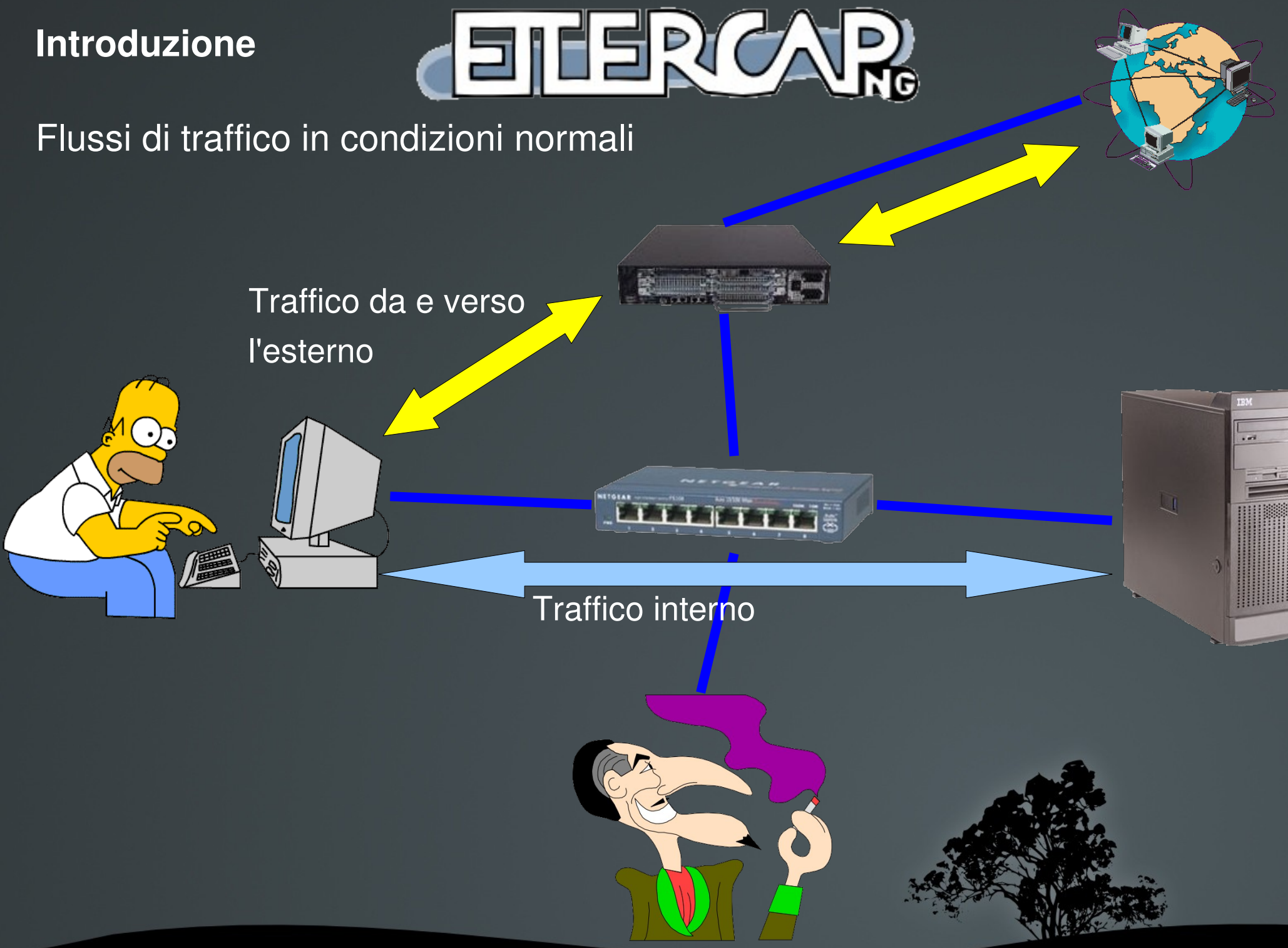
Internet

Server

Attaccante

# Introduzione

Flussi di traffico in condizioni normali

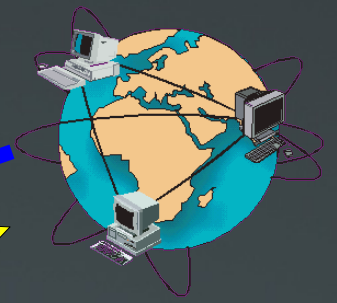


## Introduzione

Flussi di traffico dirottati

3 modi per farlo con ettercap:

1. **ARP poisoning**
2. **DHCP spoofing**
3. **Port stealing**

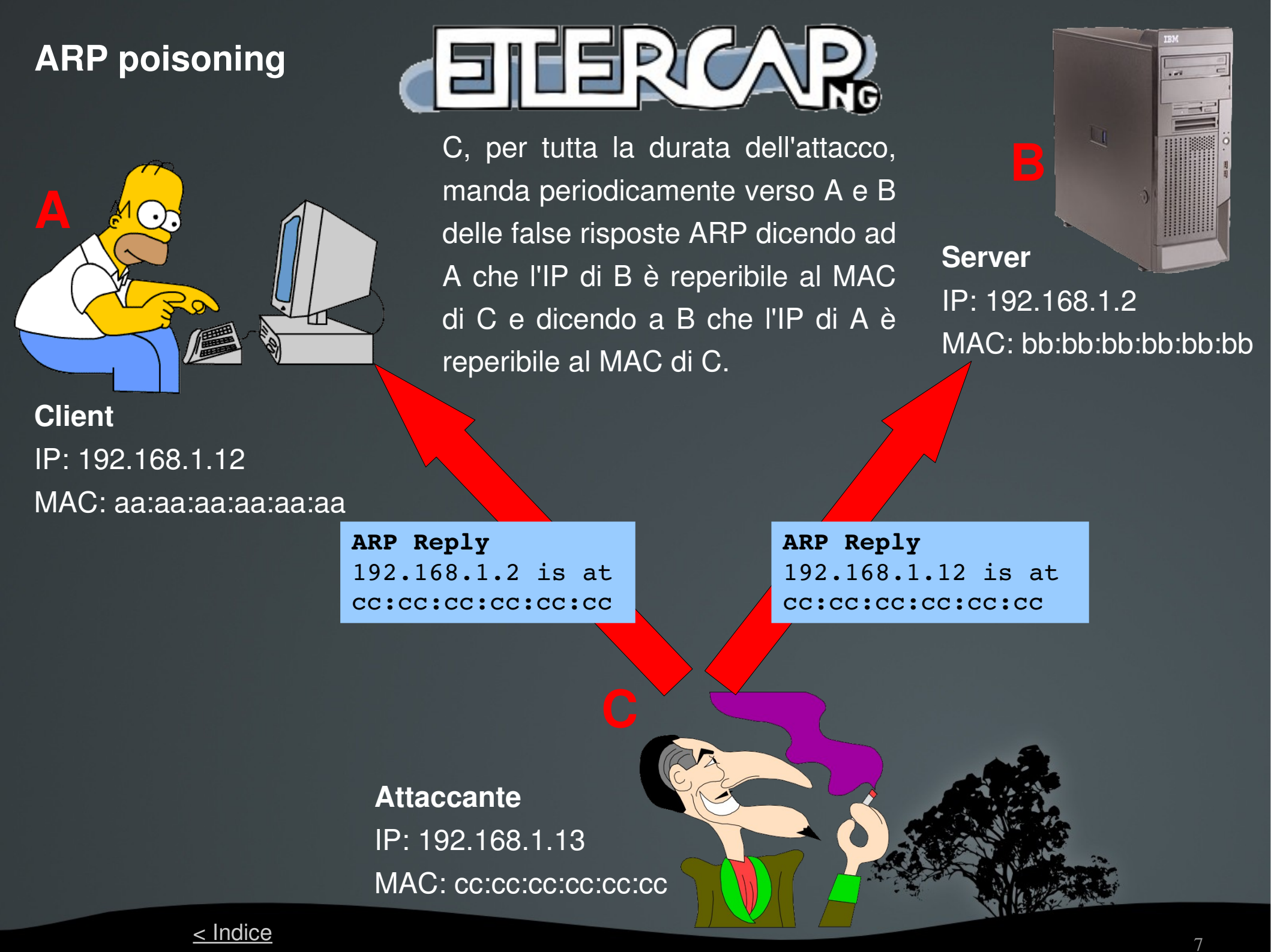


Traffico esterno

Traffico interno

L'attaccante dopo aver sniffato o modificato i pacchetti in transito provvederà a inoltrali verso la vera destinazione.

# ARP poisoning




C, per tutta la durata dell'attacco, manda periodicamente verso A e B delle false risposte ARP dicendo ad A che l'IP di B è reperibile al MAC di C e dicendo a B che l'IP di A è reperibile al MAC di C.

A

**Client**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa

B

**Server**

IP: 192.168.1.2

MAC: bb:bb:bb:bb:bb:bb

**ARP Reply**
192.168.1.2 is at
cc:cc:cc:cc:cc:cc

**ARP Reply**
192.168.1.12 is at
cc:cc:cc:cc:cc:cc

C

**Attaccante**

IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

< Indice

# ARP poisoning


A e B ricevendo questi avvisi ARP provvederanno ad aggiornare la loro cache ARP che sarà ora 'avvelenata' con l'indirizzo MAC dell'attaccante. Dunque tutto il traffico scambiato tra A e B verrà spedito in realtà verso l'attaccante.


**Client**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa

**Server**

IP: 192.168.1.2

MAC: bb:bb:bb:bb:bb:bb

```
ARP Cache
192.168.1.1 -> 00:c0:49:ac:8b:06
192.168.1.2 -> cc:cc:cc:cc:cc:cc
192.168.1.9 -> 00:40:95:d1:90:82
...
```

```
ARP Cache
192.168.1.1 -> 00:c0:49:ac:8b:06
192.168.1.12-> cc:cc:cc:cc:cc:cc
192.168.1.9 -> 00:40:95:d1:90:82
...
```

# ARP poisoning

**A**

**Client**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa

**B**

**Server**

IP: 192.168.1.2

MAC: bb:bb:bb:bb:bb:bb

A spedisce un qualsiasi pacchetto IP verso B. C dopo averlo ricevuto lo inoltra a B usando il suo MAC corretto.

**Ethernet**
From: aa:aa:aa:aa:aa:aa
To: cc:cc:cc:cc:cc:cc

**IP**
From: 192.168.1.12
To: 192.168.1.2
.....
.....

**Ethernet**
From: cc:cc:cc:cc:cc:cc
To: bb:bb:bb:bb:bb:bb

**IP**
From: 192.168.1.12
To: 192.168.1.2
.....
.....

**C**

**Attaccante**

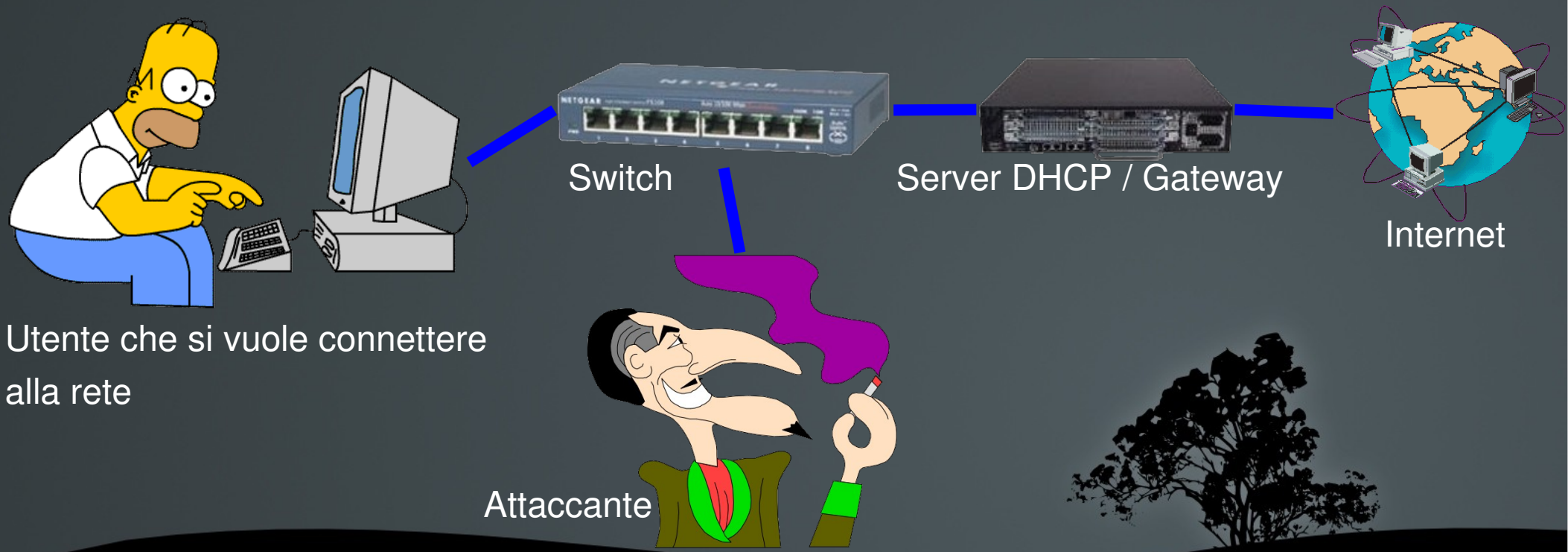
IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

< Indice

# DHCP spoofing

Un nuovo utente che si vuole connettere alla LAN, se non ha già un indirizzo IP statico preimpostato, dovrà farsene assegnare uno da un server DHCP. Oltre a far questo il server DHCP comunica al nuovo utente l'IP del gateway da utilizzare per uscire all'esterno della LAN, e quasi sempre anche un paio di indirizzi IP di server DNS.
L'attaccante tenterà di sostituirsi al vero DHCP fornedo il proprio IP come gateway così da poter intercettare tutto il traffico generato dalla vittima verso Internet. Inoltre l'attaccante potrebbe sostituirsi anche al server DNS potendo così dirottare la vittima su server fake.




< Indice

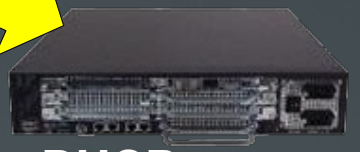
# DHCP spoofing

Il nuovo host manda un pacchetto DHCP discover, in broadcast, alla ricerca di un server DHCP. Ovviamente anche l'attaccante lo riceve.

**Utente che si connette**

MAC: aa:aa:aa:aa:aa:aa

```
DHCP Discover
From: aa:aa:aa:aa:aa:aa
To: ff:ff:ff:ff:ff:ff (Broadcast)
```



**Server DHCP**

IP: 192.168.1.1

MAC: 00:c0:49:ac:8b:06

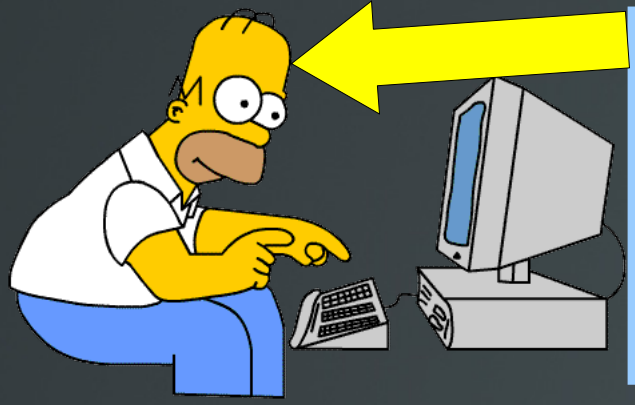

**Attaccante**

IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

# DHCP spoofing

In ettercap è possibile impostare un insieme di IP da assegnare alla vittime, ma dato che l'attaccante non sa a priori quali IP sono già usati e quale sia il piano di assegnazione di indirizzi IP del vero server DHCP, gli conviene lasciare che sia quest'ultimo ad assegnare l'indirizzo IP al nuovo host appena connesso. Il server DHCP offre l'IP 192.168.1.12 al nuovo utente.



**Utente che si connette**

MAC: aa:aa:aa:aa:aa:aa

```
DHCP Offer
From: 00:c0:49:ac:8b:06
To: ff:ff:ff:ff:ff:ff (Broadcast)

Offered IP: 192.168.1.12
for aa:aa:aa:aa:aa:aa
...
```

**Server DHCP**

IP: 192.168.1.1

MAC: 00:c0:49:ac:8b:06


**Attaccante**

IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

# DHCP spoofing

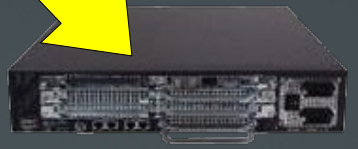
Il nuovo host accetta e richiede l'IP proposto dal server DHCP mandando un pacchetto DHCP request.

**Utente che si connette**

MAC: aa:aa:aa:aa:aa:aa

```
DHCP Request
From: aa:aa:aa:aa:aa:aa
To: ff:ff:ff:ff:ff:ff (Broadcast)

Requested IP: 192.168.1.12
for aa:aa:aa:aa:aa:aa
```



**Server DHCP**

IP: 192.168.1.1

MAC: 00:c0:49:ac:8b:06

La vittima lo accetta...

**Attaccante**

IP: 192.168.1.13

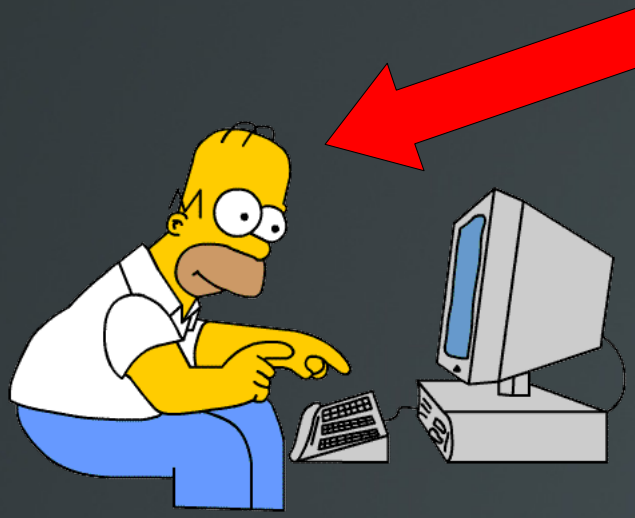
MAC: cc:cc:cc:cc:cc:cc

# DHCP spoofing

L'attaccante risponde subito, al posto del vero server DHCP, proponendosi come gateway.

```
DHCP ACK
From: 00:c0:49:ac:8b:06
To: aa:aa:aa:aa:aa:aa

Your IP: 192.168.1.12
for aa:aa:aa:aa:aa:aa
Gateway: 192.168.1.13
DNS1: 151.99.125.2
DNS2: 151.99.125.3
Lease time: 30 min
```



**Utente che si connette**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa

**Server DHCP**

IP: 192.168.1.1

MAC: 00:c0:49:ac:8b:06


**Attaccante**

IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

# DHCP spoofing

Dopo aver fatto alcune verifiche anche il vero DHCP risponde con DHCP ACK, ma non verrà più preso in considerazione, avendo, il nuovo host, già ricevuto un DHCP ACK.

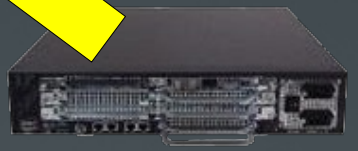
```
DHCP ACK
From: 00:c0:49:ac:8b:06
To: ff:ff:ff:ff:ff:ff (Broadcast)

Your IP: 192.168.1.12
for aa:aa:aa:aa:aa:aa
Gateway: 192.168.1.1
DNS1: 151.99.125.2
DNS2: 151.99.125.3
Lease time: 24 hours
```

**Utente che si connette**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa



**Server DHCP**

IP: 192.168.1.1

MAC: 00:c0:49:ac:8b:06


**Attaccante**

IP: 192.168.1.13

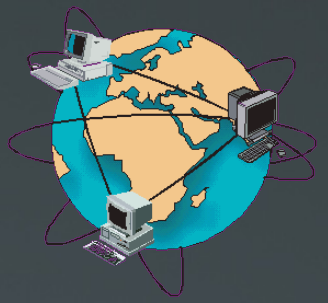
MAC: cc:cc:cc:cc:cc:cc

# DHCP spoofing

**Nuovo utente**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa



Internet

Ora l'attaccante può intecettare tutto il traffico generato dalla vittima verso Internet

**Ethernet**
From: aa:aa:aa:aa:aa:aa
To: cc:cc:cc:cc:cc:cc
**IP**
From: 192.168.1.12
To: 130.192.73.1
.....
.....

**Ethernet**
From: aa:aa:aa:aa:aa:aa
To: 00:c0:49:ac:8b:06
**IP**
From: 192.168.1.12
To: 130.192.73.1
.....
.....

**Gateway**

IP: 192.168.1.1

MAC: 00:c0:49:ac:8b:06

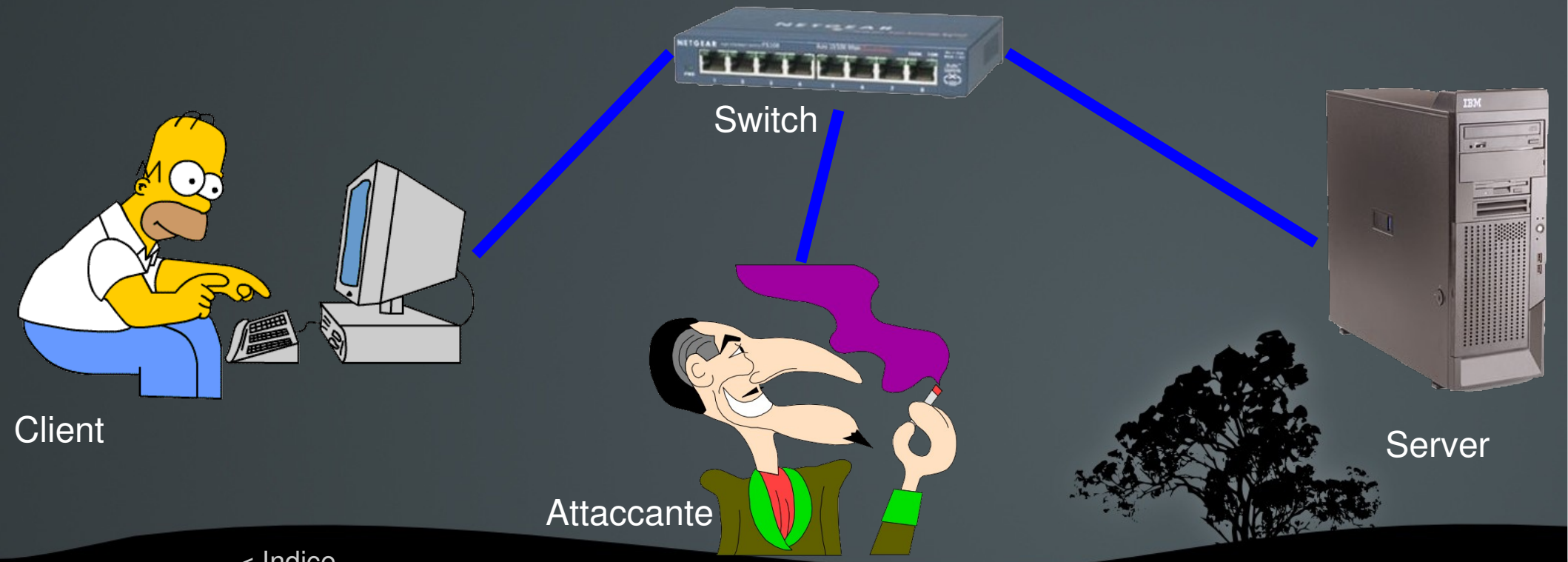
**Attaccante**

IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

## Port stealing

Durate questo tipo di attaco, si inonda periodicamente la rete con dei pacchetti ARP particolari: *Gratuitos ARP for 0.0.0.0*. Questi pacchetti vengono solitamente usati per verificare che un dato IP non sia ancora usato da nessun host. Questo procedimento permette all'attaccante di impadronirsi delle porte a cui sono collegate le vittime sullo switch di rete.
E' possibile propagare questi pacchetti ad altri switch, qualora la rete locale sia costituita da più switch, usando un indirizzo MAC di destinazione 'incerto', altrimenti ettercap usa quello dell'attaccante. Gli altri host non 'vedranno' questi pacchetti per il fatto che non sono destinati al loro indirizzo MAC.




< Indice

## Port stealing

L'attaccante inizia a mandare innumerevoli *Gratuitos ARP for 0.0.0.0.* Per gli indirizzi MAC del mittente vengono alternati i MAC delle vittime. Questo procedivento vince ogni singola gara di condizioni per ottenerne la corrispettiva porta dello switch a discapito del vero proprietario.

**Client**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa

**Gratuitos ARP for 0.0.0.0**
From: aa:aa:aa:aa:aa:aa
To: cc:cc:cc:cc:cc:cc

**Gratuitos ARP for 0.0.0.0**
From: bb:bb:bb:bb:bb:bb
To: cc:cc:cc:cc:cc:cc

**Gratuitos ARP for 0.0.0.0**
From: aa:aa:aa:aa:aa:aa
To: cc:cc:cc:cc:cc:cc

**Server**

IP: 192.168.1.2

MAC: bb:bb:bb:bb:bb:bb

**Attaccante**

IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

# Port stealing

**Client**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa

**Server**

IP: 192.168.1.2

MAC: bb:bb:bb:bb:bb:bb

**Ethernet**
From: aa:aa:aa:aa:aa:aa
To: bb:bb:bb:bb:bb:bb

**IP**
From: 192.168.1.12
To: 192.168.1.2
.....
.....

Dunque i pacchetti destinati a indirizzi MAC "rubati" saranno inoltrati dallo switch direttamente all'attaccante. Prima di poter inoltrare i pacchetti intercettati verso la vera destinazione, l'attaccante smette di mandare *Gratuitos ARP* a raffica e si assicura che la vera destinaione si 'riprenda' la sua porta dello switch.

**Attaccante**

IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

# Port stealing

**Client**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa

**Server**

IP: 192.168.1.2

MAC: bb:bb:bb:bb:bb:bb

L'attaccante fa una richiesta ARP per la vera destinazione, la risposta gli permetterà di essere sicuro che quest'ultima si è ripresa la sua porta dello switch.

```
ARP Request
From: cc:cc:cc:cc:cc:cc
To: bb:bb:bb:bb:bb:bb
Who has 192.168.1.2?
Tell 192.168.1.13.
```

```
ARP Reply
192.168.1.2 is at
bb:bb:bb:bb:bb:bb
```

**Attaccante**

IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

# Port stealing

**Client**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa

**Server**

IP: 192.168.1.2

MAC: bb:bb:bb:bb:bb:bb

**Ethernet**
From: aa:aa:aa:aa:aa:aa
To: bb:bb:bb:bb:bb:bb

**IP**
From: 192.168.1.12
To: 192.168.1.2
.....
.....

Ora l'attaccante puo procedere a inoltrare il pacchetto proveniente dalla prima vittima verso la vera destinazione.

**Attaccante**

IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

# Port stealing

L'attaccante riprende a inondare la rete con innumerevoli pacchetti *Gratuitos ARP for 0.0.0.0* come prima, in attesa di ricevere altri pacchetti da intercettare.

**Client**

IP: 192.168.1.12

MAC: aa:aa:aa:aa:aa:aa

**Gratuitos ARP for 0.0.0.0**
From: bb:bb:bb:bb:bb:bb
To: cc:cc:cc:cc:cc:cc

**Gratuitos ARP for 0.0.0.0**
From: aa:aa:aa:aa:aa:aa
To: cc:cc:cc:cc:cc:cc

**Gratuitos ARP for 0.0.0.0**
From: bb:bb:bb:bb:bb:bb
To: cc:cc:cc:cc:cc:cc

**Server**

IP: 192.168.1.2

MAC: bb:bb:bb:bb:bb:bb

**Attaccante**

IP: 192.168.1.13

MAC: cc:cc:cc:cc:cc:cc

## DNS spoofing

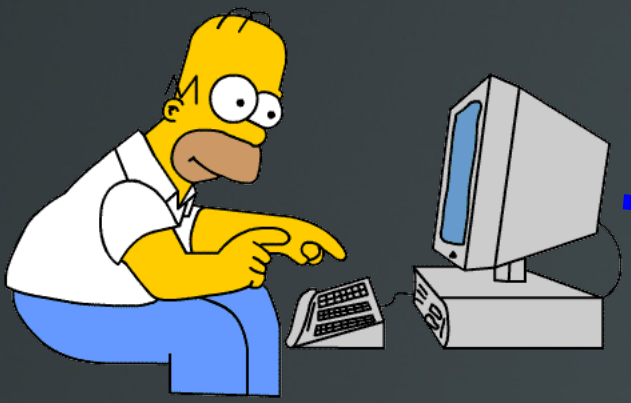

L'attaccante riesce quindi a sniffare il traffico delle vittime con i metodi precedenti, (a seconda del livello di protezione della rete, l'attaccante avrà scelto il metodo più efficace). Ettercap mette a dispozione molte funzionalità e plugin. Ad esempio con l'opzione `-e` verranno visualizzati i pacchetti contenenti un dato testo come "*carta di credito*". Il plugin *remote_browser* permette all'attaccante di visualizzare nel proprio browser 'in diretta' tutte le pagine web che sta visitando la vittima. Mentre il plugin *dns_spoof* da la possibilità all'attaccante di fare DNS spoofing, sostituendosi al vero DNS e dirottando così le vittime su falsi server con lo scopo di rubare le chiavi di accesso.

## DNS spoofing


In questo esempio il server DNS usato dalla LAN è incorporato nel gateway, se fosse esterno, l'attaccante si metterà comunque in mezzo tra GW e vittima visto che le risposte del DNS passeranno comunque per il gateway.

Internet

Gateway / Server DNS



Vittima

Switch

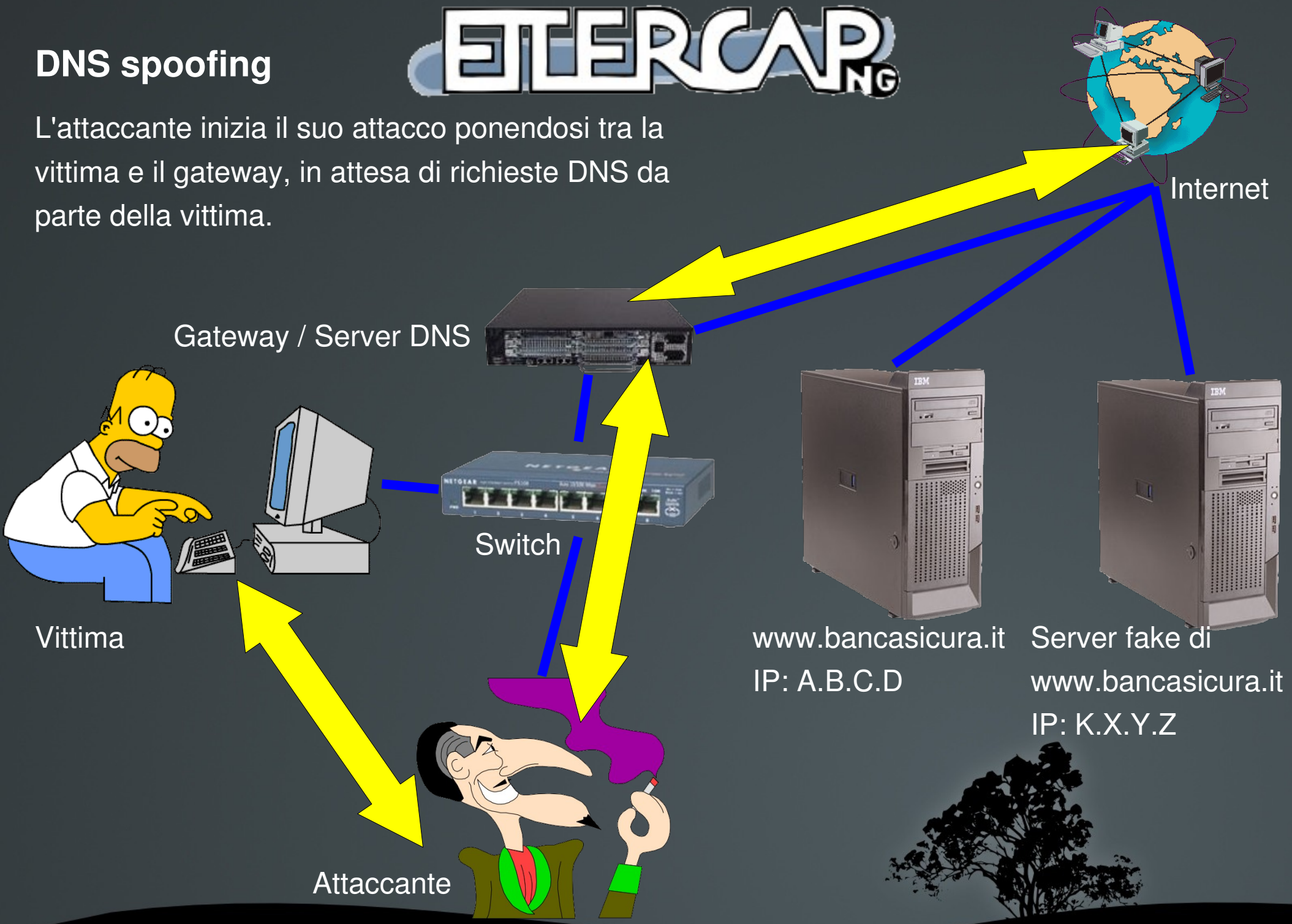
www.bancasicura.it
IP: A.B.C.D

Server fake di
www.bancasicura.it
IP: K.X.Y.Z

Attaccante

## DNS spoofing

L'attaccante inizia il suo attacco ponendosi tra la vittima e il gateway, in attesa di richieste DNS da parte della vittima.

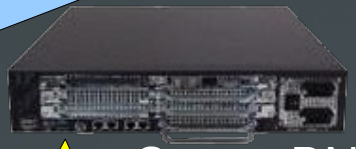


# DNS spoofing

La vittima si collega al sito della propria banca facendo una query DNS per tale indirizzo.

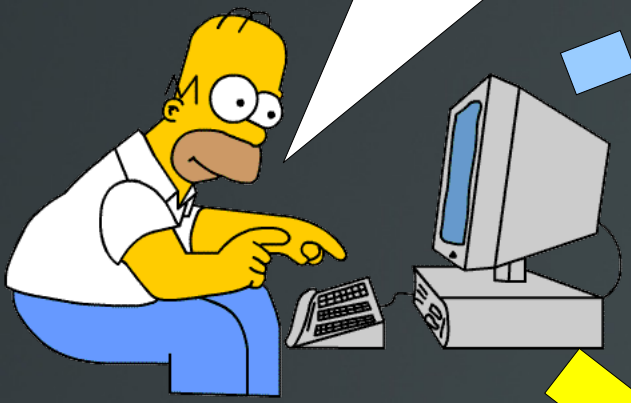
www.bancasicura.it
IP: A.B.C.D

www.bancasicura.it



Server DNS
IP: 192.168.1.1

**DNS query for www.bancasicura.it**
From: 192.168.1.12
To: 192.168.1.1



Vittima
IP: 192.168.1.12

Server fake di
www.bancasicura.it
IP: K.X.Y.Z

**DNS query for www.bancasicura.it**
From: 192.168.1.12
To: 192.168.1.1

Un'altro conto da ripulire!

La query DNS viene ovviamente intercettata dall'attaccante.

Attaccante

# DNS spoofing

L'attaccante risponde subito spacciandosi per il server DNS e fornendo l'IP di un serfer fake.

Vittima
IP: 192.168.1.12

Server DNS
IP: 192.168.1.1

www.bancasicura.it
IP: A.B.C.D

Server fake di
www.bancasicura.it
IP: K.X.Y.Z

```
DNS answer
From: 192.168.1.1
To: 192.168.1.12
www.bancasicura.it is at: K.X.Y.Z
```

Intanto il vero DNS locale, non sapendone la risposta, inoltra la query DNS all'esterno verso altri server DNS.

Attaccante

# DNS spoofing

La vittima inizia a scambiare traffico con il server fake.

Vittima
IP: 192.168.1.12

```
IP
From: 192.168.1.12
To: K.X.Y.Z
  HTTP
  GET index.html
  ......
```

Gateway

www.bancasicura.it
IP: A.B.C.D

Server fake di
www.bancasicura.it
IP: K.X.Y.Z

Nella barra dell'indirizzo nel browser della vittima resterà scritto l'indirizzo corretto, pur essendo collegati su di un server falso. Anche la cache DNS della vittima rimane inquinata con queste false informazioni.

Attaccante

# DNS spoofing

Intanto arriva anche la risposta del vero server DNS, l'attaccante potrebbe anche evitare di inoltrare queste informazioni, ma ormai è troppo tardi: la cache DNS della vittima ha già un indirizzo per tale richiesta e quindi non prenderà più in considerazione la vera risposta.

www.bancasicura.it
IP: A.B.C.D

Server DNS
IP: 192.168.1.1

Vittima
IP: 192.168.1.12

```
DNS answer
From: 192.168.1.1
To: 192.168.1.12
www.bancasicura.it is at: A.B.C.D
```

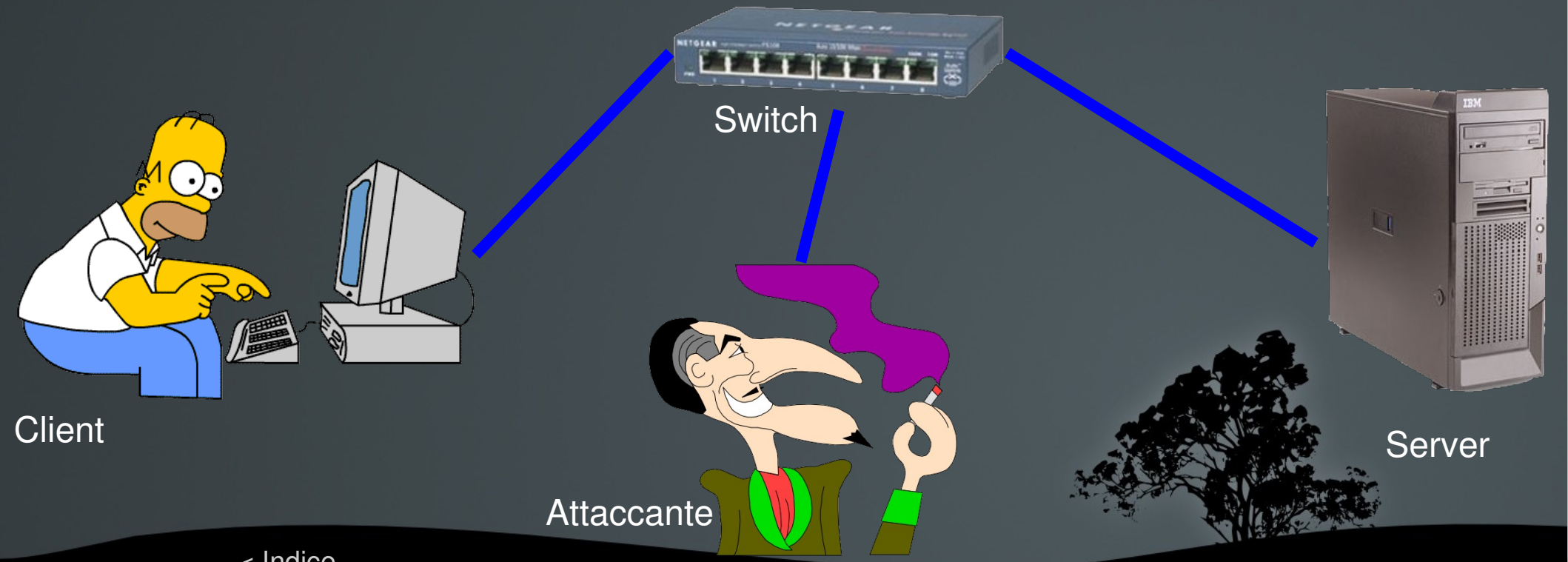
Server fake di
www.bancasicura.it
IP: K.X.Y.Z

```
DNS Cache
www.polito.it -> 130.192.73.1
www.bancasicura.it -> K.X.Y.Z
www.google.com -> 209.85.137.104
...
```

Attaccante

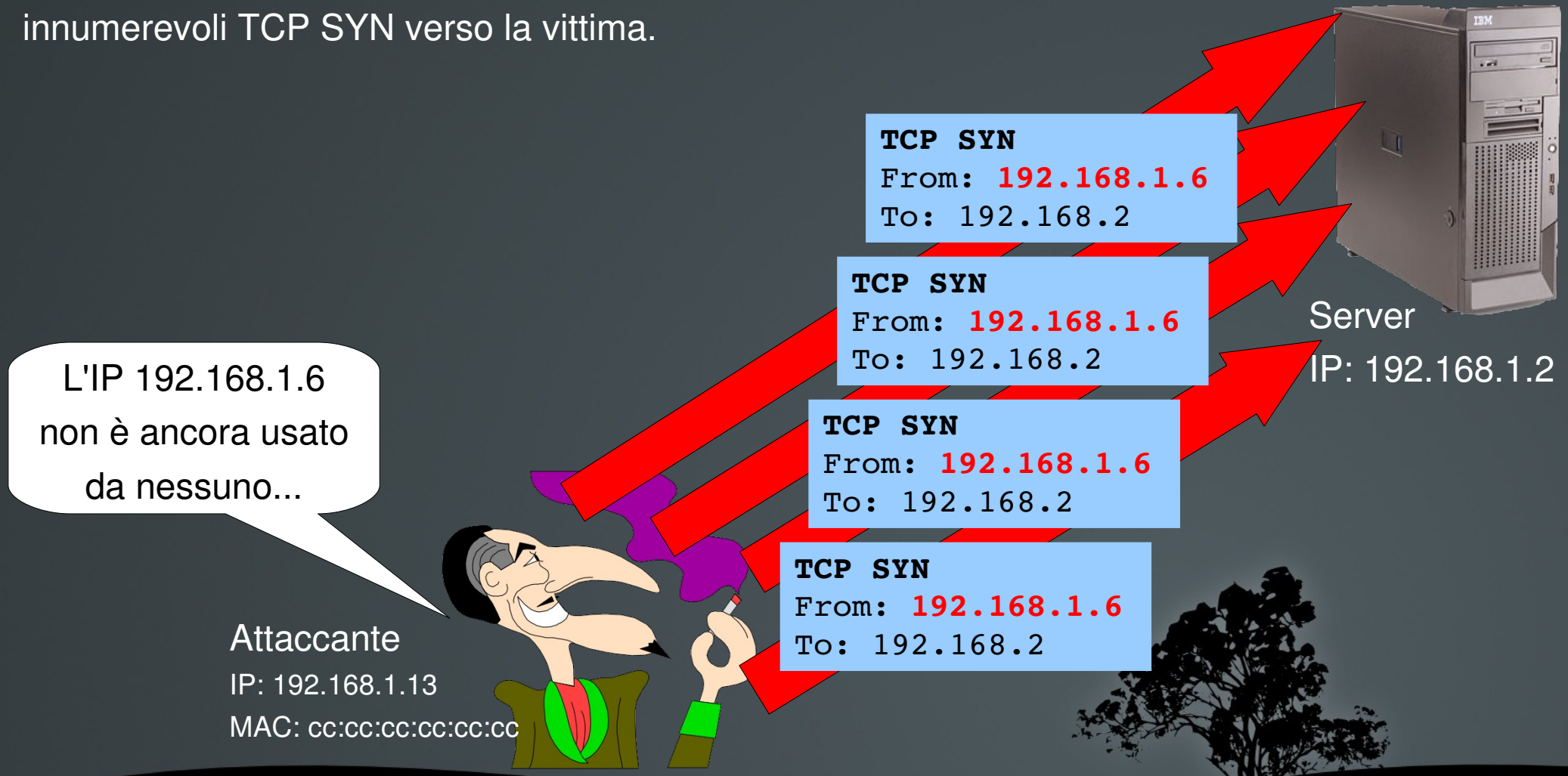
## Denial of Service

Appena instaurato un attacco MITM l'attaccante può anche rendere indisponibile un servizio usando *tcpkill* per chiudere tutte le connessioni dei client che tentano di accedere a tale servizio. Inoltre vi sono alcuni plugin di ettercap specifici per fare attacchi DoS, come per esempio: *dos_attack* che mette in atto il calssico attacco DoS basato sull'invio di pacchetti TCP SYN. Per fare questo tipo di attacco l'attaccante ha bisogno di un IP non ancora usato sulla rete da usare per madare i pacchetti TCP SYN. In ettercap troviamo uno specifico plugin per fare anche questo: *find_ip* che facendo una scansione della rete con le solite richieste ARP permette di sapere quali IP sono liberi. Fatto ciò l'attacante è pronto a iniziare l'attacco.



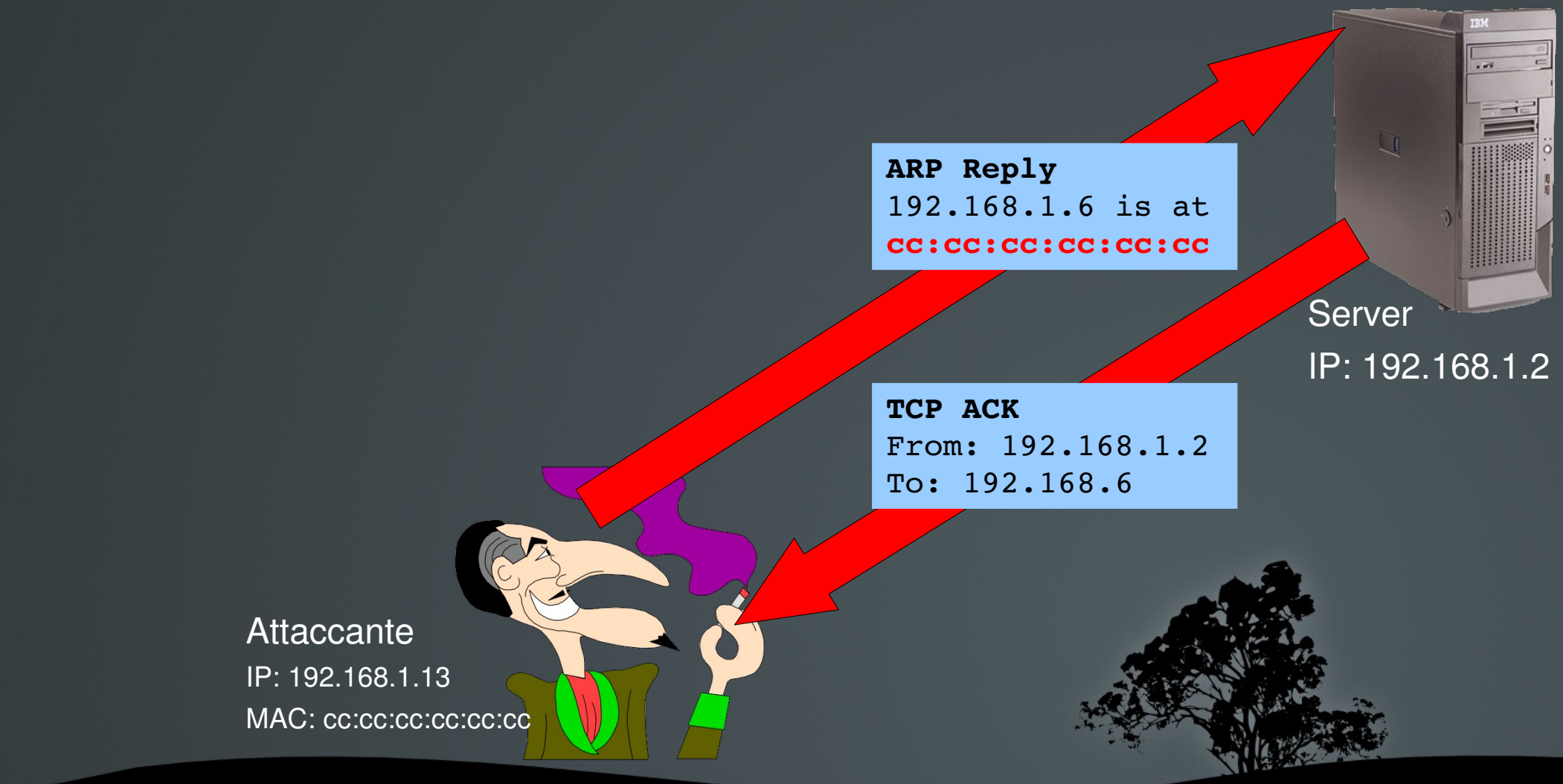
## SYN attack

I pacchetti TCP SYN vengono usati per aprire una nuova connessione TCP, quest'attacco consiste nel fare aprire al server più connessioni TCP possibili fino a saturagli i buffer, in questo modo non sarà più in grado di accettare altre connessioni, nemmeno quelle dei veri client. L'attaccante trova un IP libero da usare come IP fantasma per iniziare a mandare innumerevoli TCP SYN verso la vittima.



## SYN attack

Per far sì che la vittima apra una nuova connessione per ogni TCP SYN ricevuto, l'attaccante ha bisogno di intercettare le risposte ACK con il numero di sequenza corretto. Per questo motivo l'attaccante manda ogni tanto anche qualche falsa risposta ARP, per poter intercettare gli ACK da parte della vittima destinati all'indirizzo fantasma.



## SYN attack

Nel giro di pochi istanti la vittima non riesce più ad aprire altre connessioni e risulta quindi indisponibile a tutta la rete.


Client
IP: 192.168.1.12

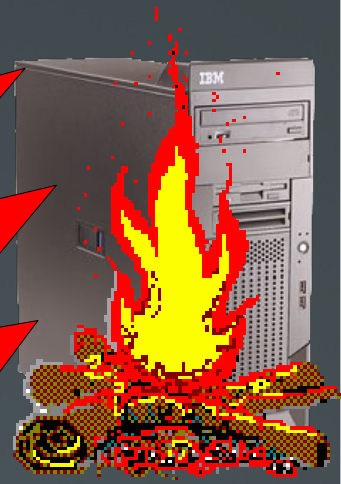
```
IP
From: 192.168.1.12
To: 192.168.1.2
.........
.........
```

```
TCP SYN
From: 192.168.1.6
To: 192.168.2
```

```
TCP SYN
From: 192.168.1.6
To: 192.168.2
```

```
TCP SYN
From: 192.168.1.6
To: 192.168.2
```

```
TCP SYN
From: 192.168.1.6
To: 192.168.2
```



Server
IP: 192.168.1.2

Attaccante
IP: 192.168.1.13
MAC: cc:cc:cc:cc:cc:cc

< Indice