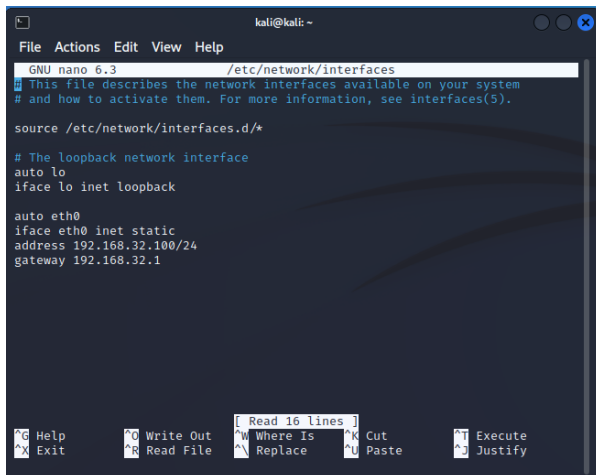
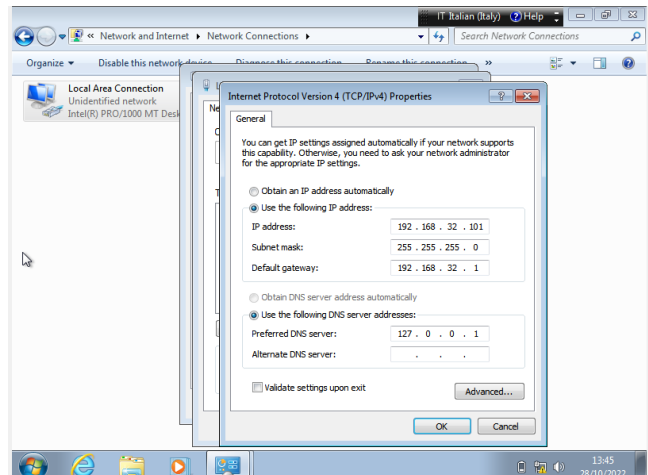


## SIMULAZIONE ARCHITETTURA CLIENT SERVER E INTERCETTAZIONE TRAFFICO CON WIRESHARK

Come prima cosa, modifichiamo nuovamente l' IP degli host in questione ( Kali Linux e Windows 7) come abbiamo fatto in una delle esercitazioni precedenti, assegnando a Kali Linux il 192.168.32.100 e a Windows 7 il 192.168.32.101:

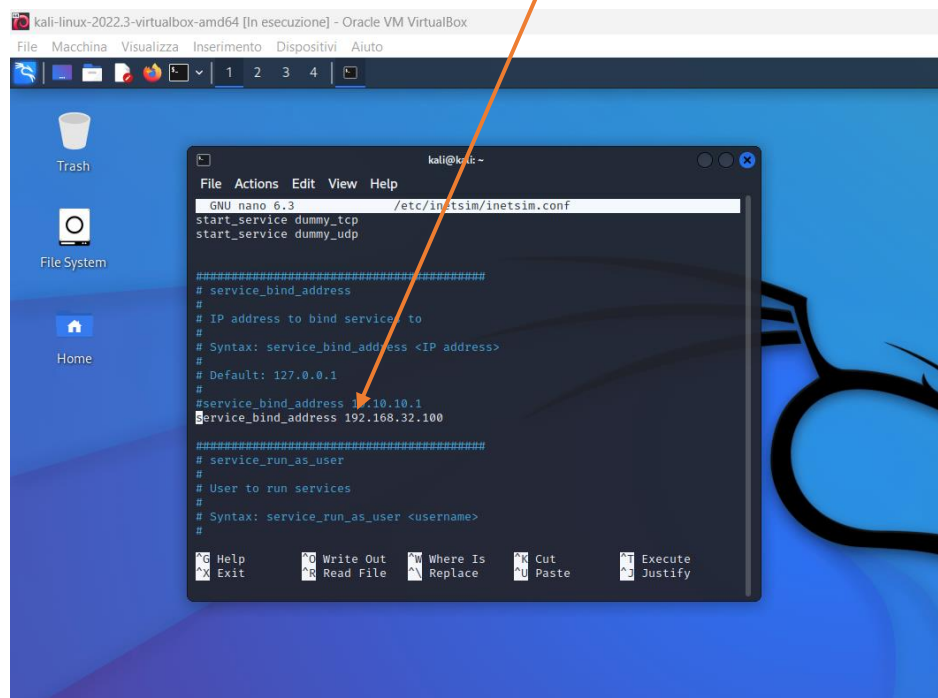


```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 6.3 /etc/network/interfaces  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
auto eth0  
iface eth0 inet static  
address 192.168.32.100/24  
gateway 192.168.32.1
```

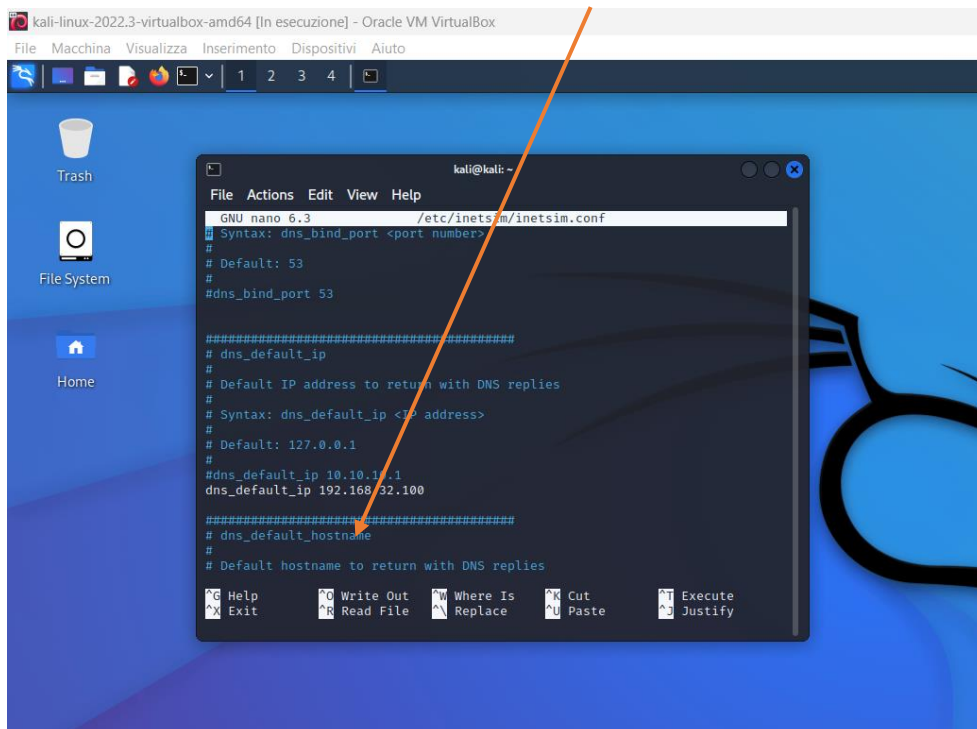


Dobbiamo poi impostare la macchina con Kali Linux in modo da rispondere anche come `epicode.internal`; andiamo quindi a settare il tutto passando per il tool Inetsim inserendo nel prompt la stringa “`sudo nano /etc/inetsim/inetsim.config`”, possiamo configurare i server di cui abbiamo bisogno:

Nel campo del server bind l'indirizzo IP di Kali

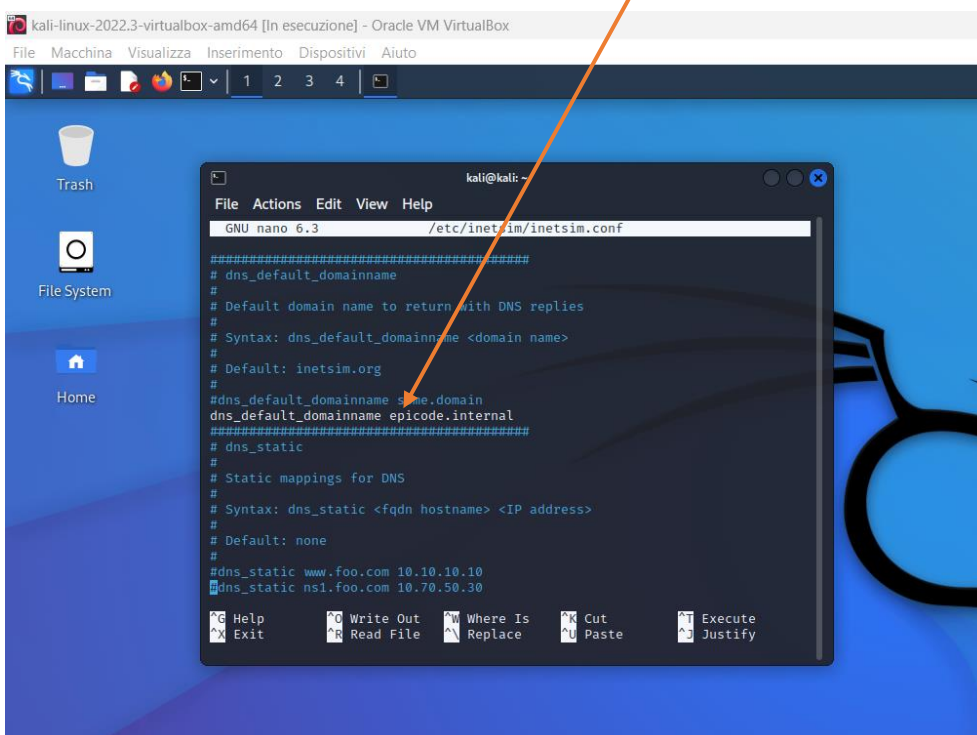


Nel campo dns\_default\_ip aggiungiamo sempre l' IP di Kali come da figura



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 6.3 /etc/inetsim/inetsim.conf  
# Syntax: dns_bind_port <port number>  
#  
# Default: 53  
#dns_bind_port 53  
  
#####  
# dns_default_ip  
#  
# Default IP address to return with DNS replies  
# Syntax: dns_default_ip <IP address>  
#  
# Default: 127.0.0.1  
#dns_default_ip 10.10.1.1  
dns_default_ip 192.168.32.100  
  
#####  
# dns_default_hostname  
#  
# Default hostname to return with DNS replies  
  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify
```

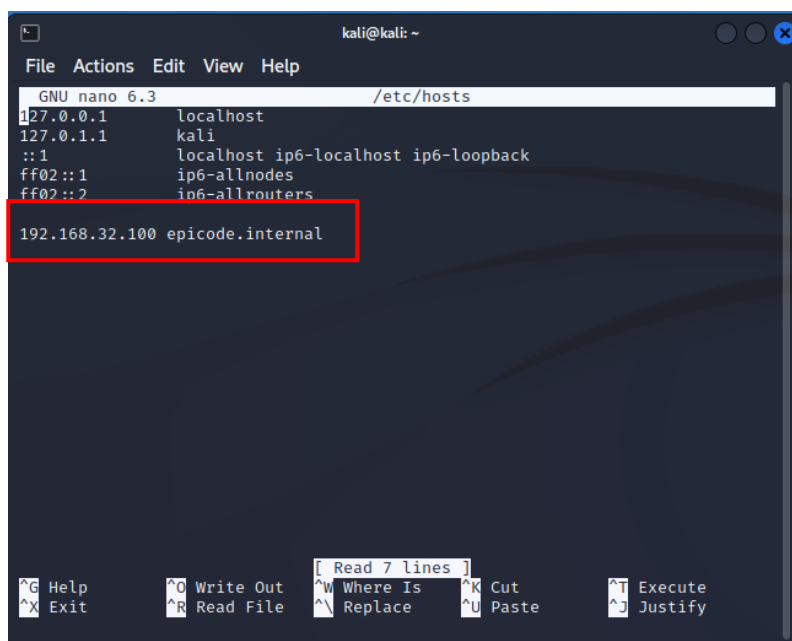
Nel campo dns\_default\_domainname il nome di dominio



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 6.3 /etc/inetsim/inetsim.conf  
  
#####  
# dns_default_domainname  
#  
# Default domain name to return with DNS replies  
# Syntax: dns_default_domainname <domain name>  
#  
# Default: inetsim.org  
#dns_default_domainname some.domain  
dns_default_domainname epicode.internal  
#####  
# dns_static  
#  
# Static mappings for DNS  
# Syntax: dns_static <fqdn hostname> <IP address>  
#  
# Default: none  
#  
#dns_static www.foo.com 10.10.10.10  
dns_static ns1.foo.com 10.70.50.30  
  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify
```

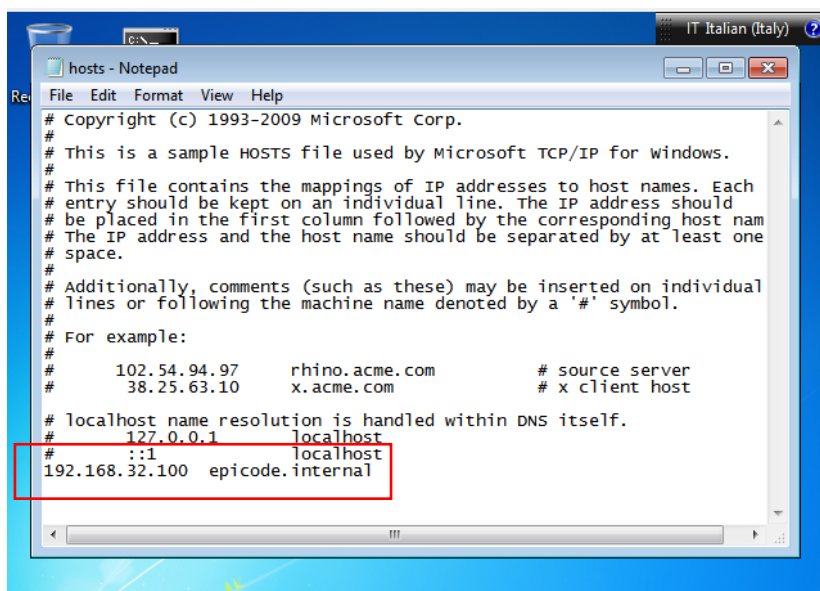
Andiamo poi ad associare il nome di dominio epicode.internal all'indirizzo IP 192.168.32.100.

Su Kali, sempre nel prompt, scriviamo “sudo nano /etc/hosts” per aprire la directory dove andremo ad associare il tutto:



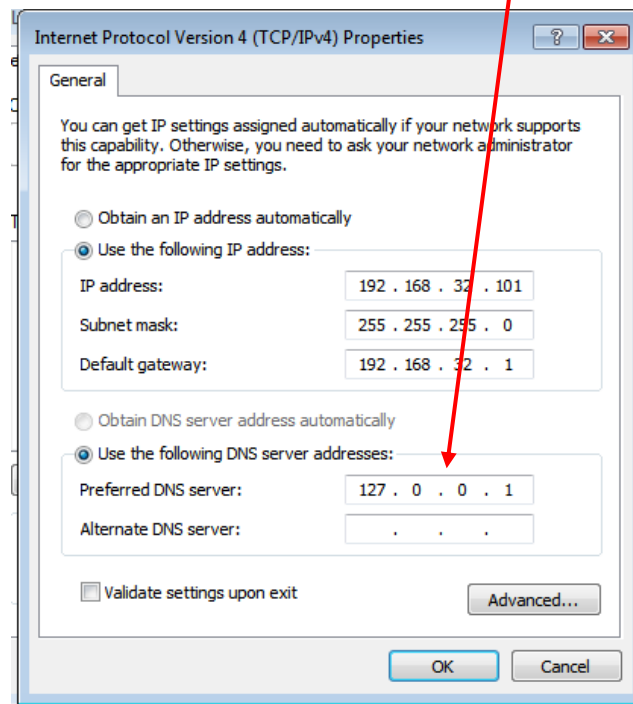
```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 6.3 /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 kali  
::1 localhost ip6-localhost ip6-loopback  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
192.168.32.100 epicode.internal  
  
[ Read 7 lines ]  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify
```

Su Windows invece runniamo il blocco note come amministratore ed apriamo il file “hosts” che si trova su C:/Windows/System32/drivers/etc ed inseriamo IP e nome di dominio:

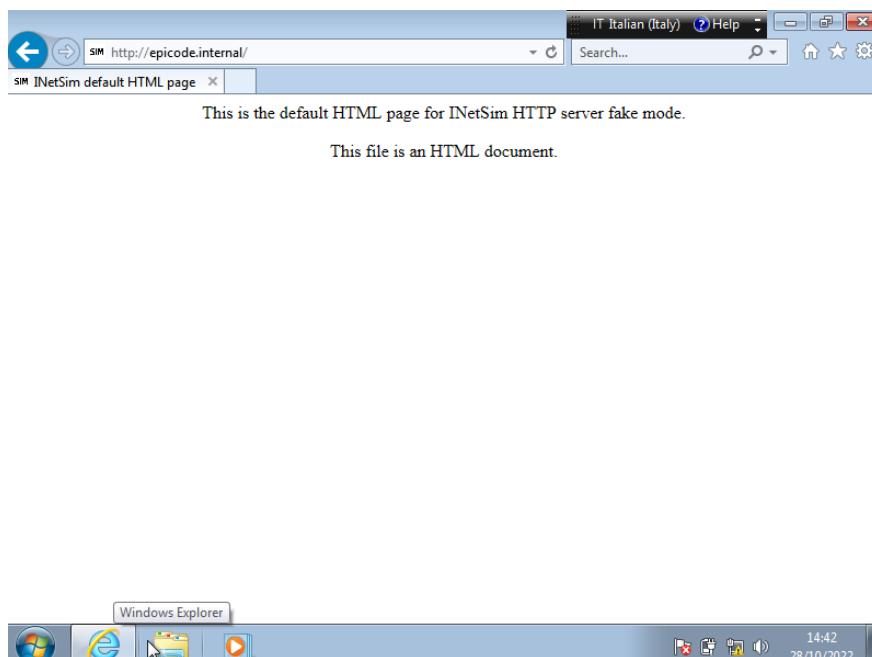


```
hosts - Notepad  
File Edit Format View Help  
# Copyright (c) 1993-2009 Microsoft Corp.  
#  
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.  
#  
# This file contains the mappings of IP addresses to host names. Each  
# entry should be kept on an individual line. The IP address should  
# be placed in the first column followed by the corresponding host nam  
# The IP address and the host name should be separated by at least one  
# space.  
#  
# Additionally, comments (such as these) may be inserted on individual  
# lines or following the machine name denoted by a '#' symbol.  
#  
# For example:  
#  
#       102.54.94.97       rhino.acme.com          # source server  
#       38.25.63.10       x.acme.com              # x client host  
#  
# localhost name resolution is handled within DNS itself.  
#       127.0.0.1         localhost  
#       ::1              localhost  
192.168.32.100 epicode.internal
```

Come ultima cosa, inseriamo l'IP del server DNS (lo troviamo su Inetsim) su Windows:



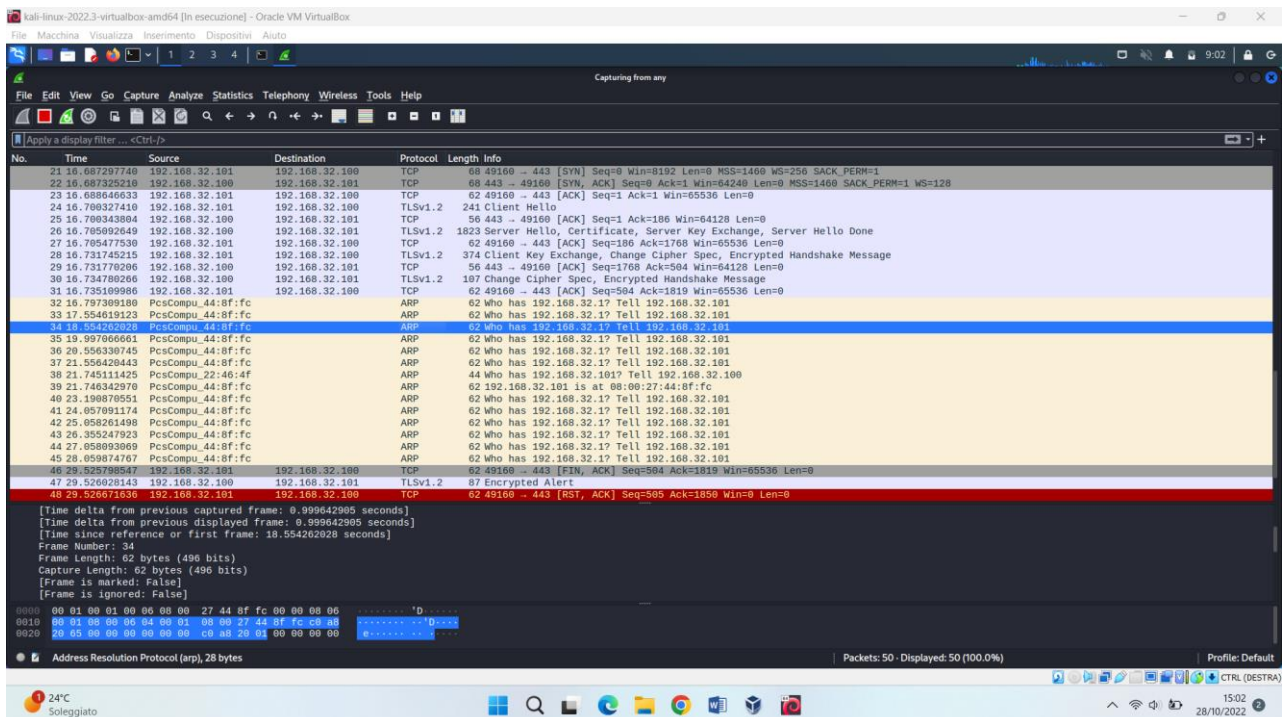
Su Kali, lanciamo Inetsim per simulare l'ambiente di rete (sudo inetsim), torniamo su Windows e, aprendo il Web browser e cercando epicode.internal, ci manda sulla pagina richiesta:



Qui possiamo vedere il MAC address del mittente ed il MAC del ricevente:

```
Sender MAC address: PcsCompu_22:46:4f (08:00:27:22:46:4f)
Sender IP address: 192.168.32.100
Target MAC address: PcsCompu_44:8f:fc (08:00:27:44:8f:fc)
Target IP address: 192.168.32.101
```

Aprendo `epicode.internal` in `https`, possiamo vedere come per prima cosa il TCP invia la richiesta di comunicazione (SYN, SYN-ACK, ACK) attraverso la porta 443; il protocollo usato per stabilire la connessione è il TLS (transport layer security) che garantisce una connessione sicura e crittografata tra client e server; come possiamo vedere nel traffico intercettato da Wireshark c'è prima uno scambio di chiavi (key exchange) seguito poi da un continuo encrypted handshake message



Mentre il traffico catturato usando il protocollo HTTP ci mostra intanto che la porta è cambiata, non è più la 443 ma la 80, inoltre possiamo vedere come la connessione sia avvenuta in modo molto più facile in quanto è bastato inviare una richiesta (GET HTTP) e, di contro, una risposta ( HTTP 200 OK), in tutto ciò, lo scambio di dati è avvenuto in plain text, cioè in chiaro.

Time	Source	Destination	Protocol	Length	Info
22.59.885824856	PcsCompu_22:46:4f		ARP	44	192.168.32.100 is at 08:00:27:22:46:4f
23.59.886263343	192.168.32.101	192.168.32.100	TCP	64	49159 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
24.59.886288415	192.168.32.100	192.168.32.101	TCP	64	80 → 49159 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1
25.59.886731247	192.168.32.101	192.168.32.100	TCP	62	49159 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
26.59.886731353	192.168.32.101	192.168.32.100	HTTP	307	GET / HTTP/1.1
27.59.886767998	192.168.32.100	192.168.32.101	TCP	56	80 → 49159 [ACK] Seq=1 Ack=252 Win=63989 Len=0
28.59.907893680	192.168.32.100	192.168.32.101	TCP	206	80 → 49159 [PSH, ACK] Seq=1 Ack=252 Win=63989 Len=150 [TCP segment of a reassembled PDU]
29.59.908252054	192.168.32.101	192.168.32.100	TCP	62	49159 → 80 [ACK] Seq=252 Ack=151 Win=64090 Len=0
30.59.908262904	192.168.32.101	192.168.32.101	HTTP	314	HTTP/1.1 200 OK (text/html)
31.59.908566946	192.168.32.101	192.168.32.100	TCP	62	49159 → 80 [ACK] Seq=252 Ack=409 Win=63832 Len=0
32.59.910219613	192.168.32.100	192.168.32.101	TCP	56	80 → 49159 [FIN, ACK] Seq=409 Ack=252 Win=63989 Len=0
33.59.910465205	192.168.32.101	192.168.32.100	TCP	62	49159 → 80 [ACK] Seq=252 Ack=410 Win=63832 Len=0

[Coloring Rule Name: ARP]  
 [Coloring Rule String: arp]  
 Linux cooked capture v1

Address Resolution Protocol (reply)

Hardware type: Ethernet (1)  
 Protocol type: IPv4 (0x0800)  
 Hardware size: 6  
 Protocol size: 4  
 Opcode: reply (2)  
 Sender MAC address: PcsCompu\_22:46:4f (08:00:27:22:46:4f)  
 Sender IP address: 192.168.32.100  
 Target MAC address: PcsCompu\_44:8f:fc (08:00:27:44:8f:fc)  
 Target IP address: 192.168.32.101