

Progetto W1

Simulazione architettura client server

Traccia

Requisiti e servizi:

- Kali Linux IP 192.168.32.100
- Windows 7 IP 192.168.32.101
- HTTPS server: attivo
- Servizio DNS per risoluzione nomi di dominio: attivo

Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.32.100.

Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.

Ripetere l'esercizio, sostituendo il server HTTPS, con un server HTTP. Si intercetti nuovamente il traffico, evidenziando le differenze tra il traffico appena catturato in HTTP ed il traffico precedente in HTTPS.

Spiegare, motivandole, le principali differenze.

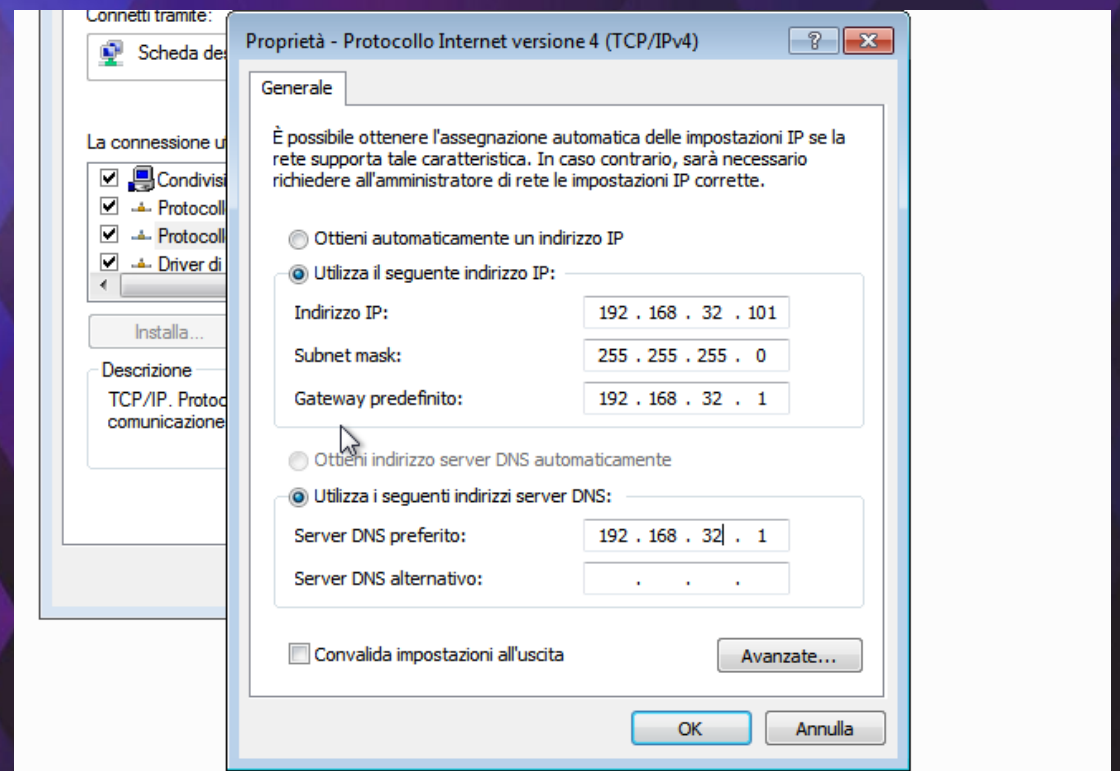
Impostazione delle macchine sulla stessa rete

Affinché le macchine virtuali fossero sulla stessa rete, è stato necessario configurare gli indirizzi IP richiesti dalla consegna.

Per fare ciò, è stato utilizzato il comando: `sudo nano /etc/network/interfaces` sulla macchina Kali Linux, mentre sono state regolate le impostazioni di rete sulla macchina Windows 7.

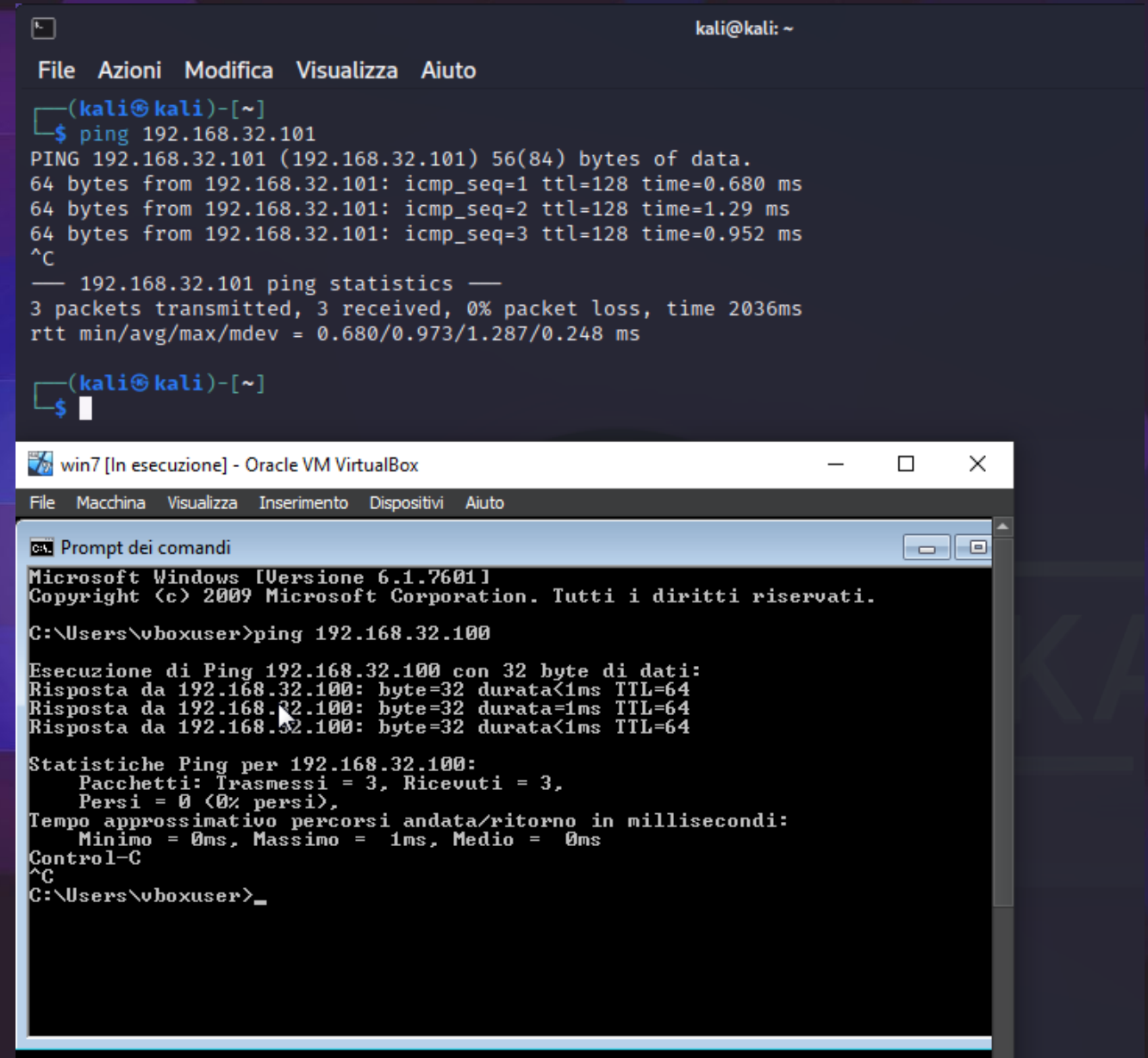
Successivamente, è stata verificata la connettività avviando pacchetti ICMP tramite il comando `ping` da una macchina all'altra utilizzando i terminali.

```
kali@kali: ~  
File Azioni Modifica Visualizza Aiuto  
GNU nano 7.2 /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  
netmask 255.255.255.0  
network 192.168.32.0  
broadcast 192.168.32.255  
gateway 192.168.32.1
```



Invio pacchetti ICMP

Nella figura è illustrato lo scambio di pacchetti tra la macchina Kali Linux e l'indirizzo della macchina Windows 7, e viceversa. Poiché sono stati inviati e ricevuti 3 pacchetti, si può confermare che le due macchine sono state messe in comunicazione.



The image shows two overlapping windows. The top window is a Kali Linux terminal with the title 'kali@kali: ~'. It displays the output of a 'ping 192.168.32.101' command. The output shows three successful ping requests with varying response times (0.680 ms, 1.29 ms, 0.952 ms) and a summary: '3 packets transmitted, 3 received, 0% packet loss, time 2036ms rtt min/avg/max/mdev = 0.680/0.973/1.287/0.248 ms'. The bottom window is a Windows 7 VM titled 'win7 [In esecuzione] - Oracle VM VirtualBox'. It shows a Windows command prompt with the command 'ping 192.168.32.100'. The output shows three successful ping requests with a response time of less than 1ms and a TTL of 64. A summary of ping statistics is also displayed: 'Statistiche Ping per 192.168.32.100: Pacchetti: Trasmessi = 3, Ricevuti = 3, Persi = 0 (0% persi), Tempo approssimativo percorsi andata/ritorno in millisecondi: Minimo = 0ms, Massimo = 1ms, Medio = 0ms'.

```
kali@kali: ~  
File Azioni Modifica Visualizza Aiuto  
(kali@kali)-[~]  
$ ping 192.168.32.101  
PING 192.168.32.101 (192.168.32.101) 56(84) bytes of data.  
64 bytes from 192.168.32.101: icmp_seq=1 ttl=128 time=0.680 ms  
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=1.29 ms  
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.952 ms  
^C  
— 192.168.32.101 ping statistics —  
3 packets transmitted, 3 received, 0% packet loss, time 2036ms  
rtt min/avg/max/mdev = 0.680/0.973/1.287/0.248 ms  
(kali@kali)-[~]  
$  
  
win7 [In esecuzione] - Oracle VM VirtualBox  
File Macchina Visualizza Inserimento Dispositivi Aiuto  
c:\ Prompt dei comandi  
Microsoft Windows [Versione 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. Tutti i diritti riservati.  
C:\Users\vboxuser>ping 192.168.32.100  
  
Esecuzione di Ping 192.168.32.100 con 32 byte di dati:  
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64  
Risposta da 192.168.32.100: byte=32 durata=1ms TTL=64  
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64  
  
Statistiche Ping per 192.168.32.100:  
Pacchetti: Trasmessi = 3, Ricevuti = 3,  
Persi = 0 (0% persi),  
Tempo approssimativo percorsi andata/ritorno in millisecondi:  
Minimo = 0ms, Massimo = 1ms, Medio = 0ms  
Control-C  
^C  
C:\Users\vboxuser>_
```


Configurazione dello strumento inetsim

Per simulare un server direttamente dalla macchina Kali Linux e metterlo in comunicazione con un'altra macchina, è stato necessario configurare inetsim tramite il comando `sudo nano /etc/inetsim/inetsim.conf`.

Poiché solo i servizi https e dns sono necessari, è stato aggiunto un `#` davanti agli altri per renderli commenti e quindi disattivarli.

Inoltre, la traccia richiedeva di associare il dominio epicode.internal all'indirizzo IP della macchina Kali tramite il servizio dns.

File Azioni Modifica Visualizza Aiuto

GNU nano 7.2

/etc/inetsim/inetsim.conf *

```
#####
#
# INetSim configuration file
#
#####

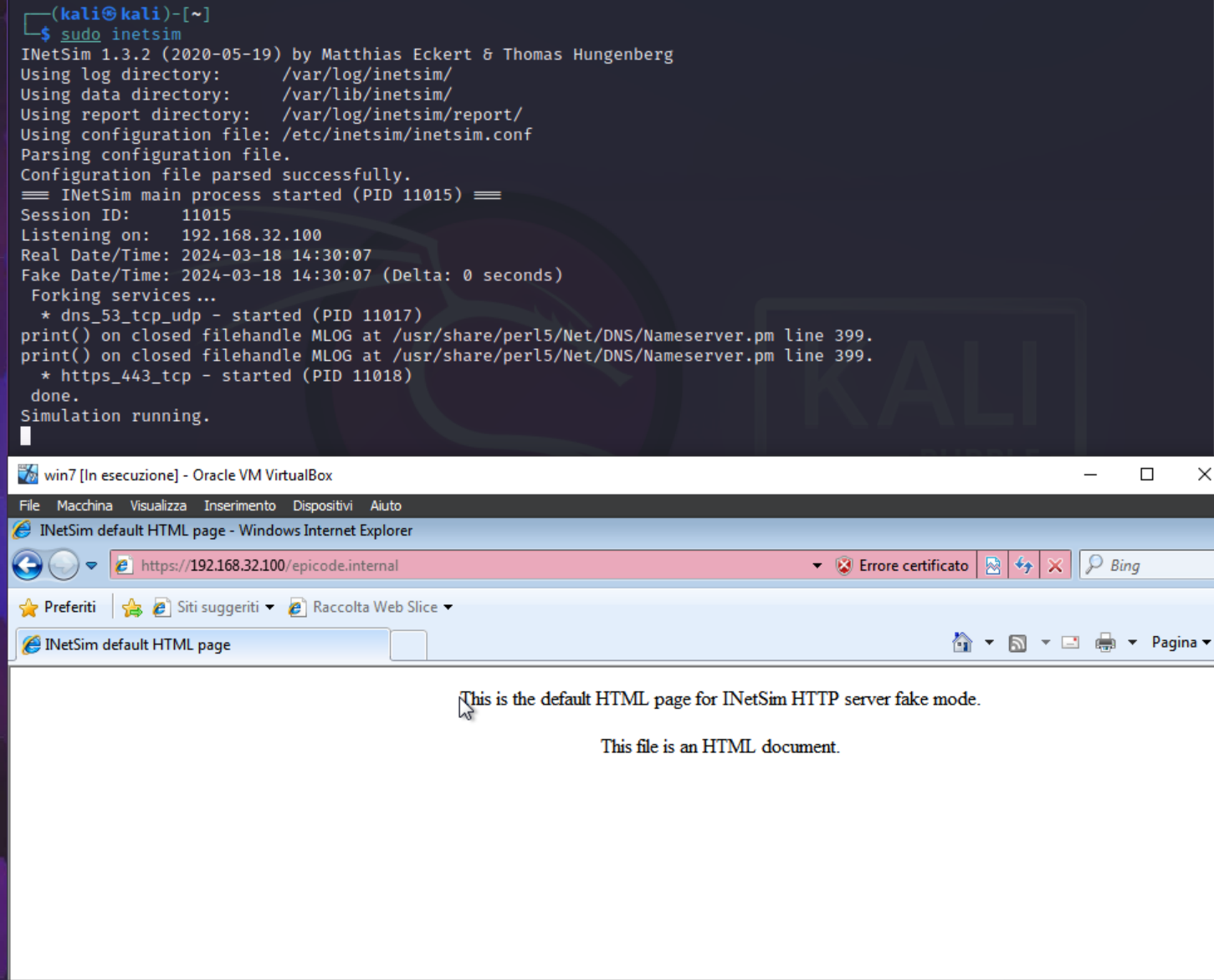
#####
# Main configuration
#####

#####
# start_service
#
# The services to start
#
# Syntax: start_service <service name>
#
# Default: none
#
# Available service names are:
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,
# time_udp, daytime_tcp, daytime_udp, echo_tcp,
# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#
start_service dns
#start_service http
start_service https
#start_service smtp
#start_service smtps
#start_service pop3

#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
#
dns_static epicode.internal 192.168.32.100
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
```

Avvio del servizio inetsim

Dopo aver avviato il servizio inetsim, è stato possibile connettersi alla pagina digitando `epicode.internal` nella barra di ricerca del browser su Windows 7.



Cattura pacchetti con Wireshark

Con l'ausilio dello strumento Wireshark, capace di intercettare la comunicazione e la trasmissione di pacchetti da una macchina all'altra, è stato possibile recuperare una vasta gamma di informazioni, tra cui gli indirizzi MAC di destinazione e di origine. Inoltre, è evidente che la comunicazione tramite HTTPS è sicura. Questo perché il canale è cifrato grazie al protocollo TLS.

30	9.289133766	192.168.32.101	192.168.32.255	NBNS	92 Name query NB WPAD<00>
31	10.039073360	192.168.32.101	192.168.32.255	NBNS	92 Name query NB WPAD<00>
32	10.788665062	192.168.32.101	192.168.32.255	NBNS	92 Name query NB WPAD<00>
33	11.542945766	PcsCompu_97:1c:91	Broadcast	ARP	60 Who has 192.168.32.1? Tell 192.168.32.
34	12.320152607	PcsCompu_97:1c:91	Broadcast	ARP	60 Who has 192.168.32.1? Tell 192.168.32.
35	13.321700131	PcsCompu_97:1c:91	Broadcast	ARP	60 Who has 192.168.32.1? Tell 192.168.32.
36	15.088325872	192.168.32.101	192.168.32.100	TLSv1	395 Application Data
37	15.102872130	192.168.32.100	192.168.32.101	TLSv1	235 Application Data
38	15.105208135	192.168.32.100	192.168.32.101	TLSv1	384 Application Data, Encrypted Alert
39	15.105587473	192.168.32.101	192.168.32.100	TCP	60 49170 → 443 [ACK] Seq=612 Ack=1891 Win
40	15.105753973	192.168.32.101	192.168.32.100	TCP	60 49170 → 443 [FIN, ACK] Seq=612 Ack=189
41	15.105763406	192.168.32.100	192.168.32.101	TCP	54 443 → 49170 [ACK] Seq=1891 Ack=613 Win
42	15.542650873	PcsCompu_97:1c:91	Broadcast	ARP	60 Who has 192.168.32.1? Tell 192.168.32.
43	16.322295601	PcsCompu_97:1c:91	Broadcast	ARP	60 Who has 192.168.32.1? Tell 192.168.32.
44	17.324099629	PcsCompu_97:1c:91	Broadcast	ARP	60 Who has 192.168.32.1? Tell 192.168.32.
45	19.544048613	PcsCompu_97:1c:91	Broadcast	ARP	60 Who has 192.168.32.1? Tell 192.168.32.
46	20.324725354	PcsCompu_97:1c:91	Broadcast	ARP	60 Who has 192.168.32.1? Tell 192.168.32.
47	21.324833945	PcsCompu_97:1c:91	Broadcast	ARP	60 Who has 192.168.32.1? Tell 192.168.32.

▶ Frame 38: 384 bytes on wire (3072 bits), 384 bytes captured (3072 bits) on	0000	08 00 27	97 1c 91 08 00	27 ff cc
▶ Ethernet II, Src: PcsCompu_ff:cc:77 (08:00:27:ff:cc:77), Dst: PcsCompu_97:1	0010	01 72 bf 4b 40 00	40 06 b8 20 c0	
▶ Destination: PcsCompu_97:1c:91 (08:00:27:97:1c:91)	0020	20 65 01 bb c0 12	43 d9 19 fb 00	
▶ Source: PcsCompu_ff:cc:77 (08:00:27:ff:cc:77)	0030	01 f5 c3 7e 00 00	17 03 01 01 20	
Type: IPv4 (0x0800)	0040	70 b7 e1 e2 19 39	79 63 a4 9e 2a	
▶ Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101	0050	07 95 be e7 88 8a	9b a2 95 24 30	
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 49170, Seq: 1560, A	0060	41 97 ec ff 3d 2b	c6 f2 e1 6b 40	
▶ Transport Layer Security	0070	b0 b1 a0 1a 8a 88	c8 4b b7 31 59	
▶ TLSv1 Record Layer: Application Data Protocol: Hypertext Transfer Protoc	0080	15 a1 14 a9 9d 60	57 9a 7b a2 ea	
▶ TLSv1 Record Layer: Encrypted Alert	0090	c6 03 e5 84 62 25	4c 5d d6 c1 c0	
	00a0	51 af 95 28 c9 da	1b a7 d7 79 d0	
	00b0	03 70 31 eb 32 b4	24 dd c3 80 b0	
	00c0	8a 04 70 39 3e 8a	8f 06 22 70 10	
	00d0	ae bd b4 c9 1d 14	85 bd 44 ac c0	

Utilizzo di HTTP

Ripetendo il medesimo procedimento ma utilizzando il servizio HTTP al posto di HTTPS, Wireshark è stato in grado di intercettare la comunicazione, la quale, essendo non criptata, non è considerata sicura.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.32.101	192.168.32.100	TCP	66	49168 → 80 [SYN] Seq=0 Win=8192
2	0.000032270	192.168.32.100	192.168.32.101	TCP	66	80 → 49168 [SYN, ACK] Seq=0 Ack=
3	0.000213582	192.168.32.101	192.168.32.100	TCP	60	49168 → 80 [ACK] Seq=1 Ack=1 Wi
4	0.000359487	192.168.32.101	192.168.32.100	HTTP	354	GET /epicode.internal HTTP/1.1
5	0.000367098	192.168.32.100	192.168.32.101	TCP	54	80 → 49168 [ACK] Seq=1 Ack=301
6	0.016681173	192.168.32.100	192.168.32.101	TCP	204	80 → 49168 [PSH, ACK] Seq=1 Ack=
7	0.018938688	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
8	0.019223756	192.168.32.101	192.168.32.100	TCP	60	49168 → 80 [ACK] Seq=301 Ack=410
9	0.019372970	192.168.32.101	192.168.32.100	TCP	60	49168 → 80 [FIN, ACK] Seq=301 A
10	0.019381552	192.168.32.100	192.168.32.101	TCP	54	80 → 49168 [ACK] Seq=410 Ack=301
11	16.963530012	192.168.32.101	192.168.32.100	TCP	66	49169 → 80 [SYN] Seq=0 Win=8192
12	16.963565802	192.168.32.100	192.168.32.101	TCP	66	80 → 49169 [SYN, ACK] Seq=0 Ack=
13	16.963843657	192.168.32.101	192.168.32.100	TCP	60	49169 → 80 [ACK] Seq=1 Ack=1 Wi
14	16.964109080	192.168.32.101	192.168.32.100	HTTP	354	GET /epicode.internal HTTP/1.1
15	16.964115463	192.168.32.100	192.168.32.101	TCP	54	80 → 49169 [ACK] Seq=1 Ack=301
16	16.980482151	192.168.32.100	192.168.32.101	TCP	204	80 → 49169 [PSH, ACK] Seq=1 Ack=
17	16.982830638	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
18	16.983173586	192.168.32.101	192.168.32.100	TCP	60	49169 → 80 [ACK] Seq=301 Ack=410
19	16.983273885	192.168.32.101	192.168.32.100	TCP	60	49169 → 80 [FIN, ACK] Seq=301 A
20	16.983282106	192.168.32.100	192.168.32.101	TCP	54	80 → 49169 [ACK] Seq=410 Ack=301
21	52.855226556	fe80::bc57:cafe:9e8...	ff02::1:2	DHCPv6	146	Solicit XID: 0x41f36c CID: 00010
22	53.851312784	fe80::bc57:cafe:9e8...	ff02::1:2	DHCPv6	146	Solicit XID: 0x41f36c CID: 00010
23	55.852090904	fe80::bc57:cafe:9e8...	ff02::1:2	DHCPv6	146	Solicit XID: 0x41f36c CID: 00010
24	59.854744799	fe80::bc57:cafe:9e8...	ff02::1:2	DHCPv6	146	Solicit XID: 0x41f36c CID: 00010
25	67.859512919	fe80::bc57:cafe:9e8...	ff02::1:2	DHCPv6	146	Solicit XID: 0x41f36c CID: 00010
26	83.867035793	fe80::bc57:cafe:9e8...	ff02::1:2	DHCPv6	146	Solicit XID: 0x41f36c CID: 00010
27	115.883052765	fe80::bc57:cafe:9e8...	ff02::1:2	DHCPv6	146	Solicit XID: 0x41f36c CID: 00010
28	347.596101319	fe80::a00:27ff:feff...	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:ff:cc:77

Frame 4: 354 bytes on wire (2832 bits), 354 bytes captured (2832 bits) on Ethernet II, Src: PcsCompu_97:1c:91 (08:00:27:97:1c:91), Dst: PcsCompu_ff:cc:77 (08:00:27:ff:cc:77)

Destination: PcsCompu_ff:cc:77 (08:00:27:ff:cc:77)

Address: PcsCompu_ff:cc:77 (08:00:27:ff:cc:77)

.....0..... = LG bit: Globally unique address (factory default)

.....0..... = IG bit: Individual address (unicast)

Source: PcsCompu_97:1c:91 (08:00:27:97:1c:91)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100

Transmission Control Protocol, Src Port: 49168, Dst Port: 80, Seq: 1, Ack: 301

Hypertext Transfer Protocol

GET /epicode.internal HTTP/1.1\r\n

Accept: */*\r\n

Accept-Language: en-US\r\n

0000 08 00 27 ff cc 77 08 00 2

0010 01 54 01 27 40 00 80 06 3

0020 20 64 c0 10 00 50 44 cd 7

0030 40 29 f2 18 00 00 47 45 5

0040 64 65 2e 69 6e 74 65 72 6

0050 2f 31 2e 31 0d 0a 41 63 6

0060 2a 0d 0a 41 63 63 65 70 7

0070 67 65 3a 20 65 6e 2d 55 5

0080 41 67 65 6e 74 3a 20 4d 6

0090 2e 30 20 28 63 6f 6d 70 6

00a0 4d 53 49 45 20 38 2e 30 3

00b0 73 20 4e 54 20 36 2e 31 3

00c0 20 54 72 69 64 65 6e 74 2

00d0 43 43 32 3b 20 2e 4e 45 5

00e0 30 2e 35 30 37 32 37 3b 2



Fine della presentazione

Amedeo Natalizi