ThessSec

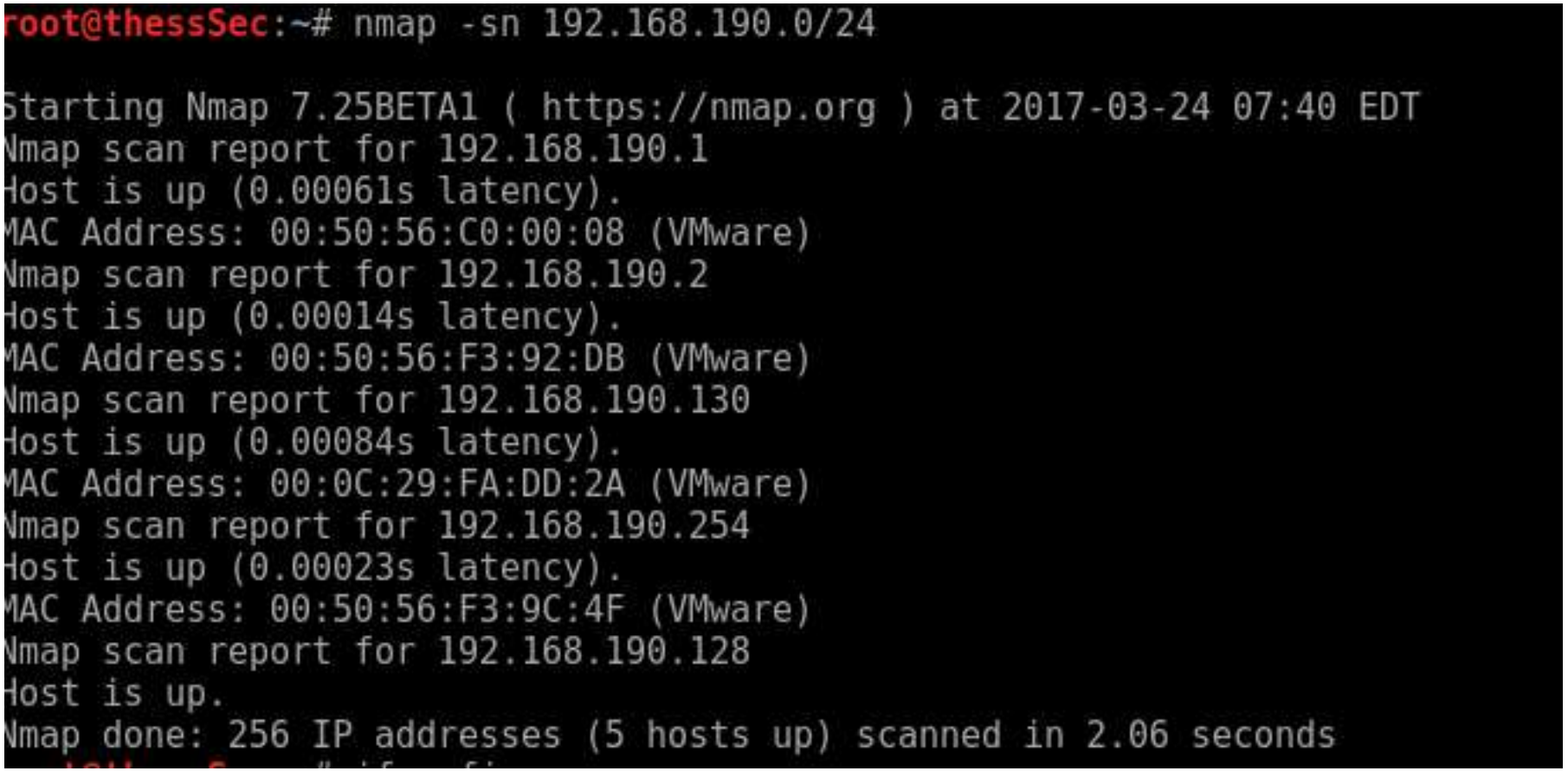
JUST ANOTHER GROUP OF PRIVACY & SECURITY ENTHUSIASTS.

3_NETWORKING_BASICS

Table of Contents

Recap

- Enumerate → nmap
  1. Ping scan whole network
  2. Find target
  3. Port scan the target


root@thessSec:~# nmap -sn 192.168.190.0/24
Starting Nmap 7.25BETA1 (https://nmap.org) at 2017-03-24 07:40 EDT
Nmap scan report for 192.168.190.1
Host is up (0.00061s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.190.2
Host is up (0.00014s latency).
MAC Address: 00:50:56:F3:92:DB (VMware)
Nmap scan report for 192.168.190.130
Host is up (0.00084s latency).
MAC Address: 00:0C:29:FA:DD:2A (VMware)
Nmap scan report for 192.168.190.254
Host is up (0.00023s latency).
MAC Address: 00:50:56:F3:9C:4F (VMware)
Nmap scan report for 192.168.190.128
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.06 seconds

```
root@thessSec:~# nmap -sS 192.168.190.130

Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-03-24 07:43 EDT
Nmap scan report for 192.168.190.130
Host is up (0.00017s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:FA:DD:2A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

```
root@thessSec:~# nmap -sV 192.168.190.130

Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-03-24 07:43 EDT
Nmap scan report for 192.168.190.130
Host is up (0.00012s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         Unreal ircd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:FA:DD:2A (VMware)
Service Info: Hosts:  metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OSs:
Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submi
t/ .
Nmap done: 1 IP address (1 host up) scanned in 14.55 seconds
```

Recap

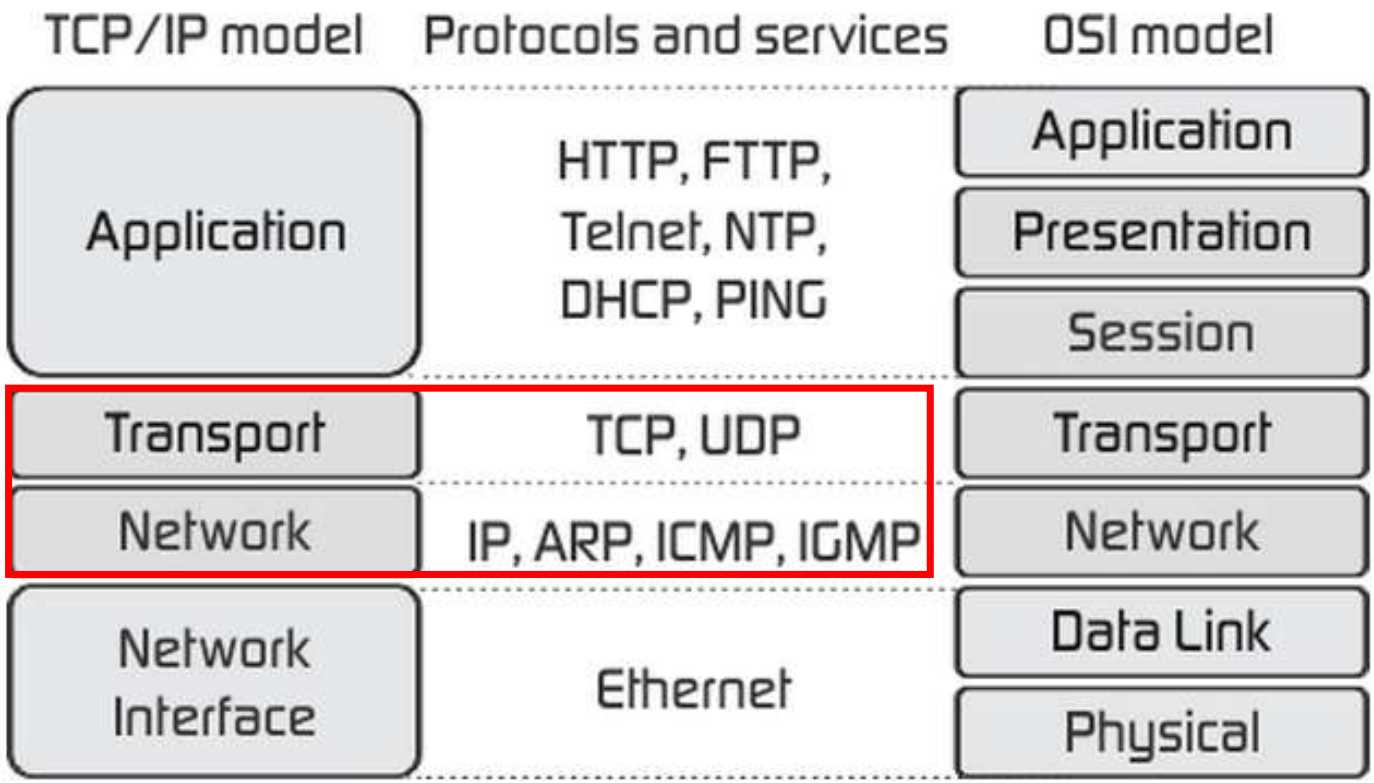
- Enumerate → nmap
  1. Ping scan whole network
  2. Find target
  3. Port scan the target
- Connect to server (IP, port)→ netcat (nc)
  1. Chat
  2. GET/POST Requests
  3. File transfers
  4. Reverse shells

```
root@thessSec:~# nc 172.217.22.14 80
GET / HTTP/1.0

HTTP/1.0 302 Found
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Location: http://www.google.gr/?gfe_rd=cr&ei=jefkWIWvAoug8wf-9bHQCw
Content-Length: 258
Date: Wed, 05 Apr 2017 12:48:13 GMT

<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.gr/?gfe_rd=cr&amp;ei=jefkWIWvAoug8wf-9bHQCw">here</A>.
</BODY></HTML>
```

TCP/IP


TCP/IP model
Protocols and services
OSI model
Application
HTTP, FTTP, Telnet, NTP, DHCP, PING
Application
Presentation
Session
Transport
TCP, UDP
Transport
Network
IP, ARP, ICMP, IGMP
Network
Network Interface
Ethernet
Data Link
Physical

Other terms…

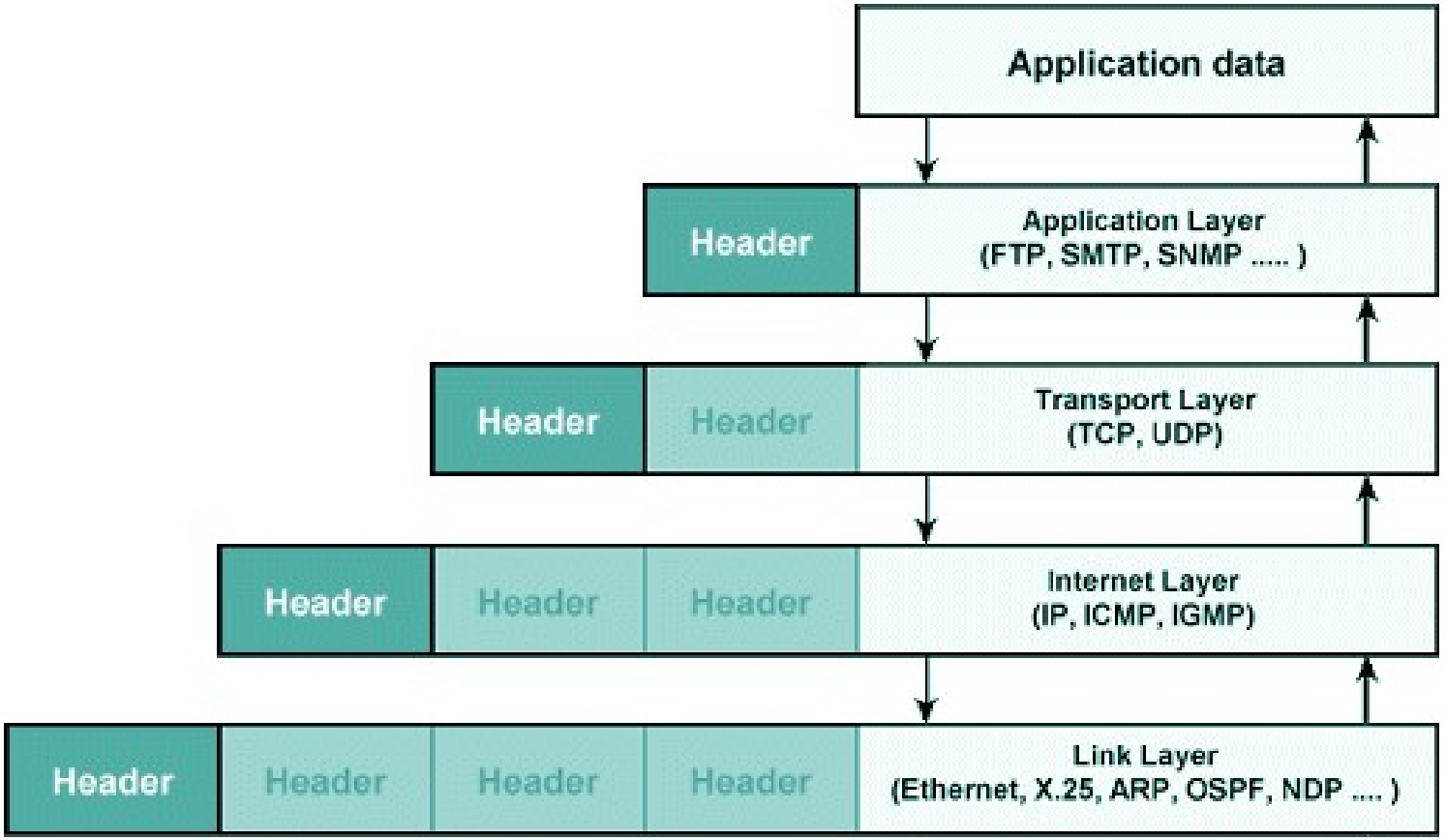
Domain Name system **(DNS)**:

- Name → IP addresses
- Found online or via commands like nslookup

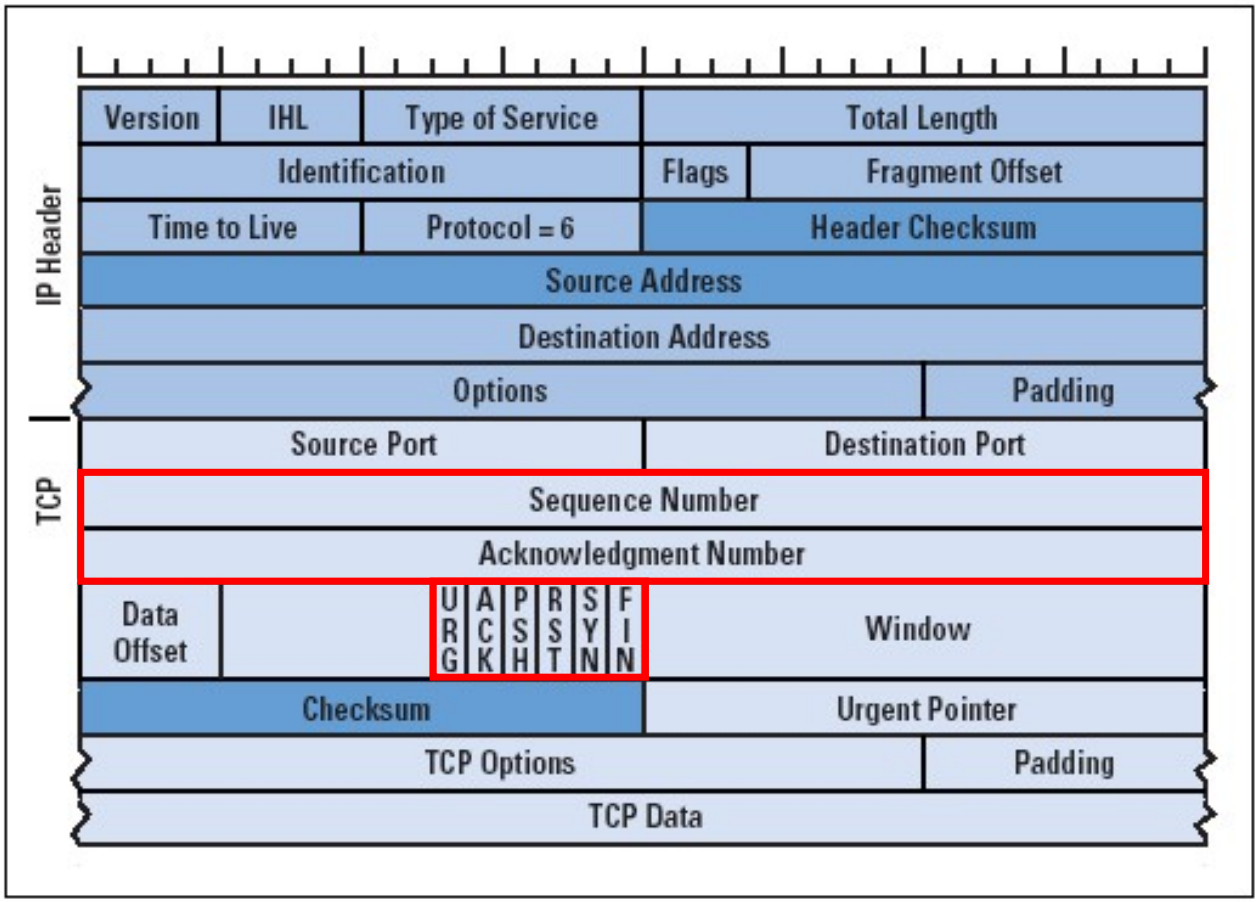
Address Resolution Protocol **(ARP)**:

- IP addresses → MAC addresses
- Found via arp command

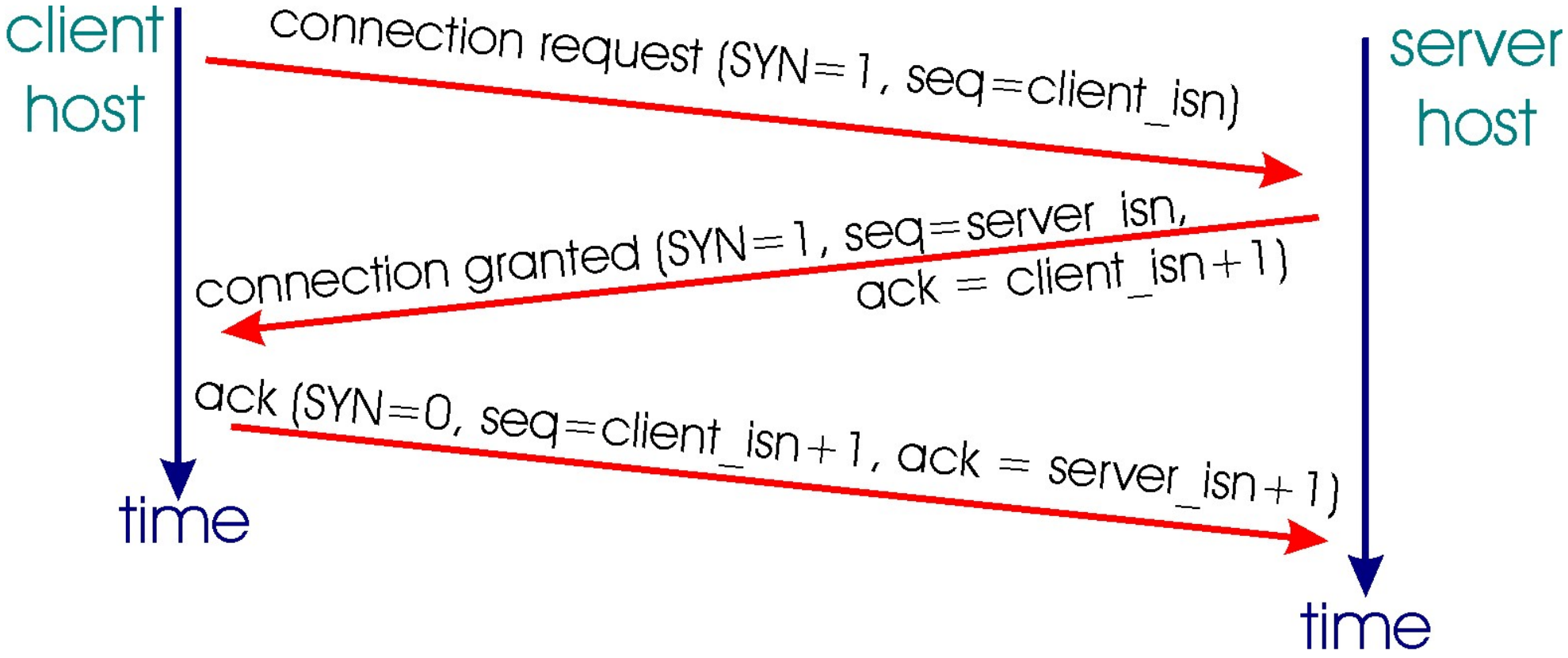
Packet Encapsulation



TCP/IP Header



Establishing TCP Connections / 3-Way Handshake



Establishing TCP Connections / 3-Way Handshake

translated to English:

Computer A: hello computer B, can you hear me?

Computer B: I acknowledge that I can hear you. Also, can you hear me?

Computer A: I acknowledge I can hear you. Let's talk about reddit.

It is done this way because it is the most efficient way to tell if the communication channels are open *in both directions*.

https://www.reddit.com/r/explainlikeimfive/comments/1ahc5s/eli5_tcpip/

Scenario 1 – ping request (hostname)

1. Resolve DNS to IP
2. Send ICMP request
3. Receive ICMP reply

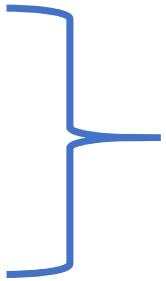
What if we ping the IP?

Wireshark : ping google.com

No.	Time	Source	Destination	Protocol	Length	Info
39	1.243173433	10.0.1.128	10.0.1.2	DNS	70	Standard query 0x8024 A google.com
40	1.243326041	10.0.1.128	10.0.1.2	DNS	70	Standard query 0x0153 AAAA google.com
41	1.281298263	10.0.1.2	10.0.1.128	DNS	86	Standard query response 0x8024 A google.com A 172.217.22.46
42	1.281975027	10.0.1.2	10.0.1.128	DNS	98	Standard query response 0x0153 AAAA google.com AAAA 2a00:1450:4001:820::200e
43	1.282578737	10.0.1.128	172.217.22.46	ICMP	98	Echo (ping) request id=0x0df5, seq=1/256, ttl=64 (reply in 44)
44	1.356664312	172.217.22.46	10.0.1.128	ICMP	98	Echo (ping) reply id=0x0df5, seq=1/256, ttl=128 (request in 43)
45	1.356854628	10.0.1.128	10.0.1.2	DNS	86	Standard query 0xe6c3 PTR 46.22.217.172.in-addr.arpa
46	1.395358496	10.0.1.2	10.0.1.128	DNS	155	Standard query response 0xe6c3 PTR 46.22.217.172.in-addr.arpa PTR fra15s16-in-
64	2.285137467	10.0.1.128	172.217.22.46	ICMP	98	Echo (ping) request id=0x0df5, seq=2/512, ttl=64 (reply in 65)
65	2.366766070	172.217.22.46	10.0.1.128	ICMP	98	Echo (ping) reply id=0x0df5, seq=2/512, ttl=128 (request in 64)
67	3.287007922	10.0.1.128	172.217.22.46	ICMP	98	Echo (ping) request id=0x0df5, seq=3/768, ttl=64 (reply in 68)

Scenario 2 – HTTP(s) request

1. Resolve DNS to IP
2. Send packet with SYN flag set
3. Receive packet with SYN, ACK flag set
4. Send packet with ACK flag set
5. Send GET request (HTTP Protocol)
6. Receive packet with ACK flag set
7. Receive HTTP response
8. Close connection with ACK – FIN/ACK sequence



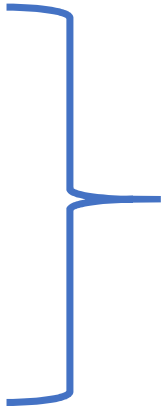
3-Way handshake.
Connection established.

Wireshark : curl google.com

No.	Time	Source	Destination	Protocol	Length	Info
8	1.549847121	10.0.1.128	10.0.1.2	DNS	70	Standard query 0x4a88 A google.com
9	1.550007571	10.0.1.128	10.0.1.2	DNS	70	Standard query 0x4917 AAAA google.com
10	1.554309425	10.0.1.2	10.0.1.128	DNS	86	Standard query response 0x4a88 A google.com A 172.217.16.174
11	1.554906890	10.0.1.2	10.0.1.128	DNS	98	Standard query response 0x4917 AAAA google.com AAAA 2a00:1450:4
12	1.563393660	10.0.1.128	172.217.16.174	TCP	74	56520→80 [SYN] Seq=3733452883 Win=29200 Len=0 MSS=1460 SACK_PERI
13	1.638200799	172.217.16.174	10.0.1.128	TCP	60	80→56520 [SYN, ACK] Seq=1232215861 Ack=3733452884 Win=64240 Len=
14	1.638303210	10.0.1.128	172.217.16.174	TCP	54	56520→80 [ACK] Seq=3733452884 Ack=1232215862 Win=29200 Len=0
15	1.638784702	10.0.1.128	172.217.16.174	HTTP	128	GET / HTTP/1.1
16	1.639415459	172.217.16.174	10.0.1.128	TCP	60	80→56520 [ACK] Seq=1232215862 Ack=3733452958 Win=64240 Len=0
17	1.712926190	172.217.16.174	10.0.1.128	HTTP	525	HTTP/1.1 302 Found (text/html)
18	1.712997782	10.0.1.128	172.217.16.174	TCP	54	56520→80 [ACK] Seq=3733452958 Ack=1232216333 Win=30016 Len=0
19	1.714620600	10.0.1.128	172.217.16.174	TCP	54	56520→80 [FIN, ACK] Seq=3733452958 Ack=1232216333 Win=30016 Len=
20	1.716437623	172.217.16.174	10.0.1.128	TCP	60	80→56520 [ACK] Seq=1232216333 Ack=3733452959 Win=64239 Len=0
21	1.786113815	172.217.16.174	10.0.1.128	TCP	60	80→56520 [FIN, PSH, ACK] Seq=1232216333 Ack=3733452959 Win=6423
22	1.786193488	10.0.1.128	172.217.16.174	TCP	54	56520→80 [ACK] Seq=3733452959 Ack=1232216334 Win=30016 Len=0

Scenario 3 – nc shell

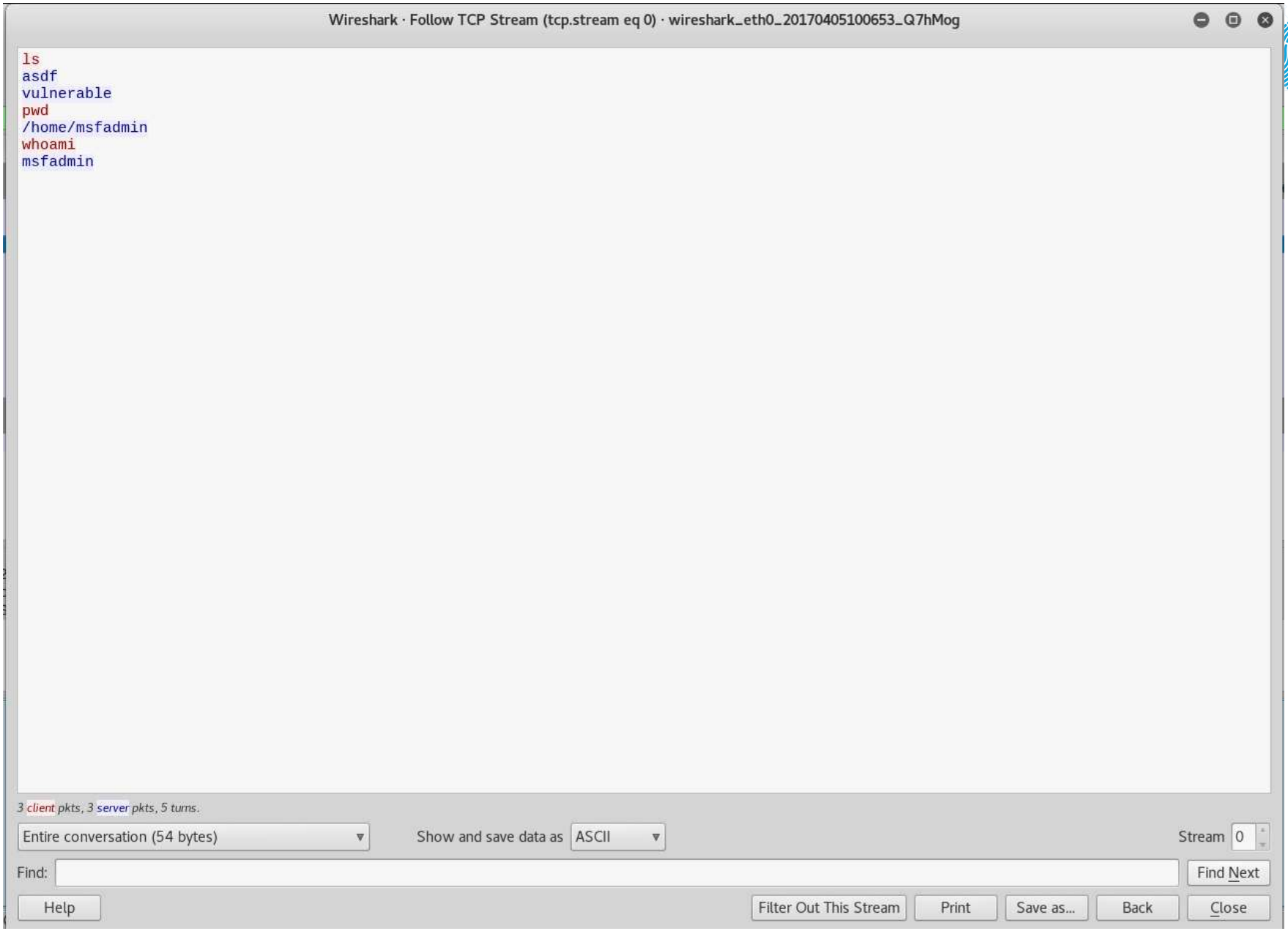
1. 3-Way handshake
2. Send Command packet (PSH/ACK)
3. Receive confirmation (ACK)
4. Receive Response packet (PSH/ACK)
5. Send confirmation (ACK)
6. …
7. Close connection with ACK – FIN/ACK sequence



UNENCRYPTED DATA

Wireshark: (3) commands over netcat interactive shell

No.	Time	Source	Destination	Protocol	Length	Info
702	785.644007637	10.0.1.130	10.0.1.128	TCP	74	44689→4444 [SYN] Seq=4081970295 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=664
705	785.644213307	10.0.1.128	10.0.1.130	TCP	74	4444→44689 [SYN, ACK] Seq=3774769288 Ack=4081970296 Win=28960 Len=0 MSS=1460
706	785.644332220	10.0.1.130	10.0.1.128	TCP	66	44689→4444 [ACK] Seq=4081970296 Ack=3774769289 Win=5856 Len=0 TSval=664250 TS
709	789.710229063	10.0.1.128	10.0.1.130	TCP	69	4444→44689 [PSH, ACK] Seq=3774769289 Ack=4081970296 Win=29056 Len=3 TSval=249
710	789.710586228	10.0.1.130	10.0.1.128	TCP	66	44689→4444 [ACK] Seq=4081970296 Ack=3774769292 Win=5856 Len=0 TSval=664652 TS
711	789.712268015	10.0.1.130	10.0.1.128	TCP	82	44689→4444 [PSH, ACK] Seq=4081970296 Ack=3774769292 Win=5856 Len=16 TSval=664
712	789.712293713	10.0.1.128	10.0.1.130	TCP	66	4444→44689 [ACK] Seq=3774769292 Ack=4081970312 Win=29056 Len=0 TSval=249053
715	791.400713198	10.0.1.128	10.0.1.130	TCP	73	4444→44689 [PSH, ACK] Seq=3774769292 Ack=4081970312 Win=29056 Len=7 TSval=249
716	791.402353389	10.0.1.130	10.0.1.128	TCP	75	44689→4444 [PSH, ACK] Seq=4081970312 Ack=3774769299 Win=5856 Len=9 TSval=6648
717	791.402407038	10.0.1.128	10.0.1.130	TCP	66	4444→44689 [ACK] Seq=3774769299 Ack=4081970321 Win=29056 Len=0 TSval=2490960
718	792.779125927	10.0.1.128	10.0.1.130	TCP	70	4444→44689 [PSH, ACK] Seq=3774769299 Ack=4081970321 Win=29056 Len=4 TSval=249
719	792.779577595	10.0.1.130	10.0.1.128	TCP	81	44689→4444 [PSH, ACK] Seq=4081970321 Ack=3774769303 Win=5856 Len=15 TSval=664
720	792.779595916	10.0.1.128	10.0.1.130	TCP	66	4444→44689 [ACK] Seq=3774769303 Ack=4081970336 Win=29056 Len=0 TSval=2491304
721	797.796391158	10.0.1.128	10.0.1.130	TCP	66	4444→44689 [FIN, ACK] Seq=3774769303 Ack=4081970336 Win=29056 Len=0 TSval=249
722	797.797314355	10.0.1.130	10.0.1.128	TCP	66	44689→4444 [FIN, ACK] Seq=4081970336 Ack=3774769304 Win=5856 Len=0 TSval=6654
723	797.797356047	10.0.1.128	10.0.1.130	TCP	66	4444→44689 [ACK] Seq=3774769304 Ack=4081970337 Win=29056 Len=0 TSval=2492558


Wireshark · Follow TCP Stream (tcp.stream eq 0) · wireshark_eth0_20170405100653_Q7hMog
ls
asdf
vulnerable
pwd
/home/msfadmin
whoami
msfadmin
3 client pkts, 3 server pkts, 5 turns.
Entire conversation (54 bytes)
Show and save data as
ASCII
Stream
0
Find:
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close

Scenario 4 – nmap Version Detection

For each port:

1. 3 –Way Handshake
2. Sends packets for specific versions
3. If response matches a version OK
4. Close connection with ACK – FIN/ACK sequence

Scenarios Demo

Thank you for your attention

georgkonst@ece.auth.gr

github.com/gakonst