

Knowledge Check - Answers

- Convert CAFE to decimal – C=12, A=10, F=15, E=14

$$12(16^3)+10(16^2)+15(16^1)+14(16^0) =$$

$$12*4096+10*256+15*16+14 = 51966$$

- Convert 7A69 to binary – 0111 1010 0110 1001

- What is the high order nibble of the low order byte in binary?

0110

- What is the low order nibble of the high order byte in hex?

A



Knowledge Check - Answers

- What ICMP type and code indicate a ping response?
 - Type 0, code 0
- What two bytes start most IPv4 packets?
 - 0x4500
- What ICMP type might indicate a traceroute?
 - Type 11
- What are the sets of flags in a normal TCP 3-way handshake?
 - SYN, SYN-ACK, ACK
- What are the two common methods of terminating a TCP connection?
 - FIN, RST



Knowledge Check - Answers

- Find SYN-ACK packets
 - `tcp[13]=0x12`
- Find packets with a source port of 80
 - `src port 80`
 - `udp[0:2]=80 || tcp[0:2]=80`
- Find packets with all TCP flags set
 - `tcp[13]=0xff`
- Find all UDP packets with a destination port of 123
 - `udp[2:2]=123`
 - `udp && dst port 123`
- Find all type 3 code 3 ICMP packets
 - `icmp[0]=3 && icmp[1]=3`



Practical Exercise – Fragmentation Answers

- frags.pcap
 - Why might only 1 host have responded to the ICMP echo request?
 - Fragmented ICMP blocked by Google, but not by local router
- frags2.pcap
 - What was the server's response code?
 - 301
 - What URL was probably requested?
 - www.google.com



Practical Exercise – Scanning Answers

-nmap.pcap

-Which host was performing scanning?

- 192.168.2.175

-tcpdump -r nmap.pcap -nn 'tcp[13]=0x02' | cut -f3 -d' ' | cut -f1-4 -d '.' | sort | uniq -c | sort -n

```
reading from file nmap.pcap, link-type EN10MB (Ethernet)
```

```
1 192.168.2.45
3 192.168.2.75
18 192.168.2.98
4537 192.168.2.175
```



Practical Exercise – Scanning Answers

-nmap.pcap

-Which host was the scan target?

- 192.168.2.75

-tcpdump -r nmap.pcap -nn 'tcp[13]=0x02' | cut -f5 -d' ' | cut -f1-4 -d '.' | sort | uniq -c | sort -n

```
reading from file nmap.pcap, link-type EN10MB (Ethernet)
```

```
1 65.55.44.109
4 192.168.2.45
16 192.168.3.249
4538 192.168.2.75
```



Practical Exercise – Scanning Answers

- nmap.pcap
 - How many TCP ports were scanned?
 - 1000
 - `sudo tcpdump -r nmap.pcap -nn 'tcp[13]=0x02 && src host 192.168.2.175 && dst host 192.168.2.75' | cut -f 5 -d ' ' | cut -f5 -d '.' | cut -f1 -d ':' | sort | uniq | wc -l`
 - How many UDP ports were scanned?
 - 30
 - `sudo tcpdump -r nmap.pcap -nn 'udp && src host 192.168.2.175 && dst host 192.168.2.75' | cut -f 5 -d ' ' | cut -f5 -d '.' | cut -f1 -d ':' | sort | uniq | wc -l`



Practical Exercise – Scanning Answers

- nmap.pcap

- Which TCP ports were open?

- 1025,1026,1027,1031,1032,1034,135,139,2869,3306,445,5357

- sudo tcpdump -r nmap.pcap -nn

- 'tcp[13]=0x12 && dst host

- 192.168.2.175 && src host

- 192.168.2.75'| cut -f 3 -d' ' | cut -f5 -d'.'|

- cut -f1 -d':' | sort | uniq

reading from file nmap.pcap

1025

1026

1027

1031

1032

1034

135

139

2869

3306

445

5357



Practical Exercise – Scanning Answers

- nmap.pcap
 - What may have caused the scan behavior after 12:04:10?
 - Firewall enabled that does not reset connections

```
56710 → 445 [SYN] Seq=0 Win=29200 L
33664 → 111 [SYN] Seq=0 Win=29200 L
56884 → 256 [SYN] Seq=0 Win=29200 L
33784 → 443 [SYN] Seq=0 Win=29200 L
33602 → 22 [SYN] Seq=0 Win=29200 L
42012 → 53 [SYN] Seq=0 Win=29200 L
36270 → 113 [SYN] Seq=0 Win=29200 L
34480 → 23 [SYN] Seq=0 Win=29200 L
52224 → 587 [SYN] Seq=0 Win=29200 L
51006 → 8080 [SYN] Seq=0 Win=29200 L
51008 → 8080 [SYN] Seq=0 Win=29200 L
52230 → 587 [SYN] Seq=0 Win=29200 L
```



ch2.pcap (from root-me.org)

- Find the FTP password
- ngrep for PASS

```
rangercha@kali:~/mnt/hgfs/vmshared/training/packet_intro_long$ ngrep -qI ch2.pcap 'PASS'
input: ch2.pcap
filter: ((ip || ip6) || (vlan && (ip || ip6)))
match: PASS
```

```
T 10.20.144.150:35974 -> 10.20.144.151:21 [AP] #11
PASS cdts3500..
```



ch3.pcap (from root-me.org)

- Find the twitter password
- `ngrep -qI ch3.pcap 'Authorization: Basic' | tr ':' '\n' | grep 'Basic' | cut -f3 -d' ' | cut -f1 -d'.' | base64 -d`

```
rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$ ngrep -qI ch
3.pcap 'Authorization: Basic' | tr ':' '\n' | grep 'Basic' | cut -f3 -d' '
| cut -f1 -d'.' | base64 -d
usertest:passwordrangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_l
ong$ █
```



2015-03-24-traffic-analysis-exercise.pcap (from malware-traffic-analysis.net)

- IP address of the infected host
 - 192.168.122.200
- Which server is probably legitimate, but compromised?
 - forums.pelicanparts.com
- What redirection techniques does this exploit kit use?
 - Script tag injection
- What types of exploits were served?
 - Adobe Reader, Flash, Java, Silverlight



sansholidayhack2013.pcap (from SANS holiday hack challenge, 2013)

- What are the MACs and IPs of the machines that ARP cache poison? Is it successful?
 - 00:0c:29:f7:f4:9a - 10.21.22.253 - successful
 - 10.25.22.252 - unsuccessful
- What are the IPs of the systems that were port scanning?
 - 10.25.22.252, 10.21.22.253, 10.25.22.253
 - `tcpdump -nn -r sansholidayhack2013.pcap 'tcp[13]=0x02' | cut -f3,5 -d' ' | tr ' ' '.' | cut -f1-4,6-10 -d'.' | sort | uniq -c | sort -rn`
 - `tcpdump -nn -r sansholidayhack2013.pcap 'tcp[13]=0x14' | cut -f5 -d' ' | cut -f1-4 -d'.' | sort | uniq -c | sort -n`



sansholidayhack2013.pcap
(from SANS holiday hack challenge, 2013)

- What systems and protocols did the scans discover?

```
-tcpdump -nn -r sansholidayhack2013.pcap  
'tcp[13]=0x12 && (dst host 10.25.22.252 || dst  
host 10.21.22.253 || dst host 10.25.22.253)' |  
cut -f3 -d' ' | sort | uniq
```



sansholidayhack2013.pcap

(from SANS holiday hack challenge, 2013)

10.16.11.5.110	10.25.22.22.80	165.254.158.56.80	208.80.154.234.80
10.21.22.22.502	10.25.22.23.80	173.194.43.47.443	208.80.154.240.80
10.21.22.23.502	10.25.22.250.80	192.190.173.45.80	216.22.25.175.80
10.21.22.24.502	10.25.22.30.80	192.204.3.75.80	54.230.49.239.80
10.21.22.253.1225	10.25.22.58.4444	199.7.57.72.80	63.245.217.36.80
10.2.2.2.8081	10.25.22.58.445	208.80.154.224.80	69.16.175.10.80
72.167.239.239.80	74.125.226.239.443	82.103.140.42.443	74.125.226.228.80



sansholidayhack2013.pcap

(from SANS holiday hack challenge, 2013)

72.21.195.198.443	74.125.226.242.443	10.25.22.22.44818
72.21.203.211.80	74.125.226.251.80	165.254.138.136.80
72.21.214.3.443	74.125.22.82.80	208.80.154.225.80
72.21.215.52.80	81.169.180.37.443	69.195.141.178.443
74.125.226.199.443	81.169.180.37.80	82.103.134.102.80



sansholidayhack2013.pcap
(from SANS holiday hack challenge, 2013)

- What account was used in an attack over the SMB protocol?
 - ernie



packet_intro.pcap

- What are the max and minimum SSL/TLS versions supported by 192.168.2.122?
 - Max: TLS 1.2
 - Min: TLS 1.0



packet_intro.pcap

The image shows a Wireshark packet capture of an SSL handshake. The top pane shows a list of packets, with packet 111 selected. The middle pane shows the details of packet 111, which is a TLSv1.2 record. The bottom pane shows the raw data of the record, which is a TLSv1.2 Client Hello message. The details pane is expanded to show the 'Handshake Protocol: Client Hello' section, which includes the 'Handshake Type: Client Hello (1)' and the 'Version: TLS 1.2 (0x0303)'. The raw data pane shows the hex and ASCII representation of the message, including the 'Random' and 'Session ID' fields.

ssl.handshake

No.	Time	Source	Destination	Protocol	Length
111	2016-05-29 ...	192.168.2.122	216.58.193.132	TLSv1.2	583
116	2016-05-29 ...	216.58.193.132	192.168.2.122	TLSv1.2	212
118	2016-05-29 ...	192.168.2.122	216.58.193.132	TLSv1.2	212

> Frame 111: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface 0

> Ethernet II, Src: VMware_f0:0b:61 (00:0c:29:f0:0b:61), Dst: AsustekC_be:1 (08:00:27:00:00:00)

> Internet Protocol Version 4, Src: 192.168.2.122, Dst: 216.58.193.132

> Transmission Control Protocol, Src Port: 40702, Dst Port: 443, Seq: 1, Ack: 340360000, Len: 583, Window: 65535, Options: None, Urgency: 0

Secure Sockets Layer

✓ TLSv1.2 Record Layer: Handshake Protocol: Client Hello

Content Type: Handshake (22)

Version: TLS 1.0 (0x0301)

Length: 512

✓ Handshake Protocol: Client Hello

Handshake Type: Client Hello (1)

Length: 503

Version: TLS 1.2 (0x0303)

Random: f2d3a94cf3a18e1e4b45f09e01c9a2353710b639e7ef06b4...

Session ID Length: 32

Session ID: dae500cd74df5dc6c9a4351d72b82efda997499260631196...



packet_intro.pcap

- Flag 1

- http.request.full_uri contains “flag”

http.request.full_uri contains flag			
	Time	Source	Dest
56	2016-05-29 22:14:30.517459	192.168.2.122	54.152.144.243

Frame 56: 175 bytes on wire (1400 bits), 175 bytes captured (1400 bits) on interface 0
Ethernet II, Src: Vmware_f0:0b:61 (00:0c:29:f0:0b:61), Dst: AsustekC_be:f7:98 (08:00:27:be:f7:98)
Internet Protocol Version 4, Src: 192.168.2.122, Dst: 54.152.144.243
Transmission Control Protocol, Src Port: 37452, Dst Port: 80, Seq: 1, Ack: 1, Len: 175
Hypertext Transfer Protocol
> GET /f490a35ac6f7ee0e686a7e2179524bf2 HTTP/1.1\r\nHost: www.flag1.com\r\nUser-Agent: curl/7.47.0\r\nAccept: */*\r\n\r\n[Full request URI: http://www.flag1.com/f490a35ac6f7ee0e686a7e2179524bf2]



packet_intro.pcap

-Flag 2

-ngrep -qI packet_intro.pcap 'flag2'

```
root@kali:/mnt/hgfs/vmshared/training/packet_intro_long# ngrep -qI packet_intro.pcap
'flag2'
input: packet_intro.pcap
filter: ((ip || ip6) || (vlan && (ip || ip6)))
match: flag2

T 192.168.2.122:57196 -> 66.235.120.113:80 [AP] #2344
GET / HTTP/1.1..Host: askjeevs.com..User-Agent: flag2:82d5927b53538c2da5c4e5eadba2b
f2a..Accept: */*....
root@kali:/mnt/hgfs/vmshared/training/packet_intro_long#
```



packet_intro.pcap

-Flag 3

-ngrep -qI packet_intro.pcap 'flag3'

```
root@kali:/mnt/hgfs/vmshared/training/packet_intro_long# ngrep -qI packet_intro.pcap
'flag3'
input: packet_intro.pcap
filter: ((ip || ip6) || (vlan && (ip || ip6)))
match: flag3

U 192.168.2.122:60663 -> 8.8.8.8:53 #4750
.V..... b768e1e8075fd6c1b7c11c84536dd467.flag3.com.....

U 8.8.8.8:53 -> 192.168.2.122:60663 #4757
.V..... b768e1e8075fd6c1b7c11c84536dd467.flag3.com.....-.....?.dns1.re
gistrar-servers.com,,.hostmaster.Mx..0.....:....._
```



packet_intro.pcap

-Flag 4

```
-echo "flag" | base64  
-ngrep -ql packet_intro.pcap 'ZmxhZ'  
-ngrep -ql packet_intro.pcap 'ZmxhZ' | grep  
'ZmxhZ' | tr -s ' ' | cut -f3 -d' ' | cut -f2- -d'=' |  
base64 -d
```



packet_intro.pcap

```
rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$ echo "flag" | base64
ZmxhZwo=
rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$ ngrep -qI packet_intro.pcap 'ZmxhZ'
input: packet_intro.pcap
filter: ((ip || ip6) || (vlan && (ip || ip6)))
match: ZmxhZ

T 192.168.2.122:56154 -> 174.36.107.130:80 [AP] #5040
GET /?a=ZmxhZzQ6MDIxMDFkZDNmNWYxZjA0NGRiYjM1YTUzMDUyMjQxOTc== HTTP/1.1..Host: supersketch.com..User-Agent: curl/7.47.
0..Accept: */*....

rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$ ngrep -qI packet_intro.pcap 'ZmxhZ' |
grep 'ZmxhZ' | tr -s ' ' | cut -f3 -d' ' | cut -f2- -d'=' | base64 -d
flag4:02101dd3f5f1f044dbb35a5305224197base64: invalid input
rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$
```



packet_intro.pcap

-Flag 5

-echo "flag5" | xxd -ps

-ngrep -qI packet_intro.pcap '666c616735'

-printf

"666c6167353a643638666337323164613331
6565663932663565373939616466643534653
139" | xxd -r -p



packet_intro.pcap

```
rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$ echo "flag5" | xxd -ps
666c6167350a
rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$ ngrep -qI packet_intro.pcap '666c6167350a'
input: packet_intro.pcap
filter: ((ip || ip6) || (vlan && (ip || ip6)))
match: 666c616735

T 192.168.2.122:51588 -> 204.79.197.200:80 [AP] #5712
GET /search=666c6167353a6436386663373231646133316565663932663565373939616466643534653139 HTTP/1.1..Host: bing.com..User-Agent: curl/7.47.0..Accept: */*....

T 204.79.197.200:80 -> 192.168.2.122:51588 [AP] #5728
HTTP/1.1 301 Moved Permanently..Location: http://www.bing.com/search=666c6167353a6436386663373231646133316565663932663565373939616466643534653139..Server: Microsoft-IIS/8.5..X-MSEdge-Ref: Ref A: FE730BDA1B4241D7AC872F3CE666680F Ref B: 36FB7548E3AF536AF11F30FF82345B4A Ref C: Sun May 29 19:15:33 2016 PST..Date: Mon, 30 May 2016 02:15:32 GMT..Content-Length: 0....
rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$ printf "666c6167353a6436386663373231646133316565663932663565373939616466643534653139" | xxd -r -p
```



packet_intro.pcap

-Flag 6

```
-tshark -r packet_intro.pcap -T fields -e  
icmp.type -Y 'icmp' | xargs printf '%x' | xxd -r -p
```



packet_intro.pcap

```
rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$ tshark -r packet_intro.pcap -T fields -e icmp.type -Y 'icmp' | xargs printf '%x' 666c6167363a3639353734643061616339373230613432383736323763373731303238393330ddddddddddddddddddddddddddddddddddddddrangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$ tshark -r packet_intro.pcap -T fields -e icmp.type -Y 'icmp' | xargs printf '%x' | xxd -r -p flag6:69574d0aac9720a4287627c771028930?????????????????????rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$rangercha@kali:/mnt/hgfs/vmshared/training/packet intro long$
```



packet_intro.pcap

-Flag 7

```
-tcpdump -r packet_intro.pcap -nnA 'icmp' |  
  grep '\.\.' | cut -f4 -d'-' | grep -v '@\.\.\.\.\.\.z' | tr  
  -d '\n'
```



packet_intro.pcap

```
rangercha@kali:/mnt/hgfs/vmshared/training/packet_intro_long$ sudo tcpdump  
-r packet_intro.pcap -nnA 'icmp' | grep '\.\.' | cut -f4 -d'-' | grep  
-v '@\.\.\.\.\.\.z' | tr -d '\n'  
[sudo] password for rangercha:  
reading from file packet_intro.pcap, link-type EN10MB (Ethernet)  
flag7:02ae9641b7052bfa4124dc41943cf36c
```

