Jonathan Lam
Prof. Chen
ECE 303
Communication Networks
2 / 22 / 20

Project 2 - Transport

Port Scanning

1. What nmap command would perform a scan of the top *x* most common ports, specifically TCP only? Let the target host be “synprint.com” and *x*=10.
Answer: nmap --top-ports 10 synprint.com

Cut and paste the result of the nmap command below:

```
Nmap scan report for synprint.com (192.241.168.54)
Host is up (0.0057s latency).

PORT     STATE    SERVICE
21/tcp   filtered ftp
22/tcp   open     ssh
23/tcp   filtered telnet
25/tcp   filtered smtp
80/tcp   open     http
110/tcp  filtered pop3
139/tcp  filtered netbios-ssn
443/tcp  open     https
445/tcp  filtered microsoft-ds
3389/tcp filtered ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 1.35 seconds
```

2. Capture the packets of your nmap run. Save the file as “<date>_nmap.pcap”.
Provide the command you used to collect the packets:
Answer: tcpdump -w 21200218_nmap.pcap

Dump the textual result of the pcap file below (tshark -r <pcap_file>):

```
    1   0.000000 192.168.1.187 → 192.168.1.1  DNS 72 Standard query 0x2c56 A synprint.com
    2   0.000022 192.168.1.187 → 192.168.1.1  DNS 72 Standard query 0x7354 AAAA synprint.com
    3   0.003001  192.168.1.1 → 192.168.1.187 DNS 88 Standard query response 0x2c56 A synprint.com A 192.241.168.54
    4   0.003374  192.168.1.1 → 192.168.1.187 DNS 140 Standard query response 0x7354 AAAA synprint.com SOA ns01.domaincontrol.com
    5   0.004065 192.168.1.187 → 192.241.168.54 TCP 74 40294 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012205
TSecr=0 WS=128
```

```
    6   0.004089 192.168.1.187 → 192.241.168.54 TCP 74 41356 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012205
TSecr=0 WS=128
    7   0.010411 192.241.168.54 → 192.168.1.187 TCP 74 80 → 40294 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1
TSval=1854221149 TSecr=3146012205 WS=64
    8   0.010425 192.168.1.187 → 192.241.168.54 TCP 66 40294 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3146012212 TSecr=1854221149
    9   0.010431 192.241.168.54 → 192.168.1.187 TCP 74 443 → 41356 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1
TSval=1854221149 TSecr=3146012205 WS=64
   10   0.010435 192.168.1.187 → 192.241.168.54 TCP 66 41356 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3146012212 TSecr=1854221149
   11   0.010467 192.168.1.187 → 192.241.168.54 TCP 66 40294 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3146012212
TSecr=1854221149
   12   0.010480 192.168.1.187 → 192.241.168.54 TCP 66 41356 → 443 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3146012212
TSecr=1854221149
   13   0.010596 192.168.1.187 → 192.168.1.1  DNS 87 Standard query 0xec3f PTR 54.168.241.192.in-addr.arpa
   14   0.013613  192.168.1.1 → 192.168.1.187 DNS 113 Standard query response 0xec3f PTR 54.168.241.192.in-addr.arpa PTR synprint.com
   15   0.013695 192.168.1.187 → 192.241.168.54 TCP 74 33208 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012215
TSecr=0 WS=128
   16   0.013712 192.168.1.187 → 192.241.168.54 TCP 74 41360 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012215
TSecr=0 WS=128
   17   0.013724 192.168.1.187 → 192.241.168.54 TCP 74 40302 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012215
TSecr=0 WS=128
   18   0.013735 192.168.1.187 → 192.241.168.54 TCP 74 59336 → 25 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012215
TSecr=0 WS=128
   19   0.013745 192.168.1.187 → 192.241.168.54 TCP 74 41576 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012215
TSecr=0 WS=128
   20   0.013757 192.168.1.187 → 192.241.168.54 TCP 74 46986 → 110 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012215
TSecr=0 WS=128
   21   0.013768 192.168.1.187 → 192.241.168.54 TCP 74 47924 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012215
TSecr=0 WS=128
   22   0.013779 192.168.1.187 → 192.241.168.54 TCP 74 57352 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012215
TSecr=0 WS=128
   23   0.013789 192.168.1.187 → 192.241.168.54 TCP 74 43522 → 3389 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012215
TSecr=0 WS=128
   24   0.013799 192.168.1.187 → 192.241.168.54 TCP 74 54516 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146012215
TSecr=0 WS=128
   25   0.021402 192.241.168.54 → 192.168.1.187 TCP 74 22 → 57352 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1
TSval=1854221151 TSecr=3146012215 WS=64
   26   0.021415 192.168.1.187 → 192.241.168.54 TCP 66 57352 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3146012223 TSecr=1854221151
   27   0.021420 192.241.168.54 → 192.168.1.187 TCP 74 80 → 40302 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1
TSval=1854221151 TSecr=3146012215 WS=64
   28   0.021425 192.168.1.187 → 192.241.168.54 TCP 66 40302 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3146012223 TSecr=1854221151
   29   0.021428 192.241.168.54 → 192.168.1.187 TCP 74 443 → 41360 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1
TSval=1854221151 TSecr=3146012215 WS=64
   30   0.021431 192.168.1.187 → 192.241.168.54 TCP 66 41360 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3146012223 TSecr=1854221151
   31   0.021468 192.168.1.187 → 192.241.168.54 TCP 66 41360 → 443 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3146012223
TSecr=1854221151
   32   0.021480 192.168.1.187 → 192.241.168.54 TCP 66 40302 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3146012223
TSecr=1854221151
   33   0.021487 192.168.1.187 → 192.241.168.54 TCP 66 57352 → 22 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3146012223
TSecr=1854221151
   34   0.607564 192.168.1.187 → 192.168.1.158 TCP 176 52230 → 8009 [PSH, ACK] Seq=1 Ack=1 Win=705 Len=110 TSval=1155283588 TSecr=7281774
[TCP segment of a reassembled PDU]
   35   0.611794 192.168.1.158 → 192.168.1.187 TCP 176 8009 → 52230 [PSH, ACK] Seq=1 Ack=111 Win=528 Len=110 TSval=7282275
TSecr=1155283588 [TCP segment of a reassembled PDU]
   36   0.611835 192.168.1.187 → 192.168.1.158 TCP 66 52230 → 8009 [ACK] Seq=111 Ack=111 Win=705 Len=0 TSval=1155283592 TSecr=7282275
   37   0.932089 192.168.1.187 → 192.168.1.169 TCP 176 33522 → 8009 [PSH, ACK] Seq=1 Ack=1 Win=614 Len=110 TSval=1818755475 TSecr=6863540
[TCP segment of a reassembled PDU]
   38   0.937463 192.168.1.169 → 192.168.1.187 TCP 176 8009 → 33522 [PSH, ACK] Seq=1 Ack=111 Win=721 Len=110 TSval=6864041
TSecr=1818755475 [TCP segment of a reassembled PDU]
   39   0.937540 192.168.1.187 → 192.168.1.169 TCP 66 33522 → 8009 [ACK] Seq=111 Ack=111 Win=614 Len=0 TSval=1818755481 TSecr=6864041
   40   1.019453 208.255.115.145 → 192.168.1.187 TLSv1.2 105 Application Data
   41   1.019589 192.168.1.187 → 208.255.115.145 TCP 66 57112 → 443 [ACK] Seq=1 Ack=40 Win=501 Len=0 TSval=33648639 TSecr=2800984375
   42   1.019654 208.255.115.145 → 192.168.1.187 TCP 66 443 → 57112 [FIN, ACK] Seq=40 Ack=1 Win=117 Len=0 TSval=2800984375 TSecr=33528618
   43   1.020431 192.168.1.187 → 208.255.115.145 TCP 66 57112 → 443 [FIN, ACK] Seq=1 Ack=41 Win=501 Len=0 TSval=33648640 TSecr=2800984375
   44   1.023556 192.168.1.187 → 192.241.168.54 TCP 74 [TCP Retransmission] 54516 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TSval=3146013225 TSecr=0 WS=128
   45   1.023583 192.168.1.187 → 192.241.168.54 TCP 74 [TCP Retransmission] 43522 → 3389 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TSval=3146013225 TSecr=0 WS=128
   46   1.023592 192.168.1.187 → 192.241.168.54 TCP 74 [TCP Retransmission] 47924 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TSval=3146013225 TSecr=0 WS=128
   47   1.023601 192.168.1.187 → 192.241.168.54 TCP 74 [TCP Retransmission] 46986 → 110 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TSval=3146013225 TSecr=0 WS=128
   48   1.023608 192.168.1.187 → 192.241.168.54 TCP 74 [TCP Retransmission] 41576 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TSval=3146013225 TSecr=0 WS=128
   49   1.023614 192.168.1.187 → 192.241.168.54 TCP 74 [TCP Retransmission] 59336 → 25 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TSval=3146013225 TSecr=0 WS=128
   50   1.028425 208.255.115.145 → 192.168.1.187 TCP 66 443 → 57112 [ACK] Seq=41 Ack=2 Win=117 Len=0 TSval=2800984385 TSecr=33648640
   51   1.115229 192.168.1.187 → 192.241.168.54 TCP 74 54518 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146013316
```

```
TSecr=0 WS=128
   52   1.115637 192.168.1.187 → 192.241.168.54 TCP 74 43528 → 3389 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146013317
TSecr=0 WS=128
   53   1.115753 192.168.1.187 → 192.241.168.54 TCP 74 47936 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146013317
TSecr=0 WS=128
   54   1.115797 192.168.1.187 → 192.241.168.54 TCP 74 47002 → 110 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146013317
TSecr=0 WS=128
   55   1.115833 192.168.1.187 → 192.241.168.54 TCP 74 41596 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146013317
TSecr=0 WS=128
   56   1.115857 192.168.1.187 → 192.241.168.54 TCP 74 59360 → 25 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146013317
TSecr=0 WS=128
   57   1.115881 192.168.1.187 → 192.241.168.54 TCP 74 33240 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3146013317
TSecr=0 WS=128
```

3. Write a script/program in a language of your choice to parse the resulting pcap file, process the parsed results, and create the same output as what nmap generated. Make sure you can handle the at least 2 states: open and filtered. Your output should look something like the following:

```
PORT    STATE  SERVICE
21/tcp   filtered ftp
22/tcp   open   ssh
23/tcp   closed telnet
…
```

Cut and paste the analysis program/script below:
nmap_analysis.sh

```
#!/bin/sh

ip_addr=$(ip address show | grep "192.168.[0-9]\+\.[0-9]\+" -o | head -n 1)
open=$(tshark -r $1 | grep "$2.\+$ip_addr.\+\[SYN, ACK\]" | awk '{print
$8}' | sort -n | uniq)
attempted=$(tshark -r $1 | grep "$ip_addr.\+$2.\+\[SYN\]" | grep -oP
'[0-9]+(?= \[SYN\]+)' | sort -n | uniq)

printf "%12s\t%8s\t%s\n" "PORT" "STATE" "SERVICE"
for port in $attempted
do
  if [ $(echo "$open" | grep -c "^$port$") -eq 0 ]; then isopen="filtered";
else isopen="open"; fi
  name=$(getent services | grep -oP ".+(?= +$port/tcp)" | head -n 1)
  printf "%8s/tcp\t%8s\t%s\n" $port $isopen $name
done
```

Usage: ./nmap_analysis.sh [pcap_in] [dst_ip]

```
$ ./nmap_analysis.sh 21200218_nmap.pcap 192.241.168.54
        PORT         STATE    SERVICE
```

```
     21/tcp      filtered    ftp
     22/tcp          open    ssh
     23/tcp      filtered    telnet
     25/tcp      filtered    smtp
     80/tcp          open    http
    110/tcp      filtered    pop3
    139/tcp      filtered    netbios-ssn
    443/tcp          open    https
    445/tcp      filtered    microsoft-ds
   3389/tcp      filtered    ms-wbt-server
```

4. Write a script/program in a language of your choice to perform a simple TCP connection to a series of ports, specified from command line as a comma delimited list.

Cut and paste the TCP connection program/script below:
tcp_ports.sh

```
#!/bin/sh

ipdst=$1
ports=$(echo $2 | tr , '\n')

for port in $ports; do
  nc $ipdst $port <<<'' >/dev/null -w 1
done
```

Usage:./tcp_ports.sh [dst_ip] [port1,port2,...]

5. Putting it all together, write a script that uses tshark (or equivalent) and your programs/scripts from Part 3 and 4 above to perform the function of nmap, without ever running nmap. Your script should accept 2 inputs: target and a comma delimited list of ports to scan.

Cut and paste the analysis program/script below:

```
#!/bin/sh

target="$(dig +short $1)"
ports="$2"

tcpdump -w tmp.pcap 2>/dev/null &
tdpid=$!
sleep 1 # sleep to allow tcpdump to initialize
```

```
        # see: https://askubuntu.com/a/746061/433872
./tcp_ports.sh $target $ports
kill $tdpid
wait $tdpid

./nmap_analysis.sh tmp.pcap $target

rm tmp.pcap
```

Run your script using target="github.com" and ports="21,22,23,25,80,443" and cut and paste the output below:

```
$ ./nmap_copy.sh github.com 21,22,23,25,80,443
        PORT         STATE    SERVICE
      21/tcp      filtered    ftp
      22/tcp          open    ssh
      23/tcp      filtered    telnet
      25/tcp      filtered    smtp
      80/tcp          open    http
     443/tcp          open    https
```