Scanning

1

Scanning

- You have data of the target from Step 1
  - These data mainly include our collection of Internet protocol (IP) addresses (or emails)
  - One of the final steps in reconnaissance was to **create a list of IP addresses** that both belonged to the target and that we were authorized to attack
  - **Step 1**: we mapped our gathered information to attackable **IP address**
  - **Step 2**: we will map IP addresses to **open ports and services**

2

Goals of Scanning

- Learn more about targets and find opening
  - ❖Determine IP address of live hosts, firewalls, etc.
  - ❖Determine network topology of the target environment
  - ❖Determine the OS types of the discovered hosts
  - ❖Determine open ports and network services
  - ❖Determine a list of potential vulnerabilities

3

3

Scan Types

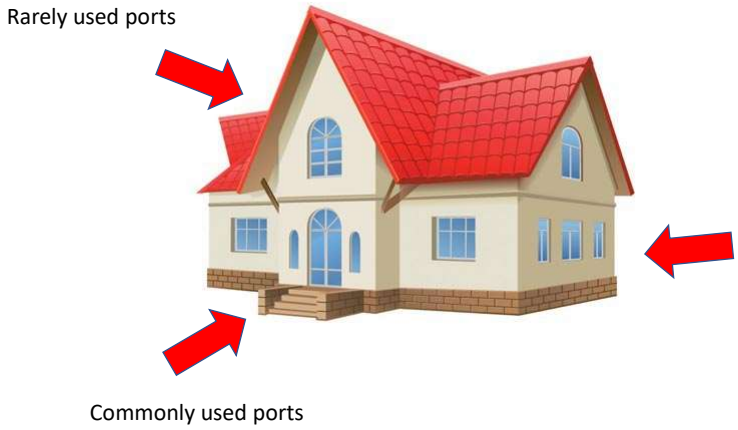
- Network Sweeping
  - ❖Identify live hosts at IP address in the target network
- Network Tracing
  - ❖Determine network topology
- Port Scanning
  - ❖Determine listening TCP and UDP ports
- OS Fingerprinting
  - ❖Determine OS type of discovered live hosts
- Version Scanning
  - ❖Determine the version of services and protocols spoken by the open ports
- Vulnerability Scanning
  - ❖Misconfigurations, unpatched services, etc

4

4

Port

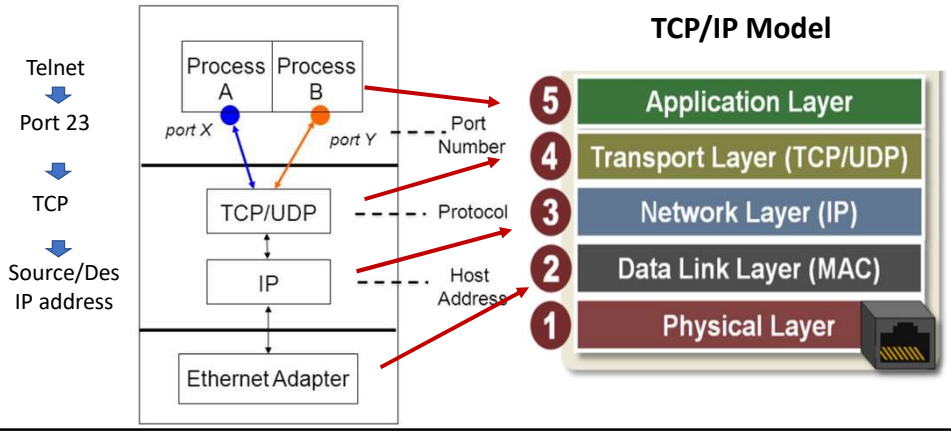
- House (**computer**) with multiple potential entry points (**ports**)



6

Communication on layers

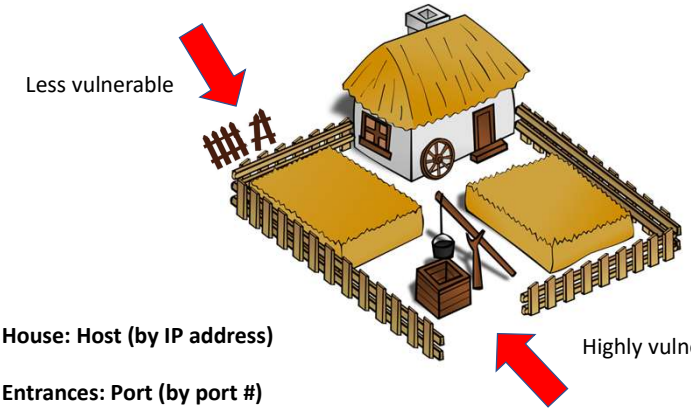
- Transport layer: **Process-to-process** communication
- Network layer (IP): **Host-to-host** communication



7

Vulnerability

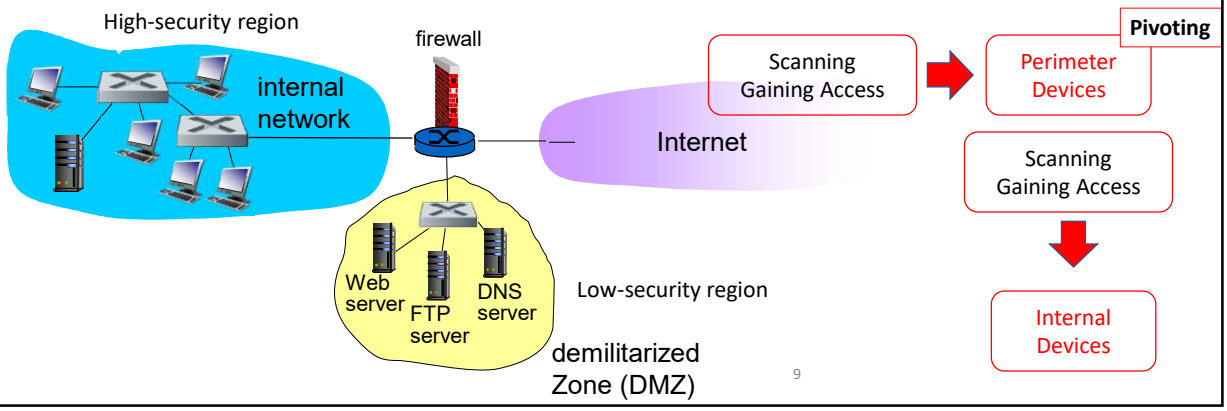
- Difference in the severity of various vulnerabilities
  - Little opportunity or complete take-over



8

Perimeter Devices

- **Perimeter devices** are the starting point of scanning
  - Firewalls, routers, servers, computers
  - Most of the information we have from step 1 belongs to perimeter devices



9

9

TCP and UDP Ports

- There are 65536 TCP and 65536 UDP ports
  - ❖Well-known ports (system ports) 0-1023 -> commonly used services
  - ❖Registered ports (user ports) 1024-49151 -> registered application (MS SQL server, OpenVPN, etc.)
  - ❖Dynamic ports 49152-65535 -> local, private, temporary use

10

10

Frequently Used TCP Ports

- Some popular TCP ports
  - ❖**21 ftp**
  - ❖**22 ssh**
  - ❖**23 telnet**
  - ❖**25 smtp**
  - ❖**80 http**
  - ❖**110 pop3**
  - ❖**135, 137, 139 NetBIOS**
  - ❖143 IMAP
  - ❖161 SNMP
  - ❖**443 https**
  - ❖**445 smb**
  - ❖**3389 RDP**

11

11

Frequently Used UDP Ports

- Some popular UDP ports
  - **53 dns**
  - 69 tftp
  - 123 ntp
  - 2049 nfs

Handling Large Scans

- Consider a pen test project to scan 1000 hosts, all TCP and UDP ports
- If it took 1s for each port, the scan alone would take (65536 x 1000 X 2)
  - 131 million seconds = 4.15 years
- Even if we scan 100 ports at a time, it would still take several days
- There must be a better way

Tips to Handle Large Scans

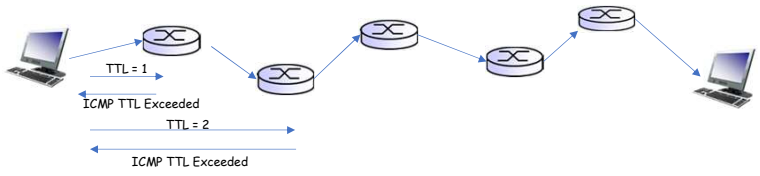
- Sample a subset of target machines
- Sample a set of ports
  - ❖Look for those most frequently used ports as shown earlier. How about other ports?
- Send packets much more quickly
- Review network firewall ruleset
- Alter firewall rules for closed ports
  - ❖You alter firewall rules so that you can scan fast?

14

14

Traceroute

- Discovers the route that packets take between two hosts
- Sends packets to target w**ith varying TTLs in the IP header**
- Linux traceroute
  - ❖Sends UDP packets with incrementing destination ports starting at 33434, going up by one port for each probe sent
  - ❖**-I:** use ICMP Echo Request instead of UDP
  - ❖**-T:** use TCP SYN instead of UDP with default destination port 80
  - ❖**-n:** don't resolve domain names



16

16

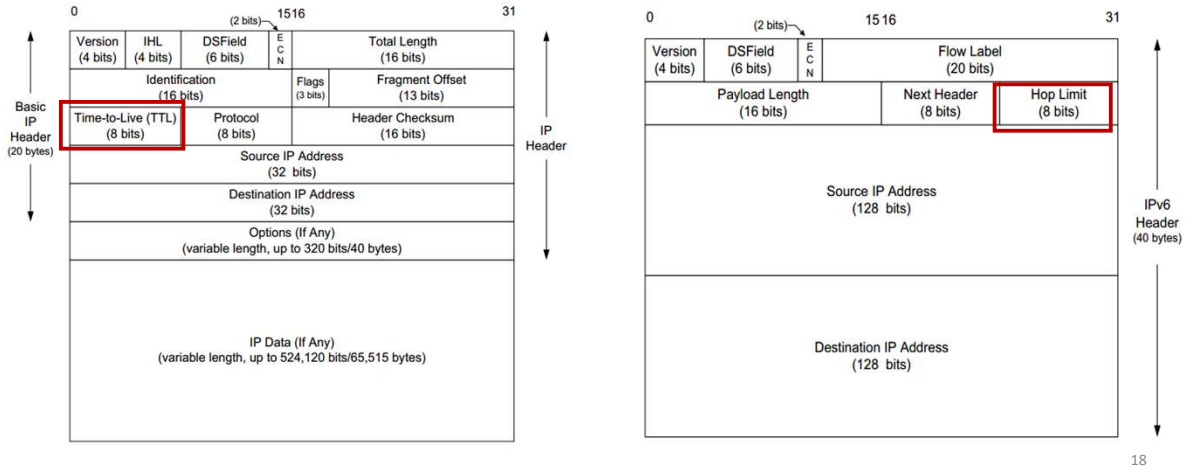
TTL (Time-To-Live, Packet lifetime)

- The TTL field is 8-bits long and indicates how many hops this packet can travel
- **Decrement by 1 at each router while hopping**
- Packet dropped once TTL becomes 0
- The last router sends a "TTL Exceeded in Transit" message back to the source IP address of the discarded packet. The source address of this ICMP message is the router itself
- Windows OS: 128 (default value)
- Linux OS: 64

17

17

IPv4 vs Ipv6 Header



18

18

Traceroute

```
┌──(kali㉿kali)-[~]
└─$ sudo traceroute -I 8.8.8.8
[sudo] password for kali:
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.795 ms  0.689 ms  0.673 ms
 2  wcm-gw.ucmo.local (153.91.155.254)  1.056 ms  1.035 ms  1.021 ms
 3  core1-wcm.ucmo.local (153.91.21.57)  0.994 ms  0.974 ms  1.334 ms
 4  ucmo-kc-bdi999.gw.more.net (150.199.100.241)  2.722 ms  2.888 ms  3.042 ms
 5  kc-dist-01-te0-1-0-18-99.mo.more.net (150.199.100.225)  4.173 ms  4.334 ms  4.462 ms
 6  kc-core-01-he0-0-0-0.mo.more.net (150.199.7.161)  4.629 ms  3.182 ms  3.135 ms
 7  ks-96-xe-1-1-1-741.greatplains.net (164.113.254.241)  3.863 ms  3.792 ms  3.731 ms
 8  AS15169.micemn.net (206.108.255.141)  17.926 ms  17.912 ms  17.899 ms
 9  108.170.243.225 (108.170.243.225)  19.284 ms  19.269 ms  19.248 ms
10  142.251.60.201 (142.251.60.201)  18.111 ms  18.516 ms  18.501 ms
11  dns.google (8.8.8.8)  18.480 ms  18.559 ms  18.193 ms
```

- **-w[N]:** wait for N seconds before giving up and writing a *
- **-6:** force use of IPv6
- **-p [port]:** set the destination port for probes
  - For UDP, set the base destination UDP port and increment
  - For TCP, set the **fixed** TCP destination port to use, default to port 80 (no incrementing)

- If a given hop doesn't return an ICMP TTL Exceeded in Transit message back (because it's configured to filter the inbound probe or omit the ICMP response), traceroute simply label that hop with a *, meaning that no address information is known for it.

- If a given network device filters all ICMP messages going back, its hop and everything thereafter will be filled with a *

19

19

Windows Tracert

- Sends ICMP Echo Request message to target
  - **-d:** don't resolve domain names
  - **-w[N]:** wait for N milliseconds before giving up and writing a *
  - **-6:** force use of IPv6

20

20

tracert wix.com

```
C:\Users\noahm>tracert wix.com

Tracing route to wix.com [185.230.60.164]
over a maximum of 30 hops:

  1     *        *        *     Request timed out.
  2     1 ms     1 ms     1 ms  cswitch1-t1-7.umkc.edu [10.255.1.9]
  3     2 ms     1 ms     1 ms  10.254.255.6
  4    54 ms     2 ms     2 ms  pa2-te1-22.umkc.edu [134.193.126.43]
  5    83 ms     4 ms   213 ms  igate.umkc.edu [134.193.126.145]
  6     8 ms     6 ms     6 ms  umicn-cn-te0-1-0-0-105.um.more.net [150.199.4.61]
  7     6 ms     6 ms     6 ms  umicn-kc-te0-1-0-0-100.um.more.net [150.199.4.42]
  8     6 ms     6 ms     5 ms  kc-gw-be20.mo.more.net [150.199.4.122]
  9     6 ms     6 ms     6 ms  kcb-b1-link.telia.net [213.248.91.77]
 10    18 ms    17 ms    17 ms  dls-b22-link.telia.net [62.115.125.159]
 11    34 ms    34 ms    34 ms  ldl-b22-link.telia.net [62.115.120.112]
 12    51 ms    51 ms    51 ms  ash-bb4-link.telia.net [62.115.125.129]
 13    50 ms    50 ms    50 ms  ash-b1-link.telia.net [213.155.136.39]
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16    47 ms    48 ms    54 ms  185.230.60.164
```

Packet loss

RTT

Missouri Columbia

Texas Dallas

Georgia Atlanta

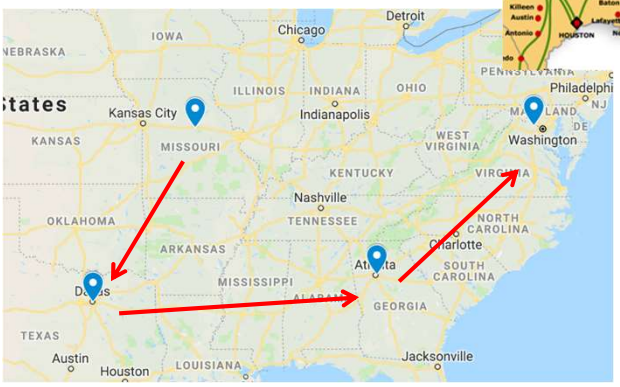
Wix.com Ltd.

Virginia Ashburn

21

tracert wix.com

- Geolocations of Routers



22

Web Based Traceroute Service

- www.traceroute.org
- Very useful in seeing if you are being shunned during a test
- Use IP addresses of target instead of domain names
- Consider the information you might be leaking to the third party

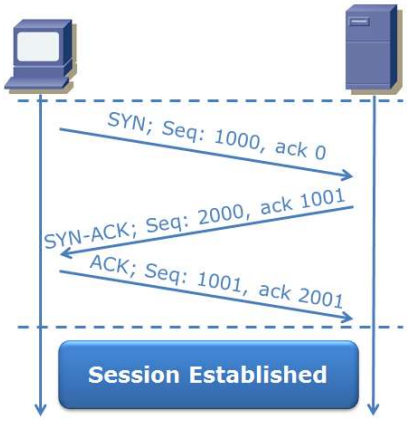
- https://geotraceroute.com/
- With map view

23

23

TCP Ports Scanning

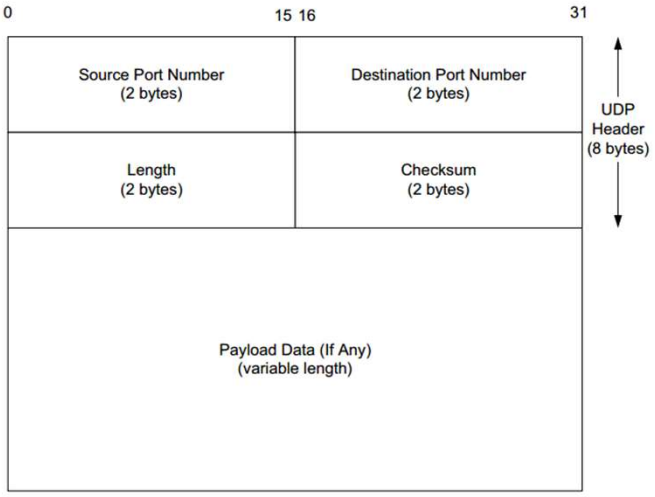
- We can figure out a specific TCP port's condition by scanning the TCP ports



24

24

UDP Header



25

25

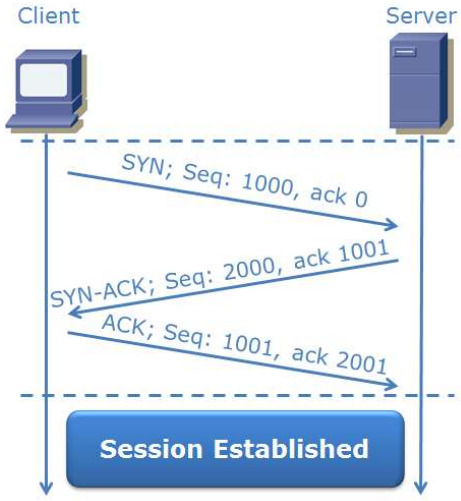
TCP Vs UDP

- TCP is a connection oriented protocol
  - ❖Communication can start only after a successful three-way handshaking
  - ❖Resend packets if lost
  - ❖Detect out of order packets and re-sequence them
- UDP is connectionless protocol
  - ❖Make no attempts to associate streams of packets together
  - ❖Each packet is completely independent and unrelated to other packets
  - ❖No attempt is made by UDP for retransmission or resequencing

26

26

TCP Three-Way Handshake

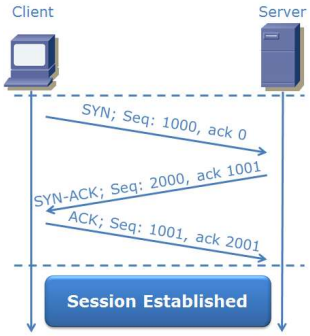


27

27

TCP Ports Scanning

- According to RFC 793, if something is listening on a TCP port and a SYN packet arrives on that port, the system responds with a SYN-ACK regardless of the payload of the SYN packet
- In other words, **if we receive a SYN-ACK in response**, we can reliably conclude that the port is listening

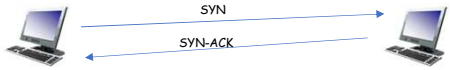


28

28

TCP Scanning Behaviors

- **Case 1:** send SYN packet in, get SYN-ACK back
  - ❖Conclusion: port is **open**



- **Case 2:** send SYN packet in, get RST-ACK back
  - ❖Conclusion: port is **closed** (or a firewall blocks it)

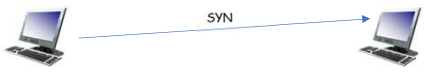
29

29

TCP Scanning Behaviors

- **Case 3:** send SYN packet in, ICMP unreachable back
  - ❖Conclusion: port is not accessible, likely blocked by a firewall (on network or end system). Nmap reports this case as "**filtered**"

- **Case 4:** send SYN packet in, nothing back
  - ❖Conclusion: port is not accessible, likely blocked by a firewall (on network or end system). Nmap reports this case as "**filtered**"



30

30

Extra Notes on TCP Scanning

- There are usually many more closed ports than open ports
- If nothing comes back, the scanning tool has to wait for a timeout to expire before moving on to the next port
- If we can get RSTs or ICMP unreachable packets back, the scan will proceed much faster

31

31

UDP Ports Scanning

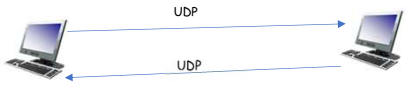
- Less options for scanning
- Much slower
- Less reliable

32

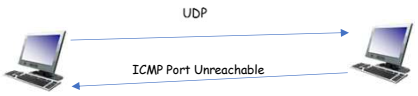
32

UDP Scanning Behaviors

- **Case 1**: send a UDP packet in, get a UDP packet back
  - ❖Conclusion: port is **open**



- **Case 2**: send a UDP packet in, get a ICMP port unreachable back
  - ❖Conclusion: port is **closed** (or a firewall blocks it)



33

33

UDP Scanning Behaviors

- **Case 3**: send a UDP packet in get nothing back
  - ❖Conclusion: the port is not accessible, why?
  - ❖Possible reasons
    - Port is closed
    - Firewall is blocking UDP probe packet
    - Firewall is blocking outbound response
    - Port is open. However, the service is looing for specific data in UDP payload. Without payload data in the probe packet, the target port silently drop the probe packet
  - ❖In other words, we do not know. Nmap reports this case as "**open | filtered**"

34

34

Nmap

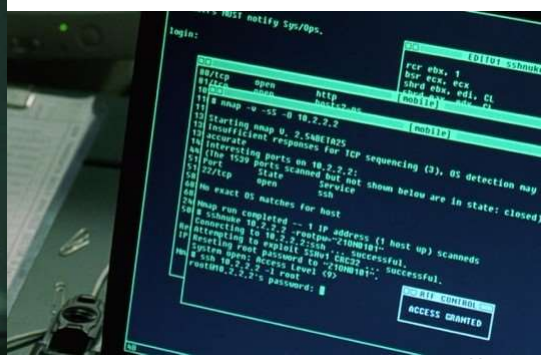
- Written and maintained by **Fyodor**
- Nmap is primarily a port scanner showing which TCP and UDP ports are open on a target machine
- www.nmap.org

35

35

Nmap, Trinity @Matrix

- Saving human race using Nmap...

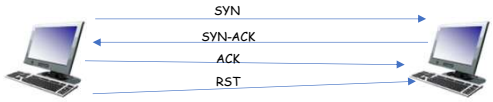


36

36

TCP Port Scan Type: -sT

- **-sT:** connect scan
  - ❖Completes three-way handshake. Connection then torn down using RST
  - ❖SYN → SYN-ACK → ACK → RST
  - ❖Slower
  - ❖More likely to be logged
  - ❖Can run with or without UID 0

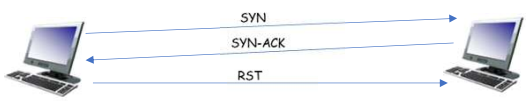


37

37

TCP Port Scan Type: -sS

- **-sS:** SYN Stealth scan or half-open scan
  - ❖Do not complete the three-way handshake
  - ❖SYN → SYN-ACK → RST
  - ❖Often not logged on the target system since there is no connection
  - ❖Firewalls, IDS and IPS tools may still detect it
  - ❖Requires UID 0 privilege to run
  - ❖If you do not specify an Nmap scan type and are running Nmap as root, the -sS SYN Stealth scan is the scan that Nmap defaults to



38

38

TCP Header, control flags (9bit)

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																								Urgent pointer (if URG set)							
20 ...	160 ...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.) ...																															

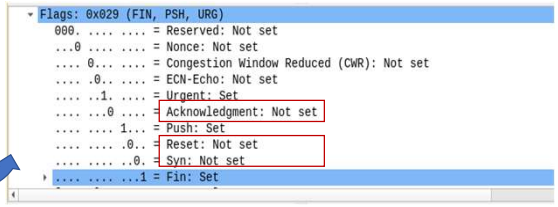
TCP Control Bits

- **SYN:** this bit is used during session establishment
- **ACK:** packets with this bit set to 1 are acknowledging earlier packets
- **RST:** the connection should be reset due to an error or other interruptions
- **FIN:** used to graceful torn down the TCP connection

Other TCP Scan Types

- **-sA**: ACK Scan
  - ❖Help get through certain kinds of packet filters
  - ❖Different systems respond in different ways to an unsolicited ACK. A response does indicate there is a system at the address
  - ❖**It is useful for identifying hosts. Unreliable to tell us whether port is open or closed**
- **-sF**: FIN Scan
  - ❖Set FIN bit for all scan packets
- -**sN**: NULL scan
  - ❖Set all control bits to 0
- -**sX**: Xmas tree scan
  - ❖Set FIN, PSH and URG bits

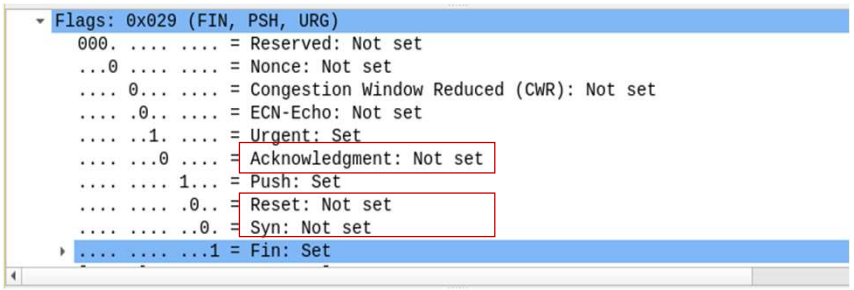
Packet flag manipulation scan



41

Nmap --scanflags

- Users can also specify arbitrary control bits setting by using **--scanflags**
  - ❖[URG|ACK|PSH|RST|SYN|FIN|ECE|CWR|ALL|NONE]
  - ❖# nmap --scanflags FINPSHURG -p 443 192.168.1.81
  - ❖Above is a Xmas tree scan (-sX)



43

43

Nmap UDP Scan

- **-sU:** UDP Scan
- Send UDP packets without payload to target for most ports
  - ❖Send a protocol specific payload only to over a dozen most common UDP services
  - ❖All other UDP ports have blank payload
  - ❖How about a common UDP service listening on an unusual port?
- Nmap attempts to detect response ICMP rate in target and slows down
  - ❖Linux sends one ICMP Port Unreachable packet per second
  - ❖Take over 18 hours to scan all 65,535 UDP ports

44

44

Nmap Port Scanning

- **By default, Nmap only checks the top 1000 most-used ports for TCP and UDP**
- The nmap-services file indicate the ranking of the most used ports
- Use the –**F flag** to scan the top 100 ports
- Use **--top-ports [N]** to scan for the N most used ports

45

45

Nmap -p

- The -p can indicate a single port, port range or port list. The flags T: and U: can be used in the list to specify TCP or UDP ports
  - -p 80
  - -p 1-600
  - -p 21,22,23,80,443
  - **-p T:21,22,23,U:53**
  - -p -65535
  - **-p**– (scan every ports)
- Ports are scanned in random order, use –r to make Nmap to scan linearly in increasing port order.

46

46

Nmap-Scanning Speed

- By default, Nmap has a dynamic timing model
  - Adapts scan timeouts based on performance of initial packets
- In addition, Nmap can be invoked with various options for scan speed by using the –T flag
  - -T0: paranoid: scan serially, send a packet every 5 minutes
  - -T1: sneaky: scan serially, send a packet every 15 seconds
  - -T2: polite: scan serially, send a packet every 0.4 seconds. Set this mode if you scan a SCADA or systems from long distance (high latency)
  - -T3: normal: scan in parallel, the default mode for Nmap
  - -T4: aggressively: scan in parallel, wait only 1.25 seconds for probe response
  - -T5: insane: scan in parallel, wait only 0.3 seconds for probe response, spend up to 15 minutes per target

47

47

Nmap Output Formats

- **-oN [filename]:** store output in normal format. The result you see on the screen
- **-oG [filename]**: store output in greppable format (deprecated, still popular)
- **-oX [filename]:** store output in XML format
- **-oA [filename]:** store in all three major formats (normal, greppable and XML), using filename.nmap, filename.gnmap and filename.xml

48

48

Nmap Host Probing

- **By default, Nmap probes a host before scanning it**
  - ❖For UID 0 users, Nmap sends
    - ARP request if the target is on the same subnet as the Nmap box
    - If the host is on a different subnet, sends ICMP Echo Request, ICMP Timestamp Request, TCP SYN to port 443 and TCP ACK to port 80
  - ❖For non-UID 0 users, Nmap sends
    - TCP SYNs to port 80 and 443
- Invoke Nmap with the **–Pn flag** will make Nmap to skip the probe phase (assuming the host is up)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.128	153.91.1.10	TCP	74	57414 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA
2	0.000097803	10.0.2.128	153.91.1.10	TCP	74	47504 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S
3	0.001177862	153.91.1.10	10.0.2.128	TCP	60	80 → 57414 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
4	0.001177887	153.91.1.10	10.0.2.128	TCP	60	443 → 47504 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
5	0.001203986	10.0.2.128	153.91.1.10	TCP	54	57414 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
6	0.001249443	10.0.2.128	153.91.1.10	TCP	54	47504 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
7	0.001280710	10.0.2.128	153.91.1.10	TCP	54	57414 → 80 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
8	0.001349660	10.0.2.128	153.91.1.10	TCP	54	47504 → 443 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0

49

49

Nmap Network Sweep

- Besides probing an individual host before port scanning, Nmap can be used to probe for target hosts (network sweeping) to identify where hosts are located in a target network address range

- # nmap –**sP** [options] (*e.g*: **nmap –sP 192.168.84.1-255**)
- # nmap –**sn** [options] (**host discovery only, disable port scan**)
  - ❖Do not confuse with the Null scan -sN

50

50

Nmap OS Fingerprinting

- **-*O***: Nmap will perform a port scan before OS fingerprinting

- Nmap uses active OS fingerprinting by sending packets out to measure target OS type

- Other tool such as p0f supports passive OS fingerprinting

53

53

Passive OS Fingerprinting

- Other tool such as p0f supports passive OS fingerprinting
- P0f is a sniffer and a database for determining system type
- **Based on packets sniffed from the line**, it determines the type of system that generated the packet
- Same false matches, but at least 90% success rate
- One caveat: everything behind a proxy will look like the proxy itself

- # p0f –s test.pcap

54

54

Nmap Version Scanning

- When Nmap identifies an open port, it displays the default service commonly associated with that port
  - ❖Consulting the **nmap-services-probes** file (about 2,200 services)
- What about services are on ports not in that list?
- What about an admin to configure a service to listen on an unusual port? For example web server on TCP 8080
- What service and protocol version the listening service is using?

55

55

Nmap Version Scanning

- For each listening port discovered in the port scan
  - ❖Make a TCP connection to and listen for 6 seconds. If response with match, done!
  - ❖Send probes to TCP and UDP ports, send data designed to elicit a response to determine service type
  - ❖Attempts SSL handshake over TCP ports. If successful, probe over SSL connection

56

56

Nmap Version Scanning: -sV

- **-sV: version scan**
  - ❖Nmap conducts a port scan before the version scan
  - ❖Use nmap-service-probes file to probe and match
- **-A: -A = -O + -sV + -sC + --traceroute**
- --version-trace flag shows the details of version probes

57

57

Other Useful Scan Flags

- **--packet-trace**
  - To display summary of each packet Nmap sends or receives
- **--reason**
  - Tell the reason why Nmap classifies a given port's open|closed|filtered state
- **-f**
  - Fragment the scan packets into 8 bytes. Might be useful to fool the firewall. To specify your own fragment size use --mtu, the size must be a multiple of 8 bytes
- **-D**
  - Causes a decoy scan to be performed together to hide your IP address
  - # nmap –n –**D RND:10** Target IP Address
    - Generates a random number of decoys
  - # nmap -D **decoy-ip1,decoy-ip2** Target IP Address

58

58

Nmap Scan for IPv6

- Many firewall and IPSs do not block IPv6 traffic
- IPv6 is auto configured on most Windows, Linux, macOS and other devices
- IPv6 has 128 bits. 8 groups of 4 hex digits separated by :
- :: means that the given bits should be populated with all zeros
- You can use :: in address only one time
- Local loopback address is ::1
- Use flag -6 to invoke Nmap to support IPv6

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.128  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::20c:29ff:fe40:f9fa  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:40:f9:fa  txqueuelen 1000  (Ethernet)
        RX packets 24  bytes 3400 (3.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1328 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

59

59

Nmap Scan for IPv6

- Locate targets
  - $ ping6 –I eth0 ff02::1 (multicast address for all link-local IPv6 hosts)
  - $ ping6 –I eth0 ff02::2 (multicast address for all link-local IPv6 routers)
- Look at neighbors
  - $ ip neigh
- Scan using Nmap
  - $nmap –Pn -6 ipv6_address%eth0