Testing for vulnerabilities

Port scanning

PURPOSE

- Identify the status of scanned ports (port scan)
  - Open – there is an active service that supports connections
  - Closed – there is no active service that supports connections
  - Filtered – the port is filtered by a firewall
- Identification of active services on systems (service scan)
  - FTP, HTTP, SMTP, …

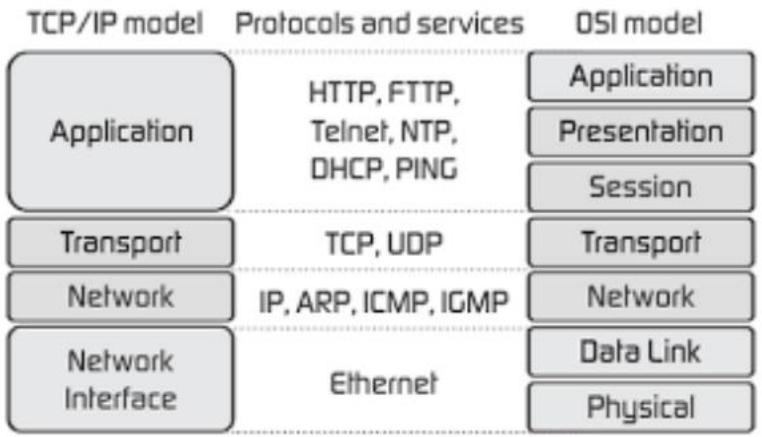
PURPOSE

- The most important step after gathering information
  - In this phase, active and/or vulnerable services are discovered
- Provides important information about the role of each system in the network
- It involves identifying both open ports and the services running on them
- Port scan vs. port sweep
  - Port scan = scans a lot of ports on a given system
  - Port sweep = scans a given port on a lot of systems

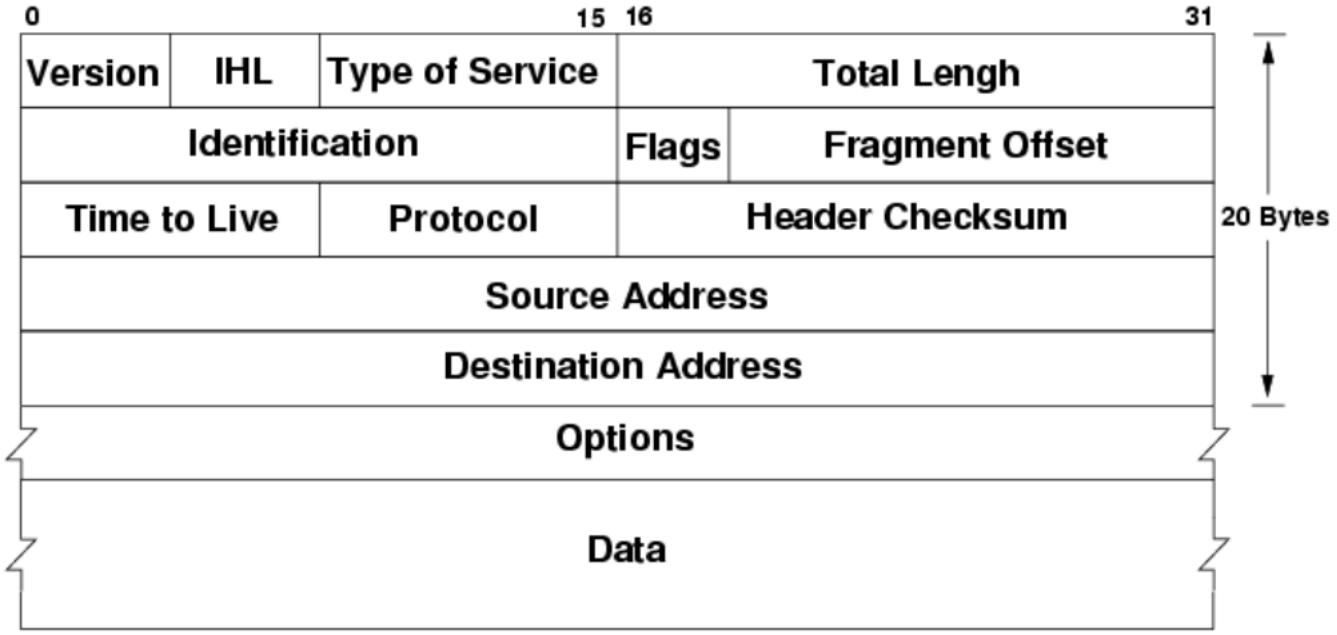
TCP/IP - review

OSI Model	Protocols
Application Layer	DNS, DHCP, FTP, HTTPS, IMAP, LDAP, NTP, POP3, RTP, RTSP, SSH, SIP, SMTP, SNMP, Telnet, TFTP
Presentation Layer	JPEG, MIDI, MPEG, PICT, TIFF
Session Layer	NetBIOS, NFS, PAP, SCP, SQL, ZIP
Transport Layer	TCP, UDP
Network Layer	ICMP, IGMP, IPsec, IPv4, IPv6, IPX, RIP
Data Link Layer	ARP, ATM, CDP, FDDI, Frame Relay, HDLC, MPLS, PPP, STP, Token Ring
Physical Layer	Bluetooth, Ethernet, DSL, ISDN, 802.11 Wi-Fi

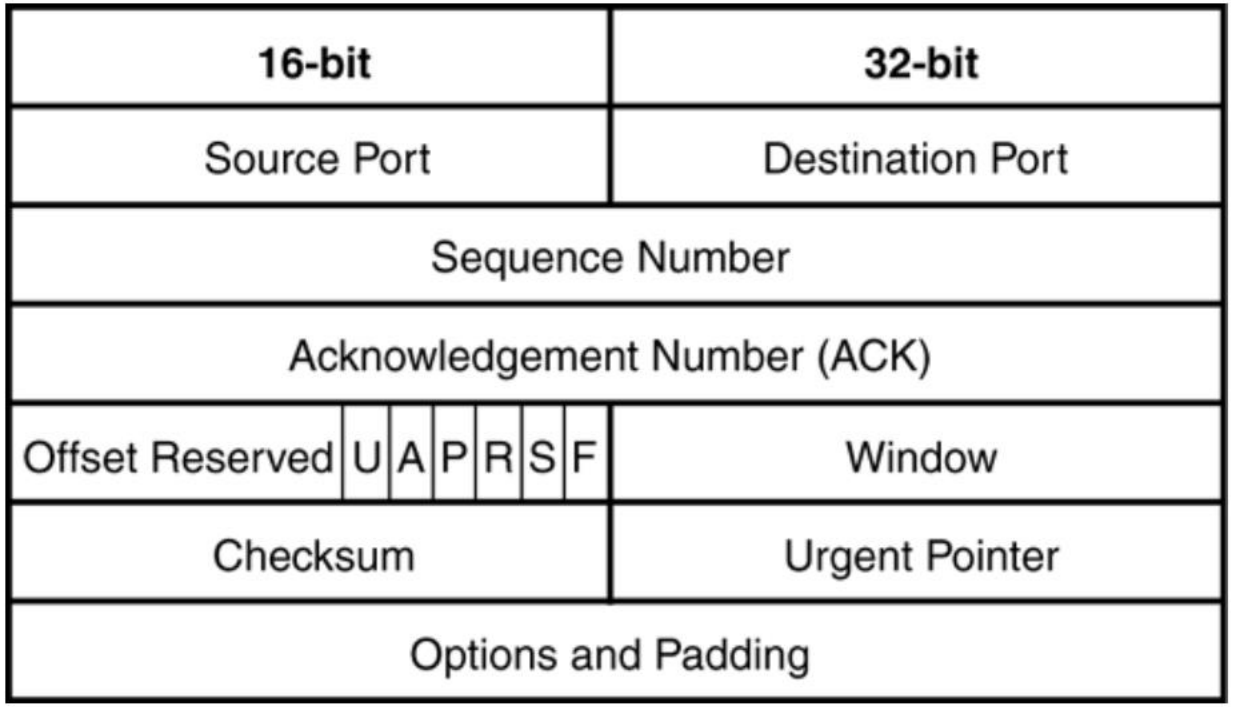
TCP/IP - review



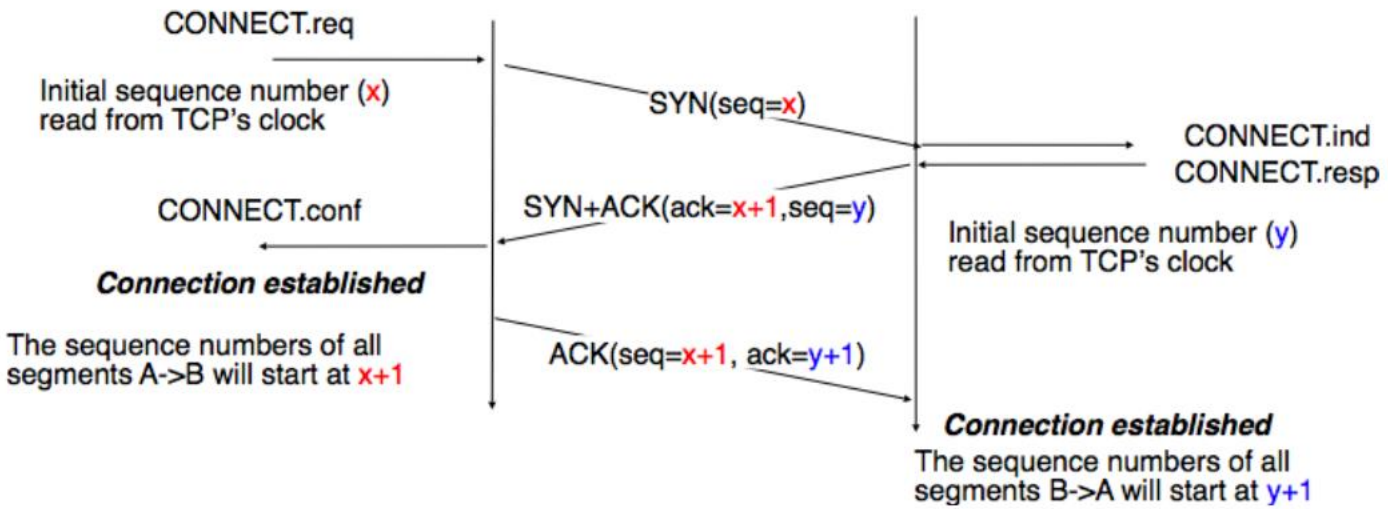
TCP/IP – review (IPv4 header)



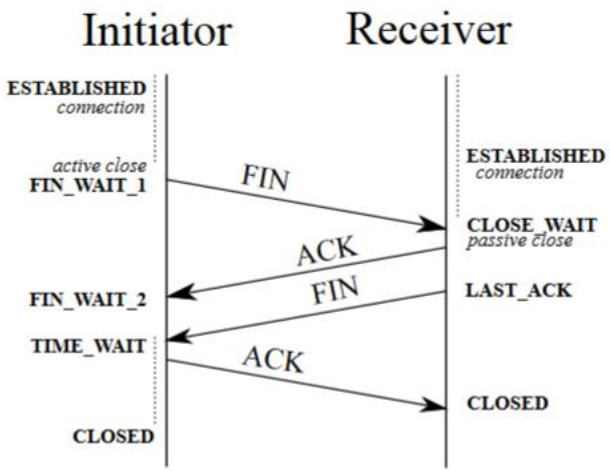
TCP/IP – review (TCP header)



TCP/IP – review (TCP open a connection)

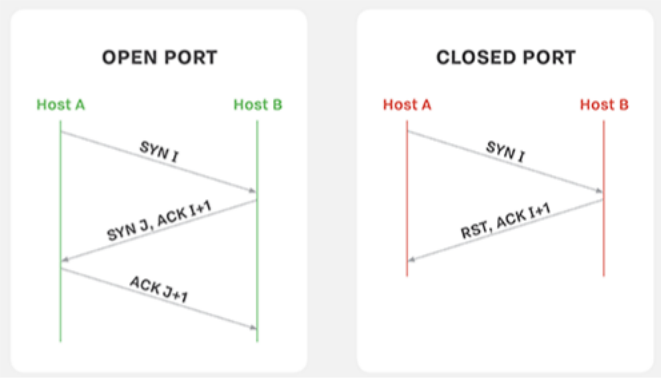


TCP/IP – review (TCP close a connection)



Port scanning techniques – SYN scan

- Send a packet with SYN set
- Fast, can scan thousands of ports/second
- Relatively stealth
- "Half open scanning" – does not open a full connection, but only sends a SYN:



- filtered port: no response is received (ICMP unreachable)

Port scanning techniques – Connect scan

- Connect scan
  - Unlike SYN scan, a full connection opens
  - Doesn't need special privileges to send raw packets
  - Relies on the operating system to establish the connection
  - Takes much longer
  - Easy to catch IDSs
  - Preferably SYN scan when possible

Port scanning techniques – NULL scan

- NULL scan, FIN scan, Xmas scan
  - NULL scan: send a packet with all 0 flags
  - FIN scan: sending a packet with FIN set
  - Xmas scan: send a packet with FIN, PSH, URG set
  - Sometimes more stealthy than SYN scan
  - Functionally, they are identical

Port scanning techniques – ACK scan

- ACK scan
  - Sending a packet with ACK set
  - Determine firewall behaviors
  - It does not determine whether a port is open/closed, but whether it is filtered/unfiltered

Port scanning techniques – maimon scan

- Maimon scan
  - Named after its inventor (Maimon, Phrack #49, '96)
  - Same as NULL, FIN, Xmas, but uses FIN/ACK
  - Response to a maimon scan:
    - open port: no response
    - closed port: RST

Port scanning techniques – window scan

- Window scan
  - Use the Window field to differentiate between open and closed ports
    - Otherwise, it's identical to ACK scan
  - It is based on certain implementations of TCP
  - Not 100% reliable
  - Response to window scan
    - open port: window > 0
    - closed port: window == 0

Port scanning techniques – UDP scan

- UDP scan
  - Many services use the UDP protocol
  - DHCP, DNS, SNMP, …
  - Slow
    - however, It must be considered

SERVICE IDENTIFICATION TECHNIQUES

- An open port is useless, if we don't know what service runs on it
- The step of identifying services and versions is critical!
- Information that can be extracted:
- Protocol (FTP, SMTP, HTTP, etc.)
- Application (Apache, ProFTPd, Postifx, etc.)
- Version
- Pc name
- Device type
- Operating System

SERVICE IDENTIFICATION TECHNIQUES

- nmap results

```
# nmap -sV -p21,22,80 10.10.10.XX
Nmap scan report for 10.10.10.XX
Host is up, received user-set (0.045s latency).
Scanned at 2017-12-08 16:07:54 EET for 827s

PORT   STATE SERVICE  VERSION
21/tcp open  ftp      vsftpd 3.0.3
22/tcp open  ssh      OpenSSH 7.5 (protocol 2.0)
80/tcp open  http     Apache httpd 2.4.27 ((Unix))
```

SERVICE IDENTIFICATION TECHNIQUES

- Nmap automatically identifies services running on a specific port
  - It can also identify other information, such as the operating system
- Sometimes, "manual identification" can be used, by reading the banner

```
# nc -nv 10.10.10.XX 2222
(UNKNOWN) [10.10.10.XX] 2222 (?) open
SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.2
```

- Sometimes manual identification or nmap doesn't help
  - Custom protocols/services
  - Not much you can do
  - Except reverse-engineering

SERVICE IDENTIFICATION TECHNIQUES

- Identifying the operating system
- Nmap can automatically identify the operating system
- Sends TCP/UDP packets and inspects every bit of the response
- Each OS has its own peculiarities in the TCP/IP stack
- Sometimes direct, from services

```
# curl -i http://10.10.10.XX
HTTP/1.1 200 OK
Date: Sun, 05 Nov 2017 10:08:52 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Fri, 22 Sep 2017 20:01:19 GMT
ETag: "89-559ccac257884"
Accept-Ranges: bytes
Content-Length: 137
Vary: Accept-Encoding
Content-Type: text/html

<!DOCTYPE html>
<html>
<body>
```

AVOIDING DETECTION

- Port scanning is relatively easy to detect by firewalls
- The port-scan sources can be automatically disconnected from the network, banned, etc.
- It is important that the scan is stealth

AVOIDING DETECTION

- Packet fragmentation
  - TCP packets are fragmented
  - Many firewalls don't handle this case (performance…)
  - Some treat this case, making the technique useless
- Decoys
  - For each scanned port, n requests are sent, each request having a spoofed source IP address
  - The victim sees that n+1 systems are scanning him, but does not know what the real source of the attack is
  - It can be mitigated by doing router path tracing or by spoof blocking

AVOIDING DETECTION

- Spoofing
  - Another source is used for scanning; example: FTP Bounce Back
- Proxy
  - Scanning is done through one or more proxies
  - The victim sees that he is scanned by the last proxy in such a chain
- Other techniques
  - Invalid checksum, MAC spoof, modified TTL, data appendage to sent packets, etc.

MITIGATIONS

- There is no real solution to prevent the porta-scan…
- Most legal entities consider a port scan action to be legal
- … unless we use it to operate a service
- Stop any service you don't need
  - An HTTP server does not ideally need an FTP server
- Different ”security" tools
  - PortSentry, TCP Wrappers, etc.

NMAP

- https://nmap.org
- Some of the most important capabilities are described below
- Target specification
  - It can scan one host at a time, or networks - both by name and by address
  - Targets to be scanned can be provided in an input file
- Host discovery
  - Supports ping scan, scan port, sweep port
  - Supports custom DNS
- Scan techniques
  - Supports all techniques listed above + idle, SCTP, IP, FTP bounce scan
- Port specification
  - Default, scans the 1000 most common ports
  - Custom ranges can be specified

NMAP

- Service/Version detection
  - Automatically detects services and versions running on each port
- Script scan
  - Dedicated scan scripts can be created
- OS detection
  - Detects the operating system
- Timing and performance
  - Supports parallel scanning (multithreaded)
  - Packets throttling
- Firewall/IDS evasion and spoofing
  - Supports all evasion techniques presented above(+)
- Output in multiple formats
  - XML, grepable,

Conclusions

- The most important step after info-gathering
- Most systems are susceptible to port- scan
- There are numerous ways to avoid detection
- The nmap utility is the "Swiss army knife" for port- scan