Module 4: Network Attacks I

14/10/23

techniques attacker will employ prior to attacking a computer n/w.

1) L1 N/w Reconnaissance with Whois (tool) n/w
   ↳ attacker tries to learn characteristics of a given
2) L2 " " "
3) L3 Network Mapping
4) L4 Network scanning Background
5) L5 Nmap Tool

DNS (tool)
eg → what IP address belong to a given n/w
↳ Domain Name service (DNS)

(how attacker can build a topology of a given n/w that he is eventually going to attack)
eg → tool to build rough map of a given n/w

practical tool
(N/w mapping & scanning tool)
↳ Hands-on experience

(techniques needed to scan a given n/w)
+ Background
eg → what hosts in n/w might be alive & what services they might be running

Lecture 1: N/w Reconnaissance with Whois

→ prerequisite required to attack a given n/w.
→ eg: what are the IP-addresses / hosts forming this n/w.
→ steal → "what to steal?"

Registrar

* organization where you register a domain name.
* verifies uniqueness of name.
* Enters domain-name into various databases: whois & DNS public databases
* (who owns the website etc)

→ Attacker can know something about the n/w from databases to maliciously attack the network.

* internic.net / Registrars
→ Godaddy

Whois databases

* I/p → domain name or company name
* output → Registrar whois server (responsible for giving away information regarding this website) dns server (the name server information for that server) (with which website is registered)
1. www.internic.net
   ↳ For com, net and org top-level domains
2. www.allwhois.org
   ↳ For country code top level domains eg. .in, .fr

(Contact info. of personnel governing that n/w)

* Name servers → take charge in providing the mapping of names of servers in this n/w & their corresponding IP addresses.
* wdkin machine in CIS n/w
  whois command web.edu

(Virtual machine)

Reconnaissance: IP Ranges

* ARIN: American Registry for Internet Numbers
  → Maintains whois database that includes IP address range in US.
* RIPE: Europe → web enter
* APNIC: Asia CIDR → n/w domain name system (0.0/16)

* what IP address ranges have been assigned to that n/w?
  [what hosts are alive? or what hosts are part of this n/w?
  what services they might be running?

first figure of IP addresses

↳ (Name-servers, IP-address range etc)

Why whois database needs to be publicly available? → Diagnosis

1) If you are under-attack, can analyze source address of packets.
   → If you are under attack by a n/w, you can do a whois to find what IP addresses are of this n/w. (attacking host)
2) Can use whois database to obtain info about the domain from where the attack is coming.
3) Can inform admin that their systems are source of an attack.

* Also for Administrative purposes

Lecture 2: N/w Reconnaissance with DNS

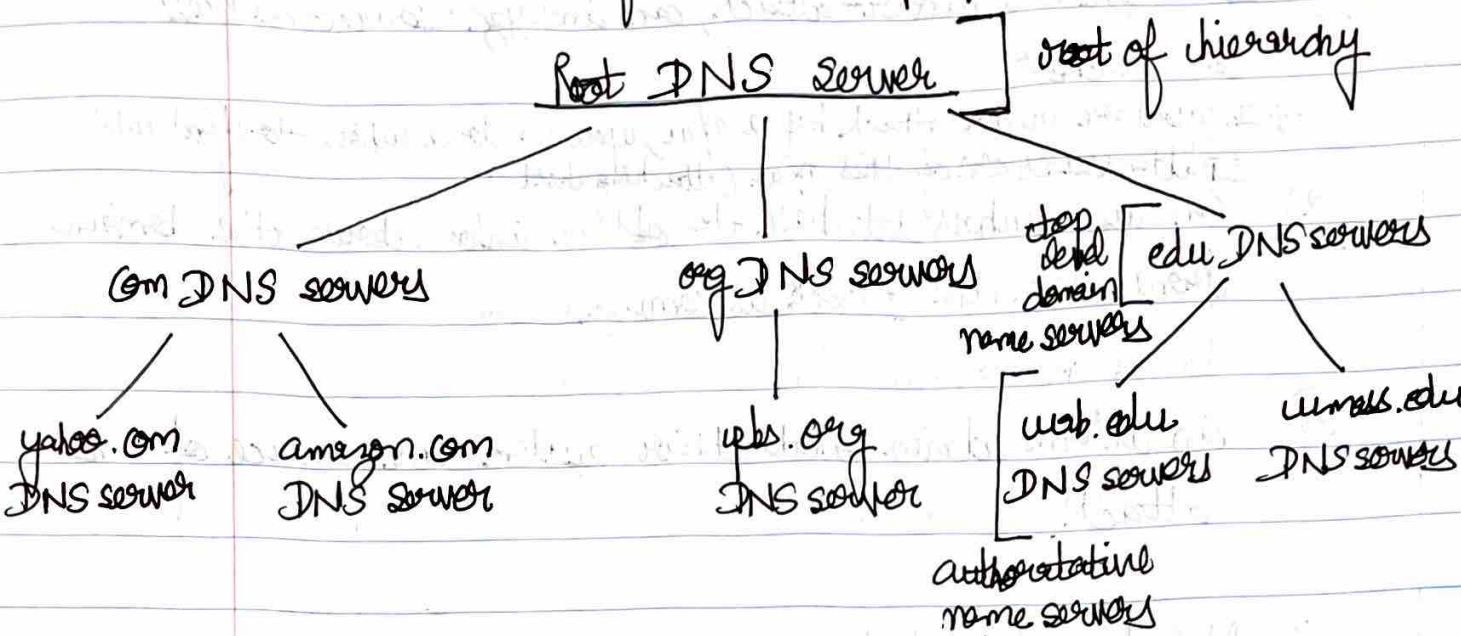
DNS:

* Distributed database implemented in hierarchy of many DNS servers.

Authoritative name server:

* for a given domain (eg web.edu), provides server name to IP address mapping for servers (web, email, ftp, etc) in domain
* Primary & secondary name server for reliability.

Portion of the hierarchy of DNS servers



* TLD → .edu (top-level domain)

DNS records

DNS: distributed database storing resource records (RR)

RR format: (name, value, type, ttl)

↳ time to leave (expiry time for certain record) ↳ after some time cache will remove

- Type = A (address)
  - * name is hostname
  - * value is IP address
- Type = NS
  - * name is domain (eg. foo.com)
  - * value is IP address of authoritative name server for this domain
- Type = MX
  - * value is name of mailserver associated with name.

DNS protocol, messages

* port 53 UDP port

* once the server resolves a query, they will cache the information for a while onto their caches. Next time when you query for this host information, you don't have to query the long procedure.

Caching!

* Improves efficiency of system although has implication on security bcoz if somehow the cache of given data server is hacked/poisoned, attacker could point the hostname google to its own ~~server~~ machine/IP address.

Interrogating DNS servers

* Attacker first gets primary or secondary authoritative server for target organization using whois.
* Attacker can then query the DNS by sending DNS query msgs.
* Tools (often available in Unix & windows machines. also available at websites):
  * nslookup * host * dig
    very popular

1) whois yahoo.com (get all name servers of yahoo) (ns1.yahoo.com)
2) nslookup enter
> server ns1.yahoo.com
Address: 68.180.131.16#53 Address: 2001:4998:130::1001#53
Default server: ns1.yahoo.com

Interrogate this particular server

> set type = any enter → [to learn all info this server has] → [hosts, name-servers, mail-exchange]

> yahoo.com enter

Reconnaissance Summary

* Obtaining information from public databases:

1) whois databases → Tool: web sites
2) DNS " → Tool: nslookup

* Defense

1) keep to a minimum what you put in the public database: only what is necessary.

Lecture 3: N/w Mapping

→ how attacker can build a rough topology of the n/w that he might be interested in attacking.

Goal → learn about a remote n/w.

* path a packet will take from attackers n/w to host within the n/w that he is attacking.
* where the gateways are located?
  firewalls " " ?

N/w mapping → tool

* Attacker often uses traceroute to determine path to each host discovered during ping sweep.

→ overlay results from traceroute to create an approximate n/w diagram.

* If you can repeat this process on multiple hosts within the n/w you can build a rough-map of the n/w.

→ Inbuilt tool in most of the linux systems.

traceroute "starting-point" "Ending-point"

o/p ⇒ entire path that packet will take going from this particular machine to other machine.

main-router

* delay from when you sent query to when you got a response.

3 packets → 1ms 1ms 2ms

↳ next gateway

* If delay measurements ↑ significantly, it means you are passing from a trans-oceanic link

* No response from gateway → (1) congested, (2) slow

* last gateway → machine you are querying too, destination of the traceroute command.

linux (CIS n/w)

$ traceroute uah.edu enter

response back

[path from web to uah.edu]

[138. → uab n/w

[146. → some third party n/w

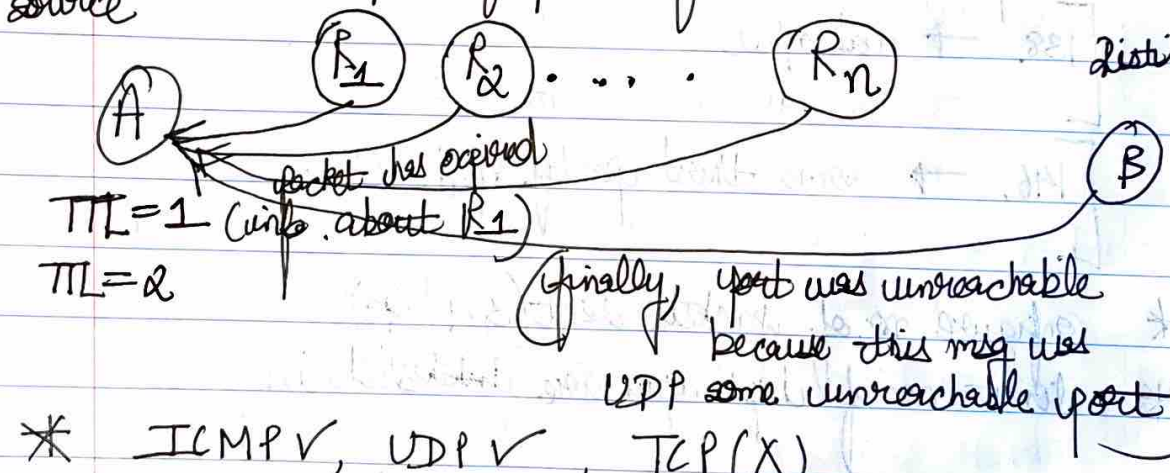
* Configure no. of packets sent (3/4/*)

* limit no. of hops you are interested in

* $ man traceroute (see all settings)

Traceroute: How it works? → Diagnostic property of IP

- ☐ source sends UDP packet to the target
  - * Each to an unlikely unreachable port
  - * 3 packets with the same TTL, then increment TTL
    - TTL → time to live | TTL=1 for first router in path & send over the destination
    - * decrement value & send it to next router on the path
    - learn info about first router | next time TTL=2 & send, first router decrement it to 1
    - ☐ when router decrements TTL to 0, sends back to source & forwards it to next router on the path. next router receives, decrements to 0 sends diagnostic msg back to the source saying packet has expired (learn about information)
      - ICMP packets
      - * type 11, code 0, TTL expired
- ☐ when target receives packet, sends back to source router
  - ICMP packets
  - * type 3, code 0, destination port unreachable
  - * finally it reaches, since port you choose to connect to destination was unreachable, source will receive msg that port is unreachable.
- → All in-built in ICMP (Internet control msging protocol)
- * exploiting benign capability of the ICMP protocol to learn path of packet from source to destination.



- * ICMP ✓, UDP ✓, TCP (X)
- * does not give information on open ports on the destination host.

Lecture 4: N/w Scanning & Background

* What host are alive in a given n/w & what services they are running?

Ping sweep

Ping (to know whether they are alive or not)

- Recall ICMP messages are directly encapsulated in IP datagrams (protocol 1)
- To ping a host:
  - * send ICMP Echo Request (ICMP type 8)
  - * Host responds with ICMP Echo reply (type 0) [If Host is alive]
- So, lets ping the entire IP address range
  - * use automated tools for this ping sweep.
- If firewall blocks ping packets: (no response)
  - * trying sweeping with TCP SYN packets to port 80. (popular, hope the web service might be running)
  - * or try sending UDP packets to possible ports
  - ↳ try opening a TCP/UDP connection to that host. (response back means machine is alive)

(open/closed ports) ← Port Scanning → (what services run on the given host)

- Now that we have a map with some hosts, lets find out what ports are open on a target host.
- 65,535 TCP ports; 65,535 UDP ports (overall 16 bit field therefore)
  - * Web server: TCP port 80 → If this port is open or not
  - * DNS server: UDP port 53 cc
  - * Mail server: TCP port 25 cc (to know these services are running)
- Port Scanning tools can scan:
  - * List of ports
  - * Range of ports
  - * All possible TCP & UDP ports

(to employ specific exploit to specific service to launch an attack)

- Attacker may scan a limited set of ports, to avoid detection.
  → poll each & every port on each & every machine, too much (you may be detected)

Inside TCP segment structure

flag

* RST → respond that you received something that you were not expecting
* SYN → initiate/synchronize a connection
* FIN → complete/finish a connection

TCP seq. #'s & ACKs

seq. #'s	Acks
* "number" of first data packet	* seq. # of next packet expected from other side!
* connection oriented protocol.	

TCP Connection Establishment

Three way handshake:

step 1: client host sends TCP SYN segment to server

- SYN = 1 ACK = 0
- specifies initial seq. # (X)
- no-data

Step2: server host receives SYN, replies with SYN-ACK segment
- SYN=1, ACK=1
- server host allocates buffers. ack# is client seq# + 1
- specifies server initial seq# (Y) (X+1)

Step3: client receives SYN-ACK, replies with ACK segment, which may contain data
- SYN=0, ACK=1
- ack# is server seq# + 1 (Y+1)
  - ↳ tells other party that u received the packet that you sent earlier.

* Once handshake is successful, you start exchanging data.
* scan a n/w to know services running on n/w, exploit this
A crucial flag to know whether port is open or not is the RST bit

Reset packet

- If machine receives a TCP packet it is not expecting (down that it responds with a TCP packet with RST bit set
- for eg → when no process is listening on destination port (port was closed) ↳ port where closed & you sent a SYN request, it will send back a RST.
- For UDP machine returns ICMP "port unreachable" instead. (to establish handshake) potentially closed

* If you receive a valid response that means port is active & open (Host is running that service)

(exploiting TCP hand-shake mechanism / UDP unreachable to know whether given port (doing a service) is open or not)

* ports that are alive → ping sweep
  Port scanning → to discern

Lecture 5: The NMap Tool

→ used for n/w scanning.

- Extremely popular
- * usually run over linux, * rich feature set, exploiting raw sockets
- * need root to use all features.

(II)
- Port Scanning → services running on ports
- * over any range of ports
- * Almost any type of TCP, UDP Packet.

(I)
- Ping Sweeping
- * over any range of IP-addresses.
- * with ICMP, SYN, ACK
- * as determination (running on machines that you are trying to attack at a point of time)

(III)
- Source IP address spoofing
- * Decoy scanning → insert fake IP addresses from which scans will be used to confuse n/w admin of victim n/w.

IMP → reference → Nmap man page

login to victim machine & do map nmap (use it)

Usage:

Input: (e.g. verbose mode) (trying to scan)

* nmap [Scan Type] [options] <target hosts>
* Default for port scanning: ports 1-1024 plus ports listed in nmap service file.

Output: (depends on what you are scanning)

PORT SCANNING TYPE

* open ports: syn/ack returned; port is open
* unfiltered (closed) ports: RST returned: port is closed but not blocked by firewall.

* filtered ports: nothing returned; port is blocked by firewall.

Examples

(I) Nmap : ping sweep

(38.1 to 38.255) all addresses

Nmap -sP -v 116.27.38/24

* Sends ICMP echo request (ping) to 256 addresses [If ICMP blocked because of firewall]
* Can change options to that pings with SYN, ACKS ... Also performs TCP (SYN, ACK scans)
* -sP = ping sweep
* -v = verbose

(II) Nmap : polite port scan

verbose mode

nmap -sT -v target.com → given host (scanning host on most standard TCP port eg web service)

* Attempts to complete 3 way handshake with each target port.
* Sends SYN, waits for SYN ACK, sends ACK, then sends FIN to close connection. (host is running the service, port is open)
* If target port is closed, no SYN ACK returned
* Instead RST packet is typically returned

(polite :) → DISADVANTAGE

* TCP connects scans are easy to detect
* Target (eg web server) may log completed connections.
* Gives away attackers IP address.
* If dont receive anything, packet is filtered by the firewall

(III)

Nmap : TCP SYN port scan

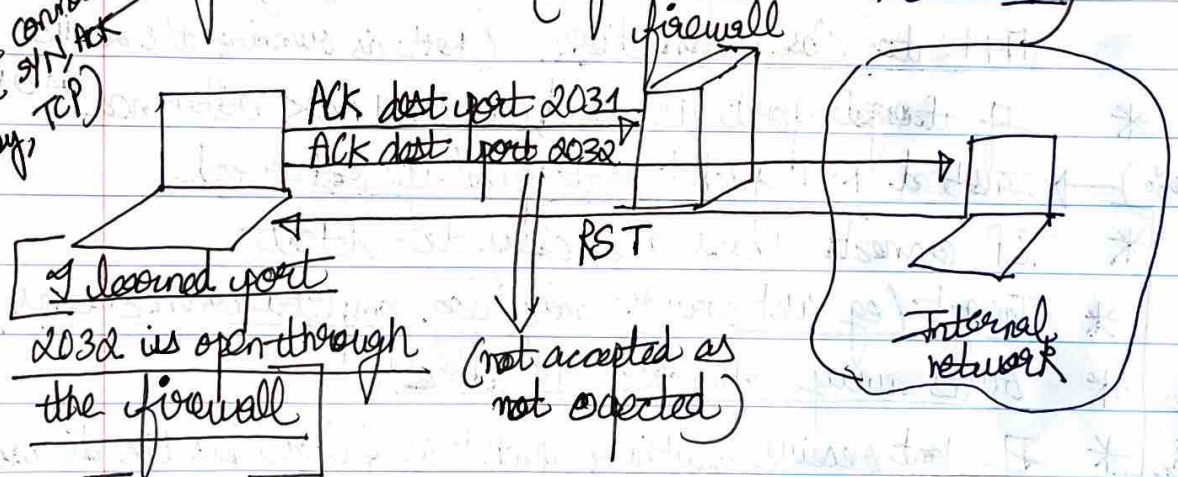
* nmap -sS -v target.com
* stealthier than polite scan.
* sends SYN, receive SYNACK, send RST (don't do full connection)
* stealthier : hosts do not record connection → (as connection not fully established)
* But router with logging enabled will record the SYN packet.
* Faster : don't need to send FIN packet.

(IV)

Nmap : TCP ACK scans (port filtered or not)

* nmap -PA -v target (particular if firewall does not accept incoming connection)
* Many filters (in firewall and routers) only let internal system hosts initiate TCP connections
* Drop packets for which ACK=0 (i.e SYN packet): no session initiated externally.
* To learn what ports are open through firewall, try an ACK scan (segments with ACK=1)

[firewall will block SYN packet] [but accept ACK packet as for internal connection (they will send SYN, ACK only, TCP)]



* Not learn whether port is open or not as you are receiving RST packet but learn whether port was filtered or not filtered.

[If TCP SYN incoming packets are getting blocked]

(V) Nmap : UDP port scans

* UDP does not have SYN, ACK, RST packets.
* nmap simply sends UDP packet to target port
  eg → nmap -PU target ; (may require root access)

→ ICMP port unreachable : interpret port closed.

→ Nothing comes back : interprets port open (maybe open/ maybe firewall blocking it)

• False positives common.

(VI) Nmap : obscure source

* Attacker can enter list of decoy source IP addresses into Nmap.
* for each packet it sends, Nmap also sends packets from decoy source IP addresses.

→ for 4 decoy sources, send 5 packets.

* Attacker's actual address must appear in atleast one packet to get a result.
* If there are 30 decoys, victim n/w will have to investigate 31 different sources. (n/w admin task running system task becomes difficult)

  Eg: nmap -n -D IP1, IP2, ...

  (no. of decoy IP addresses)

(VII) Nmap : TCP stack fingerprinting

* In addition to determining open ports, attacker wants to know OS on targeted machine :
  - → exploit machines known vulnerabilities
  - → sophisticated hacker may set up lab environment similar to target network.

* TCP implementations in different OSes respond differently to illegal combinations of TCP flag bits

  eg: nmap -O target

* O.S/ version of O.S of machine

→ Nmap sends : Fingerprinting

- 0 SYN to open port
- 0 NULL to open port (no flag bits sent)
- 0 SYN/ FIN/ URG (urgent)/ PSH to open-port
- 0 SYN to closed port
- 0 ACK to closed port
- 0 FIN/ PSH/ URG to closed port
- 0 UDP to closed port

* Nmap includes a database of OS fingerprints for hundreds of platforms

→ Nmap has signatures of how diff. OS respond to this combination & determine OS/ version running.

Nmap : more eqs

⇒ -sX → certain pattern of flags to do scan.

practical eqs (vulcan machines)

→ for os-scans / fingerprinting → need root access.

1) $ ifconfig → to configure IP-address of host
inet addr: 138.26.64.232

2) $ nmap -sP 138.26.64/24 (scan host/whole n/w)
↳ to see host on this n/w that are actually alive & running.

3) $ nmap -sP 138.26.64.26
↳ Host is up

4) $ nmap -sT 138.26.64.232 [cannot do stealthier scan in user mode]

→ what services this host was actually running.

↳

PORT	STATE	SERVICE
22/tcp	open	ssh
80/tcp	open	http
111/tcp	open	rpcbind

→ not recommended to do scan on all ports in a given n/w → that way you can be detected.

---

* nmap -sP 10.10.40.0/24

10.10.40.1 — 255

0 — 127 — 128

256/2 = 128

128
15
143

Notes and Warnings when using nmap (Diagnostic tool, also used for attack)

- GUI version available : zenmap :
  - http://nmap.org/zenmap/
- USE carefully
- Do not scan entire network (or all ports on given host) (not allow scans that require root access)
- Scanning a host for testing/learning purpose is fine.
- Please keep in mind the ethics of security education
- Lab will be the safest platform to try.

Defenses against network scanning (Diff. post)

- Filter using firewalls and packet-filtering capabilities of routers (block getting from services running on ports)
  - Block incoming ICMP packets, except to the hosts that you want to be pingable.
  - Filter Time Exceeded ICMP msg leaving your n/w.
- close all unwanted ports.
- scan your own system to verify that unneeded ports are closed. (penetration testing)
- Intrusion Detection System
  - eg → Snort (popular)
    - ↳ builds signature of benign scanning (admin) v/s malicious scanning (attacker)
  - → as part of ML algos learn