W1

15/09/2022

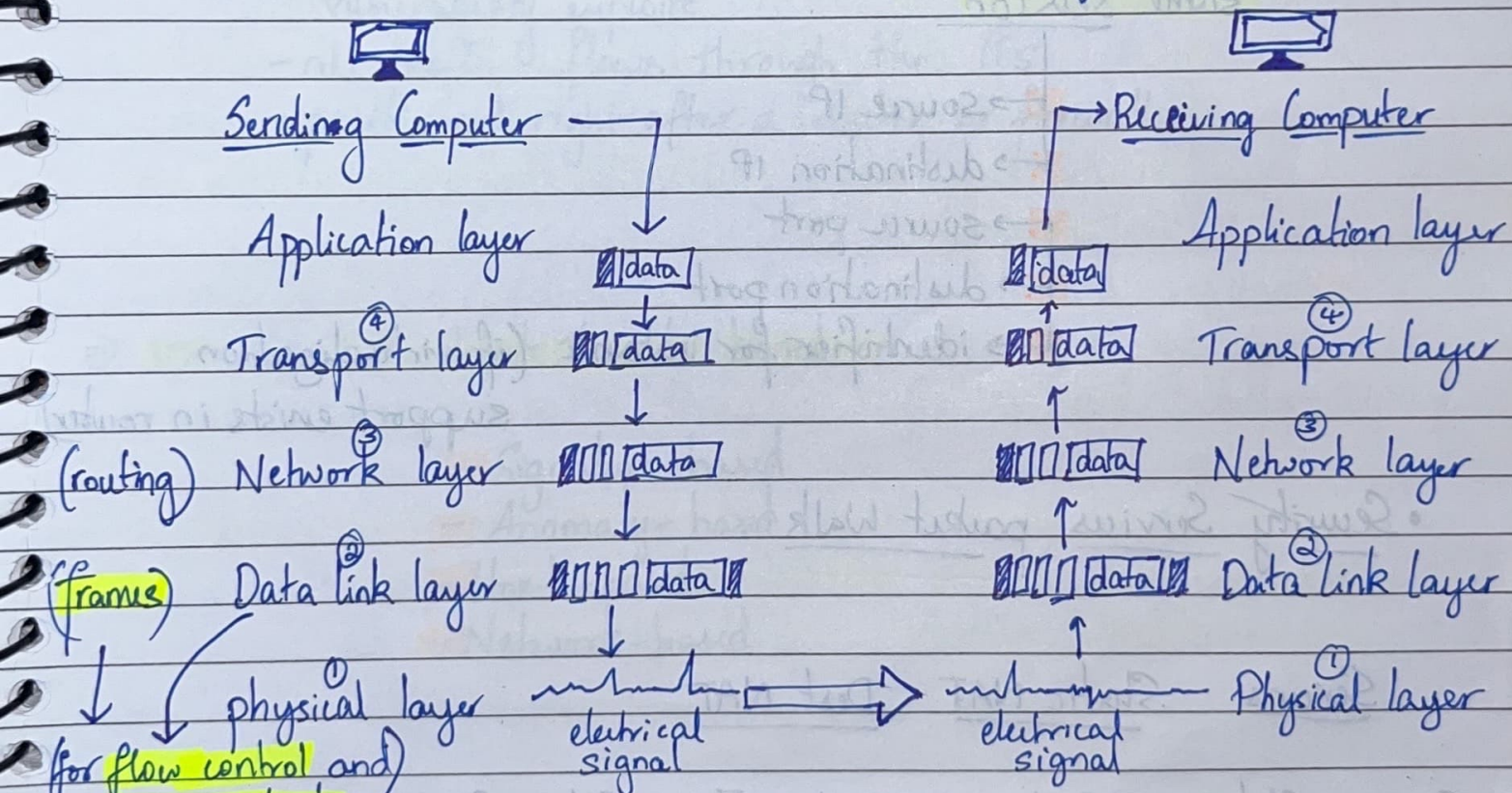
Module-4 Network Security and Database Vulnerability

Week-1

⇒ Introduction to the TCP/IP protocol framework

↳ **Stateless inspection**

- Stateless inspection has no concept of session table



- if our server is listening to that specific service for traffic from outside or even within our company's network, then the packet will be forwarded to server
- checks source and destination IP addresses and ports.

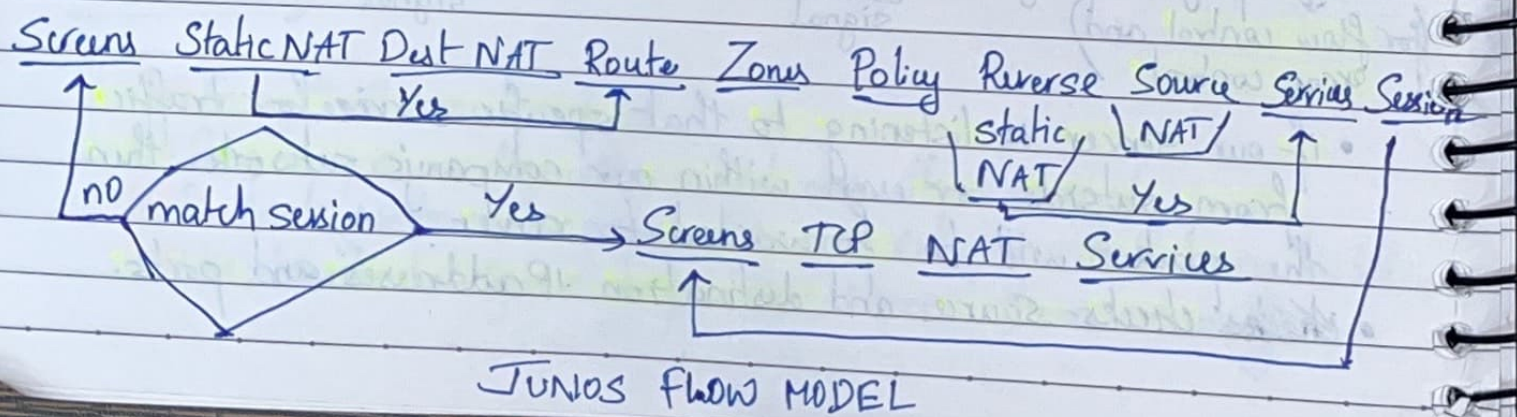
- Some use cases
  - To control traffic going in or out of organization
  - to control traffic routing
  - To perform Dos/Cos (prioritize traffic)
  - To troubleshoot purposes

↳ Stateful Inspection

- each packet is inspected with knowledge of all other packets that have sent or recived from same session
  - → source IP
  - → destination IP
  - → source port
  - → destination port
  - → identifier for instance (if virtualization support exists in router)

- Security Services packet Walk

Screens Static NAT Dest NAT



JUNOS FLOW MODEL

Intrusion detection System

- built for detecting vulnerability exploits against a target application or computer.
- listen only device
- monitors traffic and reports result to admin

Intrusion prevention system

- examins network traffic flow to detect and prevent vulnerability exploits.
- all traffic flows through the IPS
- positioned right after a router or firewall

⇒ IDS and IPS detect Threats based on

- Signature-based
- Anomaly-based
- Host-based
- Network-based.

⇒ IPS vs IDS

	IPS	IDS
• plaument in network infrastructure	inline (part of direct line of communication)	offline
• System type	monitors as well as defend ← Active/passive	passive

- detection mechanism
  - 1) statistical anomaly-based detection
  - 2) Signature detection:
    - exploit facing signature
    - vulnerability facing signature
  - 1) Anomaly based

↳ Network address translation (NAT)

- masquerades real IP address (IP configured on system) with a different IP address for use on network
- ex: IP addresses from internal network are translated from private IP address that is on the computer to an external or public IP that is routable on the internet.
- remaps one IP space into another by modifying network address information in Internet protocol datagram packet headers while in transit across a traffic routing device.
- NAT allows organizations with non-globally routable addresses to connect to internet by translating those addresses into globally routable address space.

↳ Types of NAT

- static: allows one-to-one mapping between local and global addresses
- Dynamic NAT: maps unregistered IPs to registered IPs from a pool of registered IPs

M T W T F S S

Page No.:	YOUVA
Date:	

- Overloading: maps multiple unregistered IPs to a single registered IP using different ports.
  Also known as Port Address translation.

M T W T F S S

Page No.: YOUVA

Date:

⇒ Network protocols over Ethernet and Local area networks

↳ Network Addressing

	Data link → Layer 2 Address	Network → Layer 3 Address
A.K.A	MAC Addresses Hardware Addresses Physical Addresses	IP Addresses logical addresses
ex:	00:90:69:9f:ea:46 octets of 6 ~~bits~~; 8 bits each total of 48 bits	172.16.12.1
Description	• Identify stops made along the way • Change with each stop along the route	• Identify communicating computers or end points • Doesn't change

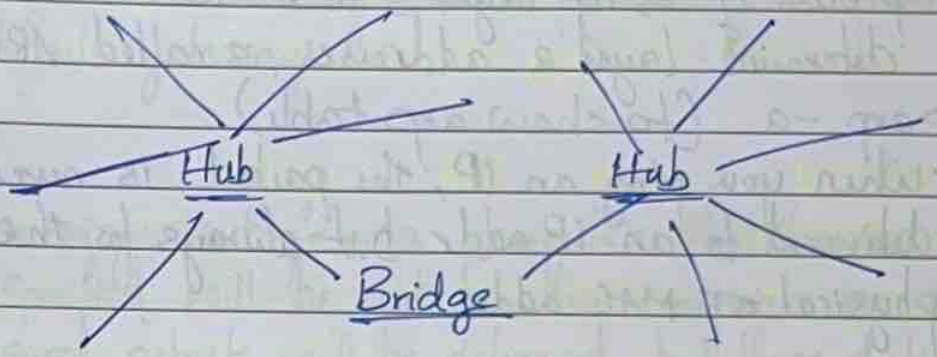
↳ The frame or header of layer 2, the data link layer contains the source and destination MAC IP addresses, the protocol type, data and checksum.

↳ to ensure integrity

- All modern network support full-duplex communications.

↳ ability to send and recieve data at the same time.

↳ Bridge ÷ similar to a hub, but it doesn't send the signal to all connected ports, but only to the port the destination computer is attached to.



- forwards frames
- learning mac frames
- controlling traffic

↳ Switch ÷ modern version of a bridge
- full duplex & data transmission
- each port is dedicated to single device; bandwidth isn't shared
- Virtual lans possible

M T W T F S S

Page No.: YOUVA

Date:

⇒ Basics of Routing and Switching, Network packets and Structures

↳ Address Resolution protocol (ARP)

- process of using layer 3 addresses to determine layer 2 addresses is called ARP.
  > arp -a (to show arp table)

!! - When you ping an IP, the packet is never delivered to an IP addr but always to the physical or MAC addr

- To summarize ARP translate IP addresses into MAC addresses in order for systems to communicate.
  - only required when data is being transmitted within same broadcast domain
  - when data is being transmitted to another broadcast domain, only the MAC of the router or gateway is required.

↳ To summarize, to send packets accross different network segments, our default gateway will make sure that the packets are delivered to the closest layer 3 device until it finds a layer 3 device that has destination system directly connected to one of its interfaces.

↳ Routing tables are maintained by any network connected devices.

↳ When a NIC reads a packet header and sees the destination address is not its own address, it discards the packet.

↳ When a router needs to send a packet to an address that isn't in its routing table, it forwards the packet to the default gateway.

↳ messages sent from a computer that has no gateway specified will be sent to other computers on the same subnet will be delivered but those destined to computers on other networks will not be delivered.

↳ Three types of routes found in a routing table

- Direct
- Default
- Dynamic

M T W T F S S

Page No.: YOUVA

Date:

17/09/22

Week-2 IP Addressing - The Basics of Binary

↳ Convert from decimal to binary

Q) 235

235 - 128 ? ✓ = 107	1
107 - 64 ? ✓ = 43	1
43 - 32 ? ✓ = 11	1
11 - 16 ? X = X	0
11 - 8 ? ✓ = 3	1
3 - 4 ? X = X	0
3 - 2 ? ✓ = 1	1
1 - 1 ? ✓ = 0	1

$(235)_{10} = (11101011)_2$

↳ IP protocol

IPv4 → 4 octects of 8 bits each from 0.0.0.0 to 255.255.255.255 (32 bits)

→ 10.195.121.10 → host portion

Network portion

→ nowadays computers are set up to allow DHCP or Dynamic

Host Configuration protocol to dynamically configure IP addresses for you.

→ Here if we take 192.168.52.3/24, This whole number is called CIDR range.

→ The /24 defines how many bits of the IP are dedicated to the network portion of the address.

→ Classful Addressing

subnet mask ⇓

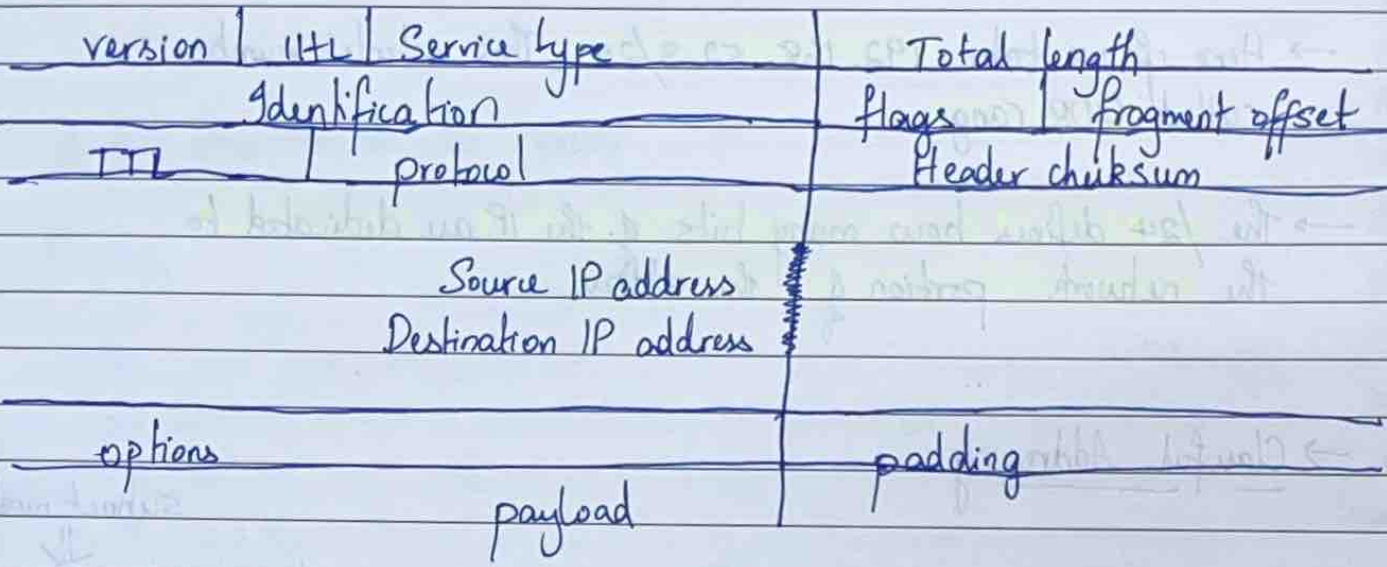
- class A → 0.0.0.0 to 127.255.255.255 → Unicast/ → 255.0.0.0
- class B → 128.0.0.0 to 191.255.255.255 → spcial use → 255.255.0.0
- class C → 192.0.0.0 to 223.0.0.0 → Unicast/special use → 255.255.255.0
- class D → 224.0.0.0 to 239.255.255.255 → multicast → N/A
- class E → 240.0.0.0 to 255.255.255.255 → Reserved → N/A

A 192 . 168 . 102 . 32
B 192 . 168 . 102 . 32
C 192.168.102 . 32

network portion — host portion

→ A routable protocol is a protocol whose packets may leave your network, pass through your router, and be delivered to a remote network.

→ IP protocol header



→ Broadcast address

- special address is reserved to be the subnet broadcast address
- The subnet broadcast address is formed by setting the network/subnet portion of an IPv4 to appropriate value and all bits in host portion to 1
- IP → 192.168.102 . 32 → 00100000
- Broadcast → 192.168.102 . 255 → 11111111

network portion host portion

→ IPv6

- 128 bits in length
- 8 hexadecimal digits of 16 bits each
- 65b3 : b834 : 45a3 : 0000 : 0000 : 762e : 0270 : 5224
- can use :: instead of consecutive zeros.
- can use :: only once in address

↳ Addressing schemes (IPv4)

- **Unicast**: One computer communicates with just one other system

- **Broadcast**: One computer communicates with all other systems on the same subnet.
  : Network portion of broadcast address is same as network portion of all other computers on that subnet

- **multicast**: one to many arrangement.
  : Anything sent from that system will only be received by those systems that are subscribed to receive multicast from that system.

↳ Addressing schemes (IPv6)

- **Unicast**: one to one communication
- **multicast**: used to send data to multiple systems at one time
- **Anycast**: refers to group of systems providing service
  : packet sent to anycast address is routed to the nearest interface that has that address

M T W T F S S

Page No.: YOUVA

Date:

⇒ TCP/IP layer 4, Transport layer overview

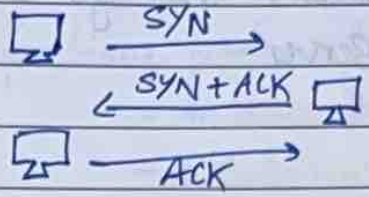
↳ TCP sets up a connection, so sending and receiving computers know which packets have been sent and received, and which is the correct order. [slower]

↳ UDP doesn't set up a connection the same way TCP does. The receiver can only reassemble the data stream in the order the packets arrive. [faster]

↳ Common protocols using UDP

- TFTP (trivial file transfer protocol) – port 69
- DNS (Domain name system) – port 53
- SNMP (Simple network mgt protocol) – port 161 and 162
- DHCP (Dynamic Host configuration protocol) – port 67
  - ↳ manages pool of IPs to systems subscribed to it
- VoIP (Voice over IP) – port 5060
  - ↳ used to send voice over internet
- IPTV (IP television) – port 80, 5004, 12000
  - ↳ streams TV signal

↳ Application and transport protocols



- HTTP — 80
- HTTPS — 443
- SMTP — uses TCP (25)
- FTP — 21/20

⇒ TCP/IP layer 5, Application layer

↳ DHCP handshake consists of four packets that go between requesting system and DHCP server—

- Discover
- Offer
- Request
- Acknowledgement

↳ The DHCP Server will be listening on port 67. Once it recieves DHCP request it will chuk its pool of IP addresses to see if one is available. If it has an IP to lease, it will reply back to the requesting endpoint with a DHCP offer containing the proposed IP addressed and related info like DHCP Server address, gateway address, subnet mask and lease duration.

↳ Syslog protocol

- standard protocol for messaging and Logging.
- Used ~~for~~ to forward all these messages to a centralized Syslog Server where system management and auditing can take place
- 3 layers:
  - Syslog Content : Information contained in message
  - Syslog Application : allows syslog message to be routed, analyzed and stored. Also handles

~~sending message across network~~

-Syslog Transport: handles sending the syslog message across the network

- functions performed at each conceptual layer:
  - An originator (Router, switch, server, etc)
  - A collector (Syslog server for further analysis)
  - A relay (Syslog forwarder)
  - A "transport sender" (passes syslog messages to specific transport protocol
    - common - UDP, defined in RFC 5426)
  - A "transport receiver" (takes syslog messages from specific transport protocol)

- Syslog message components includes the facility code and severity level.
- Syslog software adds information such as:
  - originator process ID
  - a timestamp
  - the hostname or IP address of device.
- facility names reflect the names of UNIX processes and daemons
- recognizes 23 facility codes.
- Syslog security levels go from 0 to 7, 0 being emergency and 7 being debug.

M T W T F S S

Page No.: YOUVA

Date:

↳ Netflows take sample of the flows traversing an interface

- In a flow one can see
  - Usage (packet/byte count)
  - Timestamp
  - Port Utilization
  - QoS (TCP flags, protocol)
  - To/from (Source or dustination IP)
  - Application (sourc or dust TCP/UDP port)
  - Routing and peering (TCP flags, protocol)

↳ port mirroring is when a switch is configured to make a copy of all the traffic traversing one or more ports on that switch, and send the copied packets out to a single destination port.

- AKA Switched port Analyzer (SPAN)
- most of the mirrored packets will be sent to IDS that is configured to monitor network all the time.

- So if an IDS or any network Analysis tool to work, it must have a NIC that's configured in promiscuous mode, so it can read all the frames that are sent to its incoming port.
- basically port mirroring is done so as to provide a stream of data entering or leaving a specific port for debugging or analysis work.

⇒ Firewalls, Intrusion detection and Intrusion prevention systems

↳ Next Generation firewall / Deep packet Inspection firewall

- main difference is the word sessions.
- able to inspect traffic that is encrypted with TLS/SSL, and even website filtering.
- Sessions
  - allow a firewall to permit returning traffic if its part of a previously established session
- NGFW provides intelligence to distinguish business applications and non-business applications and attacks.

↳ Traditional firewall uses layer 3 or 4 to perform blocking decisions whereas NGFW inspect from layer 2 to 7.
- Can make decisions based on application layer too

[traffic entering network boundary]

↳ ⇒ flow of traffic between ingress and egress interfaces on next Gen FW

- At the interface level, we can configure traditional firewall rules that basically states layer 3 and 4 of osi model.
- Then it flows through flow module which has next gen securing capabilities
- Then it sends it out of the egress interface.

[traffic exiting network boundary]

M T W T F S S

Page No.: YOUVA

Date:

↳ A. NGFW is application aware.
- provides higher level granularity.
- ex: palo alto networks
  Cisco
  Juniper
  McAfee

- ex of open Source NGFW
  - pfSense
  - clear OS } Linux firewall
  - IPCop

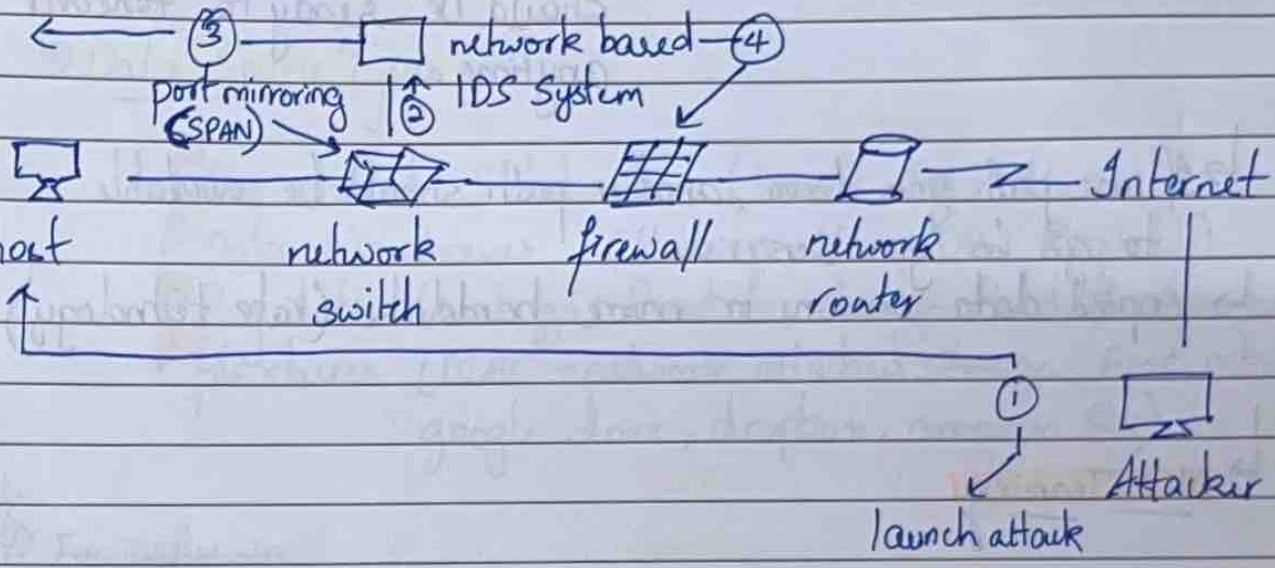
↳ IDS

- Signature based : analyses content of each packet at layer 7 with set of predefined signatures

- Anomaly based : monitors network traffic and compares it against an established baseline for normal use.

- Types
  - Host based IDS (anti threat apps)
  - Network based IDS (anti threat software)

↳ IDS location on network



⇒ Clustering and high Availability Systems

- high availability system is basically a component that is continuously operational 100% of time.
- The requirements are
  - Hosts in a virtual server cluster should have acess to same shared storage and must have identical configuration too.
  - Same OS level
  - DNS must be configured correctly too.
  - you should have a connection between primary and secondary node.

↳ To create HA systems, we need 3 characteristics

- Redundancy: multiple components can perform same task. Allows second server to take over if the first one goes down.

- monitoring and failover : Secondary device should always monitor primary device and should be ready to takeover anytime.

↳ If one path goes down, another path should be available to go to your servers.
↳ copied data on one or more harddisks (rate technology)

↳ NIC Teaming

- protection against NIC failures
- fault tolerance in the event of network adapter failure.
- If one Network interface card fails, the other should hop in.

↳ Notable quiz answers

- Time to live or TTL is the number of layer 3 devices (hubs, routers, etc) the ~~packet~~ packet is allowed to pass through before it is dropped)

M T W T F S S
Page No.:
Date:
YOUVA

18/9/2022

Week-3 Network Security database Vulnerabilities

Data Source Types :-

① Distributed Databases (Oracle, DB2, MySQL)
② Data Warehouses (Hadoop, MongoDB, BigTable)
③ Big data (Netezza, Enadata, Amazon Redshift, Apache Hive)
④ file shares (NAS - network attached Storage, EMC, net App, google drive, dropbox, amazon S3)

~~Typically in~~

↳ ① structured data
③ Semi-structured data
② Structured data
④ Unstructured data

↳ ① structured data is data that has been organized into a formatted repository, typically a database so that its elements can be addressable for effective processing and analysis.

③ hasn't been organized into specialized repository - database, but has associated information such as metadata, that makes it amenable to processing than raw data

④ Information in many different forms, and typically isn't good fit for mainstream relational databases.

Securing your data

Discover > Harden > Monitor & Protect > Repeat

Identification & Baseline

- Discovery and classification → (where) (sensitive data)
- entitlements reporting → (who) (accss)
- Vulnerability Assessment → (how) (secure repository)

Raise Bar

- Reconfigure, Mask & Encrypt → (how) (sensitive data)

Real-Time monitor and protection

- activity monitoring → (what)
- Blocking quarantine → (how) (prevent unauth accss)
- Dynamic data masking → (how) (sensitive data protection)

↳ A database administrator will login through a db client to make big changes to a database.

↳ for added Security, a firewall is often placed between the database and the hardened data repository.

Data Activity monitoring / auditing / logging

1) Does your product log all key activity, including key generation, retrieval/ usage, etc?
2) Demo data access activity monitoring and logging of activity monitoring?
3) Does your product monitor for unique user identities with access to the data?
4) At the storage level, it can detect/identify access to highly ~~developed~~ privileged users such as database admins, system admins or developers?
5) Does your product generate real time alerts of policy violations while recording activities?
6) Does your product monitor user data access activity in real time with customizable security alerts and blocking unacceptable user behaviour, access patterns or geographic access, etc?
7) Does your product generate alerts?
8) Demo the capability for reporting and metrics using information logged?
9) Does your product create audible reports of data access and security events with customizable details that can address defined regulations or standard audit process requirements?
10) Does your product support the ability to log security events to a centralized incident and ~~information~~ ~~man~~ event management system? (security)
11) Demo monitoring of non-relational DBMS systems such as Cognos, Hadoop, spark, etc
12) Demo the following events attributes and to what level of granularity?
13) Demo when the product provides the following event attributes and to what level of granularity?
14) Demo sufficient information in the log record to establish what events occured and who or what caused them?
15) Demo configurations configured to monitor user account additions and changes
16) Demo configurations to monitor significant instances of failed passwords

attempts and against multiple accounts within a short time frame which may indicate hacking attempts

17) Demo configs to monitor significant instances of failed accss attempt to the database not authorized to account ID.
18) Demo configs to monitor attempts to select the list of users and passwords.
19) Demo configs to monitor all direct accss to databases from accounts which should be limited to accss through the application
20) Demo configs to monitor use of nonstandard tools to directly accss DBMS (excel/Accss)
21) Demo configs to monitor use of application ID from a source other than the defined owner application location (based on host name or IP address)
22) Demo the configs to monitor log failures, manual logging shut down and attempts to purge

⇒ Week 3 notable quiz answers and Sdutions

- flat file databases store all the data in a single table.
- An access rule is set up to automatically terminate a session if an attempt is made to accss data in a sensitive table, social security ID number (SSN).
- By configuring bidirectional communication between monitoring and SIEM systems, if available and by configuring monitoring system to write to the SIEM system syslog file, security events ~~can be~~ collected by a data monitoring tool can be logged to a security incident and event management system.

M T W T F S S
Page No.: YOUVA
Date:

19/09/2022

Week-4 Deep-Dive Injection Vulnerability

↳ Injection flaw allow attackers to relay malicious code through the vulnerable application to another system. (OS, Database Server, LDAP Server (lightweight directory access protocol), etc)

↳ Injection vulnerability is OWASP #1

↳ OS Command Injection

- abuse of vulnerable application functionality that causes execution of attacker-specified OS Commands
- applies to windows, Linux, MacOS

- OS Command Injection can lead to:
  - full system takeover
  - Denial of Service
  - Stolen sensitive Information (passwords, crypto keys, sensitive info, confidential data)
  - lateral movement on the network, launching pad for attacks on other systems.
  - Use of system for botnets or cryptomining.

↳ Prevention

(1)

→ Use library functions to significantly reduces the attack surface

- Instead of rm, use java.nio.file.Files.deleteIfExists(file)
- Instead of cp, use java.nio.file.Files.copy(source, destination)

②

→ Run at the least possible privilege level.
- If attacker sneaks an OS command (rm -rf /) as root, it is game over, but can cause less damage if is run as user.

③ → Do not run commands through shell interpreters
- this command allows additional rm:
> /bin/sh -c "/bin/rm /var/app/logs/x; rm -rf/"

- But this won't
> /bin/rm /var/app/logs/x; rm -rf/ (fails)

④

→ Use Explicit paths when running executables.
- If a writable folder is referenced in the path before the folder containing the valid executable, an attacker may install a malicious version of the application there.
- execution of malicious application:
> nmap 123.45.67.89

- can be avoided by referencing executable by full path:
> /usr/bin/nmap 123.45.67.89

- explicit references help avoid DLL hijacking

Dynamic link library - shared library in windows

(5)

→ Use safer functions when running system commands
- use functionality that helps prevent command injection
- following function call is vulnerable to new parameter injection.

```
> Runtime.getRuntime().exec("/usr/bin/nmap " + ipAddress);
```

↳ new parameter

- but this isn't vulnerable

```
> Runtime.getRuntime().exec(new String[] {"/usr/bin/nmap", ipAddress});
```

(6)

→ If possible, do not let user input such command execution unchanged
- modifying user input, or replacing user-specified values with others (translation tables) help protect against injection.
- Instead of allowing user to specify file to delete, specify an unique file ID.

```
> action = delete & file = 457
```

- When submitted:

```
> realName = getRealFileName(fileID);
  Runtime.getRuntime().exec(new String[]{"/bin/rm",
  "/var/app/logs/" + realName});
```

→ Sanitize user input with strict whitelists (not blacklists!)
- hard to build blacklists
- whitelist filename as [A-Za-z0-9._]+
- Blacklist ; and |
  > x`rm -rf /` (evasion)
- Blacklist ; and |`
  > x$(rm -rf /) (evasion)
- Blacklist spaces
  > x;rm${IFS:0:1}-rf${IFS:0:1}/ (evasion)

↳ SQL Injection

- suppose we have a login dialog
- Backend code would be:
  > stmt.executeQuery("SELECT * from users where user='"+user"' and pass='"+pass"'")
- with regular input:
  > select * from users where user='bob' and pass='secret'
- malicious input
  > select * from users where user='' OR 1=1;-- ' AND pass=''
  - Here first single quote closes value for username
  - then adds a boolean cause 1=1 → always true
  - ; seperates SQL queries
  - -- is a comment
  - so ~~a quote~~ ' and pass='' is commented out.

- Consequences
  - Authentication bypass
  - Data enfiltration
  - copy of OS commands
    > copy (Selut 1) to program 'rm -rf /'
  - DOS
    > drop table sales
    > selut * from users where user = '' ; drop table sales ; -- ' and pass = ''

- Types
  - error based (application displays database errors)
  - Union based
    > selut name, text from log where date = '2018-04-01' UNION selut user, password from users -- '
  - Blind Injection (based on behaviour of system)
    - Boolean based or time based
    - one of two possible outcomes
    - immediate vs delayed execution
      > IF (password like 'a%', sleep(10), 'false')
      ↓
      indicates if first letter of password is 'a', sleeps for 10 seconds, ie delays
  - Out of band
    - through a seperate channel
    - by sending an HTTP request

b) prevention

①
→ Use prepared statements
- seperates query structure from query parameter
- use:
> PreparedStatement ps = conn.prepareStatement ("selut * from users where user = ? and pass = ?");
ps.setString(1, user);
ps.setString(2, pass);
} doesn't allow where clause or union query

②
→ Sanitize user input
- do not allow user reach database, instead use mapping table to translate.

③
→ Do not expose native database errors to user.
- belong to internal log file.
- application errors shouldn't expose internal information to user.

④
→ limit database user permissions.
- use user with read-only perms

⑤
→ use stored proudures
- mitigates risk by moving SQL queries into database engine.

⑥

→ use ORM libraries (Object Relational mapping)
- example: Java Persistence API (JPA) implementations like hibernate
- create in-memory set of objects that map to your database structure, so its hard for attacker to modify with malicious syntax.

↳ NoSQL Injection

- In MongoDB $where is interpreted as javascript
- suppose
  > $where: "$expression"
- harmless
  > $where: "this.userType == 3"
- However, DoS attack:
  > $where: "d = new Date; do {c = new Date;} while (c - d < 100000;"

↳ XPath Injection

- operate on XML, XML trees
- malicious
  > //Employee[UserName/text() = '' or 1=1 or '1'='1' And Password/text() = '']

↳ LDAP Injection

- common mechanism for managing user identity information.
- command to find username & pass
  > find ("(&(cn=" + user + ")(password=" + pass + "))")
- malicious users may tweak username to force expression to find any user
  > find ("(&(cn=*)(cn=*))(|cn=*)(password=any))")

↳ Injections flaws also exist in Templating engines

⇒ Notable websites → OWASP cheat Sheets

- Injection flaws : https://owasp.org/www-community/Injection-flaws
- OS Command Injection : https://owasp.org/www-community/attacks/Command_Injection
- SQL Injection : https://owasp.org/www-community/attacks/SQL_Injection
- LDAP Injection : ~~https://owasp.org/~~ https://cheatsheetseries.owasp.org/cheatsheets/LDAP_Injection_prevention_Cheat_Sheet.html

→ The Database Hacker's Handbook : Defending Database Servers (book)

→ pentestmonkey

- MSSQL Injection cheat sheet: http://pentestmonkey.net/cheat-sheet/sql-injection/mssql-sql-injection-cheat-sheet
- Oracle Injection cheat sheet: /sql-injection/oracle-sql-injection-cheat-sheet
- DB2 Injection cheat sheet: /sql-injection/db2-sql-injection-cheat-sheet
- Postgres Injection cheat sheet: /sql-injection/postgres-sql-injection-cheat-sheet
- MySQL Injection cheat sheet: /sql-injection/mysql-sql-injection-cheat-sheet