# Application Security

# Vulnerability Assessments

# *. Cyber Threat Intelligence : (CHAPTER-1)

- Threat is a potential occurence of an unwanted event which damages the functionality of an organization.
- A threat can affect the integrity and availability factors of an organization.
- The impact of threats is high.
- The threat intelligence, usually known as TI, is defined as the collection and analysis of information about threats and adversaries and drawing patterns that provide an ability to make knowledgeable decisions for the preparedness, prevention and response actions against various cyber attacks.
- Threat intelligence can be effectively leveraged to enhance the following areas of cyber security
  1. Identify and protect Detect
  2. Respond
  3. Recover

Types :

1. Strategic Threat Intelligence — cyber attacks
2. Tactical Threat Intelligence — Collect about attacks
3. operational Threat Intelligence — methodologies to prevent
4. Technical Threat Intelligence — attack vectors

Ex: A malware used to perform an attack is tactical threat intelligence, whereas the details related to the specific implementation of malware come under technical threat intelligence.

*. Threat intelligence - informed risk management provides organizations with the following:

① Information assurance

② Detailed information about security vulnerability

③ Determining possible threat actors to organization.

④ Identification of new vulnerabilities.

*. Threat Intelligence is used to understand the direction and operations needed to perform risk Management. → Frame, Assess → Respond, Monitor

*. INCIDENT RESPONSE MANAGEMENT:

• Phases involved of escalation in the incident RM.

1. Pre-planning → Incident Response (IR), Breach Response (BR)

2. Event operational & Tactical threat intelligence

3. Incident once an adversary sets a foothold.

- standard operating procedures (SOPs) play an important role in improving incident response.

## *. RISK ASSESSMENT : (CHAPTER-3)

- Risk = Threat * Vulnerability

| Risk | Threats | Vulnerabilities |
|---|---|---|
| → Business Disruption | Disgruntled employees | Zero-day Vuln. |
| → Different Losses | Criminals, Terrorists | Improper Sec. Contol |
| → Legal Liabilities | Competitors, Hackers | Software/Hardware flaws |
| → Physical destruction | Natural Disasters | Inadequate Bandwidth |

- Prioritizing Risks : Risk = Threat x Vulnerability x Impact
- Organization typically employ a 3-step process to determine the overall likelihood of threat events :

1) Assess if threat events will be initiated or will occur.
2) Assess the impact or harm it can cause to organization.
3) Assess the combination and compound impact of all likelihoods.

- Goals of Risk Assessment :

1) Identifying probable threats and dangers.
2) Assessing the impact caused by a threat.
3) Analyzing the uncertainity of the nature of threat.
4) Taking necessary measures to reduce the impact of the threat.
5) Analyzing the solutions that can be used.

- Risk Assessment Process :

1) Identify Risk.
2) Develop Assessment Criteria.
3) Assess Risks.
4) Assess Risk Interactions.
5) Prioritize Risks.
6) Respond to risks.

- Management Security :

1) Assigning of responsibilities.
2) Providing support.
3) Incident Response aptitude.
4) Review of Security Control
5) Security & Tech training.
6) Risk assessment.
7) System Security plan.

- Technical Security :

1) Communications
2) Cryptography
3) Discretionary audit control
4) Identification & Authentication
5) Intrusion detection
6) System Audit

- Likelihood is divided into 3 categories:

1) High : The threat source is motivated and proficient and the security controls in place are not enough to prevent the vulnerability from being exercised.

2) Medium : The threat source is motivated & proficient but the security controls are in place to prevent the vulnerability from being successfully exercised.

3) Low : The threat sources lacks motivation & proficient and the security controls are in place prevent the vulnerability from being exercised atall.

- IOT Vulnerability Management :

1) Improper Authentication
2) Improper Physical Security
3) Absense of Keys

- Areas where we find IOT Vulnerabilities:

1) Device memory
2) Ecosystem & Access Control
3) Decommisioning system
4) Device Physical Interface
5) Device web Interface
6) Device Firmware
7) Device Network services
8) Admin Interface
9) Third-party API's
10) Local data storage
11) Ecosystem communication.

- Phases of Risk Management :

1) Risk Assessment.
2) Risk Mitigation.
3) Risk Management Plan Evaluation.

- Steps involved in Risk Assessment :

1. Identify all relevant resources and infrastructure boundaries. (system characterization).
2. Threat identification.
3. Vulnerability identification.
4. Control Analysis
5. Likelihood Analysis
6. Impact Analysis

---

## WEB APPLICATION SECURITY (CHAPTER 6)

- Web applications provide an interface between end users and web servers through a set of web pages that are generated at the server end or contain script code to be executed dynamically within the client web browser.

- Components of Web Application:

1) Login
2) Web Server
3) Session Tracking Mechanism
4) User Permissions
5) Application content
6) Data Access
7) Data store

8) Application Logic 9) Logout

- Web Application Architecture:

1) client or Presentation Layer
2) Business logic layer
3) Database Layer

- Website Defacement is a process of changing the content of a website or web page by hackers. Hackers break into the web server and will alter the hosted website by creating something new.

- Reasons for Web Servers' compromise:

1) Webmasters' concern
2) Bugs in software programs
3) Illegal Authorization
4) Network Admin Concern
5) End User concern
6) User Account Compromise

- Impact of Webserver Attacks:

1) Data Tampering
2) Secondary attacks from websites
3) Root access to other application or Server
4) Whitelisting / Blacklisting: (Tools)

- Apility.io
- Autoshun
- Cisco Umbrella
- APT Groups & operations
- I-Blocklist
- CINS Army List
- FireHOL IP Lists
- Rutgers Backlisted IPs

- Web Content Filtering Tools :

1) open DNS
2) inCompass
3) WebTitan
4) Smoothwall SWG
5) NetSentron
6) Symantec Secure Web Gateway

- Web Proxy Tools :

1) Proxy switcher
2) Proxy Workbench
5) Hotspot Shield
3) Cyber Ghost VPN
4) Tor Burp Suite
6) Proxifier

- Web Application Fuzz Testing Steps :

1) Identify the target system
2) Identify inputs
3) Generate fuzzed data
4) Execute the test using fuzz data
5) Monitor the system behavior
6) Log the defects

Tools

1) WS Fuzzer
2) Webscarab
3) Burp Suite
4) Appscan
5) Peach Fuzzer

- Web Application Security Testing Tools :

1) N - Stalker Web Application Security Scanner
2) OWASP Zap
3) Arachni
4) Vega
5) Nessus
6) Skipfish
7) WebReaver
8) WSSA - Web Site Security Audit
9) Syhunt Hybrid

- The Veracode report shows that the most common types of flaws are:

1) Information leakage (64%)
2) Cryptographic issues (62%)
3) CRLF injection (61%)
4) Code Quality (56%)
5) Insufficient input validation (48%)
6) Cross-site scripting (47%)
7) Directory traversal (46%)
8) Credentials management (45%)

- Application security is made of 4 factors:

1) Breach cost (Bc)
2) Vulnerability density (Vd) = $\frac{\text{Vulnerabilities}}{\text{Size of software}}$
3) Counter measure efficiency (Ce)
4) Compliance index (CI) = $\frac{\text{no. of compliance requirements}}{\text{total no. of compliance req}}$

App Sec Risk Model

$$ASRM = \frac{Vd \times Bc}{Ce \times CI}$$ *IMP

- The method of designing the ASRM includes 6 stag

1) classification of Applications
2) Quantification of Breach cost

3) Application Vulnerability Density

4) Countermeasure Efficiency

5) Compliance Index

6) ARSM Formulation

---

## WEB APP SECURITY & AUDIT TRAIL

- Security relies on the following elements :

1) Authentication
2) Authorization
3) Auditing
4) Confidentiality
5) Integrity
6) Availability

- Audit trail are the manual or electronic records that chronologically catalog events or procedures to provide support documentation and history that is used to authenticate security and operational actions, or mitigate challenges.

- Numerous industries use versions of an audit trail to provide a historical record of progression based on a sequence of events. These records provide proof of compliance and operational integrit

## * SECURITY PRINCIPLES :

- Vulnerability - weakness in the system or network that is dangerous because an attacker can exploit it

- Attack vector - the path / approach used by an attacker to hack or exploit into system, network
  e.g.: Malwar, virus, email attachments, web pages, pop-ups, social erginering

- Attack surface - The environment (software surface) where the attacker can perform an attack.
  - → physical Attack surface - related to hardware, devices (Router, switch, Mobile, printer) cc camera, USB ports
  - → Digital Attack surface - software, OS services, apps, ports, networks.
    - ↓ accessed from remote access location
    - → Network Attack surface - vulnerabilities over an networ with weak authorizati weak DOS, detection systems
    - → Software Attack surface - vulnerabilities in apps, software like bugs in code
    - → Human Attack surface - vulnerabilities created by outsiders onto employees like social erginering, trusted insides

- Social Erginering - pretending genuine details / change the identity & create trust with people to manuplate things inorder to gain access.
  (phishing attacks)

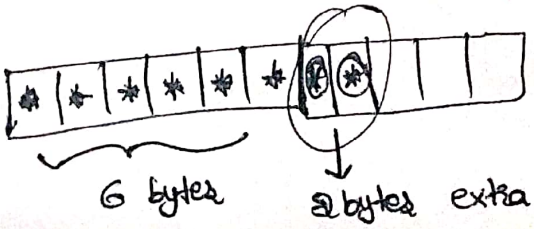
- Sphear phishing – attack a particular & perform phishing.
- Keylogging = keyboard Capturing / action of recording Keys by a spyware / malicious code.
- Adversary – enemy / attacker (in cyber world)
- Zero day – software security flaw that is known but doesn't have a patch in place

## *. SECURE CODING :

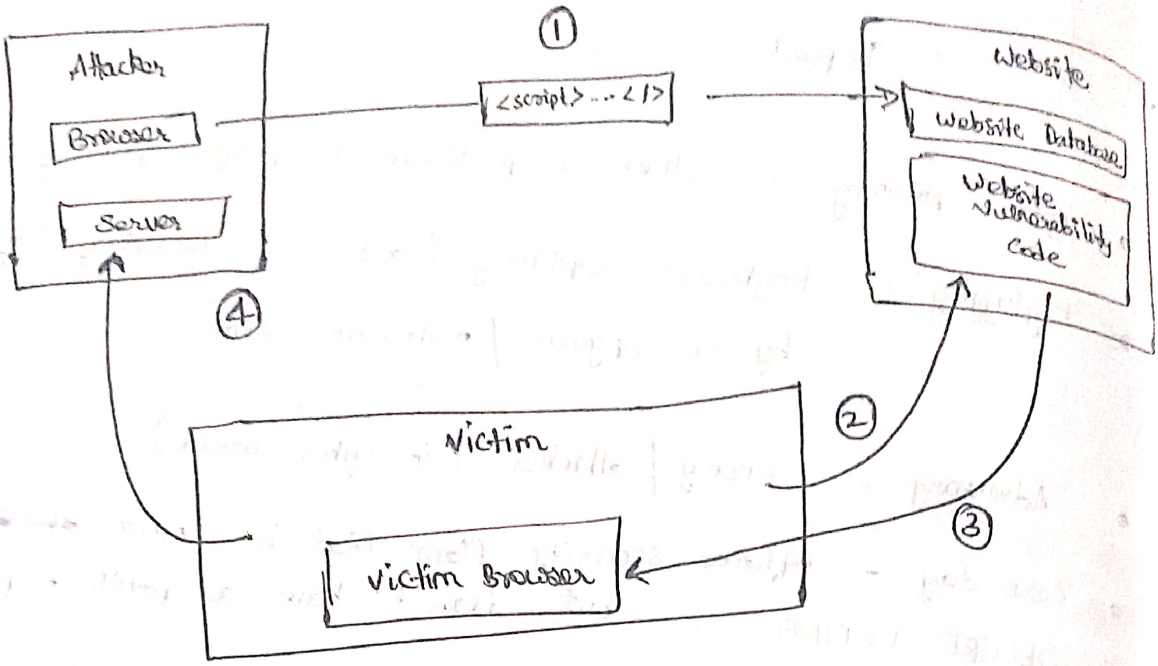
- The practice of developing computer software to avoid introduction to intruder or other security vulnerabilities.
- OWASP – open Web Application Security Project
- Risks involved :
  i. Denial of service to single user.
  ii. Compromised secrets.
  iii. Loss of service.
  iv. Damage to systems of 1000s of users.
  v. Loss of life.

## *. COMMON SECURITY VULNERABILITIES :

1. Buffer overflow Attack :



## 2. Cross site Scripting (XSS) :




① Attacker discovers a website for having script injection vulnerabilities.

② The attacker injects a payload in websites database with malicious javascript that steal Cookies.

③ The website transmits the victim's browser the page with the attackers' payload, The victim's browser executes the malicious script.

④ After script execution victim sends his Cookie to the attacker.

⑤ The attacker extracts victim's Cookie, after which he use it for session hijacking.

3. SQL Injection Attack :

If a SQL sever is vulnerable to an injection attack, it may be possible for an attacker to go to website empty fields and type in code that would force these siter SQL sever to dump all of its stored data for this site.

*. IMPLEMENTING BEST SECURE CODING PRACTICES :

(1) Follow OWASP guidelines :

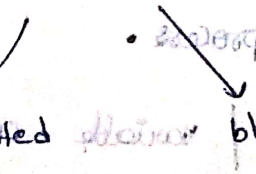
- For web Application security, OWASP enforces secure coding efforts for offering free application testing resources.
- OWASP continuously updates its web security testing guide for software developers to implement.

(2) Avoid unbounded write operators in C++ :

- strcpy (or) strcat in C++ dont have account for the limited capacity of buffers, so they may cause buffer overflow. So use strncpy & strncat operators.

(3) Implement proper input validation :

- Avoid browser (cross site scripting) attacks. check the validity of user input in the browser before executing it.



④ Dynamic Application Security Testing :

- After software has been fully developed, it should then be run through a series of cyber-attack scenarios it might encounter when deployed. (DAST)

⑤ Monitor the security status of your Vendors :

- You may strictly follow secure coding practices & security policies but your vendors may not protect their coding with equal vigor.
- To prevent 3rd party security breaches from affecting your business, a barrier needs to be established between your internal sensitive data & your vendors.

*. SOFTWARE SECURITY FUNDAMENTALS :

- Software security is idea of making software so that it continues to function correctly under malicious attacks.

- Threat – A new incident with potential harm : Attacker, person, malicious code

- Vulnerability – known weakness exploited by hackers : Hacker

| Threat + Vulnerability = Risk. → exploits vulnerability |
|---|

## * SOFTWARE RISK MANAGEMENT :

- Risk Management is a business process that helps to evaluate / Track the risks present in the business environment.

(See back)

Step-1 : Identify Risk : what type of Risk.

Step-2 : Analyse Risk : high | low ? Impact of Risk
how ?

Step-3 : Evaluate Risk : Ranking all your Risks.

Step-4 : Treat Risk : → eliminate
→ reduce severity
→ Ignore (low risk)

Step-5 : Monitor & Review Risk :

Continuously study the risk status, Analyse.

* Three Pillars of Software security :

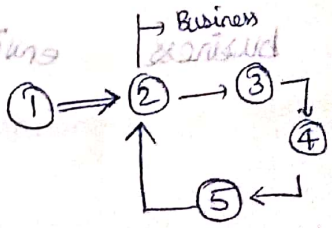
IMP
* 1. Risk Management framework
2. Touchpoints
3. Knowledge

Risk = Vulnerability + Threat

* RISK MANAGEMENT FRAMEWORK :

• Phases :

1. Understanding the Business Context.
2. Identify and link the Business and Technical Risks.
3. Synthesis & Rank the Risks
4. Define the Risk Mitigation strategy.
5. Carry out Fixes & Validate.



Stage 1 : Understanding Business Context

- A key task of an analyst.
- Extract and describe business goals.
- Set priorities.
- Understanding what risks to care about.

Stage 2 : Identify link & Business Technical Risks.

- Business risks impact business goals.
- Identify Business Risks.
- Identify Technical Risks and map them to business goals.
- Evaluating software artifacts.
- Use cases, class Diagrams, and other Unified Modeling language (UML) models, requirements & design documents.

Stage 3 : Synthesis & Rank the Risks.

- Prioritize the risks based on business goals.
- Risk metrics : Risk likelihood, Risk impact, Risk severity.

Stage 4 : Define the Risk Mitigation Strategy.

- Create an efficient strategy for mitigation the risk that takes into account :
  - → Cost.
  - → Implementation time.
  - → Livelihood of success.
  - → Impact of it.
- Identify the Validation techniques. - Techniques to check whether the strategy works or not.

Stage 5 : Fix the problems & Validate the Fixes :

- Implementation of mitigation strategy.
- Application of validation techniques.
- ~~Process~~ Progress is measured interms of "completeness against risk mitigation strategy."

IMP

*. RMF is a Multilevel Loop : (Why ?)

- RMF is a loop because some risks maynot be solved at once like (Business risk or Marketing risk) they need to applied over & over again & restart again to solve contempory problems.
- RMF is a multilevel because this is applied in every stage of Software Development Life cycle. (analysis, design, implementation, testing, deployment, ...)

  In every phase of software development — this is important.

## *. TOUCHPOINTS :

- Software Security Touchpoints specifies one set of touchpoint
- These are set of security best practices.
- These are implimented on security artifacts.
- Touchpoints are mix of destructive & constructive activities.
  - destructive ← attacks, exploits, breaking
  - constructive → design, defense, function

Examples :

1. Code review (Tools) : Static, dynamic Analysis tools.
2. Architectural Risk Analysis : design, specification, artifacts.
3. Pentration Testing : entering into software & testing to find vulnerabilities
4. Risk based Security Testing : System Testing
5. Abuse cases : User validations,
6. Security Requirements : Authentication, Authorization, requirements
7. Security Operations : work together, learn from one another & build secured system.

| | |
|---|---|
| ① Code : Artifact<br>Example : Buffer overflow | ② Artifact : Design<br>Example : Design flaws |
| ③ ~~Artifact~~ : System environment<br>Example : DDOS Attack | ④ ~~Artifact~~ : System<br>Example : Data leakage |

| | | |
|---|---|---|
| ⑤ ~~Artifact~~ : Use case<br>Example : Tampering attack | ⑥ ~~Artifact~~ : Requirements<br>Example : Data protection | ⑦ Artifact : Deployment system<br>Ex : Insufficient logging |

* Types of Penetration Testing :

1. Vulnerability scanning
2. Network scanning
3. Log Views
4. Penetration Testing
5. File Integrity checkers
6. War Dialing
7. Virus Detect
8. Password Cracking

## (III) KNOWLEDGE : (3rd pillar of Software security)

- It involves gathering & sharing security information.
- It includes : Principles, Guidelines, rules, vulnerabilities, risks, attacks.

* SECURITY PRINCIPLES :

→ CIA Model : Confidentiality, Integrity, Availability

→ Privacy : Keep it private

→ Identification & Authentication.

→ Password Management, Generator, Checker, Analyser.

→ Honeypot : A fake info stored inorder to get notified when attacker uses this to login.

→ Limited login attempts

→ Token devices - (Security Questions, Answers)

## *. SECURITY GUIDELINES :

- Physical security.
- Physical security perimeter (boundaries).
- Physical Access Controls, Monitoring this.
- Security operation Centre
- System and network security.
- Incident Response & Actions.
- Protecting Sensitive Information.

## *. RULES & POLICIES :

- Rules for granting, controlling, monitoring & removal
- Roles and responsibilities.
- Infrastructure security - Access Control (Admin priviliegs, access controls)
- Asset policies - BYOD. (Admin) (Bring your own device).

## *. VULNERABILITLES & EXPLOITS :

1. Buffer overflows
2. Integer overflows
3. Format string attack
4. Shell Code injection
5. Failing to handle errors
6. Cross site scripting (XSS)
7. Failing to protect Network Traffic
8. Improper use of SSL, TLS
9. Use of weak passwords
10. Improper Access Control
11. Information Breach / Leaks
12. Failure to store data Securely, privacy issue.