DOMAIN 6

CH: IS SECURITY ASSESSMENT & TESTING.

So far

Build and Implement security controls

This chapter

Regularly test security controls and ensure they are working properly, and effectively safeguarding information assets.

COMPONENTS OF

Security Assessment Program

1. Security Test
2. Security Assessment
3. Security Audits

P.T.O

Is security control functioning Properly?

- configuration Drift
  - check operational evidence
    - Automated Scans
    - Tool assisted Pen tests
    - Manual attempts

Risk assessment / Exposures

NIST 800-53A

Four components of Assessment

P.t.o

NIST 800-53A

1. **SPECIFICATIONS:** Docs associated with audit
   - Procedures
   - Policies
   - Requirements
   - Specifications

2. **MECHANISMS:** These are controls within Information system & to meet specifications.
   - Based on
     - hardware
     - software

3. **ACTIVITIES:** Action carried out by People within Information System.
   - Perfor Backup
   - Log export
   - Review A/C histories

4. **INDIVIDUALS:** People who implement
   - specifications
   - Mechanisms
   - Activities

Audit is different from Assessment & Testing

internal use only

To show buyer/3rd party that security doors are installed with triple lock.

House is clean from inside. security doors/windows are working for internal protection.

Auditors = Unbiased view

Internal Assessment & Testing = likely biased

3 Types of AUDITS

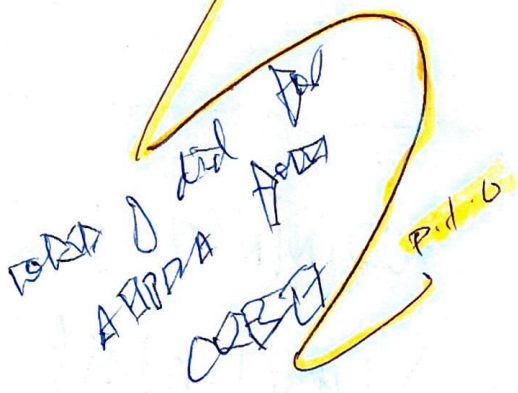
INTERNAL

- Internal Audit staff for internal Audience
- These audits normally have a reporting line that is completely independent of functions they evaluate.

EXTERNAL

- Auditing firms
- Has no conflict of interest with organization.

THIRD-PARTY



Third-party Audits

- Conducted by / on behalf of other organization

↓

Burden if large number of client

→ AICPA released SSAE 16

(Statement on standard for Attestation Engagement Document 16)

Provides one common standard for auditing

Type I report SOC

- Description of control + opinion of auditor based on control description
- No actual testing of control (-VE)
- Controls are implemented as they described

RELIABLE Type II report SOC

- Covers min. 6 months + opinion of auditor on effectiveness of the control
- Actual testing of control by Auditor (+VE)

AUDITING STANDARDS

- COBIT – Control objectives for Info. & related technologies
- ISO 27001
- ISO 27002

ISO – International organization for Standardization

PERFORMING VULNERABILITY ASSESSMENTS

Actually, it's vulnerability testing tools, not vul. assessment tools.

Vulnerability Scanning + Pen Test → To find weakness in System / Applications technical controls.

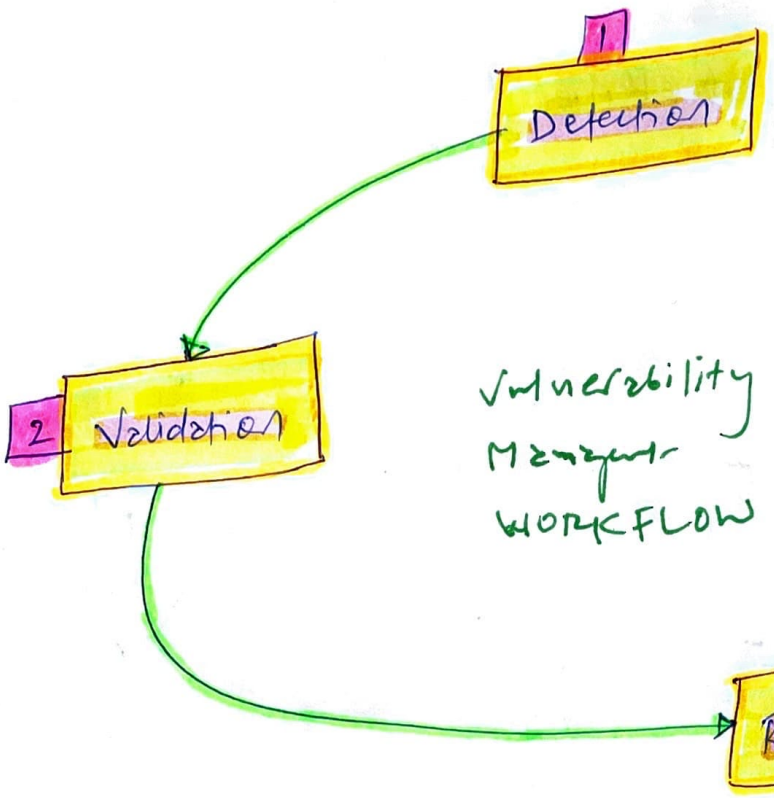
1. Network Discovery Scans
   - opens half-connection → TCP SYN scanning
   - opens Full-connection → TCP connect scanning
   - TCP ACK scanning
   - Xmas scanning.

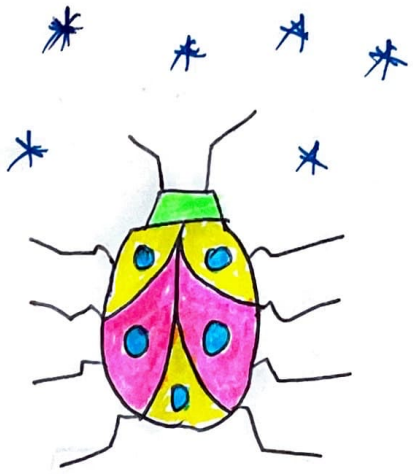
   (NESSUS, Rapid7) NMAP

2. Network Vulnerability Scanning.
   - Scan open port + find known vuls.
   - False negative is dangerous than False positive
   - To Reduce → To perform authenticated scans over unauthenticated scan

3. Web Vulnerability Scanning
   - Similar scan to N/w vul. but scan more on web application structure.
   - Web App scan required for compliance such as PCIDSS.

4. Database Vulnerability Scanning
   - sqlmap: open source tool to detect vuls.
   - Database security issues (ch:9)





Vulnerability Management WORKFLOW

Prioritize vulnerability remediation based on:

- Severity of the vulnerability
- Likelihood of the exploitation
- Difficulty of remediation

REMEMBER AGAIN

Security vulnerability assessment is NOT about assessment, but it's about Security testing.

Penetration Testing

Phases

RECONNAISSANCE

Planning → Testing scope is crucial. UI + backend

SCANNING

Info. Gathering & Discovery ← Perform Reconnaissance → Automated / manual tools to collect info. about target system or environment. E.g. nmap for n/w discovery.

Vulnerability scanning → Database vul. scan; Web vul. scan; Network vul. scan

EXECUTING

Exploitation → ~~Kali Tools~~. Metasploit tool: for common attacks. Automatic exploitation on target system.

Reporting → Summary of test + recommendation; Raise risk

Ethical hacker use only this type of pentest method

4. Double Blind

3 Kinds of Pen Test

1. Whitebox Pen Test

- Provide full targeted system to Attacker
- short time
- likelihood to find security flaws.

2. Gray Box Pen Test (Hybrid)

- Balanced / Hybrid / Partial knowledge test
- Use when you want Black Box test but have no time & money

3. Blackbox Pen Test (Si

- Doesn't provide any info to Attackers
- more time + money $$$
- Simulates genuine external attack

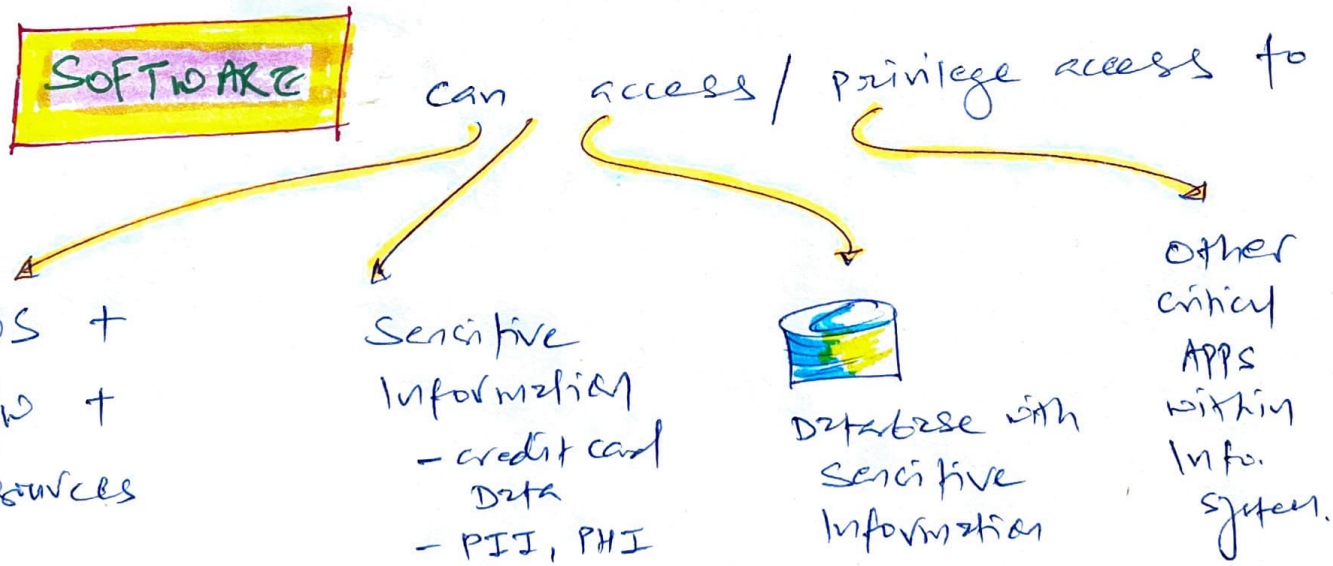
Industry standard Pentest methodologies

- OWASP testing guide (Application security)
- PCI DSS Information Supplement (Financial Data Security).

so far.....

SOFTWARE TESTING.

why take software seriously ?
or
why software is the heart of modern enterprise



Software testing → Essential to CIA requirement of modern Organization

SDLC / ch: 20.

Critical components of software testing

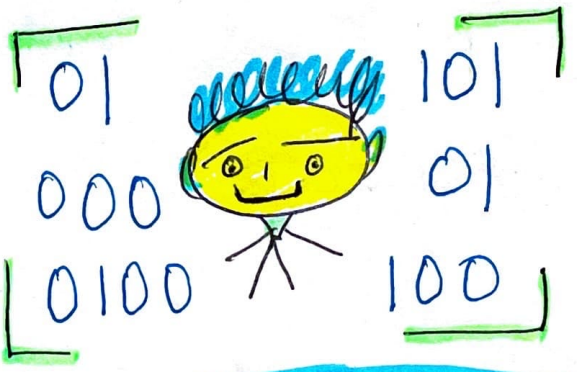
Code Review

FAGAN = most rigid code review process

1. Planning
2. Overview
3. Preparation
4. Inspection
5. Rework
6. Follow-up

Testing

- static (SAST)
- dynamic (DAST)
- Fuzz
- Interface
- misuse case
- website monitoring



Human can do below types of code review

- software inspections
- software walkthrough
- Code Review

But not this

- Static program Analysis as it's done by Automated Tools

Static testing

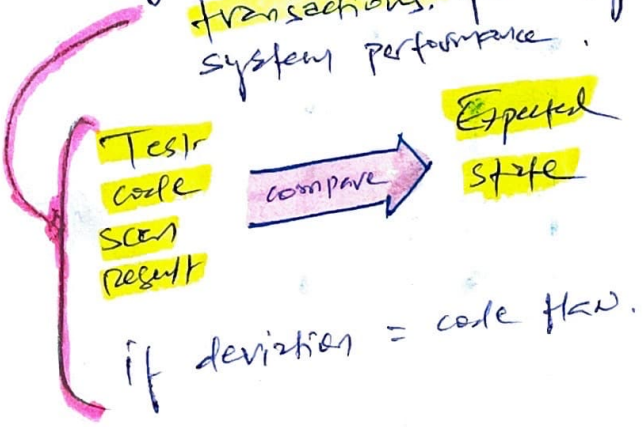
- Evaluates software security without running software. Instead, it analyze / scan source code.

- SAST: We can use this during Pre-prod (Design, build & test) of CI/CD pipeline

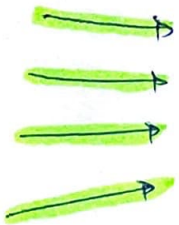
Dynamic Testing

- Not focused on / scan source code as apps / s/w is deployed in prod / runtime environment.

- May include synthetic transactions to verify system performance.



Fuzz Testing ← Black box testing



S/W = s/w crash, buffer overflow, undetected flaws.

2 types

Dumb Fuzzing (Mutation)

- Fuzz tool manipulates input (bit flipping) to confuse s/w (Osh page 684-685)

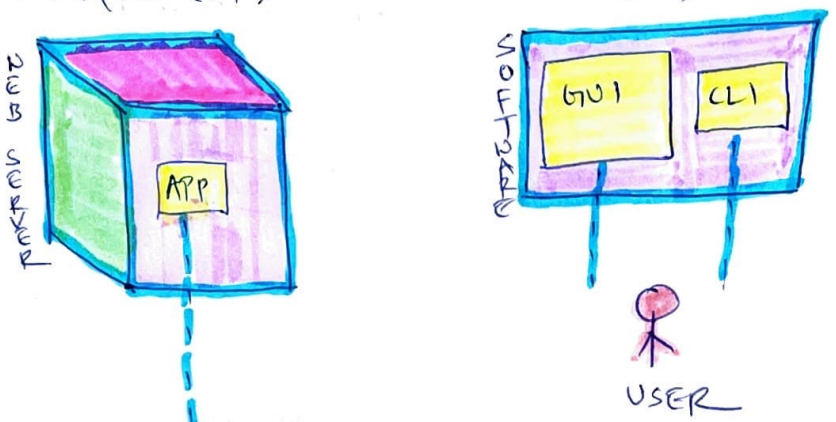
Intelligent Fuzzing (Generational)

- Develop data models to create fuzz input based on the understanding of types of O/p.

Interface testing

3 types of Interface to test during s/w testing :

- Application Programming Interface (API)
- User Interface (UI)
- Physical Interface



User clicks, Data requests, Login, Url

Website monitoring

Passive

- Real-time live traffic for real user interaction
- Detect issues for real user after it occurs
- Tool: Real User monitoring (RUM)

Synthetic / Active

monitoring reports

- Initiate artificial transaction with website to check performance
- Can't detect real user issues but we can detect issues before it actually occurs.

Variant of passive monitoring, it records users interactions with App / website to ensure quality and performance

Misuse case Testing

s/w developed use → - Abuse case testing to evaluate vulnerability of s/w to known risks

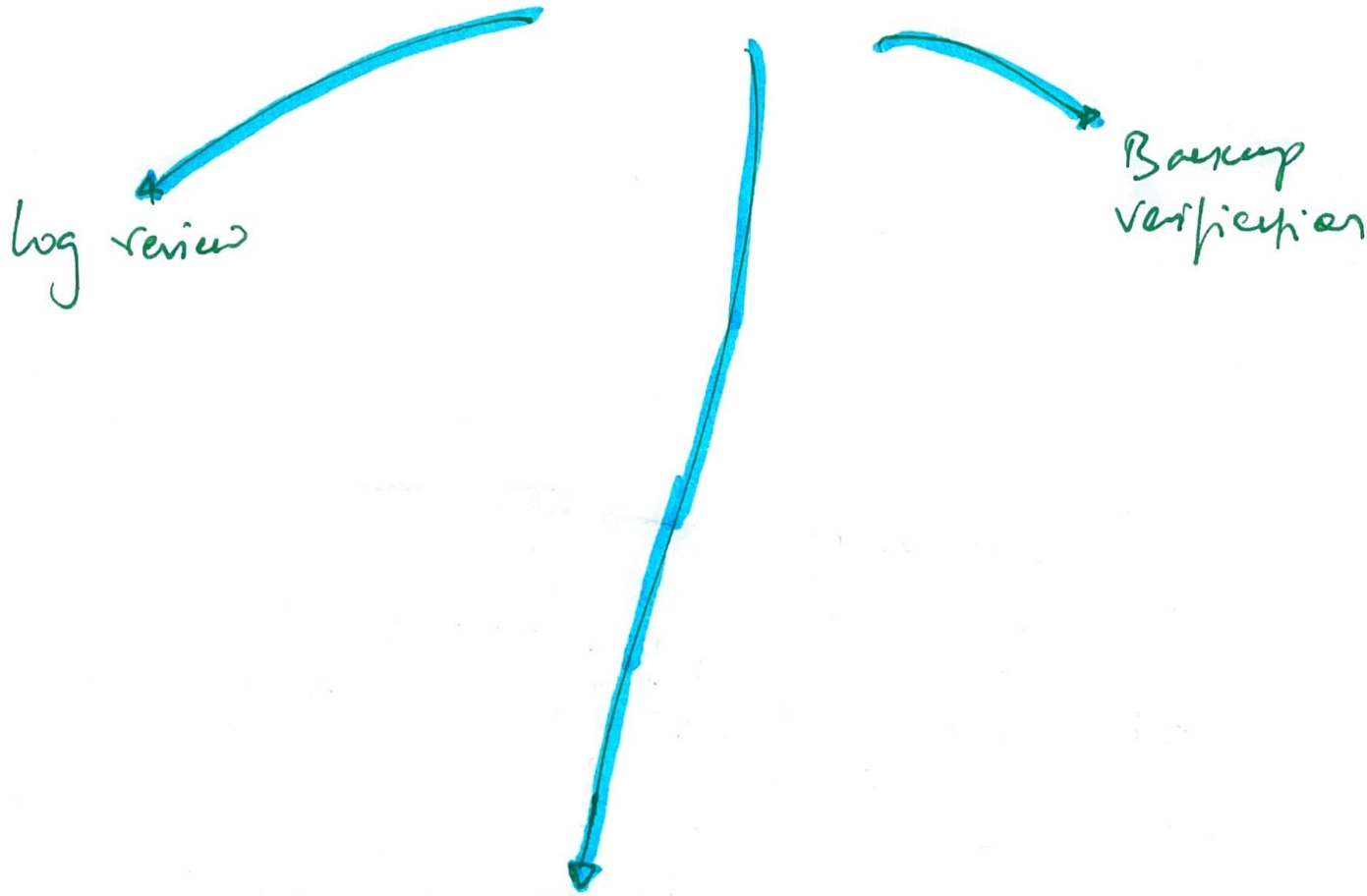
Test coverage Analysis — For new softwares

s/w testing professionals use below formula for new s/w

$$\text{test coverage} = \frac{\text{no. of use cases tested}}{\text{total no. of use cases}}$$

- Branch coverage - if/else conditions
- conditional coverage - logical test in the code
- Functional coverage
- Loop coverage - conditions that cause code execution multiple times
- Statement coverage - test every line of code

Implementing Security Management Process



Post Q&A - wrong answer topics

* Describing Vulnerabilities

NIST provides SCAP (Security content Automation Protocol)

A framework to describe common vulnerabilities.

Components of SCAP

CVE
- Common Vul. & Exposure
- Identifies vul. generated by diff. security Assessment Tools
- consistent naming system/ reference to identify vul.

CCE
- Common config. Enumeration
- Naming system for system configuration issues.

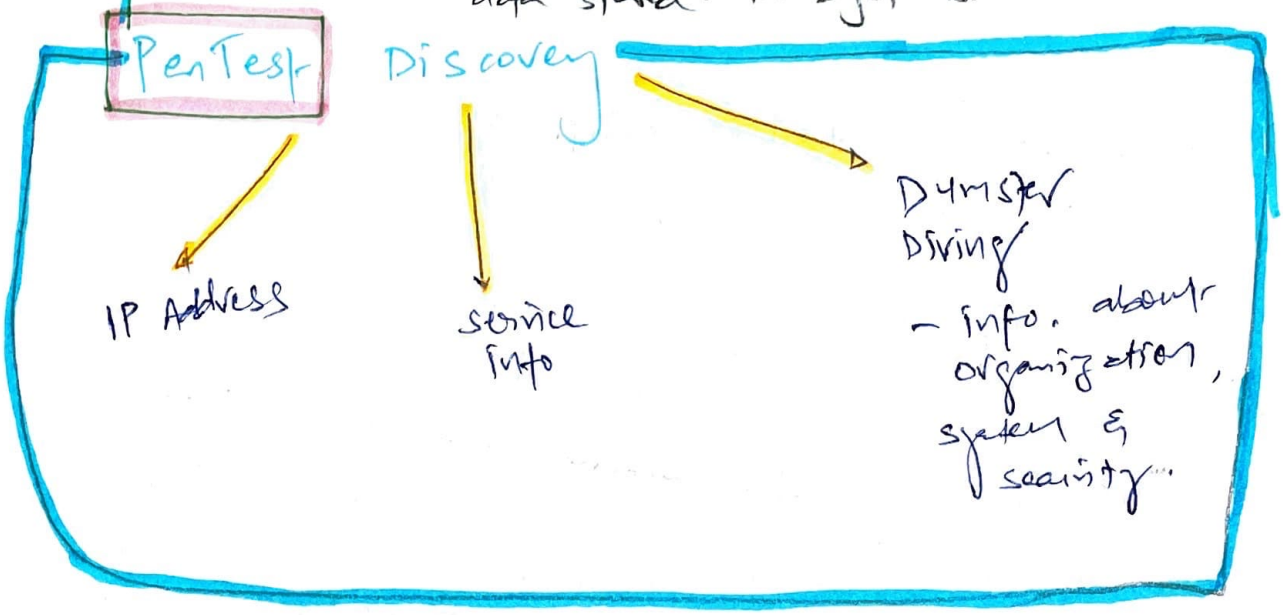
CPE
- Common platform Enumeration
- Naming system to refer consistent name for operating system

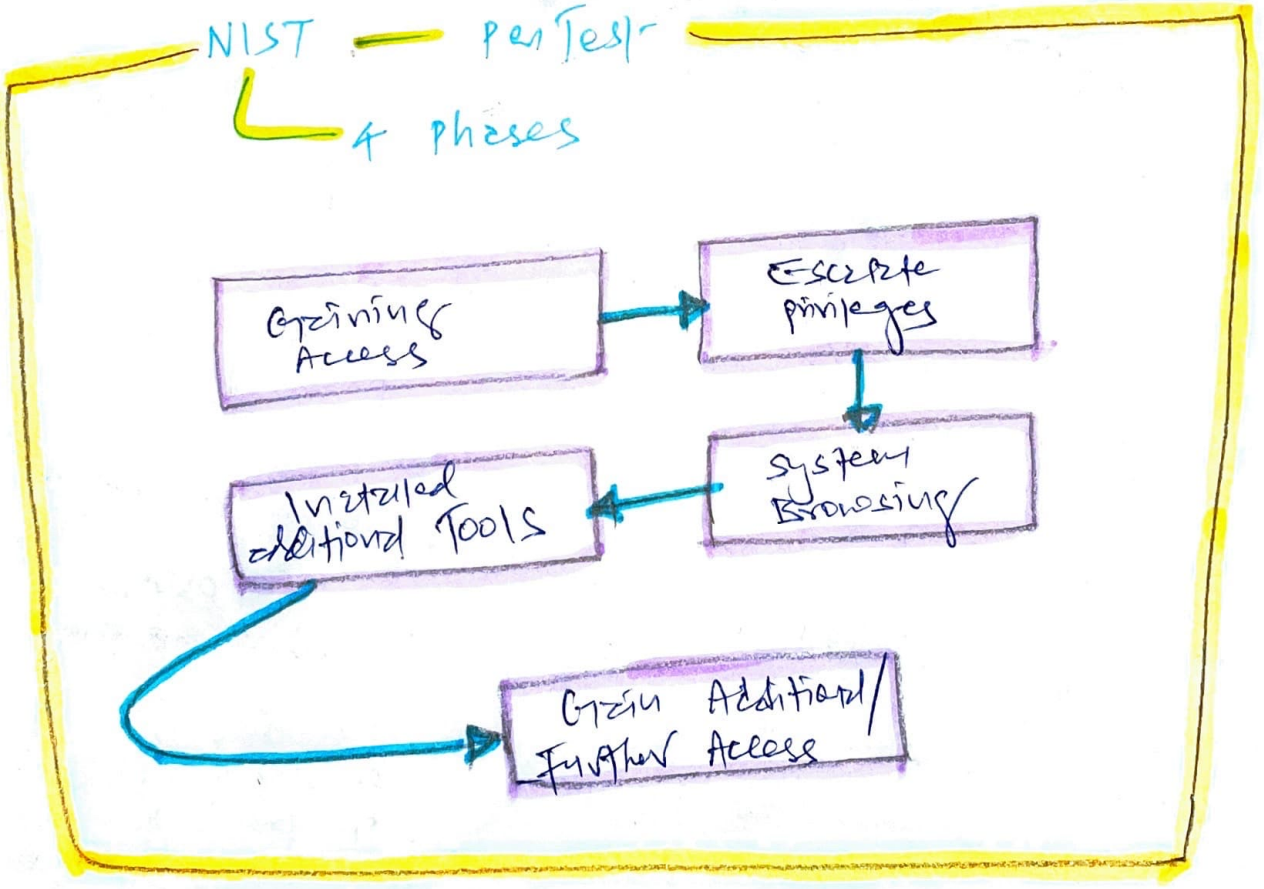
CVSS
- common vul. scoring system
- Describes severity of security vul.

OVAL
- open vul. & Assess. language
- Describes security condition of system
- Provides language for describing security testing procedure.

XCCDF
- Extensible configuration checklist Description Format
- Provides language for specifying security checklist

Hazard: May disrupt system access or corrupt data stored in systems.





PenTest Report doesn't have sensitive Data. it has list of vuls, mitigation guide & risk rating.

Port
- is open, no access to App — closed
- is open, access to App — Open
- not accessible — Filtered (could be firewall)
  (Port could be open or close)

Reports
- Test coverage Report
  - how many test cases have been completed
  - Provides metric when uses test cases
- PenTest Report
  - upon conducting PenTest
- Line coverage Report
- code coverage Report
  - How much code has been tested

kind of

Coding issues → Bound checking, Input validation, pointer manipulation

Environment issue → Race condition
- code testing in manual & prod arise in fixing issue

Unlikely to find in web vulnerability scanner

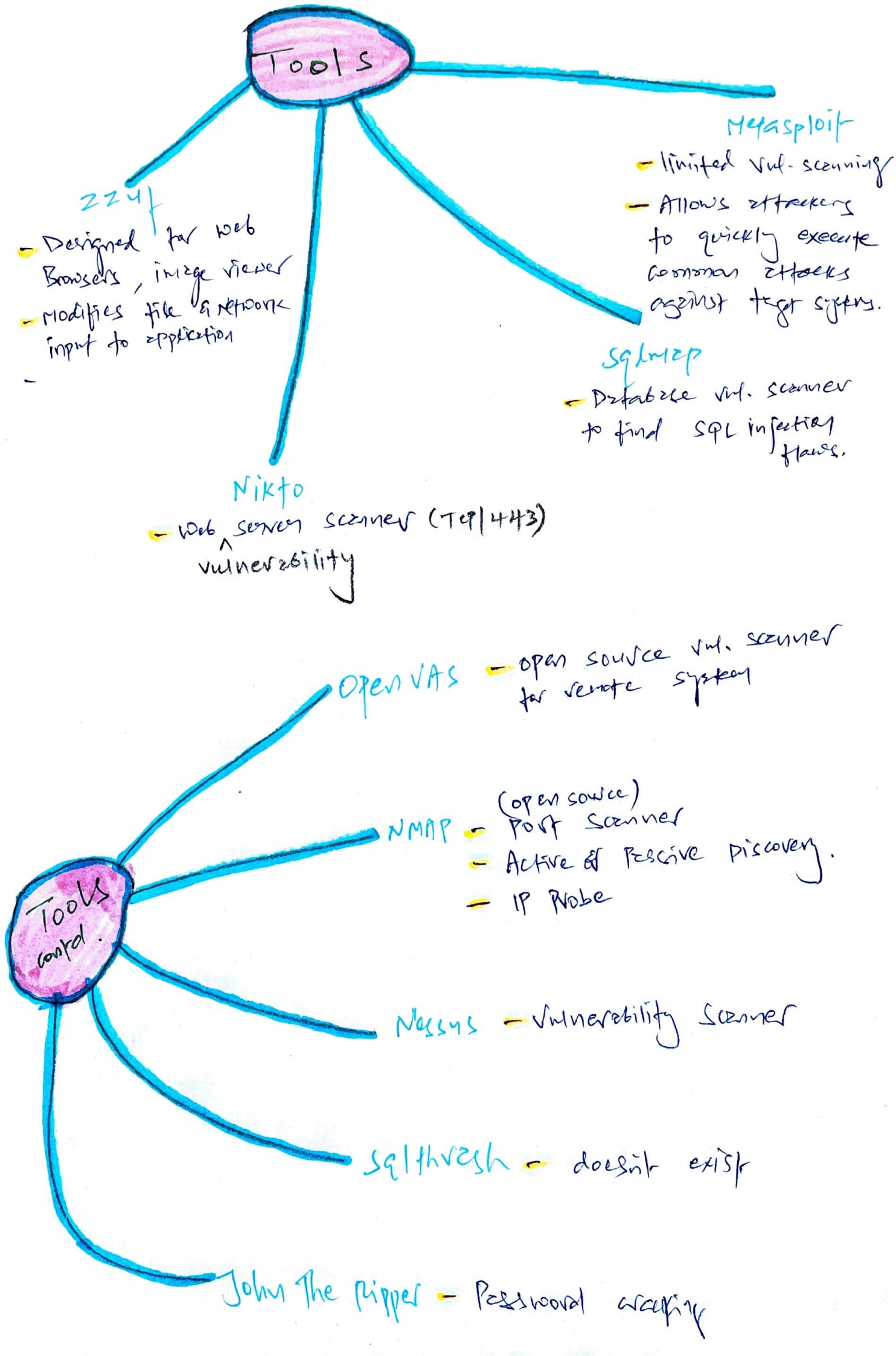
s/w Testing

Static → Analyze source code without running software

Dynamic → NO source code, for runtime
→ Synthetic transactions to verify system performance

Fuzz → Mutation (Dumb)
- modifies program in a small way
- takes input from actual operations manipulates to create fuzzed input

Generational (Intelligent) — 224+ tool
- Develops data models & creates fuzz inputs


Tools
ZZUF
- Designed for web Browsers, image viewer
- Modifies file & network input to application
Metasploit
- limited vul. scanning
- Allows attackers to quickly execute common attacks against target systems.
sqlmap
- Database vul. scanner to find SQL injection flaws.
Nikto
- Web server scanner (TCP/443)
vulnerability
Tools contd.
OpenVAS - open source vul. scanner for remote systems
NMAP (open source)
- Port Scanner
- Active & Passive discovery.
- IP Probe
Nessus - Vulnerability Scanner
sqlthrash - doesn't exist
John The Ripper - Password cracking

WEP ——— lower security

WPA 2 ——— Better than WEP

- Enterprise mode
  Use RADIUS Authentication
  rather than preshared key

For consistent logging to SIEM

Use Group policy, not windows client

Preemive Monitoring

only works after issue has occured
or if required
actual / real-time
traffic

Synthetic Monitoring

Use simulated / artificial traffic
as proactive approach
to identify issues
before they occurs

only if it's in test script