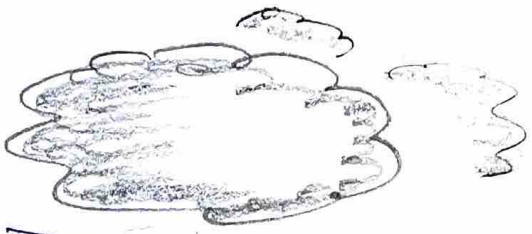
* Cloud-Based Systems and Cloud computing



PCIDSS → Answer
HIPPA
GDPR

Cloud is good but

- Privacy concerns
- Compliance difficulties (Regulations) (Jurisdiction)
- Is data secured there?
- Standards
- Inconsistent security policy b/w cloud provider & organization
- Lost privacy control

Hypervisor
(VMM - virtual machine monitor)

→ create, manage & operate virtual machines

Type I Hypervisor (Runs on BARE METAL Native)

- No host OS
- Hypervisor directly installs onto hardware where host OS reside
- used to support Server virtualization
- most efficient as it requires no OS loading
- highly secure as it eliminate OS vulnerabilities

max. utilization H/W, reduction of risk by host OS

Type II Hypervisor – Runs on top of OS

- Regular H/W OS + hypervisor instal another S/W app.
- Used for Desktop deployment (Guest OS, Sandboxing)
- cheaper than type I but not secure

Cloud storage

- cost effective backup but not high speed
- ? Is data secured?

Elasticity

Flexibility of virtualization resources on demand

- Autoscale

Cloud concepts

CASB — **DLP**

Gmail not responsible if you leak confidential info. Use below solns.

PaaS

- AWS Elastic Beanstalk
- Azure
- ~~[illegible]~~

H/w & S/w tools available on Internet

- looks after infra, physical security, maintenance but you have to worry about software & code

SaaS

- Google Drive
- MS 365

S/w by 3rd party on Internet

IaaS

- AWS EC2
- Google Compute Engine (GCE)

Cloud-based service, pay-as-you-go for storage, networking, virtualization

Deployment concepts

Hosted Solution

Organization owns/license s/w, then operate + maintain the s/w

Cloud Solution

Organization outsource h/w + s/w to 3rd party cloud provider for monthly fee.

- **Private cloud**
  - Virtual private cloud
- **Public cloud**
- **Hybrid cloud**
- **Community cloud**
  - shared benefit + collaborative work

Investigate Security of cloud provider

- Compliant?
  - PCI DSS
  - HIPPA
  - SOX
- How they do Disaster Recovery?
- Review patch management policy
- How they respond to zero-day attacks?
- How often they perform Pentesting?
- Review their Data Encryption solutions
  - How Data is backed up?
  - How they destroy data?
- How they provide availability of resources?

Is CASB Implemented?

(Cloud Access Security Broker)

- To enforce Proper security measures are implemented or not.

Security as a Service (SECaaS)

- cloud provider concept
- From managing security locally to online = reduce cost + overhead

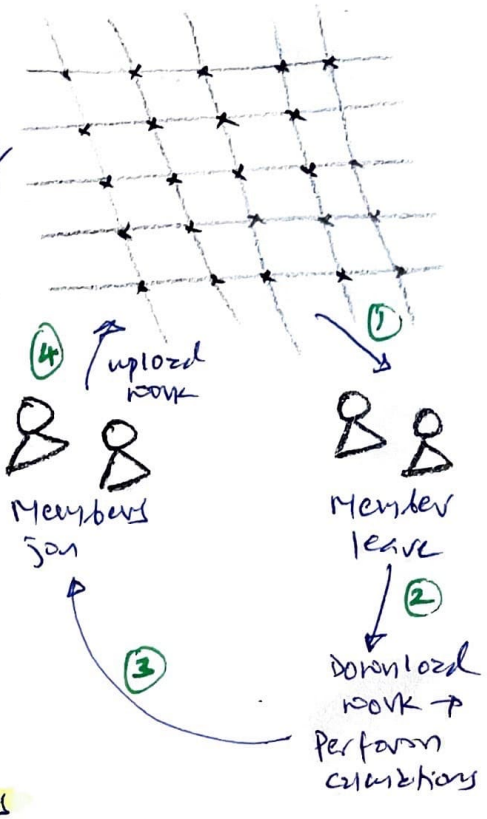
Cloud shared Responsibility Model

Who will do?

- Security
- Troubleshooting
- management

* Grid Computing

- Form of parallel distributed processing that loosely groups a significant number of processing nodes to work towards a specific processing goal.



Security concerns

- Contents are exposed to the world = no confidentiality / privacy of data
- Compromise of grid server = leverage grid members to perform malicious actions.

* Peer to Peer (P2P) - TORRENT

Networking & distributed application solution that share tasks & workloads among peers.

Similar to grid computing with 2 differences

- No central management system
- Provided Services = Realtime Not collection of computing power

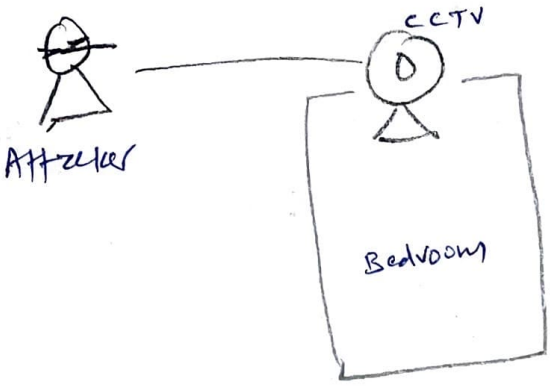
Security concerns

- Eavesdrop distributed content
- Pirated copyrighted materials
- No control mgmt / filtering
- Can consume all available bandwidth

INTERNET OF THINGS (IoT)

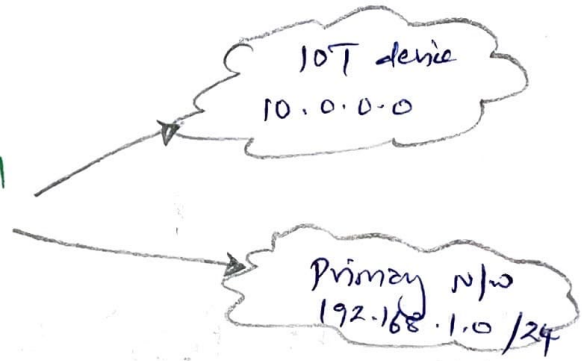
Security concerns ---- NIST Initiative for IoT

- Access
- Encryption



isolate IoTs into seperate network

Secure Implementation for IoT (THREE DUMB ROUTERS)



INDUSTRIAL CONTROL SYSTEM (ICS)

- ICS controls industrial processes and machines.

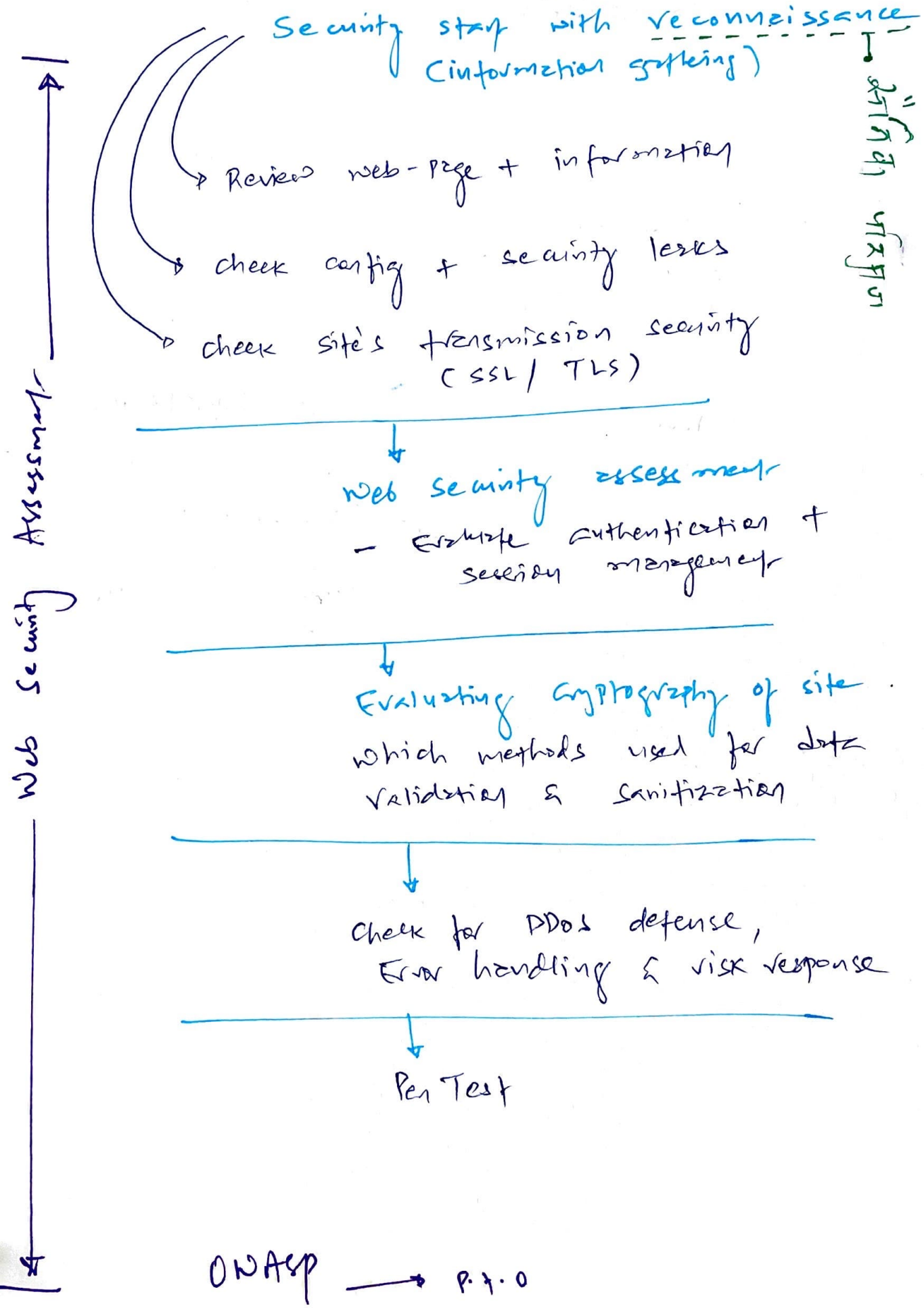
3 forms

- Distributed control system (DCS)
- Programmable Logic Controller (PLC)
- Supervisory control and Data Acquisition (SCADA)

Needs less Human interaction = little security was Built

Until Stuxnet delivered root kit in SCADA

ASSESS AND MITIGATE VULNERABILITIES IN WEB-BASED SYSTEMS



Few of OWASPS Top 10 Web Risks

INJECTIONS

SQL Injection

Use of unexpected / malicious code as query to compromise web application / database.

↓

static to dynamic web page story on How did we reach here.

2 techniques

- Input Validation
  - type + Format & length of input
  - E.g six numeric for DoB
  - 230283
- Limit account Privilege
  - least privilege

LDAP Injection

Focus on LDAP directory, not database

XML injection

Backend target is XML Application

- Sanitizing of input & defensive coding as prevention

XML Exploitation

- Programming attack either use false information to visitor or cause system to give-up information without authorization.

↓

SAML abuse = web-based authentication

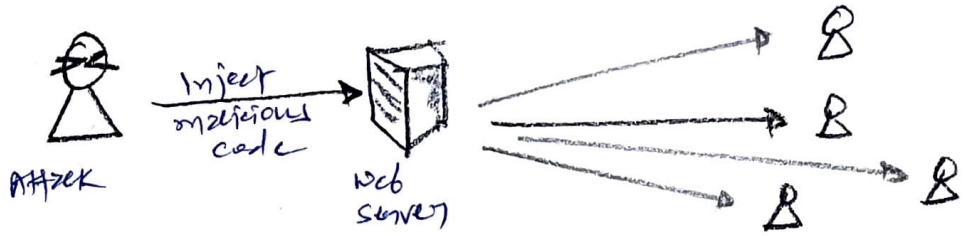
Used for web SSO solution

Attacker can falsify SAML communication or steal visitor's access token, to bypass authentication & gain unauthorised access to site.

SAML = XML use

Cross-site Scripting (XSS)

- Malicious-code injection attack



- Defense → WAF
  - Patched web server
  - HIDS
  - Input validation
  - metacharacter filtering.
  - Avoid malicious / non-mainstream websites
  - Run Antivirus s/w

Cross-site Request Forgery (XSRF)

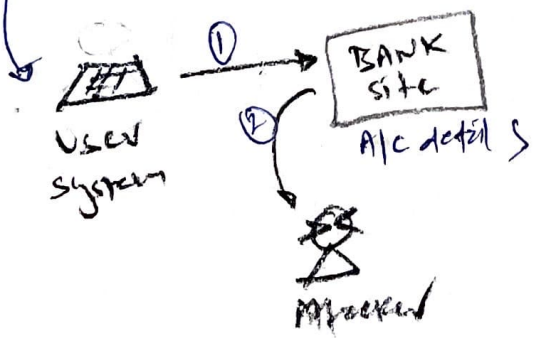
- Attacker Focus = visiting user's web browser, not visiting website

Purpose

To make user / user's browser to perform unattended action
- Make a purchase
- Download A/c details
- change A/c information

XSRF

Eg: ZEUS



Defense

- Sensitive action = Reauthenticate (Netcode)
- CAPTCHA → Human / machine
- Nonce: Add random string to each URL + session mgmt
- WAF, HIDS
- Patch browser, clear cookie + temp Internet files.

ASSESS AND MITIGATE VULNERABILITIES IN MOBILE SYSTEMS

To steal internal confidential data & expose to outside world

⇩

WIKI LEAKS

Dangers of Mobile Devices

- Camera to capture sensitive information
- USB connection to transfer files/ photos/ video
- Eavesdropping/ conversations
  - Be careful when talking in public place

Mobile Security

- Device Security
  - No value of feature till they are enabled
- Full Device Encryption
  - Good feature but useless if device is stolen or system has backdoor vulnerability
  - Voice encryption makes eavesdropping useless.

Remote wiping (Remote Sanitization)

- If device is stolen but no guarantee of data security
- Encrypt device so even after recovery, attack struggle to decipher

Erase all data if incorrect code entered 10 times

Lockout

3 attempts = Account / device is locked

Screen Locks

Pattern, PIN, FaceID

GPS

- Find my iphone

Application control

- Limit which APP can be installed = reduce exposure for malicious apps

Storage segmentation

Mobile device

- Artificially compartmentalize various types of data on a storage medium

Asset Tracking (Active / passive)

To verify that device is still assigned to authorised user. Prevent / locate missing devices.

Inventory Control

Mobile camera = scan bar codes to track physical goods

Mobile Device Management (MDM)

manage
- BYOD
- company device

security, mgmt, monitoring & troubleshooting of devices

Device Access control

- To reduce unauthorised access
- MDM to force screen-lock configuration

Removable storage

Mobile → micro SD
Mobile → USB
Mobile → Bluetooth / Wifi

Disabling unused features

- Turn-off SIRI
- Disable location based service

Application Security

Key management
- Don't store keys locally on mobile. Consider TPM or removable hardware
- Google KMS

↳ Hashicorp Vault

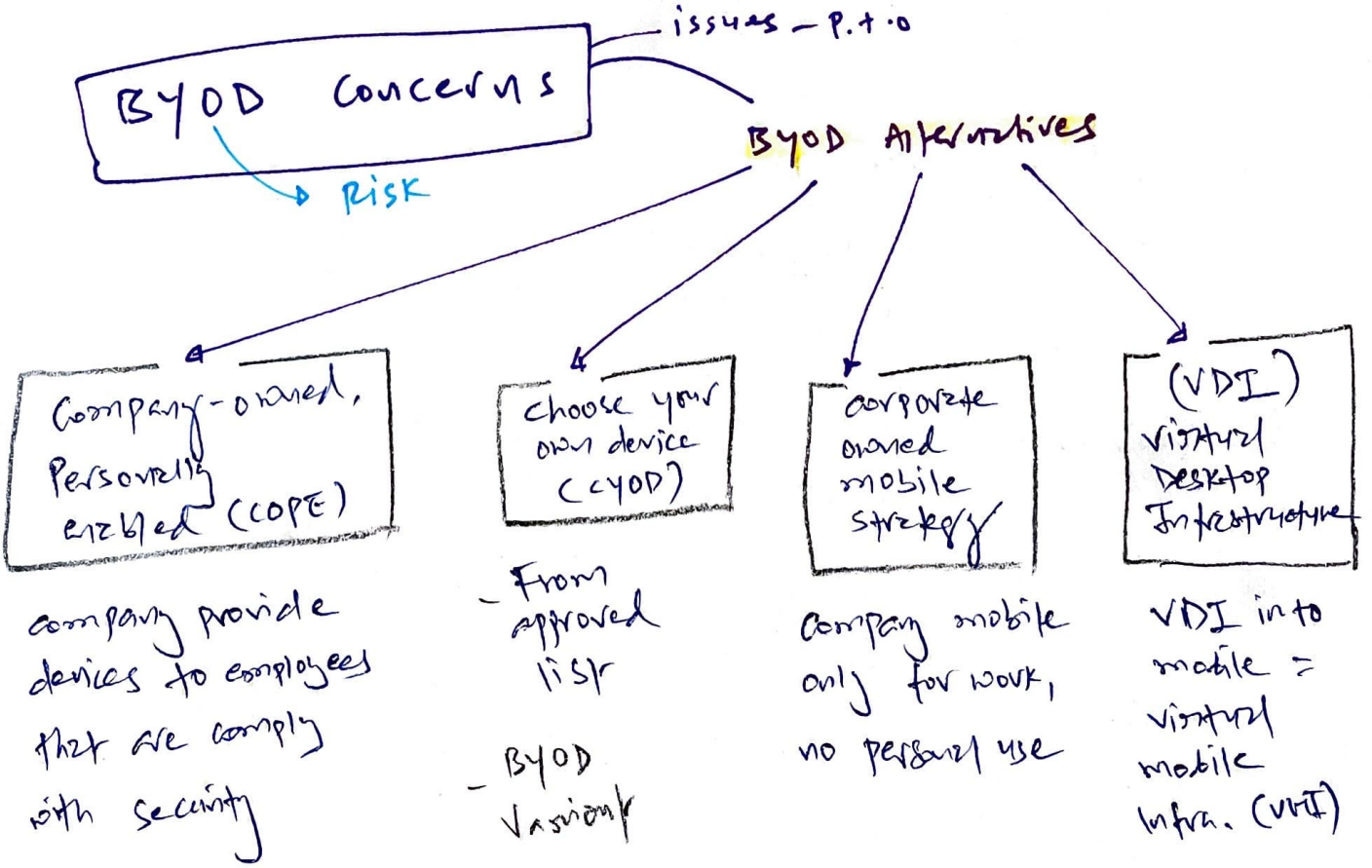
Geotagging
Don't like Insta Photos & Google reviews attached with locations

Credential management
- Cyber Ark

Encryption
- At Rest
- In Transit

Authentication
To combine
- Consider multi-factor authentication + device encryption

Application whitelisting
- Implicit deny
- Prohibit unauthorised software from being able to execute.



BYOD Related Issues

Data Ownership

Problem
- Have a clear policy as remote wipe of business & personal data = individual's significant loss

Solutions
- MDM solution = data isolation + segmentation
- Backup solution for personal & business data in case of remote wipe

reduce the risk of data loss in event of remote wipe / device failure / damage

Support Ownership

- Policy for = if employee phone → theft / repair / damage

who will do it?

Patch Management

- Policy for mobile device updates
  - is user responsible?
  - Enforce patch / updates

Antivirus Management

- Policy should dictate which Antivirus installed on mobile device

Forensics

- Make mobile user aware = device will involved in case of security violation / crime

Privacy

- Policy should address privacy & monitoring
- Employee to agree of tracking work mobile after business hours.

On-boarding / off-boarding — have policy for both

- On-boarding
  - Install / keep update
  - Security mgmt
  - Business Apps
- off-boarding
  - Full wipe
  - Factory Reset

Adherence to corporate Policies

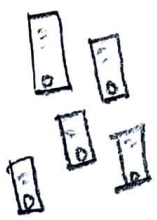
Personal ~~work~~ mobile as BYOD work → comply → work + security policy

→ x Tinder / Personal use

User Acceptance

organization → Educate About Personal mobile Policy → User → Acceptance → on-boarded

Architecture / Infrastructure Considerations



Move Personal BYOD mobile Device

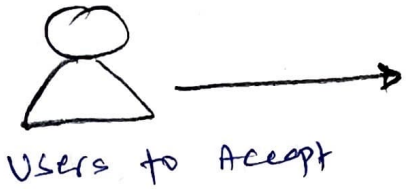
- Increase Intra coop $$↑
- IP Assignment
- Load on IPS / IDS
- Bandwidth consumption
- Wifi congestion / interference

Legal concerns — Attorney has job to do!

Personal Devices for Business Activities

- → Increased burden / liability
- → Risk of data leakage

AUP : Acceptable Use Policy

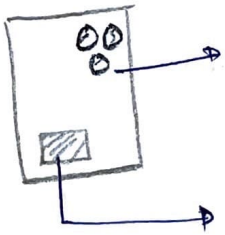


Users to Accept

Company policy (OR) Personal Device policy

Restrict user to access inappropriate content + information disclosure

On-board camera / video + storage



when not to use camera

what they can / can't store

define in policy

Assess and Mitigate Vulnerabilities in Embedded Devices and Cyber-Physical Systems

Don't alter need / surprising elements

Static Environment / system

Conditions / Events that don't change = Reduce risk / security

Embedded System

- Designed around a limited set of specific functions in relation to larger product of which it's a component.

N/w Enabled devices
- Smart appliances, smart TV
- medical devices,
- n/w attached printers

OS
h/w
s/w
n/w
→ Configured for specific need / function & set to remain unaltered

Examples of embedded & static systems

Cyber-physical systems
- Robotic = movement Element
- Sensor = physical condition

Extension = IOT

Devices that offer means to control in physical world.

Mainframe systems
- Perform high complex calculations & provide bulk data processing

Home consoles

Vehicle-computing systems → TESLA

No hair about security? — P.T.O

IOT
P.T.O End

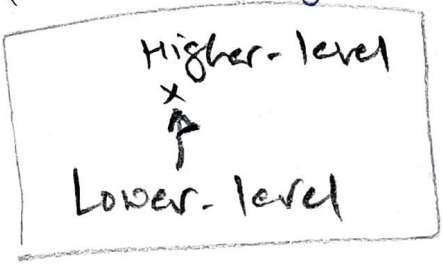
Methods of Securing Embedded and Static Systems

Network Segmentation

- Assign different network range for IoTs.
- VLAN, Application filtering, Routing, Access control mgmt.

Security Layers

- different level of classification / sensitivity grouped together & isolate from other group.



- Isolation
  - Physical = N/W segmentation, Air Gaps
  - Logical = classification

Application Firewalls + Network Firewalls

Manual updates

- First read release notes before upgrade / downgrade

Firmware Version Control

Wrappers = controlled channel to check integrity & authentication before manual updates are applied

Monitoring - SIEM

Control Redundancy & Diversity = Availability

Don't rely one security solution = Defense-in-depth

ESSENTIAL SECURITY PROTECTION MECHANISMS

Everything starts from here:

Software should not be trusted.

⇓

SECURE ARCHITECTURE

Technical Mechanisms

- Layering
  - similar to Ring model
  - centre of ring = Highest level = more privileged
  - Lowest = least privileged
  - sensitive info = core circle
  - less secure info = outer rings
- Abstraction
  - ~~Access control per groups~~ (AD Groups)
  - P.t.o - Revised misunderstanding
- Data Hiding = containers
  - multilevel secure system
  - Placing objects in security containers
- Process Isolation (logical)
  - OS enforce process seperation with dedicated memory space
  - VM
- Hardware segmentation (physical)
  - Same enforcement as process but use of physical hardware

Policy Mechanisms

- Principle of Least privilege
  - Run in user mode whenever possible
  - More process in privilege mode = VM + malicious $$$
- Seperation of Privilege
  - Builds on top of least privilege
  - Granular level access = Permission per privilege
  - Least privilege for Administrators.
- Accountability
  - Need authorization & authentication system
  - \+
  - Auditing & monitoring system

Common Architecture Flaws and Security Issues

Covert channel

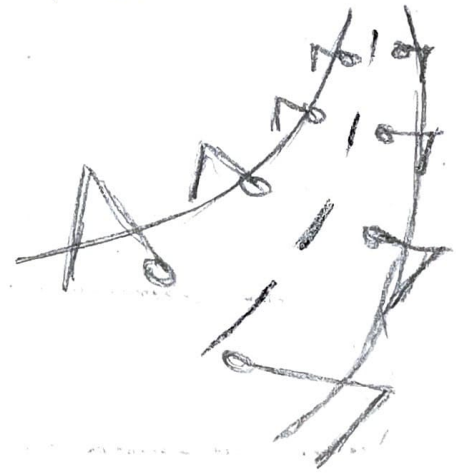
Road less travel : a path not usually for passing information because it's not protected by system's normal security controls.

covert timing channel

- Pass information by altering resource's timing
- secretly transfer data, hard to detect

covert storage channel

- convey information by writing data to storage so other process can read it.



Defense to covert channel = Audit & Analyze log files of covert channel

* Attacks Based on Design or Coding Flaws and security issues

- Seperate testing for security issues = trending.

P. 1. 0 → common attack source or vulnerability of security architecture

↳ Trusted Recovery

- Ensures security controls remain intact in the event of system crash.

→ Database crash while writing data classified as top-secret

↳ unprotected system = unauthorized access

→ Trusted Recovery = ensures no confidentiality violation occur during crash

↳ Input and Parameter checking

Most notorious security violation = buffer overflow

- Happens when we don't have sufficient limit on input data
- Programmers to blame who ignore proper data validation in the code

↳ Maintenance Hooks & Privilege Programs

Back Doors = only developer know the hidden entry point in the system

originally designed for maintenance activity

Common system vulnerability = Practice of executing a program whose security level is elevated during execution.

↳ Incremental Attacks

slow, gradual incremental attacks

DATA DIDDLING

- Subtle damage on system storage, input/output
- Hard to detect unless files are protected from integrity check or encryption
- Tripwire : good tool to address data diddling

SALAMI ATTACK

- Metaphore : Stealing thin slice of salami by customer each time

↯

deducting small amount from financial records routinely.

- Defense : seperation of duties + proper control over code

↳ Programming - other flaws

Programs that doesn't handle exception well = unstable state

Write a secure code. Difficult. Not impossible.

If attacker successfully crash the program = ↓ Gain

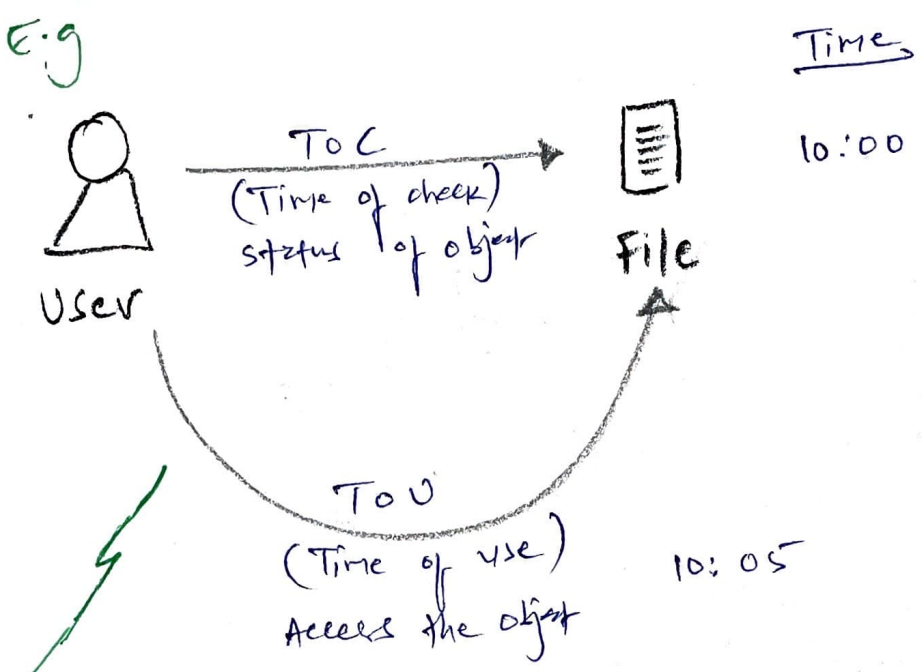
Get high-security level = ↓ Consequence

CIA compromise

Install latest version of software & be aware about existing vulnerabilities

↳ Timing, State changes, and Communication Disconnects

Attackers attack based on predictability of task execution (repetitive task)



Attack based on Timing

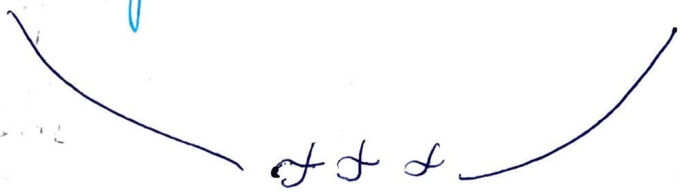
TOCTTOU (Race conditions) — In this a minute of time difference, attacker can replace normal file with malicious code.

Communication Disconnects — Small window of opportunity when attacker exploit

System state — Attacker attempt to trace action b/w two known states when state of resource or entire system changes.

State Attacks

↳ New Technology and Process Integration



Intertwine for new business function

=

Security problems

⇓

FOCUS

- Single point of failure
- Emerging weakness in Service oriented Architecture (SOA)

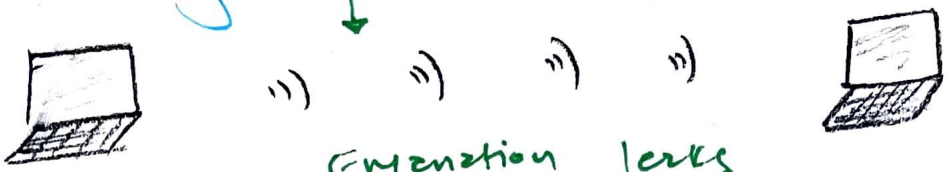
SOA is BAD ~~bad stuff~~

- SOA constructs new APP | functions out of existing ~~soa~~ but seperated software services.

New deployment / function = must be Securely evaluated & tested

- New APP
  - insecure
  - untested
  - unprotected

Electromagnetic Radiation.

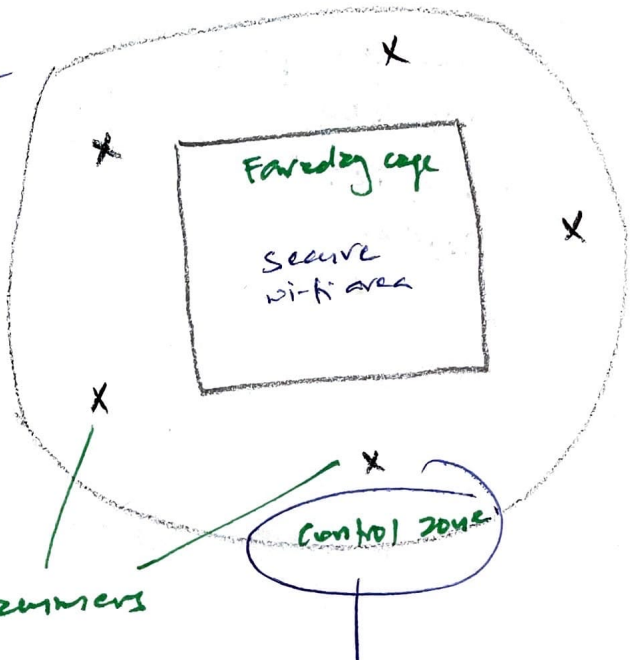


Emanation leaks

→ Intercept to recreate keyboard input, monitor output

Sol.

- Jamming
  - Noise Generators
- Faraday cage
  - Control Zone



Jammers

Wireless device works within control zone & blocks encryption outside

finally.

Abstraction = Black Box = Object

- Fundamental principle behind object-oriented programming
- It is Black Box Doctrine
- Core concept that uses Black-Box Approach: Users of the object don't need to know how object works or how object is implemented.

Database security Issues