Intro

Why Security Matters

Ubiquitous Computing

- Personal devices
- Embedded Systems
- Smart/IoT devices.

> Computers are everywhere...

Always Online

We are always connected to a form of network, be it wifi, ethernet connections, or mobile connections... Which means that location information is almost always present.

> You can already imagine how bad it is to be traceable at all times...

Online Services

Or using the cloud, basically any form of data storage and such...

> Our data isn't only for us, it is also stored somewhere online...

High Level Complexity

Using high level programming is really helpful for production, however, this complexity of frameworks poses a threat to how much we know on security.

> Who is to say that this framework is completely safe...

Human Error

Unaware employees pose high risks, even to the most secured system. Humans are highly manipulate-able, can be black-mailed and such.

Consequences of Attacks

Financial Loss: Be it from the attack itself, or from after-taxes due to endangering consumer data. Recovery also requires huge sums of money to get a business back online.

Productivity Loss: Loss in productivity of the business for variable reasons and disrupting the business.

Reputation Damage: In the business world, reputation is one of the most important asset to a company, losing it's reputation to an attack means that the business will go down and suffer huge losses.

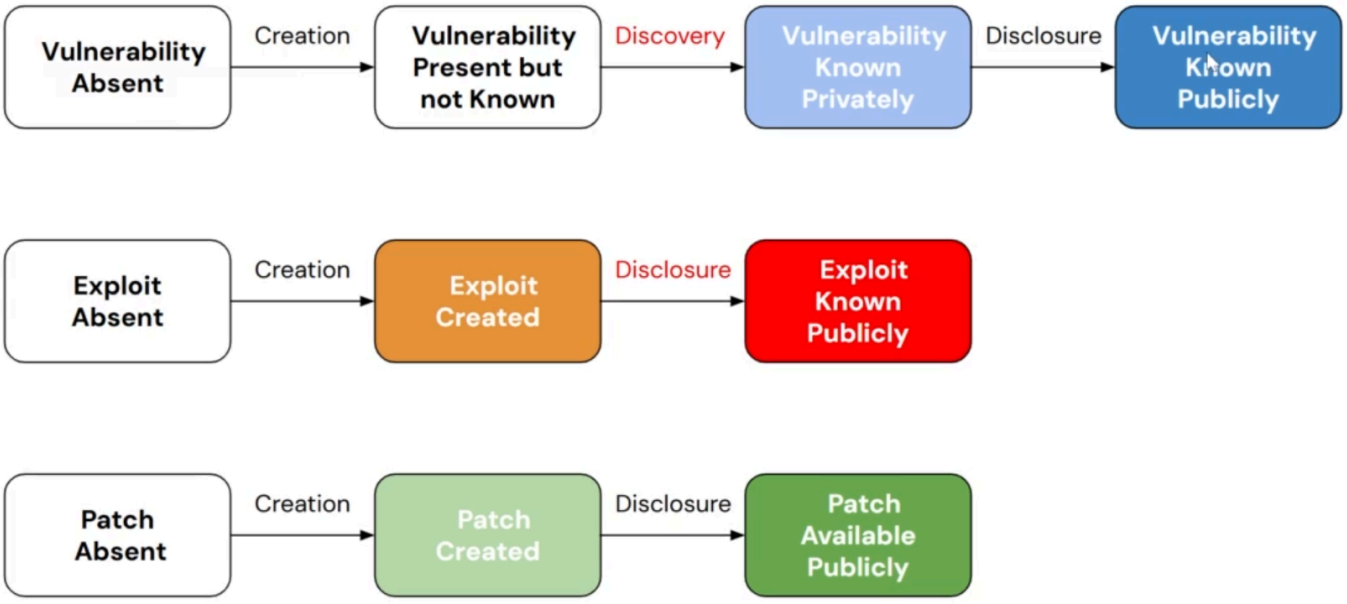
Example Attack

WannaCry Attack.

Basic Terminology

Threat: anything that potentially be harmful.
Vulnerability: weakness that threats can exploit.

Vulnerability Lifecycle



> Depending on who finds the vulnerability and what they might do with it, the lifecycle of the vulnerability might take different directions.

Common Vulnerabilities and Exposures (CVE)

A public database of known security flaws.

Hackers

Black Hat

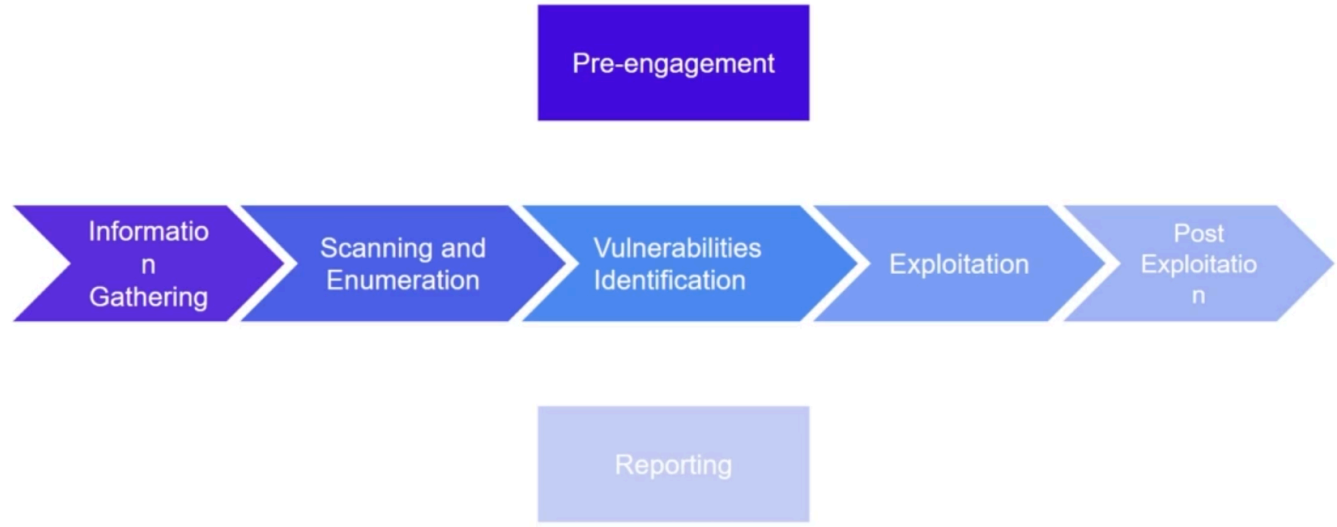
Criminal hackers, violate the law, use the exploits to get benefits.

White Hat

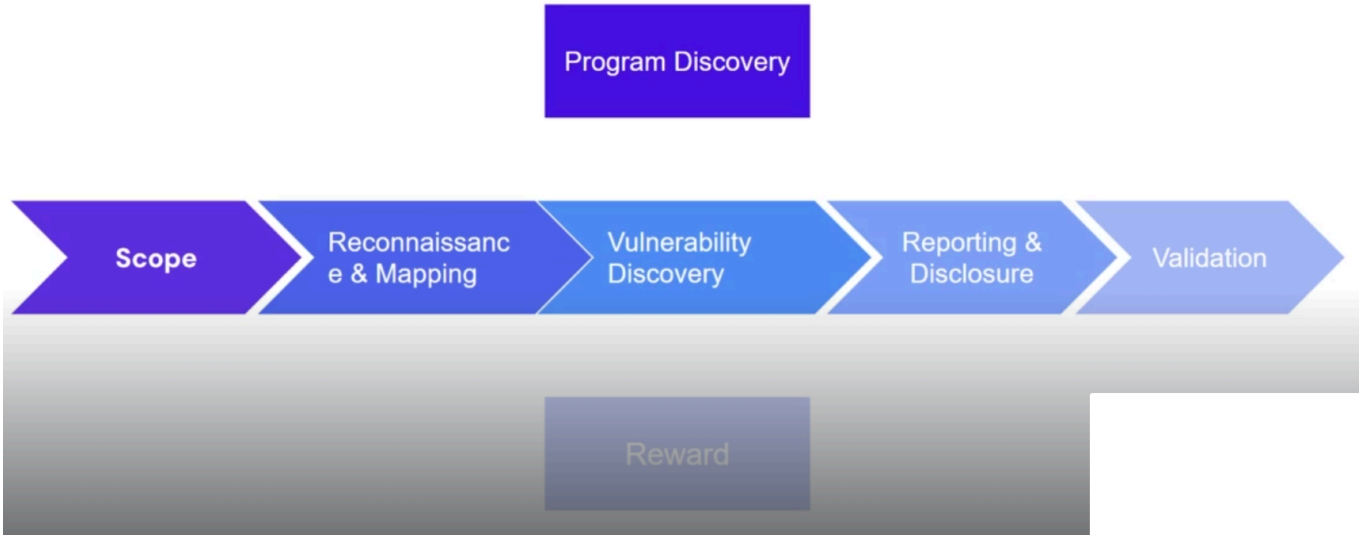
Work with the companies themselves, whenever they find any vulnerability, they would report it to be patched.

Pentesting

Phases of Penetration Testing



Bug Hunting



Pentesting vs Bug Hunting

Penetration Testing	Bug Hunting
Performed during a specific time period.	Typically run continuously.
Follow a systematic and structured approach.	Decentralized and crowd-sourced approach.
Paid per project.	Per valid report.

Red and Blue

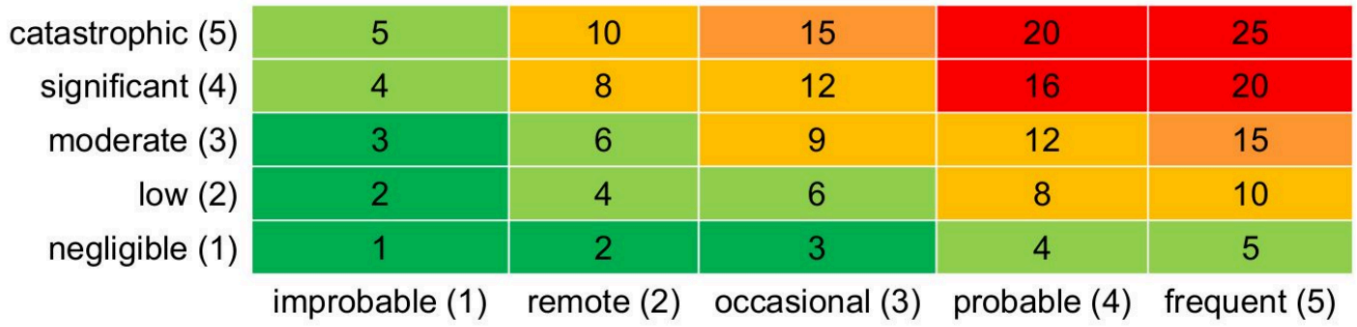
Red team, attacking side. They are a team selected by the company, their job is to find a way to hack into the organization, and report their findings to make the organization more secure.

Blue team, defending side. Their job is to defend the organization and block/patch any exploit. They are tasked with designing the systems of the organization in a secured way, or securing an already made system. They are also supposed to handle attacks properly.

Purple team, a middle ground, in between both teams. They are supposed to do both jobs.

Risk Estimation and Management

Risk Matrix



CIA Triad

Confidentiality

Secrecy in data, ensuring that the data is only accessed by authorized people.

Integrity

Protecting the data from unauthorized changes.

Availability

Ensuring that authorized people can always access their data and services without delay or interruptions.

Security By Design

Least Privilege

Always grant the least amount of access, for the shortest duration possible.

Duress Code

Design some kind of a duress code that is silently triggered in emergencies.

Defense In Depth

Data Layer

Most basic layer, which should always be implemented.
encryption, backups, access controls...

Application Layer

Content filtering, data validation...

Host Layer

Auth, ips/ids, passwords, hashing...

Network

ids, ips, logging...

Perimeter

dmz, vpn, logging, firewall, proxy...

Zero Trust

No Trust to anything, not a user, not a device, and not a network.

Least Privilege should be properly applied, never give anyone access to your data.

Log and Inspect any and all network traffic, always monitor what is going on in your network.

Always **verify** and **validate** all requests **continuously**.

Authentication

Single

Only the client proves their identity.
While more secured from the servers side, the client won't be safe this way.

Mutual Authentication

Both the client and the server prove their identities to each other.
Making it safer for both sides.

Data Origin Authentication

Ensures that the recipient can verify the origin of the data, making sure it is not forged.

Authorization

Determines whether a user has permissions to access data/services.

Ensuring that a person cannot deny performing an action.

> through different means like signatures, ip addresses, logs, timestamps and such

---

Classes of Attacks

Overview

Basic Problems

- Network Insecurity
- Weak Authentication
- Services Full of Bugs

Replay

An attack where a valid message is intercepted and sent again.

Counter Measures

Counter: Implementing a counter that is sent with the message to confirm that the request is sent correctly.
Weakness: After a couple of messages the attacker will understand the sequence and use it.

Timestamp + Lifetime: Sending the time of the request with it and making it only effective for short period of time.

IP Spoofing

Forging the source network address.

Counter Measures

NEVER USE ADDRESS BASED AUTHN

Packet Sniffing

Read the packets addressed to another node.

Counter Measures

Encryption of the packet payload.

DoS

Keep a host busy so it can't provide services.

Be it from buffer overflow, syn attack, ping flooding, mail/log saturation...

Counter Measures

Monitor requests and oversize servers.

DDoS

Distributed DoS.

Over the network.

Counter Measures

Monitor, human confirmation...

Shadow/Fake Server

Could be through:

- Intercepting and then spoofing, impersonating the server.
- DNS.

Counter Measures

Server authentication.