Cyber Security

PLURALSIGHT

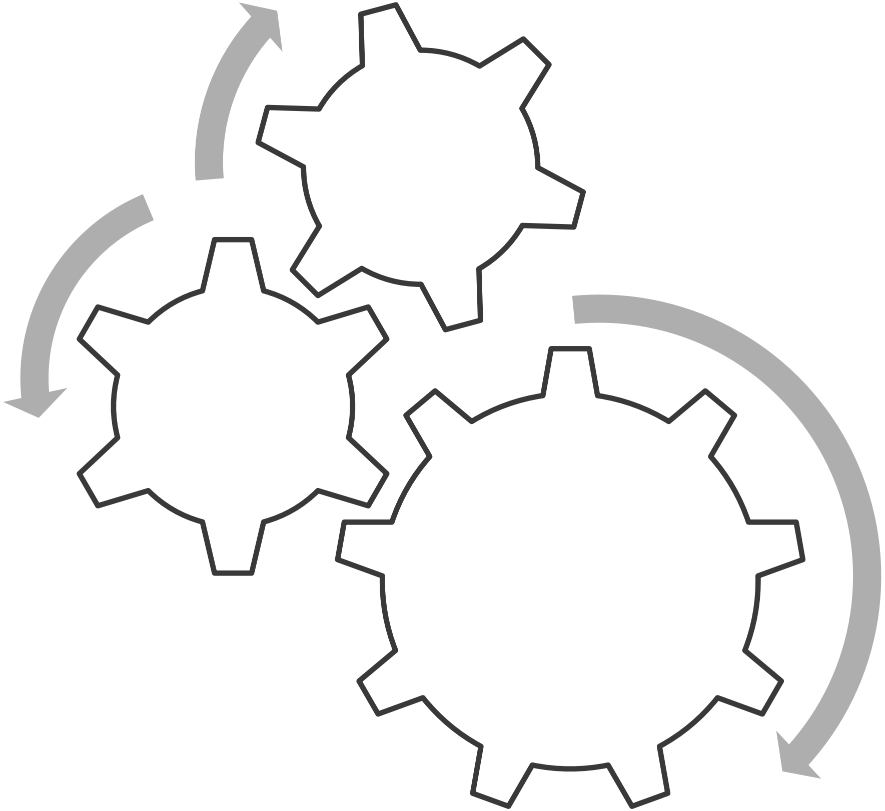
AGENDA

- Introduction to Software Security
- Security vs. Privacy Implications in Software Design
- Common Threats and Mitigations
- Identity and Access Management
- Security as Software Requirement
- Threat Modelling
- Vulnerability Assessment vs. Penetration Testing
- Code Review & Penetration Testing
- Aspects of Good Code Review
- OWASP ASVP
- Privacy by Design and by Default

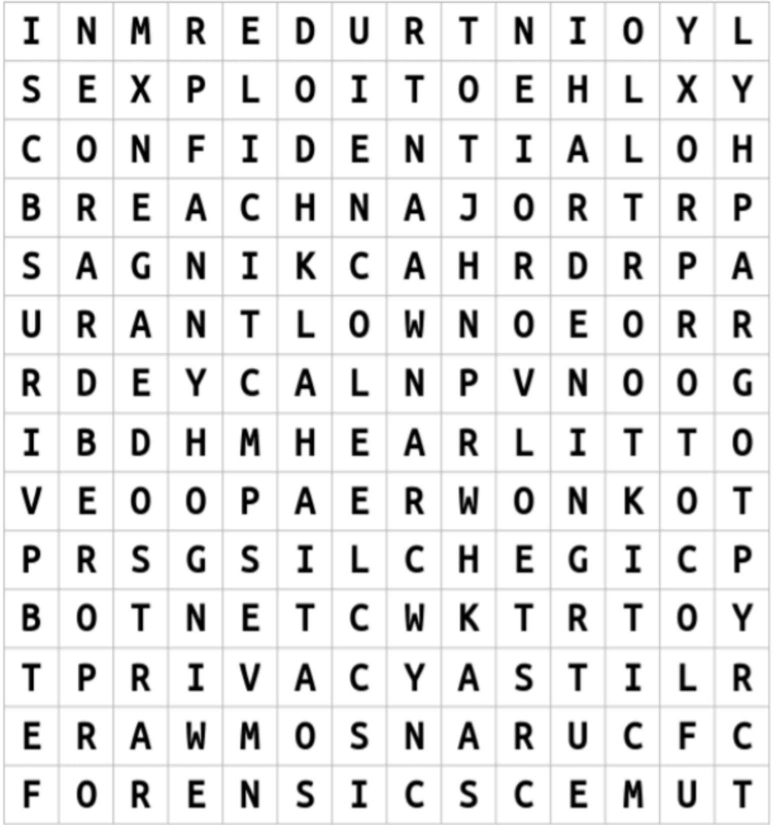
Introduction to Software Security

PLURALSIGHT

How we're going to work together

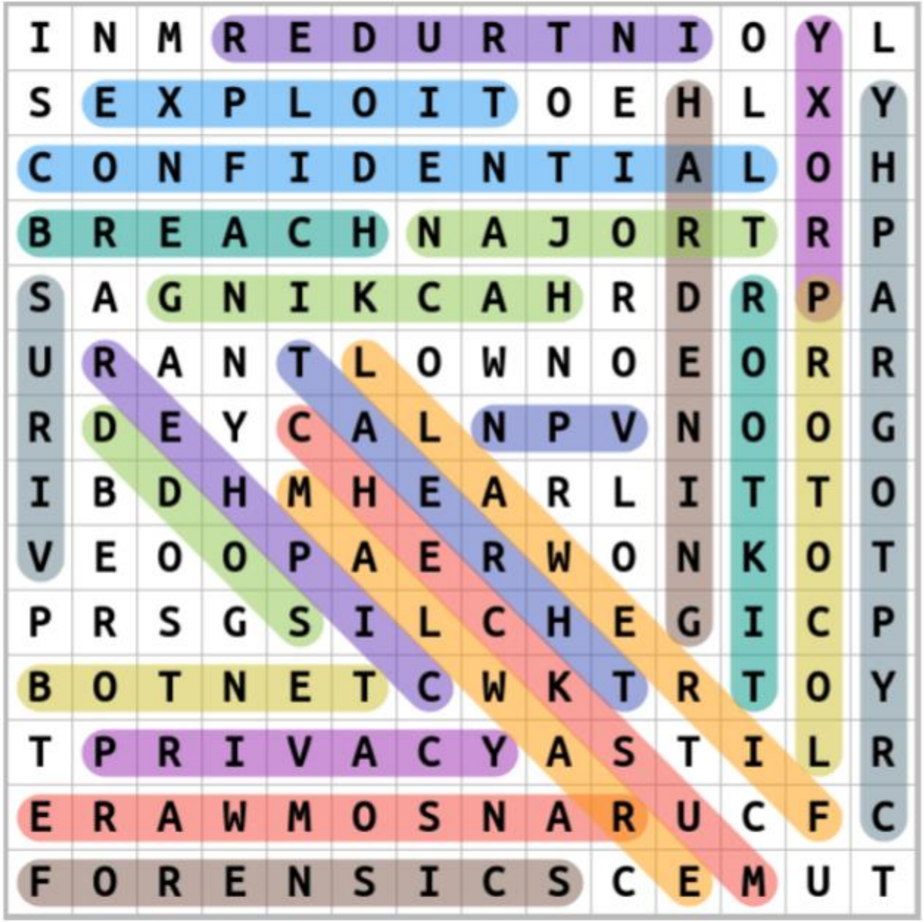


Let's try this Word Search Puzzle



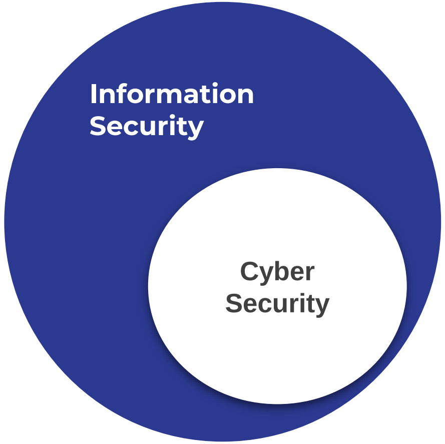
I	N	M	R	E	D	U	R	T	N	I	O	Y	L
S	E	X	P	L	O	I	T	O	E	H	L	X	Y
C	O	N	F	I	D	E	N	T	I	A	L	O	H
B	R	E	A	C	H	N	A	J	O	R	T	R	P
S	A	G	N	I	K	C	A	H	R	D	R	P	A
U	R	A	N	T	L	O	W	N	O	E	O	R	R
R	D	E	Y	C	A	L	N	P	V	N	O	O	G
I	B	D	H	M	H	E	A	R	L	I	T	T	O
V	E	O	O	P	A	E	R	W	O	N	K	O	T
P	R	S	G	S	I	L	C	H	E	G	I	C	P
B	O	T	N	E	T	C	W	K	T	R	T	O	Y
T	P	R	I	V	A	C	Y	A	S	T	I	L	R
E	R	A	W	M	O	S	N	A	R	U	C	F	C
F	O	R	E	N	S	I	C	S	C	E	M	U	T

Solution



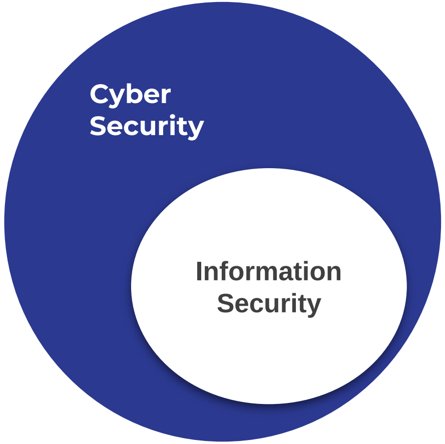
I	N	M	R	E	D	U	R	T	N	I	O	Y	L
S	E	X	P	L	O	I	T	O	E	H	L	X	Y
C	O	N	F	I	D	E	N	T	I	A	L	O	H
B	R	E	A	C	H	N	A	J	O	R	T	R	P
S	A	G	N	I	K	C	A	H	R	D	R	P	A
U	R	A	N	T	L	O	W	N	O	E	O	R	R
R	D	E	Y	C	A	L	N	P	V	N	O	O	G
I	B	D	H	M	H	E	A	R	L	I	T	T	O
V	E	O	O	P	A	E	R	W	O	N	K	O	T
P	R	S	G	S	I	L	C	H	E	G	I	C	P
B	O	T	N	E	T	C	W	K	T	R	T	O	Y
T	P	R	I	V	A	C	Y	A	S	T	I	L	R
E	R	A	W	M	O	S	N	A	R	U	C	F	C
F	O	R	E	N	S	I	C	S	C	E	M	U	T

Information Security vs. Cyber Security



Information Security

Cyber Security



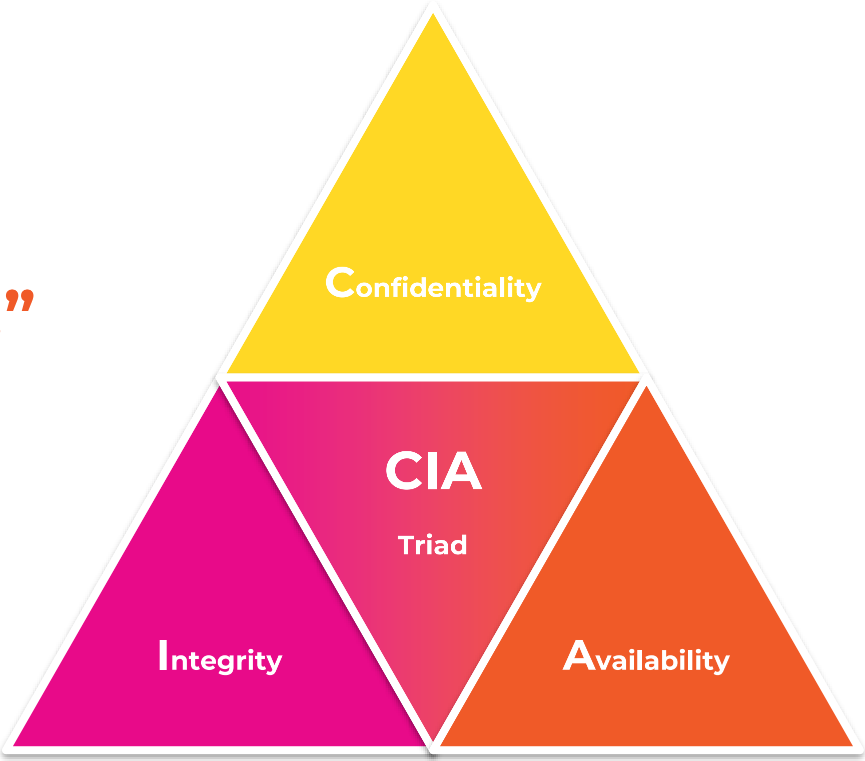
Cyber Security

Information Security

It’s all about the CIA

Before moving ahead, always remember that

“It’s always about CIA!”



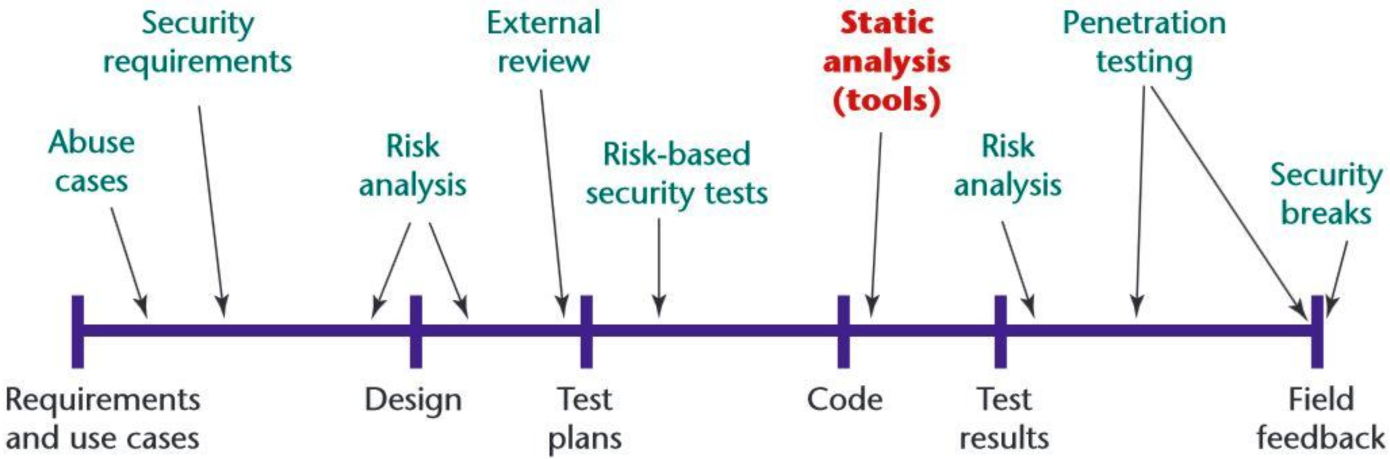
Information Security Standards / Frameworks

- **ISO 27001**, ISO 27002, ISO 27003, ISO 27004, ISO 27005 ...
- **NIST** Cyber Security Framework
- **PCI DSS**
- **HIPAA**
- **GDPR**
- **HITRUST CSF**
- **IT Act**
- **CIS**
- **and many more ...**

Effective Software Security

- Software security is the idea of engineering software so that it continues to function correctly under malicious attack.
- Most technologists acknowledge its importance, but need help in understanding how to tackle it.
- Here, we will explore software security best practices.
- It is a relatively new field, with first books and courses appearing in 2001, demonstrating how recently developers, architects and computer scientists have started systematically studying how to build secure software.
- This recency is one of the reasons why best-practices are neither obvious nor widely adopted.

Effective Software Security



Challenges in Achieving Effective Software Security

- Speed of Software Development
- Risks of Using Open Source Components
- Vulnerabilities in Code
- Lack of AppSec Planning
- Security Testing during a Secure Software Development Life cycle

I'M AFRAID WE HAD TO CANCEL THE 'PERFECT PLANNING' SEMINAR. WE FORGOT TO BOOK THE HOTEL UNTIL IT WAS TO LATE AND THE SPEAKER WE'D HOPED TO USE DIED IN 1958
FRAN
©Original Artist
Reproduction rights obtainable from
www.CartoonStock.com

Importance of Getting Software Security Right

- Software security and security testing are often overlooked aspects of software development
- Many businesses don’t realize the importance of it until it’s too late.

1

Software is now a critical part of most businesses

The majority of the companies heavily rely on some kind of software, and any vulnerability in the code can have serious consequences. Some recent large-scale breaches include the 2021 attack on Facebook, which resulted in exposing the personal information of 1.5 billion users.

2

The number and severity of attacks are increasing

Hacker s are increasingly targeting vulnerabilities as a way to gain access to networks and data. Once the software is compromised, it can take years for a business to recover from damage and losses.

3

The stakes are higher than ever

The competition within the software industry has seen an explosion in recent years with each business fiercely trying to distinguish themselves with their product or service offering. A vulnerability in software is not only harmful for the users but also for company’s reputation & image.

Security vs. Privacy: Implications in Software Design

What is Data Privacy?

- Data privacy refers to the proper use and processing of personal data by restoring control over their data to individuals.
- Simply put, data privacy enables individuals to decide and limit access to the use and sharing of their personal data.
- Protecting personal information ensures that the data is kept secure.
- Here, data privacy transitions into the realm of data security and protection.

Data Privacy Laws

- If you’re a business owner with an online presence, you have probably heard about recently enacted numerous data privacy laws in different geographies.
- Here are some of the major ones:
  - EU: General Data Protection Regulation (GDPR)
  - Canada: Personal Information Protection and Electronic Documents Act (PIPEDA)
  - CA, USA: The California Online Privacy Protection Act (CalOPPA)
  - CA, USA: The California Consumer Privacy Act (CCPA)
  - UT, USA: The Utah Consumer Privacy Act (UCPA)

How Data Security affects Data Privacy

- Most online businesses and websites collect personal data, from email addresses to phone numbers, credit cards, and log-in details.
- Ideally, these entities shouldn’t keep more information than is necessary, nor should they keep it longer than necessary.
- However, you cannot operationalize data privacy without ensuring the security of data.
- For example, if you fail to protect people’s credit card details against hackers and they get access to this data, they can sell it on the dark web.
- Therefore, data security is a prerequisite to data privacy

What is Data Security?

- Data security is the concept of protecting digital data from theft, corruption, or unauthorized access throughout its entire lifecycle of:
    - Creation
    - Storage
    - Use
    - Sharing
    - Archiving
    - Destruction
- Data security involves physical security of the storage devices and hardware, administrative access controls, and the security of software applications.
- It also includes organizational policies and procedure

Data Security vs. Data Privacy

Data Privacy ≠ Data Security

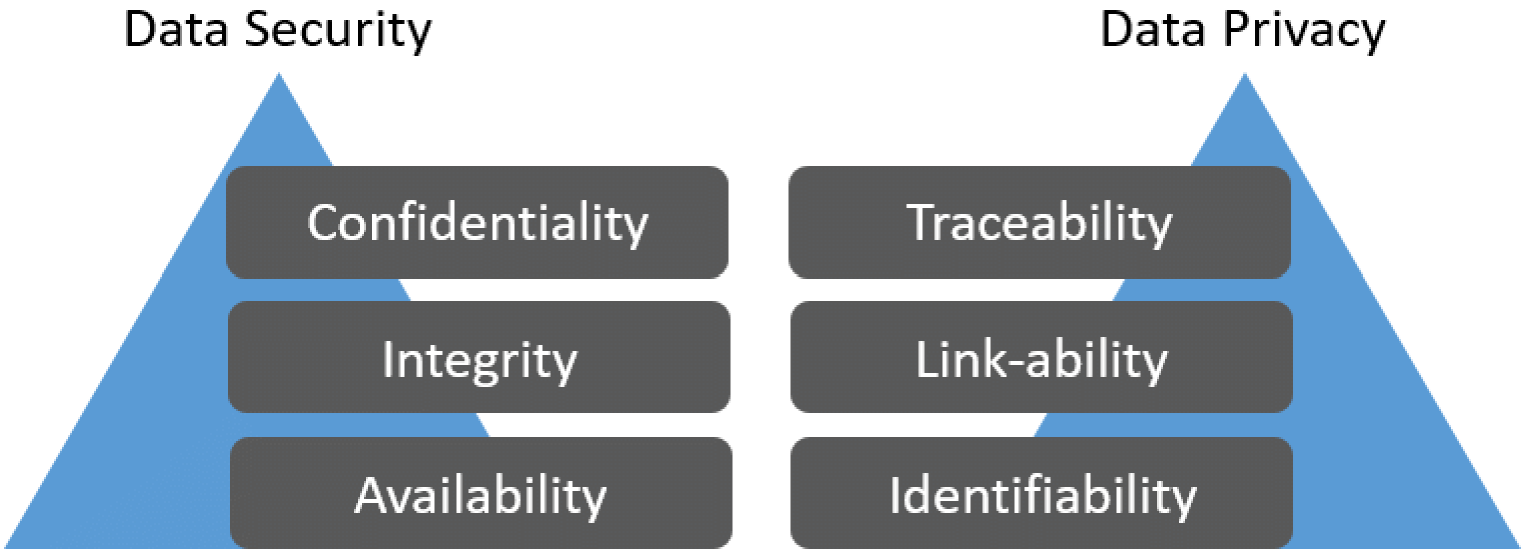
Data Privacy

Compliance with data protection laws and regulations. Focus on how to collect, process, share, archive and delete the data

Data Security

Measures that an organzation is taking in order to prevent any third party from unauthorized access.

Data Security vs. Data Privacy



Data Security

Confidentiality

Integrity

Availability

Data Privacy

Traceability

Link-ability

Identifiability

Example: Data Privacy

Example.com sells unique products via its eCommerce shop and it collects many pieces of data from its online shoppers such as:

- Email addresses and log-in details
- Shipping addresses
- Billing addresses

To ensure proper handling of personal data and to give individuals control over access to and sharing of their data, Example.com does the following:

- It allows its customers to unsubscribe from its email marketing & newsletter lists.
- It does not disclose its customers’ email addresses and purchases data to data brokers without getting its customers’ consent.
- It stores customers’ purchase information in accordance with data storage periods determined by applicable laws.

These efforts are all part of Example.com’s data privacy strategy.

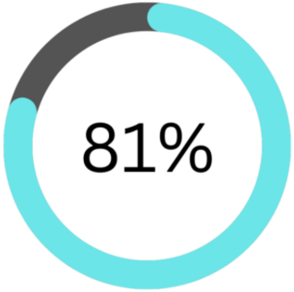
Example: Data Security

- The executives recently decided to update Example.com’s data security policy.
- As a result, they hired a data security analyst who brought to their attention that more staff members had access to shoppers’ information than was necessary — weakening the company’s overall data security.
- After reviewing which staff members needed access to this information, they reduced the number of “*need-to-know*” players from 26 to only seven.
- Additionally, they allowed an outlet for some other members to request access under special circumstances, for a limited period or a specific task only.
- By reducing the number of staff members who could access shoppers’ data by nearly three-quarters, Example.com significantly strengthened its data security plan.

Common Threats and Mitigations

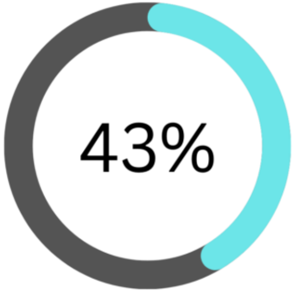
Identity Attacks

Stolen or Weak Password



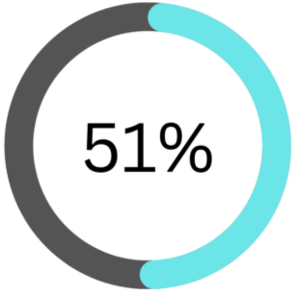
of breaches leveraged either stolen and/or weak passwords

Social Attacks



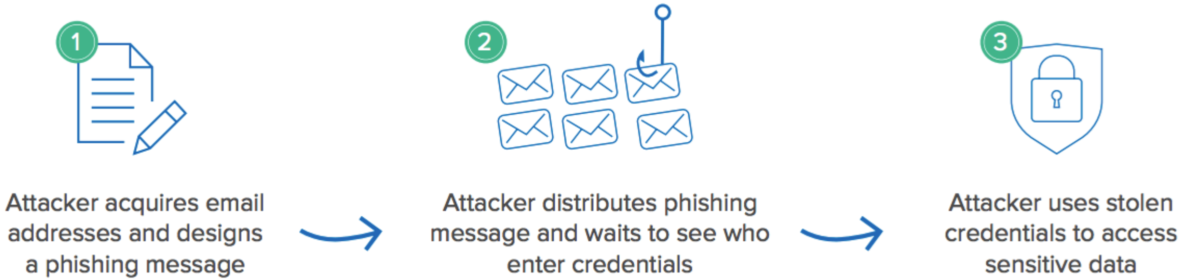
attacks resulted in a data breach

Credential Stealing Software



of data breaches include some form of credential stealing software

Broad-based Phishing Campaign Attack

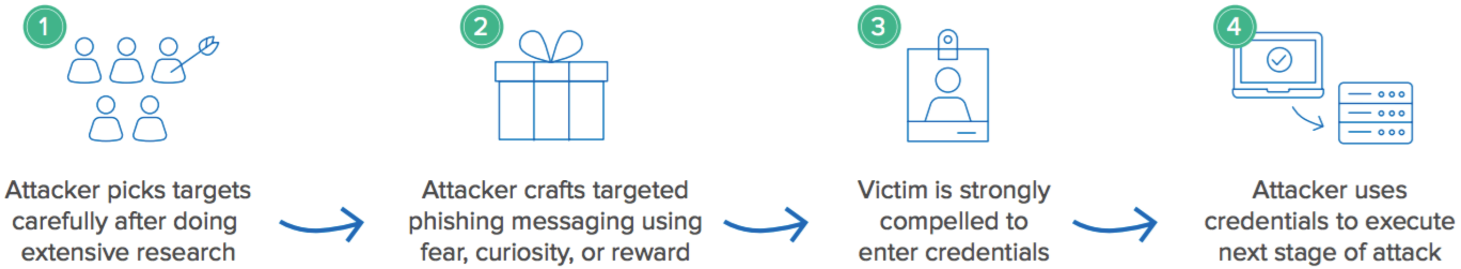


1 Attacker acquires email addresses and designs a phishing message

2 Attacker distributes phishing message and waits to see who enter credentials

3 Attacker uses stolen credentials to access sensitive data

Spear Phishing Campaign Attack



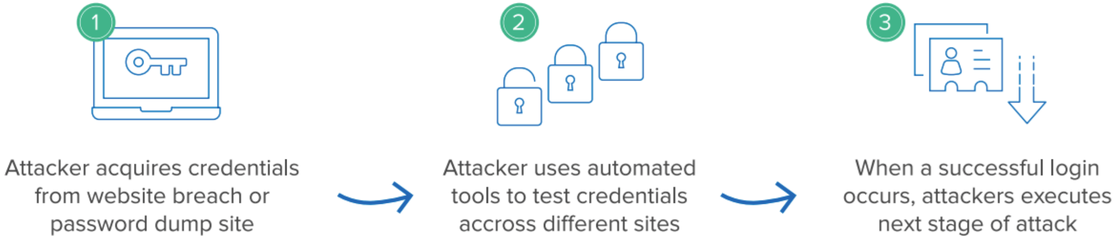
1 Attacker picks targets carefully after doing extensive research

2 Attacker crafts targeted phishing messaging using fear, curiosity, or reward

3 Victim is strongly compelled to enter credentials

4 Attacker uses credentials to execute next stage of attack

Credential Stuffing Attack

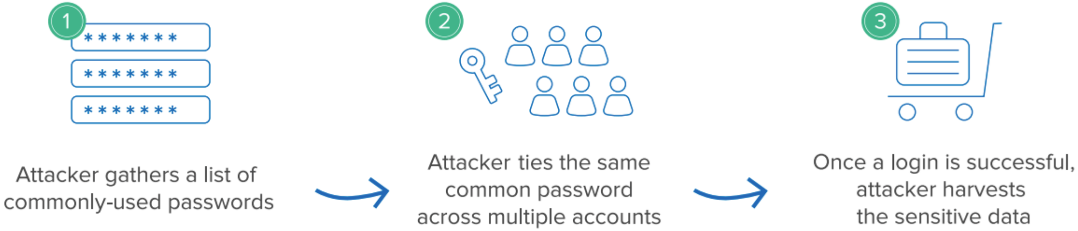


1 Attacker acquires credentials from website breach or password dump site

2 Attacker uses automated tools to test credentials accross different sites

3 When a successful login occurs, attackers executes next stage of attack

Password spraying attack

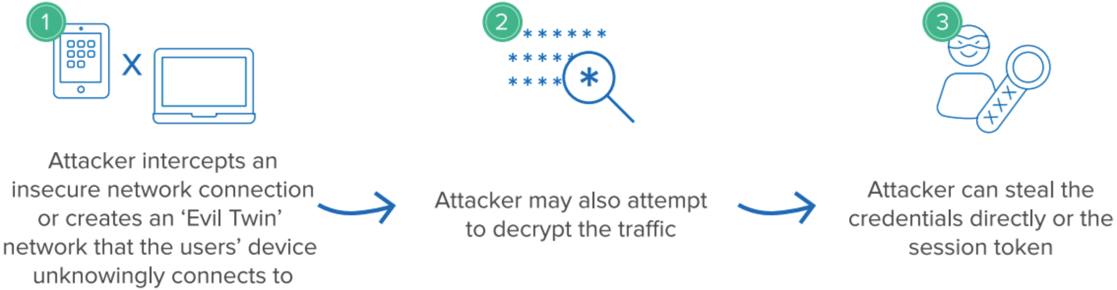


1. Attacker gathers a list of commonly-used passwords

2. Attacker ties the same common password across multiple accounts

3. Once a login is successful, attacker harvests the sensitive data

Man-in-the Middle attack



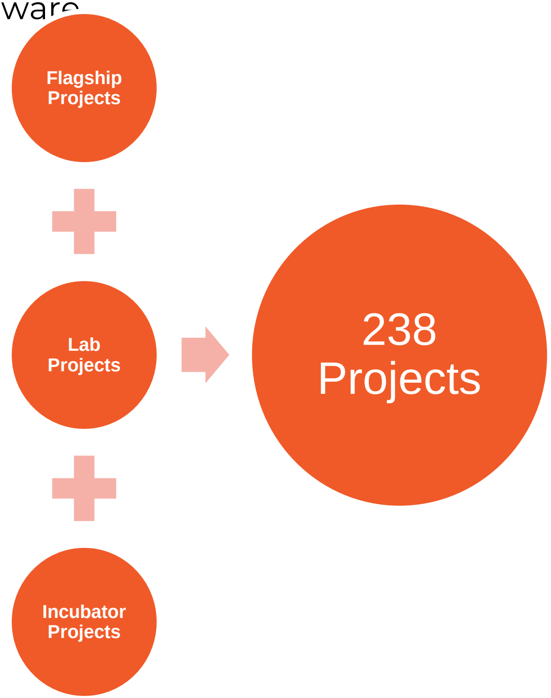
Attacker intercepts an insecure network connection or creates an ‘Evil Twin’ network that the users’ device unknowingly connects to

Attacker may also attempt to decrypt the traffic

Attacker can steal the credentials directly or the session token

What is OWASP ?

- The Open Web Application Security Project® (OWASP) is a nonprofit foundation that works to improve the security of software



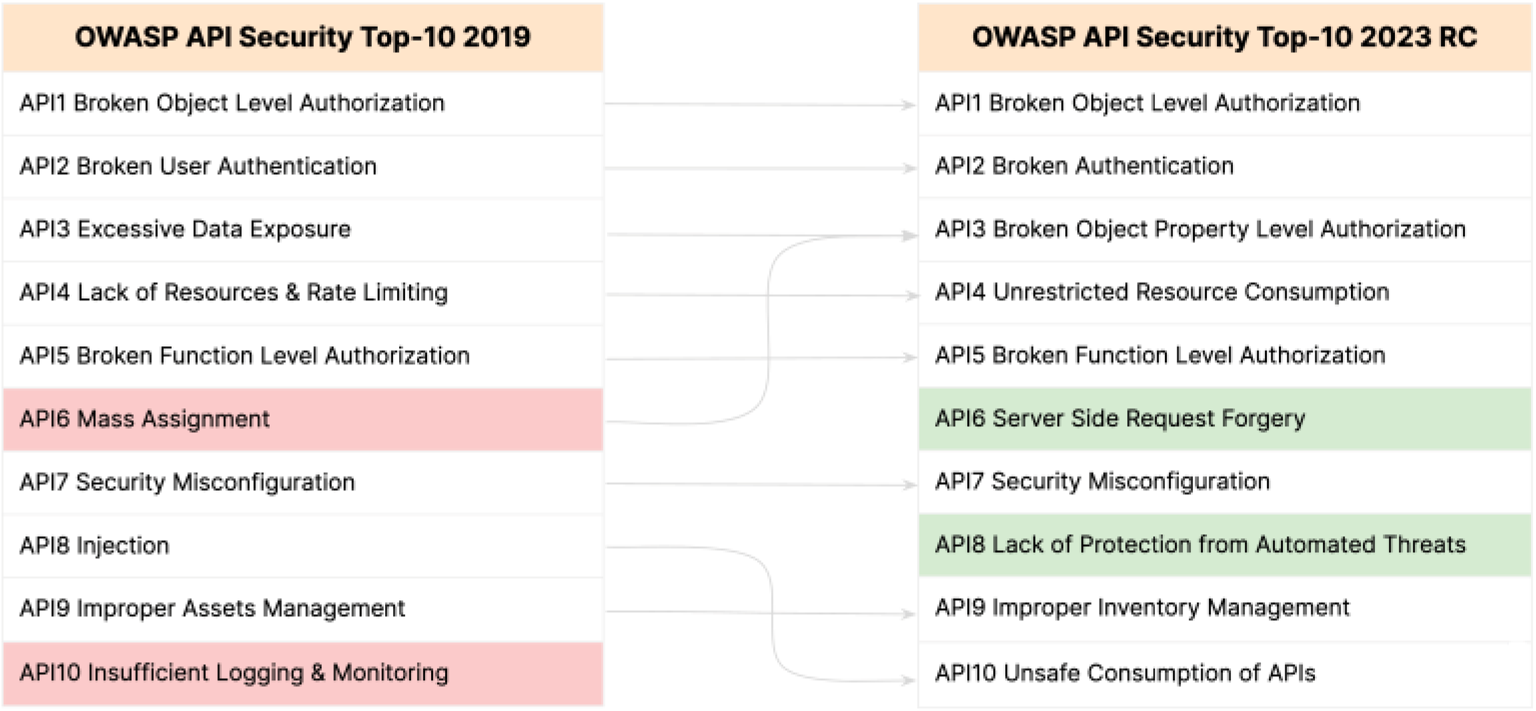
Flagship Projects

Lab Projects

Incubator Projects

238 Projects

OWASP Top 10



OWASP API Security Top-10 2019
API1 Broken Object Level Authorization
API2 Broken User Authentication
API3 Excessive Data Exposure
API4 Lack of Resources & Rate Limiting
API5 Broken Function Level Authorization
API6 Mass Assignment
API7 Security Misconfiguration
API8 Injection
API9 Improper Assets Management
API10 Insufficient Logging & Monitoring

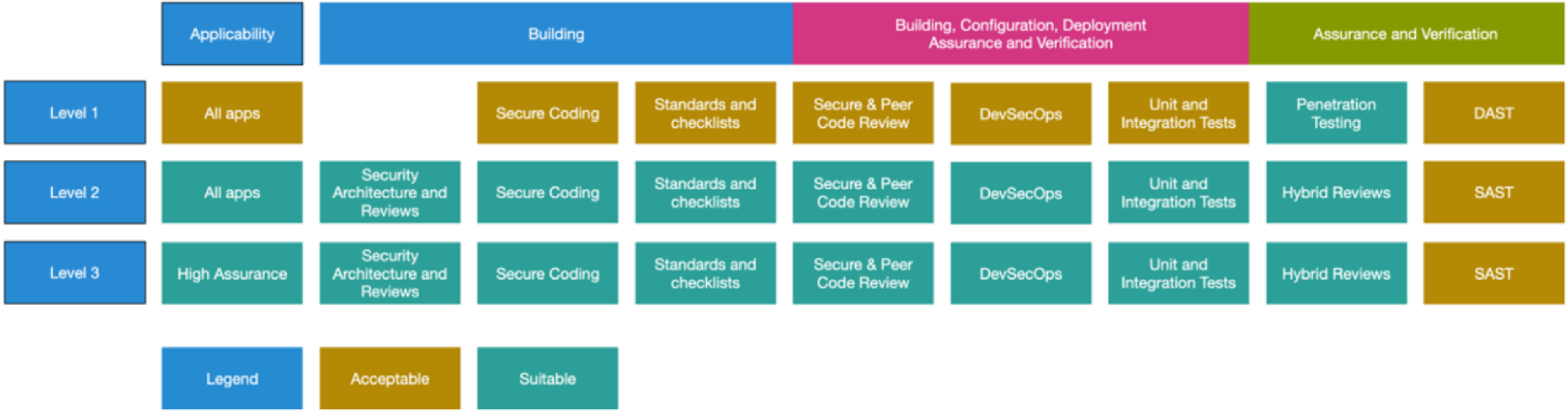
OWASP API Security Top-10 2023 RC
API1 Broken Object Level Authorization
API2 Broken Authentication
API3 Broken Object Property Level Authorization
API4 Unrestricted Resource Consumption
API5 Broken Function Level Authorization
API6 Server Side Request Forgery
API7 Security Misconfiguration
API8 Lack of Protection from Automated Threats
API9 Improper Inventory Management
API10 Unsafe Consumption of APIs

Image Source: https://securityboulevard.com/2023/03/insights-into-the-new-owasp-api-security-top-10-for-cisos/

Application Security Verification Standard (ASVS)

- ASVS has two main goals:
  - to help organizations develop and maintain secure applications; and
  - to allow security service vendors, security tools vendors, and consumers to align their requirements and offerings.

ASVS v4 – Checklist Walkthrough



	Applicability	Building			Building, Configuration, Deployment Assurance and Verification			Assurance and Verification	
Level 1	🟨 All apps		🟨 Secure Coding	🟨 Standards and checklists	🟨 Secure & Peer Code Review	🟨 DevSecOps	🟨 Unit and Integration Tests	🟩 Penetration Testing	🟨 DAST
Level 2	🟩 All apps	🟩 Security Architecture and Reviews	🟩 Secure Coding	🟩 Standards and checklists	🟩 Secure & Peer Code Review	🟩 DevSecOps	🟩 Unit and Integration Tests	🟩 Hybrid Reviews	🟨 SAST
Level 3	🟩 High Assurance	🟩 Security Architecture and Reviews	🟩 Secure Coding	🟩 Standards and checklists	🟩 Secure & Peer Code Review	🟩 DevSecOps	🟩 Unit and Integration Tests	🟩 Hybrid Reviews	🟨 SAST

Legend	🟨 Acceptable	🟩 Suitable

Image Source: https://owasp.org/

Identity and Access Management

IAM Overview

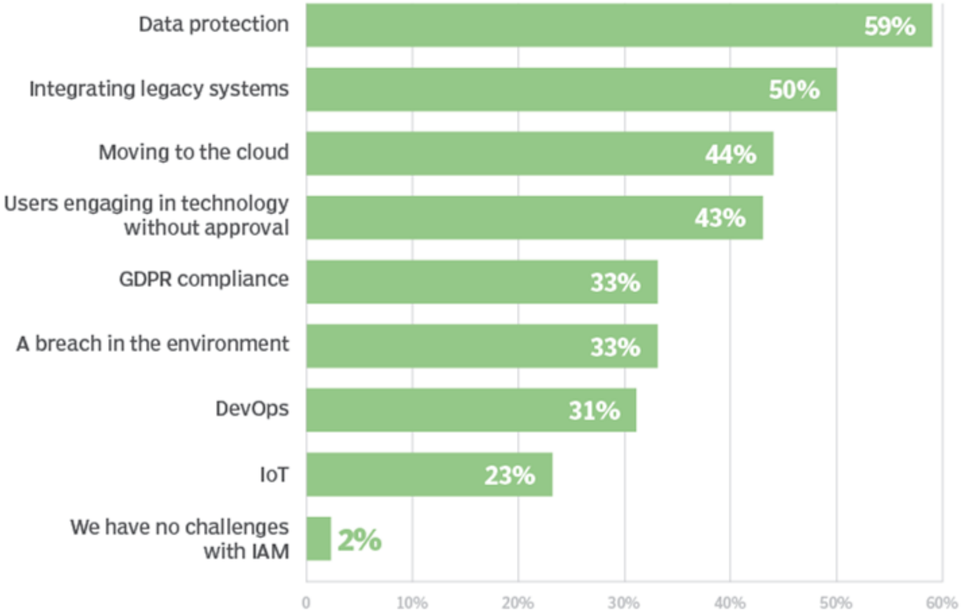
- Identity and access management (IAM) is a framework of business processes, policies and technologies that facilitates the management of electronic or digital identities.
- With an IAM framework in place, information technology (IT) managers can control user access to critical information within their organizations .
- Systems used for IAM include single sign-on systems, two-factor authentication, multifactor authentication and privileged access management.
- These technologies also provide the ability to securely store identity and profile data as well as data governance functions to ensure that only necessary and relevant data is shared.
- IAM systems can be deployed on on-prem systems, through a third-party cloud-based subscription model, or in a hybrid model.

IAM Overview

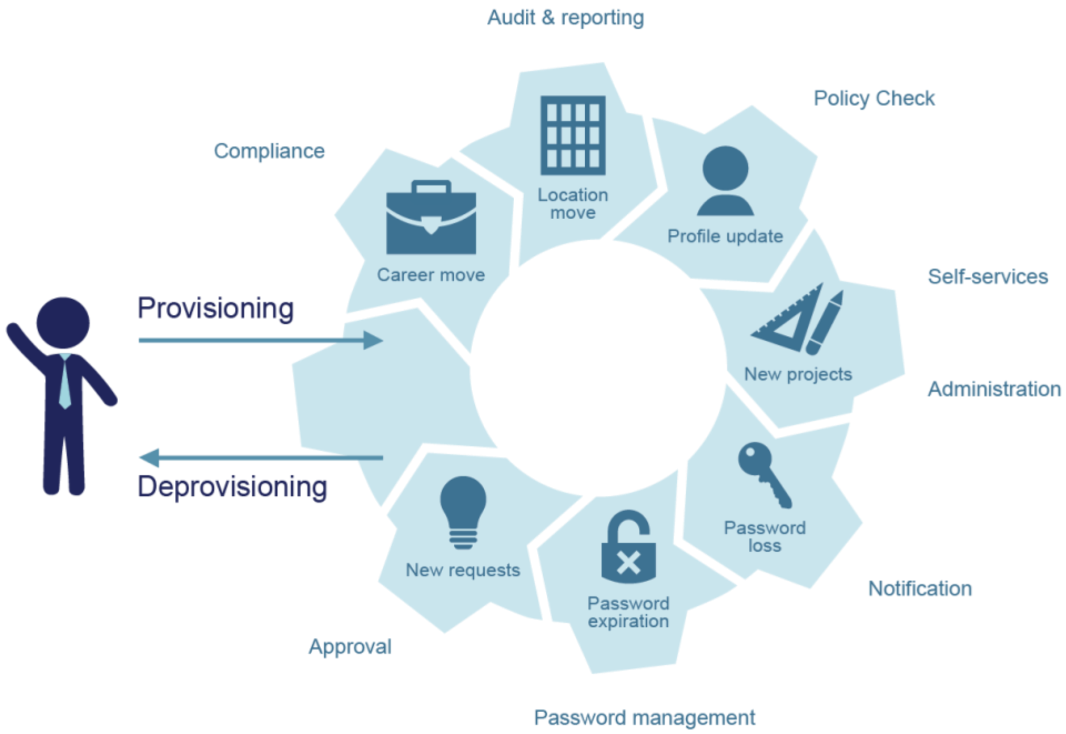
On a fundamental level, IAM encompasses the following components:

- how individuals are identified in a system (understand the difference between identity management and authentication);
- how roles are identified in a system and how they are assigned to individuals;
- adding, removing and updating individuals and their roles in a system;
- assigning levels of access to individuals or groups of individuals; and
- protecting the sensitive data within the system and securing the system itself.

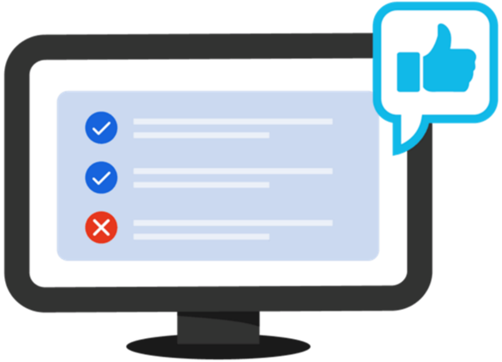
Breakdown of reported IAM Challenges



Identity and Access Life Cycle



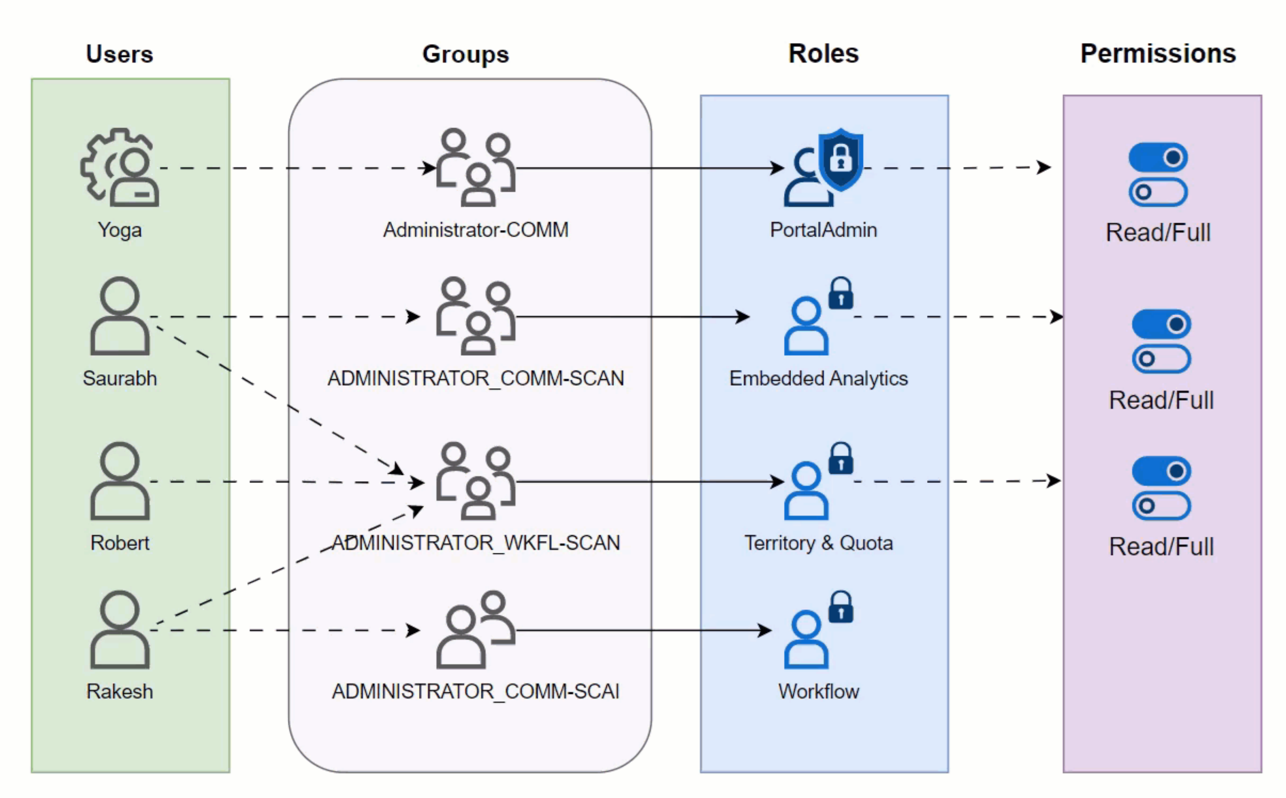
Authentication & Authorization

Authentication	Authorization
	
Who are you?	**Are you allowed to do this?**
Validate that system is accessed by the right person	Check if the user has required permissions to access the data

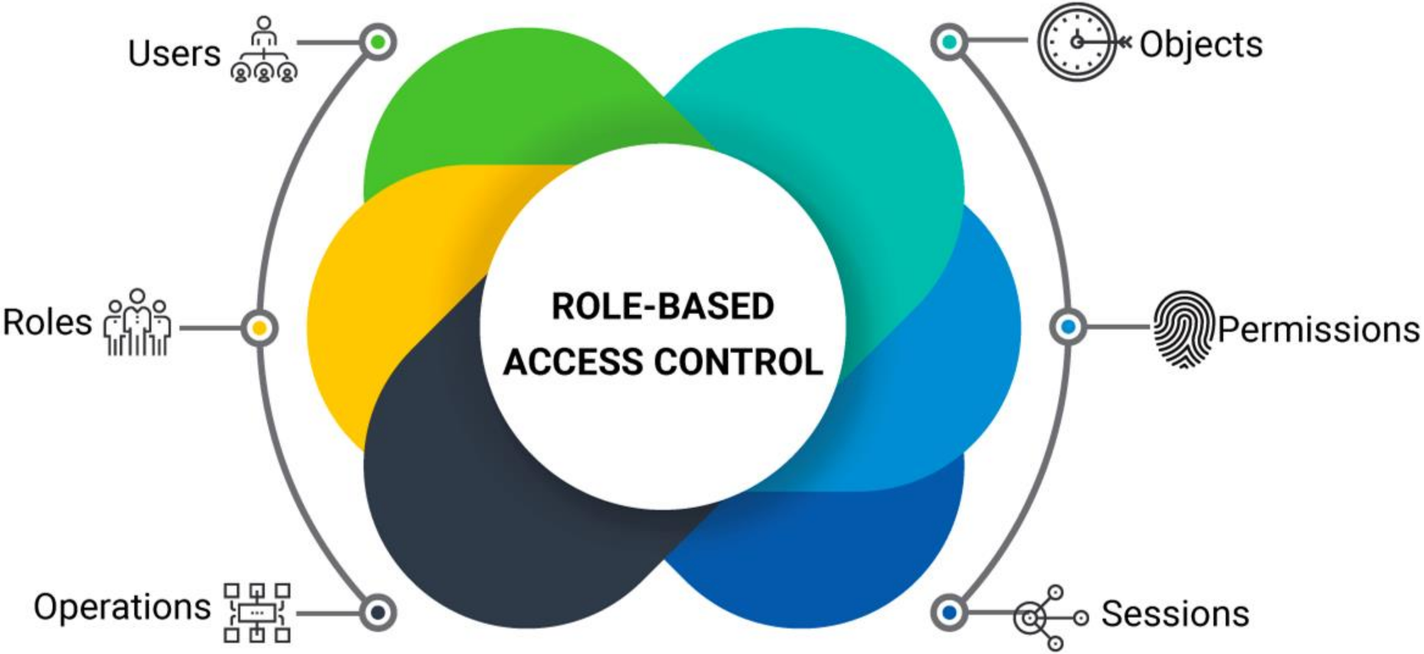
Role Based Access Control (RBAC)

- RBAC restricts network/system/application access based on a person's role within an organization and has become one of the main methods for advanced access control.
- The roles in RBAC refer to the levels of access that employees have to the network/systems/applications.
- Employees are only allowed to access the information necessary to effectively perform their job duties.
- Access can be based on several factors - authority, responsibility, and job competency.
- Access to computer resources can be limited to specific tasks - the ability to view, create, or modify a file.
- Employees at lower-levels usually have no or limited access to sensitive data to fulfill their responsibilities.
- It can also be extended to third-parties and contractors who have access to your network/systems/applications.
- Using RBAC will help in securing your company's sensitive data and important applications.

Role Based Access Control (RBAC)



Key components of RBAC



Single Sign-On (SSO)

- Single sign-on (SSO) is a one-to-many authentication method that allows users to access multiple resources using a single username/password combination.
- *Use case: Consider a multi-speciality medical hospital, where medical users (doctors, nursing staff, etc.) may need access to the electronic medical records systems, then toggle over to the laboratory software, pharmacy system, bed management system, and so on. If at each point they need to fill in credentials, then ir can become frustrating.*

  With SSO, they'll login only once a day into the system and all applications will share their common session.

Single Sign-On (SSO) Benefits

Single sign-on offers several benefits:

- **Productivity**: SSO allows users to access multiple resources efficiently and securely, reducing potential downtime lost to account lockouts or forgotten credentials.
- **User experience**: Accessing applications on multiple devices with one sign-on improves the working experience of your staff.
- **Compliance**: SSO provides a centralized, secure, and auditable authentication mechanism in compliance with applicable regulatory requirements, including HIPAA and Payment Card Industry Data Security Standard (PCI-DSS).
- **Security**: SSO has been shown to improve adoption of password complexity and multi-factor authentication among employees dealing with critical and sensitive data, resulting in improved security.
- **Monitoring**: With SSO, IT departments an easily monitor user activity and limit data access through robust account administration and auditing features.

Potential pitfalls of Single Sign-On (SSO)

Despite the many positive benefits of SSO, there are some notable pitfalls of SSO, including:

- **Access**: SSO introduces new layers of complexity and potential reliance on 3rd party platforms. This could impact access to applications should a logon portal become inaccessible or malfunction.
- **Interoperability**: Legacy applications that don't support SSO can pose challenges for IT teams, who may need to create exceptions and workarounds to ensure these applications continue functioning.
- **Threat detection:** Over-reliance on one single authentication mechanism can make it harder for security teams to detect suspicious behaviors as it can be difficult to distinguish between legitimate and unauthorized user activities.
- **Credential stuffing**: SSO can increase exposure to credential stuffing attacks, which is when hackers use compromised sessions or stolen credentials from one application to gain access to another application connected to the same SSO system.
- **New risks**: There are situations when SSO can introduce new risks, such as increasing the impact of compromised accounts.

Multi-Factor Authentication (MFA)

- MFA enhances security by requiring users to provide multiple forms of identification to access an application.
- With MFA, users typically provide two or more authentication factors, such as a password and a frequently changing one-time verification code.
- This approach significantly reduces the risk of unauthorized access to systems, even if a password is compromised.
- Security experts separate MFA factors into three main categories:

Something you know

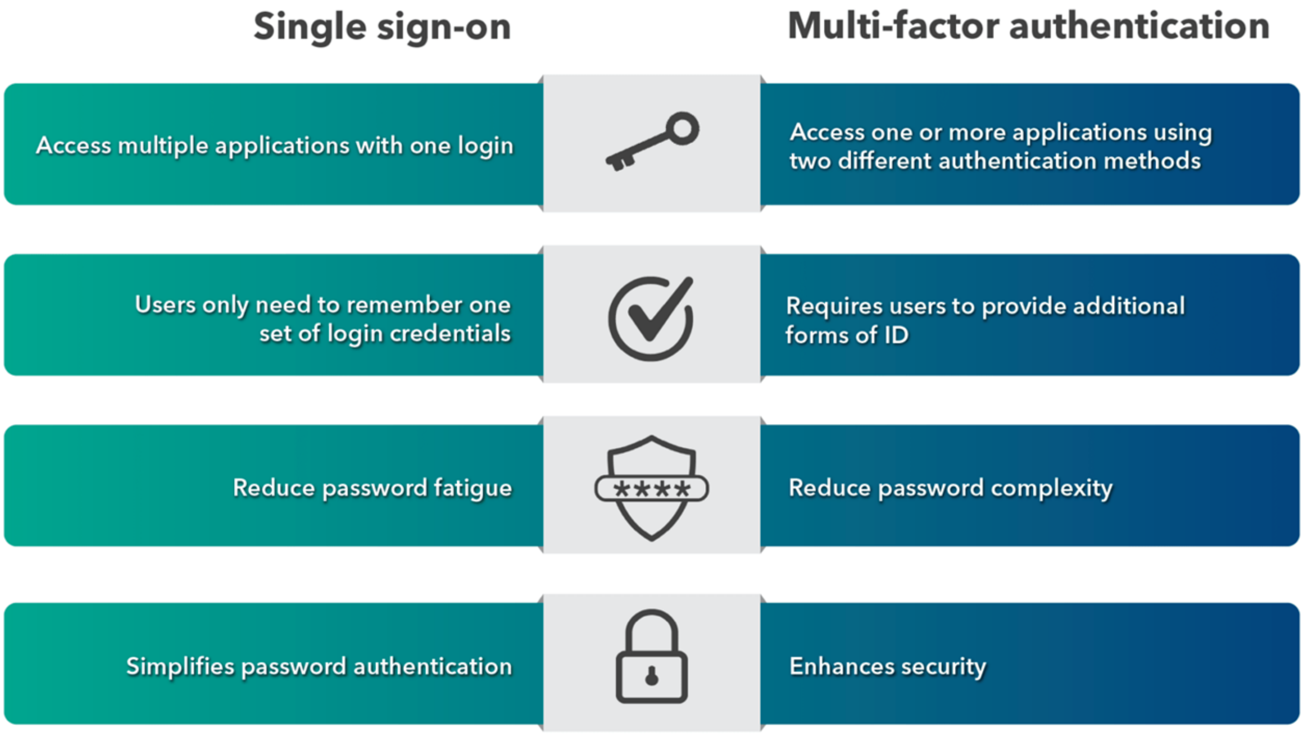
Something you have

Something you are

Implementing Multi Factor Authentication

- When implementing MFA, organizations choose two or more authentication factors, for example, a password and a verification code,
- IT team can require MFA with every login (a zero trust model), or only when users login from a new device or an unknown network.
- After verifying their identity, users can safely access the applications.
- To minimize the perceived inconvenience for stakeholders, it is crucial to choose authentication factors that are user-friendly.
- Alternatively, you could enhance the security measures with productivity and usability benefits, such as those provided by single sign-on (SSO).
- By incorporating MFA with SSO, we can add an extra layer of security while also making the authentication process more efficient and streamlined for employees.

Single Sign-On (SSO) vs. Multi-Factor Authentication



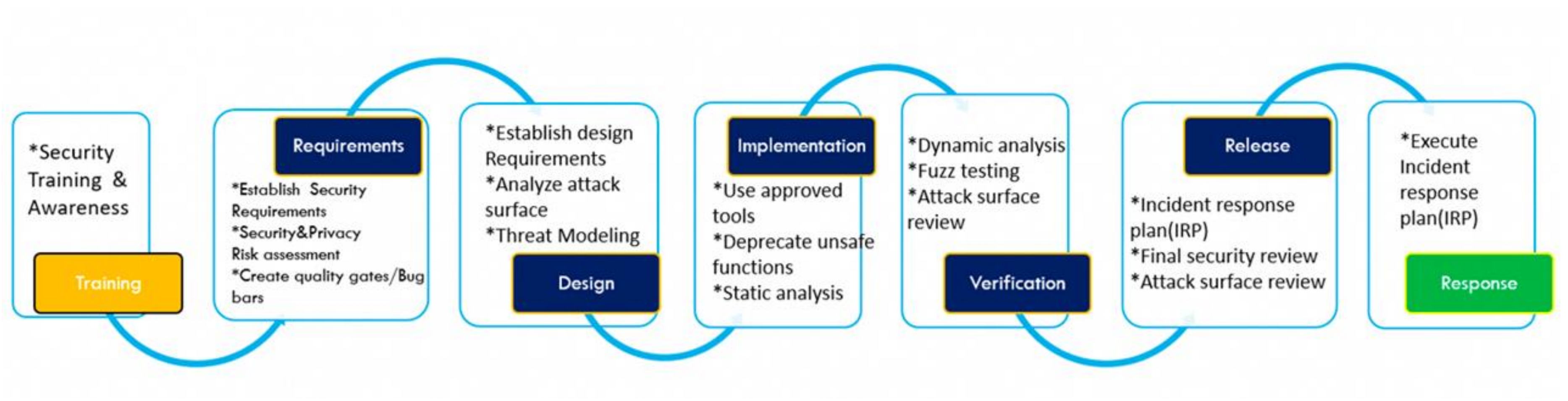
Single sign-on	Multi-factor authentication
Access multiple applications with one login	Access one or more applications using two different authentication methods
Users only need to remember one set of login credentials	Requires users to provide additional forms of ID
Reduce password fatigue	Reduce password complexity
Simplifies password authentication	Enhances security

Security as Software Requirement

Secure Software Development Life Cycle

- A secure SDLC involves integrating security testing and other activities into an existing development process.
- Examples include writing security requirements alongside functional requirements and performing an architecture risk analysis during the design phase of the SDLC.
- Many secure SDLC models are in use, but one of the best known is the Microsoft Security Development Lifecycle (MS SDL), which outlines 12 practices organizations can adopt to increase the security of their software.
- There is also the Secure Software Development Framework from the National Institutes of Standards and Technology (NIST), which focuses on security-related processes that organizations can integrate into their existing SDLC.

Secure Development Life Cycle



Training

*Security Training & Awareness

Requirements

*Establish Security Requirements
*Security&Privacy Risk assessment
*Create quality gates/Bug bars

Design

*Establish design Requirements
*Analyze attack surface
*Threat Modeling

Implementation

*Use approved tools
*Deprecate unsafe functions
*Static analysis

Verification

*Dynamic analysis
*Fuzz testing
*Attack surface review

Release

*Incident response plan(IRP)
*Final security review
*Attack surface review

Response

*Execute Incident response plan(IRP)

Advantages of Secure Development Life Cycle

- Your software is more secure.
- All stakeholders are aware of security considerations.
- You detect design flaws early, before they're coded into existence.
- You reduce your costs, thanks to early detection and resolution of defects.
- You reduce overall intrinsic business risks for your organization.

Threat Modelling

PLURALSIGHT

What is Threat Modeling?

- Threat Modeling is all about:
    - Identifying types of threat agents
    - Adopting perspective of hackers/intruders
    - Thinking how much damage is possible
- What to Model?
    - The application as a whole
    - Security & Privacy Features
    - Features whose failures have security or privacy implications
    - Features that cross trust boundaries

Why to perform Threat Modeling?

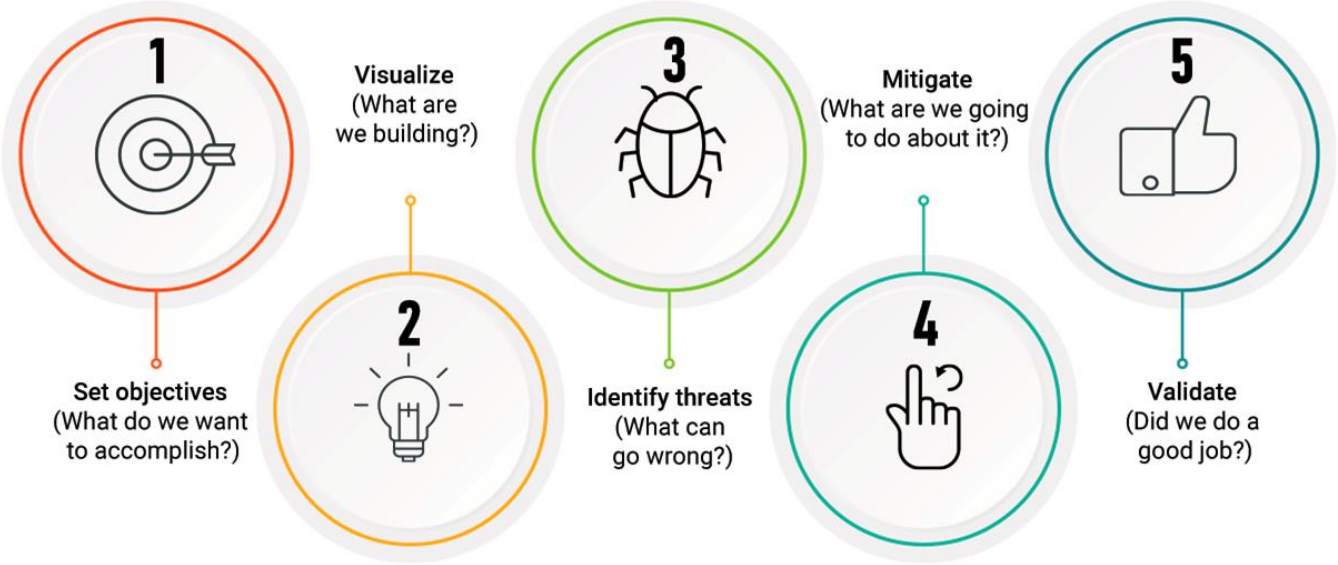
- Identifying likely threats and the probable consequences of successful attack is the method of investigation to identify an appropriate set of defenses.
- Better to identify threats early in the development process.
- Can help other security assessment activities.

When to perform Threat Modeling?

- Threat modeling can be used as part of requirements engineering to derive security requirements, based on a first architecture overview.
- It can also be used as a design analysis technique applied to the software design before coding starts.

BUT IT’s NEVER TOO LATE

How to perform Threat Modeling?



1

Set objectives
(What do we want
to accomplish?)

2

Visualize
(What are
we building?)

3

Identify threats
(What can
go wrong?)

4

Mitigate
(What are we going
to do about it?)

5

Validate
(Did we do a
good job?)

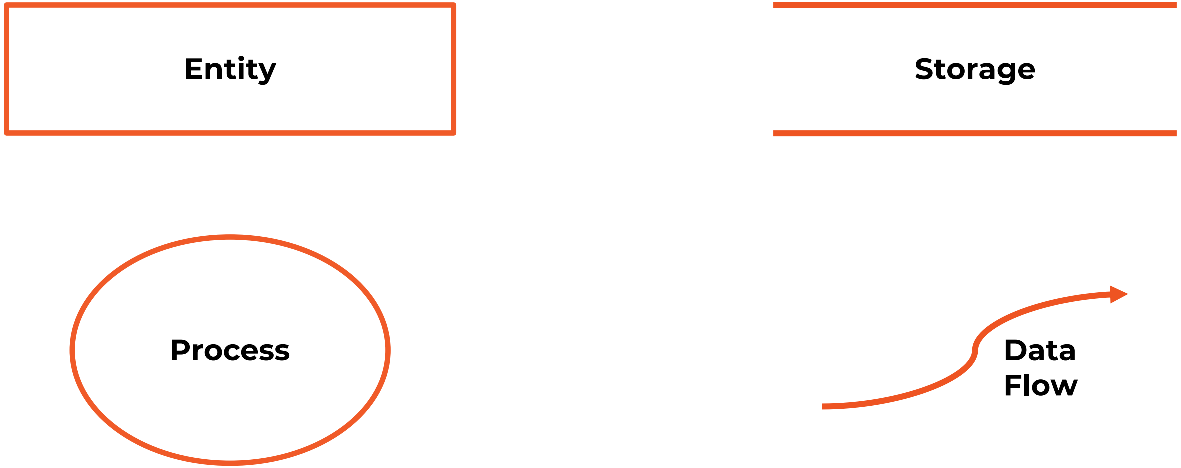
STRIDE Model

- The STRIDE approach to threat modeling was proposed by Loren Kohnfelder and Praerit Garg (Kohnfelder, 1999) and later adopted by Microsoft.
- The framework and mnemonic were designed to help developers identify the types of attacks that software tends to experience.

Attack	Attacks	Description
Spoofing	Authentication	Impersonating something or someone else.
Tampering	Integrity	Modifying data or code.
Repudiation	Non-repudiation	Claiming to have not performed an action.
Information Disclosure	Confidentiality	Exposing information to someone not authorized to see it.
Denial of Service	Availability	Deny or degrade service to users.
Elevation of Privilege	Authorization	Gain capabilities without proper authorization.

Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system. It shows how information is input to and output from the system, the sources and destinations of that information, and where that information is stored.



DFD and STRIDE

Elements	S	T	R	I	D	E
▭	X		X			
◯	X	X	X	X	X	X
═		X	X	X	X	
⤴		X		X	X	

Types of STRIDE models

STRIDE-per-element	STRIDE-per-interaction
● It makes STRIDE more prescriptive by observing that certain threats are more prevalent with certain elements of a diagram. ● For example, a data store is unlikely to spoof another data store (although running code can be confused as to which data store it is accessing) ● By focusing on a set of threats against each element, this approach makes it easier to find threats. ● STRIDE-per-element does have two weaknesses. First, similar issues tend to crop up repeatedly in a given threat model; second, the chart may not represent your specific issues.	● In reality, threats don't show up in a vacuum. They show up in the interactions of the system. ● This model is a simplified approach to identifying threats, designed to be easily understood by the beginners. ● STRIDE-per interaction is an approach to threat enumeration that considers tuples of (origin, destination, interaction) and enumerates threats against them. ● For example, unauthorized access to sensitive customer details through a compromised login page on an e-commerce website.

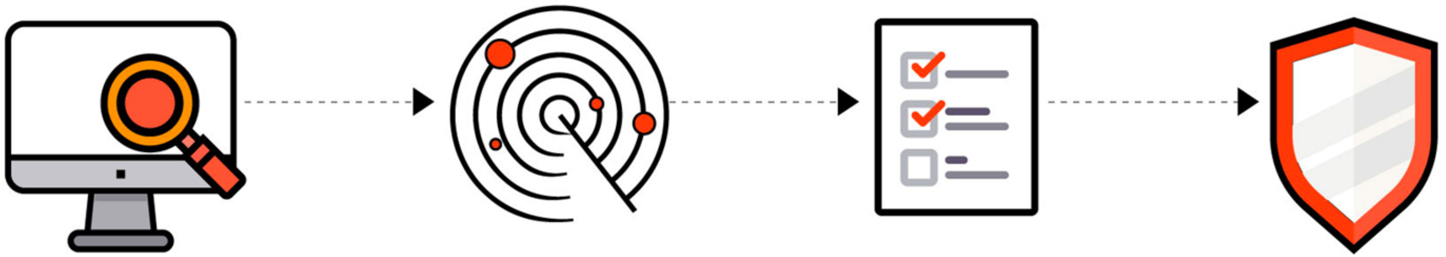
Vulnerability Assessment vs. Penetration Testing

PLURALSIGHT

Vulnerability Assessment

- Vulnerability assessment (VA) is the process of defining, detecting, categorizing, and prioritizing security vulnerabilities in a computer system, application, or network.
- Organizations rely on vulnerability assessments to provide the crucial intelligence and risk context to understand and respond to cybersecurity threats.
- The vulnerability assessment process aims to identify threats and their associated risks. It usually involves using an automated testing tool, such as a network security scanner.
- The vulnerability assessment report lists the vulnerabilities in the system under test and possible reasons for the same.

Vulnerability Assessment



Asset discovery

Detect and manage local and remote endpoints, roaming devices, and closed network (DMZ) machines.

Vulnerability scanning

Spot all OS and third-party vulnerabilities, including vulnerabilities in content management systems, web servers and database software.

Vulnerability assessment

Visualize, analyze, and prioritize vulnerabilities based on CVSS scores, age, exploitability, patch availability, impact type, affected asset count and more.

Vulnerability remediation

Deploy automatically correlated patches to seal vulnerabilities, and leverage alternative mitigation measures if no patch is available.

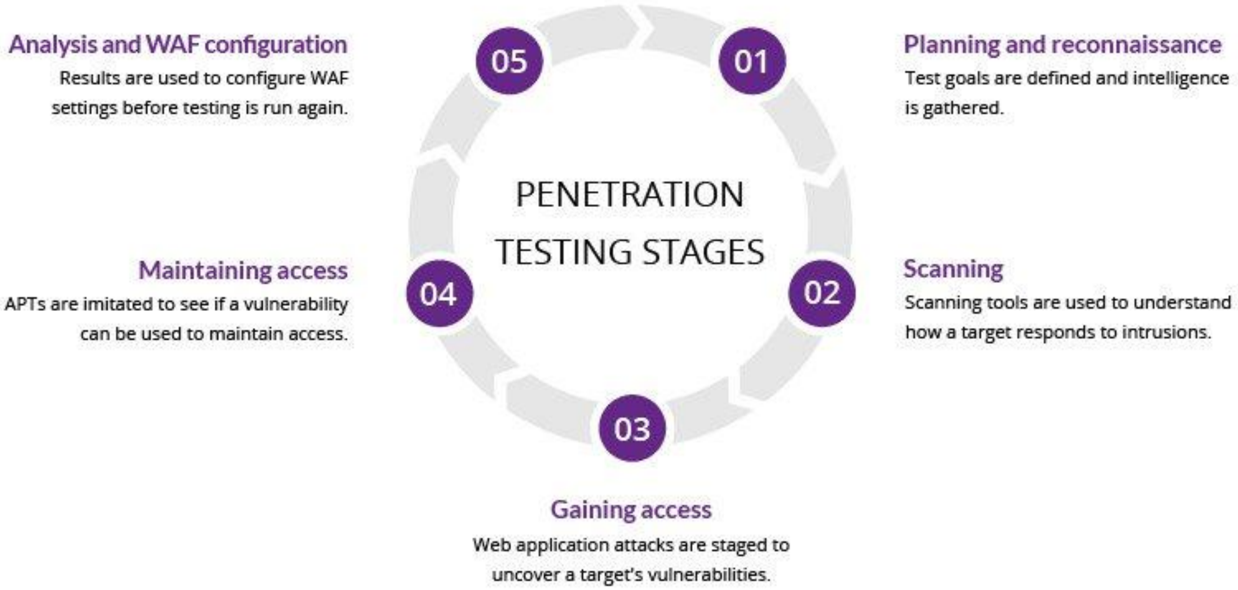
Why Vulnerability Assessment?

- A vulnerability assessment provides an organization detailed information about security vulnerabilities in their environment.
- It also offers guidelines for assessing the risks associated with these vulnerabilities.
- This process allows organizations to better understand their assets, security vulnerabilities, and overall risk, making it less likely for attackers to compromise their systems and steal their information.
- Vulnerability assessments help identify flaws and threats proactively and as soon as possible allowing orgs to take remedial action to patch the gaps in the organizational infrastructure.
- They are also important for ensuring organizations meet cybersecurity compliance requirements, such as the HIPAA and PCI DSS standards.

Penetration Testing

- Penetration testing is a security method that allows organizations to identify, test, and prioritize vulnerabilities in computer systems and networks.
- Usually, ethical hackers — internal employees or third-party contractors – perform penetration tests.
- Penetration testers imitate the tactics and behaviors of attackers to assess the security posture of an organization’s network, computer system, or web application.
- Organizations can also use penetration testing to test compliance with industry standards and regulations.

Penetration Testing



PENETRATION TESTING STAGES

01

Planning and reconnaissance

Test goals are defined and intelligence is gathered.

02

Scanning

Scanning tools are used to understand how a target responds to intrusions.

03

Gaining access

Web application attacks are staged to uncover a target's vulnerabilities.

04

Maintaining access

APTs are imitated to see if a vulnerability can be used to maintain access.

05

Analysis and WAF configuration

Results are used to configure WAF settings before testing is run again.

WAF: Web Application Firewall
APT: Advanced Persistent Threat

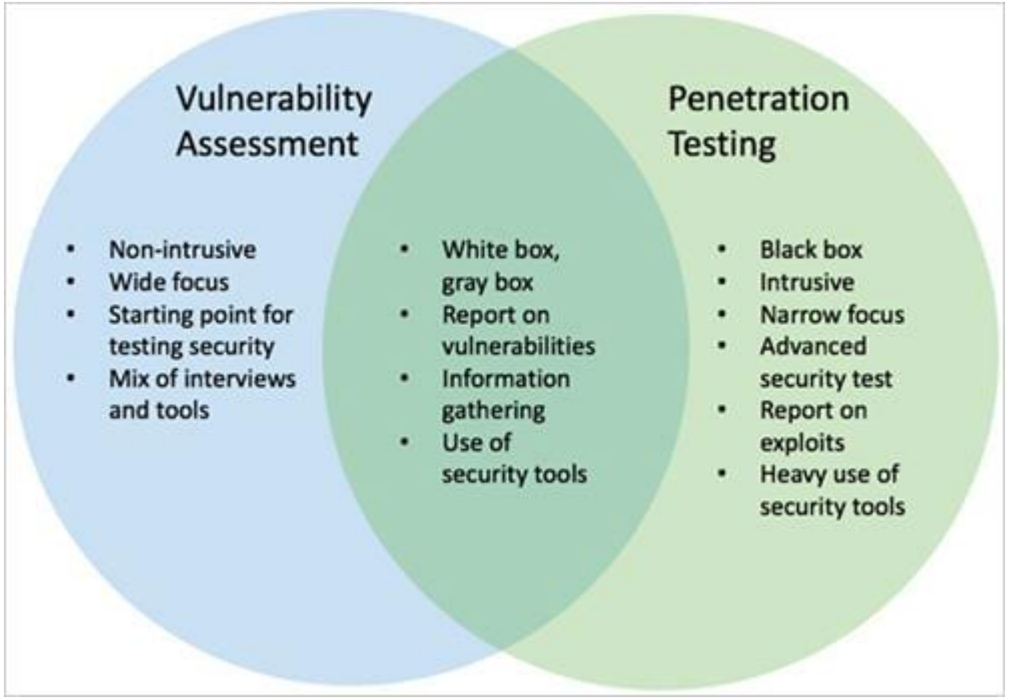
Why Penetration Testing?

- The incidence of distributed denial of service (DDoS), phishing, and ransomware attacks is increasing rapidly, placing all internet-based businesses at risk.
- The consequences of successful a cyber-attack are greater than ever, given businesses' reliance on digital technologies.
- Penetration testing leverages a hacker's perspective to identify, prevent, and mitigate security risks before a malicious actor can exploit them.
- It helps the IT leadership implement smart security upgrades to minimize the chances and likelihood of a successful attack.
- To protect their assets from penetration attacks effectively, businesses must be able to update their security measures simultaneously.

Vulnerability Assessment & Penetration Testing (VAPT)

- VAPT is a comprehensive suite of security assessment services that help identify and mitigate cybersecurity threats and the associated risks to an organization's IT assets.
- It provides businesses with a highly detailed assessment of their applications, offering deeper insights than individual penetration tests.
- The VAPT approach helps organizations better understand the threats their applications face, allowing them to protect their data and systems from malicious attacks.
- Vulnerabilities are often present in internally created or third-party applications and software. However, most issues are easy to fix once discovered.
- VAPT providers allow security teams to focus on addressing critical flaws while the providers continue to discover, triage, and prioritize vulnerabilities.

Vulnerability Assessment & Penetration Testing (VAPT)



Vulnerability Assessment

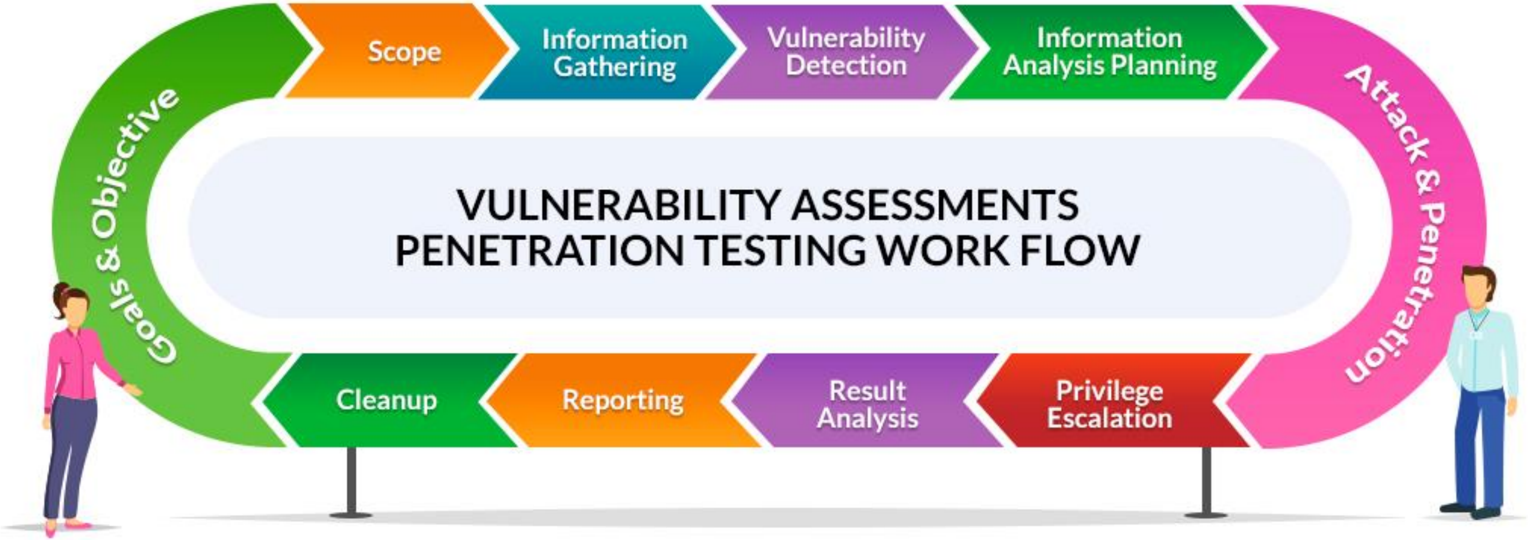
- Non-intrusive
- Wide focus
- Starting point for testing security
- Mix of interviews and tools

- White box, gray box
- Report on vulnerabilities
- Information gathering
- Use of security tools

Penetration Testing

- Black box
- Intrusive
- Narrow focus
- Advanced security test
- Report on exploits
- Heavy use of security tools

VAPT Workflow

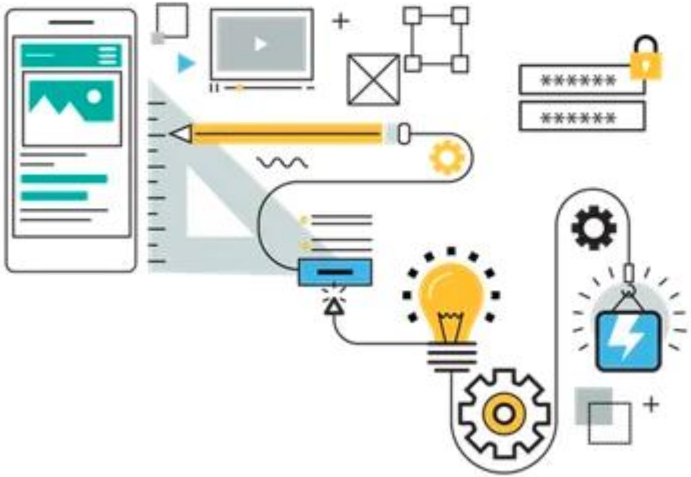


Code Review & Penetration Testing

PLURALSIGHT

Secure Coding

- Secure coding is the practice of developing software that is safeguarded from security vulnerabilities.



Code Review

- Code review is an important step in the software development process to get a second opinion on the solution and implementation.
- It is a peer review of code that helps developers ensure or improve the code quality before they merge and ship it.
- The reviewer can also act as a second step in identifying bugs, logic problems, uncovered edge cases, or other issues.
- Reviewers can be from any team or group as long as they’re a domain expert.
- If the lines of code cover more than one domain, two experts should review the code.

Approaches of Code Review

Pair Programming

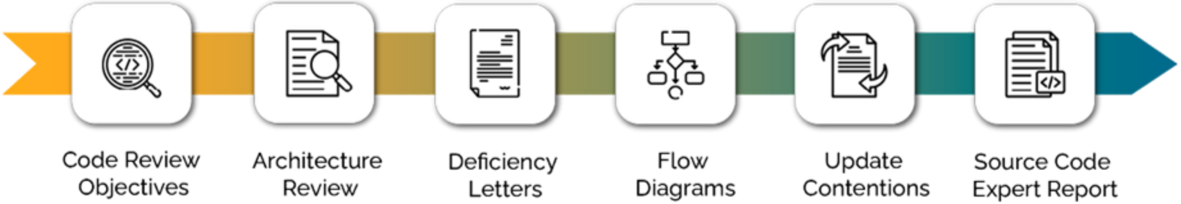
Over-the-shoulder Review

Tool Assisted Review

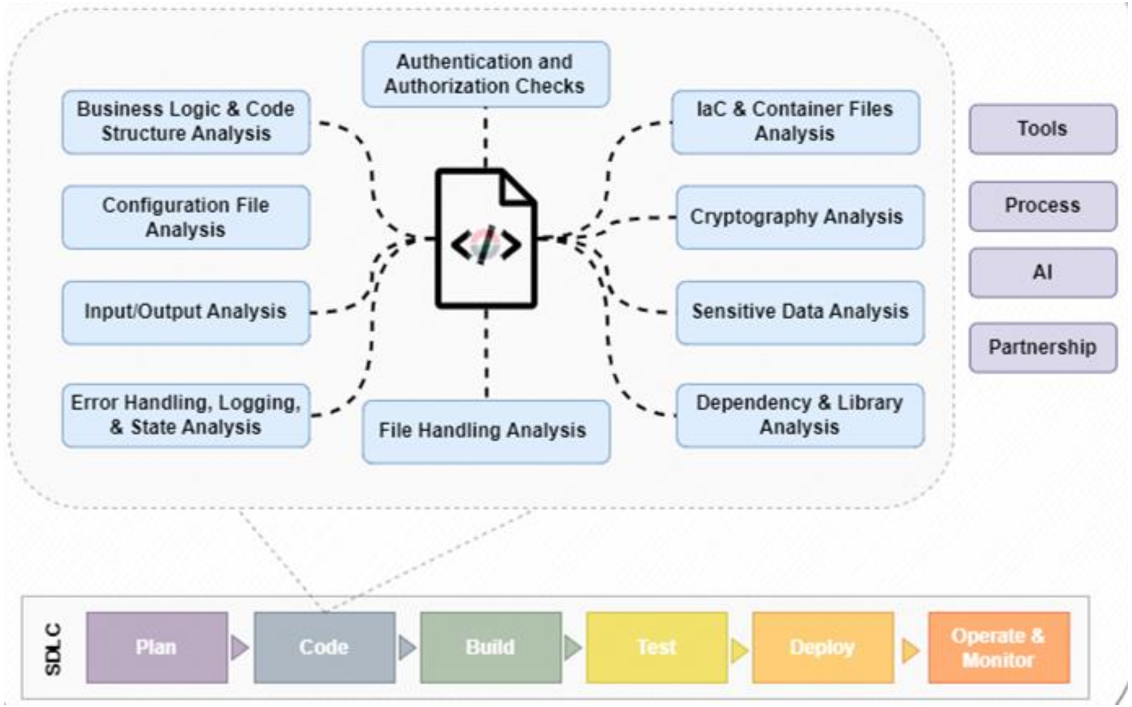
Email Pass-around

Secure Code Review

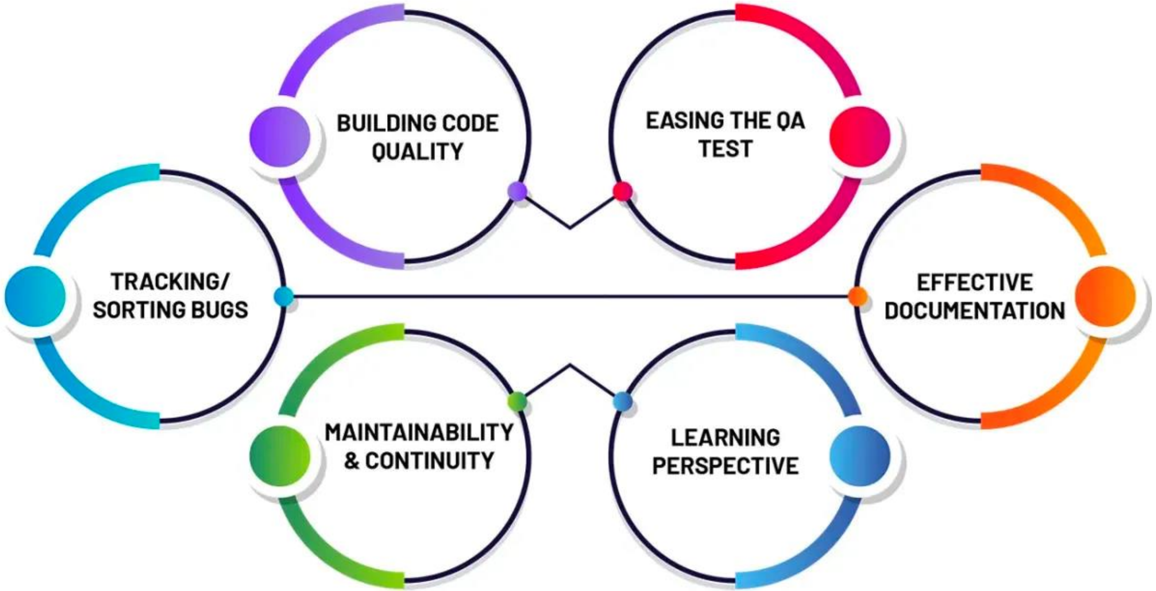
- A secure code review is a specialized task involving manual and/or automated review of an application's source code in an attempt to identify security-related weaknesses (flaws) in the code.
- It does not attempt to identify every issue in the code, but instead looks to provide insight into what types of problems exist and to help the developers of the application understand what classes of issues are present.
- The goal is to arm the developers with information to help them make the application's source code more sound and secure.



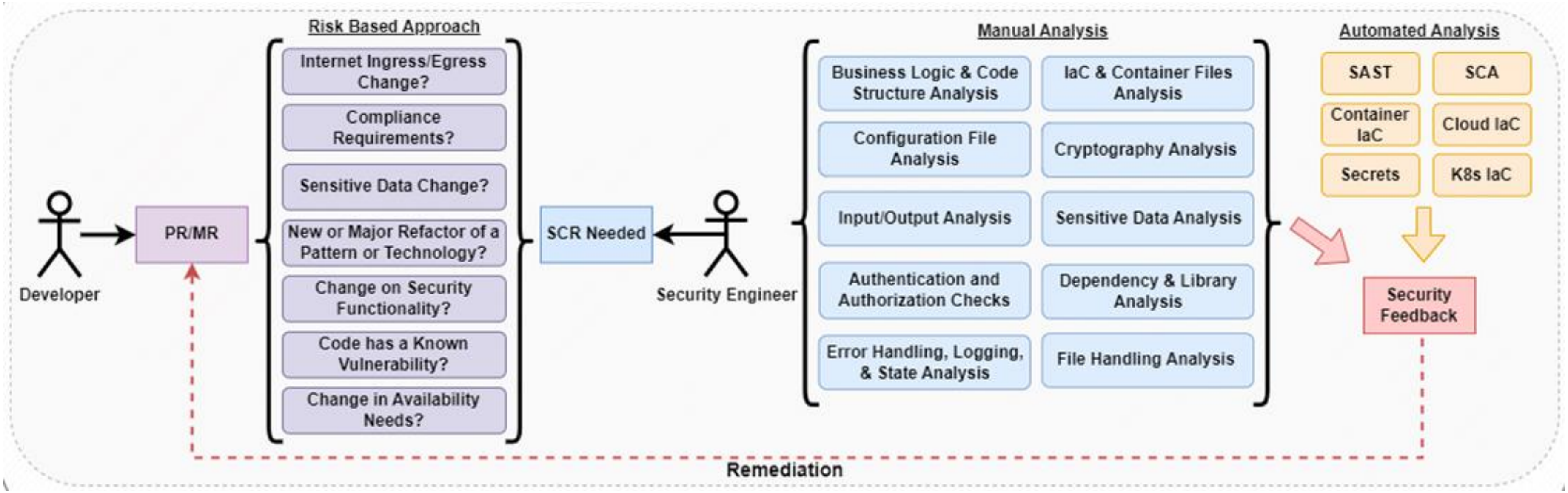
Secure Code Review


Authentication and Authorization Checks
Business Logic & Code Structure Analysis
IaC & Container Files Analysis
Configuration File Analysis
Cryptography Analysis
Input/Output Analysis
Sensitive Data Analysis
Error Handling, Logging, & State Analysis
File Handling Analysis
Dependency & Library Analysis
Tools
Process
AI
Partnership
SDLC
Plan
Code
Build
Test
Deploy
Operate & Monitor

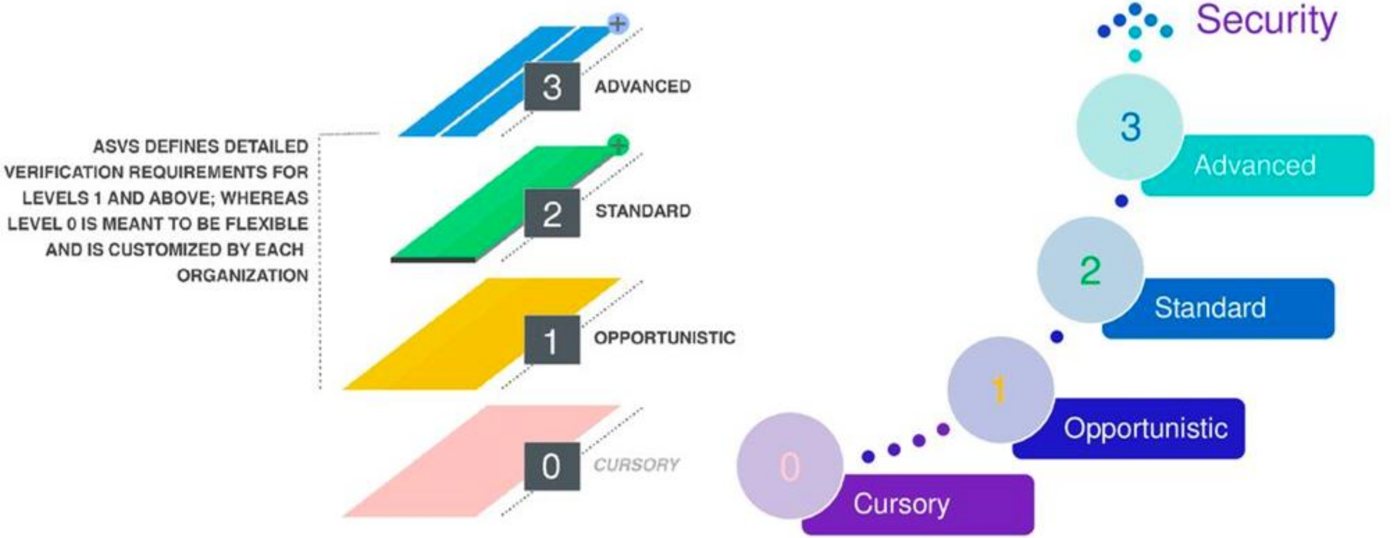
Secure Code Review Advantages



Building effective Secure Review Code


Risk Based Approach
Internet Ingress/Egress Change?
Compliance Requirements?
Sensitive Data Change?
New or Major Refactor of a Pattern or Technology?
Change on Security Functionality?
Code has a Known Vulnerability?
Change in Availability Needs?
Developer
PR/MR
SCR Needed
Security Engineer
Manual Analysis
Business Logic & Code Structure Analysis
IaC & Container Files Analysis
Configuration File Analysis
Cryptography Analysis
Input/Output Analysis
Sensitive Data Analysis
Authentication and Authorization Checks
Dependency & Library Analysis
Error Handling, Logging, & State Analysis
File Handling Analysis
Automated Analysis
SAST
SCA
Container IaC
Cloud IaC
Secrets
K8s IaC
Security Feedback
Remediation

OWASP ASVP Levels



ASVS DEFINES DETAILED VERIFICATION REQUIREMENTS FOR LEVELS 1 AND ABOVE; WHEREAS LEVEL 0 IS MEANT TO BE FLEXIBLE AND IS CUSTOMIZED BY EACH ORGANIZATION

Privacy by Design and by Default

PLURALSIGHT

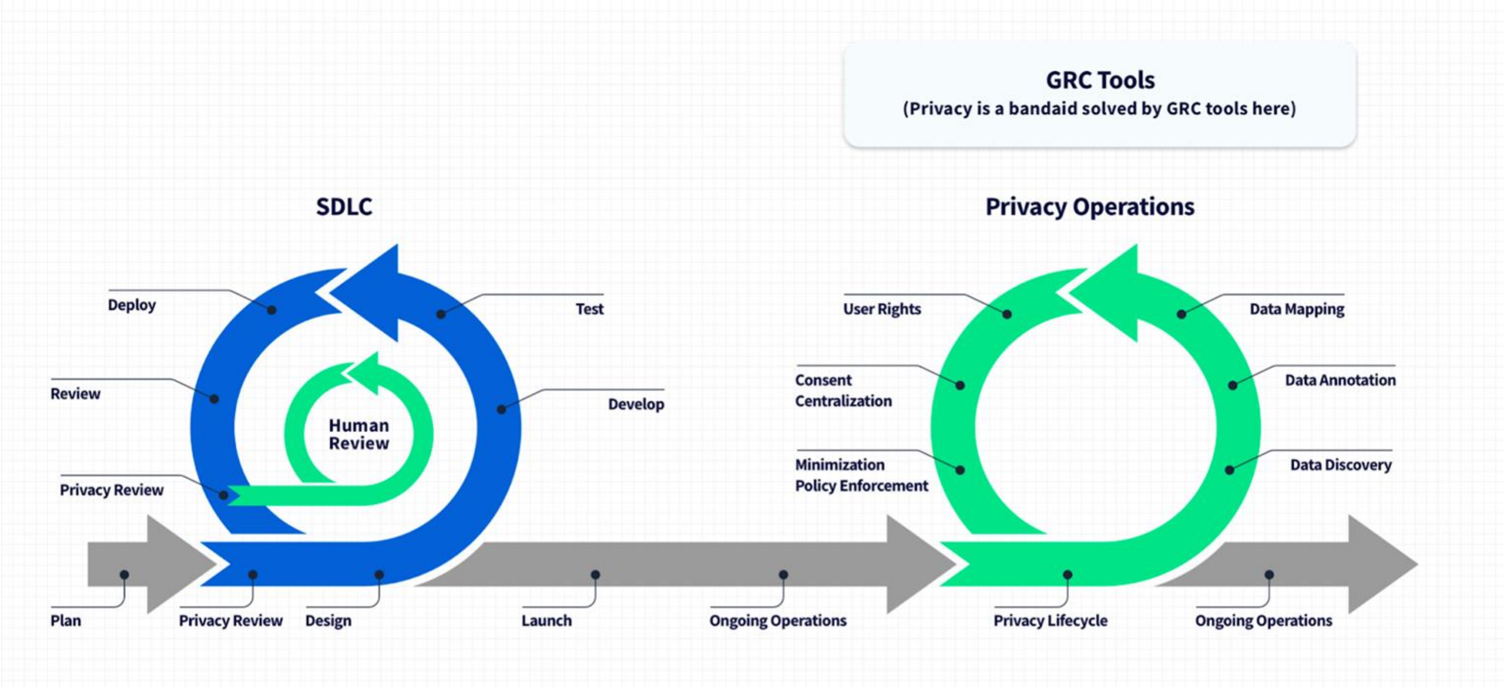
Privacy by Design & Privacy by Default

- In this complex electronic business environment, a “check the box” compliance model leads to a false sense of security.
- *Privacy-by-Design* is a framework based on proactively embedding privacy into the design and operations of IT systems, networked infrastructure, and business practices.
- *Privacy-by-Default* means that when a system or service includes choices for the individual on how much personal data he/she shares with others, the default settings should be the most privacy friendly ones.

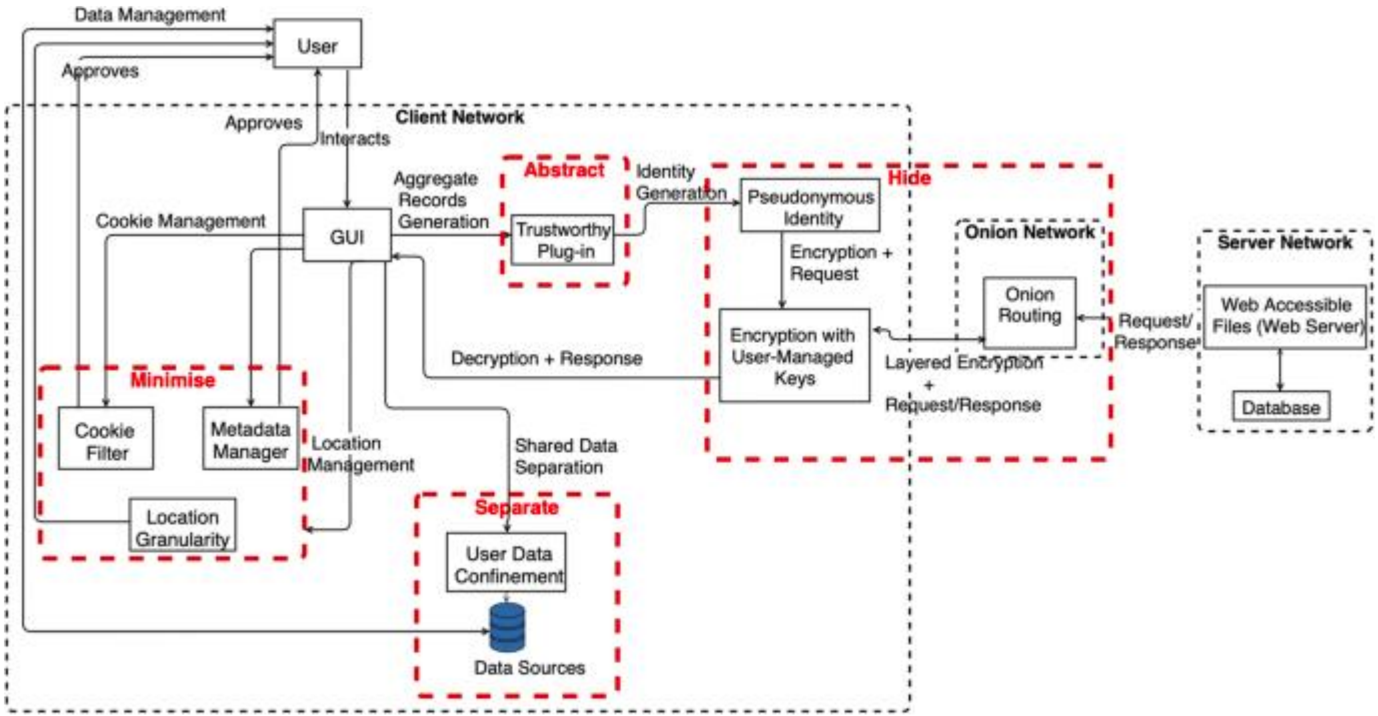
Why Privacy in Software Development?

- Being transparent in how you collect consumers' personal data and what you do with it not only helps you comply with data privacy laws, but it also establishes trust with your customers.
- A good way to start is to understand what customers want in terms of data privacy, including transparency, security, and portability, along with timely communication and a clear understanding of their rights.
- Software development teams should be considering these business needs when developing software applications.
- Both customers and businesses benefit from awareness about data privacy laws and the rights of the consumer.
- Building software solutions with the flexibility to adapt and respond to the shifting tides of data privacy legislation will help better position your company for success.

Privacy in SDLC



Security & Privacy in Software Development



Lab Excercise

PLURALSIGHT