



# Security Essentials

## Key Takeaways

# All rights reserved to nnSoftware GmbH

No part of this publication may be reproduced, copied, transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of nnSoftware GmbH

# About TechWorld with Nana

TechWorld with Nana is an established name in the DevOps and Cloud industry, and it stands for the quality trainings helping 1,000s of engineers acquire the most in-demand skills in this field.



Our mission is enable individual engineers as well as companies to take advantage of the recent developments in Cloud and DevOps fields, to use technologies and concepts in order to create efficient, automated, streamlined DevSecOps processes in organisations.

# Importance of Security



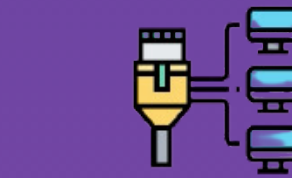
# Targets for hacker attacks



Security is defence against malicious actors

## What needs to be secured

- Customer data
- Company data
- Internal Applications
- User Applications



Internal Network



Online Systems



Servers



Apps



Database



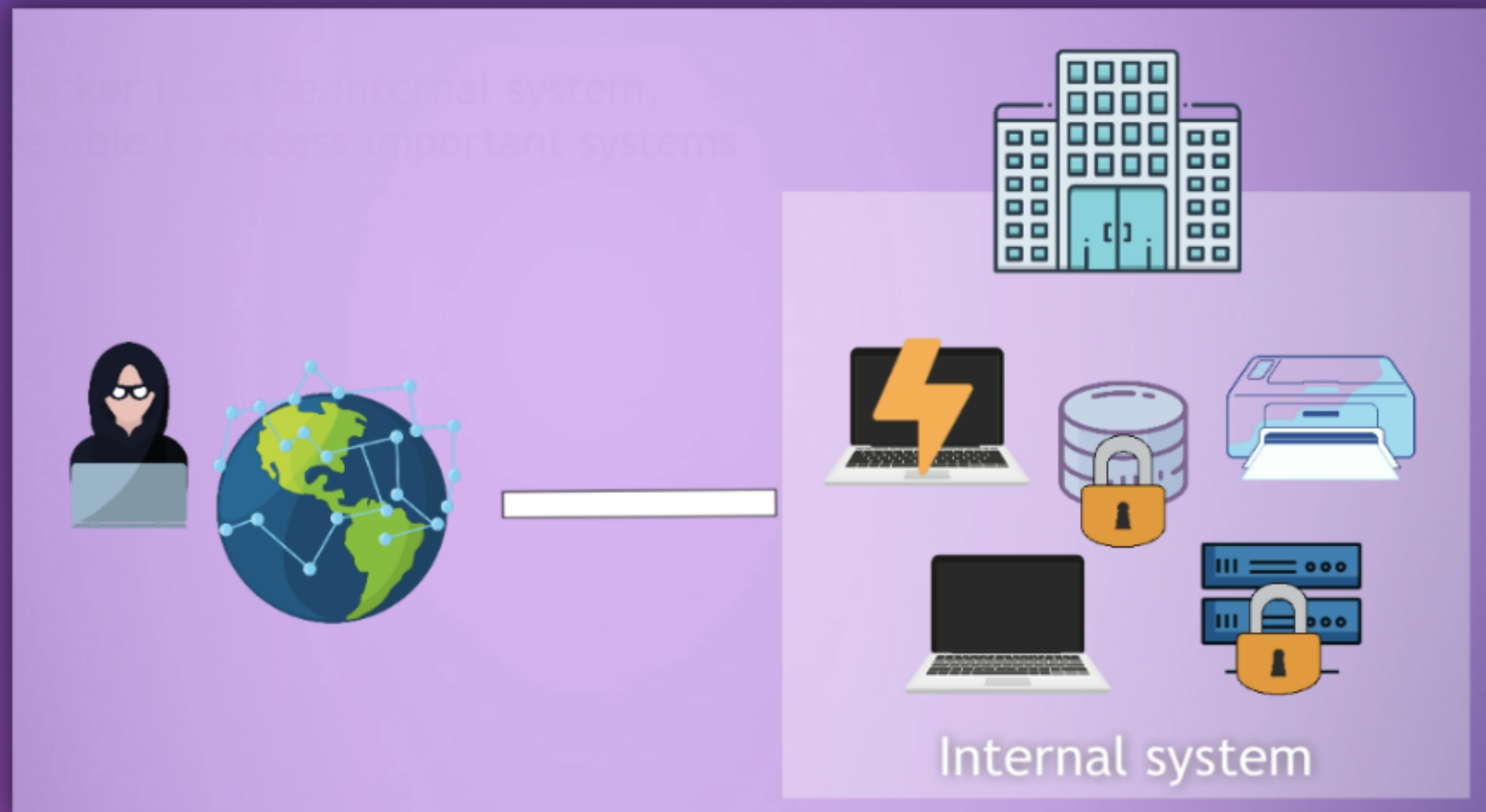
Employee's hardware



Printer

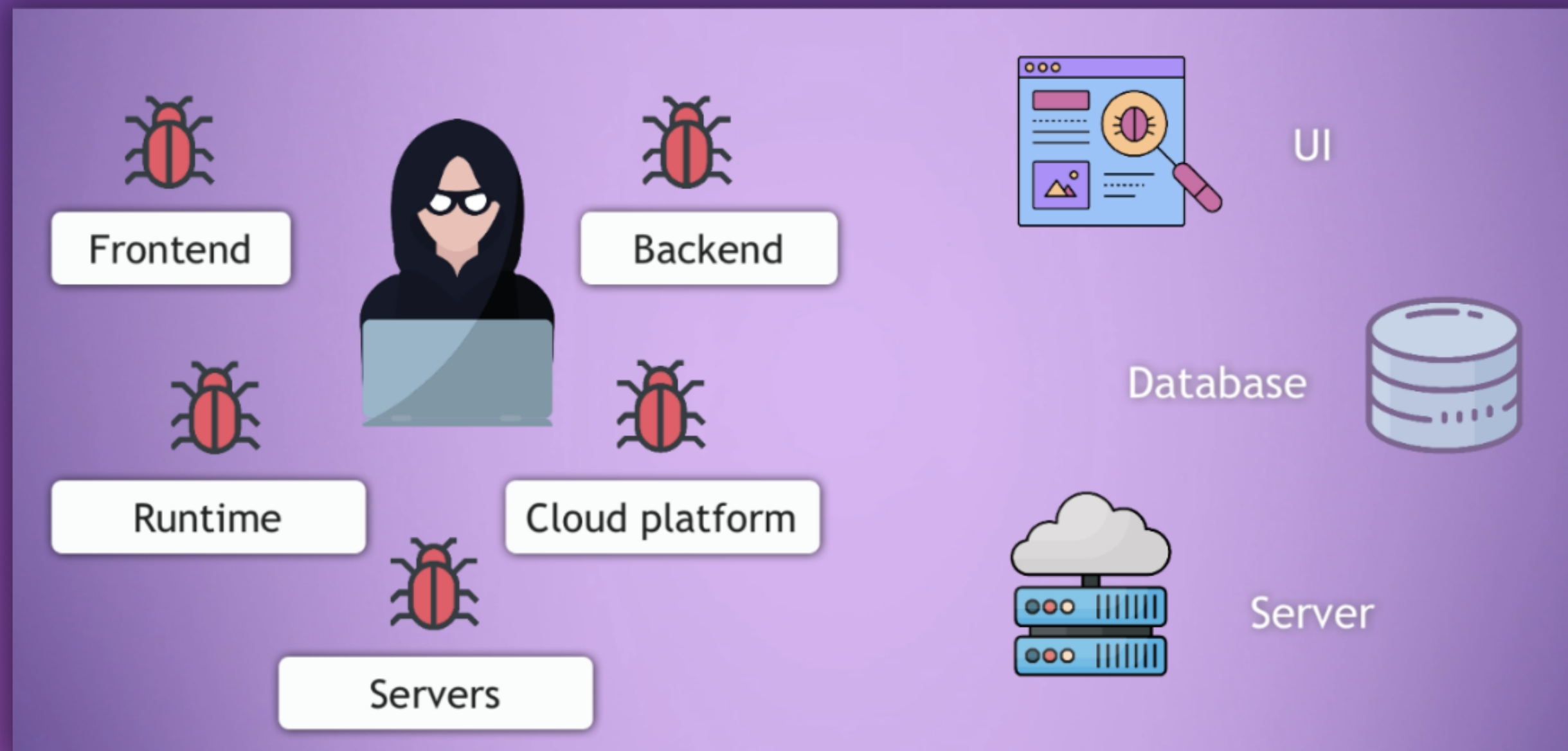
# Importance of security at many levels

Hacking into an internal system should not give access to all systems



# Application security

All application components need to be secured





# Types of Security Attacks

# Types of attacks

Different hacking methods and tactics to get into different systems





# Phishing Attack

A human as a hacking target, instead of a system

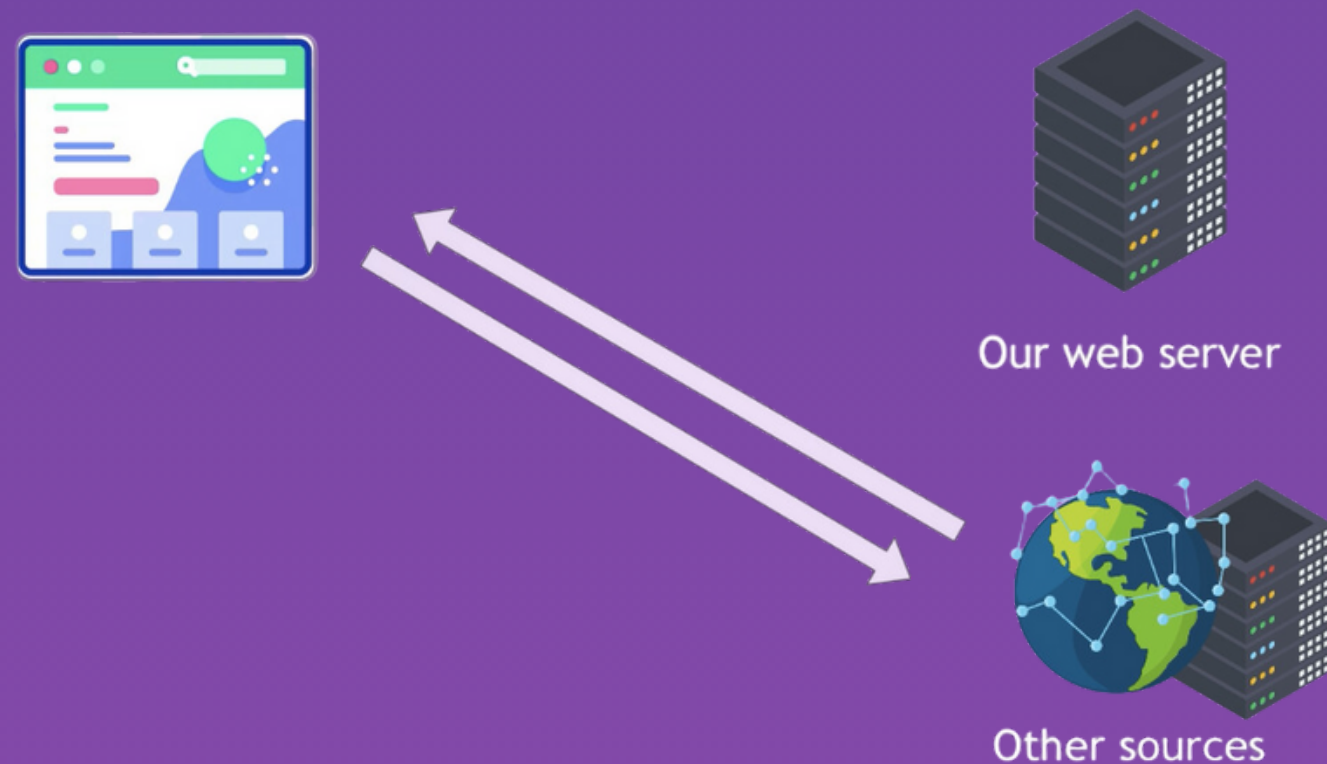
## How it works

1. Personalised phishing emails
2. Redirects to malicious pages
3. Links to malicious websites
4. Downloading malicious scripts



# XSS - Cross Site Scripting

Execute scripts from malicious sources



- Application allows fetching script from anywhere and executes malicious code

## How it works

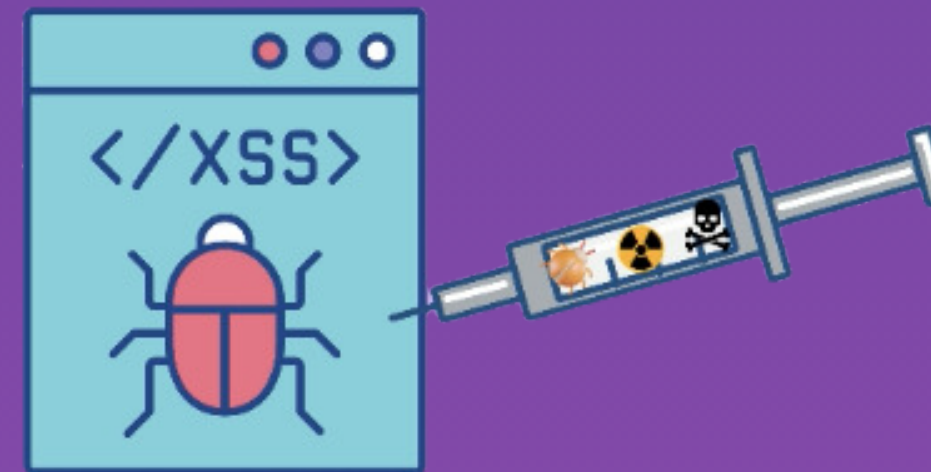
- An attacker often exploits input fields on a website (like comment sections) to inject malicious code
- When other users visit the affected page, the script is served to their browsers, which is executed





# CSRF - Client Side Request Forgery

- Common impact of XSS: Stealing user identity
- Attacker forging request = Pretending to be another user
- Client-side = Attack happens on user/client side



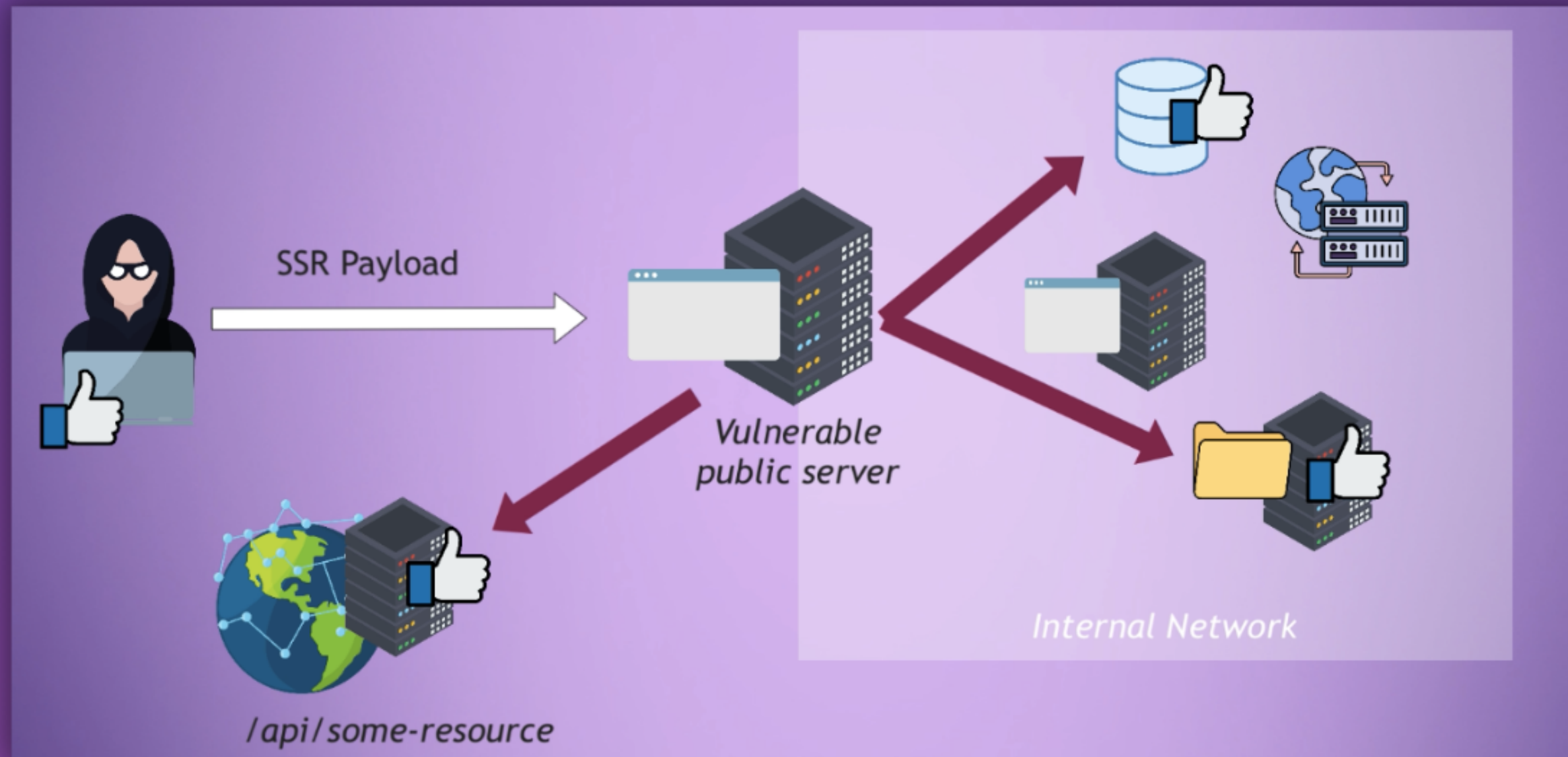
Use a website's script injection vulnerability

That malicious code steals session information

# SSRF - Server Side Request Forgery

## Forging server request

- Attacker manipulates server to send requests to protected resources



# SQL Injection



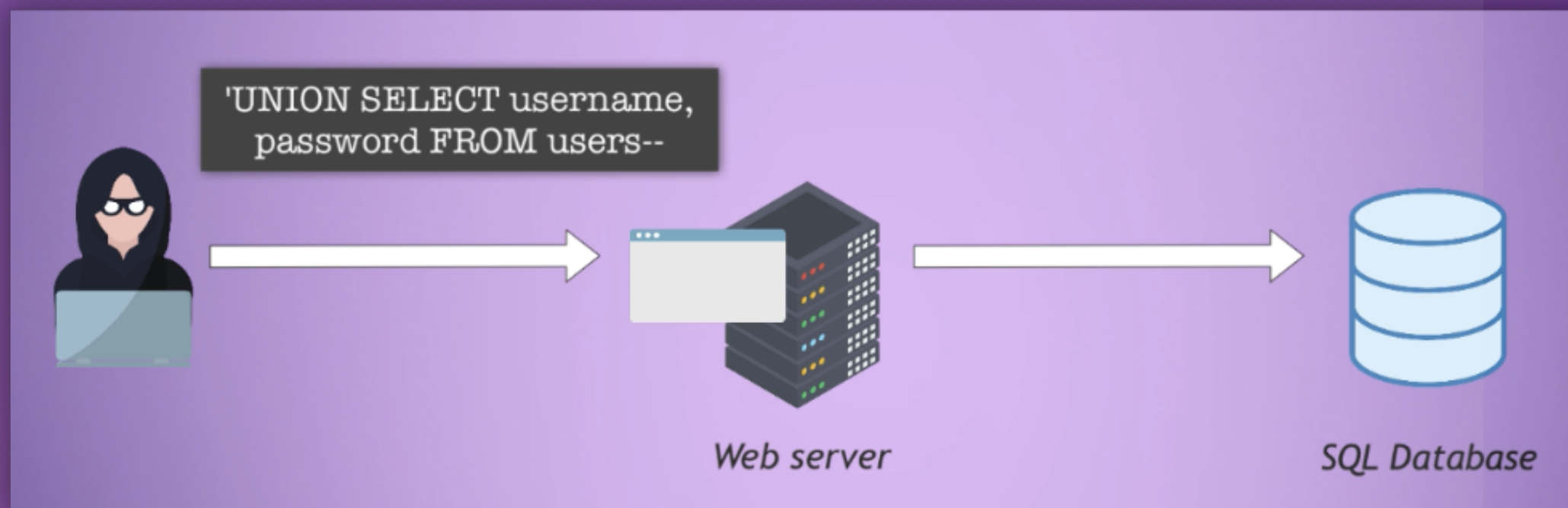
## Access to Database

### How it works

- SQL injection typically occurs when a web application doesn't properly validate or sanitize user inputs before incorporating them into SQL queries
- User inputs include data entered into web forms, URL parameters or cookies

### Exploitations include

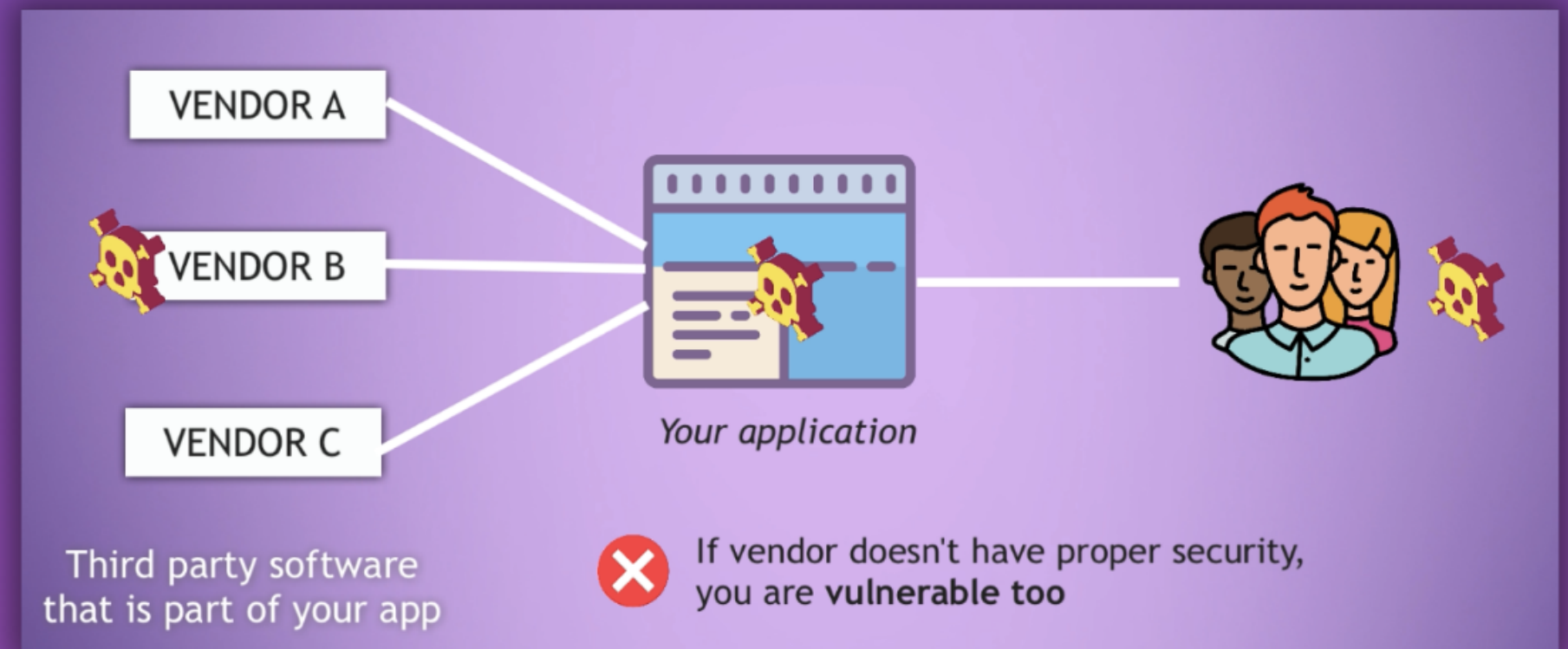
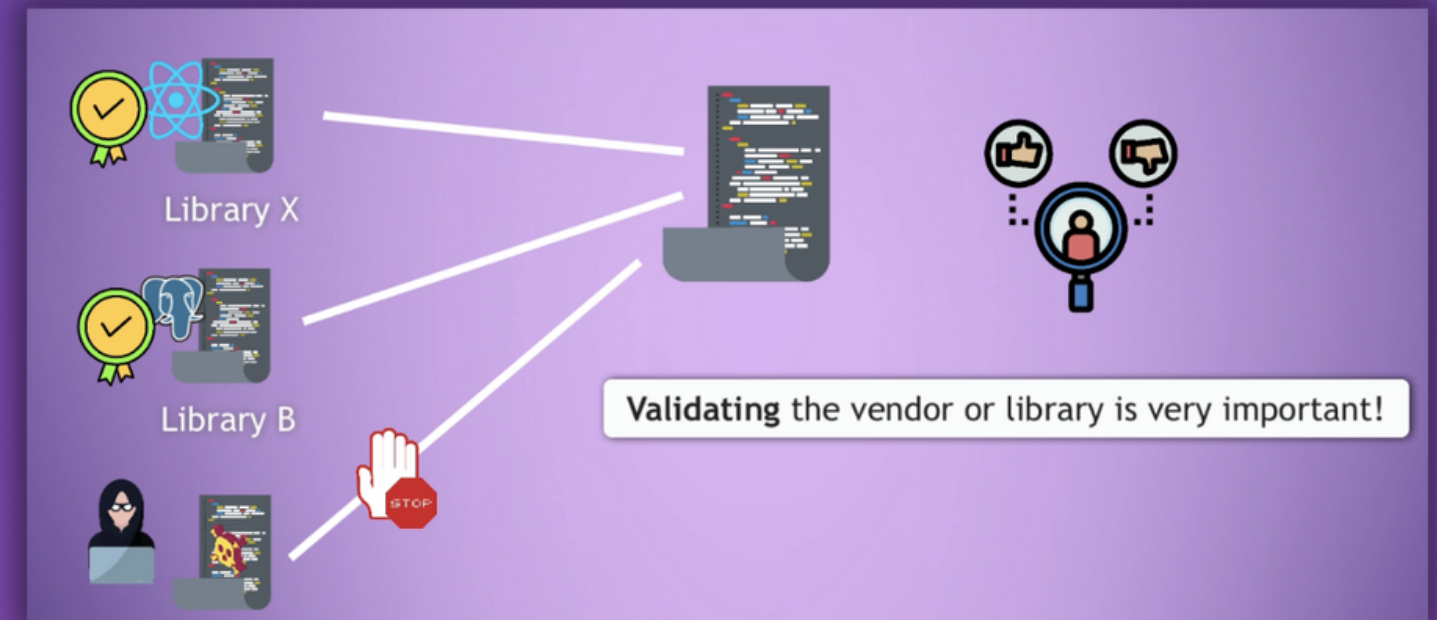
- ✗ Fetch sensitive user data
- ✗ Manipulate DB data
- ✗ Delete data





# Security Issues in Libraries

- An application often has many dependencies (libraries, frameworks), which is a main part of your software
- These libraries can have vulnerabilities, just like your own code
- Many applications don't know they are using vulnerable 3rd-party code
- Hackers can exploit those 3rd-party security holes



# Weak Password



## Brute Force Attacks

- Cybercriminals use trial and error to try and break into a network or website
- They typically use hacking tools to automate these login attempts

## Weak Passwords

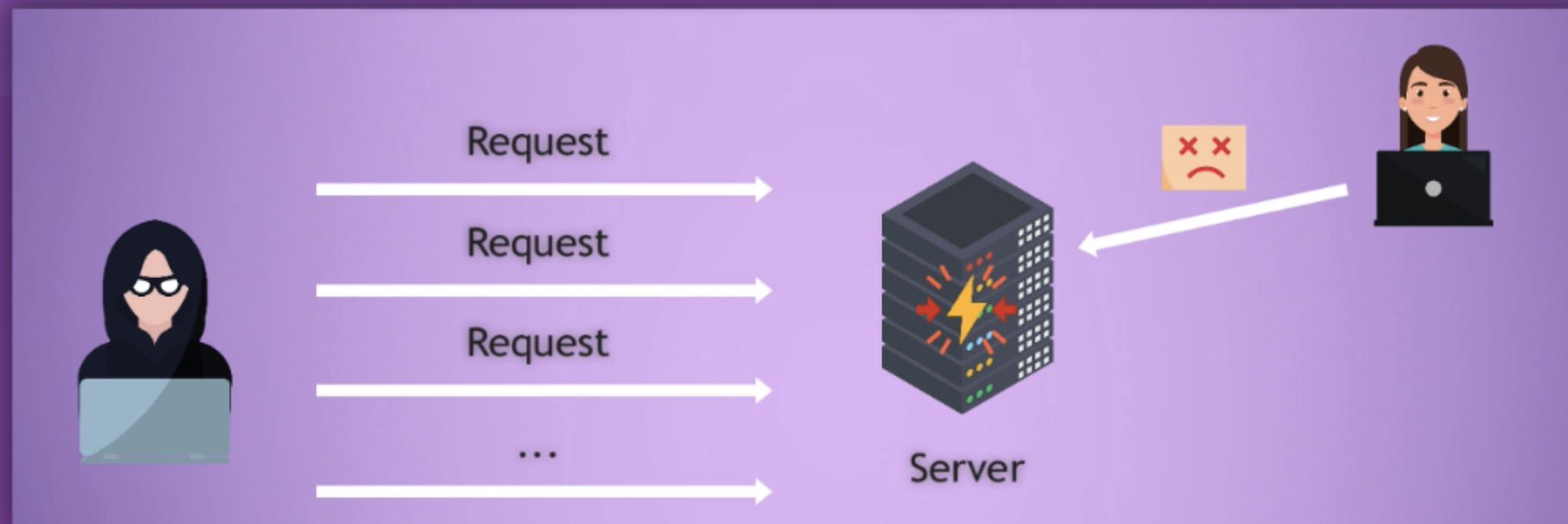
- But people also use weak passwords and use same passwords for multiple systems
- For companies, it's essential to enforce strong password policies
- Password manager tools can help



# DoS - Denial of Service Attack

## How it works

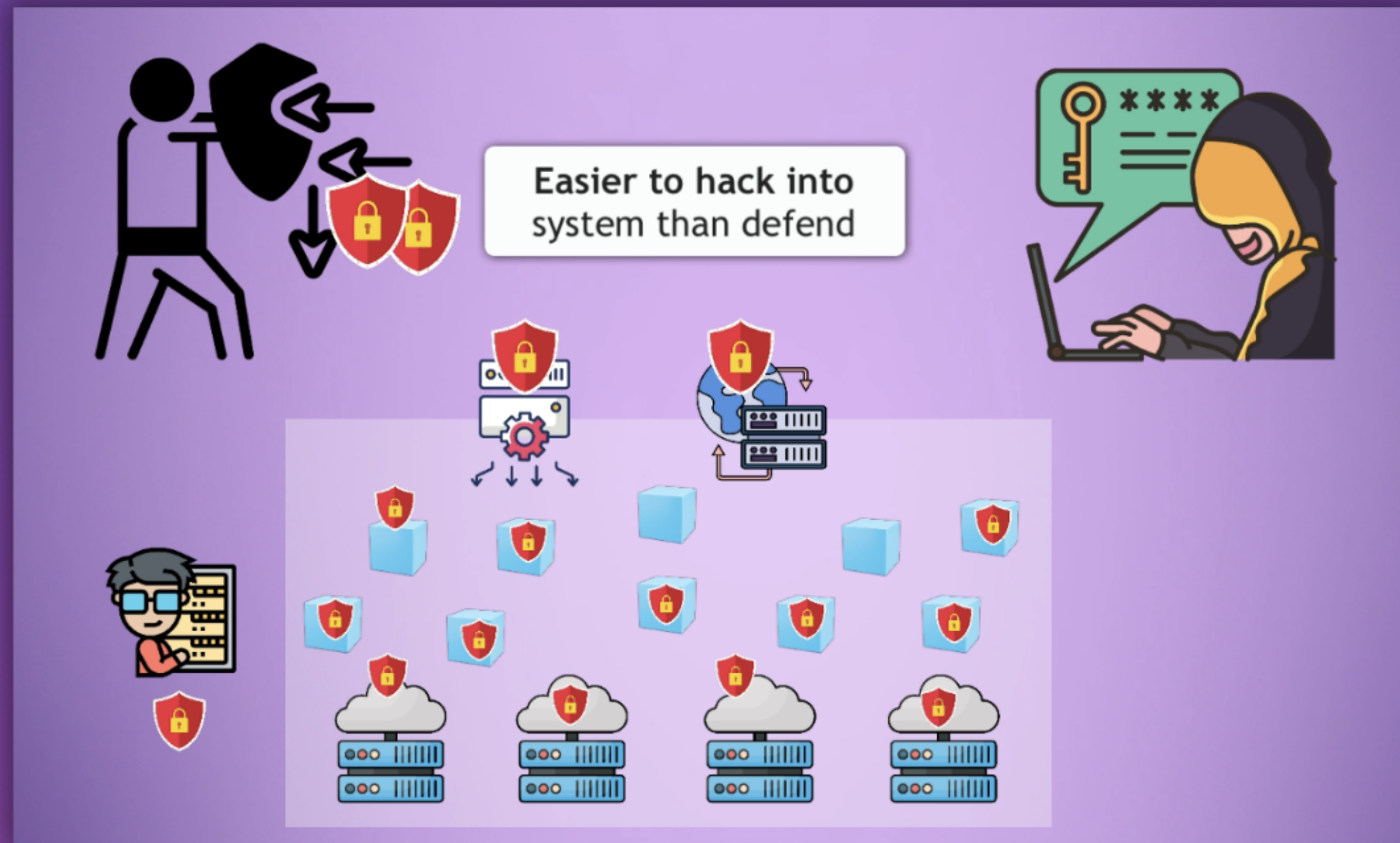
- DoS makes an application unable to respond to legitimate user requests.
- Often 1000s of bots sending requests at once
- This results in the attacker consuming all available **bandwidth**, consuming too many **connections** or exhausts **server's resources**





# Security Principle

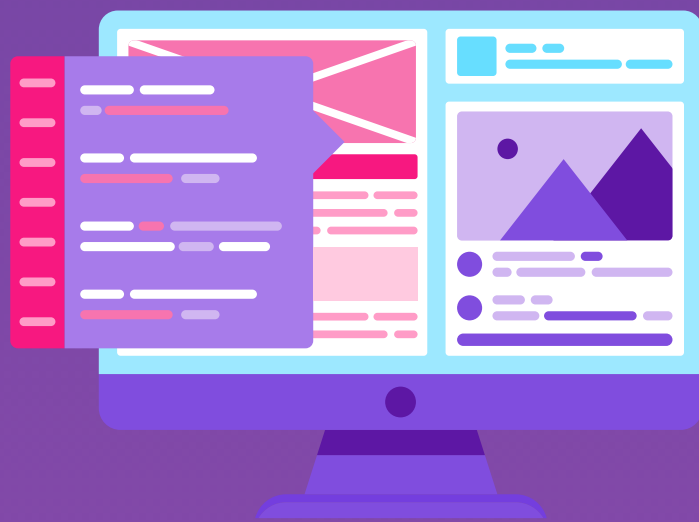
Easier to hack into systems than to defend them





# Security Posture

- Refers to organisation's overall security status and ability to defend against and respond to security threats and incidents



We need to know how secure the applications are?



How secure is the underlying infrastructure?

# OWASP and OWASP Top Ten

# What is OWASP

**OWASP stands for Open Web Application Security Project**

- A nonprofit organization dedicated to improving the security of software
- Raises awareness about common web application security risks
- Provides resources, tools and documentation to help organizations and developers enhance the security of their web applications
- One of OWASP's most well-known initiatives is the OWASP Top Ten

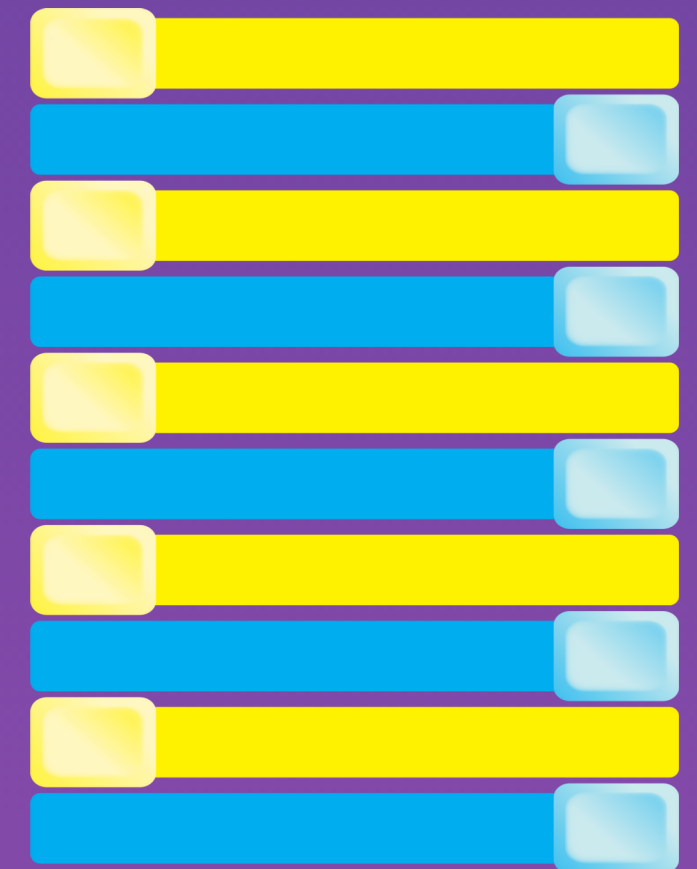


# OWASP Top Ten

**Lists the most critical web application security risks**

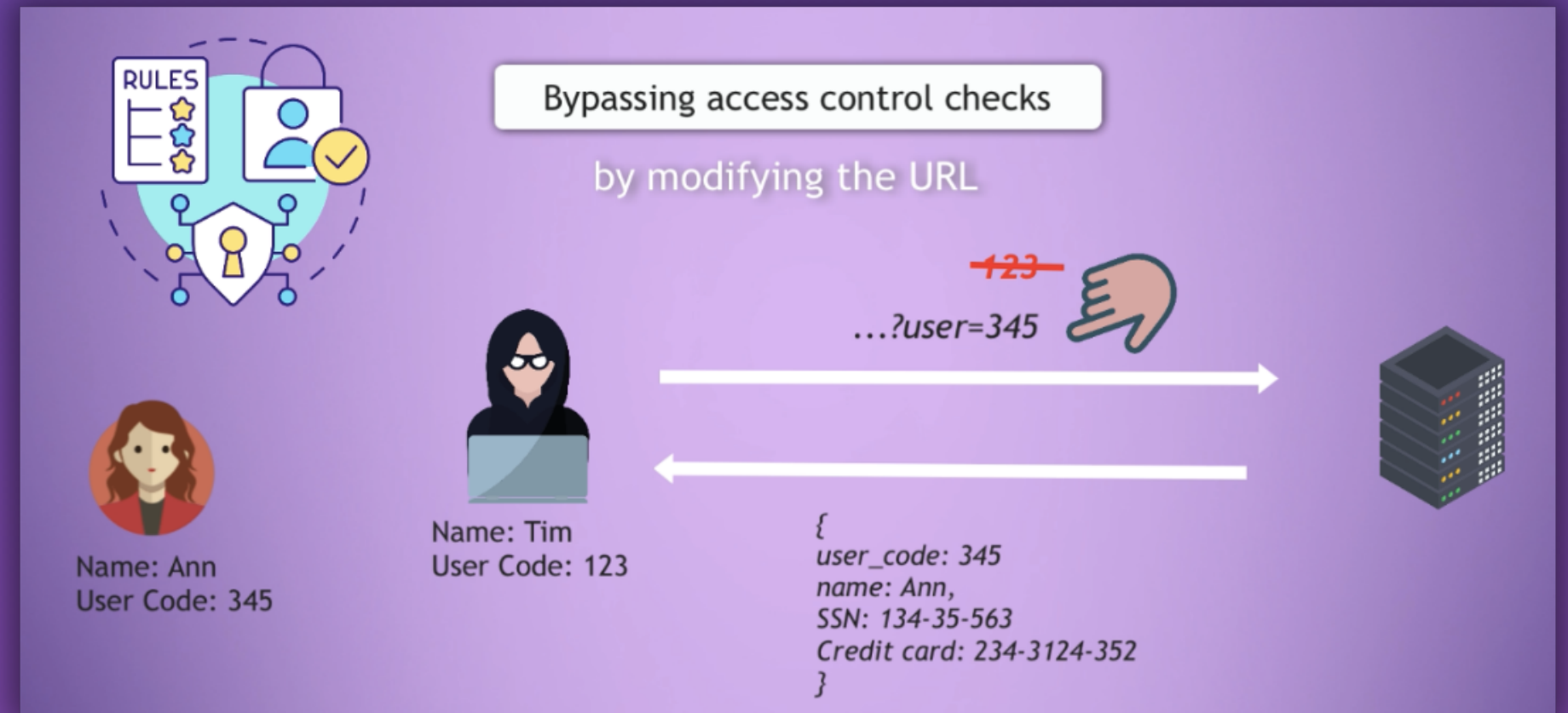
- Companies should adopt this document to minimize these risks
- Regularly updated to reflect emerging security trends

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery (SSRF)



# Broken Access Control

- Access control enforces policies so that users cannot act outside of their intended permissions
- Failure of that **leads to unauthorized access**



## Common vulnerabilities of this category include

- Violation of least privilege principle
- Bypassing access control checks by modifying the URL
- Fails to detect request forgery
- Returning unauthorized data
- Phishing attacks, session hijacking

# Cryptographic Failures

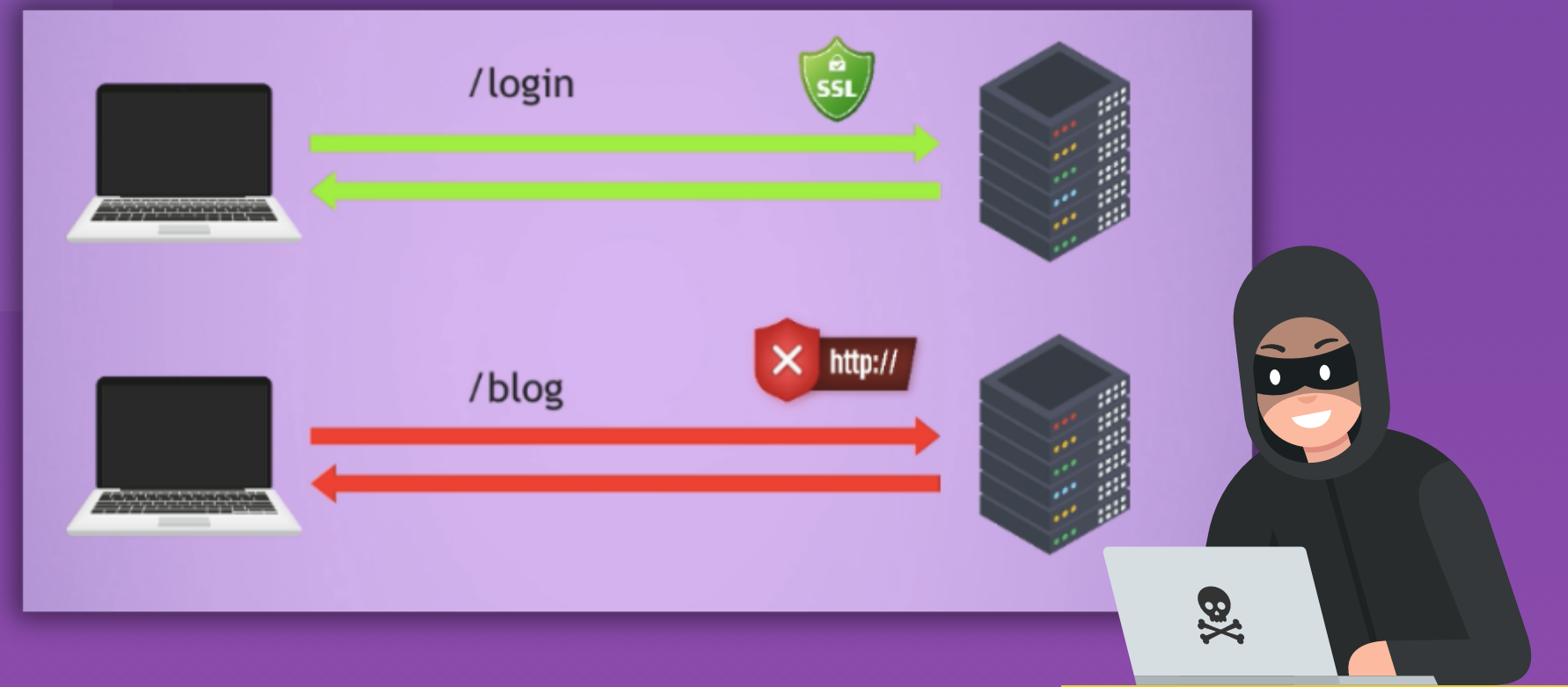


## Common vulnerabilities of this category include

- Lack of or weak cryptography
- Hard-coded sensitive data
- Use of broken or risky crypto algorithms
- Using insecure protocols
- Protect data in transit and at rest

## Leads to

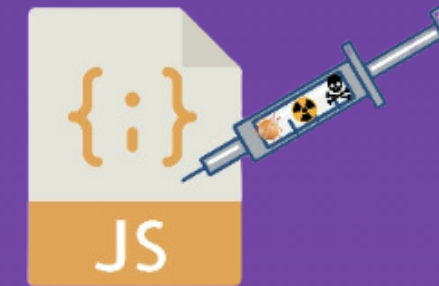
- Sensitive data exposure
- Access to unauthorized resources





# Injection

- Application allows hackers to inject malicious code
- That code can be JavaScript, SQL, NoSQL, OS command templates



Cross-site Scripting

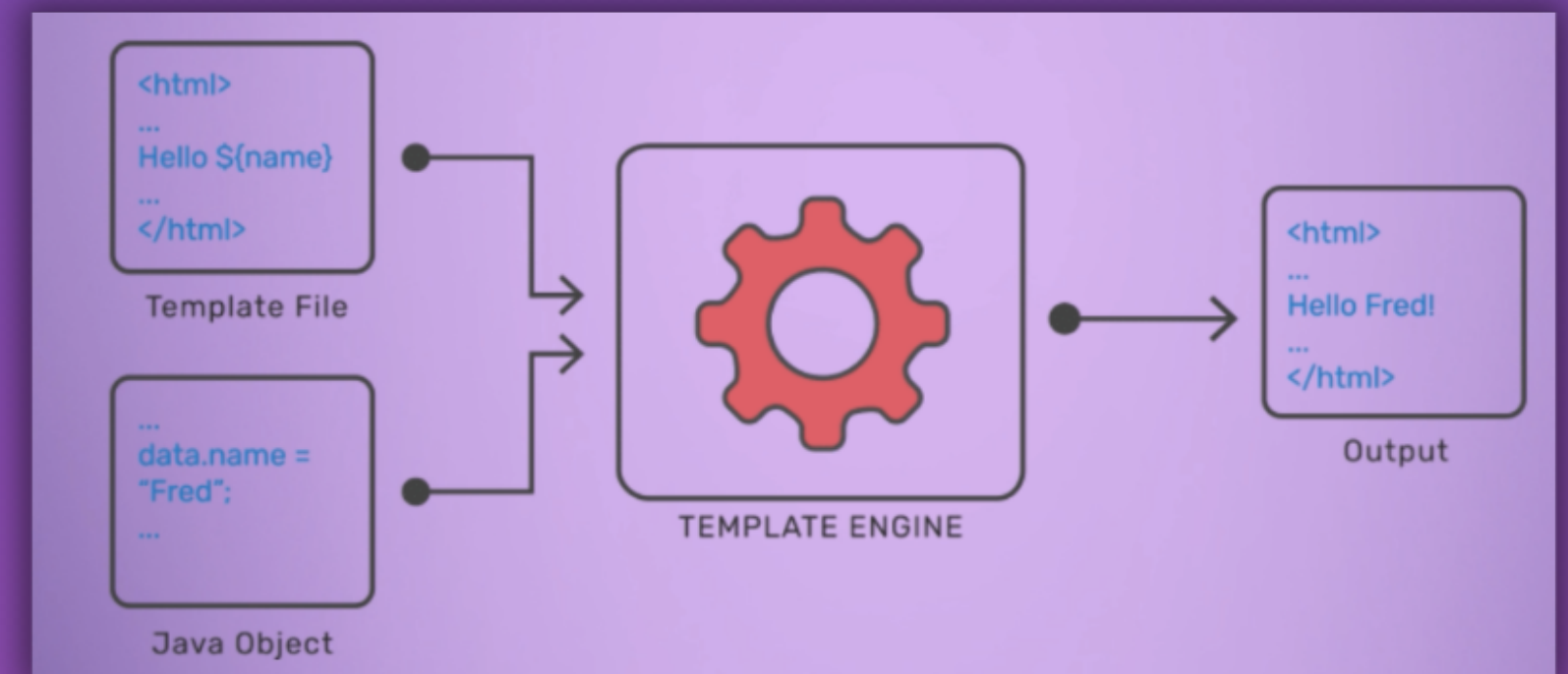


SQL Injection

## How to avoid



- Validate and sanitize user input
- Avoid creating templates from user input
- Always expect malicious user input by default



- Template Injection



# Insecure Design

## Insecure design vs insecure implementation

- Focuses on risks related to design and architectural flaws
- Pre-implementation phase

### A call for more

- ✓ Threat modelling
  - What are possible threats for a specific application, or specific systems
- ✓ Secure design patterns
- ✓ Reference architectures



### Missing or ineffective control design

- Bad credential management
- Bad permission handling
- Insecure infrastructure configuration

# Security Misconfiguration

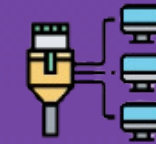
## Some examples

- Improper access configuration
- Unnecessary open ports
- SSH port allowing access from any source
- Debugging features enabled
- Using default accounts & passwords

Storage



Network Config



Application Config

App



Can be on any level

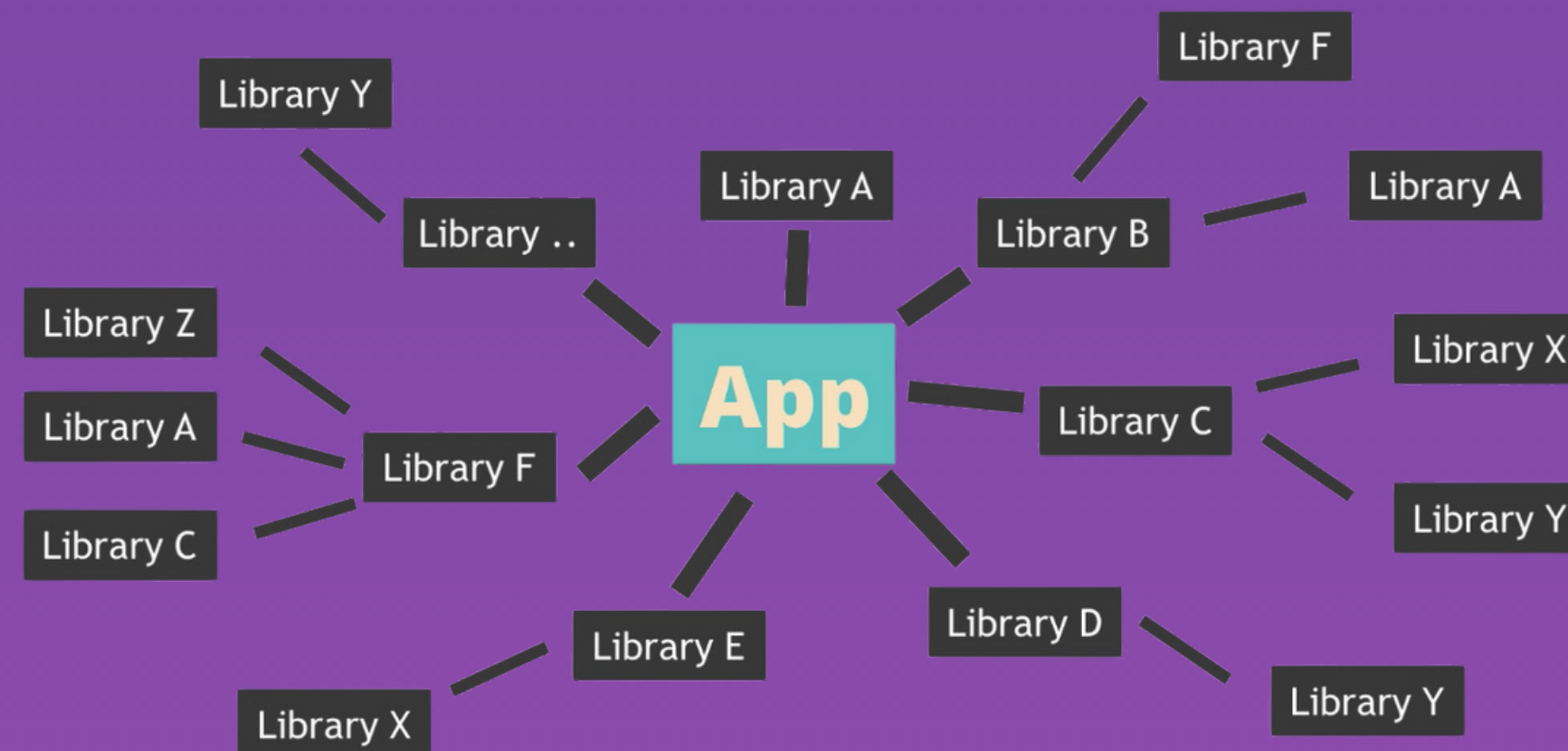
More chances of misconfigurations

Apps have become more **complex** and **highly configurable**



# Vulnerable and Outdated Components

- No difference between your own code and library code, for security threats
- Vulnerability in third-party library affects your whole system
- You are likely vulnerable if:
  - you don't know the versions of all components you use
  - you do not scan for vulnerabilities regularly
  - developers do not test the compatibility of updated, upgraded or patched libraries



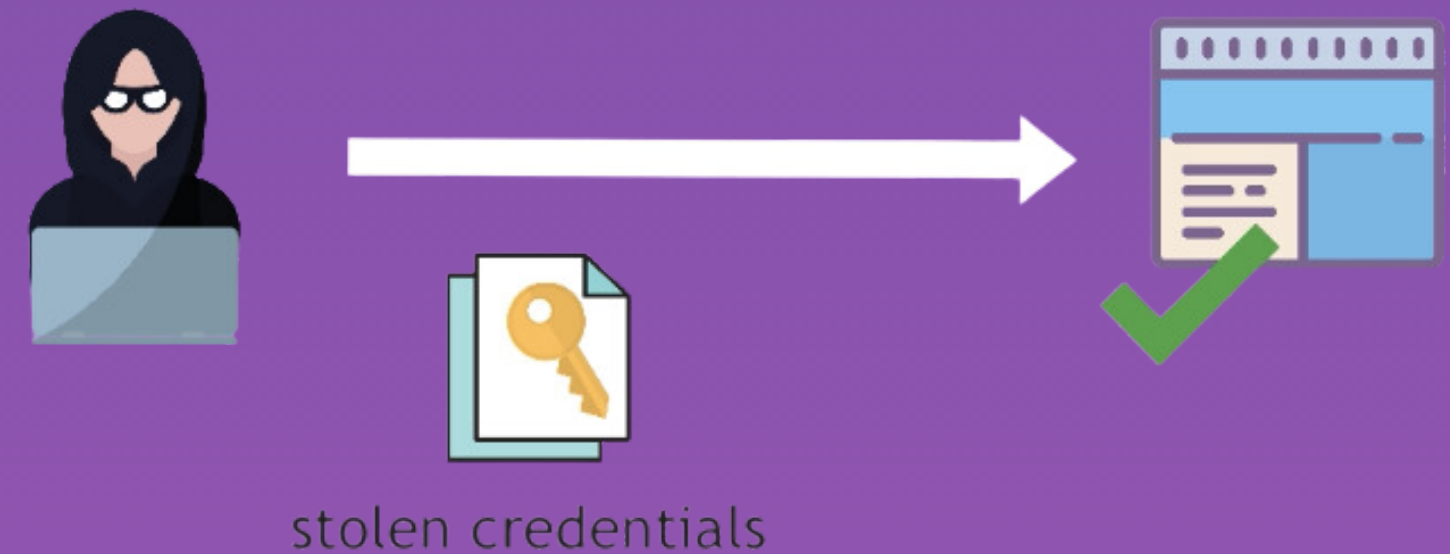
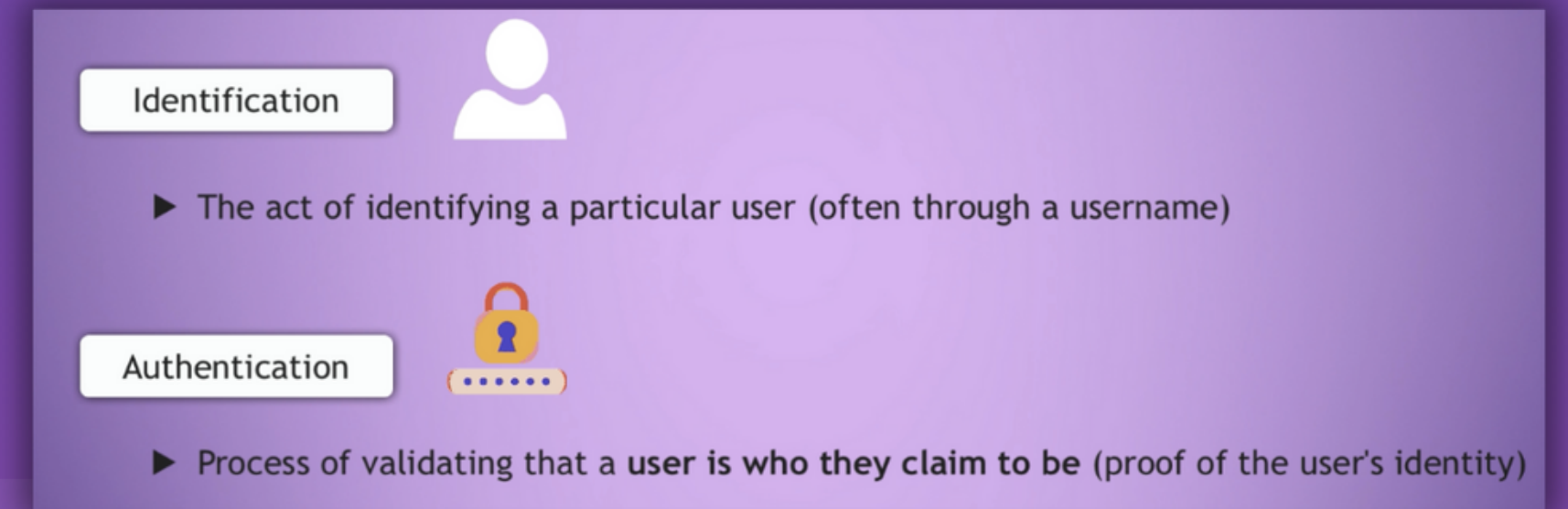


# Identification and Authentication Failures

- Relates to **failure to properly identify and authenticate user**

## Common vulnerabilities of this category include

- Weak confirmation of user identity
- Allow weak passwords
- Missing or ineffective multi-factor authentication
- Weak credential password recovery process
- Plain-text or weakly hashed passwords
- User sessions not properly invalidated





# Software and Data Integrity Failures

- Relate to code and infrastructure that does not protect against integrity violations

## Some examples

- Using libraries & plugins from unverified sources
- Auto-update downloading without integrity verification
- Weak digital signature



web server



web server



web server



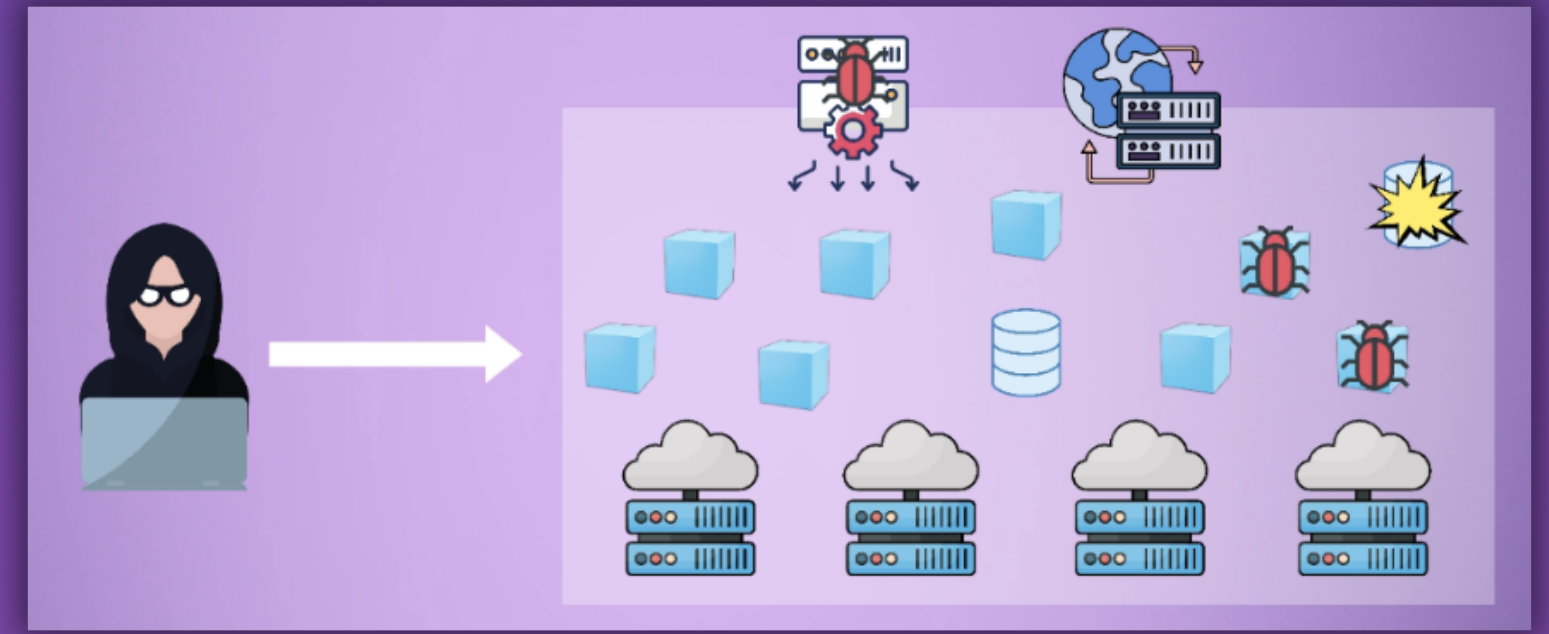
web server





# Security Logging and Monitoring Failures

- Category is to help detect, escalate and respond to active breaches
- Without logging and monitoring, breaches cannot be detected



Logging

Monitoring

Alerting



## Logging and Tracing

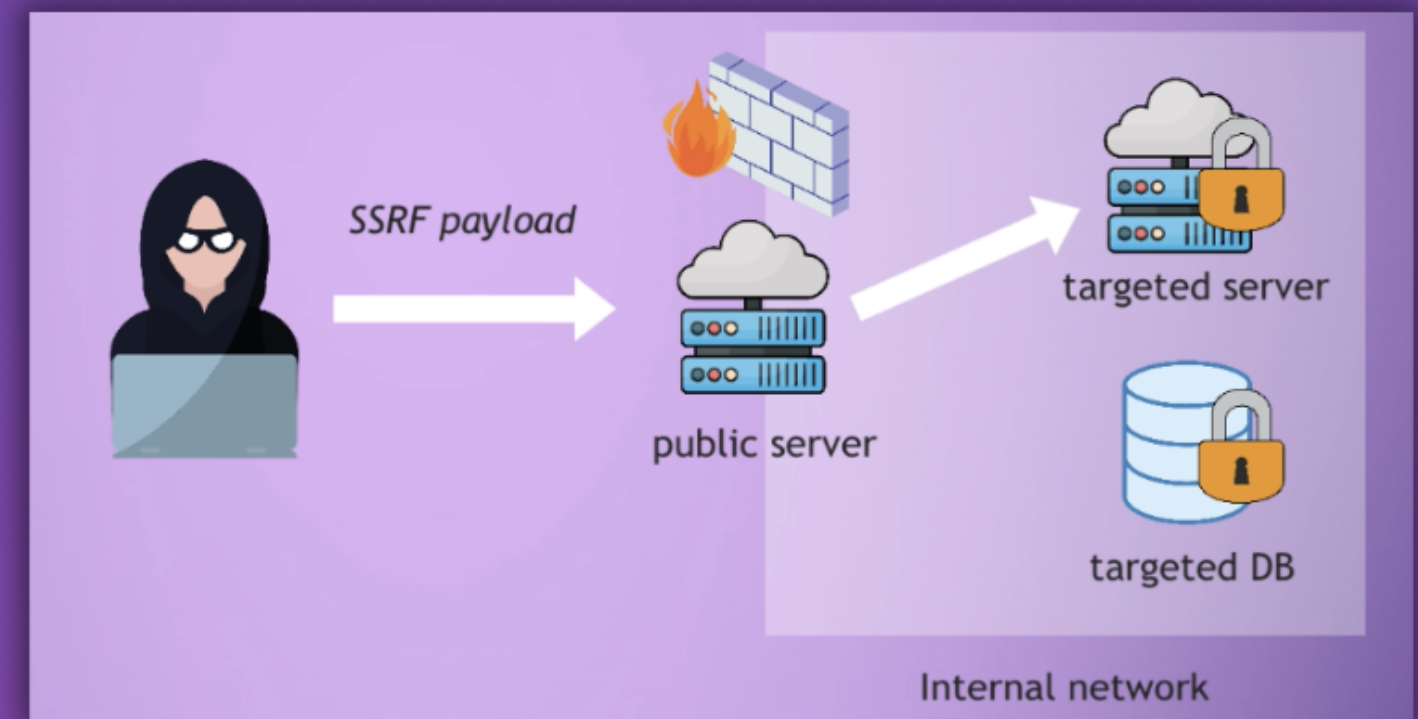
- Without it, we have no insights on system behavior
- Recording events, actions, errors

## Monitoring and Alerting

- Notify on attack attempts
- Without monitoring, you are blind and deaf to attacks
- Enough time for attackers to probe around in systems

# Server-Side Request Forgery (SSRF)

- SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL
- Allows attacker to coerce the application to send a crafted request to an unexpected destination (accessing a protected resource)
- SSRF attacks circumvent firewall, VPN or network access control lists



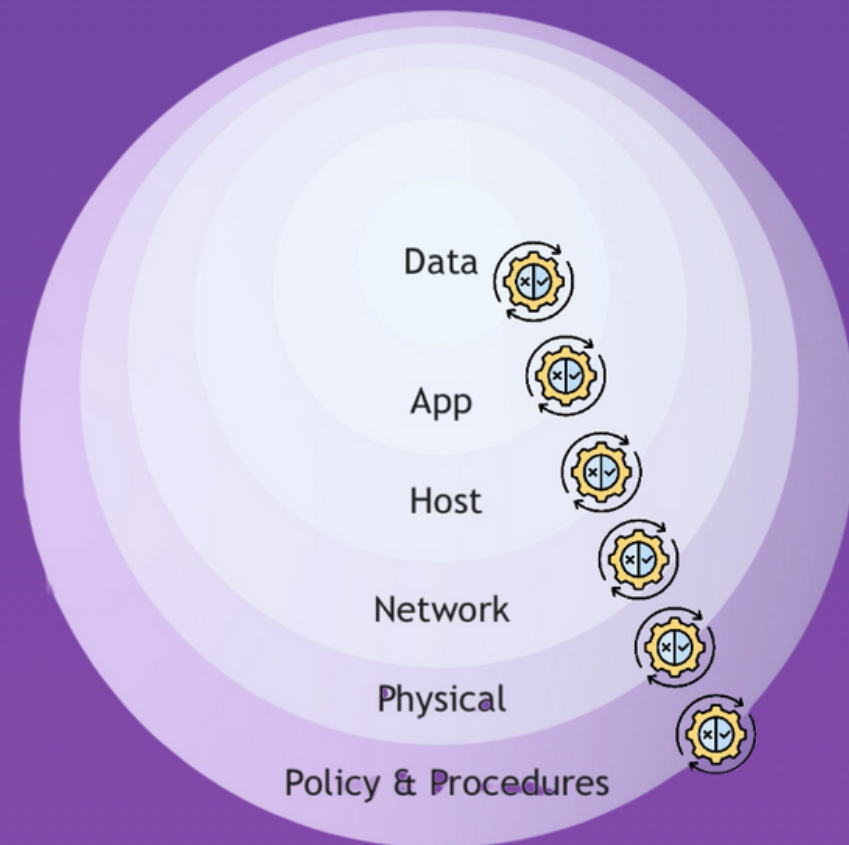
## Attack scenarios

- Port scan internal servers
- Sensitive data exposure
- Access metadata storage



# Security in Layers

# Layered Security



## It includes:

- Access management
- Network security
- Application security
- Logging
- Monitoring

We must secure our system on **multiple layers**



- If attackers gain access to some part of our system, they should not get access to our whole system