fit@hcmus

Software Testing

CSC13003

Security Testing

Content

- What is Security Testing?
- Why to do Security Testing?
- Types of Security Testing
- Security Testing Tools

Content

- **What is Security Testing?**
- Why to do Security Testing?
- Types of Security Testing
- Security Testing Tools

What is Security Testing?

Security testing is a type of non-functional testing that uncovers vulnerabilities, threats, risks in a software application.

What is Security Testing?

- The main goal is to
  - identify assets
  - identify threats and vulnerabilities
  - identify risk
  - perform remediation

What is Security Testing?

- Key principles
  - **Confidentiality** – limiting access to sensitive access managed by a system
  - **Integrity** – ensuring that data is consistent, accurate, and trustworthy throughout its lifecycle and cannot be modified by unauthorized entities
  - **Authentication** – ensuring sensitive systems or data are protected by a mechanism that verifies the identity of the individual accessing them
  - **Authorization** – ensuring sensitive systems or data properly control access for authenticated users according to their roles or permissions
  - **Availability** – ensuring that critical systems or data are available for their users when they are needed
  - **Non-repudiation** – ensures that data sent or received cannot be denied, by exchanging authentication information with a provable time stamp

What is Security Testing?

- Example Security Test Cases
  - Authentication
    - Check password rules—test the password security level and quality required by the site
    - Identify username enumeration vulnerabilities—check if the error differs depending on whether there is a user
    - Test password strength—the minimum requirements to create a password
    - Identify account recovery vulnerabilities—check if attacks can recover accounts (i.e., by changing emails or passwords)
    - Check username strength—ensure usernames are unique
    - Identify fail-open authentication—check if the system provides open access even when authentication fails
    - Verify cookie scoping—check if cookies are scoped to the domain or if attackers can steal them

What is Security Testing?

- Example Security Test Cases
  - Input Validation
    - Fuzz request parameters—check for reflected parameters and open redirection
    - Identify SQL injection vulnerabilities—check if the system handles parameters as SQL
    - Identify SOAP injection vulnerabilities—check if the application responds to SOAP
    - Identify LDAP injection vulnerabilities—test for failure to sanitize inputs
    - Identify XML injection vulnerabilities—check if injected XML impacts the application
    - Identify XXE injection vulnerabilities—check if attackers can inject external entities

What is Security Testing?

- Example Security Test Cases
  - Application and Business Logic
    - Determine the application logic attack surface—what the application does
    - Check data transmission from clients—see if information transfers differ between applications
    - Identify input validation on the client-side—check where the application bases its logic
    - Identify logic flaws in multi-step processes—check if bypassing steps is possible
    - Test incomplete input handling—check if the application processes faulty input
    - Check trust relationships—for example, if users can access admin functions

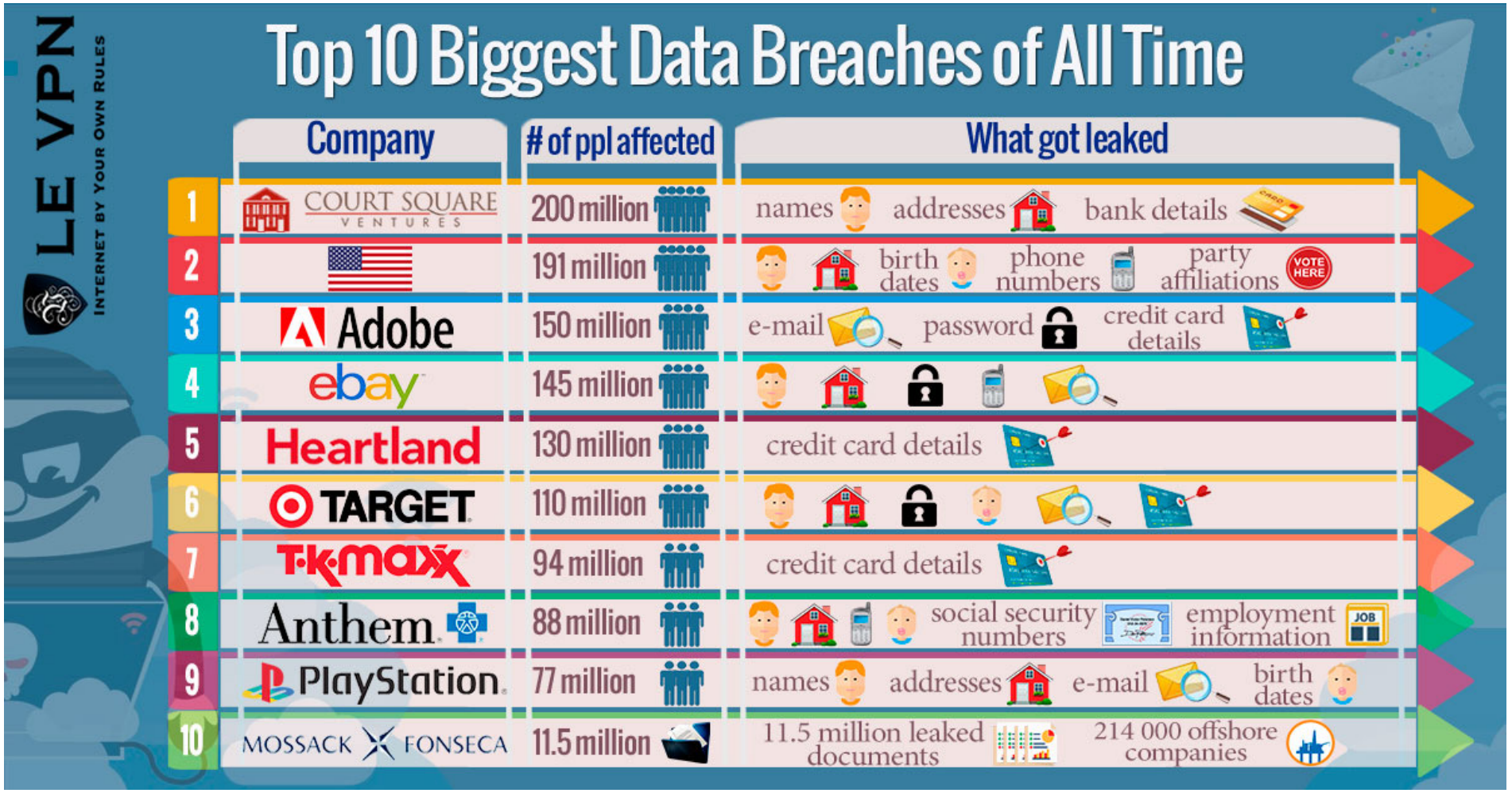
What is Security Testing?

- Example Security Test Cases
  - Other Tests
    - DOM vulnerabilities like XSS
    - Lack of HTTP security headers
    - Local privacy vulnerabilities
    - Weak and persistent cookies
    - Weak SSL ciphers
    - URL parameters containing sensitive information

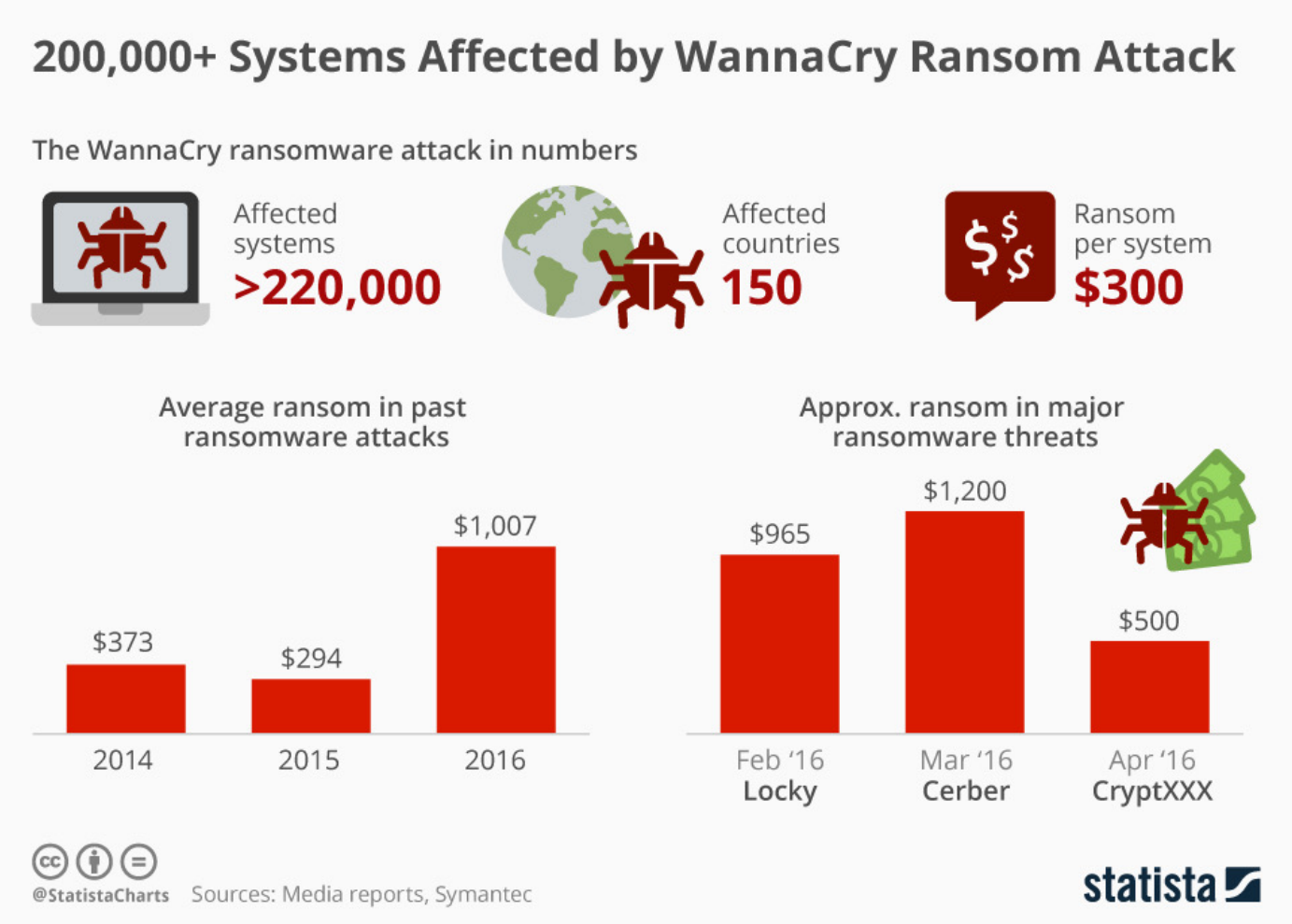
Content

- What is Security Testing?
- **Why to do Security Testing?**
- Types of Security Testing
- Security Testing Tools

Why to do Security Testing?



Why to do Security Testing?



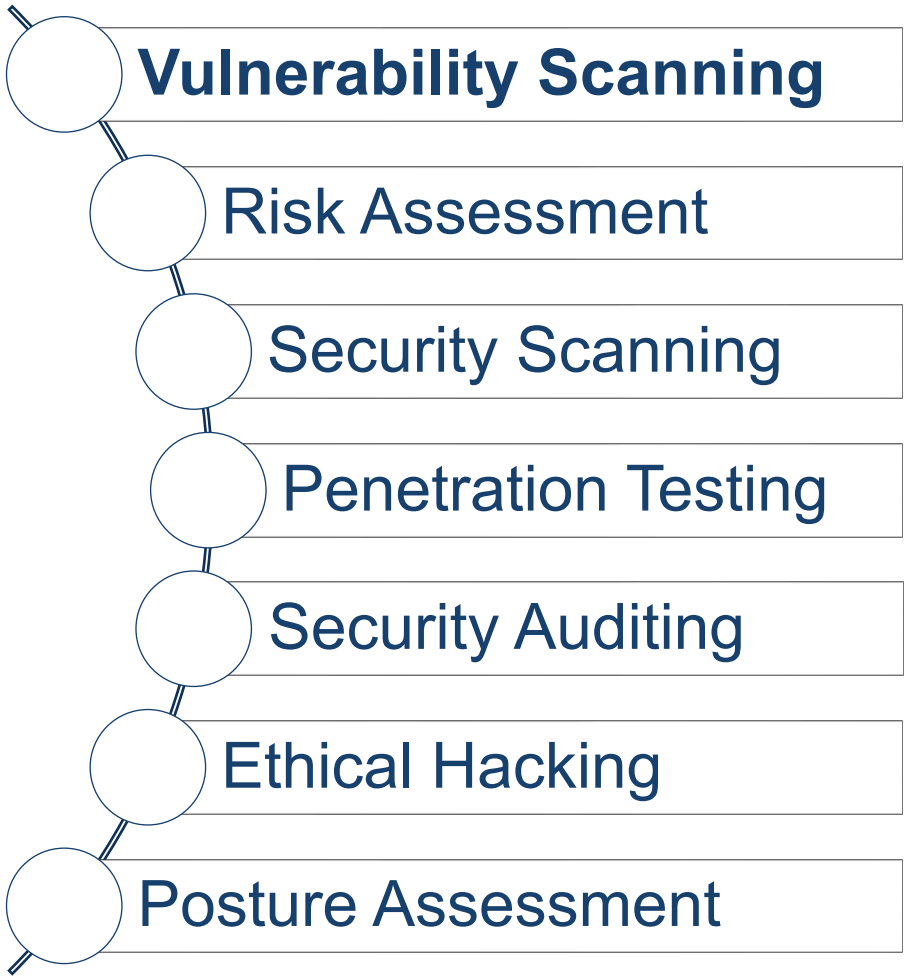
Why to do Security Testing?

- Protecting Sensitive Information
- Preventing Unauthorized Access
- Maintaining Customer Trust
- Compliance with Regulations
- Preventing Financial Loss
- Ensuring Business Continuity
- Adapting to Evolving Threats

Content

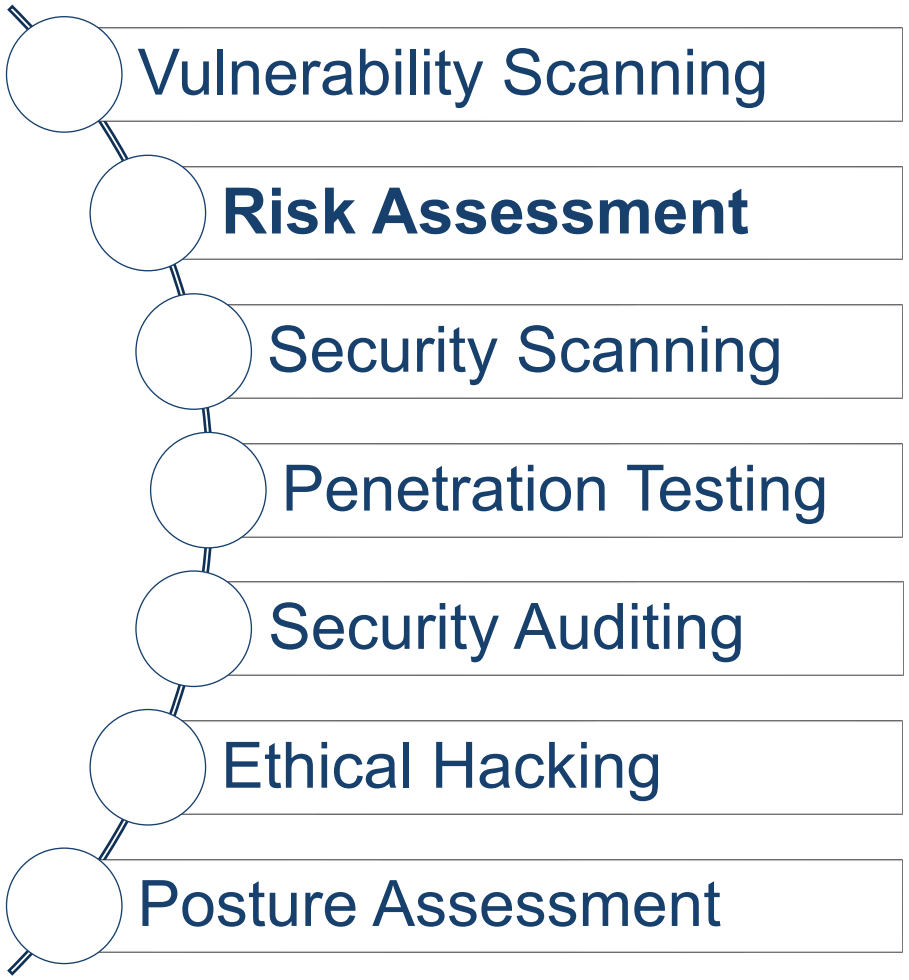
- What is Security Testing?
- Why to do Security Testing?
- **Types of Security Testing**
- Security Testing Tools

Types of Security Testing



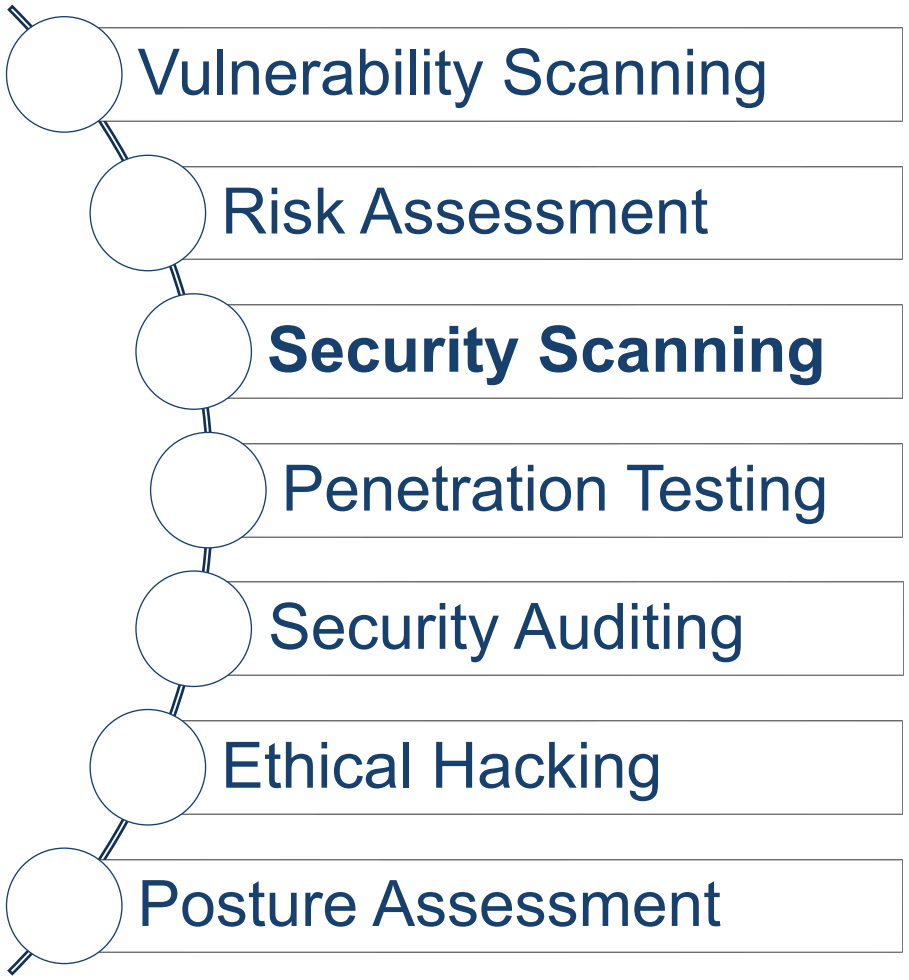
- Involves use of an automated software tool to scan systems against predetermined vulnerabilities

Types of Security Testing



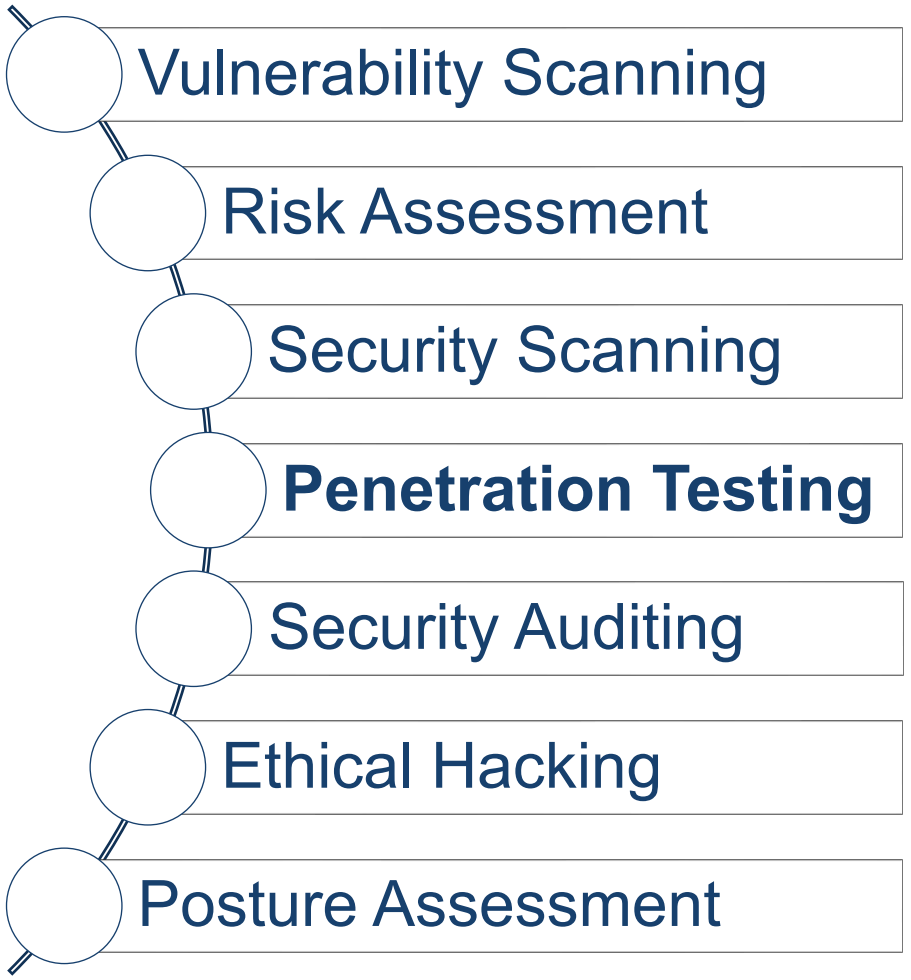
- Consists of an analysis of security risks in the application, software, or network
- Once identified, they are classified as low, medium, high, or critical and mitigation measures can be enacted based on priority

Types of Security Testing



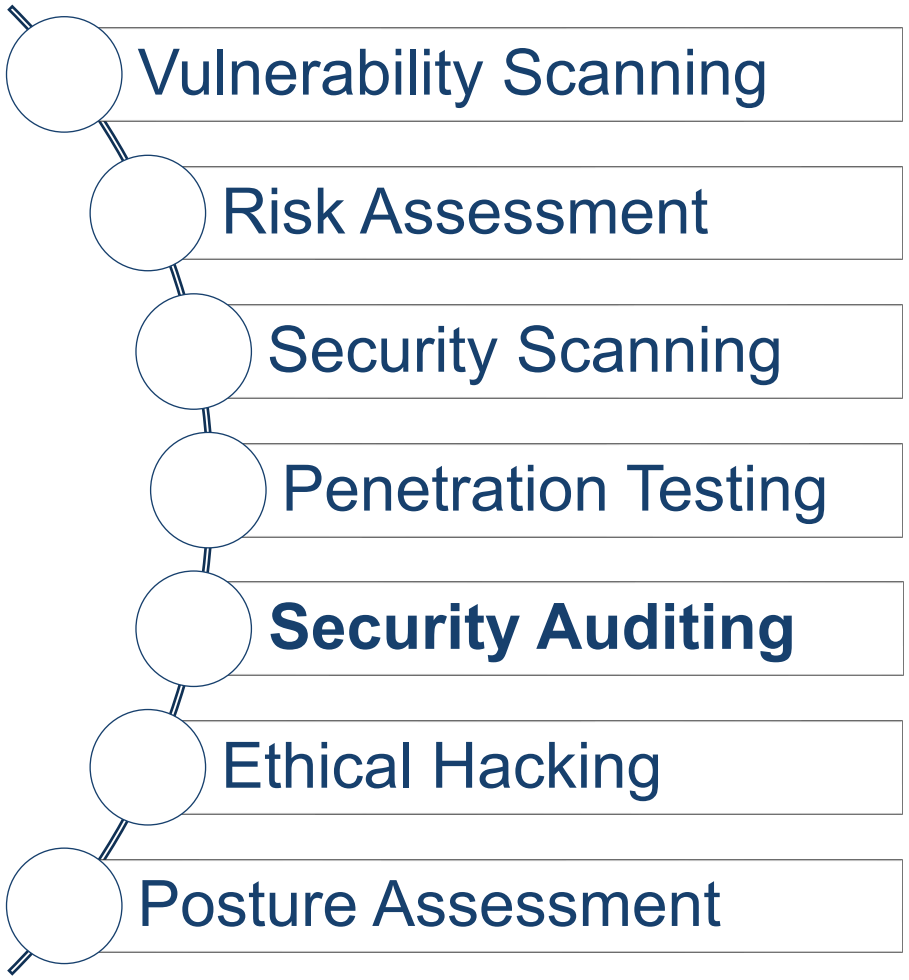
- Can be done with manual or automated testing and serves as a means for locating network or system weaknesses

Types of Security Testing



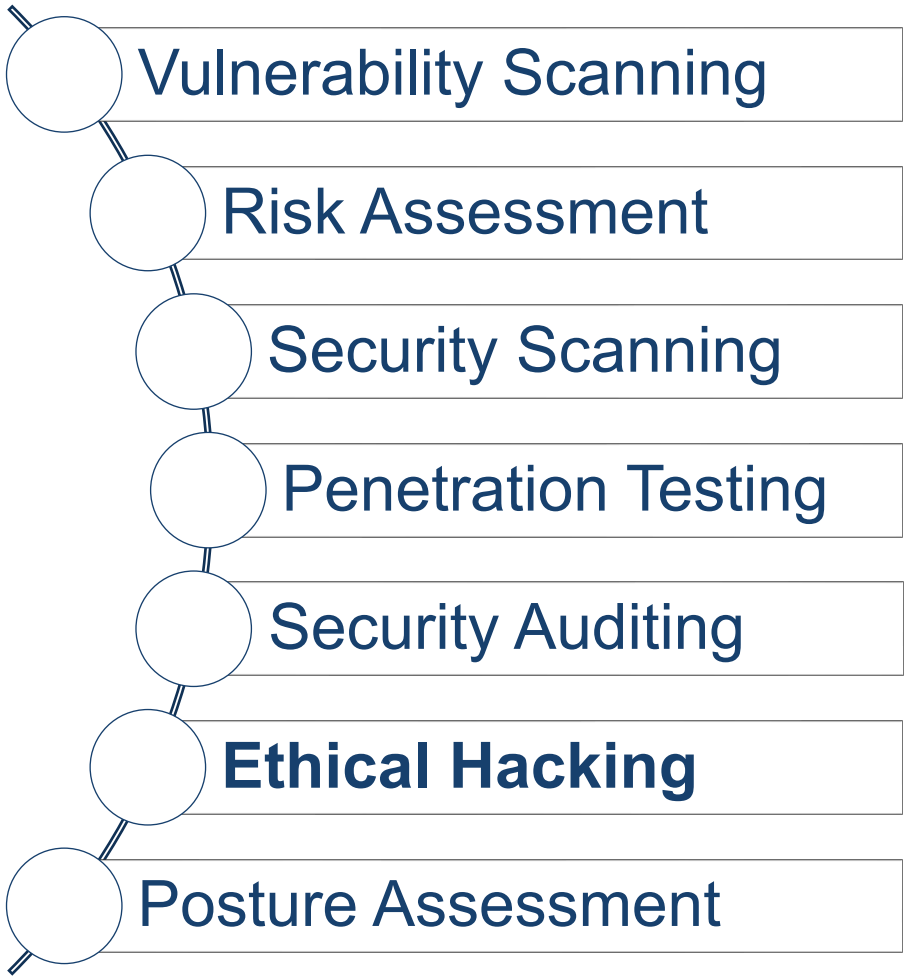
- Simulates an attack from a malicious party or hacker and helps to clearly identify critical vulnerabilities in the system, software, or application

Types of Security Testing



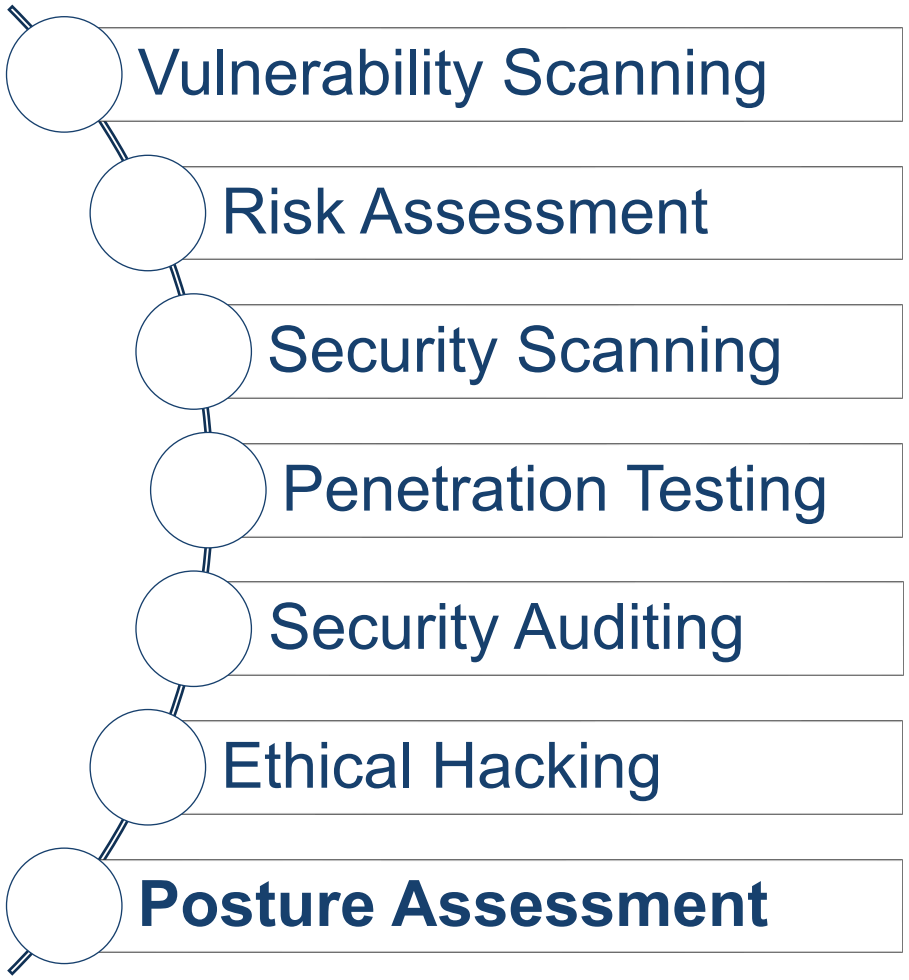
- An internal inspection of all the operating systems and applications with the intent of finding security flaws
- The results from the audit can then be passed to the applicable teams for follow up and correction

Types of Security Testing



- Hired experts attempt to hack into a system or network with the goal of exposing flaws and gaps in the existing security measures

Types of Security Testing



- A combination of ethical hacking, security scanning, and risk assessments to give a snapshot of the overall security within the organization

Content

- What is Security Testing?
- Why to do Security Testing?
- Types of Security Testing
- **Security Testing Tools**

Security Testing Tools

- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Interactive Application Security Testing (IAST)
- Software Composition Analysis (SCA)

Security Testing Tools

- Static Application Security Testing (SAST)
  - assess the source code while at rest
  - identify exploitable flaws
  - detect issues in source code
    - input validation
    - numerical errors
    - path traversals
    - race conditions
  - can also be used on compiled code

Security Testing Tools

- Dynamic Application Security Testing (DAST)
  - examine the application during runtime
  - detect exploitable flaws while running
  - uses fuzzing
    - throw large volumes of known invalid errors and unexpected test cases
    - try to detect conditions during which the application can be exploited
  - check a wide range of components
    - scripting, sessions, data injection, authentication, interfaces, responses, and requests

Security Testing Tools

- Interactive Application Security Testing (IAST)
  - leverage both static and dynamic testing to create a hybrid testing process
  - determine if known source code vulnerabilities are exploitable during runtime
  - reduce the number of false positives
  - combines various testing techniques
    - create multiple advanced attack scenarios
    - using pre-collected information about the data flow and application flow
    - recursively perform dynamic analysis

Security Testing Tools

- Software Composition Analysis (SCA)
  - a technology used to manage and secure open-source components
  - track and analyze the open-source components deployed in projects
  - How?
    - detect all relevant components, libraries, direct and indirect dependencies
    - identify vulnerabilities and suggest remediation


Q
A