Playbook Methods Repository

# **Security Review**

Review a software product for security. Use techniques such as code review, dependency audits, static analysis, and penetration testing to assess security risks and potential vulnerabilities.

### Remote Agility: **•** High

### Linked Tactic(s): Technical Validation

## Why we do it:

Today, businesses not only deploy web-based payroll systems, shopping malls, banking, and stock trading software; they are also supplied as products.

This implies that clients and users have developed a sense of trust in online applications due to its critical feature called SECURITY. Without a doubt, that security component is critical for desktop applications as well. If an online system is unable to safeguard transaction data, no one will consider using it.

The primary objective of security testing is to discover threats inside the system and to quantify its possible vulnerabilities, ensuring that threats can be detected and the system does not cease to operate or is not exploited. Additionally, it assists in identifying any potential security concerns inside the system and assisting developers in resolving them through code.

## 

## When to apply it:

* Xx

## Best Practices & Considerations:

* Shift Security Testing Left: Organizations are incorporating security practises earlier in the development process as a result of the shift to DevSecOps. Shifting security testing to the left may assist developers in identifying security vulnerabilities and implementing security best practises when developing software. Additionally, it may assist testers in identifying security concerns prior to the programme going into production.
* Automate and Test Often: While it is important to perform manual security testing, such as full penetration tests or security audits, organisations must automate security testing and perform it frequently
* Static Analysis: Static analysis enables you to fully inspect all aspects of the software's source code when it is not in use. This will assist you in identifying any possible back door or vulnerability in your software application that would make it susceptible to prospective assaults. Static analysis assists in finding potential vulnerabilities that the developer may have overlooked during the code review process.
* Dynamic Analysis (Penetration Testing): After static analysis, dynamic analysis is performed. It is carried out in a runtime context while the application is running. Dynamic analysis assists in identifying possible problems that may have gone unnoticed during static analysis.
* Determine the level of data protection: Your data security is contingent upon data storage, usefulness, and visibility. Excellent security testing procedures are required to ensure that user data is always safeguarded.
* Test Access Points: Open access might attract unwelcome intrusion. To avoid such assaults, you must verify your application's access points.
* Test Error Handling: Testing for error codes is critical. This includes 400, 404, and 408 errors, among others.
* Test Session Management: Management of testing sessions is an essential need. Session on a website refers to the response transactions between the user's browser and the website's server.
* Test for the Malicious Script: SQL and XSS injection are both techniques used by hackers to compromise a website. Testing for malicious scripts might assist you in protecting your website from such attacks.
* Conduct Additional Functionality Testing: Payments and file uploads are two more services that need testing. These functionalities need exhaustive and specialised testing techniques.

## Responsible roles:

* xx

## Tools:

### Online tools/platforms/services

* + NetSparker, ImmuniWeb, Vega, Wapiti, W3af, SQLMap

### Websites

* + xx

### Databases

* + xx
* Other
  + xx

## 

## Thoughtworks Examples - Linked

### Client working docs, airtable, miro/mural boards

* + xx

### Client polished presentations/deliverables

* + xx

### Internal assets - clinic materials / guild docs

* + xx

## 

## Learn more: How we do this?

### Templates (docs, decks, sheets, miro, etc.)

* + xx

### How-To Resources (external or internal)

* + <https://www.softwaretestinghelp.com/how-to-test-application-security-web-and-desktop-application-security-testing-techniques/>
  + <https://www.guru99.com/what-is-security-testing.html>
  + <https://www.testrigtechnologies.com/learn-10-best-practices-for-security-testing/>
  + <https://brightsec.com/blog/security-testing/#security-testing-best-practices>
  + <https://www.cigniti.com/blog/10-open-source-web-security-testing-tools/>

### Outside References (articles, books, etc.)

* + xx

### Sub-set Activities

* + xx

## 