Тестування, верифікація та валідація програмного забезпечення

Лекція №8

Тема: Тестування безпеки програмного забезпечення

ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Лектор: асистент кафедри комп'ютерних наук Українець Микола Олександрович

Питання лекції

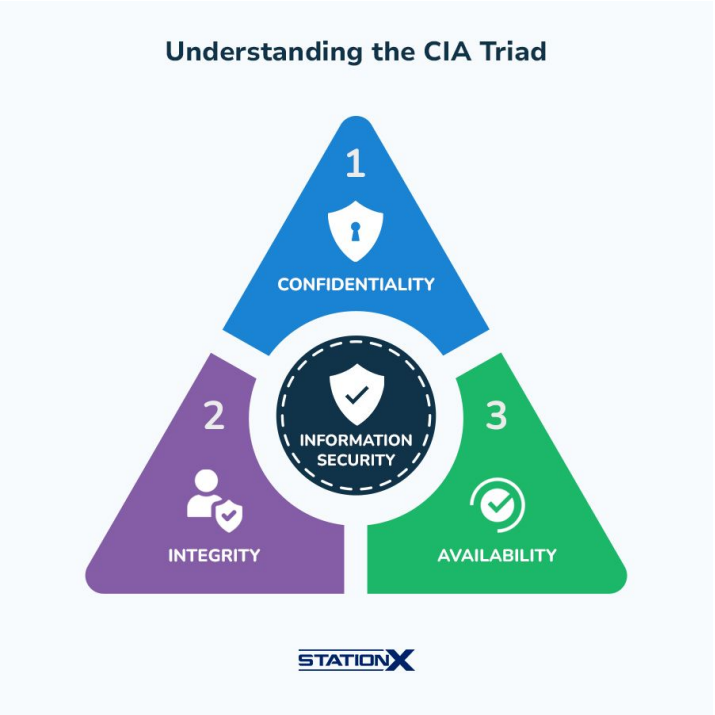
1. Основні принципи інформаційної безпеки
2. Методи тестування безпеки
3. Типові вразливості

Основні принципи інформаційної безпеки

Тріада CIA (Конфіденційність, Цілісність, Доступність) — це фундаментальна модель інформаційної безпеки. Вона слугує основою для розробки організаціями політик безпеки.

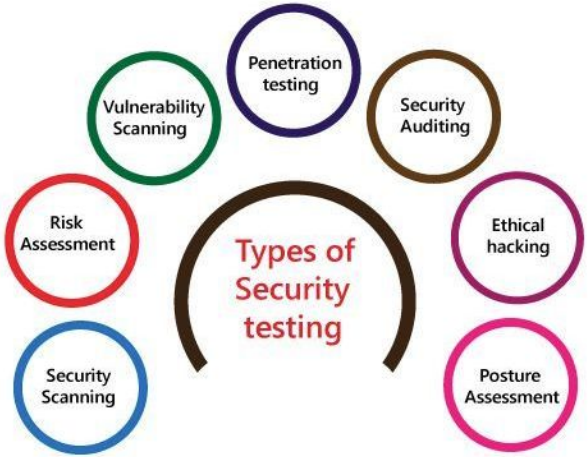
Основні принципи інформаційної безпеки:

- Confidentiality (Конфіденційність) — дані доступні лише тим, хто має дозвіл.
- Integrity (Цілісність) — дані не можуть бути змінені без дозволу.
- Availability (Доступність) — система має працювати, коли користувач її потребує.



Основні принципи інформаційної безпеки

- **Vulnerability Scanning -** Автоматизований процес пошуку відомих вразливостей у системі.
- **Security Auditing** - Комплексна перевірка відповідності системи вимогам безпеки.
- **Penetration Testing / Pentest** - Імітація реальної атаки на систему з метою виявлення вразливостей, які важко знайти автоматичними засобами.
- **Risk Assessment** - Процес ідентифікації та аналізу, який зосереджений на запобіганні дефектам і вразливостям у безпеці програмного забезпечення.
- **Security Scanning** - Використання автоматизованих інструментів для перевірки мережі, додатка чи системи на наявність вразливостей та слабких місць, які можуть бути використані зловмисниками.
- **Ethical hacking** - Процес виявлення вразливостей у комп'ютерних системах та мережах з дозволу їхніх власників з метою підвищення безпеки.
- **Posture Assessment** - Комплексна оцінка захисту кібербезпеки організації для виявлення вразливостей, прогалин і областей для вдосконалення



Методи тестування безпеки

За метою тестування

- Penetration Testing
- Vulnerability Assessment
- Security Audit
- Red Teaming
- Blue Team Testing
- Purple Teaming

За глибиною впливу

- Surface Testing
- Deep Testing
- Infrastructure Testing
- Social Engineering Testing

За рівнем обізнаності тестера

- White Box Testing
- Grey Box Testing
- Black Box Testing

За способом виконання

- Ручне
- Автоматизоване

За виконанням системи

- SAST (Static Application Security Testing)
- DAST (Dynamic Application Security Testing)
- IAST (Interactive Application Security Testing)

Типові вразливості

Вразливість	Приклад	Як перевірити
Injection (SQL/Command)	"' OR '1'='1" у полі паролю	Спробувати некоректні запити через Postman
Broken Authentication	Паролі без обмежень спроб	Тестувати brute-force сценарії
Sensitive Data Exposure	Відправка паролів у відкритому вигляді	Перевірити HTTPS та зберігання даних
XXE (XML External Entities)	Неконтрольована обробка XML	Відправити шкідливий XML-запит
Broken Access Control	Користувач бачить чужі дані	Змінити userId у запиті
Security Misconfiguration	Відкриті debug-сторінки	Перевірити /admin, /debug URL
Cross-Site Scripting (XSS)	<script>alert(1)</script>	Ввести у форму HTML-код
Insecure Deserialization	Підробка serialized-даних	Тестування через interception tools
Using Components with Known Vulnerabilities	Застарілі бібліотеки	Перевірка через Dependency Scanner
Insufficient Logging & Monitoring	Відсутність запису спроб зламу	Перевірити журнали подій

Типові вразливості

Authorization

User admin

Pass ‘ OR 100=100 --

id	username	password	email
0	admin	_____	admin@com.ua

Приклад SQL-ін’єкції

```
Select *
From users
Where
   user_name = ‘admin’
   AND password = ‘’
   OR 100=100 --’
```

```
<?php

if(isset($_POST['btnSign']))
{

   $message = trim($_POST['mtxMessage']);
   $name    = trim($_POST['txtName']);

   // Sanitize message input
   $message = stripslashes($message);
   $message = mysql_real_escape_string($message);

   // Sanitize name input
   $name = mysql_real_escape_string($name);

   $query = "INSERT INTO guestbook (comment,name) VALUES
   ('$message','$name');";

   $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>' );

}

?>
```

Приклад XSS-ін’єкції

Приклади інцидентів

- Widespread npm Supply Chain Attack Puts Billions of Weekly Downloads at Risk
- Spill the Tea: “Tea” the #1 Women’s Dating App Faces Major Data Breach — Private Data Publicly Exposed
- Cloudflare DDoSed itself with React useEffect hook blunder

Корисні посилання

1. https://www.testrail.com/blog/performance-testing-types/
2. https://blog.sentry.io/whats-the-difference-between-api-latency-and-api-response-time/
3. https://training.qatestlab.com/blog/technical-articles/performance-testing/

Дякую за увагу!