Applied!

Data & Network Security

Behnam Amiri

ans.dailysec.ir

aNetSec.github.io

Spring 2025

Web Security

Website hacking

- Websites are public and easy access targets.
- Websites have few developers.
- There are many tools for web hacking.

Web Attacks

- Common web attacks
  - Brute Force
  - XSS
  - CSRF
  - RFI, LFI
  - SQLi
  - …

DVWA

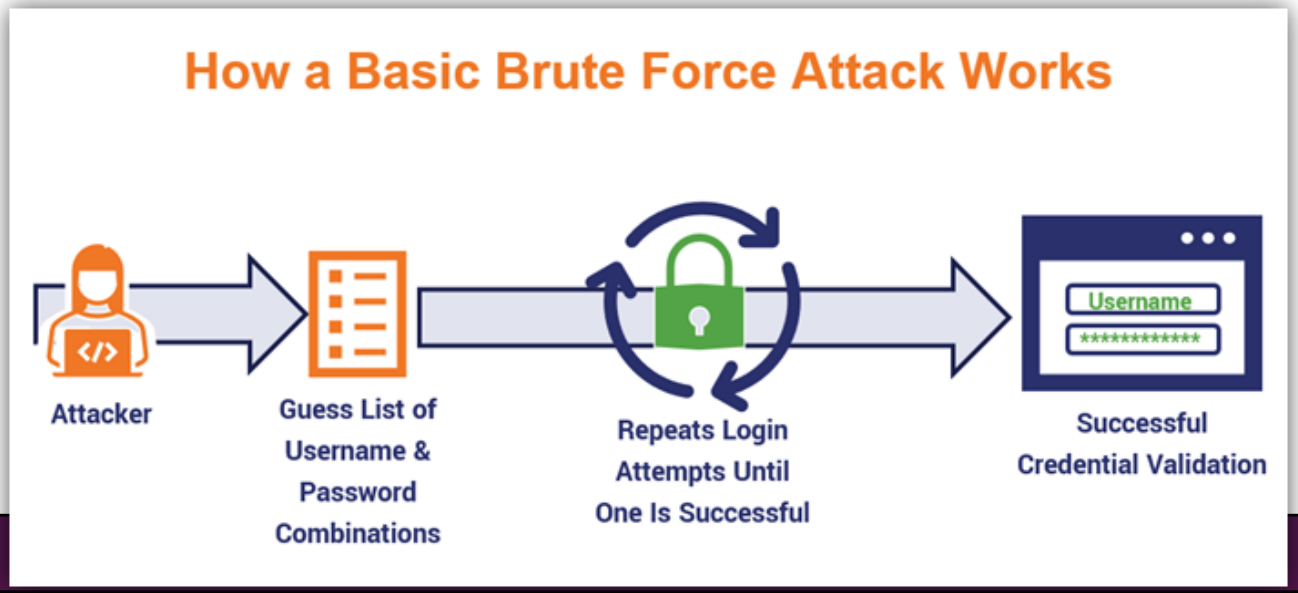
- Damn Vulnerable Web Application (DVWA)
- is a PHP/MariaDB web application that is damn vulnerable.
- Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment.
- https://github.com/digininja/DVWA

Webgoat

- WebGoat is a insecure application that allows you to test vulnerabilities commonly found in web.
- https://owasp.org/www-project-webgoat/

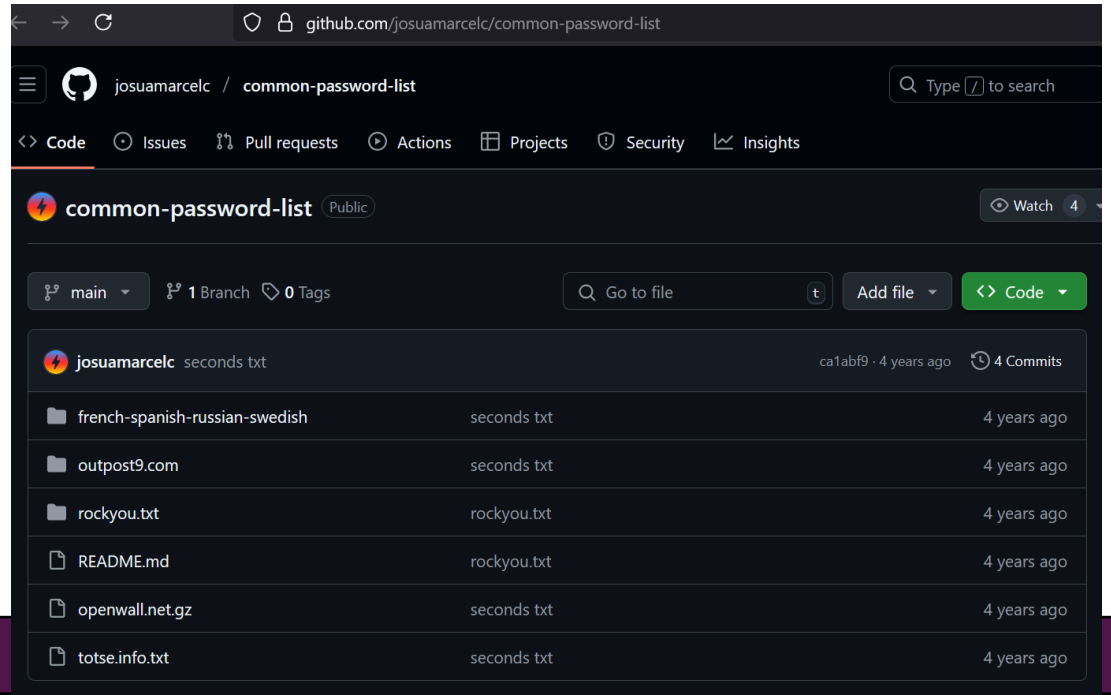
Brute Force

- Brute Force Attack
- a brute force attack is a method used to gain unauthorized access to a system by systematically trying all possible combinations of passwords until the correct one is found.
- This method can be time-consuming.



Dictionary Attack

- is a method to break passwords or encryption keys.
- Unlike a brute force attack, which tries every possible combination of characters, a dictionary attack uses a pre-defined list of potential passwords, often derived from common words, phrases, or previously leaked passwords.

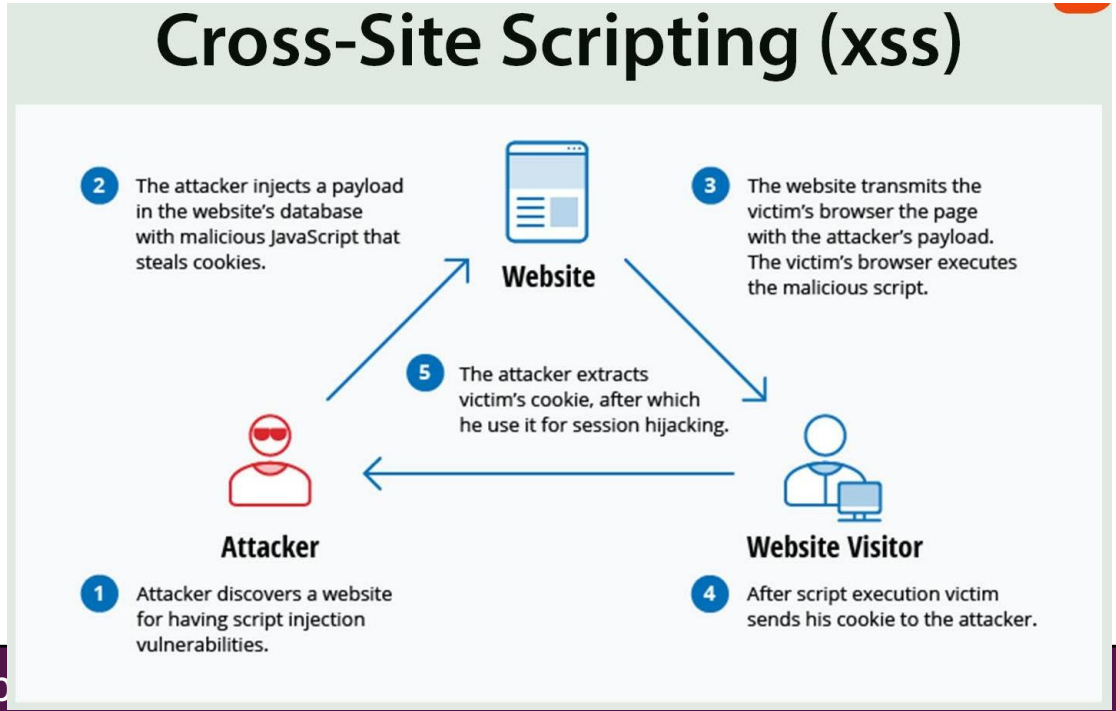


Brute Force Mitigation

- Captcha
- 2FA
- Rate limit
- …

XSS attacks

- Cross-Site Scripting (XSS)
- Session and some useful information stored in cookies.
- If attacker can run JavaScript on victim browser he/she can access cookies.
- XXS runs java script code.

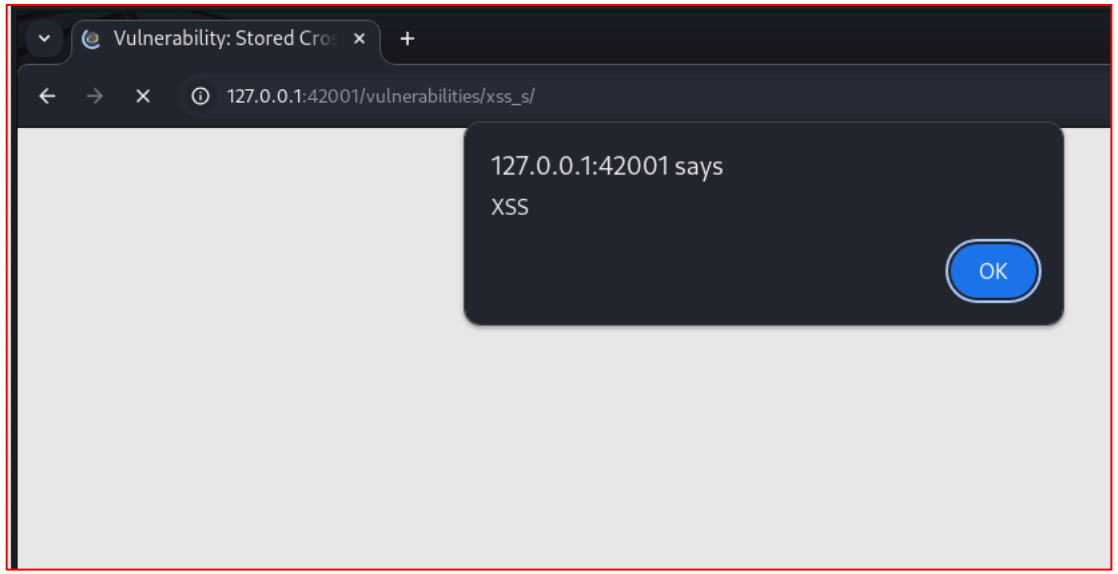


XSS Types

- **Stored XSS (Persistent XSS)**:
  - In this type of attack, the malicious script is stored on the server (e.g., in a database, message forum, or comment field) and is served to users when they access the affected page.
- **Reflected XSS (Non-Persistent XSS)**:
  - In reflected XSS, the malicious script is not stored but is reflected off a web server.
  - The attack typically occurs when a user clicks on a specially crafted link that includes the malicious script as part of the URL or form submission.
- **DOM-based XSS**:
  - In DOM-based XSS, the vulnerability exists in the client-side code rather than the server-side.
  - The attack occurs when the client-side JavaScript modifies the Document Object Model (DOM) and executes the injected code.

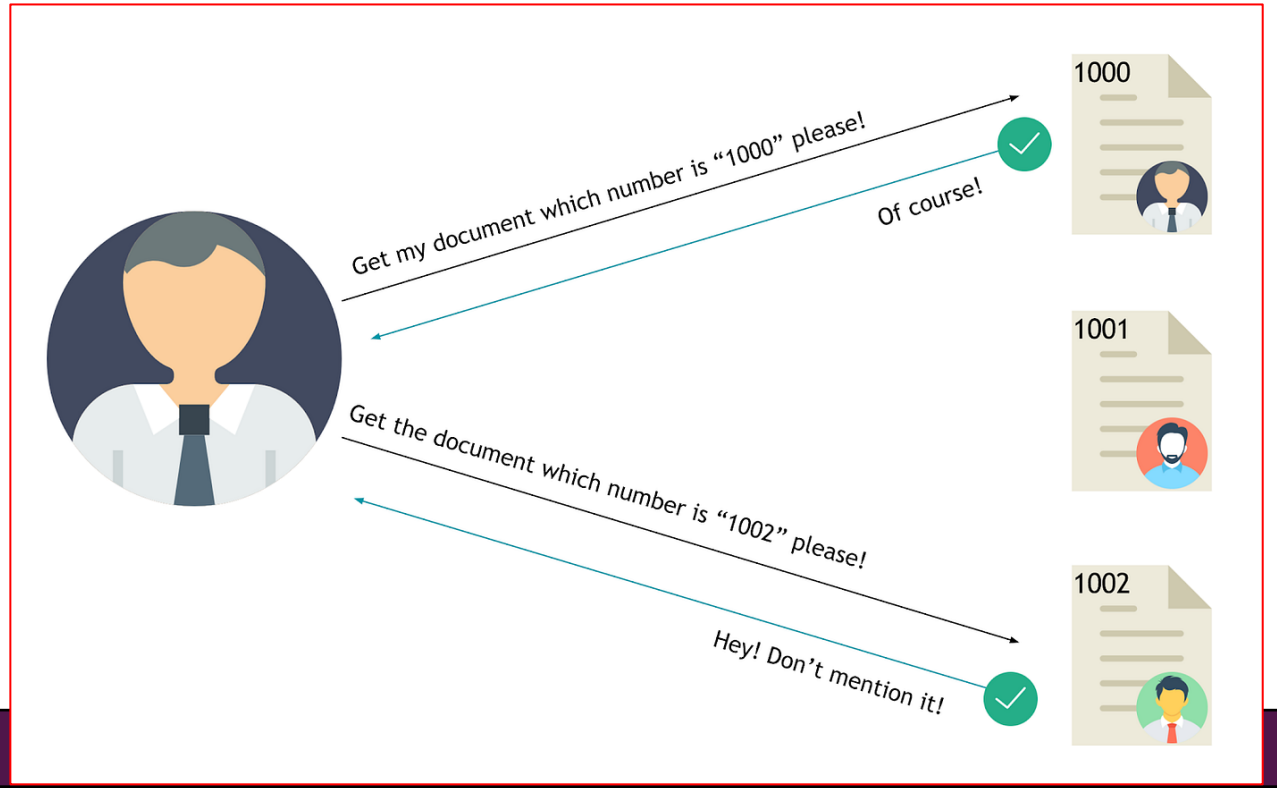
XSS Demo

- DVWA: Vulnerability: Stored Cross Site Scripting (XSS)
  - Set security to low
  - http://127.0.0.1:42001/vulnerabilities/xss_s/
  - <script>alert('XSS');</script>



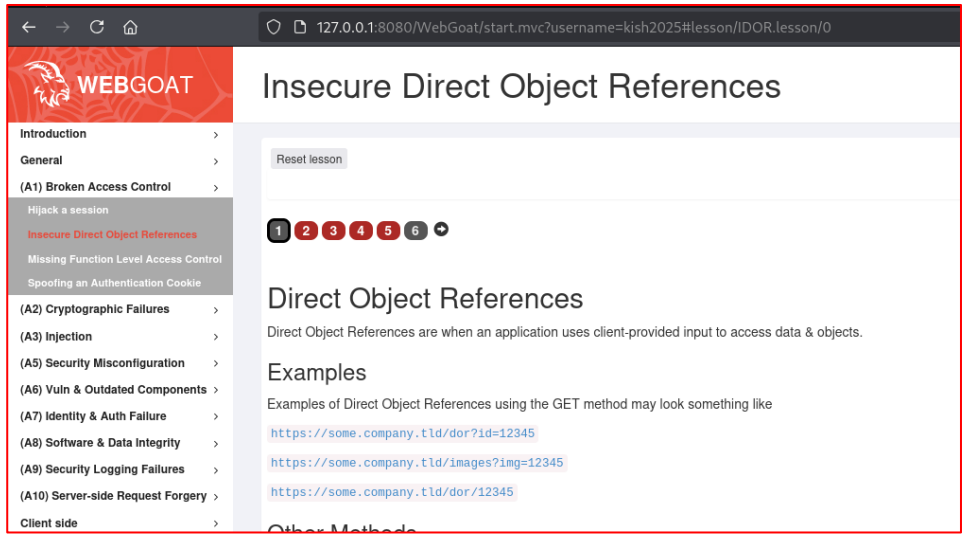
idor

- Insecure direct object references (IDOR)
- IDOR are a type of access control vulnerability that arises when an application uses user-supplied input to access objects directly.
- Example: https://insecure-website.com/account?customer_number=132355



idor Demo

- Start point:
  - http://127.0.0.1:8080/WebGoat/start.mvc?username=kish2025#lesson/IDOR.lesson/0
  - WebGoat/IDOR/profile/2342384
  - GET /WebGoat/IDOR/profile/UserID HTTP/1.1



- src:
  - https://thehackerish.com/idor-tutorial-hands-on-owasp-top-10-training/

SQLi

- SQL Injection (SQLi)
- is a type of security vulnerability that allows an attacker to interfere to application database.
- It occurs when an application includes untrusted data in a SQL query without proper validation or escaping, allowing attackers to manipulate the query and execute arbitrary SQL code.

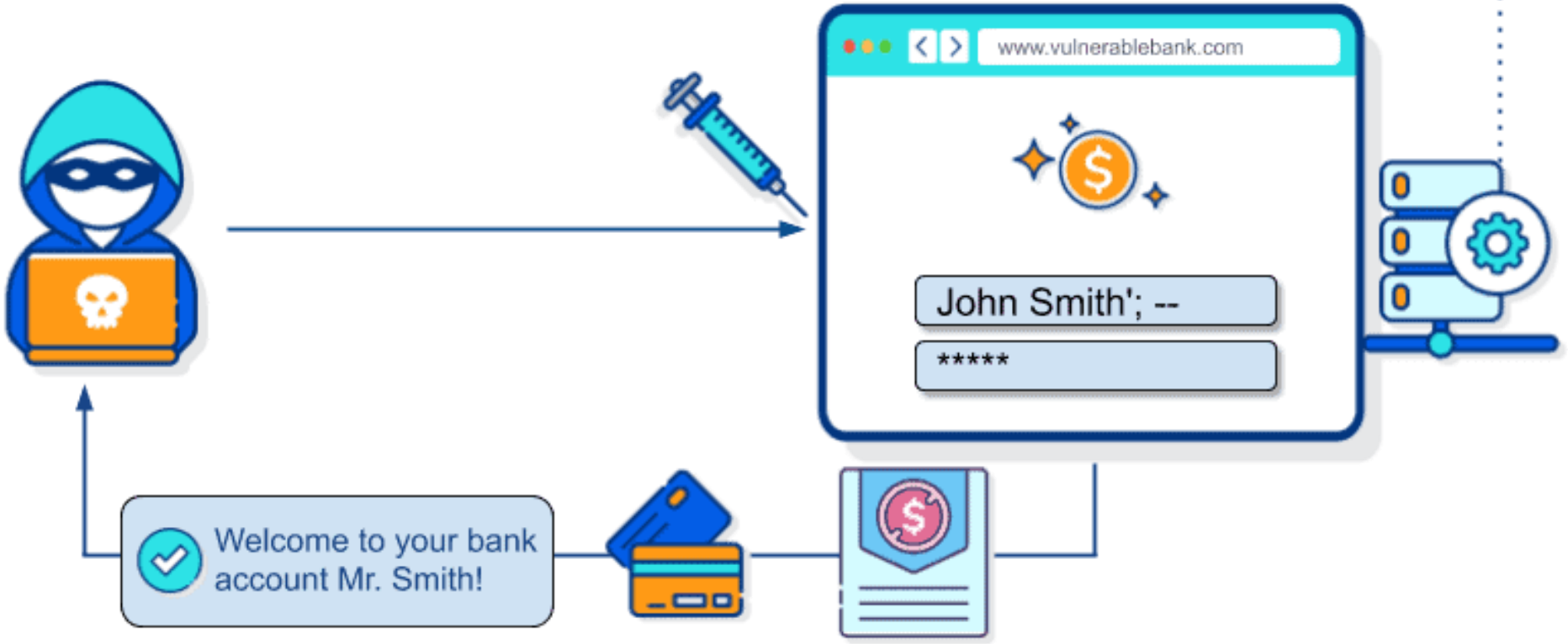
SQLi Types

- **Error-Based SQL Injection**
  - The attacker see application errors.

- **Blind SQL Injection**
- The attacker cannot see the results of the query directly but can infer information based on the application's behavior.
  - eg: Time-Based Blind SQL Injection

SQLi

```
SELECT * FROM users WHERE name='John Smith'; --' and password='wrong'
```



SQLi Demo

- DVWA: Vulnerability: SQL Injection
  - http://127.0.0.1:42001/vulnerabilities/sqli/
  - Normal id: 1,2,3, …
  - Use single quote ‘ for test
  - SELECT first_name, last_name FROM users WHERE user_id = ‘$id’”;
  - 1' OR '0'=‘0
  - 1 ' or 0=0 union select null, version() #
  - 1' or 0=0 union select null, user() #

Web vulnerability scanners

- sqlmap
- Accunetix
- HP Webinspect
- Nessus
- Burp Suit
- Zaproxy
- ...

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3.4.44#dev}
|_ -| . [,]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent i
s illegal. It is the end user's responsibility to obey all applicable local, state and fed
eral laws. Developers assume no liability and are not responsible for any misuse or damage
 caused by this program

[*] starting @ 10:44:53 /2019-04-30/

[10:44:54] [INFO] testing connection to the target URL
[10:44:54] [INFO] heuristics detected web page charset 'ascii'
[10:44:54] [INFO] checking if the target is protected by some kind of WAF/IPS
[10:44:54] [INFO] testing if the target URL content is stable
[10:44:55] [INFO] target URL content is stable
[10:44:55] [INFO] testing if GET parameter 'id' is dynamic
[10:44:55] [INFO] GET parameter 'id' appears to be dynamic
[10:44:55] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
 (possible DBMS: 'MySQL')
```

More training

- https://application.security/

WAF

- Firewall can’t detect and block web attacks.
- We must use **Web Application Firewall (WAF)**
- WAF is a security solution designed to protect web applications by monitoring, filtering, and analyzing HTTP traffic between a web application and the internet.
- WAFs are specifically tailored to defend against various types of attacks that target web applications, such as SQL injection, cross-site scripting (XSS), and other vulnerabilities.
- **Mode Security** is a free and common WAF.

Pentest

Penetration test

- How we find vulnerabilities in a web site?
- Which attacks we must consider?
- Is my knowledge enough?
- What is start and end point?
- **Penetration Test**, often referred to as a **pen test**.
- is a simulated cyber attack against a computer system, network, or web application to identify vulnerabilities that an attacker could exploit.
- The goal of a penetration test is to evaluate the security of the system and provide insights into how to improve its defenses.

Pentest methodology

- Penetration testing (pentesting) methodologies provide a structured approach to conducting security assessments.
- These methodologies outline the steps and best practices that testers should follow to ensure thorough and effective testing.
- The **Web Security Testing Guide (WSTG)** is a comprehensive resource developed by the Open Web Application Security Project (OWASP) that provides a framework for testing the security of web applications.
- https://owasp.org/www-project-web-security-testing-guide/

OWASP

- The **Open Web Application Security Project (OWASP)**.
- is a nonprofit organization focused on improving the security of software.
- It provides a wealth of resources, tools, and community-driven projects aimed at helping organizations and developers understand and mitigate security risks in web applications.
- https://owasp.org

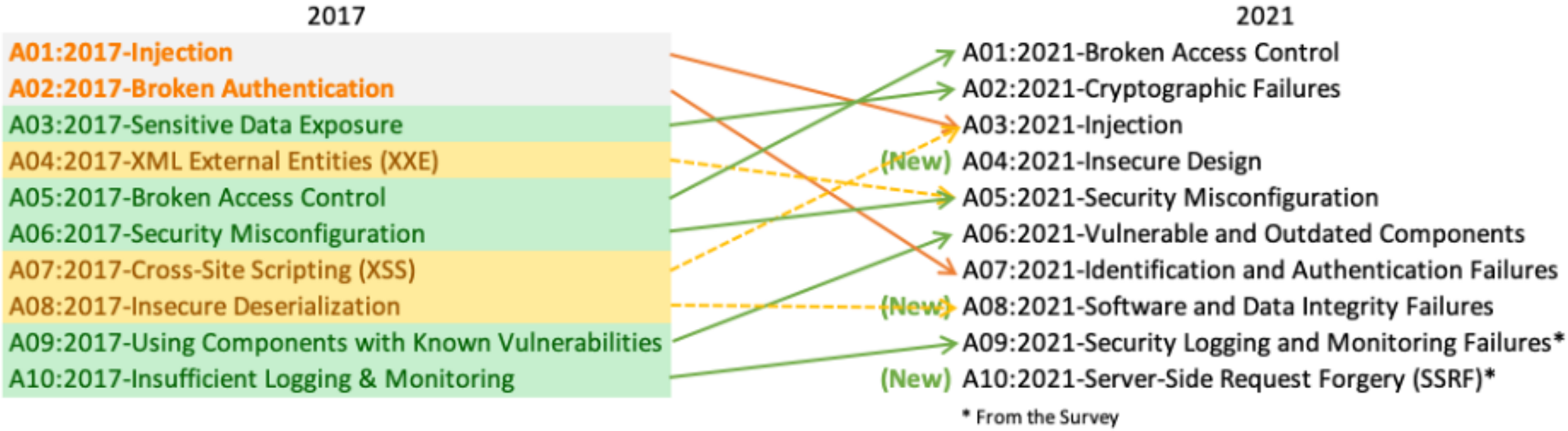
Evaluation

- We find some vulnerability in web application.
- We don’t have enough time and money to fix all of them!
- Which one is important?
- We need evaluation method to find important vulnerabilities.

OWASP top 10

- The **OWASP Top Ten** is a widely recognized list that highlights the ten most critical web application security risks.
- It serves as a foundational resource for organizations, developers, and security professionals to understand and prioritize security vulnerabilities in web applications.
- The list is updated periodically to reflect the evolving threat landscape and emerging vulnerabilities.
- https://owasp.org/www-project-top-ten/

OWASP top 10


2017
A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring
2021
A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New) A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New) A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New) A10:2021-Server-Side Request Forgery (SSRF)*
* From the Survey

OWASP top 10 common mistakes

- It’s not fixed and you can change order based on your web application.
- It’s not Pentest methodology!
- It’s order is sorted based on Risk not Critical vulnerabilities.

Feedback Time