HyperionDev

Welcome to this session:
Web Security Essentials

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators answering questions.

HyperionDev

What is Safeguarding?

Safeguarding refers to actions and measures aimed at protecting the human rights of adults, particularly vulnerable individuals, from abuse, neglect, and harm.

To report a safeguarding concern reach out to us via email:

safeguarding@hyperiondev.com

HyperionDev

Live Lecture Housekeeping:

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
- No question is daft or silly - ask them!
- For all non-academic questions, please submit a query: www.hyperiondev.com/support
- To report a safeguarding concern reach out to us via email: safeguarding@hyperiondev.com
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

HyperionDev

Lecture Overview

- ➔ Discuss Common Web Security Threats
- ➔ Best Practices for Web Application Security
- ➔ Authentication and Authorization Mechanisms
- ➔ Secure Coding and Testing Principles

Learning Outcomes

- Explain and identify common web security threats and their impact.
- Apply best practices to secure web applications.
- Demonstrate knowledge of authentication and authorization protocols.
- Implement secure coding principles and basic security testing methods.

How familiar are you with common threats like XSS or SQL Injection?

A. Beginner
B. Intermediate
C. Advanced

Which of these security practices do you already implement?

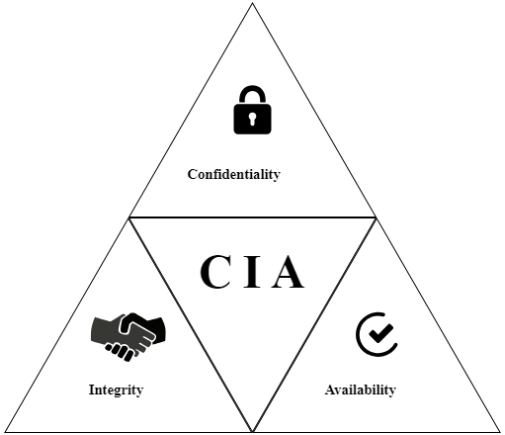
A. Input validation
B. HTTPS
C. Secure authentication

Web Security Essentials

What comes to mind when you think about web security?

CIA Triad

- Confidentiality, Integrity, and Availability of Information
- Guides defending against threats and detecting problems



HyperionDev

Confidentiality

- Protecting against the risk of unauthorised access and leaking of information.
- Includes personal or proprietary information, especially sensitive data related to a person's health or finances.

HyperionDev

Integrity

- Protecting information from unauthorised modification.
- Also involves addressing several social-technical issues.

HyperionDev

Availability

- Preventing unauthorised access that denies illegitimate users from accessing and modifying information.
- Also refers to creating systems that promote security while maintaining efficiency.

HyperionDev

Common Web Security Threats

- **Cross-Site Scripting (XSS):** Attacker injects malicious scripts.
  - Example: Popup stealing user data.
- **Cross-Site Request Forgery (CSRF):** Unauthorized actions on behalf of a user.
  - Example: Automatic transfers in banking apps.
- **SQL Injection:** Injecting malicious SQL code into queries.
  - Example: Accessing entire databases via a login form.

HyperionDev

Injection Attacks

- Injection attacks take advantage of an application's failure either to sanitize or validate user-supplied input.

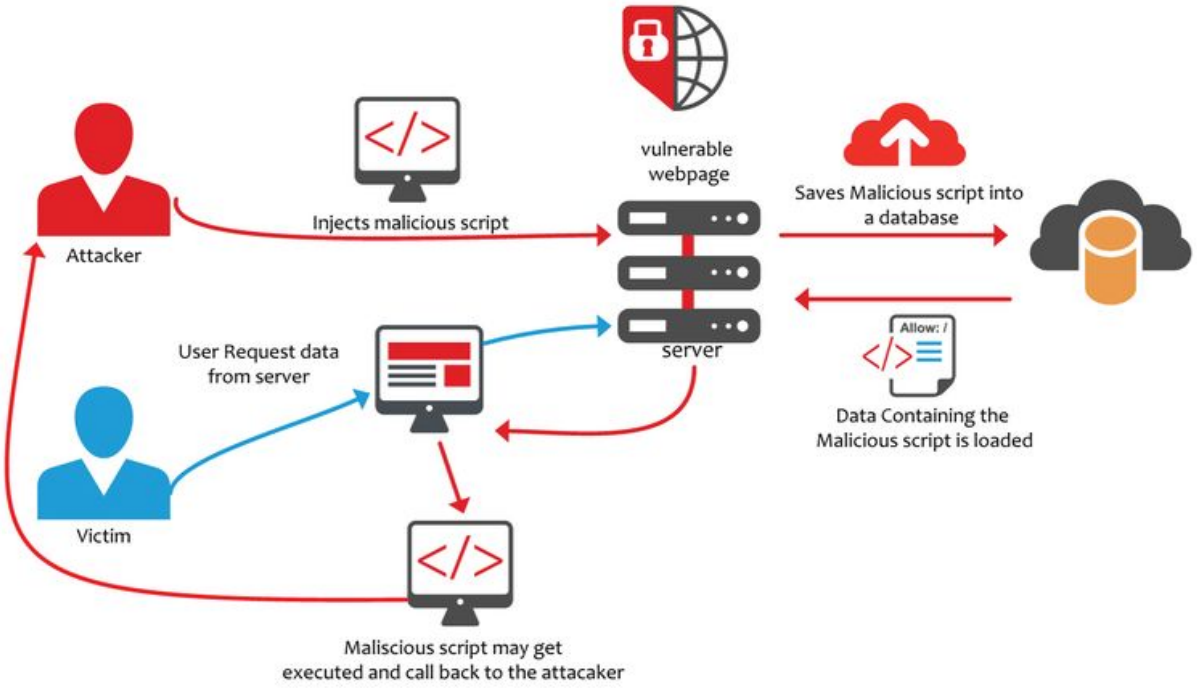
HyperionDev

Types of injection

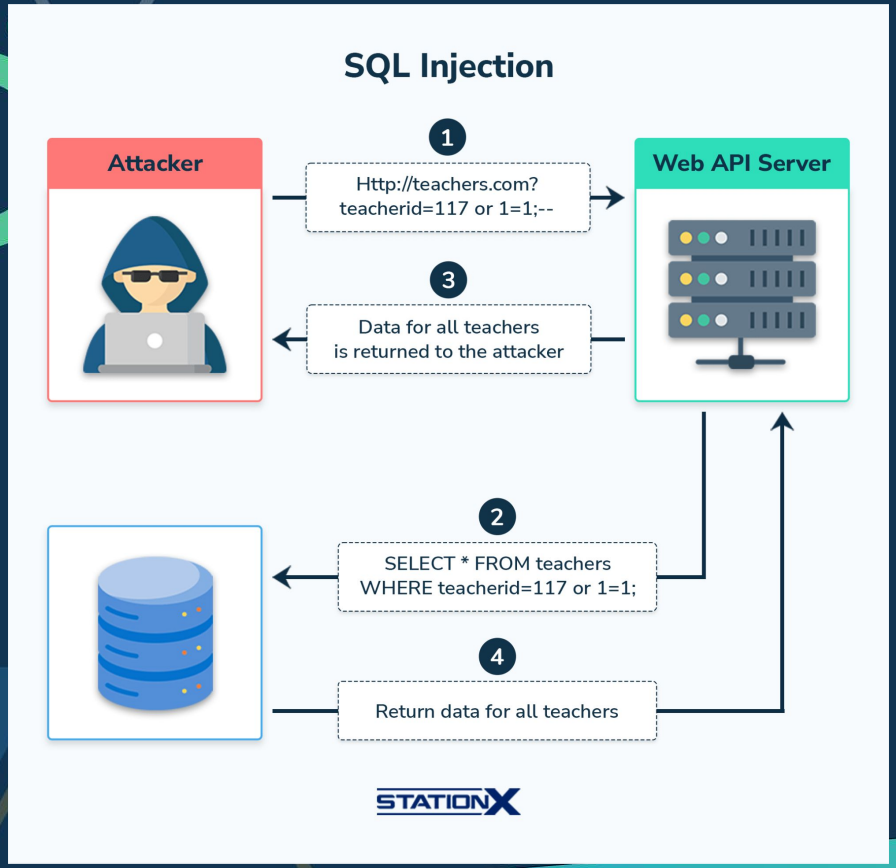
- SQL
- Command
- XML
- NoSQL

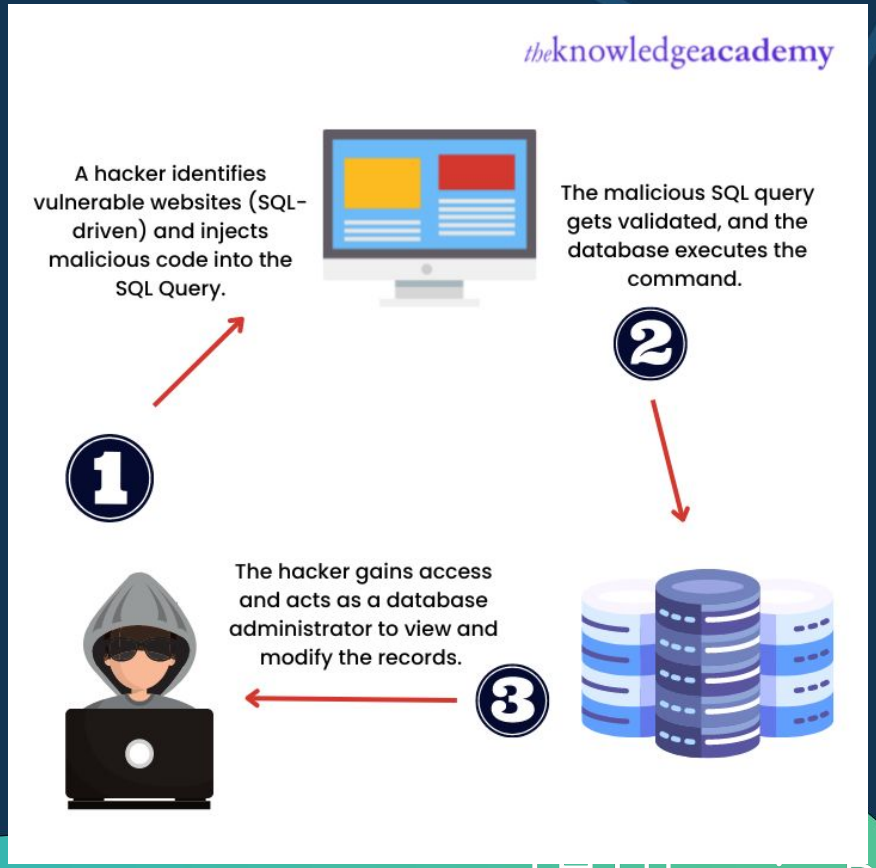
HyperionDev

XSS Attack


vulnerable
webpage
Attacker
Injects malicious script
Saves Malicious script into
a database
server
User Request data
from server
Allow: /
Data Containing the
Malicious script is loaded
Victim
Maliscious script may get
executed and call back to the attacaker

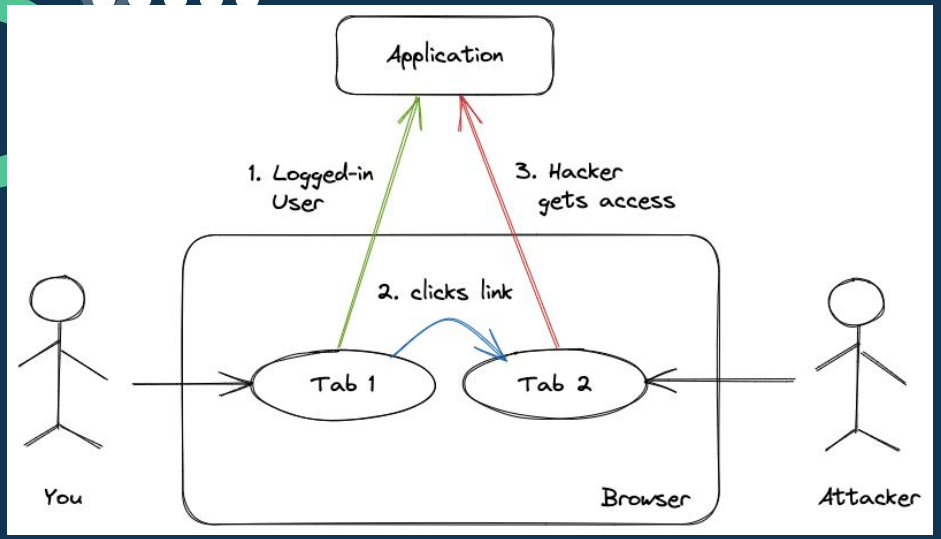
SQL Injection

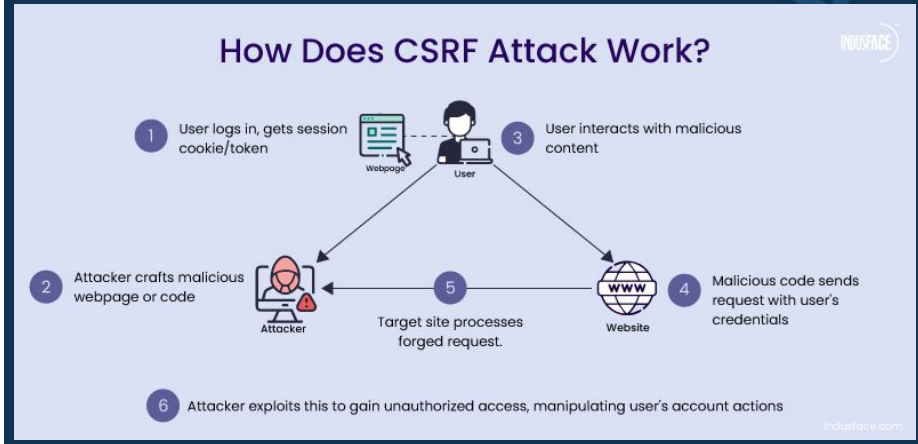




HyperionDev

CSRF





HyperionDev

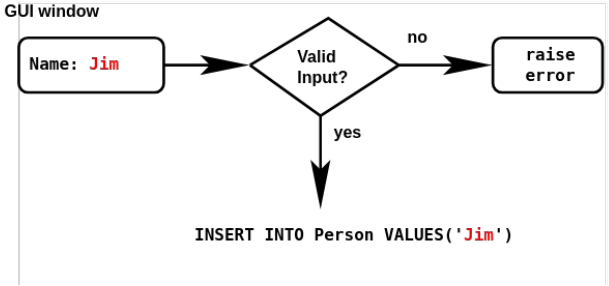
Let’s take a break

HyperionDev

Best Practices for Securing Web Applications

- Use HTTPS: Encrypt data in transit.
- Input Validation: Whitelist and sanitize inputs.
- Content Security Policy (CSP): Prevent unauthorized content loads.
- Regular Updates: Patch vulnerabilities promptly.

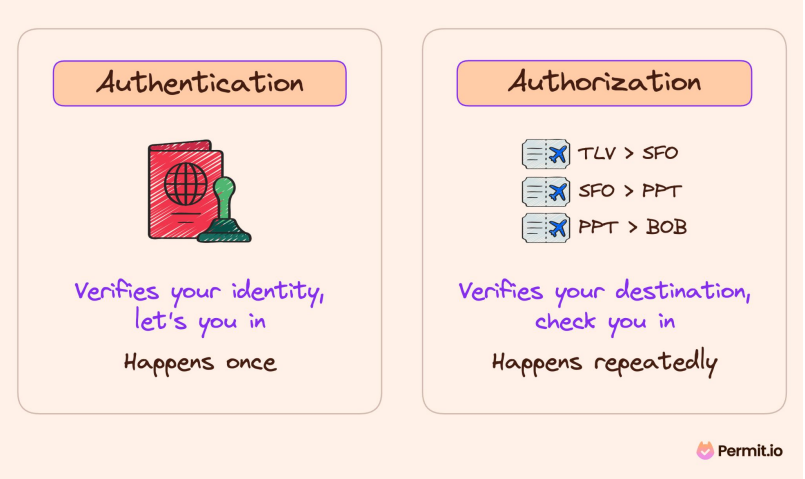
shutterstock.com · 533371321



HyperionDev

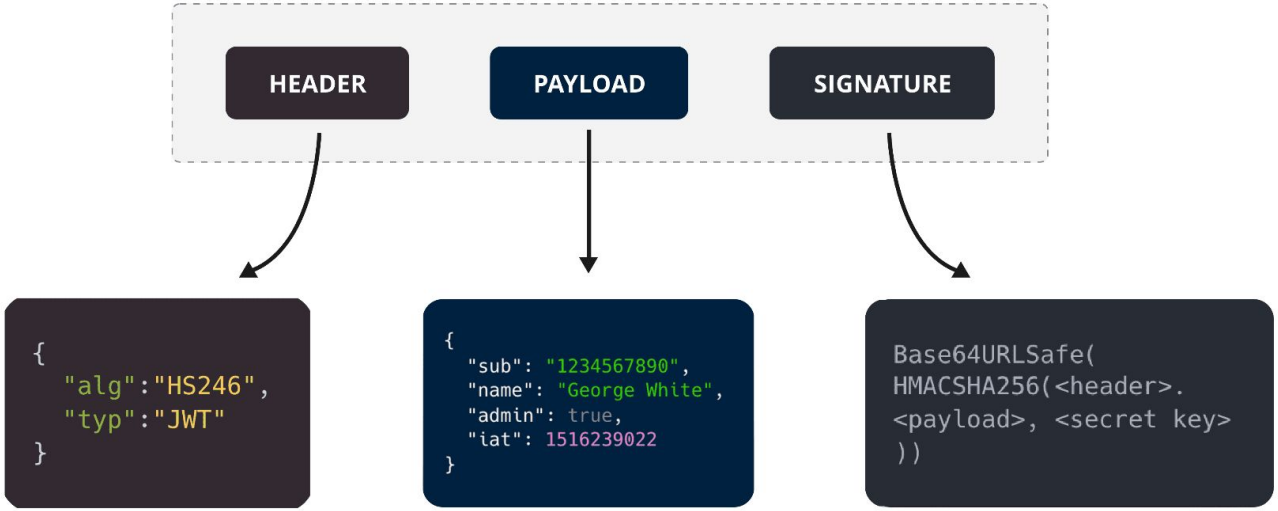
Authentication and Authorization Mechanisms

- Authentication: Verifying user identity.
  - Tools: JWT, OAuth, SSO.
- Authorization: Defining user permissions.
  - Example: Role-based access controls.



How JWT Works

Structure of a JSON Web Token (JWT)

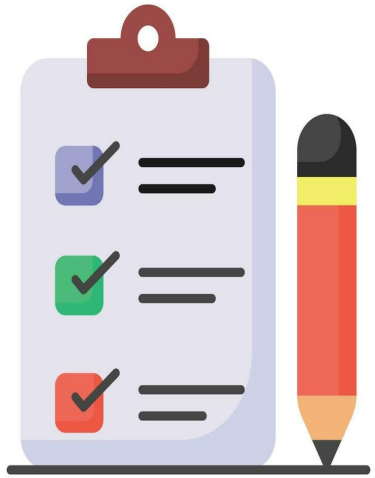


SuperTokens

HyperionDev

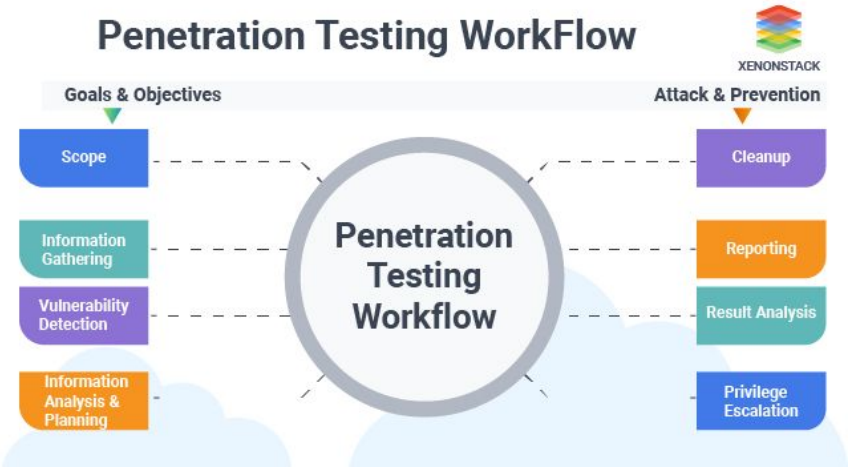
Secure Coding Principles

- **Least Privilege:**
  - Give users the minimum access needed.
- **Error Handling:**
  - Avoid leaking sensitive data in error messages.
- **Code Reviews:**
  - Regular peer audits.
- **Use Established Libraries:**
  - Reduce risks of custom coding errors.



Security Testing Methods

- **Static Code Analysis:**
  - Identifying vulnerabilities in the codebase.
- **Dynamic Application Testing:**
  - Testing applications in runtime.
- **Penetration Testing:**
  - Ethical hacking to uncover weaknesses.



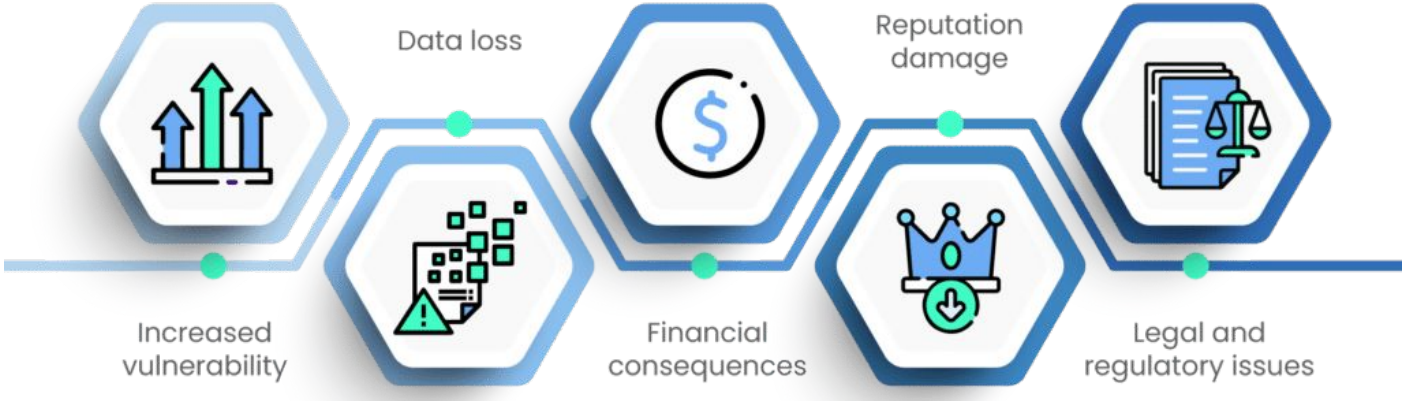
Rule of Law in Web Security

- Legal obligations such as GDPR and Data Protection Act 2018.
- Example: Securing APIs with OAuth for compliant data handling.

Activity

- How could weak security lead to legal violations.

What happens if you have poor cyber hygiene?



HyperionDev

Key Takeaways

- Web security is essential to protect users and comply with laws.
- Understand threats (e.g., XSS, CSRF, SQL Injection).
- Use best practices, authentication, and secure coding principles.
- Regularly test and update your applications.

Which method helps prevent SQL Injection?

A. Using prepared statements
B. Storing passwords in plaintext

What is the purpose of anti-CSRF tokens?

A. Encrypt data
B. Prevent cross-site request forgery

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.


Thank you
for attending
HyperionDev