CoGrammar

Welcome to this session:

Skills Bootcamp - Web Security Fundamentals

The session will start shortly...

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

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Skills Bootcamp Cloud Web Development

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Skills Bootcamp Cloud Web Development

- For all **non-academic questions**, please submit a query: ***www.hyperiondev.com/support***
- **Report a safeguarding incident: *www.hyperiondev.com/safeguardreporting***
- We would love your feedback on lectures: ***Feedback on Lectures.***
- Find all the lecture **content** in your **Lecture Backpack** on GitHub.
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

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.

Have you ever worked on securing a web application?

A. Yes
B. No

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

Question

Key Questions:

- What makes web applications attractive targets for attackers?
- How do threats like XSS and CSRF exploit web application vulnerabilities?
- What practices and tools can developers use to prevent these attacks?

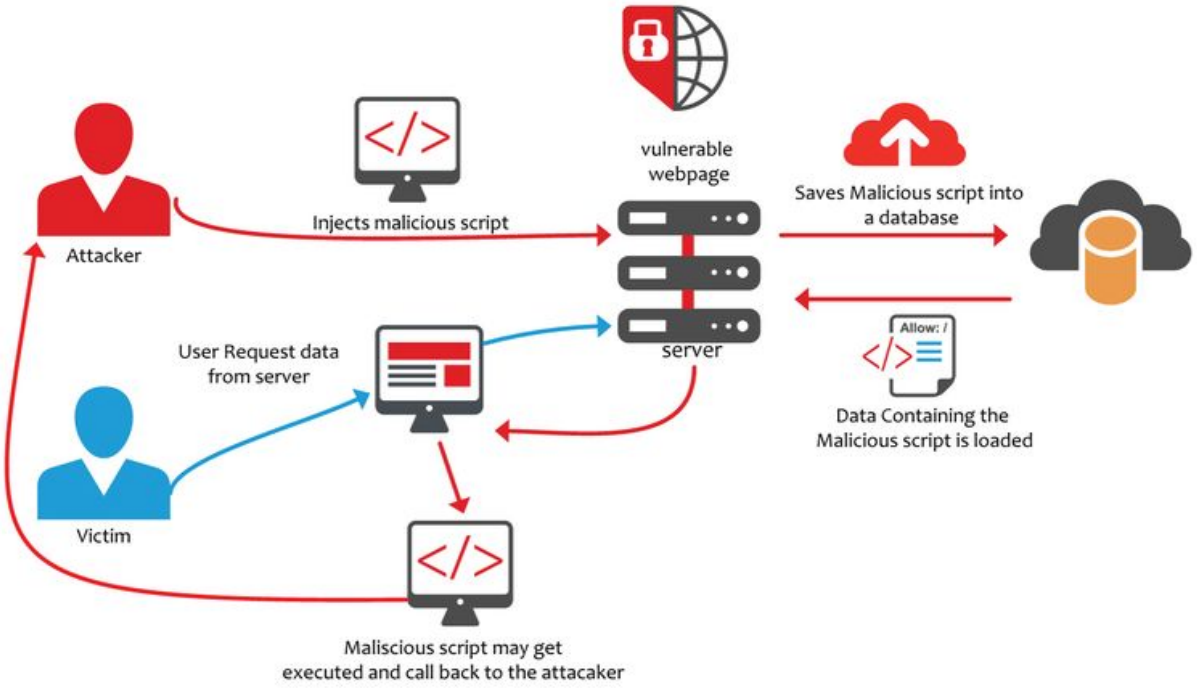
Lecture Overview

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

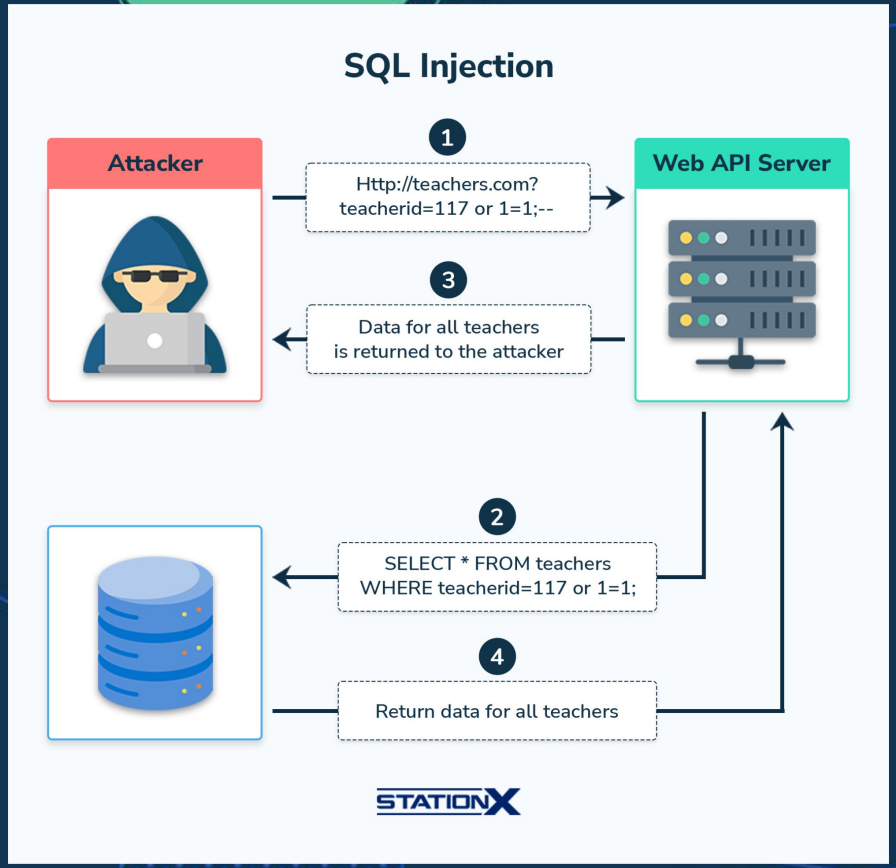
Common Web Security Threats

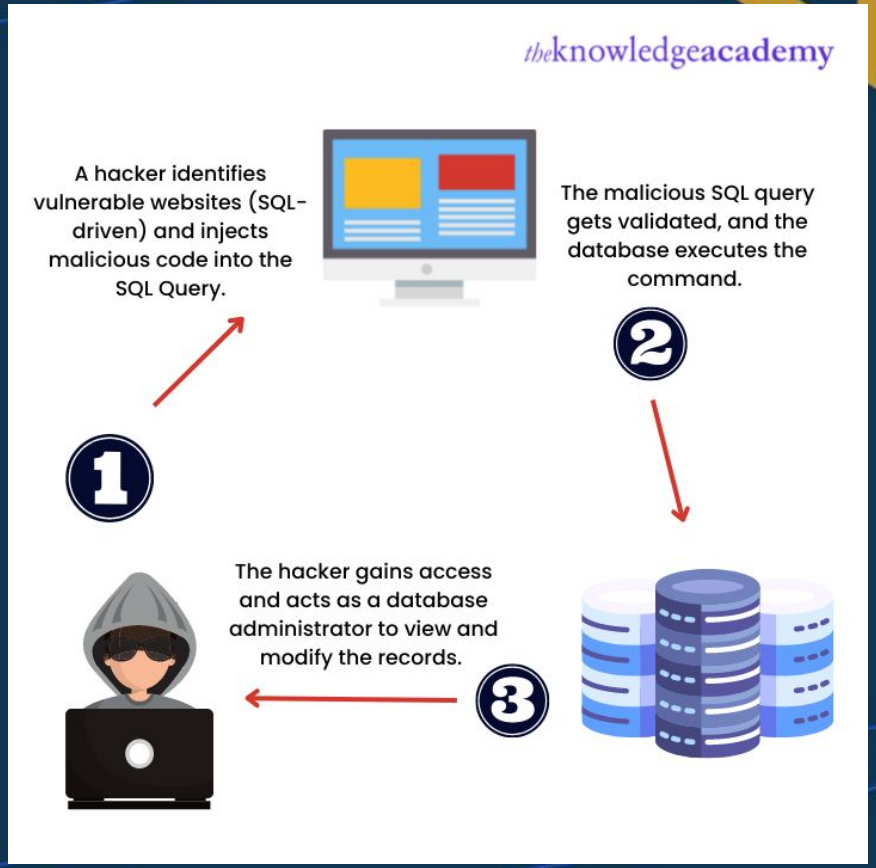
- **Cross-Site Scripting (XSS):** Attacker injects malicious scripts.
  - Example: Popup stealing user data.
- **Cross-Site Request Forgery (CSRF):** Unauthorized actions on behalf of a user.
  - Example: Automatic transfers in banking apps.
- **SQL Injection:** Injecting malicious SQL code into queries.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

XSS Attack



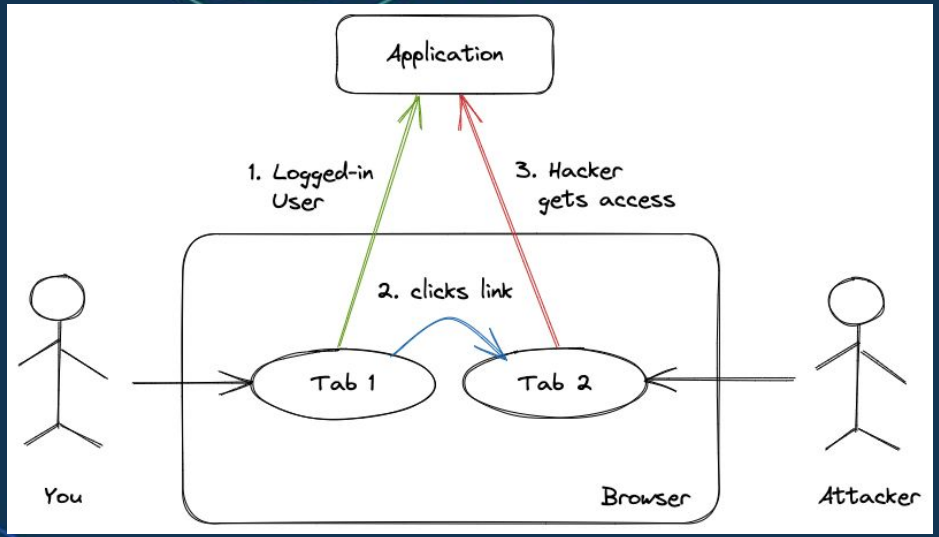
SQL Injection

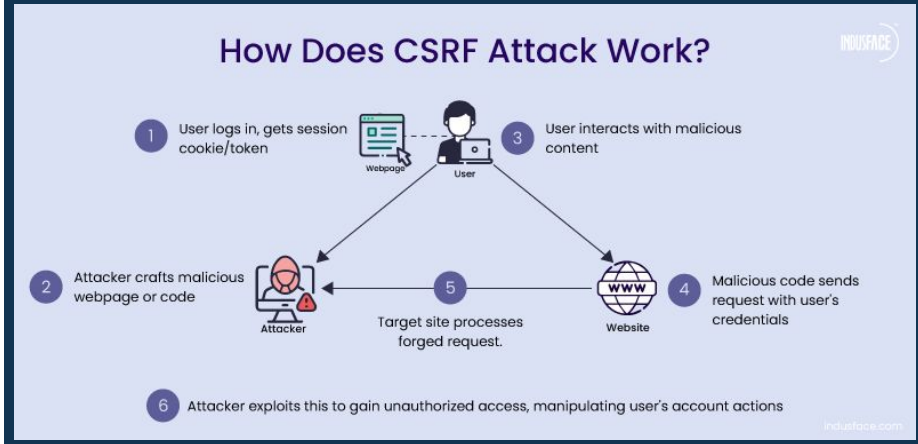




CoGrammar

CSRF



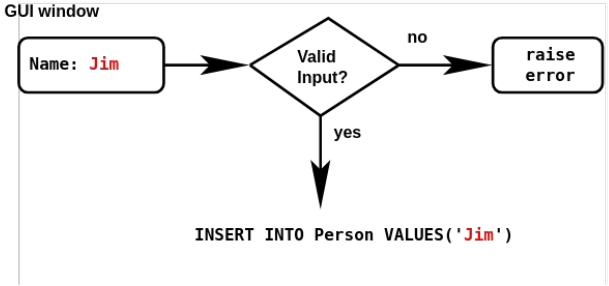


CoGrammar

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



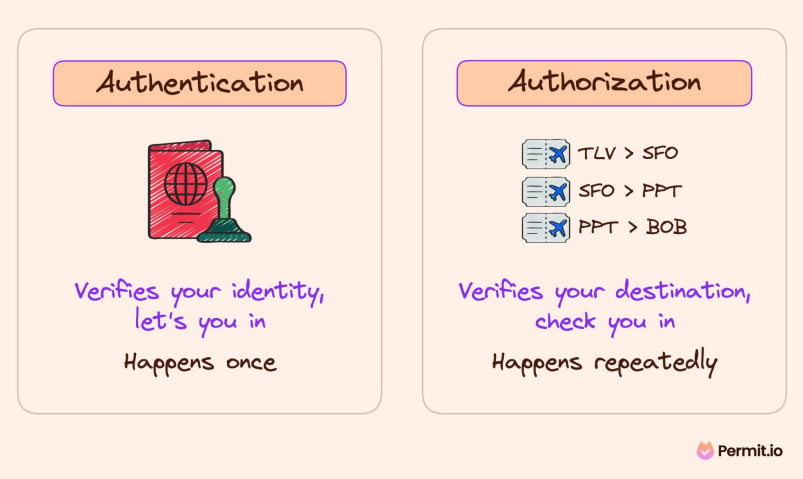
CoGrammar

Let's take a break

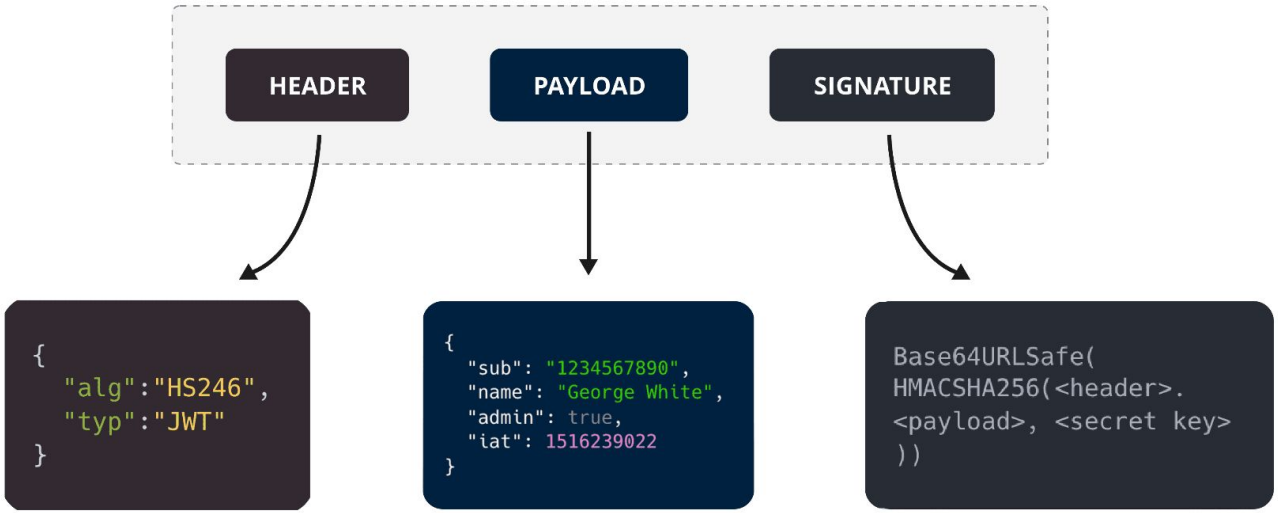
CoGrammar

Authentication and Authorization Mechanisms

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



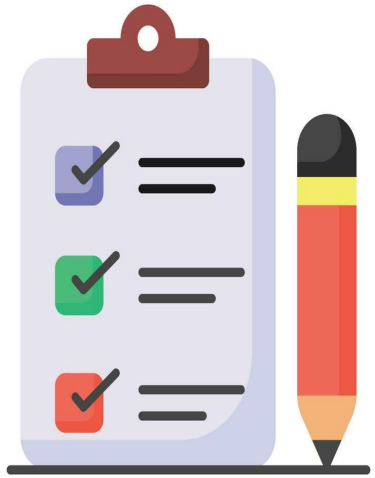
How JWT Works



CoGrammar

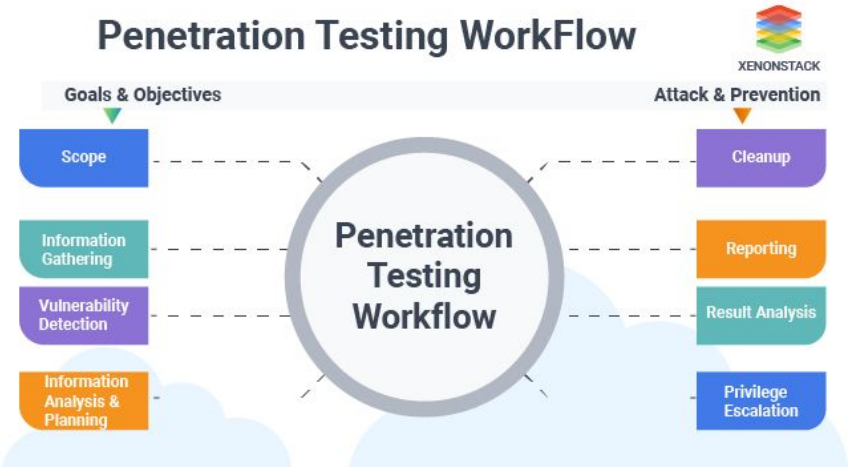
Secure Coding Principles

- **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.



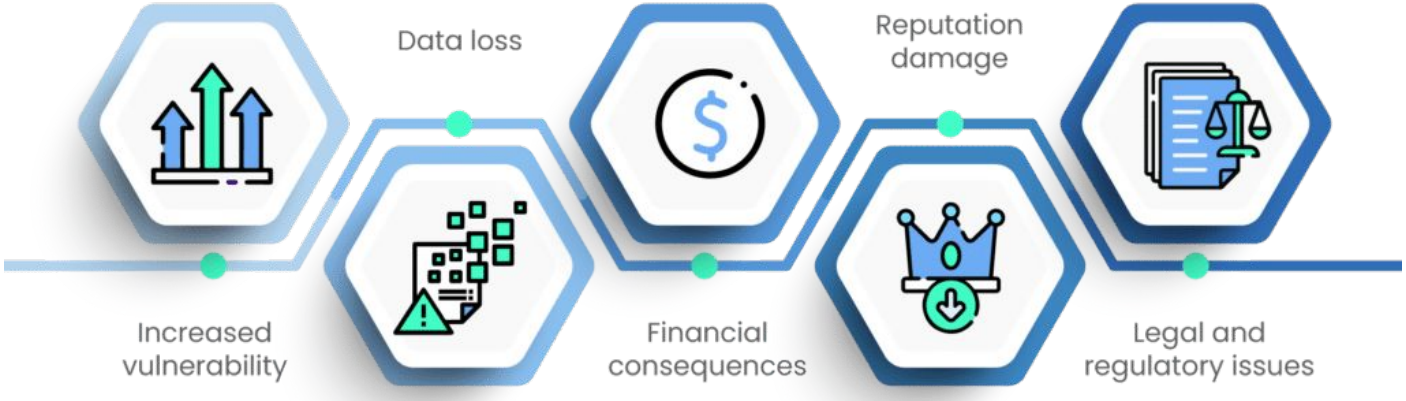
Rule of Law in Web Security

- 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?



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

Questions and Answers

CoGrammar

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education