Injection Attacks

START

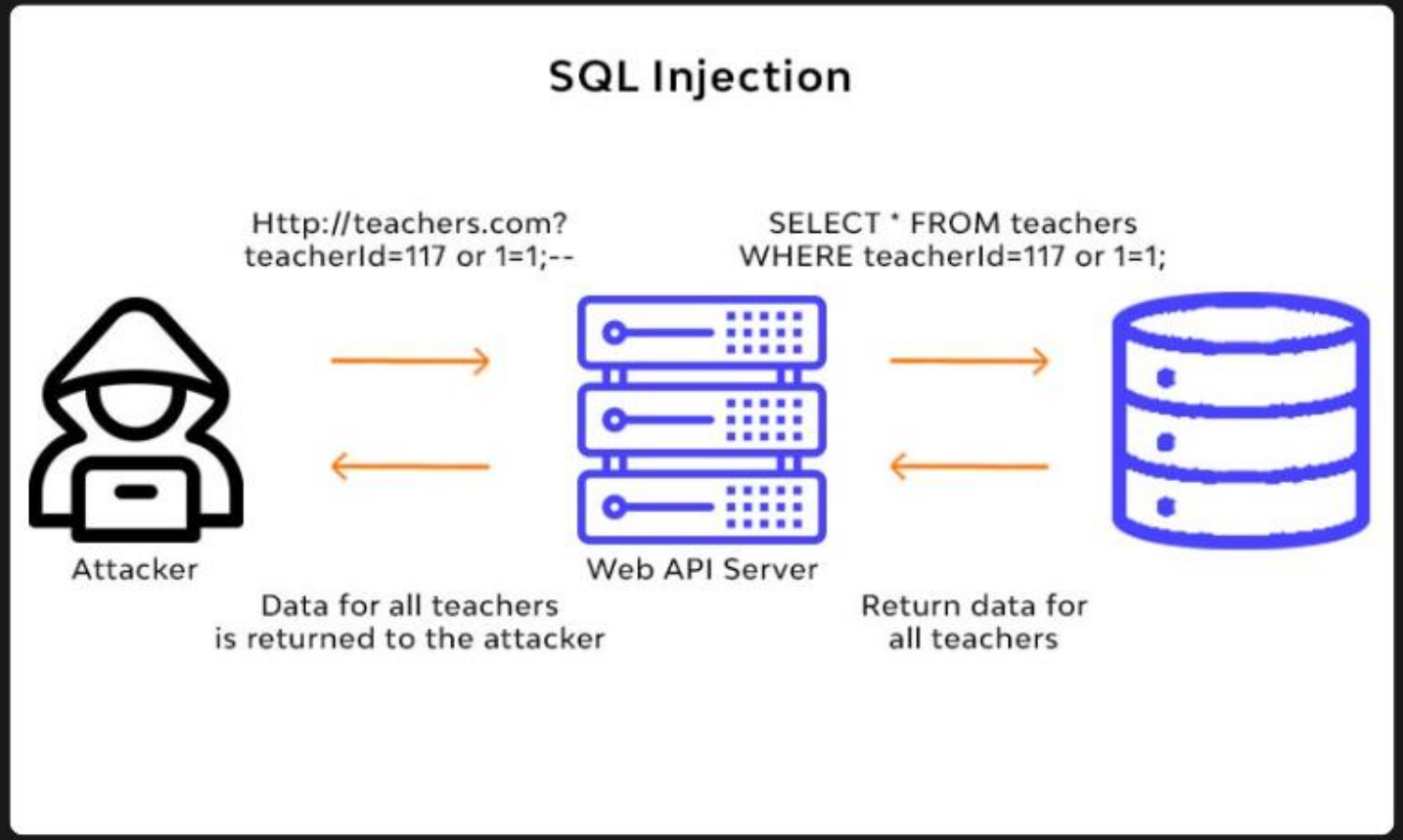
Definition

Injection attacks refer to a broad class of attack vectors. In an injection attack, an attacker supplies untrusted input to a program. This input gets processed by an interpreter as part of a command or query. In turn, this alters the execution of that program.

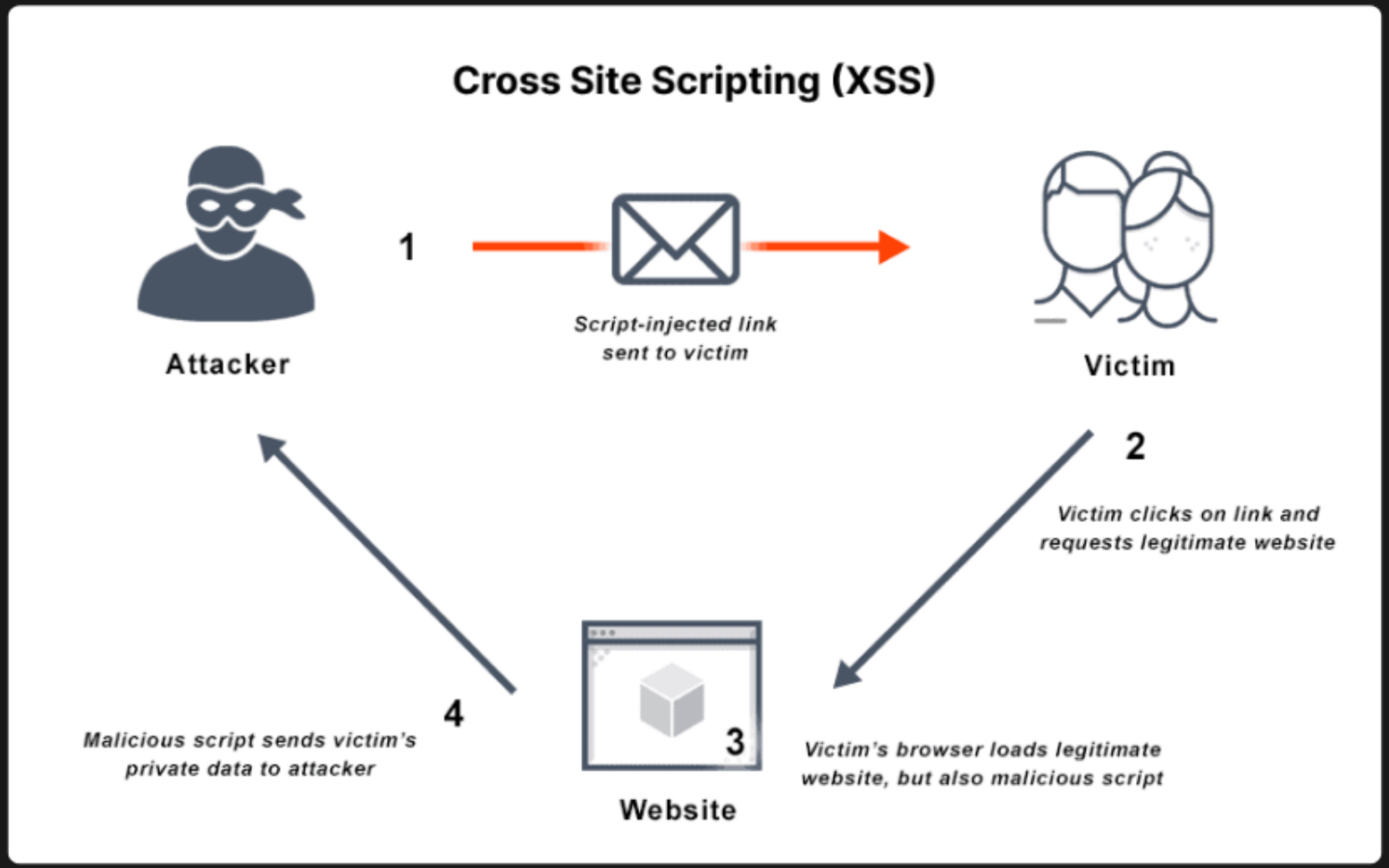
Rating

This attack type is considered a major problem in web security. It is listed as the number one web application security risk in the OWASP Top 10 – and for a good reason. Injection attacks, particularly SQL Injections (SQLi attacks) and Cross-site Scripting (XSS), are not only very dangerous but also widespread, especially in legacy applications.

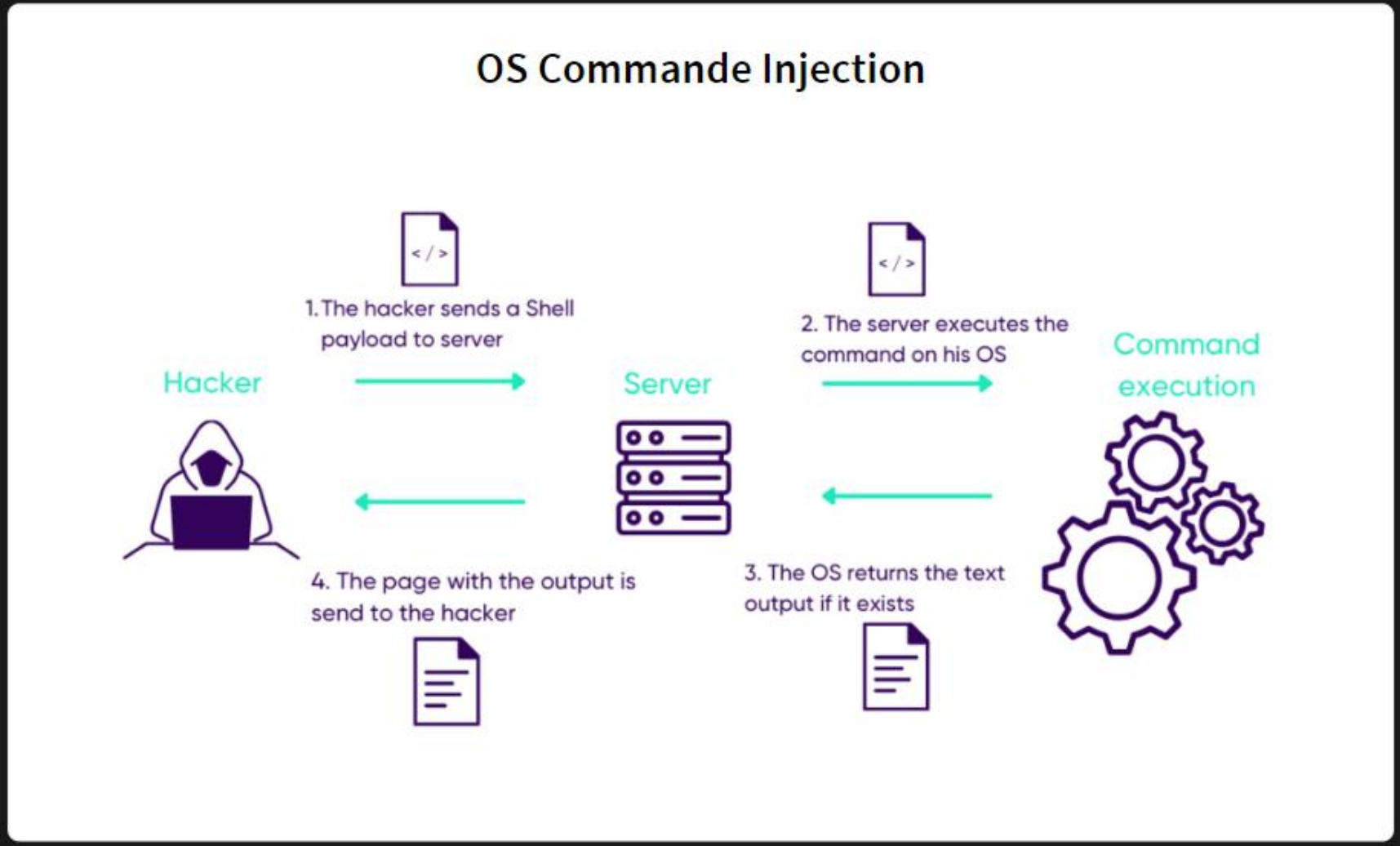
Types of injection attacks



Types of injection attacks



Types of injection attacks



Dammages

- Authentication bypass
- Information disclosure
- Data loss
- Sensitive data theft
- Loss of data integrity
- Denial of service
- Full system compromise.

Prevention

Input Validation and Sanitization

Implement strict input validation to ensure that user inputs adhere to expected formats and patterns. Validate both client-side and server-side inputs to ensure consistency and security.

Prevention

Input Validation and Sanitization

Use parameterized queries and prepared statements in database interactions to separate user input from the actual SQL or other command structure. This helps to prevent attackers from injecting malicious code into queries.

Prevention

Input Validation and Sanitization

Apply input sanitization techniques to filter out potentially harmful characters or escape special characters. For example, replacing single quotes with double quotes or using libraries that automatically handle input sanitization can be effective.

Prevention

Use of Least Privilege Principle

Limit the permissions and access levels of database accounts, web application components, and users to the minimum necessary for their functionality. Adopt the principle of least privilege to reduce the potential impact of a successful injection attack.

Prevention

Use of Least Privilege Principle

Avoid using overly permissive database accounts with unrestricted access. Instead, create specific accounts with the minimum privileges required for their intended tasks, such as read-only or write-only access.

Prevention

Use of Least Privilege Principle

Regularly review and update permissions to ensure they align with the current needs of the application, and revoke unnecessary privileges.

Prevention

Web Application Firewalls (WAF) and Security Testing

Implement a Web Application Firewall (WAF) to monitor and filter HTTP traffic between a web application and the Internet. A WAF can help detect and block injection attacks by analyzing patterns and behaviors indicative of such attacks.

Prevention

Web Application Firewalls (WAF) and Security Testing

Conduct regular security testing, including static analysis and dynamic analysis, to identify vulnerabilities in the application code and database configurations. Automated tools and manual penetration testing can help uncover potential injection points.

Prevention

Web Application Firewalls (WAF) and Security Testing

Keep software frameworks, libraries, and the underlying infrastructure up-to-date to patch known vulnerabilities. Regularly check for security updates and apply them promptly to address any potential weaknesses that could be exploited in injection attacks.

Conlusion

In conclusion, defending against injection cyber attacks demands a multifaceted strategy. Employing rigorous input validation, adhering to the least privilege principle, and integrating tools like Web Application Firewalls (WAF) create a robust defense. Regular security testing and proactive software updates are essential for staying ahead of evolving threats. A vigilant, layered approach is key to safeguarding data integrity and maintaining user trust in the face of persistent injection vulnerabilities.