



SQLI



HyperionDev



Lecture – Housekeeping

- ❑ The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment - please engage accordingly.
- ❑ 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.
- ❑ You can also submit questions here:
hyperiondev.com/sbc4-cs-questions
- ❑ For all non-academic questions, please submit a query:
www.hyperiondev.com/support
- ❑ Report a safeguarding incident:
hyperiondev.com/safeguardreporting
- ❑ We would love your feedback on lectures:
<https://hyperiondev.wufoo.com/forms/zsgv4m40ui4i0g/>

Previously:

- Be introduced to the syntax for:
- Creating tables (inserting)
- SELECT Statement
- Conditions using WHERE
- Functions & Deleting

Objectives

- Recap on what SQL injections are
- Take a look at how these are executed
- Discover the severity of these attacks
- Learn how SQLI can be mitigated



Poll for the day:



**Do you know how attackers
actually inject malicious SQL?**



SQLI Recap

An SQL injection occurs when an attacker uses malicious input to influence a web application's database query. It happens when user-supplied data is used to build SQL queries without first being properly validated or sanitized. Attackers may use this flaw to run unauthorized SQL commands, change data, or gain unauthorized access to confidential data.

Severity of SQLI?

Unauthorized Access: By inserting SQL statements that change or obfuscate login checks, attackers can get around authentication mechanisms. They can then gain unauthorized access to private information or even administrative rights.

Data Breaches: SQL injections can expose or steal sensitive data, including user credentials, personal information, financial information, and intellectual property. Significant reputational harm and legal repercussions may result from this.

Remote Code Execution: In some circumstances, SQL injection flaws can give attackers access to the web server and allow them to run arbitrary code, completely compromising the application and the underlying infrastructure.


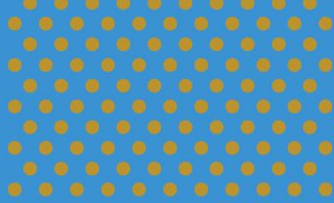
How to prevent SQLI?

- Ensure that all user input complies with expected formats and is free of malicious characters and SQL code by validating and sanitizing it. **Strict input validation** procedures, like regular expressions, can be used to accomplish this, as well as input sanitization methods like escape or removal of special characters.
- Use **parameterized queries** or **prepared statements**, also referred to as parameter binding, to distinguish between SQL code and user input. In this method, placeholders are created in the SQL statement, and the user input is then bound to these placeholders. This effectively mitigates SQL injection attacks by preventing user input from being directly concatenated into the SQL statement.
- Make sure that the database user accounts used by the application have the fewest privileges required for them to carry out their assigned tasks according to the **least privilege principle**. Only grant the account access to the necessary tables, columns, and operations. It is possible to lessen the effects of a successful SQL injection attack by adhering to the principle of least privilege.


What we've covered?

Next up

Network Security & Firewall Management



Poll for the day:
Do you know how to
create/manipulate a table in SQL?





Hyperiondev

Thank you for joining us

1. Take regular breaks
2. Stay hydrated
3. Avoid prolonged screen time
4. Practice good posture
5. Get regular exercise



Questions and Answers

