

SQL Injection

Troy Hunt
troyhunt.com
@troyhunt



pluralsight 
hardcore developer training

Outline

- Understanding SQL injection
- Testing for injection risks
- Discovering database structure via injection
- Harvesting data via injection
- Automating attacks with Havij
- Blind SQL injection
- Secure app patterns

The significance of SQL injection

T10

OWASP Top 10 Application Security Risks – 2013

A1 – Injection

Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

A2 – Broken Authentication and Session Management

Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.

A3 – Cross-Site Scripting (XSS)

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

A4 – Insecure Direct Object References

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

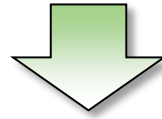
A5 – Security

Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings

Understanding SQL injection

Trusted

http://www.mysite.com/Widget?Id=1



SELECT * FROM Widget WHERE ID = 1

Untrusted

Breaking out of the data context

/Widget?Id=1 or 1=1



```
SELECT * FROM Widget WHERE  
ID = 1 or 1=1
```

Types of SQL injection attack

- **Explicit**
 - Union-based: append a result set that's rendered to the markup
 - Error-based: disclose information in an unhandled exception
- **Implicit (blind)**
 - Boolean-based: test if a particular condition is true
 - Time-based: cause the response to be delayed in response to a test

Summary

- **SQL injection risks exploit the ability to break out of the data context and enter the query context**
 - Remember untrusted data – here it is again!
- **Once a risk is established, a creative attacker may exploit it to discover the internal database structure and then extract data**
 - Table names
 - Column names
- **It could go even further than that – the account the web app is using may be able to drop or create objects in the database**
- **SQL injection is a repeatable pattern that's frequently automated with tools like Havij**
- **Even with no *explicit* feedback, “blind” injection can still be leveraged**
- **Always check for parameterisation and whitelisting of untrusted data**