- Filename: eccouncil-ceh31250-v11-15-1-1-sql-injection-concepts.md
- Show Name: CEHv11 (312-50)
- Topic Name: Web Application Hacking: SQL Injection
- Episode Name: SQL Injection Concepts
  ==

SQL Injection Concepts

Objectives:

- Define what SQLi is
- List and explain the different types of SQLi
- Explain and demonstrate methods used to discover SQLi vulnerabilities
- Explain the process of a SQLi attack and why it works
- List and explain common types of SQLi IDS signature evasion techniques
- List and describe common security controls and best practices to secure systems against SQLi

---

- What is SQL Injection and why are we concerned about it?
    - HOW?
        - Modifying back-end SQL queries
            - User can 'inject' T-SQL into original query
    - WHY CONCERN?
        - CIA of data is compromised
            - Data can be:
                - Extracted
                - Modified
                - Deleted
                - Access of target's local file-system
                - Remote access to Target's system commands
- How does this happen?
    - Insecure coding
        - Trusts user input
- SQL injection types?
    - Authentication Bypass
        - DEMO an AUTH Bypass and show the code from DVWA to explain
    - Error-based SQLi
    - Blind SQLi
- How do we discover a possible SQL injection point?
    - Manual Discovery
        - Look for visible input
            - Login forms
            - Dynamic site pages
            - Search boxes
            - URLs with things like '?id=1'

        - Invisible input
            - Page source
            - API calls (DEMO)(just show the API injection point)
    - Automation
        - Vulnerability Scanners
        - SQLi Specific vulnerability scanners
        - SQLmap
        - SQLNinja
        - Mobile sqli tools
- What are some common defenses against SQL Injection?
    - Regex filtering aka Input Validation
        - Look for special characters and strings used in SQLi
    - WAFs
    - Least privilege
    - Parameterized statements
        - Prepared statements
- Are there ways around these defenses?
    - Query Obfuscations
        - Inline Comments
        - Null bytes (%00)
        - Use variables
    - Encoding special chars
        - Hex
        - URL
    - Concatenation
    - Uncommon queries
        - Look for 'OR DOG=DOG' instead of 'OR 1=1'