Command Injection Execution

By Inlighn Tech

Objective:

This project helps you understand how attackers exploit command injection vulnerabilities to execute system commands on the server. You'll practice basic and blind command injection, bypassing filters, and executing a reverse shell using vulnerable web applications.

Tools & Labs Required

- **DVWA (Damn Vulnerable Web Application)**
- **bWAPP (Buggy Web Application)**
- **Kali Linux / Parrot OS**
- **Burp Suite** (for intercepting/filtering payloads)
- **Netcat (nc)** (for catching reverse shell)
- **TryHackMe** (optional for blind command injection challenges – "Injection" or "Basic Pentesting" rooms)

Part 1: DVWA – Command Injection

Setup

- Set DVWA **Security Level** to both **Low** and **High** (test both).
- Navigate to **Command Injection** vulnerability.

Tasks

- **Basic Command Injection:**
  - Input: `127.0.0.1; whoami`

    - Try different delimiters: `&&`, `|`, `&`, backticks (`` `ls` ``), `||`.
    - Observe the output.

- **Blind Command Injection:**
    - Use: `127.0.0.1 && ping -c 3 127.0.0.1`
    - Check for server delay to confirm execution.

- **Command Injection via Burp Suite:**
    - Capture the request, send to Repeater.
    - Try URL-encoded payloads: `127.0.0.1%26%26id`
    - Analyze response timing/content.

Part 2: bWAPP – OS Command Injection

Setup

- Open bWAPP and login.
- Choose "**OS Command Injection**" from the dropdown.
- Security level: Test both **Low** and **Impossible**.

Tasks

1. **Basic Injection:**
    - Try: `127.0.0.1; whoami`
    - Use `&`, `|`, `&&`, and backticks (`` `whoami` ``)

2. **Advanced Filtering Bypass:**
    - Use URL encoding, double encoding.
    - Try breaking out of filters with payloads like:
        - `127.0.0.1%26%26cat%20/etc/passwd`
        - `` 127.0.0.1`id` ``

3. **Blind Command Injection:**
    - Use ping or sleep-based delays.
    - Try payload: `127.0.0.1 && sleep 5`
    - Time the response.

4. **Reverse Shell:**
    - Same as in DVWA: try Bash, Netcat, or PHP shells.
    - Upload a custom shell if necessary.

What to Submit

Students must submit the following:

1. **Screenshots**:
    - Successful basic and blind command injection (DVWA + bWAPP).
    - Burp Suite request showing payload.
    - Terminal output for reverse shell (if achieved).

2. **Payload List**:
    - Document the payloads used, and which worked or failed.
    - Include bypass examples (URL encoded, alternative syntax).

3. **Short Report (200–300 words)**:
    - Describe the vulnerability, how you exploited it.
    - Mention differences between Low and High security levels.
    - Explain how command injection can be prevented in real-world apps.