BSH201 - DEFENDING BASH

Course Learning Objectives

Defending Bash is a course designed to provide you with an overview of security concepts related to Bash and shell script development. We'll begin in module one with an overview of Bash and Shell server environment vulnerabilities and maliciously manipulated environment variables in scripts. In module two, we look at how attackers can use files and directories that scripts write or read from to commit malicious activities. In the third module, we examine attacks that focus on malicious input and injection attacks. In module four, we discuss different ways to protect a script from unintended access as well as best practices regarding permission levels between users and scripts. Finally, in module five, we cover common vulnerabilities in Bash and Shell scripts and security best practices.

Description

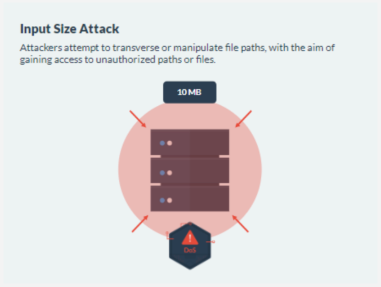
Defending Bash is a course designed to cover best security practices along the SDLC for Developers, SysAdmins, DevOps professionals, Data Engineers, and more. This course is designed for intermediate students who have already familiar with basic security concepts. While this course is designed to address security concerns Bash, a Unix command language, it also applies to any command-line interface shell.

Audience

SysAdmins
DevOps Professionals
Data Engineers

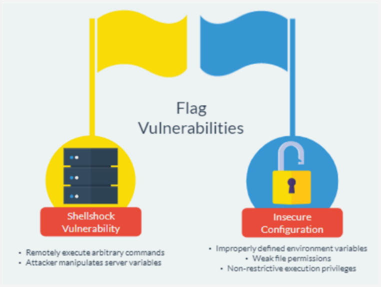
Time Required

Tailored learning - 45 minutes total (approx.)



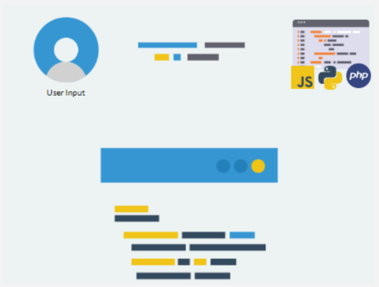
Input Size Attack

Attackers attempt to transverse or manipulate file paths, with the aim of gaining access to unauthorized paths or files.



- Remotely execute arbitrary commands
- Attacker manipulates server variables

- Improperly defined environment variables
- Weak file permissions
- Non-restrictive execution privileges



Security Compass

BSH201 - DEFENDING BASH

Course Outline

1. Bash and Shell Environment Vulnerabilities

- Path environment attacks
- Code: safe_command execution
- Best practices against path environment attacks
- Untrusted third-party input
- Code: injection attack
- Path traversal and file path manipulation
- Code: vulnerable script and mitigation
- Input aize attack
- Code: shell metacharacter
- Code: publicly writable and readable directories
- Code: adding security
- Code: grep and find
- Best practices: file and directory

2. File and Directory Attacks

- Input file attacks
- Code: Bash script
- Best practices: input file attacks
- Using unmask
- Code: grant read/write permissions
- Code: rewrite Bash script
- File upload vulnerabilities
- Code: Bash script to check file type
- Code: text-only Bash script
- Denial of service attack
- Code: stat command
- Code: ulimit command
- Input validation
- Code: maximum size set
- Using a configuration file
- Overwriting files
- Code: validating file content against executable code
- Defenses and best practices

3. Injection Attacks

- What are injection attacks?
- Command injection (Python)
- Code: input, executes a shell script command
- Code: mitigate against command injection attacks
- Code: using the ProcessBuilder class
- Code: secure this code
- Command injection (Java)
- Code: quoting and escaping
- Code injection attacks
- Code: code injection attacks
- Code: quoting handled securely
- File path injection
- Code: vulnerable to file path injection
- SQL injection
- Code: SQL injection
- Defenses and best practices

4. Authentication and Authorization

- Differences in authentication and authorization
- Authentication attacks
- Code: password-based authentication
- Code: password hashing and salting
- Rate limiting and password policies
- Code: rate limiting
- Code: input sanitization and validation
- Code: SSH keys, LDAP, and authentication
- Permissions and access controls
- Code: permissions and access controls
- File permissions
- Permission types
- Code: permissions and access controls
- User privilege verification

5. Common Vulnerabilities and Security Practices

- Environmental vulnerabilities
- Environment variable attack
- Code: environment variable attack
- Code: environment input injection
- Shellshock attack
- Code: shellshock attack
- Code: log vulnerabilities
- Log injection
- Code: log injection attack protection
- Sensitive information exposure in logs
- Code: appropriate permissions on log files
- Improperly managed processes
- Code: improperly managed processes
- Poor cryptography
- Avoid hardcoding keys
- Code: Validate data before encrypting
- Handle cryptography operation errors
- Code: Handle cryptography operation errors
- Flags
- Vulnerabilities and defenses

SecurityCompass