Command Injection

이정호

목 차

01. Command Injection이란?

02. 공격 원리

03. 공격 실습

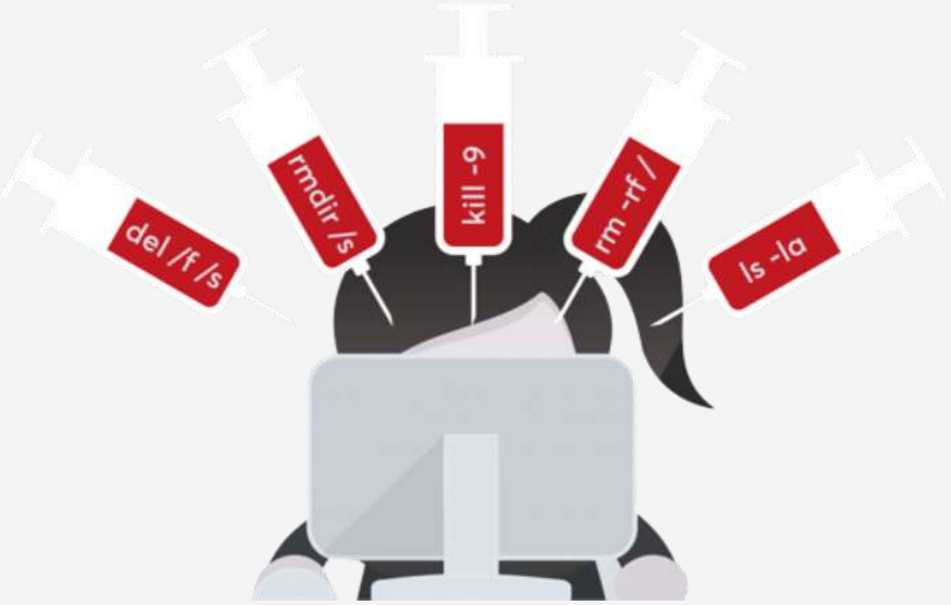
04. 대응 방안

05. Q&A

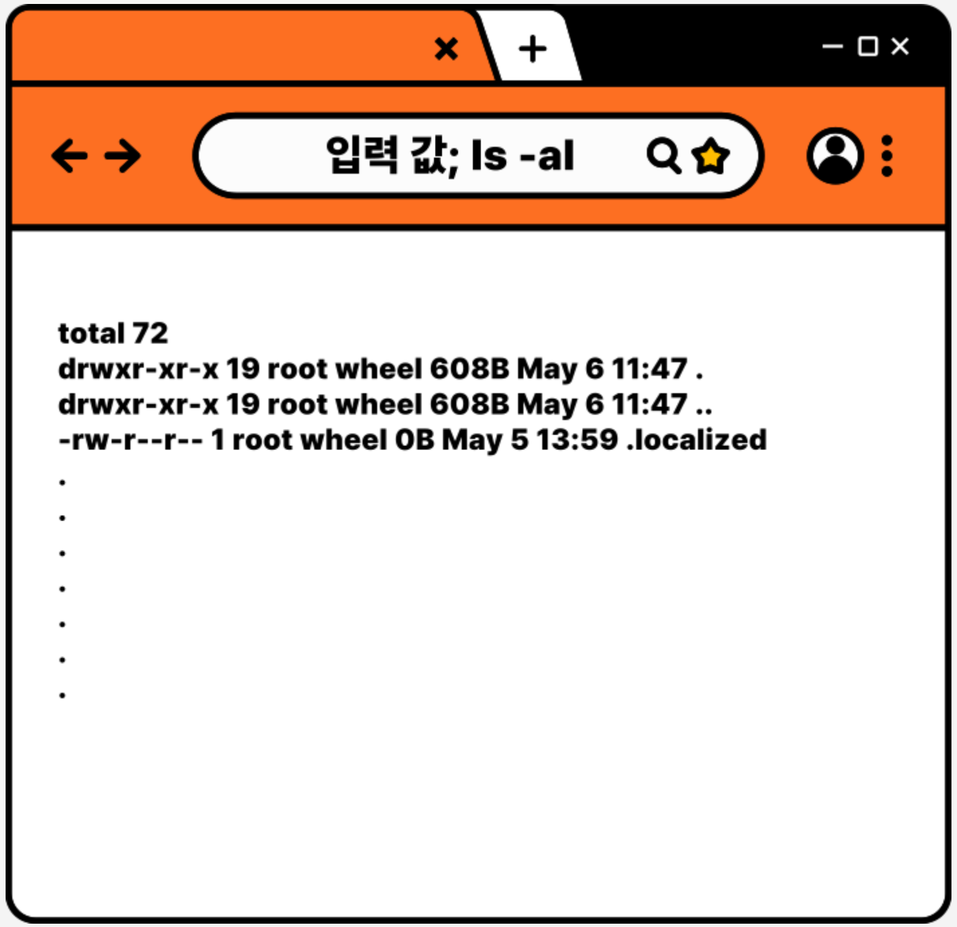
01. command injection이란?

Command Injection : 명령어 삽입

Command Injection은 해커가 웹 애플리케이션에 악성 코드를 삽입하여 해당 시스템에서 실행되는 명령어를 조작하는 공격 기법



02. 공격 원리



[일반적인 입력]; [실행하고자 하는 명령]

;	앞 명령어 실행 후 뒤 명령어 실행(리눅스 전용, 복수 명령어 실행 시 사용하는 문자)
&&	명령어 연속 실행, 한 줄에 여러 명령어를 사용하고 싶을 때 사용. 앞 명령어가 참이여야 뒷 명령어가 실행됨
|	파이프 앞 명령어 결과가 뒷 명령어 입력으로 들어감
그 외	||, >, >>, &>, >&, <, {}, ?, *, ~등등

02. 공격 원리

03. 공격 실습

Low Medium High Impossible

03. 공격 실습

LOW

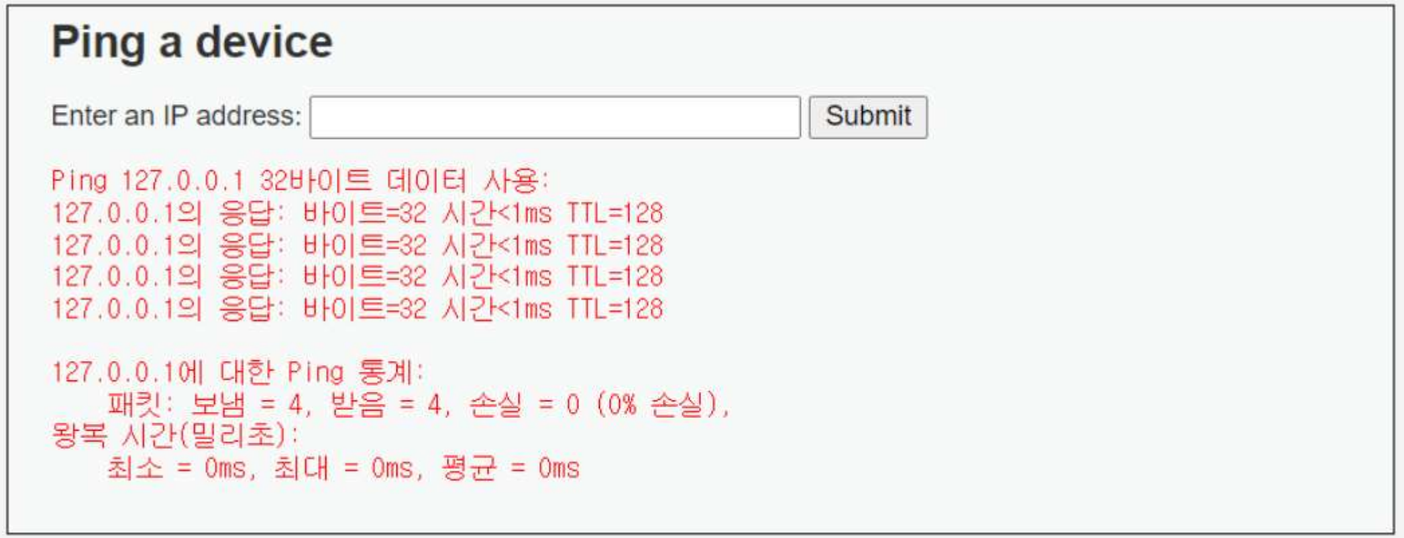
시나리오 가정

IP 주소를 입력하면 해당 IP 주소의 시스템이 활성화되어 있는지 알려주는 웹 서비스에 command injection을 사용하여 시스템 정보 획득

03. 공격 실습

LOW

IP 주소: 127.0.0.1



03. 공격 실습

LOW

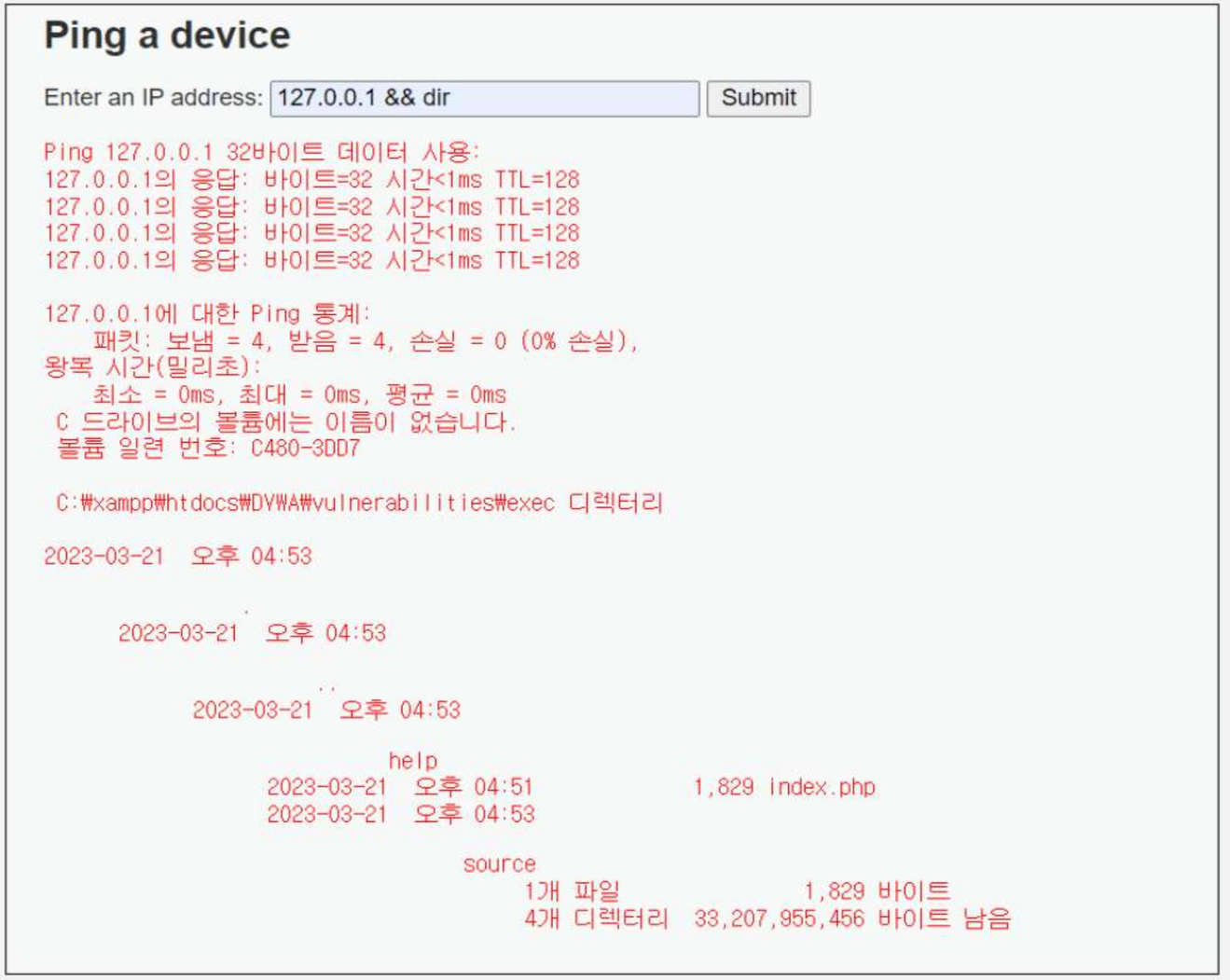
PHP 소스코드

```php
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Get input
    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping  ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping  -c 4 ' . $target );
    }

    // Feedback for the end user
    echo "<pre>{$cmd}</pre>";
}

?>
```

03. 공격 실습

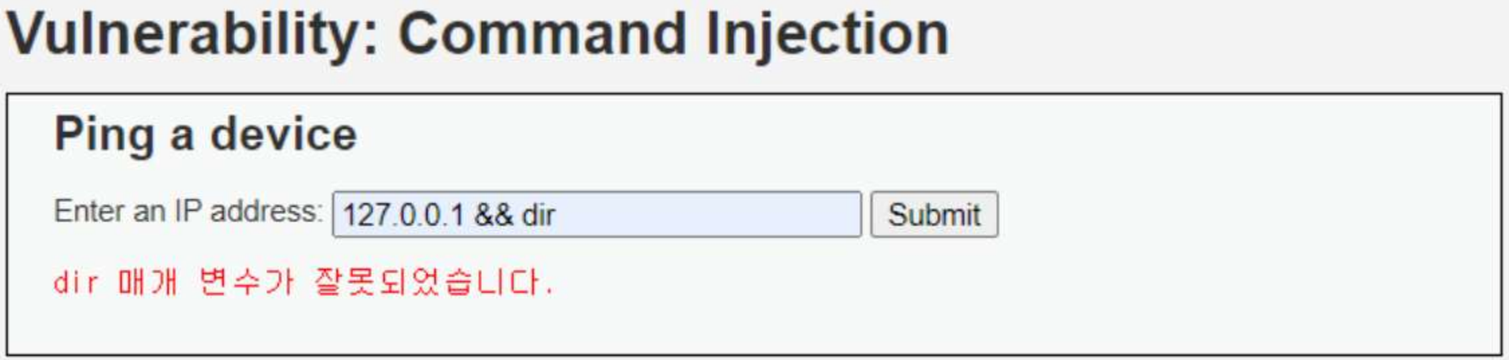
LOW

127.0.0.1 && dir



03. 공격 실습

High


Vulnerability: Command Injection
Ping a device
Enter an IP address: 127.0.0.1 && dir
Submit
dir 매개 변수가 잘못되었습니다.

03. 공격 실습

High

PHP 소스코드

```php
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Get input
    $target = trim($_REQUEST[ 'ip' ]);

    // Set blacklist
    $substitutions = array(
        '&'  => '',
        ';'  => '',
        '| ' => '',
        '-'  => '',
        '$'  => '',
        '('  => '',
        ')'  => '',
        '`'  => '',
        '||' => '',
    );

    // Remove any of the characters in the array (blacklist).
    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

    // Determine OS and execute the ping command.
    if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping  ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping  -c 4 ' . $target );
    }

    // Feedback for the end user
    echo "<pre>{$cmd}</pre>";
}

?>
```

```php
// Set blacklist
$substitutions = array(
    '&'  => '',
    ';'  => '',
    '| ' => '',
    '-'  => '',
    '$'  => '',
    '('  => '',
    ')'  => '',
    '`'  => '',
    '||' => '',
);
```

Vulnerability: Command Injection

Ping a device

Enter an IP address: 127.0.0.1 && dir Submit

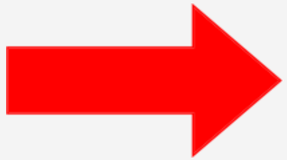
dir 매개 변수가 잘못되었습니다.

03. 공격 실습

High

127.0.0.1|dir

'|' => '',



Ping a device

Enter an IP address: 127.0.0.1|dir Submit

```
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: C480-3DD7

 C:₩xampp₩htdocs₩DVWA₩vulnerabilities₩exec 디렉터리

2023-03-21  오후 04:53

      2023-03-21  오후 04:53

           2023-03-21  오후 04:53

                help
          2023-03-21  오후 04:51             1,829 index.php
          2023-03-21  오후 04:53

                source
                    1개 파일                 1,829 바이트
                    4개 디렉터리  33,205,186,560 바이트 남음
```

04. 대응 방안

Impossible

```php
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Check Anti-CSRF token
    checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'index.php' );

    // Get input
    $target = $_REQUEST[ 'ip' ];
    $target = stripslashes( $target );

    // Split the IP into 4 octects
    $octet = explode( ".", $target );

    // Check IF each octet is an integer
    if( ( is_numeric( $octet[0] ) ) && ( is_numeric( $octet[1] ) ) && ( is_numeric( $octet[2] ) ) && ( is_numeric( $octet[3] ) ) && ( sizeof( $octet ) == 4 ) ) {
        // If all 4 octets are int's put the IP back together.
        $target = $octet[0] . '.' . $octet[1] . '.' . $octet[2] . '.' . $octet[3];

        // Determine OS and execute the ping command.
        if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
            // Windows
            $cmd = shell_exec( 'ping  ' . $target );
        }
        else {
            // *nix
            $cmd = shell_exec( 'ping  -c 4 ' . $target );
        }

        // Feedback for the end user
        echo "<pre>{$cmd}</pre>";
    }
    else {
        // Ops. Let the user name theres a mistake
        echo '<pre>ERROR: You have entered an invalid IP.</pre>';
    }
}

// Generate Anti-CSRF token
generateSessionToken();

?>
```

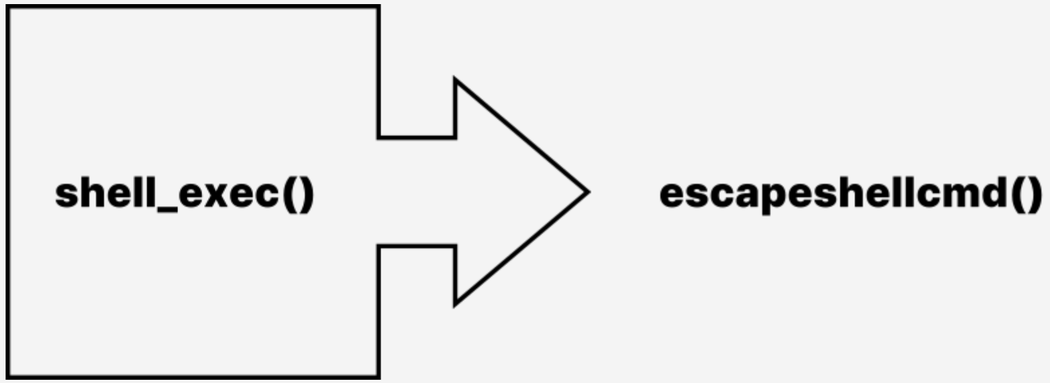
```php
// Check IF each octet is an integer
if( ( is_numeric( $octet[0] ) ) && ( is_numeric( $octet[1] ) ) && ( is_numeric( $octet[2] ) ) && ( is_numeric( $octet[3] ) ) && ( sizeof( $octet ) == 4 ) ) {
    // If all 4 octets are int's put the IP back together.
    $target = $octet[0] . '.' . $octet[1] . '.' . $octet[2] . '.' . $octet[3];
```

04. 대응 방안

1. 입력값 검증

;	앞 명령어 실행 후 뒤 명령어 실행(리눅스 전용, 복수 명령어 실행 시 사용하는 문자)
&&	명령어 연속 실행, 한 줄에 여러 명령어를 사용하고 싶을 때 사용. 앞 명령어가 참이여야 뒷 명령어가 실행됨
|	파이프 앞 명령어 결과가 뒷 명령어 입력으로 들어감
그 외	||, >, >>, &>, >&, <, {}, ?, *, ~등등

2. 안전한 API 사용



Q&A

Thank you