2000.00.00

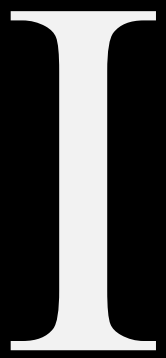
# Authentication

## 가디언 웹 보안 세미나 4

임준서

# Authentication

Concept



# Authentication

Concept

내가 누군지 증명하는 과정

NOT Authorization

알고 있는 정보를 통해 확인

이메일, 아이디, 그리고 비번 등

중심에서 0.3cm 떨어진 소속 18px

# Authentication vs Authorization

The difference

## Authorization: 내가 무슨 권한을 가졌는지

Are you admin? user?

중심에서 0.3cm 떨어진 소속 18px

# 공격 유형

무슨 권한을 얻는가에 따른 분류

## 수평적 권한 상승

user의 계정을 볼 수 있다.

## 수직적 권한 상승

admin의 계정을 볼 수 있다.

중심에서 0.3cm 떨어진 소속 18px

# Authentication Process

Outline

0. 사용자의 정보를 저장 – id / pw

1. 사용자가 로그인 요청을 form을 통해 보냄

2. 네트워크를 통해 서버로 전달

중심에서 0.3cm 떨어진 소속 18px

# Authentication Process

Outline

3. 서버에서 DB에 저장된 정보와 비교

4. 성공 여부를 토큰/세션으로 전달

5. 클라이언트에서 로그인 상태를 유지

중심에서 0.3cm 떨어진 소속 18px

# Authentication

0. 사용자의 정보를 저장 – id / pw

"The password is too simple"

BruteForce를 통해 뚫린다.

– Rockyou.txt

id/pw에서 문제가 발생하는 경우

중심에서 0.3cm 떨어진 소속 18px

# Homework [brute brute]

http://dreamhack.io/wargame/challenges/1688

```
sudo apt-get install john -y
```

```
zip2john target.zip > hash.txt && \
john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```

```
sudo apt-get install wordlists -y && \
    sudo gzip -d /usr/share/wordlists/rockyou.txt.gz
```

web에서는 requests 기반 패키지를 통해서 brute force를 시행한다.

중심에서 0.3cm 떨어진 소속 18px

# Brute Force

종류

Rockyou로 해시를 구해도 됨

https://crackstation.net/ 활용

Rainbow Table Attack

중심에서 0.3cm 떨어진 소속 18px

# Authentication

Brute force protection

## 시도 횟수 제한

매우 강력하다. 은행 생각하면 된다.

## 시도 빈도 제한

우회 가능한 각종 테크닉들도 존재

중심에서 0.3cm 떨어진 소속 18px

# Authentication

0. 사용자의 정보를 저장 – id / pw

## Hash the password

## Salt and Pepper

mypassword → mypassword_asdf1234

4cfd1dd64a0c3be2e4f3fa592e162f5ff818e8d86000ab7171d131b15bd12ae7

저장 방식에서 문제가 발생한 경우

중심에서 0.3cm 떨어진 소속 18px

# Authentication

1. 사용자가 로그인 요청을 form을 통해 보냄

사용자가 XSS 또는 Plugin에 노출

→ 관련 로그인 정보 탈취 가능

중심에서 0.3cm 떨어진 소속 18px

# Authentication

2. 네트워크를 통해 서버로 전달

## MitM

DNS spoofing

## Compromised Network

이번 웹 세미나의 범위를 벗어나기에 다루지 않음

MitM: Man in the middle

중심에서 0.3cm 떨어진 소속 18px

# Authentication

3. 서버에서 DB에 저장된 정보와 비교

## SQLi

## Logic Error

== in js

백엔드 서버 로직에서는 strict하게 값을 체크해야 한다.

중심에서 0.3cm 떨어진 소속 18px

# Authentication

4. 성공 여부를 토큰/세션으로 전달

## JWT

간편 but 잘못쓰면 위험

## Session

복잡 but 안전

곧 자세히 다룰 예정

중심에서 0.3cm 떨어진 소속 18px

# Authentication

5. 클라이언트에서 로그인 상태를 유지

## Cookie 형태로 세션을 저장

HttpOnly, Secure, SameSite

→ XSS으로 인한 탈취 방지

중심에서 0.3cm 떨어진 소속 18px

# Managing State

JWT

II

# Json-Web-Token

Concept

Json 형태의 토큰을 발행

header - payload - Signature 구성

각 부분은 base64로 인코딩

Firfox: Storage 〉 Cookies 에서 확인

중심에서 0.3cm 떨어진 소속 18px

# JWT

Header

alg(필수) : 암호화 방식

이 부분이 조작되어 성공적으로 들어가면 검증없이 사용된다.

kid : 서명한 key

fs에서 key를 찾는다면 path traversal 위협

jku : 공개키를 url에서 받아온다

공격자의 site에서 가져오게 조작 위협

kid: key id

jku: jwk set url

중심에서 0.3cm 떨어진 소속 18px

# JWT

Header

jwk : JWT 공개키

중간자 공격 취약

이 외에도 있지만, major한 것들은 이 정도

중심에서 0.3cm 떨어진 소속 18px

# JWT

Header - Protection

## 서버는 절대 Header를 믿지 말자

## 예시처럼 하드코딩

```
jwt.decode(token, key, algorithms=["RS256"])
```

## kid 값도 path traversal 조심

중심에서 0.3cm 떨어진 소속 18px

# JWT

Payload

## Payload

## 전달하고자 하는 데이터

중심에서 0.3cm 떨어진 소속 18px

# JWT

Signature

주어진 JWT가 조작되진 않았는지 확인

alg 기반으로 key를 사용해서 검증

중심에서 0.3cm 떨어진 소속 18px

# JWT

Signature – alg

## HMAC 계열

HS-256, HS-384, HS-512

## RSA 계열

RS-256, RS-384, RS-512, PS-256, …

## 그 외…

ES-256, EdDSA

중심에서 0.3cm 떨어진 소속 18px

# JWT

Signature – alg – Vulnerability

## alg: none

사용자가 alg를 none으로 → 믿으면 검증 없음

중심에서 0.3cm 떨어진 소속 18px

# JWT

Signature – alg – Vulnerability

## alg 혼용

RSA 계열과 HMAC 계열 혼용

RSA 공개키로 HMAC 키를 줘버린다

중심에서 0.3cm 떨어진 소속 18px

# Side Note.

Replay attack

Replay Attack

서버 입장에서는 문제가 없음

JWT의 취약점은 아님

중심에서 0.3cm 떨어진 소속 18px

# JWT

Conclusion

잘 쓰면 문제 없음

주로 개발자 실수로 구멍이 생긴다.

JWT의 header은 변조가 가능함에 유의

중심에서 0.3cm 떨어진 소속 18px

# Homework [KeyCat]

https://dreamhack.io/wargame/challenges/905

Hint: JWT 취약점

Python library를 활용하자

중심에서 0.3cm 떨어진 소속 18px

# Managing State

Session

III

# Session

Concept

사용자 정보를 서버에 저장

사용자 로그인시 메모리에 login을 저장

user에게는 login 상태를 기록한 값을 전달

Redis 같은 메모리 기반 non-relational sql을 사용

중심에서 0.3cm 떨어진 소속 18px

# Session

Concept

User는 임의의 key를 받으므로 변조 불가

중심에서 0.3cm 떨어진 소속 18px

# Session

Vulnerability

Session 구조는 안전

→ 검증 과정의 Logic error이 원인

또는 Framework의 CVE

단, session을 저장할 때 object sqli 주의

중심에서 0.3cm 떨어진 소속 18px

# Bonus Homework [Admin Feature]

https://dreamhack.io/wargame/challenges/2148

Hint: Redis에는 문제가 없다.

중심에서 0.3cm 떨어진 소속 18px

# Trivia

잡다한 지식들

## Remember_me Token에 static 값은 위험

암호화해도 로그인 페이지를 통해 대신 암호화 "해줘" 할 수 있다.

## 유저의 비번 초기화 / 변경 페이지

공격자가 dummy user를 활용해 다른 유저에 영향을 주지 않도록 해야한다.

중심에서 0.3cm 떨어진 소속 18px

# Cookies

Options

Secure : https 에서만 사용

Httponly : js는 쿠키 접근 불가

Samesite : Cross-Site에서 cookie 안 감

Strict, Lax, None

중심에서 0.3cm 떨어진 소속 18px

2000.00.00

# Q&A

## 질문이 있다면 하십시오

발표자

# 2.5cm-3.5cm 떨어진 제목 36px

제목 하단의 부제목 18px

## 3.5cm 떨어진 내용 1 32px

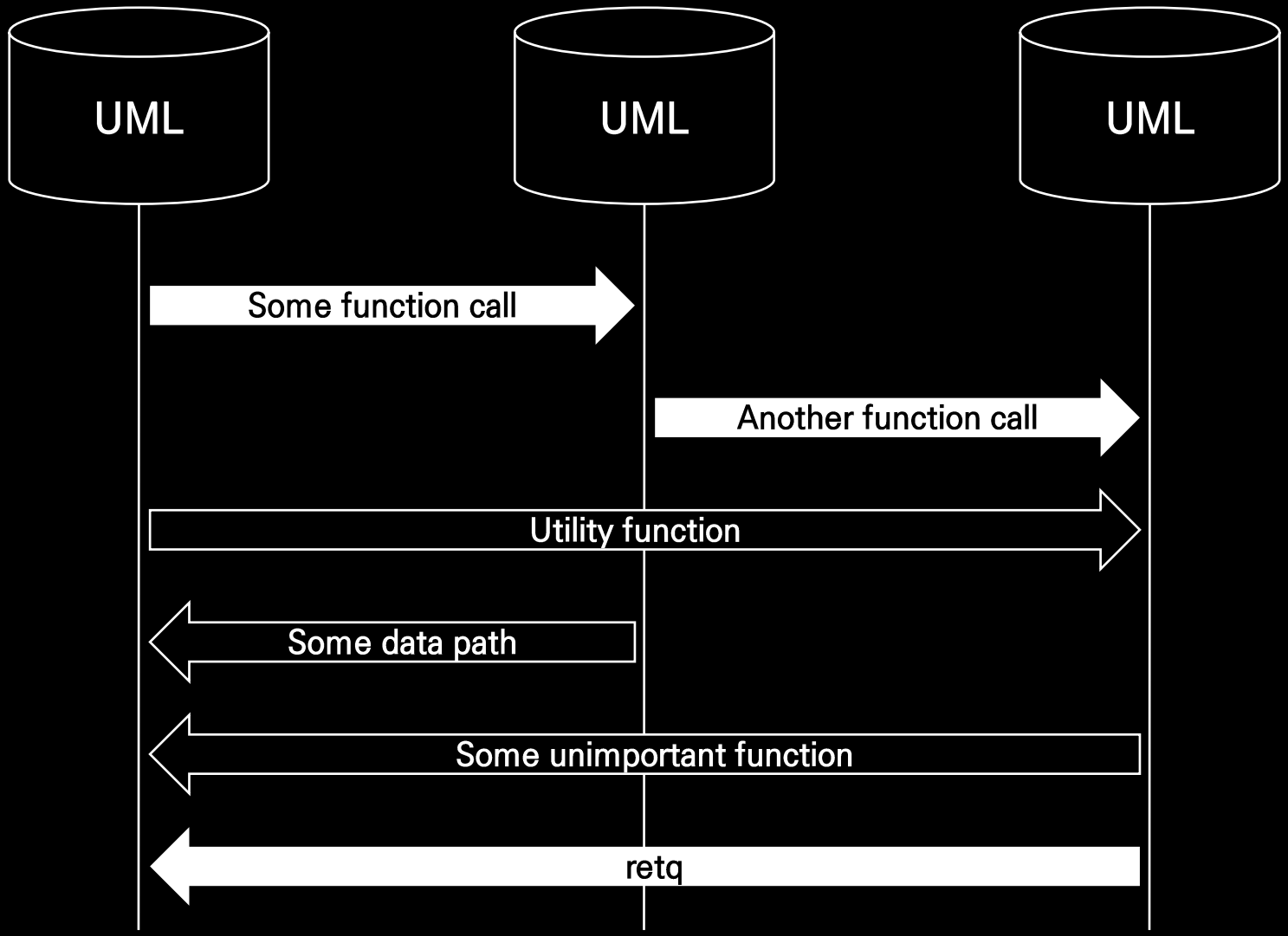
```
Git init
Git status
Git add text.txt
Git add .
Git commit
Ctrl+C
Git commit -m “genesis”
Git log
Git log --oneline
```

```
Git add .
Git reset .
Git commit -m “add README”
Git log --oneline -n 3
Git commit –a –m “hello”
```

1cm-1cm 떨어진 주석 12px

1cm-1cm 떨어진 주석 12px

중심에서 0.3cm 떨어진 소속 18px




중심에서 0.3cm 떨어진 소속 18px

# 2.5cm-2.5cm 떨어진 제목 36px

제목 하단의 부제목 18px

## 3.5cm 떨어진 내용 1 32px

좌측으로 0.5cm 떨어진 내용 하단의 설명 18px

## 3.5cm 떨어진 내용 2 32px

좌측으로 0.5cm 떨어진 내용 하단의 설명 18px

## 3.5cm 떨어진 내용 3 32px

좌측으로 0.5cm 떨어진 내용 하단의 설명 18px

1cm-1cm 떨어진 주석 12px

1cm-1cm 떨어진 주석 12px

중심에서 0.3cm 떨어진 소속 18px