1인 개발 서비스에서 발생하는 클라우드 보안 사고 예방

BCG LAB
장건희

발표할 내용

- 클라우드의 기본 보안 조치 및 세부 설명
- 소규모 프로젝트의 클라우드 보안 중요성

클라우드의 기본 보안 조치:

- 다중 인증 요소(MFA) 설정
- 웹 애플리케이션 방화벽(WAF) 도입
- 데이터 암호화 적용

MFA - Multi-Factor Authentication 다중 인증 요소

- 사용자가 시스템 또는 계정에 접근할 때 여러 단계의 인증을 필요로 하는 보안 절차
- 일반적으로 둘 이상을 조합하여 사용자를 확인
- **비밀번호 (Something you know):**
- 기존의 계정 암호나 비밀번호와 같은 지식 기반의 인증 요소
- **보안 토큰 (Something you have):**
- 사용자가 보유한 물리적인 장치나 앱을 통한 동적인 인증 코드를 사용하는 것으로, 주로 OTP(One-Time Password)가 해당.
- **바이오메트릭 데이터 (Something you are):**
- 생체 인식 정보를 사용하는 것으로, 지문, 홍채, 얼굴 인식 등이 해당.

웹 애플리케이션 방화벽(WAF)
Web Application Firewall

- 시스템을 보호, 애플리케이션의 취약성 감소
- **입력 필터링:**
- 악의적인 입력을 차단하고, SQL 삽입, 크로스사이트 스크립팅(XSS), 크로스사이트 요청 위조(CSRF) 등과 같은 공격을 방어.
- **패턴 인식:**
- 특정한 패턴이나 시그니처를 가진 악성 행위를 감지하고 차단.
- **액세스 제어:**
- 특정 IP 주소, 국가, 사용자 에이전트 등을 기반으로 액세스를 제어하거나 차단.
- **세션 관리:**
- 세션 하이재킹, 세션 고정, 세션 폐기 등과 같은 공격으로부터 세션을 보호.

발생 가능한 보안 사고: i

- 1인 개발 서비스의 취약성
- Firebase DB 권한 관리와 관련된 실제 사례 소개
- 보안뉴스 링크

발생 가능한 보안 사고: ii

- DB 계정 탈취와 관련된 위협
- 사용자 정보 노출과 관련된 리스크

실제 사례

보안뉴스 모바일 애플리케이션 백엔드 정보, 무방비로 노출되어 있다

모바일 애플리케이션과 연결된 데이터 저장소가 비밀번호도 없이 노출되어 있는 경우가 많아도 너무 많다는 조사 결과가 발표됐다. 클라우드 도입이 너무나 빠르게 진행되고 있어 사용자들이 익숙해질 새가 없었기 때문이다. 장점만큼 단점도 있다는 걸 잠깐 쉬어가며 생각할 차례다.

[보안뉴스 문가용 기자] 공개된 파이어베이스(Firebase) 데이터베이스를 간단히 검색했더니 2100개가 넘는 모바일 에플리케이션용 데이터저장소가 있는 그대로 노출되어 있다는 사실이 발견됐다. 이를 통해 기업의 민감한 정보인 은행 잔액, 가족 사진, 건강 관련 정보 등을 손쉽게 구할 수 있는 것으로 나타났다. 이러한 상황에 대해 보안 업체 체크포인트(Check Point)가 발표했다.

[이미지 = utoimage]

이 조사에서 문제가 되는 것으로 분석된 애플리케이션들의 종류는 1만 번도 다운로드 되지 않은 데이팅 앱에서부터 1천만 번 이상 설치된 유명 백화점 체인 앱까지 매우 다양했다. 이 유명 백화점 체인의 앱의 경우 API 게이트웨이 크리덴셜과 키 정보를 노출시키고 있었다. 그 외에 GPS 위치 정보, 사용자 심장 박동 수 등과 같은 건강 정보 등도 데이터베이스에 포함되어 있었다.

많은 개발자들과 기업들이 클라우드 네이티브 기술들을 도입하는 추세다. "하지만 이러한 최신 근황을 보안은 쫓아가지 못하고 있습니다. 클라우드로 이주하는 이유가 주로 생산성 향상인데, 생산성에 지나치게 초점을 맞추다 보니 보안은 우선순위에서 뒤로 밀리는 것이죠." 체크포인트의 위협 첩보 수장인 로템 핑켈스틴(Lotem Finkelsteen)의 설명이다. "그래서 온프레미스 개발 환경에 익숙한 개발자들은 클라우드의 각종 보안 사항들을 잊어버립니다."

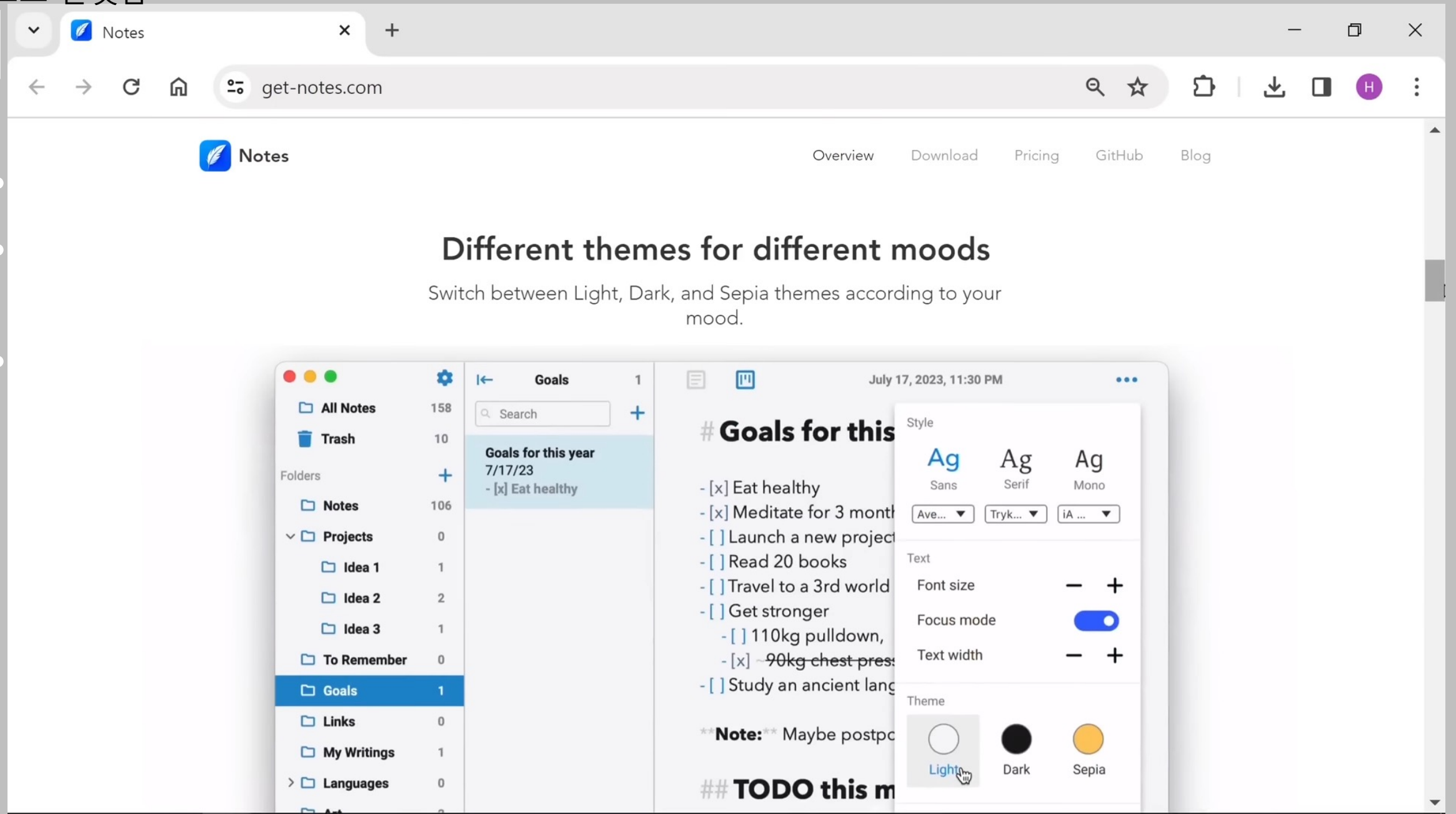
모바일 애플리케이션이 자유롭게 접근할 수 있도록 만들어진 데이터베이스 백엔드나 클라우드 서비스에서 정보가 보호되지 않은 채 노출되어 있는 사례는 꽤나 흔히 발견된다. AWS S3 버킷에서 수천만 건

대기업 서비스 vs. 1인 서비스 한계 및 관리 책임:

- 대기업의 경우: SECaaS(서비스형 보안) 채택
- 1인 서비스의 경우: 현실적으로 신경 쓰지 않는 상황 토이 프로젝트로 시작한 게임이 인기를 얻게 되는 경우
- 1인 서비스의 덩치가 커질 경우 보안 책임이 필요

Notes app : 사이드 프로젝트로 개발시작
- 크로스 플랫폼

대

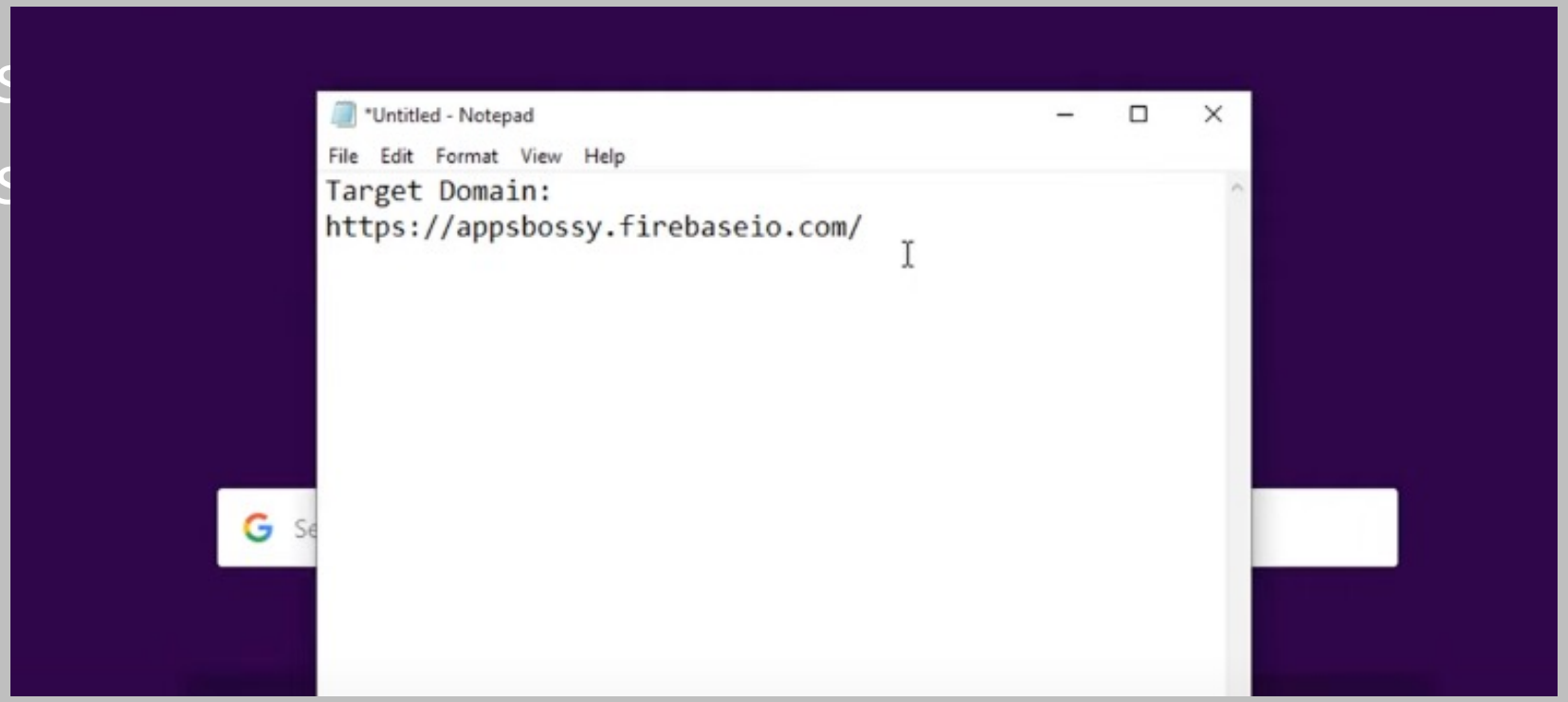


Firebase:1

- Firebase 권한 설정 잘못하면..
- Firebase 계정 액세스 및 권한 관리

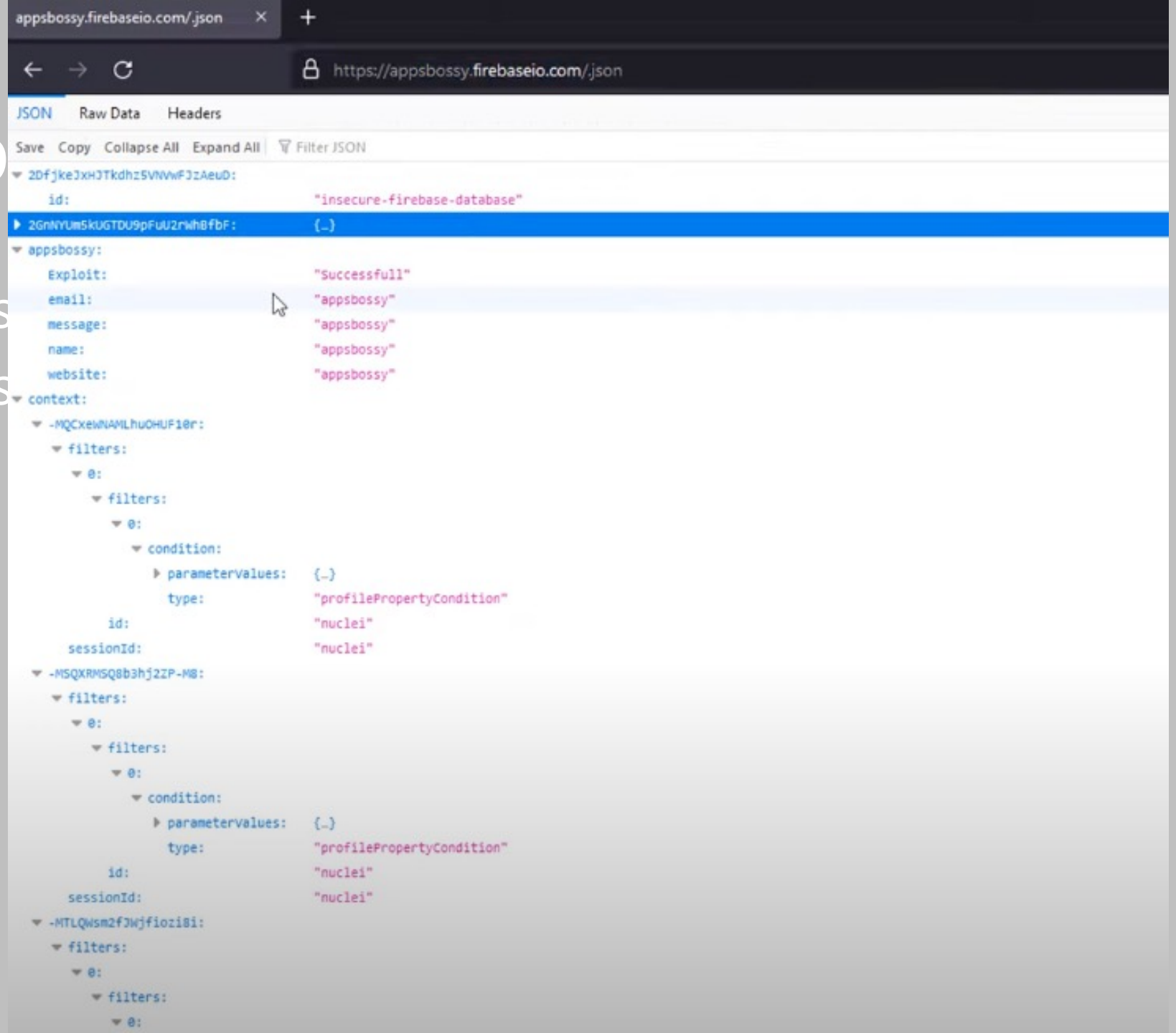
Firebase:2

- Firebas
- Firebas



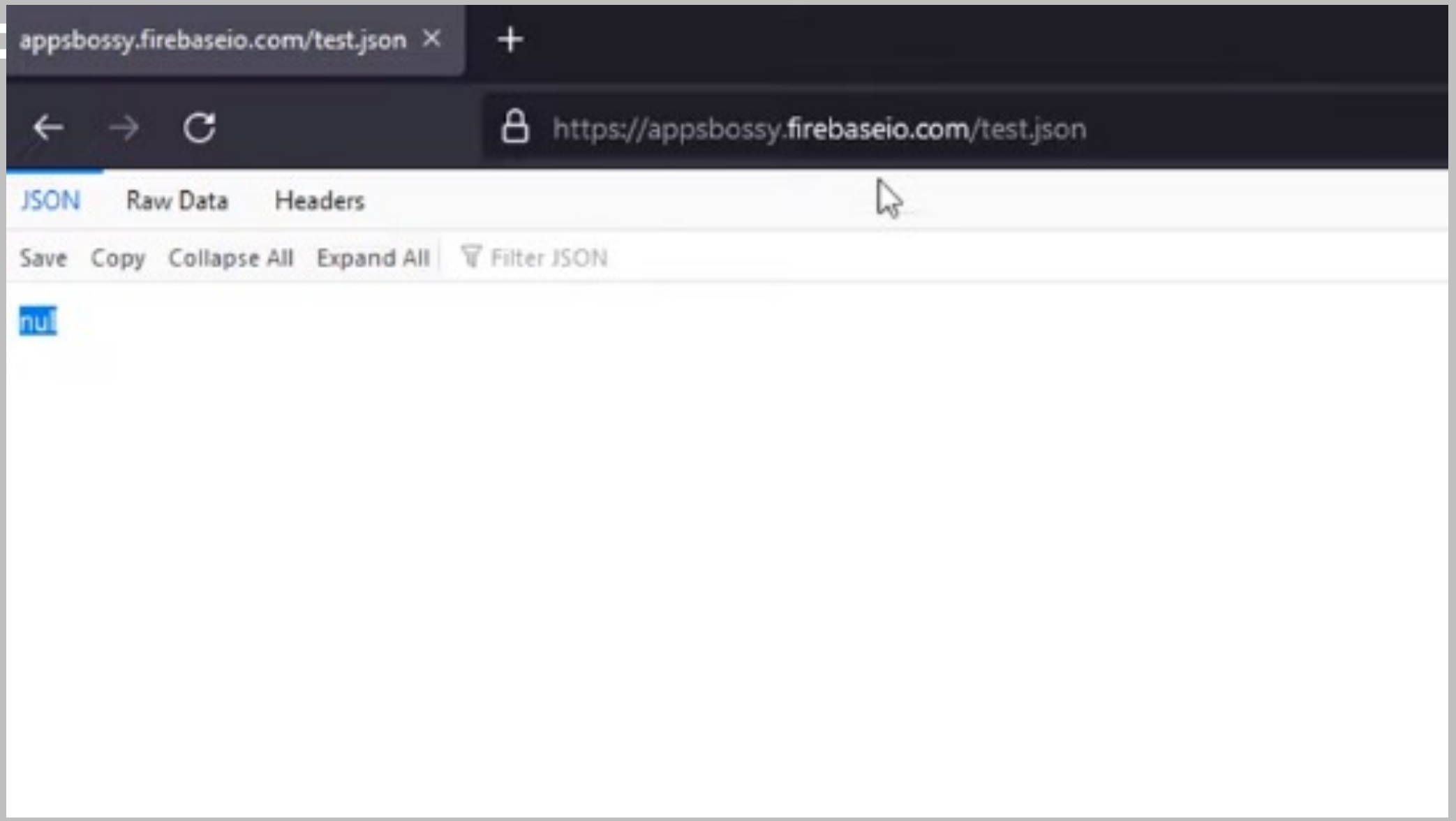
Fireb

- Firebas
- Firebas



F

-
-


appsbossy.firebaseio.com/test.json
https://appsbossy.firebaseio.com/test.json
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
null

CC

```python
def check_firebase_vulnerabilities(url):
    # Check for unauthenticated read access
    response = requests.get(url)
    if response.status_code == 200:
        print("Unauthenticated read access is possible.")
    else:
        print("Unauthenticated read access is not possible.")

    # Check for unauthenticated write access
    response = requests.put(url + "/test.json", json={"test": "test"})
    if response.status_code == 401:
        print("Unauthenticated write access is not possible.")
    else:
        print("Unauthenticated write access is possible.")

    # Check for insecure rules
    response = requests.put(url + "/.settings/rules.json", json={
        "rules": {
            ".read": "true",
            ".write": "true"
        }
    })
    if response.status_code == 200:
        print("Insecure rules can be set.")
    else:
        print("Insecure rules cannot be set.")

if __name__ == "__main__":
    url = input("Enter Firebase URL to check vulnerabilities and Exploit: ")
    check_firebase_vulnerabilities(url)
```

200 OK

401
Unauthorized

정리
1인 서비스 관리의 책임:

- 규모 상승 시 필요한 추가적인 보안 서비스 적용

마무리:

- 개발자를 위한 AWS 클라우드 보안 (1) - 클라우드 설계 원칙과 IAM
- https://tech.scatterlab.co.kr/aws-cloud-security-for-devs-1/
- **[Android + Firebase] 안드로이드 앱에 파이어베이스 추가하기**
- https://m.hanbit.co.kr/channel/category/category_view.html?cms_code=CMS3455201881
- 모바일 애플리케이션 백엔드 정보, 무방비로 노출되어 있다
- https://www.boannews.com/media/view.asp?idx=105481
- **Firebase database takeover vulnerability | firebase exploit**
- https://www.youtube.com/watch?v=lWowt5xixIs