Q&A 형식으로 알아보는 Apache log4j 취약점 대응 가이드

- 목 차 -

Q1. log4j 가 무엇인가요? ····· 2
Q2. log4j 1.x 버전에서도 영향을 미치나요? ····· 2
Q3. 취약한 log4j 를 사용하고 있는지 어떻게 확인하나요? ····· 2
Q4. log4j 의 버전 확인 방법은 무엇인가요? ····· 3
Q5. 버전에 따라 어떻게 조치해야 하나요? ····· 4
Q6. 보안 업데이트는 어떻게 하나요? ····· 4
Q7. 보안 업데이트를 하지 않으면 어떻게 되나요? ····· 5
Q8. 해당 취약점을 탐지할 수 있는 패턴은 어떻게 작성할 수 있을까요? ····· 5

2021.12

과학기술정보통신부
Ministry of Science and ICT
KISA 한국인터넷진흥원
KOREA INTERNET & SECURITY AGENCY

log4j 취약점 대응 FAQ

Q1. log4j가 무엇인가요?

○ log4j의 기능은 서비스 동작 과정에서 일어나는 일련의 모든 기록을 남겨 침해사고 발생 및 이상징후를 점검하기 위해 필수적으로 필요한 기능입니다. 무료로 제공되는 오픈소스 프로그램으로 Java 기반의 모든 어플리케이션에서 사용할 수 있습니다.

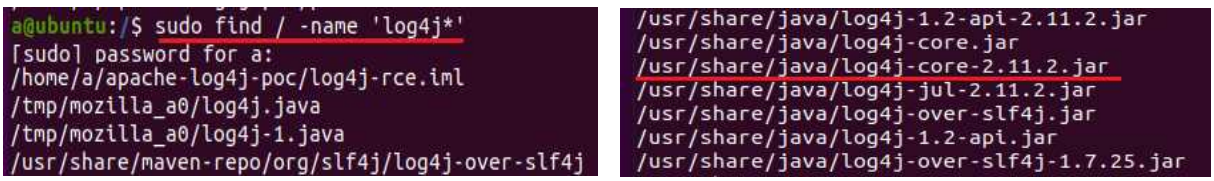
Q2. log4j 1.x 버전에서도 영향을 미치나요?

○ log4j 1.x 버전은 이미 2015년 이후 기술지원(업데이트)이 종료되었으므로 보안위협들에 노출될 가능성이 높습니다. 최신버전으로 업데이트 적용하기를 권고합니다.

Q3. 취약한 log4j를 사용하고 있는지 어떻게 확인하나요?

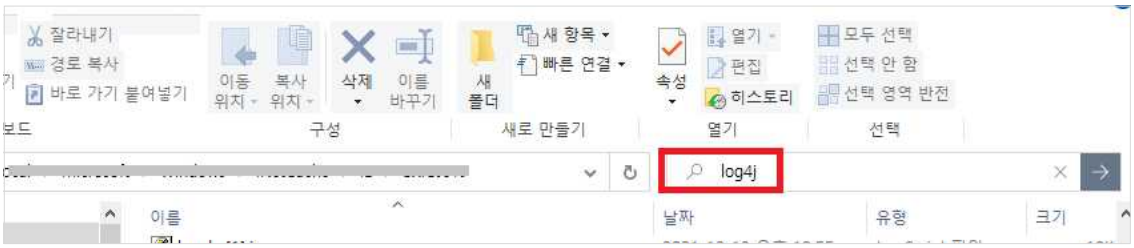
○ log4j를 사용하는 제품에 관한 정보는 아래 링크에서 확인할 수 있습니다.
- 상용제품 사용 중인 경우 아래 링크에 접속하여 해당 제조사를 찾아 제조사가 안내하는 방법으로 확인
  * https://gist.github.com/SwitHak/b66db3a06c2955a9cb71a8718970c592

○ log4j 설치 여부 확인(linux)
- dpkg –l | grep log4j
- find / -name 'log4j*'



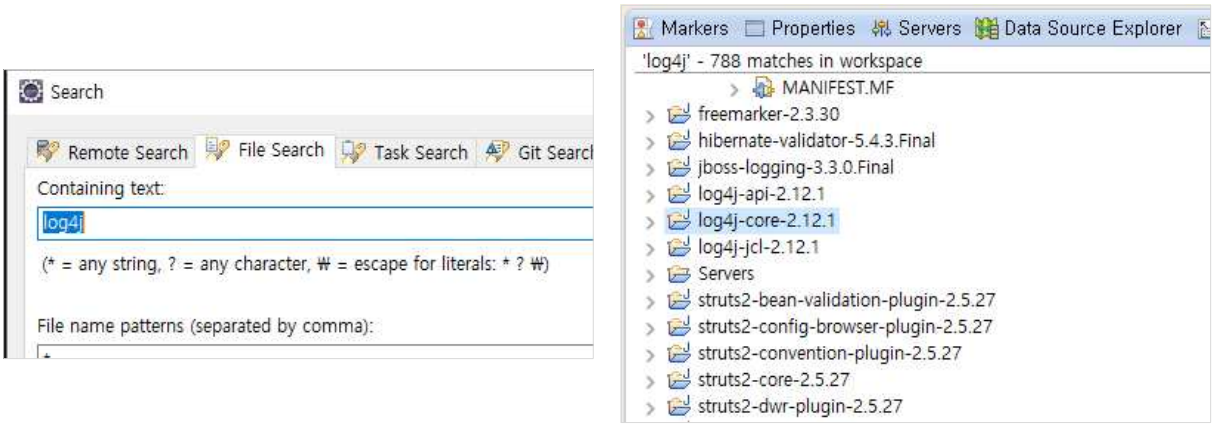
<linux에서 log4j 설치 여부 확인>

○ log4j 설치 여부 확인(Windows)
- window explorer의 검색 기능 (log4j 검색)을 이용



<windows explorer의 검색기능을 이용하여 검색>

- (참고) 도구를 활용하는 방법 – eclipse가 설치되어 있는 경우, eclipse의 찾기 기능 활용



<eclipse 도구의 '찾기' 기능을 이용하여 검색>

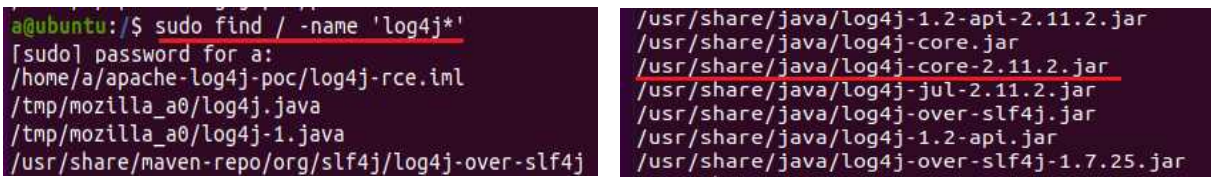
○ 참고 – 공개되어 있는 도구를 사용하는 방법

- Syft와 Grype 도구를 사용하는 방법
  * https://github.com/anchore/grype
- 취약점 스캐너를 사용하는 방법
  * 아래 링크에서 스캐너를 실행하여 취약 여부 확인
  * https://github.com/logpresso/CVE-2021-44228-Scanner

Q4. log4j의 버전 확인 방법은 무엇인가요?

○ log4j-core 버전 확인

- dpkg –l | grep log4j
- find / -name 'log4j*'



<linux에서 log4j-core 버전 확인>

○ Java Spring Framework Maven 사용 시 log4j가 설치된 경로의 pom.xml 파일을 열어 "**log4j-core**"로 검색

○ 검색결과 "<version>사용버전</version>" 으로 확인

```
<!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.15.0</version>
</dependency>
```

<log4j-core 버전 정보>

Q5. 버전에 따라 어떻게 조치해야 하나요?

○ **log4j 2.15버전 이상으로 업데이트를 수행**하여야 합니다.
- (최신버전 다운로드) https://logging.apache.org/log4j/2.x/download.html

○ 즉시 업데이트가 어려운 경우 log4j 버전에 따른 해결 방안은 아래와 같습니다.
- **2.0-beta9 ~ 2.10.0 : JndiLookup 클래스를 경로에서 제거**
  * zip –q –d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
- **2.10 ~ 2.14.1**
  * log4j2.formatMsgNoLookups 또는 LOG4J_FORMAT_MSG_NO_LOOKUPS 환경 변수를 true로 설정

Q6. 보안 업데이트는 어떻게 하나요?

[주의] 시스템 환경에 따라 다양한 라이브러리가 추가로 적용되어 있을 수 있으므로, 테스트 후 적용을 권고드립니다.

○ log4j-core.jar 파일 위치 및 파일 백업 후 jar 파일을 아래의 경로에서 다운로드
- (최신버전 다운로드) https://logging.apache.org/log4j/2.x/download.html

○ Java Spring Framework Maven을 사용하고 있는 경우 버전 정보 수정 후 재설치, 버전 재확인

① pom.xml을 다음과 같이 수정

```
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.14.1</version>
</dependency>
```

<취약한 버전의 pom.xml>

```
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.15.0</version>
</dependency>
```

<업데이트한 버전의 pom.xml>

② mvn install

③ ./mvnw dependency:list | grep log4j or sudo find / -name 'log4j*'로 버전 2.15.0 이상 여부 재확인

○ gradle을 사용하고 있는 경우 버전 정보 수정 후 재설치, 버전 재확인

①-1) build.gradle에서 log4j2.version 업데이트

```
ext['log4j2.version'] = '2.15.0'
```

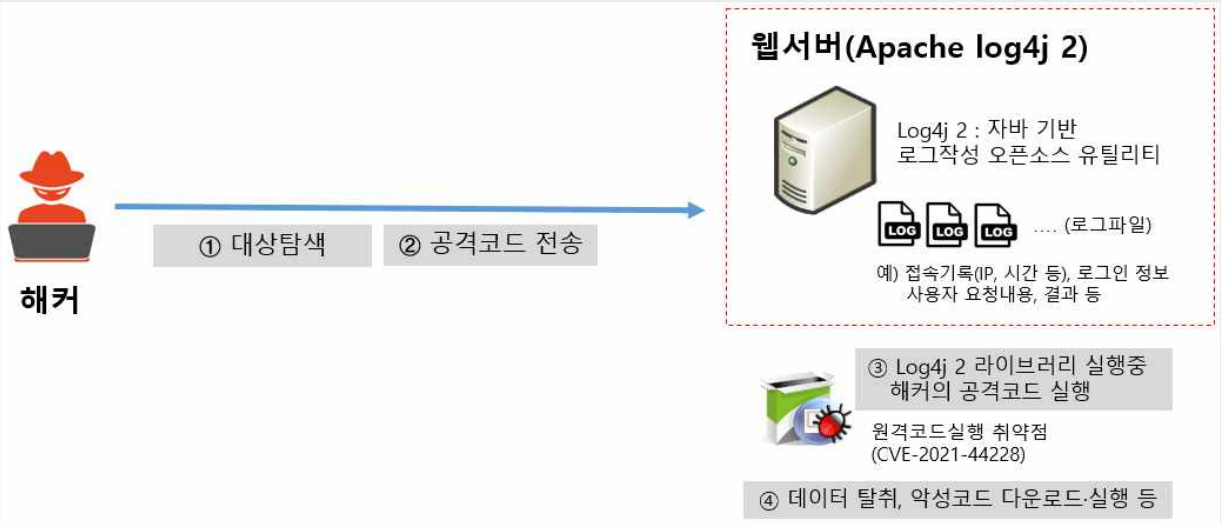
①-2) Gradle 플랫폼 지원을 사용하는 경우

```
implementation(platform("org.apache.logging.log4j:log4j-bom:2.15.0"))
```

② ./gradlew dependencyInsight —dependendy log4j-core로 버전 2.15.0 이상 여부 재확인

Q7. 보안 업데이트를 하지 않으면 어떻게 되나요?

○ 원격의 공격자가 이 취약점을 이용하여 악성코드 유포, 중요 데이터 탈취, 임의의 파일 다운로드 및 실행 등이 가능합니다.



<log4j 취약점 개요도>

Q8. 해당 취약점을 탐지할 수 있는 패턴은 어떻게 작성할 수 있을까요?

○ 아래 링크를 참조하여 log4j로 검색 후 탐지 정책을 확인할 수 있습니다.

- https://rules.emergingthreatspro.com/open/suricata-5.0/rules/emerging-exploit.rules

※ 본 탐지 정책은 내부 시스템 환경에 따라 다르게 동작할 수 있으며 시스템 운영에 영향을 줄 수 있으므로 **충분한 검토 후 적용 바랍니다**. 또한 **우회의 가능성이 있으므로 지속적인 업데이트가 필요함**을 알려드립니다.