

Elasticsearch 7.8.1 Log4j2 오류 해결

엘라스틱 서치 공식 답변

<https://discuss.elastic.co/t/apache-log4j2-remote-code-execution-rce-vulnerability-cve-2021-44228-esa-2021-31/291476>

Apache Log4j2 원격 코드 실행(RCE) 취약점 - CVE-2021-44228 -

■ 공지사항 ■ 보안 공지

Elasticsearch 완화 요약 매트릭스

참고: 아래의 완화는 완전한 것으로 간주되지만 전반적인 권장 사항은 버전 7.16.2 또는 6.8.22 이상으로 업데이트하는 것입니다.

예는 해당 버전이 해당 취약점의 영향을 받는다는 것을 나타내고, 아니오는 취약하지 않음을 나타냅니다. 버전 범위가 포함됩니다.

엘라스틱서치	JDK	CVE ID	정보 유출	원격 코드 실행	완전한 완화
7.16.1 - 7.16.2	≥ 8	CVE-2021-44228, CVE-2021-45046	아니	아니	N/A(취약하지 않음)
7.0.0 - 7.16.0	≥ 9	CVE-2021-44228, CVE-2021-45046	아니	아니	N/A ³ (취약하지 않음)
7.0.0 - 7.16.0	< 9	CVE-2021-44228, CVE-2021-45046	네	아니	시스템 속성 ¹
6.8.21	≥ 8	CVE-2021-44228, CVE-2021-45046	아니	아니	N/A(취약하지 않음)
6.0.0 - 6.8.20	≥ 9	CVE-2021-44228, CVE-2021-45046	아니	아니	N/A ³ (취약하지 않음)
6.4.0 - 6.8.20	< 9	CVE-2021-44228, CVE-2021-45046	네	아니	시스템 속성 ¹
6.0.0 - 6.3.2	< 9	CVE-2021-44228, CVE-2021-45046	네	아니	JndiLookup ² 제거
5.6.11 - 5.6.16	8	CVE-2021-44228, CVE-2021-45046	네	네	시스템 속성 ¹
5.0.0 - 5.6.10	8	CVE-2021-44228, CVE-2021-45046	네	네	JndiLookup ² 제거
< 5.0.0	어느	CVE-2021-44228, CVE-2021-45046	아니	아니	N/A(취약하지 않음)

¹ 각 노드에서 JVM 옵션 -Dlog4j2.formatMsgNoLookups=true를 설정하고 각 노드를 다시 시작합니다. 이것은 위에서 언급한 완전한 완화입니다. Elasticsearch에는 CVE-2021-45046의 스레드 컨텍스트 메시지 및 컨텍스트 조회에 대한 알려진 취약점이 없습니다.

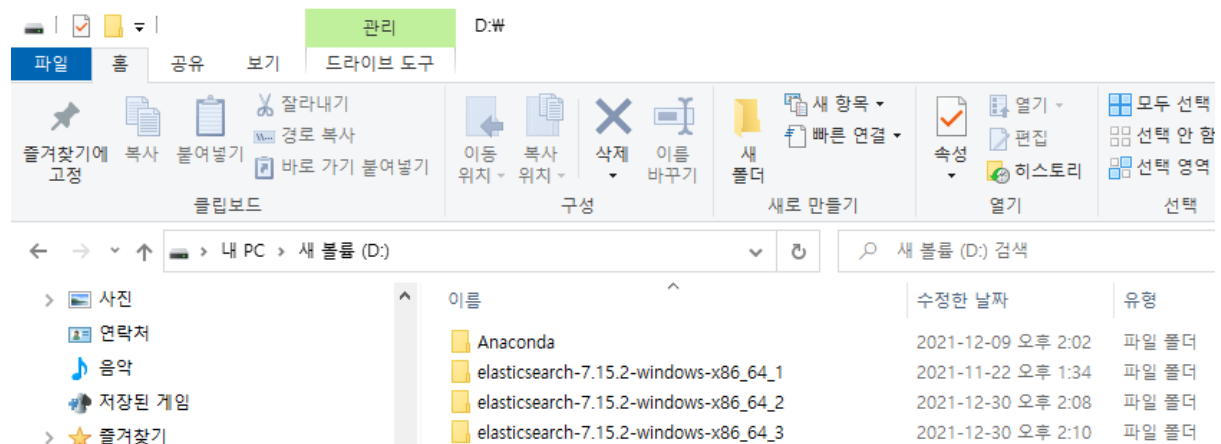
² Log4j 라이브러리에서 JndiLookup 클래스 제거. [여기에서 지침](#) 1.9k.

³ 이 구성은 취약하지 않으므로 완화가 필요하지 않습니다. Elasticsearch 6.4.0 이상에서는 여전히 주의 깊게 시스템 속성을 추가할 수 있습니다.

엘라스틱은 JAVA를 사용한 툴이라서 Log4j2오류를 가지고 있습니다.

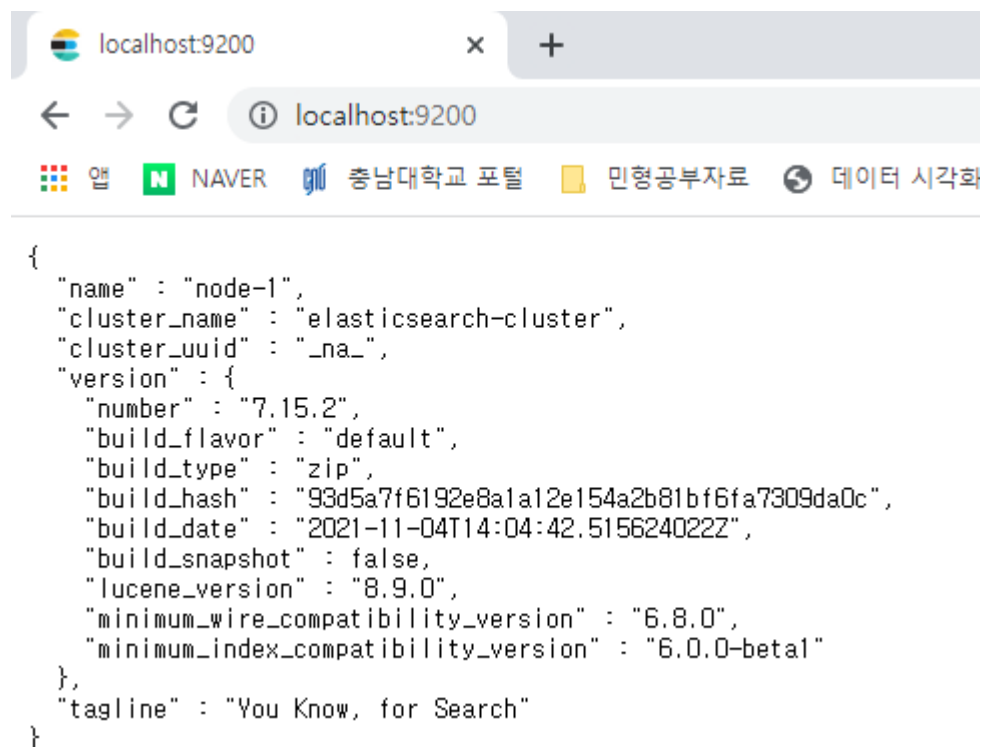
엘라스틱은 버전과 JDK 버전에 따라 해결방법이 다릅니다.

1.엘라스틱 버전 확인하는 방법



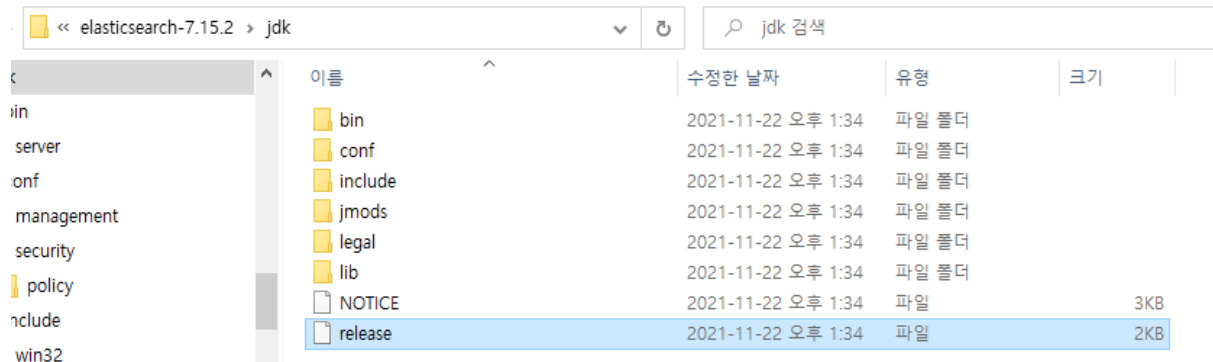
엘라스틱서치를 설치한 파일을 확인하면 어떤 엘라스틱서치 버전을 설치했는지 확인 가능합니다. 저희 경우는 7.15.2버전을 설치했군요.

다른 방법은,



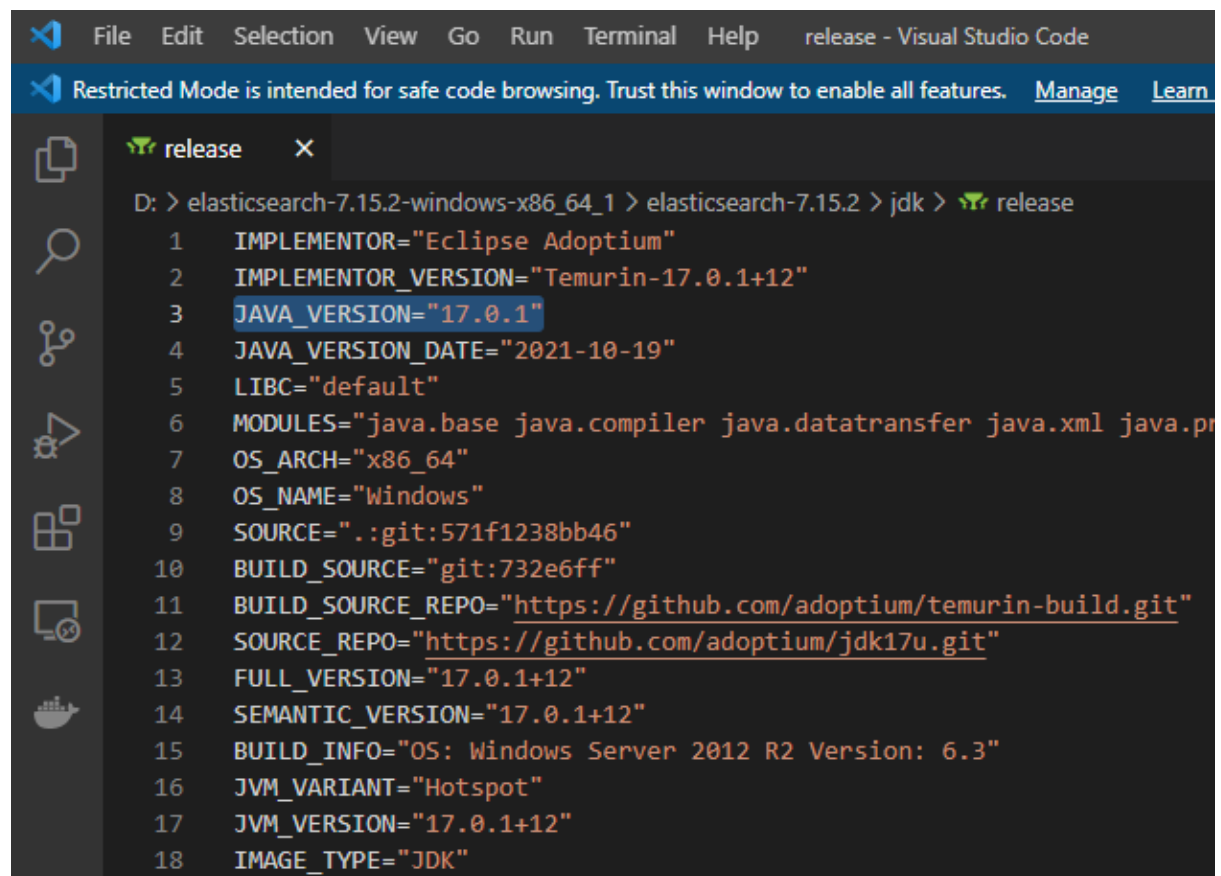
엘라스틱서치의 url, 저의 경우는 localhost:9200으로 들어가면 엘라스틱서치 버전을 확인할 수 있습니다. 제가 설치한 엘라스틱서치 버전이 7.15.2인 것을 알 수 있습니다.

2.JDK 버전 확인하는 방법



이름	수정한 날짜	유형	크기
bin	2021-11-22 오후 1:34	파일 폴더	
conf	2021-11-22 오후 1:34	파일 폴더	
include	2021-11-22 오후 1:34	파일 폴더	
jmods	2021-11-22 오후 1:34	파일 폴더	
legal	2021-11-22 오후 1:34	파일 폴더	
lib	2021-11-22 오후 1:34	파일 폴더	
NOTICE	2021-11-22 오후 1:34	파일	3KB
release	2021-11-22 오후 1:34	파일	2KB

elasticsearch > jdk > release 들어가서 확인하면 JAVA_VERSION을 확인할 수 있습니다.



```
D: > elasticsearch-7.15.2-windows-x86_64_1 > elasticsearch-7.15.2 > jdk > release
1  IMPLEMENTOR="Eclipse Adoptium"
2  IMPLEMENTOR_VERSION="Temurin-17.0.1+12"
3  JAVA_VERSION="17.0.1"
4  JAVA_VERSION_DATE="2021-10-19"
5  LIBC="default"
6  MODULES="java.base java.compiler java.datatransfer java.xml java.p
7  OS_ARCH="x86_64"
8  OS_NAME="Windows"
9  SOURCE=".:git:571f1238bb46"
10 BUILD_SOURCE="git:732e6ff"
11 BUILD_SOURCE_REPO="https://github.com/adoptium/temurin-build.git"
12 SOURCE_REPO="https://github.com/adoptium/jdk17u.git"
13 FULL_VERSION="17.0.1+12"
14 SEMANTIC_VERSION="17.0.1+12"
15 BUILD_INFO="OS: Windows Server 2012 R2 Version: 6.3"
16 JVM_VARIANT="Hotspot"
17 JVM_VERSION="17.0.1+12"
18 IMAGE_TYPE="JDK"
```

저의 JAVA_VERSION 즉, JDK는 17.0.1 입니다.

따라서 저희 JDK는 9이상입니다.

Apache Log4j2 원격 코드 실행(RCE) 취약점 - CVE-2021-44228 - ■ 공지사항 ■ 보안 공지

Elasticsearch 완화 요약 매트릭스

참고: 아래의 완화는 완전한 것으로 간주되지만 전반적인 권장 사항은 버전 7.16.2 또는 6.8.22 이상으로 업데이트하는 것입니다.

예는 해당 버전이 해당 취약점의 영향을 받는다는 것을 나타내고, 아니오 는 취약하지 않음을 나타냅니다. 버전 범위가 포함됩니다.

엘라스틱서치	JDK	CVE ID	정보 유출	원격 코드 실행	완전한 완화
7.16.1 - 7.16.2	≥ 8	CVE-2021-44228, CVE-2021-45046	아니	아니	N/A(취약하지 않음)
7.0.0 - 7.16.0	≥ 9	CVE-2021-44228, CVE-2021-45046	아니	아니	N/A ³ (취약하지 않음)
7.0.0 - 7.16.0	< 9	CVE-2021-44228, CVE-2021-45046	네	아니	시스템 속성 ¹
6.8.21	≥ 8	CVE-2021-44228, CVE-2021-45046	아니	아니	N/A(취약하지 않음)
6.0.0 - 6.8.20	≥ 9	CVE-2021-44228, CVE-2021-45046	아니	아니	N/A ³ (취약하지 않음)
6.4.0 - 6.8.20	< 9	CVE-2021-44228, CVE-2021-45046	네	아니	시스템 속성 ¹
6.0.0 - 6.3.2	< 9	CVE-2021-44228, CVE-2021-45046	네	아니	JndiLookup ² 제거
5.6.11 - 5.6.16	8	CVE-2021-44228, CVE-2021-45046	네	네	시스템 속성 ¹
5.0.0 - 5.6.10	8	CVE-2021-44228, CVE-2021-45046	네	네	JndiLookup ² 제거
< 5.0.0	어느	CVE-2021-44228, CVE-2021-45046	아니	아니	N/A(취약하지 않음)

¹ 각 노드에서 JVM 옵션 -Dlog4j2.formatMsgNoLookups=true를 설정하고 각 노드를 다시 시작합니다. 이것은 위에서 언급한 완전한 완화입니다. Elasticsearch에는 CVE-2021-45046의 스레드 컨텍스트 메시지 및 컨텍스트 조회에 대한 알려진 취약점이 없습니다.

² Log4j 라이브러리에서 JndiLookup 클래스 제거. [여기에서 지침](#) 1.9k.

³ 이 구성은 취약하지 않으므로 완화가 필요하지 않습니다. Elasticsearch 6.4.0 이상에서는 여전히 주의 깊게 시스템 속성을 추가할 수 있습니다.

따라서 저희 상황은 노란색에 해당됩니다.

버전이 낮은 건, Log4j2가 아니라 Log4j1을 사용해서 문제가 되지 않습니다.

N/A (취약하지 않음)³ 에 해당이 되어서 완화가 필요하지 않습니다.

하지만 시스템 속성¹과 JndiLookup²제거도 알아보도록 하겠습니다=)

1.시스템 속성¹

elasticsearch-7.15.2 > config >					config 검색	
이름	수정한 날짜	유형	크기			
jvm.options.d	2021-11-04 오후 2:06	파일 폴더				
user_dic	2021-12-03 오후 5:06	파일 폴더				
elasticsearch.keystore	2021-11-22 오후 1:52	KEYSTORE 파일	1KB			
elasticsearch.yml	2022-01-05 오후 2:53	Yaml 원본 파일	3KB			
jvm.options	2022-02-05 오후 10:17	OPTIONS 파일	4KB			
log4j2.properties	2021-11-22 오후 1:34	Properties 원본 파...	19KB			
role_mapping.yml	2021-11-22 오후 1:34	Yaml 원본 파일	1KB			
roles.yml	2021-11-22 오후 1:34	Yaml 원본 파일	1KB			
users	2021-11-22 오후 1:34	파일	0KB			
users_roles	2021-11-22 오후 1:34	파일	0KB			

elasticsearch>config>jvmoptions 에 들어갑니다.

```
≡ jvm.options X
D: > elasticsearch-7.15.2-windows-x86_64_1 > elasticsearch-7.15.2 > config > ≡ jvm.options
64  ## JVM temporary directory
65  -Djava.io.tmpdir=${ES_TMPDIR}
66  -Dlog4j2.formatMsgNoLookups=true
67
```

JVM temporary directory 밑에

-Dlog4j2.formatMsgNoLookups=true 을 넣어줍니다.

그 다음 Elasticsearch 서비스를 재실행하면 간단하게 적용이 완료됩니다.

N/A (취약하지 않음)³ 에 해당이 되어도 시스템 속성¹은 적용하는 것이 보안상 좋다고는 합니다. (<https://shinwusub.tistory.com/m/147>)

2021.12.12 03:54:27

최대의 보안사고가 터지려나 봅니다

토비리

조회 수 5912



최대의 보안사고 중 하나가 터지는 것 같군요. CVE-2021-44228 혹은 Log4Shell 이라는 취약점으로, log4j 라는 모듈 때문인데, Apache 서버를 사용하는 분들은 우선 조심하셔야겠습니다.

apache-log4j1.2 혹은 apache-log4j2 라는 것이 설치되어 있으면 영향을 받는 것 같습니다.

예를들면, Ubuntu/Debian 서버라면

```
% apt list --installed | grep "log4j"
```

```
% dpkg -l | grep log4j (또는)
```

해서 목록이 뜨는지 체크해보세요. 만약 해당사항이 있다면 추가 조치가 필요합니다.

(<https://xetown.com/topics/1636343>)

```
root@DESKTOP-G9DVJ1H: /mnt/c/Users/MongTa# df -h -T
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/sdd         ext4      251G  1.2G  237G   1% /
tmpfs            tmpfs      9.4G  365M   9.0G   4% /mnt/wsl
/dev/sdc         ext4      251G  11G   228G   5% /mnt/wsl/docker-desktop-data/isocache
none            tmpfs      9.4G   16K   9.4G   1% /mnt/wsl/docker-desktop/shared-sockets/host-services
/dev/sdb         ext4      251G  123M  239G   1% /mnt/wsl/docker-desktop/docker-desktop-proxy
/dev/loop0      iso9660    334M  334M   0 100% /mnt/wsl/docker-desktop/cli-tools
tmpfs            tmpfs      9.4G  392K   9.4G   1% /mnt/wsl/docker-desktop-data/tarcache/entries/docker.tar/493d8ec8dece3d01
2df0d53051d65b1515e457a5263c9dc79019709357534026/containers/services/docker/tmp
overlay          overlay    9.4G  392K   9.4G   1% /mnt/wsl/docker-desktop-data/tarcache/entries/docker.tar/493d8ec8dece3d01
2df0d53051d65b1515e457a5263c9dc79019709357534026/containers/services/docker/rootfs
tools            9p        465G  118G  347G  26% /init
none            devtmpfs   9.4G   0   9.4G   0% /dev
none            tmpfs      9.4G   0   9.4G   0% /run
none            tmpfs      9.4G   0   9.4G   0% /run/lock
none            tmpfs      9.4G   0   9.4G   0% /run/shm
none            tmpfs      9.4G   0   9.4G   0% /run/user
tmpfs            tmpfs      9.4G   0   9.4G   0% /sys/fs/cgroup
drivers          9p        465G  118G  347G  26% /usr/lib/wsl/drivers
lib              9p        465G  118G  347G  26% /usr/lib/wsl/lib
C:\              9p        465G  118G  347G  26% /mnt/c
D:\              9p        1.9T   21G  1.8T   2% /mnt/d
root@DESKTOP-G9DVJ1H: /mnt/c/Users/MongTa# cd /mnt/d
root@DESKTOP-G9DVJ1H: /mnt/d#
```

```
df -h -T
cd /mnt/d
```

```
ls
cd elasticsearch-7.15.2-windows-x86_64_1
dpkg -l | grep log4j
```


log4j-core-2.11.1.jar는 elasticsearch>lib안에 있습니다.

log4j-core-2.11.1.jar가 elasticsearch>lib 안에 있는지 확인을 해주세요.

혹은 리눅스 검색어로

```
ls -l lib/log4j-core-*.jar
```

을 검색했을 때, 출력이 파일 목록이면 올바른 디렉토리에 있는 것입니다.

다음을 사용하여 취약한 log4j JAR 파일을 백업하십시오.

```
zip ./backup-log4j.zip lib/log4j-core-*.jar
```

이 백업 파일은 후속 단계에서 실수를 하거나 업데이트된 JAR 파일에 문제가 발생한 경우 유용할 수 있습니다. Elasticsearch 노드가 올바르게 작동하고 있다고 확신하면 안전하게 삭제할 수 있습니다.

참고 : 위의 명령을 작성된 대로 따르는 것이 중요합니다. jar 파일을 백업하는 다른 방법은 실행 중에 Elasticsearch 클래스 경로에 포함되어 노드가 올바르게 시작되지 않을 수 있는 위험이 있습니다.

다음을 사용하여 log4j JAR 파일에서 취약한 클래스를 제거하십시오.

```
zip -d lib/log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
```

출력이 `deleting: org/apache/logging/log4j/core/lookup/JndiLookup.class` 이면 명령이 성공한 것입니다.

출력이 `zip error: Nothing to do!` 이면 클래스가 이미 삭제되었음을 의미합니다.

다른 모든 오류는 문제를 나타내는 것이며 단계를 검토해야 합니다.

참고 : 위의 명령을 실행하면 다음과 같은 일련의 줄이 생성됩니다.

```
zip warning: Local Version Needed To Extract does notmatch CD: org/apache/logging/log4j/core/util/WatchManager$WatchRunnable.class
```

다음을 사용하여 취약한 클래스 제거를 확인합니다.

```
jar tvf lib/log4j-core-*.jar | grep -i JndiLookup
```

제거가 성공적이면 출력이 없습니다.

출력에 JAR 파일의 항목 목록이 포함되어 있으면 제거에 실패한 것입니다. 오류 메시지가 있는지 이전 단계를 검토하십시오.

이 단계가 끝나면, Elasticsearch 노드 다시 시작합니다.

그리고 클러스터의 모든 Elasticsearch 노드에 대해 이 단계를 반복합니다.

참조:

아래의 방법들은 Elasticsearch와 관련없는 Log4j 보안 취약점 동작 원리 및 조치 방법에서도 찾을 수 있는 조치 방법들입니다.

(<https://devocean.sk.com/blog/techBoardDetail.do?ID=163523>)

조치 방법

이 문제는 Log4j 버전 2.0-beta-9 부터 2.14.1 사이를 사용하는 경우에 해당 됩니다. 글을 작성하는 21년 12월 13일 기준에서는, 관련 취약점이 조치된 [Log4j v2.15.0 버전](#)을 다운로드 받아 조치가 가능했으나, 2.15 버전도 완벽한 해결이 된 버전이 아니라는 이유로 12월 14일 기준 [Log4j v.2.16.0 버전](#)이 릴리즈 되었습니다.

문제점을 해결할 수 있는 방법은 크게 3가지가 있는데,

첫번째 방법은, 관련 문제가 해결된 버전으로 위 주소의 log4j 버전으로 업그레이드 하는 방식 입니다. 하지만, log4j 는 단독으로 동작하는 라이브러리가 아니고, Elasticsearch, Hadoop, Druid 등 자바 기반의 큰 프로젝트 안에 포함되는 로깅용 라이브러리 이기 때문에 단순히 log4j 버전만 올리는 경우 전체 프로젝트 빌드 과정에서 에러가 발생할 수 있습니다. 따라서, 이 방법의 경우에는 일부 에러가 발생하는 부분의 코드 수정이 필요합니다.

두번째 방법은, 현재 사용중인 log4j jar 파일 안에 있는 JndiLookup.class 파일을 삭제해 버리는 것 입니다. 2

```
zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
```

위와 같이 삭제를 실행할 수 있는데, 이후에는 당연히 JndiLookup 기능이 동작하지 않게 될 것 입니다. 실제로 이 방법으로 조치를 한 경우 로그에 WARN 이 발생하기도 했습니다. (Hive 의 경우)

마지막 방법은, 사용중인 log4j 의 버전이 2.10 부터 2.14.1 까지의 경우에만 적용이 가능한 방법입니다. 1

설정 파일에 아래와 같은 설정 두가지 중 하나를 넣어주시면 됩니다.

```
log4j2.formatMsgNoLookups=true
```

혹은

```
LOG4J_FORMAT_MSG_NO_LOOKUPS=true
```

두개의 설정은 같은 의미를 같습니다.



log4j 가 매우 널리 사용되는 라이브러리 이기 때문에, 상당히 많은 시스템에 영향이 있을거라 생각이 되는데, 시스템 담당자라면 빠른 확인과 조치가 필요하겠습니다.

elasticsearch 공식홈페이지에서 나와있는 방법과 거의 비슷한 방법으로 Log4j2를 해결하고 있음을 알 수 있습니다.