오즈 Log4j 2.25.4 패치 방법

2026-04-24

목차

1. 개요 2
2. 오즈 서버 log4j 패치 3
3. 오즈 스케줄러 log4j 패치 4
4. 디버깅 7
5. 오즈 API 8

1. 개요

◎ 아파치 보안 취약점 및 최신 log4j

- 아파치 보안 취약점 : https://logging.apache.org/log4j/2.x/security.html
- 최신 log4j 다운로드 : https://logging.apache.org/log4j/2.x/download.html

◎ 오즈 대상제품

- OZ Server
- OZ Scheduler Server

◎ 오즈 log4j 패치 파일

- 다운로드 : https://www.oztn.net/ozguide/oz_log4j/log4j-2.25.4_oz/

패치 파일	설명
log4j-api-2.25.4.jar log4j-core-2.25.4.jar ozlog4j2.11.jar ozlogextra.jar	▪ 2026-04-24 기준 보안 취약점 개선된 최신 버전 ▪ jre 1.8 이상 지원

◎ 내용

오즈 서버, 스케줄러에 배포되는 기본 log4j는 릴리즈 시점에 따라 달라집니다. 제품 안정성을 위해 항상 최신 log4j로 릴리즈 되는 것은 아니기 때문에 임의로 최신 log4j를 받아 적용하실 수 있습니다.

모든 log4j 유형에 대한 사항은 🔗가이드 문서를 참고하시기 바랍니다.

이 문서는 2026년 4월 기준 최신 log4j인 2.25.4 버전 적용 방법에 대한 가이드 문서입니다.

2. 오즈 서버 log4j 패치

① 오즈 서버가 사용하는 WAS의 **java 버전이 1.8 이상**인지 확인합니다.

② oz80/WEB-INF/lib/ 폴더에 다음과 같은 **이전 버전 log4j 관련 파일을 삭제**합니다.

- ✓ log4j.jar
- ✓ log4j-api-*.jar, log4j-core-*.jar, ozlog4j2.jar
- ✓ logback-*.jar, ozlogback.jar

③ http://www.oztn.net/ozguide/oz_log4j/log4j-2.25.4_oz/ 에서 받은 다음 파일을 **oz80/WEB-INF/lib에 업로드** 합니다.

- ✓ log4j-api-2.25.4.jar
- ✓ log4j-core-2.25.4.jar
- ✓ ozlog4j2.11.jar

④ **oz80/WEB-INF/conf/log.properties** 의 레이아웃을 ISO8601에서 **DEFAULT로 변경**합니다.

레이아웃을 변경하지 않으면 날짜와 시간 사이에 '**T**' 글자가 표시됩니다.
: 2021-12-20**T**20:37:29,053 [http-nio-80-exec-3] INFO

- 기존

```
ROLLING_FILE.Type=RollingFile
ROLLING_FILE.Layout=%d{ISO8601} [%t] %-5p %c{1} %x - %m%n
```

- 변경

```
ROLLING_FILE.Type=RollingFile
ROLLING_FILE.Layout=%d{DEFAULT} [%t] %-5p %c{1} %x - %m%n
```

⑤ 오즈 서버를 재시작 하면 설정이 반영됩니다. 보고서가 실행되는지 확인합니다.

⑥ oz80/WEB-INF/logs/server.log 에 ready up 이란 문구가 나오면 서버가 정상적으로 구동된 것입니다. (단, log.properties 의 Priority=INFO 또는 DEBUG 인 경우)

3. 오즈 스케줄러 log4j 패치

HTML5(SVG) 방식에 사용되는 **OZ HTML5 Manager**도 Scheduler와 같은 방식으로 패치합니다.

- {oz스케줄러폴더} : 보통 OZ Scheduler 8.0, ozsch80, ozmng80 와 같은 폴더 경로입니다.

<Window/Unix/Linux 공통>

① {oz스케줄러폴더}/lib 폴더에 **이전 버전 log4j 관련 파일을 삭제**합니다.

- ✓ log4j.jar
- ✓ log4j-api-*.jar, log4j-core-*.jar, ozlog4j2.jar
- ✓ logback-*.jar, ozlogback.jar

② http://www.oztn.net/ozguide/oz_log4j/log4j-2.25.4_oz/ 에서 받은 **다음 파일들을 {oz스케줄러폴더}/lib에 복사**합니다.

- ✓ log4j-api-2.25.4.jar
- ✓ log4j-core-2.25.4.jar
- ✓ ozlog4j2.11.jar

③ {oz스케줄러폴더}/conf/scheduler_server_log.properties 의 레이아웃을 ISO8601에서 **DEFAULT**로 변경합니다.

레이아웃을 변경하지 않으면 날짜와 시간 사이에 'T' 글자가 표시됩니다.
: 2021-12-20**T**10:51:34,909 [Thread-7] DEBUG

- 기존

```
ROLLING_FILE.Type=RollingFile
ROLLING_FILE.Layout=%d{ISO8601} [%t] %-5p %c{1} %x - %m%n
```

- 변경

```
ROLLING_FILE.Type=RollingFile
ROLLING_FILE.Layout=%d{DEFAULT} [%t] %-5p %c{1} %x - %m%n
```

<Windows 서비스 구동 방식>

위의 공통 부분 설정 후 다음과 같이 조치합니다.

① {oz스케줄러폴더}/bin/**OZSchedulerInstallService.bat** 을 텍스트 편집기로 열고, jre버전이 1.8 이상인지 확인합니다.

```
set JRE_PATH=C:₩Program Files (x86)₩Java₩jre1.8.0_251₩bin₩client₩jvm.dll
set OZSCHEDULER_HOME=..₩
set OZSCHEDULER_HOME_BIN=C:₩Program Files (x86)₩Forcs₩OZ Enterprise Server 8.0₩OZ Scheduler 8.0₩bin
```

② {oz스케줄러폴더}/bin/**OZSchedulerInstallService.bat** 의 이전 log4j 관련 라이브러리를 **삭제**합니다.

- ✓ log4j.jar
- ✓ log4j-api-*.jar, log4j-core-*.jar, ozlog4j2.jar
- ✓ logback-*.jar, ozlogback.jar

```
set OZSchLib=%OZSchLib%;%OZSCHEDULER_HOME%₩lib₩log4j.jar
```

③ {oz스케줄러폴더}/bin/**OZSchedulerInstallService.bat** 의 신규 log4j 관련 라이브러리를 아래와 같이 **추가**합니다.

- ✓ log4j-api-2.25.4.jar
- ✓ log4j-core-2.25.4.jar
- ✓ ozlog4j2.11.jar

```
rem -------------------------------------------------
rem             Library for Log4j 2.25.4
rem -------------------------------------------------
set OZSchLib=%OZSchLib%;%OZSCHEDULER_HOME%₩lib₩log4j-api-2.25.4.jar
set OZSchLib=%OZSchLib%;%OZSCHEDULER_HOME%₩lib₩log4j-core-2.25.4.jar
set OZSchLib=%OZSchLib%;%OZSCHEDULER_HOME%₩lib₩ozlog4j2.11.jar
```

④ bin 폴더의 OZSchedulerUninstallService.bat, OZSchedulerInstallService.bat 순서대로 관리자 권한으로 실행합니다.

⑤ 서비스에서 OZScheduler를 시작합니다.

⑥ {oz스케줄러폴더}/logs/OZSchedulerMainLog.log 에 ready up 문구가 나오면 정상적으로 구동된 것입니다.

참고로, 서비스를 이용하지 않고 배치 파일로 직접 실행하는 경우 OZSchedulerInstallService.bat 대신 scheduler.bat, scheduler_stop.bat 파일을 수정하시면 됩니다.

<Unix,Linux 환경>

위의 공통 부분 설정 후 다음과 같이 조치합니다.

① {oz스케줄러폴더}/bin/scheduler.sh 을 텍스트 편집기로 열고, jre버전이 1.8이상인지 확인합니다.

```
OZSCHEDULER_HOME=/apps/ozsch80
JAVA_HOME=/usr/java8/
```

② {oz스케줄러폴더}/bin/scheduler.sh, scheduler_stop.sh을 텍스트 편집기로 열고, log4j 관련 라이브러리를 아래와 같이 변경합니다. (기존 log4j.jar 삭제하고 신규 jar 4개 추가)

```
# ------------------------------------
#            Library for Log4j 2.25.4
# ------------------------------------
#OZSchLib=$OZSchLib:$OZSCHEDULER_HOME/lib/log4j.jar      ← 삭제 또는 주석처리
OZSchLib=$OZSchLib:$OZSCHEDULER_HOME/lib/log4j-api-2.25.4.jar
OZSchLib=$OZSchLib:$OZSCHEDULER_HOME/lib/log4j-core-2.25.4.jar
OZSchLib=$OZSchLib:$OZSCHEDULER_HOME/lib/ozlog4j2.11.jar
```

③ 스케줄러를 재시작 합니다. (.sh 파일명은 다를 수 있습니다)

oz_stop.sh // 스케줄러 중지

ps –ef | grep scheduler // 스케줄러가 중지되면 java로 Scheduler를 구동한 프로세스가 없어야 함

oz_start.sh // 스케줄러 구동

④ {oz 스케줄러폴더}/logs/OZSchedulerMainLog.log 에 ready up 문구가 나오면 정상적으로 구동된 것입니다.

4. 디버깅

- **ozlog4j2.11.jar 가 없는 경우 아래 에러 발생**

```
javax.servlet.ServletException: <OZ Servlet> Error in configure. :
java.lang.NoClassDefFoundError: org/apache/log4j/Category
        at oz.server.OZServlet.a(OZServlet.java:273)
        at oz.server.OZServlet.a(OZServlet.java:331)
        at oz.server.OZServlet.init(OZServlet.java:86)
        at org.apache.catalina.core.StandardWrapper.initServlet(StandardWrapper.java:1115)
        at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1064)
        at org.apache.catalina.core.StandardWrapper.allocate(StandardWrapper.java:748)
        at
org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:115)
        at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:90)
        at
org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:494)
```

- **ozlog4j2.11.jar 대신 ozlog4j2.jar 을 넣은 경우**

```
java.lang.NoSuchMethodError:
org.apache.logging.log4j.core.layout.PatternLayout.createLayout(Ljava/lang/String;Lorg/apach
e/logging/log4j/core/config/Configuration;Lorg/apache/logging/log4j/core/pattern/RegexReplac
ement;Ljava/nio/charset/Charset;ZZLjava/lang/String;Ljava/lang/String;)Lorg/apache/logging/log
4j/core/layout/PatternLayout;
        at
com.forcs.log4oz.ver2.OZConsoleAppender.createAppender(OZConsoleAppender.java:48)
        at com.forcs.log4oz.ver2.OZCategory.addAppender(OZCategory.java:167)
        at oz.framework.log.c.a(PropertiesParser.java:578)
        at oz.framework.log.c.a(PropertiesParser.java:250)
        at oz.framework.log.OZLogManager.a(OZLogManager.java:660)
```

5. 오즈 API

OZ API (서버바인딩, 스케줄러를 이용하여 서버에서 export 작업을 하는 경우 등) 사용시 API 를 구현한 jsp 나 서블릿이 올라간 웹 모듈에 log4j 패치가 필요합니다. 예를 들어 oz 서버가 아닌 업무 웹모듈에 ozsfw80.jar 가 있는 경우 업무 웹모듈도 로그 관련 패치를 해주셔야 합니다.

OZ API를 구현한 jsp, servlet 등 java 클래스가 오즈 서버가 아닌 다른 웹 모듈에 올라간 경우, 해당 WEB-INF/lib/ 폴더에도 ozsfw80.jar, log4j.jar가 존재합니다.

이 log4j를 패치하려면 다음과 같이 파일을 추가해 주셔야 합니다.

Log4j 2.25.4 패치

다음 jar를 해당 웹모듈 WEB-INF/lib/에 복사합니다.

- ✓ log4j-api-2.25.4.jar
- ✓ log4j-core-2.25.4.jar
- ✓ ozlog4j2.11.jar
- ✓ ozlogextra.jar → BasicConfigurator, Category 클래스를 참조한 경우에만 필요

UDE, UDL, IDL와 같은 오즈 API를 사용한 경우에는 상속받는 인터페이스명을 뒤에 OZLog가 붙은 인터페이스로 변경해야 합니다. 또한 메소드들의 Category를 모두 OZLog로 변경해야 합니다.

자세한 내용은 🔗가이드 문서 21페이지를 참고하시기 바랍니다.