# ゼロデイ脆弱性Webアプリ型検出器の作成と評価

二ノ方 理仁（芝中学校　３年）

## 序論：Webアプリ型脆弱性検出器の利点と課題

### ゼロデイ脆弱性とは

ソフトウェア中に発見された脆弱性が公表前・修正前の状態であること。対策がないまま攻撃される可能性があり脅威が大きい。

**Webアプリ型**検出器の利点

- URLにアクセスするだけでWebサービスの脆弱性が検出できる。
- ボタンのクリック等で直感的に操作できる。
- ユーザの環境に依存せず利用できる。（Windows、Mac、Linux）

想定した**ユーザ**と**背景**

- ノーコード開発ツールの利用、サーバ運用のコスト減等により、個人でもWebサービスを作れるようになった。
- 個人向け・操作が簡易なゼロデイ脆弱性検出器が必要である。

⇒ Webアプリ型ゼロデイ脆弱性検出器の作成と評価

- 個人ユーザの脆弱性検出器利用を促進する。
- サーバ停止などの脆弱性対応を早めることができる。

### 目的

ブラウザ上で簡易に動く検出器を作成し、ゼロデイ脆弱性検出の直感的な操作を可能にする。検出器の正解率・適合率・再現率を評価する。

## 方法：Log4j detectorの作成と特徴

本研究では、検出するゼロデイ脆弱性として**Log4Shell**を取り上げ、検出器Log4j detectorを作成した。

### Log4Shellとは

Javaのロギングライブラリ Apache Log4jでリモートコード実行が可能になってしまう脆弱性。2021年11月に発見され、12月には脆弱性評価尺度であるCVSSスコアで10.0と高い危険性が指摘された。

### Log4Shellを利用した攻撃とは

Log4jにはログを収集する機能、ログの文字列を置き換える機能が備わっている。脆弱性により攻撃者が送った悪意ある文字列がログに出力され、Log4jが文字列をコードとして認識することで攻撃者のコードが実行されてしまう。

### 仕様

- Log4j detectorの記述言語はJavaである。
- WebフレームワークはSpring Frameworkを使用した。GUIはCSSライブラリbootstrapで作成した。

### 脆弱性検出の流れ

1. ターゲットに**擬似攻撃**を送信する。
2. 脆弱性があった場合、検出器のLDAPサーバにアクセスがあるので検知する。
3. アクセスがあったら、LDAPサーバはリクエストをWebサーバに送信する**リモートコード**を返却する。
4. ターゲットがリモートコードを実行し、Webサーバにリクエストが来たら、ユーザに脆弱性ありの結果を表示する。

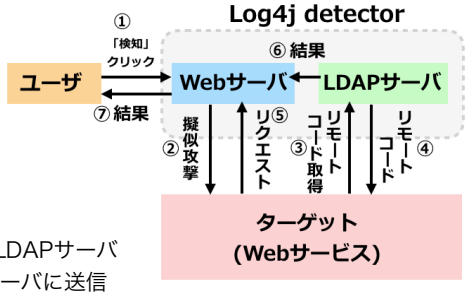


**図1. 脆弱性がある場合の検出の流れ**

### Log4j detectorの特徴

- URLにアクセスすることで利用できる。
- ボタンをクリックする簡易な操作で検出を開始できる。
- 検出結果は脆弱性あり・なしの2種類である。脆弱性ありだった 場合の対策としては、サーバ停止を想定している。


**図2. Log4j detectorのGUI**

## 評価：検出結果の評価

### 対象

テスト用に作成したWebサービスを評価の対象とした。

- 記述言語はJava、WebフレームワークはSpring Framework
- サーバのOSはUbuntu20.04、CPU Dual 20-Core Intel Xeon E5-2698 v4、メモリ512GB
- テストWebサービスは3種類のJava（ver. 6, 7, 8 ）をインストールしたコンテナ上で、**リクエストヘッダをLog4jで**出力する。
- それぞれのJavaのバージョンに対応するLog4jを用いる。

**表1. JavaとLog4jの組み合わせ**

| Javaのバージョン | Log4jのバージョン |
|---|---|
| 6 | 2.2, 2.3.1, 2.3.2 |
| 7 | 2.11.2, 2.12.0, 2.12.1, 2.12.2, 2.12.3, 2.12.4 |
| 8 | 2.13.0, 2.13.1, 2.13.1, 2.13.3, 2.14.0, 2.14.1, 2.15.1, 2.16.0, 2.17.0, 2.17.1 |

**(計19通り、うち12に脆弱性あり)**

### 分類と評価指標

- 予測値と実測値について脆弱性あり・なしの二値分類で評価した。
  - True Positive(**TP**)：実測値と予測値共に正
  - False Positive(**FP**)：実測値が負にも関わらず正と予測（偽陽性）
  - True Negative(**TN**)：実測値と予測値共に負
  - False Negative(**FN**)：実測値が正にも関わらず負と予測（偽陰性）
- 予測結果の評価指標は以下の３種類を用いた。
  - Accuracy 正解率…予測したデータ数あたりの正解データ数
  - Precision 適合率…予測が正であったデータ数あたりの実際正であったデータ数
  - Recall 再現率…実際に正であったデータ数あたりの、実際正であったデータの中で正しく予測できたデータ数

## 結果と考察

### 結果

Log4j detectorを用いてテスト用Webシステムの脆弱性を検出した結果は以下の通りである。

- 実際に脆弱性が含まれているWebサービス（実測値正）は12、含まれていないWebサービス（実測値負）は7であった。Log4j detectorは全てを**正しく検知**することができ、**偽陽性・偽陰性共に０**だった。
- 評価指標はAccuracy 正解率が１（100%）、Precision 適合率が１（100%）、Recall 再現率が１（100%）であった。

**表2. 検出結果**

| | 実測値　正 | 実測値　負 |
|---|---|---|
| 予測値　正 | TP 12 | FP 0 |
| 予測値　負 | FN 0 | TN 7 |

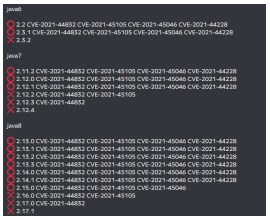

**図3. 検出結果（raw data）**

## 結論

### 結論

ゼロデイ脆弱性は対応策のない状態の攻撃されやすい脆弱性である。個人でWebサービスを運用する時、発見された脆弱性が含まれているかを早期に検出し、場合によってサーバを停止させるのが望ましい。ブラウザ上で動く検出器は簡易な操作で利用でき、マルチプラットフォームで使え、脆弱性を正確に検出できる利点があると考える。

### 課題

本研究では直感的に使えるGUIを採用したので、検出器の完成までに時間がかかる欠点がある。短期間でゼロデイ攻撃に対処するためには、開発時間の短縮は今後の課題である。

## 参考文献

pimps/JNDI-Exploit-Kit
https://github.com/pimps/JNDI-Exploit-Kit
分類問題の予測結果の評価指標（Accuracy, Precision, Recall, F値, AUC）について整理してみた
https://tech.ledge.co.jp/entry/metrics