情報ネットワーク応用 演習課題 2A

学籍番号:1280391
細川 夏風

2026 年 2 月 7 日

1　背景

昨今，IoT デバイスを用いたセキュリティ事案が多く発生していることは言うまでもない．2018 年に多く確認されたウイルス “Mirai” は記憶に新しいだろう [1]．IoT デバイスはその性質上，膨大な上のやり取りがされている．必然的に乗っ取りや盗聴されると盗まれる情報の量や希少性も大きくなってくる．そのため，我々クライアントは IoT デバイスにも多くの意識を割く必要がある．本実習ではその一例として IoT デバイスに対して辞書攻撃のパスワードクラックを仕掛け，デバイスのカメラ機能を乗っ取るという攻撃からパスワードクラックからの対策と IoT デバイスの脆弱性についての議論を行う．

2　内容

2.1　環境

本実習の環境は以下の図の様になっている 図1．すでに接続している Windows 環境のサーバからサーバ内にインストールされている TeraTerm を用いて Linux 環境の攻撃者端末に接続する．その後，複数のツールを用いて IoT デバイスに接続する．本レポートは IoT デバイスのパスワードを変更するまでのものとする．

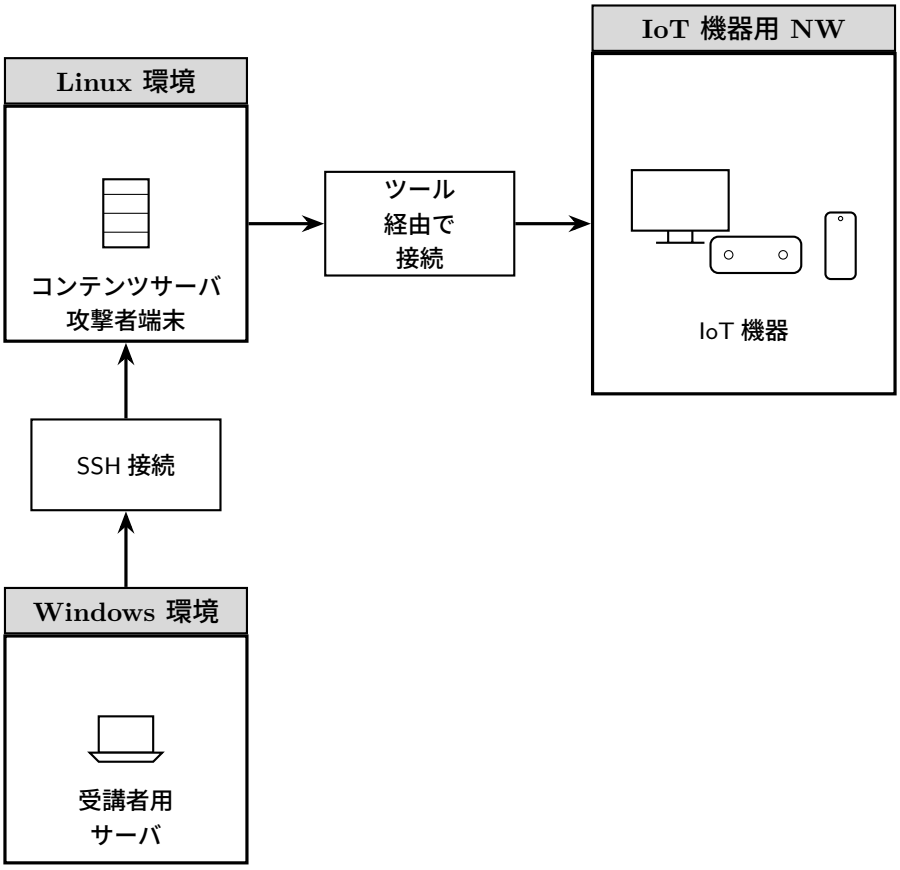
IoT 機器用 NW
Linux 環境
ツール
経由で
接続
コンテンツサーバ
攻撃者端末
IoT 機器
SSH 接続
Windows 環境
受講者用
サーバ

図 1　演習環境の構成図

2.2 使用ツール

本実習で使用するツールを以下に示す.

Nmap

Nmap(“Network Mapper”) はネットワーク調査およびセキュリティ監査を行うためのオープンソースツールである [2].

Nmap の引数として，IP アドレス (2 進数表記も可能)，ホスト名が使用可能である．しかし，この際 IP アドレスと同じホスト名を引数内に 2 つ指定しても同じ結果が得られる．また，IP アドレスは末尾にサブネットマスクを指定することによって，そのセグメント内に対してブロードキャストにポートスキャンが可能である.

- オプションの指定
  - -sn
    - Ping スキャン
    - ホストの発見のみ実行し，結果を一覧表示する
    - ポートスキャンは行わない
  - -sL
    - List スキャン
    - 検査対象をリストし DNS 解決するのみで検査対象へパケットを送らない
    - 検査対象が正しいかを確認するために用いられる
  - -n
    - 発見した IP アドレスに対して DNS の逆引きを行わない
    - 処理高速化のために用いられる
  - -sS
    - TCP SYN スキャン
    - 対象ポートに SYN パケットを送り，応答を確認する
    - 実行には管理者権限が必要
  - -sT
    - TCP Connect スキャン
    - 対象ポートに接続を行い応答を確認する
    - SYN スキャンにクラ b て，対象のホストのログに記録される可能性が高い
  - -A
    - OS 検知，バージョン検知，スクリプトスキャンなど，様々な処理を有効にする
    - サービスの詳細を調査する場合に利用される
  - -p
    - 対象のポートを選択できる

  - 複数選択が可能
- -iL
  - 引数にファイル名をとり，そのファイル名からターゲット指定を読み込む
  - コマンドラインで非常に巨大なホストリストを渡すのは不適切である場合が多いため，その対策として用いられる
- -iR
  - インターネット上のサーバに対して無作為にポートスキャンが可能
  - 引数にどれほどのサーバをポートスキャンするかを指定できる
- --execlude
  - ホスト名を指定したあと，[] 内にコンマ区切りで入力することによってスキャン対象から除外できる
- --excludefile
  - 上記の--execlude と同様の機能を提供するが，こちらはファイルに記述されたものを読み取り除外する

hydra

hydra とは，並列化さらたログインクラッカーである．また，攻撃対象として多数のプロトコルをサポートしている [3].

- オプションの指定
  - -l LOGIN / -L FILE
    - ログイン名 (ユーザ名) を指定する
    - LOGIN
      - 単一のログイン名を指定する場合に用いる
    - FILE
      - ログイン名が列挙されたファイル (リスト) を読み込む場合に用いる
  - -p PASS / -P FILE
    - パスワードを指定する
    - PASS
      - 単一のパスワード候補を指定する場合に用いる
    - FILE
      - パスワード候補が列挙されたファイル (辞書ファイル) を読み込む場合に用いる
  - -x MIN:MAX:CHARSET
    - 辞書ファイルを使わず, パスワードを自動生成 (ブルートフォース) して試行する
    - MIN:MAX:CHARSET
      - 最小文字数: 最大文字数: 文字種 の形式で指定する (例: 3:5:a1 なら英小文字と数字で 3〜5 文字)

・-e nsr
  * 辞書リスト等に加え, 特殊なパターンのパスワードを追加で試行する
  * n
    ・空のパスワード (パスワードなし) を試行する
  * s
    ・ログイン名と同じ文字列をパスワードとして試行する
  * r
    ・ログイン名を逆さにした文字列をパスワードとして試行する
・-o FILE
  * 解析結果をファイルに出力する
  * FILE
    ・発見されたログイン名とパスワードのペアを書き出すファイル名を指定する
・-t TASKS
  * 並列処理の数を指定する
  * TASKS
    ・ターゲット 1 台あたりの同時接続数 (スレッド数) を指定する (デフォルトは 16)
・-M FILE
  * 複数のターゲットに対して攻撃を行う
  * FILE
    ・攻撃対象のサーバ (IP アドレスやドメイン) が列挙されたファイルを指定する
・-f
  * 1 つでも有効なログインペアが見つかった時点で, そのホストへの攻撃を直ちに終了する
・-V
  * 詳細表示モード (Very Verbose)
  * 試行中のログイン名とパスワードをリアルタイムで画面に表示する

2.3 方法

(1). Tera Term のホスト名に対象の IP アドレスを入力し，他項目はデフォルトのまま OK を押す.
(2). ユーザ ID 欄，パスワード欄に既知の ID とパスワードを入力する.
(3). 次に，以下のコマンドから Nmap を用いたポートスキャンを行う.

```
1        nmap -Pn -n -A -p <IoTデバイスのIPアドレス>
```

(4). 結果から，ポート番号 45678 に http サービスが稼働している事がわかった.
(5). Tera Term の機能から SSH 転送を利用し，Windows 端末から IoT デバイスへの HTTP 通信を実現する.

(6). Tera Term の設定から SSH 転送を押し，追加を選択する．この際，ローカルポートに 1 から 1024 でない任意のポート番号を入力する．ただし，65535 までである．リッスンの項目は 127.0.0.1 を入力し，リモート側ホストは IoT デバイスの IP アドレスそ指定する．最後にポートの項目は先程 HTTP サービスの稼働を確認した 45678 である．

(7). Web ブラウザから http://127.0.0.1:45678/にアクセスする．アクセスすると認証画面が表示される．

(8). hydra によって辞書攻撃を行う．このとき，ユーザ名とパスワードリストは /opt/iot/mirai/user.txt，/opt/iot/mirai/pass.txt 内に格納されている．

```
1        hydra -t 5 -L /opt/iot/mirai/user.txt /opt/iot/mirai/pass.
            txt http-get://<IoTデバイスのIP >:45678/
```

(9). これにより，ユーザ名 admin に対して，パスワードが 88888 であることがわかった．

(10). これを使用し，FireFox の Sign in から管理画面へ侵入する．

(11). 管理画面に侵入したので，そこからこのユーザに対して任意の行動を取ることができる．パスワード変更もその 1 つである．

3 考察

今実習から以下のことが得られた．

(i). ユーザ名を推測しやすいものにすると，パスワードクラックにかかる時間が低下する場合がある．

(ii). パスワードを推測しやすいもの，何かの単語を用いると辞書攻撃の際にパスワードクラックにかかる時間が大幅に低下する事がある．

辞書攻撃は非常に強力な攻撃の 1 つであると言える．なぜなら，パスワードやユーザ名には何かしらの単語を用いることが非常に多い．それはログインに用いる際の人為的負荷を軽減するためである．ログインの際に正規のユーザ自体も覚えられないような文字列である場合，パスワード強度としては非常に強力であるが，認証自体では適切とは言えない．その解決策として，パスワードマネージャーが多く挙げられる．Web ブラウザ等やその他アプリケーションのログインに用いられる．これにはいくつか効果があり，単純に長く難解なパスワードを記憶する必要がなくなることがある．他にも，Web サイトの場合，偽サイトかどうかの判断に用いることも可能だ．しかし，これにもデメリットは存在する．そのパスワードリスト自体が漏洩した場合，多くのサイトへの侵入を許してしまう．次に特定の環境に依存する場合があるということである．これはそのパスワード自体をどの媒体，どのプラットフォームに保存しているかによって環境が変化した場合，利用が単純ではないということである．しかし，それらを共有すればするほど漏洩のリスクが上昇する．そのため，他の解決策として 2 要素認証で身体情報等を用いることができればそれが最も良いが，その

ようなサービスは多くない．また，最も良いとされているのは定期的にパスワードを変更することである．これらの機能や対策を総合的に用いてパスワードを作成することが現時点における最良の手段であると私は考える．

また，今実習については攻撃対象が IoT デバイスであることから，高負荷の攻撃を行うことが難しい．IoT デバイスには多くの制約がつきまとう．計算能力もその 1 つである．そのため，多くの場合に用いられるのがソーシャルエンジニアリングである．計算機の特定の機能を悪用すること無く対象に攻撃を行うことである．これによって計算機に依存せず攻撃を行うことができる．

最後に IoT デバイスに侵入されることのデメリットについて議論を行う．私が考える IoT デバイスがクラックされることの最大のデメリットはその IoT デバイスが接続されているネットワークについての情報が攻撃者に把握される可能性があることであると考える．些細な IoT デバイスからその上位ネットワークについて，それに接続される可能性さえ孕んでいる．もちろんこれは稀なケースであることには変わりない．しかし，いくつか事例は存在する．ロジテック製のルータに存在した脆弱性を用いて，外部から操作され攻撃の拠点として使われた事例もその 1 つである．これに対する対策は VLAN での隔離という方法が取られることが多い．他にも根本的に孤立したデバイス (IoT デバイスや Web カメラなど) のセキュリティを強化することや内部ネットワークに直接アクセスできないようにすることなどが挙げられる．最終的に必要なのはどのデバイスにも簡単にアクセスさせないこと，もしアクセスがある場合はその部分のみ VLAN 等で切り離すことである．そして，ソーシャルエンジニアリングに対する理解を深め自身が攻撃を受けたことを理解し，それに対する適切な処置を取れるようになる必要がある．

こうした，IoT デバイスの乗っ取りや盗聴はこれから IoT デバイスが増えていく中で増加していくだろう．そのため，それらを利用する立場に置かれる私たちがこの問題を理解し，対処することができるようになることがこの実習での意義の最たるものである考える．更にはこの問題が世に周知され，多くの被害が緩和されることを期待し本報告の結びとする．

参考文献

[1] 情報セキュリティ白書 2019，独立行政法人情報処理推進機構 (IPA)，2019 年 8 月 8 日

[2] Nmap 公式ページ，https://nmap.org/

[3] Kali Linux Tools，https://www.kali.org/tools/hydra/