卒業論文

ハニーポットによる不正ファイルの入手と分析

2024 年度

拓殖大学工学部情報工学科

吉村　直将

指導教員 教授　蓑原 隆

助手　田島 信行

第 1 章

はじめに

近年，サイバー攻撃の発生件数が年々増加してきており，その攻撃手法も多様化している．多様化した新しい攻撃に対処するためには攻撃手法の分析が必要である．攻撃手法の分析のために，攻撃者を誘き寄せ，不正アクセスを受けるハニーポットを用いて攻撃者の情報を収集する方法がある．例えばハニーポットを利用して，ログイン試行時に使われる ID やパスワード，ログイン後に攻撃者から送られるシェルコマンド等の情報を収集する方法が提案されている [1].

本研究では，より具体的な攻撃者の攻撃手法の情報を得るため，攻撃者がログイン成功後に行う攻撃に着目し，ハニーポットを用いて，攻撃者から送信されるコマンドやそのコマンドから入手できるファイルの情報を収集し，解析するシステムを構築する．そして，攻撃の分析を行い，最新の攻撃内容について警告を発することを目的とする．

第 2 章

攻撃収集分析システム

攻撃者がダウンロードさせようとしてくる不正なソフトウェアの解析を実現する為のシステムの構成を図 5.1 に示す.

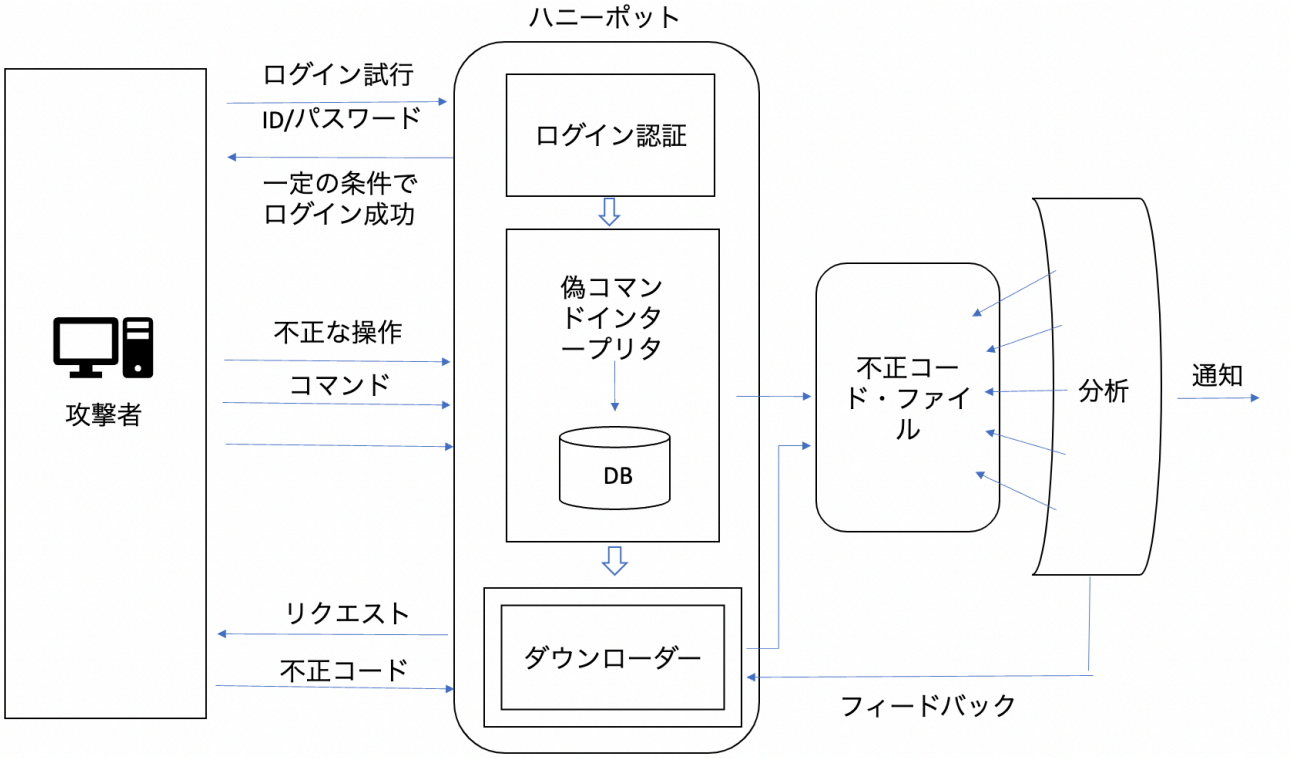

図 2.1　システムの構成

ハニーポットは，攻撃者からの何度かのログイン試行を受け，一定の条件で，攻撃者にログイン成功したと思わせる．その後，攻撃者にコマンドインタープリタの様な返答を見せ，不正な操作のコマンドをデータベース DB に収集する．収集したコマンドから，攻撃者が不正なサイトからダウンロードさせようとする不正なファイルを安全に入手する．また，コマンドの中には，ハニーポット内に不正ファイルの作成を試みるものもあり，安全にファイルを作成し，収集を行う．収集した不正ファイルのコードから，どの様な不正ファイルかを分析し警告を発する．また，その情報からダウンローダーに生かせるものをフィードバックする．

本研究では，ハニーポットとして DShield (Distributed Intrusion Detection System)[2] と呼ばれるグローバルなセキュリティコミュニティによって構築された分散型侵入検知システムを使用する．これまで我々の研究室では主に攻撃頻度の時系列解析のために Dshield ハニーポットを利用している [3]．が，本研究では図 5.1 に示すように，Dshield の cowrie[4] のログイン認証部，コマンドインターポリタ部に必要な機能を追加する，また，コマンドからファイル又は，URL などを取集するダウンローダー部，不正ファイルの分析部は，新しくプログラムを作成する．

第 3 章

Dshiled の環境構築と運用

Raspberry Pi に Dshield をインストールし，パスワードや接続する無線 LAN の設定を行なった．Rassberry Pi のファイアウォールの設定から SSH を有効にする事で，外部からの接続を cowrie が対応するように設定した．また，研究室内のネットワークからの接続は攻撃と見さないように設定した．

攻撃者からコマンドを収集するために Dshield のプログラムを調査し，コマンドを収集できているのか確認した．まず初めに，Cowrie は，攻撃者からのコマンドをどう対処しているのかを調べ，t 攻撃者に返すコマンドのディレクトリとして extcoms ファイルがあることが分かった．次に，このファイルを使っているものを次のコマンドで検索し，`find. -exec grep, textcmds {}\; -and -print 2 > /dev/null` srv/cowrie/shell/protocol.py で使われていることが分かった．調査の結果，Dshield は攻撃コマンドを取集し，/srv/cowrie/var/log/cowrie の場所に保存していることが分かった．また，ログイン試行に対応しているプログラムが/src/cowrie/core/の場所にある auth.py であることを突き止め，解読したところ，Dshield は外部からの攻撃者からのログイン試行を 1 回以上のランダム数行うと，ログイン可能とするように設定されていることが分かった．

実際にハニーポットを運用し 5/19 から 5/30 の期間中にコマンドを取集した．収集したコマンドの中には特定のパターンのコマンドが多く発見された．例えば組み込み Linux で複数のコマンドをまとめるために使われる busyBox が含まれていて，この期間中に多くの攻撃が組み込み Linux の機器を対象としていることが分かった．また，収集したコマンド中には，不正なファイルをダウンロードさせるため wget コマンドを用いているものや，不正なファイルをハニーポット内に作成するために，echo コマンドを用いているものが多かった．

第 4 章

ダウンローダー部の作成

ダウンローダー部のシステムプログラムは，主に 3 つの動作を行う．

(1) 不正ファイルデータのダウンロードコマンドを模倣して不正データを入手する
(2) 不正データを安全に扱うためにヘッダ情報を追加する
(3) データを保存するファイル作成し，ヘッダ情報と不正ファイルデータを書き出す．

(1) について，Dshield での攻撃コマンドを模倣している部分を調査し，srv/cowrie/src/cowrie/commands の下に Python プログラムとして実現されていることが分かった．例えば，不正ファイルデータのダウンロードに使われている wget の動作は，wget.py 中で行われる．主なそのファイル内の動作プログラムとして，123 行目の `self.deferred = self.download(self.url, self.outfile)` で不正サイト情報である defeerred を，引数として URL と out ファイルで download 関数によって取得する.124 から 128 行内の `if self.deferred:` の if 文で不正サイト情報が得られたら `self.deferredaddCallback(self.success)`success メソッドへ移動し，上手く得られなかったら `self.deferred.addErrback(self.error, self.url)` で error メソッドへ行く．そして，success メソッド内で (2) と (3) の処理を行うコードを追加した．

(2) において，ヘッダ情報を不正データに追加するのは，ダウンロードしたデータをそのままファイルとして保存することを避け，不正データのファイルを誤って動作させてしまった場合でも問題が起きないようにする安全対策のためである．具体的なヘッダ情報としては，ダウンロードに使用した URL の前に 4 桁の URL 文字数を追加したものとし，URL の文字数は，4 桁の数字を追加したものとする．プログラムとして URL の文字数は,format メソッドを利用し，0 埋めを行う．`hd = '{:04}'.format(hdct)`.

(3) については，Dshield ハニーポットが攻撃者に見せているファイルシステムの外の領域として外付け HDD に作成する，そしてファイル名とする fname は，ファイル場所と攻撃があった日時を入れ，

`fname = "/HD/malwares/tmp/MW" + str(now.date()) + "_" + str(now.time())`，open 関数でファイルを作成する．`self.file = open(fname, "wb")`.

実際に研究室で運用しているハニーポットに組み込んで確認したところ，攻撃があったときにデータが記録されていないことが判明した．原因は DShield が 1 日に 1 回,18 時 28 分に，配布元を確認し，システムのアップデートがあった時，自動的に更新しているためであった．そこでハニーポットの/etc/cron.d/dshield にアップデートの情報が持ち，そのファイルを wget.py へ周期的に変更を加えたもので上書きをするように設定し，正しく動作していることを確認した．

第 5 章

不正ファイルの分析

不正ファイルの分析として実際の不正ファイルからの情報取得を行った．具体的には, 直接不正ファイルを作成させようとしてくる攻撃の 1 つとして, 複数の echo コマンド を次のように送るものについて, `echo -ne "\x7f\x45.." > niggabox` 実際にファイルを作成して調査した．以下，作成したファイルを対象データと呼ぶ．まず，対象データは 2,720 バイトのバイナリーデータであることがわかった．バイナリーデータのファイルは，最初の数バイトがファイル形式を示している場合が多い．対象データについて先頭部分を調べたところ，最初の 4 バイトが，“x7f x45 x4c x46” であり，ELF 形式 [5] のファイルであると判別した．

次に ELF 形式のヘッダ情報の解析を行う．ELF 形式のファイル構造では，具体的に，1 から 4 バイトがファイル形式を示し，5 バイト目が 32or64 ビットのファイルか．6 バイト目がリトルエンディアンかビックエンディアン方式か，18 バイト目が対象としているアーキテクチャの種類，と判別することができる．そのため，今回の対象データは，32 ビットのリトルエンディアン形式で，ARM アーキテクチャ用であることがわかった．そこで，次のコマンドで逆アセンブリを行い，`objdump --print-imm-hex` –disassemble -s niggabox! 内容を確認したところ，マルウェアの Mirai のソースコード [6] として公開されているコードの一部に酷似していた．さらに，リンク先から別のファイルをダウンロードする機能を持っていることが分かり．具体的にこの不正ファイルでは，リンク先はアムステルダムで，別のファイルは jklarm というファイルであることも分かった．

5.1　ELF 形式のファイルから情報取得の自動化

ELF 形式のファイルについて，自動で情報を解析するためのプログラムを作成した．ELF 形式は図 2 に示すように, 実行時にメモリ上に展開される.text，.rodata などの複数セクションで構成されている．セクションは，ELF ヘッダ，プログラムヘッダテーブル，セクションヘッダテールの 3 種類あり，ELF ヘッダでは，プログラムヘッダテーブルとセクションヘッダテーブルのサイズやオフセットの情報が取得できる．また，プログラムヘッダテーブルでの情報で，読み出し可能なデータのセグメントに分けられている．各セクションの境界はセクションヘッダテーブルの情報で切り分けられる．

自動情報取得として，プログラムに埋め込まれた文字列などのリテラルが格納されている.rodata から情報を取得する処理を実装した．ELF 形式では各セクションが記録される順番は規定されていないが，各セクションの名前のテーブルが格納されたセクション (.shstrtab) からの番号は ELF ヘッダから取得できる．セクションヘッダテーブルは，エントリという項目で分かれており，各エントリは一つ一つのセクションの情報について書かれている．各エントリ内の一部が.shstrtab セクションのテーブルの番号の情報をもつため，その番号で各エントリがどのセクションかを判断ができる，そこで，.rodata を検索し，対応するセクションの内容をダンプするようにプログラムした．手作業での分析に使用した対象データ (niggabox) について，作成したプログラムを実行し，図 3 のようにダウンロードしようとするファイル名 (jklarm7m) などの情報が取得できることを確認した．また，このプログラムを応用して，

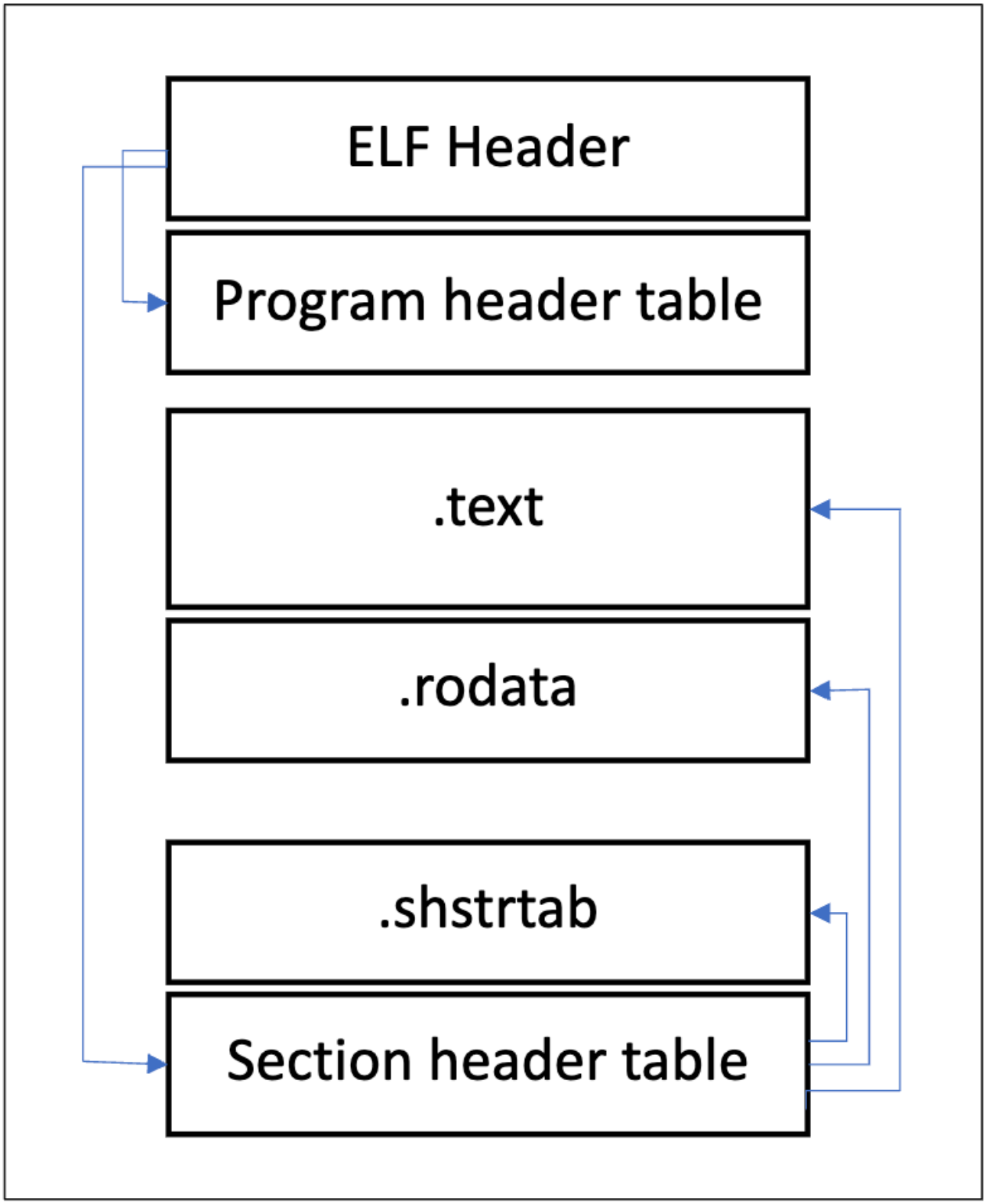

図 5.1　ELF 形式のファイル構造

第 6 章

まとめ

本研究では，Dshield ハニーポットに機能を追加することで，攻撃コマンドが送り込もうとしている不正データを所得し，ELF 形式の場合に解析を行うシステムの開発を行なった．我々の研究室で運用しているハニーポットに実装して，不正データの取得および解析結果の表示を行えることを確認した．しかし，解析結果を管理者に通知する部分は未実装で残された課題となっている．

参考文献

[1] 中山楓, 鉄穎, 楊笛, 田宮和樹, 吉岡克成, 松本勉：IoT 機器への Telnet を用いたサイバー攻撃の分析, 情報処理学会論文誌, Vol. 58, No. 9, pp. 1399–1409 (2017).

[2] DShield Honeypot https://isc.sans.edu/tools/honeypot/ (accessed accessed 2024/6/14).

[3] 西田圭介：インターネット上のサイバー攻撃のハニーポットを用いた分析と可視化, 拓殖大学工学部情報工学科卒業論文 (2022).

[4] Welcome to Cowrie's documentation! https://cowrie.readthedocs.io/en/latest/index.html (accessed accessed 2024/5/31).

[5] elf - 実行可能リンクフォーマット (ELF) ファイルのフォーマット https://manpages.ubuntu.com/manpages/trusty/ja/man5/elf.5.html （参照参照 2024/6/14).

[6] Mirai BotNet https://github.com/jgamblin/Mirai-Source-Code (accessed accessed 2024/5/31).