卒業論文

2018年度 (平成30年)

低対話型Honeypotのコマンド拡張による高対話型Honeypotへの近似

慶應義塾大学

総合政策学部 総合政策学科

徳田· 村井· 楠本· 中村· 高汐·
バンミーター· 植原· 三次· 中澤· 武田合同研究プロジェクト

菅藤 佑太

卒業論文　2018年度 (平成30年)

低対話型Honeypotのコマンド拡張による
高対話型Honeypotへの近似

論文要旨

PCの普及やIoTデバイスのシステム高度化により,高度な処理系を組むことが可能になり,デバイス上にLinux系などのOSが動く機器が広く人々に使われるようになった.そうした中でSSHの攻撃によって自分の機器が踏み台にされ,自身の機器から第三者のネットワーク機器に攻撃されていたり,ウイルスやバックドアが設置されて自身の機器が不正にウイルスが実行されるような環境を作られる問題が発生している. 侵入された際に侵入者がどのような挙動をしているのかを知る手段として,Honeypotというものがある.これはSHELLの擬似的な挙動をアプリケーション上で実現し,敢えてSSHで侵入しやすいような環境を作ることで,侵入者にログイン試行に成功したと検知させ,その際に実行したコマンドを観測する.本研究では,Honeypotの機能を拡張することでより多くの,高精度な攻撃ログを取得し,これを評価した.

キーワード:

SSH, Honeypot, 機械学習, 自然言語処理

総合政策学部総合政策学科, 慶應義塾大学

菅藤 佑太

卒業論文　2018年度 年度（平成 年度）

Title

カテゴリー：keyword1, keyword2

論文要旨

This is the abstract.

キーワード：

keyword1, keyword2

慶應義塾大学総合政策学部総合政策学科

I am the author

目次

第1章　序論

1　序論

1.1　研究の背景

PCの普及やIoTデバイスのシステム高度化により,高度な処理系を組むことが可能になり,デバイス上にLinux系などのOSが動く機器が広く人々に使われるようになった.そうした中でSSHの攻撃によって自分の機器が踏み台にされ,自身の機器から第三者のネットワーク機器に攻撃されていたり,ウイルスやバックドアが設置されて自身の機器が不正にウイルスが実行されるような環境を作られる問題が発生している.

侵入された際に侵入者がどのような挙動をしているのかを知る手段として,Honeypotというものがある.これはShellの擬似的な挙動をアプリケーション上で実現し,敢えてSSHで侵入しやすいような環境を作ることで,侵入者にログイン試行に成功したと検知させ,その際に実行したコマンドを観測する.

SSHのHoneypotは大きく二種類に分けることができ,一つは低対話型Honeypot,もう一つは高対話型Honeypotである.低対話型Honeypotは実際のShellの挙動をエミュレートしたアプリケーションシステムである.高対話型Honeypotは実際のOSを用いたシステムであり,低対話型Honeypotと比較して,本物のOSを用いている分高精度な攻撃ログを取得することができるが,乗っ取られ他のホストに攻撃をしたりウイルスに犯されてしまうなどの危険を孕んでいるため設置コストが高く,普及率も非常に低い.一方で低対話型Honeypotはアプリケーションシステムであるため,root権限を取られるような危険が極めて少なくアプリケーションシステム内での脆弱性だけに限った問題しか存在しないため,設置コストが低く,ドキュメントが多く存在し比較的誰でも安全に設置できるため,普及率が高い.しかし,実際ののShellとは異なる挙動や,Honeypotに特有な挙動をしてしまうため,設置したHoneypotに侵入した悪意のあるユーザーに侵入先がHoneypotであると検知されてしまう可能性があるという問題がある.

この低対話型Honeypotと高対話型Honeypotの各々のデメリットを解消し,リスクを小さく,Honeypotに侵入してきた悪意のあるユーザーに侵入先がHoneypotであると検知されないように攻撃ログを取得するためには,設置コストを抑えたアプリケーションシステムで動作する低対話型Honeypotの挙動をできるだけ実際のShellに近似すれば良いのではないかと考えられる.

本研究では低対話型Honeypotを実際のShellの挙動に近似するために,実際のShellに実装されているもので低対話型Honeypotに実装されていないコマンドの実装や,低対話型Honeypotに特有の異常な

挙動を修正を行うことでこの問題を解決できるのではないかと考えた.

1.2 論文の構成

本論文の構成を以下に示す.第1章では,本研究の背景と目的について述べた.第2章では,本研究の要素技術となるShellとHoneypotと時系列データの扱いについて整理する.第3章では,本研究の問題の定義をする.第4章では,本提案手法のシステムについて解説する.第5章では,実装方法や実装例について述べ,第6章で提案システムの評価手法について説明する.第7章では先行研究や事例について紹介する.第8章では本研究のまとめと展望について言及する.

第2章　本研究の関連技術

2　本研究の関連技術

本章では,本研究の要素技術となるShellとHoneypotと時系列データの扱いについて各々整理する.

2.1　Honeypot

使われているデバイスで不正なSSHによって侵入された際に侵入者がどのような挙動をしているのかを知る手段としてのHoneypotについて,その技術について解説する.

2.1.1　低対話型Honeypot

SSHのHoneypot [4]は低対話型Honeypotと高対話型Honeypotの大きく二種類に分けることができ本節ではSSHの低対話型について解説する.SSHの低対話型Honeypotは実際のShellの挙動をエミュレートしたアプリケーションシステムである.エミュレートされたものであるのでコマンドやその挙動についての機能が限定されており,実際のShellの機能として不足があり,侵入者に侵入先がHoneypotであると検知され,exitされてしまう可能性を含んでいるため,収集ログの精度に問題がある.以下に侵入者がSSHで不正に機器に侵入してから踏み台にして他の機器に攻撃を仕掛けるまでの一般的なフローを図2に示す.

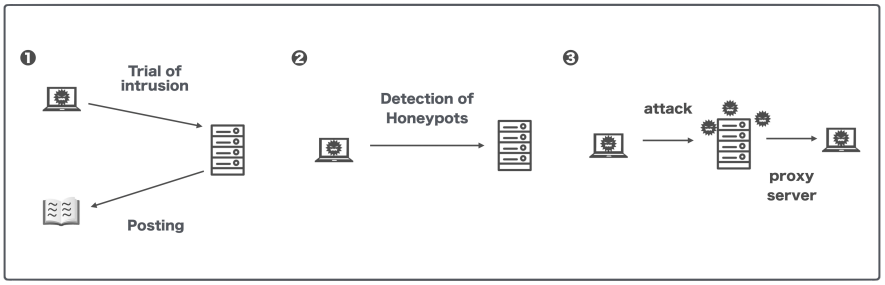

図1: 不正なSSH侵入者の想定行動フロー

しかし,実際のShellの挙動をエミュレートしただけのアプリケーションシステムなので,脆弱性がアプリケーションシステム内に限られ,root権限を侵入者に許してしまい,踏み台にされてしまうなどの危険が極めて少ないため,比較的安全に運用することができる.続いてSSHの低対話型Honeypotの種類についての紹介をする.

2.1.1.1 Kippo

Kippoは,悪意のあるSSHのログイン試行者や侵入者の挙動やログを記録するために使用されるPythonで実装されたSSHの低対話型Honeypotであり [1],Kippoは前身のKojoney [2]に大きく影響を受けている.実装はPythonで行われており,ネットワークはTwisted [3]というフレームワークで組まれている.Kippoのプロジェクトは低対話型Honeypotとして2009年に登場し,Raspberry Pi [5]などを筆頭としたシングルボードコンピュータ [6]の普及により2010年頃から勃興したIoTデバイスの高度化 [7]と相まって広く設置された.Kippoは2014年頃を最後に現在はプロジェクトが進んでいない. [8] KippoはIoTデバイスの高度化広く設置されたSSHの低対話型Honeypotのうちの一つであったが,コマンドの機能が少なく,またKippo特有の異常な挙動があったりと多くの問題があった.以下の図2にその一例を示す.

```
nadechin@cpu:~$ echo -n test
testnadechin@cpu:~$
```

```
s15445ys@s15445ys-neco:~$ echo -n hello
-n hello
```

図2: 正しいShellの挙動とKippo特有の異常な挙動の例

上図が通常の挙動で,下図がKippoの挙動である.echoコマンドの-nオプションは改行をしないようにするというものであるが,実際のShellの挙動が改行がされることなく正しく出力されているのに対して,Honeypotの挙動ではオプション部分も出力されてしまっているという問題がある.これは有名なKippo特有の挙動であるため,これによってHoneypotであると検知されてしまう可能性がある.

第3章　問題定義

3　問題定義

本章では,現状のHoneypotの問題点を整理し,この問題をどのように解決すれば良いのかを定義する.

3.1　SSH Honeypotの現在の問題

Honeypotには運用する上で大きな問題が2つある.一つは設置したHoneypotに悪意のある侵入者が侵入先をHoneypotであると検知してしまう問題である.もう一つはHoneypotに侵入を許した侵入者にHoneypotを設置した機器から攻撃が仕掛けられてしまう危険があることである.以降の節でこの問題について説明する.

3.1.1　SSHの低対話型Honeypotにおける問題

前章の要素技術のところでも説明した通り,SSHの低対話型Honeypotは実際のShellの挙動をエミュレートしたものであるのでコマンドやその挙動についての機能が限定されており,実際のShellの機能として不足があり,侵入者に侵入先がHoneypotであると検知され,exitされてしまう可能性を含んでいるため,収集ログの精度に問題がある.

3.1.2　SSHの高対話型Honeypotにおける問題

一方で高対話型Honeypotは,実際に脆弱性を残した実際のOSやアプリケーションシステムを利用したものであるので,侵入を許すと設置したOSの予期せぬ脆弱性を突かれたり新たなウイルスによってroot権限を取られ,設置したOSから他のホストへと攻撃してしまう可能性を含んでいるため,設置コストやリスクに問題がある.

3.2　仮説

以上のSSHのHoneypotにおける問題点を踏まえ,SSHの低対話型Honeypotを設置し,かつその挙動をできる限り本物のShellの挙動に近似することで収集ログの精度や数を落とすことなくセキュアな環境でログを収集できるのではないか.

第4章　本研究の手法

4　本研究の手法

本章では,現状のHoneypotの問題を解決するための手法と,それが本当に解決したのかを評価する手法を説明する.

4.1　問題解決の為のアプローチ

本研究では,以下の三種類のHoneypotを設置する.

1. 広く利用されているSSHの低対話型Honeypot
2. 実際のShellには実装されているが,1.のHoneypotで未実装のコマンドを実装したHoneypot
3. 広く利用されている高対話型Honeypot

これ以降,1.の広く利用されているSSHの低対話型Honeypotのことを”純正の低対話型Honeypot”, 2.の実際のShellには実装されているが,1.のHoneypotで未実装のコマンドを実装したHoneypotのことを”修正済みの低対話型Honeypot”, 3.の広く利用されている高対話型Honeypotのことを”高対話型Honeypot”と呼ぶこととする.

三種類のHoneypotは同じ期間で同じ環境で設置し,それぞれ攻撃ログを収集した.

4.2　問題解決したのかを評価する為のアプローチ

収集したログの内,純正の低対話型Honeypotの攻撃ログを始点に,高対話型Honeypotの攻撃ログを終点に定め,修正済みの低対話型Honeypotの攻撃ログがその間のどこに位置するのかで,高対話型Honeypotにどれほど近づいたのかを評価する. この時,攻撃ログはコマンドの時系列データであるので,実行コマンドの実行の順番を機械学習で分析することによってこれを評価する.

第5章　実装

5　実装

5.1　実装

第6章　評価

6　評価

6.1　評価手法

6.1.1　コマンドの時系列性

6.1.2　コマンドの時系列データのベクトル表現

6.1.2.1　文章のベクトル表現

6.1.3　SSHの低対話型Honeypotの攻撃ログの比較

6.1.4　攻撃ログのベクトル表現によるログの精度

第7章　先行研究

7　先行研究

7.1　SSHのHoneypot

7.1.1　低対話型Honeypot

7.1.1.1 kiipo

7.1.1.2 cowrie

7.1.2　高対話型Honeypot

7.1.2.1 honeynet

7.2　時系列データの処理

7.2.1　確率分布モデル

7.2.1.1 マルコフモデル

7.2.1.2 隠れマルコフモデル

7.2.2　ニューラルネット

7.2.2.1 畳み込みニューラルネットワーク

7.2.2.2 リカレントニューラルネットワーク

第8章　結論

8　結論

8.1　本研究のまとめ

8.1.1　展望

SSHの低対話型Honeypotに実装するコマンドの選定について,OSごとに異なるはずであるが,今回はBusyBoxをそっくりそのまま移植しただけであったので,厳密に実際のShellやOSの挙動を模して本研究を再検証したい.

謝辞

生まれてきてよかったー！

付録

参考文献

[1] Kippo・https://github.com/desaster/kippo

[2] Kojoney・http://kojoney.sourceforge.net/

[3] Twisted・https://twistedmatrix.com/trac/

[4] Honeypot・https://ja.wikipedia.org/wiki/ハニーポット

[5] Raspberry Pi・

[6] Single Board Computer・

[7] Iotデバイスの普及・

[8] Kippoのプロジェクトの現在・

[9]