

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  
菅藤 佑太

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

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った. これによりデバイス上に Linux 系などの OS が搭載された機器が広く人々に使われるようになった. また, Linux 系 OS にリモートログインする手法として SSH がある. これを用いて不正に侵入する攻撃が行われている. 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 がある. 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のログを収集するものである. また現在では Shell の挙動をエミュレートした Honeypot が広く使用されており, この Honeypot は実行できるコマンドが少ない実装になっている. そのため Honeypot への侵入者に侵入先が Honeypot であると検知されてしまう. そこで事前実験では Honeypot のコマンドを拡張し, 拡張をしていない Honeypot とコマンドの拡張をした Honeypot で侵入ログを収集した. 収集したログを確率的な算出方法を使用することで比較した結果, より多くの侵入者のコマンド実行ログのパターンを取得できることを示した. 本研究ではコマンドを拡張した Honeypot の侵入ログがどれほど実際の OS に不正な SSH の侵入をされた際の侵入ログに近似したのかを示すために, 拡張をしていない Honeypot とコマンドの拡張をした Honeypot と, さらに実際の OS を使用した Honeypot で侵入ログを収集した. この 3 つの侵入ログを自然言語処理の意味解析を用い, コマンドの一つ一つの意味をベクトル表現することで, 拡張した Honeypot で収集した侵入ログが, 拡張をしていない Honeypot で収集した侵入ログよりも, 実際の OS を使用した Honeypot で侵入ログに意味的に近くなることを評価した.

キーワード:

1. SSH, 2. Honeypot, 3. 機械学習, 4. 自然言語処理

慶應義塾大学 総合政策学部  
菅藤 佑太

Approximation of high-interaction honeypots  
by Command Extension of low-interaction honeypot

I was able to make high processing system, and a moving apparatus is wide, and the OS's such as the Linux system came to be in this way used for people on a device, and, meanwhile, one's apparatus is stepped over by an unjust invasion of SSH, and I am attacked to the network apparatus of the third party from an apparatus of oneself by the spread of PCs and the system advancement of the IoT device, and a problem made in the environment that a virus and a back door are installed, and is carried out a virus an apparatus of oneself illegally occurs. Let an intruder detect it by making the environment that there is ,Honeypot as a window what kind of ways an intruder behaves when was invaded, and is easy to invade it daringly in SSH when succeeded in a login trial, and Honeypot which collect the log of the command that carried out on this occasion, and emulated behavior of Shell now again is wide, and is used, and there are few commands that this Honeypot can carry out; as a result of being implemented, and therefore being detected by an intruder to Honeypot when invasion ahead was Honeypot, and expanding the command of Honeypot by the there prior experiment, and having compared the log that collected invasion log in Honeypot which expanded Honeypot and the command that did not expand, and collected by using a probabilistic calculation method, with Honeypot which expanded Honeypot and the command that did not expand, collected invasion log more in Honeypot using the real OS, and invasion log of Honeypot which showed that could acquire a pattern of the command practice log of more intruders, and expanded the command in this study used semantic analysis of the natural language processing for the real OS in the invasion log of these three how to show approximation したのかをを to invasion log when was invaded of unjust SSH, and evaluated a semantically thing nearby in invasion log than the invasion log that collected in Honeypot which the invasion log that collected in Honeypot which expanded because a vector expressed one one meaning of the command did not expand in Honeypot using the real OS.

Keywords :

1. SSH, 2. Honeypot, 3. Machine Learning, 4. Natural Language Processing

Keio University, Faculty of Policy Management Studies  
Yuta Sugafuji

# 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	通信機器の普及	1
1.2	honeypot	1
1.3	本研究の問題と仮説	1
1.4	予備実験	2
1.5	提案手法の実装	2
1.6	本研究の評価	2
1.7	本論文の構成	2
<b>第2章</b>	<b>本研究の要素技術</b>	<b>4</b>
2.1	Honeypot	4
2.1.1	低対話型 Honeypot	4
2.1.2	製造責任と知的財産権に関する法制度	5
2.1.3	高対話型 Honeypot	5
2.1.4	SSH の Honeypot の比較	5
2.1.5	Shell	6
2.1.6	自然言語処理	7
<b>第3章</b>	<b>本研究における問題定義と仮説</b>	<b>8</b>
3.1	本研究における問題定義	8
3.1.1	SSH Honeypot の現状の問題	8
3.1.2	本研究の問題	10
3.2	問題解決のための要点	10
3.3	仮説	10
<b>第4章</b>	<b>事前実験</b>	<b>11</b>
4.1	概要	11
4.2	要素技術	11
4.3	問題定義	11
4.4	予備実験の手法	12
4.5	実装	12
4.6	評価	12
4.7	結果	12

<b>第 5 章</b>	<b>本研究の手法</b>	<b>15</b>
5.1	問題解決の為のアプローチ . . . . .	15
5.1.1	コマンドの追加実装 . . . . .	15
5.1.2	既実装コマンドの修正 . . . . .	15
<b>第 6 章</b>	<b>実装</b>	<b>16</b>
6.1	実装環境 . . . . .	16
6.1.1	純正の Honeypot で未実装のコマンドの実装 . . . . .	16
<b>第 7 章</b>	<b>評価および考察</b>	<b>17</b>
7.1	評価手法 . . . . .	17
7.1.1	評価手法の実装 . . . . .	19
<b>第 8 章</b>	<b>第 8 章</b>	<b>22</b>
8.1	関連研究 . . . . .	22
8.1.1	SSH の Honeypot . . . . .	22
8.1.2	時系列データの処理 . . . . .	22
<b>第 9 章</b>	<b>結論</b>	<b>23</b>
9.1	本研究のまとめ . . . . .	23
9.2	本研究の課題と展望 . . . . .	23
9.2.1	うんちっち～ . . . . .	23
9.2.2	うんちへの応用 . . . . .	23
	<b>謝辞</b>	<b>24</b>

# 目 次

2.1	SSH の低対話型 Honeypot と SSH の高対話型 Honeypot の比較 . . . . .	6
3.1	不正な SSH 侵入者の想定行動フロー . . . . .	8
4.1	収集した SSH の低対話型 Honeypot のデータ . . . . .	13
4.2	純正の Cowrie と修正済みの Cowrie のスコアリングによる比較 . . . . .	14
7.1	予備実験の評価の概念図 . . . . .	18
7.2	本研究の評価の概念図 . . . . .	18
7.3	評価のフロー [1][2] . . . . .	20
7.4	評価のフロー [1][2] . . . . .	21

# 表 目 次