侵入監視のための Honeypot の実装と評価

宮本　大輔，　久保　聡之，　大江　将史，　門林　雄基
奈良先端科学技術大学院大学 情報科学研究科

概　要

既存の不正アクセス対策手法は，既知の脆弱性に対する不正アクセスに対しては攻撃を遮断するなどの対策を行うことが可能である．しかし，未知の脆弱性に対する不正アクセスに対しては，不正アクセス対策を行うことができない．現在，悪意ある第三者の行動分析を行うことで，未知の脆弱性を発見する手法が注目されている．しかし不正アクセス手法は多様化し，そのツールは急激に増加しているため，行動分析が追いつかないことが問題とされている．本研究では，Honeypot を用いて悪意ある第三者にホストに侵入させ，このホストにおける行動を監視し，記録する．これより，行動分析にかかる時間を短縮し，未知の脆弱性を効果的に発見することを目的とする．

An Implementation and Evaluation of Honeypot for Intrusion Auditing

Daisuke Miyamoto, Toshiyuki Kubo, Masafumi Oe, and Youki Kadobayashi
Graduate School of Information Science, Nara Institute of Science and Technology

Abstract

Traditional defensive measures can drop well known attacks but these systems can't reject attacks which exploit new vulnerability. Attacking analysis has been a critical strategy against those penetration techniques. But penetration techniques are being diversified, and the number of exploiting tools are increasing much faster than that of analysis tools. In this paper, we present a new Honeypot to solve the analysis problem and discover new vulnerabilities efficiently.

1　はじめに

近年，オンライン取引などのミッションクリティカルなサービスを提供するホストの増加に伴い，ホストにおけるシステムの脆弱性が指摘されるようになった．悪意ある第三者は，脆弱性を持つホストに対して不正アクセスを行うことで，正当な許可を得ることなくホストの提供するサービスを制御することが可能である．

このような不正アクセスへの対策技術として，Firewall や IDS(Intrusion Detection System: 侵入検知システム) がある．Firewall は定められたルールに従い，トラフィックを破棄することができる．また，IDS はトラフィックを分析し，このトラフィックが不正アクセスであるかどうかを検知することができる．この 2 つの対策手法の組み合わせにより，IDS がトラフィックを不正アクセスと検知した場合，Firewall にこのトラフィックを破棄するよう通知することで，ホストに対する不正アクセスを防止する手法が採られてきた．

しかし IDS は既知の脆弱性に対する不正アクセスを検知することはできても，未知の脆弱性に対する不正アクセスを検知することはできない．このため，攻撃者が未知の脆弱性に対する不正アクセスが行った場合，これらの不正アクセス対策技術では，不正アクセスを防止することができない．したがって未知の脆弱性を効果的に発見し，不正アクセスに用いられないようにすることが重要である．

現在，不正アクセスにおけるトラフィックをデータとして収集し，攻撃者の行動分析を行う手法が，未知の脆弱性を発見する効果的な方法として注目されている．しかし不正アクセス手法の多様化や，不正アクセスを行う

ツールの急激な進化に，行動分析が追いつかないという問題が指摘されていた．

そこで本研究では Honeypot を用いて，ホストに侵入した悪意ある第三者の不正アクセスを監視し，後に分析を行うために記録する環境を実現する．これにより，分析にかかる時間を大幅に短縮し，未知の脆弱性を効果的に発見することを目的とする．なお，本研究でいうホストへの侵入とは，悪意ある第三者が正当な許可を得ることなく，ホストの演算装置や記憶装置等の資源にアクセスできる状態を指す．

以下，2 節において関連研究を説明し，3 節において本研究で実装する Honeypot について議論する．4 節では仮想ターミナル監視型 Honeypot を説明し，5 節ではシステムコール監視型 Honeypot を説明する．6 節では Honeypot に関する考察を行い，最後に 7 節でまとめと今後の課題について述べる．

2 関連研究

本研究は Honeypot と呼ばれる研究領域に分類される．関連する研究として The Honeynet Project [4] や Steganographic File System[2] がある．

Honeypot Project は，悪意ある第三者が攻撃に用いたツールや，攻撃の戦略などから得られた情報を公開し，実際の攻撃から学ぶことを理念として活動している．しかし，Honeypot Project の報告は主に不正アクセスに対する啓蒙を行っており，その監視手法について明白な部分が少ない．また攻撃者の行動内容を NFS や SMB を用いて記録する方針をとっているが，侵入者がネットワークへのトラフィックを監視した場合，侵入したホストが Honeypot であると検知できる．このため，Honeynet Project の目的は，ホストに侵入する前の攻撃者の行動を監視することにあると思われる．本研究ではホストに侵入した後の攻撃者の行動に着目し，監視を行う．

Steganographic File System(StegFS) は，重要なファイルを安全に記録し，ユーザから隠蔽するファイルシステムである．本研究で行った実装では，Honeypot は，監視した内容をファイルシステムに記録する．この際に StegFS のようなファイルシステムを組み合わせて用いることで，監視した記録を侵入者から隠蔽して，記録することが可能である．

3 Honeypot

本研究では Honeypot の主要な要素を以下の通りに定義する．

警告能力

侵入者の存在を検知した Honeypot は，侵入者に気付かれないよう Honeypot 管理者に警告する．

監視能力

Honeypot は不正アクセスによって侵入した悪意ある第三者の行動を監視する．監視した内容は，管理者が今現在行われている不正アクセスの内容を把握するために役立てられる．

記録能力

Honeypot は監視によって得られた侵入者の行動内容を，侵入者に気付かれないように記録する．管理者は，記録した監視内容から，ホストにおける侵入者の行動を分析することができる．

偽装能力

Honeypot は，Honeypot ではない普通のホストに似せて設計する．また Honeypot の行う全ての監視やその記録は，全て侵入者に隠さなければならない．

攻撃誘致能力

Honeypot は意図的に脆弱性を設置することで，悪意ある第三者の不正アクセスを誘致する．

防御能力

Honeypot は監視している侵入者の行動が危険であると判断した場合，侵入者のトラフィックを遮断する．また Honeypot 自身を悪用され，他のホストを攻撃されないよう注意する．

分析能力

Honeypot は，攻撃者の行動から攻撃者の能力や攻撃の動機を分析する．

本研究では，Honeypot の要素である監視能力，記録能力，及び偽装能力に着目して実装を行った．これ以外の Honeypot の要素については，関連研究との組み合わせにより補う．

Honeypot の実装には様々な方法がある．本研究では攻撃者をホストに侵入させることを意図して Honeypot を実装したが，攻撃者の侵入を許すことがないように実装した Honeypot も考えられる．この Honeypot は，不正アクセスのトラフィックをデータとして収集し，分析を行う目的で設置される．本研究では，悪意ある第三者が不正アクセスにより侵入したホストにおける行動分析を行うことで，未知の脆弱性を効果的に発見するため，ホ

ストへの侵入させる Honeypot を実装した．

また本研究で実装した Honeypot は，カーネル空間において侵入者の行動を監視し，その内容を記録するように設計した．ユーザ空間において監視を行った場合，侵入者はホストのプロセスを調査することでこのホストが Honeypot であると看破できる．この場合，Honeypot の偽装能力が満たされない．さらに 4 節で説明するバックドアを用いた侵入者の行動を監視をするには，カーネル空間における監視が必要であった．このため本研究における Honeypot の実装は，カーネルを改造することで行った．

4 仮想ターミナル監視型 Honeypot

この節では仮想ターミナルを監視することにより，侵入した攻撃者の行動を監視し，記録を行う Honeypot について述べる．

ホストの脆弱性を利用した不正アクセスは，配送経路上のルータなどでトラフィックを観測し，記録することができる．悪意ある第三者がホストの脆弱性に対する不正アクセスを繰り返し行った場合，管理者は記録した通信内容を分析することで不正アクセスの痕跡を発見することができる．しかし，不正アクセスにより侵入した悪意ある第三者は，次回からの侵入を容易にするため，ホストにバックドア（裏口）を設置することが確認された．このバックドアが暗号化された通信路を提供する場合，配送経路上のルータでトラフィックを観測・記録しても，通信内容は暗号化されているため，管理者は行動分析を行うことができない．実際に，筆者らのオペレーションの結果，悪意ある第三者の侵入を許したホストから改造された ssh サービスプログラムが発見された．ssh はリモートからホストの仮想ターミナルを使用可能にするプログラムであり，ユーザに暗号化された通信路を提供する．

データが暗号化されているため，配送経路上のルータから，ホストへの侵入者の行動を監視することは難しい．そこで本研究では，悪意ある第三者が侵入しているホストにおいて監視を行う．

4.1 設計

本研究で提案する Honeypot は，仮想ターミナルのデバイスドライバを改造し，侵入者の仮想ターミナルデバイスへのデータの入出力を監視する．たとえホストとホストの間の通信路が暗号化されていたとしても，仮想ターミナルデバイスには，復号化された通信内容が渡される．このため仮想ターミナルデバイスへの入出力を監視することで，侵入者の行動を監視することが可能である．

表 1: プログラムを追加したファイル群

`driver/char/tty_io.c`	`driver/char/console.c`
`fs/open.c`	`fs/ext2/file.c`
`net/ipv4/tcp_ipv4.c`	`net/ipv6/tcp_ipv6.c`
`net/ipv4/udp.c`	`net/ipv6/udp.c`
`includelinux/fs.h`	`include/linux/sched.h`
`kernel/printk.c`	

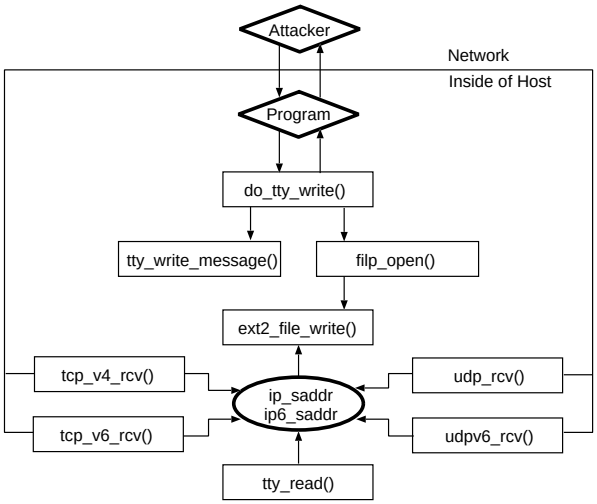

図 1: 仮想ターミナルを監視する Honeypot の設計

また Honeypot は監視だけでなく，不正アクセスが行われたことをホストの管理者に警告しなければならない．このため，仮想ターミナルへのデータ入出力をディスプレイに表示することで，システム管理者の注意を促す．

なお行動記録には，行動を監視した時刻と侵入者の IP アドレスを併せて記録する．IP アドレスは偽造することが可能であるため，この値を完全に信用することは難しいが，攻撃者がどのくらいの間隔で侵入を繰り返すのかを分析する際に役立てることができる．

4.2 実装

実装は Linux カーネル 2.2.16 上で行った．ディストリビューションは Kondara 2000 [3] であり，開発には C 言語を用いた．開発の対象は Linux カーネル 2.2.16 であり，開発したプログラムは表 1 に示す /usr/src/linux ディレクトリにおけるファイル群に追加した．

以下，図 1 を用いて説明する．仮想ターミナル監視型 Honeypot は，仮想ターミナルデバイスに入力されたデータ，仮想ターミナルデバイスから出力されたデータを `do_tty_write()` 関数において監視する．監視したデー

```
% ssh -p 9999 root@192.168.1.2
root@192.168.1.2's password: sshiscool
Last login: Mon Aug 27 18:01:31 2001 from 192.168.1.1
No mail.
[root@192.168.1.2 /root]# ls -a
.   .Xdefaults     .bash_logout   .bashrc  .jedrc  .tcshrc
..  .bash_history  .bash_profile  .cshrc   .ssh
[root@192.168.1.2 /root]# who
root     pts/0    Aug 27 18:02
[root@192.168.1.2 /root]#
```

図 2: 悪意ある第三者の不正アクセスの様子

```
Last login: Mon Aug 27 18:01:31 2001 from 192.168.1.1
No mail.
[root@192.168.1.2 /root]# ls -a
.   .Xdefaults     .bash_logout   .bashrc  .jedrc  .tcshrc
..  .bash_history  .bash_profile  .cshrc   .ssh
[root@192.168.1.2 /root]# who
root     pts/0    Aug 27 18:02
[root@192.168.1.2 /root]#
```

図 3: ディスプレイに表示される警告

タは tty_write_message() 関数を用いてディスプレイに表示し，ext2_file_write() 関数を用いて監視した内容をファイルとして記録する．

なお記録を行う際には，データの送信元 IP アドレスと，現在の時刻を合わせて記録する．この情報は，複数の悪意ある第三者が同時にホストに侵入したとき，各侵入者の行動を識別する際に役立てることができる．

4.3 評価

ここでは ssh を利用したバックドアの一実装である SSH Backdoors [1] を用いてホストに不正アクセスを行う例を示す．このバックドアは，通常の ssh サービスだけでなく，万能のパスワード sshiscool を用いて様々なユーザの権限でホストに侵入できる機能を提供する．悪意ある第三者は，このパスワードを用いて，管理者権限でホストへ無制限に侵入することができる．

実験は，攻撃者がこのプログラムを設置した後で，システムに再侵入を行うケースを想定して行った．

図 2 に攻撃者がバックドアを利用して侵入を行う様子を示す．悪意ある第三者は，ssh クライアントを用いて，9999 番ポートに起動したバックドア付きの sshd からの侵入を試みている．

図 3 に，仮想ターミナルの監視した内容を，ディスプレイに表示している様子を示し，図 4 に仮想ターミナルへの入力を，図 5 に仮想ターミナルからの出力を記録したファイルの内容を示す． 管理者はディスプレイを監視することで，侵入が行われていることを検知することが

```
3b8a1876,192.168.1.1,l
3b8a1876,192.168.1.1,s
3b8a1876,192.168.1.1,
3b8a1876,192.168.1.1,-
3b8a1877,192.168.1.1,a
3b8a1877,192.168.1.1,^M
3b8a187c,192.168.1.1,w
3b8a187c,192.168.1.1,h
3b8a187c,192.168.1.1,o
3b8a187c,192.168.1.1,^M
```

図 4: 仮想ターミナルへの入力を監視したファイル

```
3b8a1871,192.168.1.1,Last login: Mon Aug 27 18:48:20 2001 192.168.1.1^M

3b8a1871,192.168.1.1,No mail.

3b8a1871,192.168.1.1,[root@192.168.1.2 /root]#
3b8a1876,192.168.1.1,l
3b8a1876,192.168.1.1,s
3b8a1876,192.168.1.1,
3b8a1876,192.168.1.1,-
3b8a1877,192.168.1.1,a
3b8a1877,192.168.1.1,

3b8a1877,192.168.1.1,.   .Xdefaults     .bash_logout   .bashrc  .jedrc  .tcshrc

3b8a1877,192.168.1.1,..  .bash_history  .bash_profile  .cshrc   .ssh

3b8a1877,192.168.1.1,[root@ /root]#
3b8a187c,192.168.1.1,w
3b8a187c,192.168.1.1,h
3b8a187c,192.168.1.1,o
3b8a187c,192.168.1.1,

3b8a187c,192.168.1.1,root     pts/0    Aug 27 18:52

3b8a187c,192.168.1.1,[root@ /root]#
```

図 5: 仮想ターミナルからの出力を監視したファイル

可能である．また，仮想ターミナルへの入出力を監視し，その内容を記録したファイルは，後に行動分析を行う際に役立てることができる．

以上，仮想ターミナル監視型 Honeypot を実装し，暗号化された通信路を提供するバックドアが設置された場合においても，侵入者のホストにおける行動を監視し，その内容を記録できることを示した．

5 システムコール監視型 Honeypot

2 節で説明した Honeypot は，仮想ターミナルデバイスを監視することによりバックドアを用いた侵入を行う悪意ある第三者の行動を監視する．しかし，仮想ターミナルデバイスへの入出力を監視するという手法が普及した場合は，侵入者は，仮想ターミナルデバイスを用いないバックドアを設置し，侵入経路を確保するようになると考えられる．

この将来的に起こりうる問題を解決するため，本研究ではホストにおいてファイルの読み書きや実行，プロセスの制御を行うシステムコールを監視し，その内容を記録する機能を持つ Honeypot も実装した．

表 2: プログラムを追加したファイル群

kernel/signal.c	kernel/printk.c
fs/open.c	fs/ext2/file.c
fs/exec.c	mm/filemap.c
net/ipv4/tcp_ipv4.c	net/ipv6/tcp_ipv6.c
net/ipv4/udp.c	net/ipv6/udp.c
includelinux/fs.h	include/linux/sched.h
driver/char/console.c	

5.1 設計

システムコール監視型の Honeypot は，ファイルの読み書きなどを行う際に使われるシステムコールを監視する．悪意ある第三者が，仮想ターミナルを利用せずに侵入しても，侵入したホストでファイルを実行したり，読み書きを行う際には，ホストにおいてシステムコールが呼び出される．このためシステムコールを監視することで，侵入者の行動分析を行うことが可能である．

システムコール監視型 Honeypot は，仮想ターミナル監視型 Honeypot と同様に，システムコールを監視した内容をディスプレイに表示することで警告を行い，後に分析を行うために記録する．

5.2 実装

実装環境は仮想ターミナル監視型 Honeypot 同じ環境である．開発したプログラムは表に示す /usr/src/linux ディレクトリにおけるファイル群に追加した．

以下，図 6 を用いて説明する．

本研究で実装した Honeypot は，Linux における signal システムコールである `send_sig()` 関数，Linux で標準的に用いられる Ext2 ファイルシステムにおける open/read/write/execve システムコールである `filp_open()` 関数，`generic_file_read()` 関数，`ext2_file_write()` 関数，`do_execve()` 関数を監視し，その内容を記録する．また，open/write システムコールにおいて監視した内容を記録する際にも，open/write システムコールが実行されるため，再帰的な呼び出しを防ぐ必要がある．このため，新たに作成した `honey_filp_open()` 関数を用いて監視内容を記録するためにファイルを開き，`honey_filp_write()` 関数を用いて監視内容をファイルに記録する．さらに，仮想ターミナル監視型 Honeypot と同様に，記録を行う際には，データの送信元 IP アドレスと，現在の時刻を合わせて記録する．

Honeypot は，監視した内容を `tty_write_message()` 関数を用いて，ディスプレイに表示し，管理者に警告を行う．本実装では，全てのシステムコールの監視内容をディスプレイに表示することも可能であった．しかし Honeypot が警告を行うという意味において，管理者がホストに侵入した悪意ある第三者の行動が把握できれば十分であると考えた．このため，`do_execve()` 関数や `send_sig()` 関数の監視内容のみを表示するように実装した．

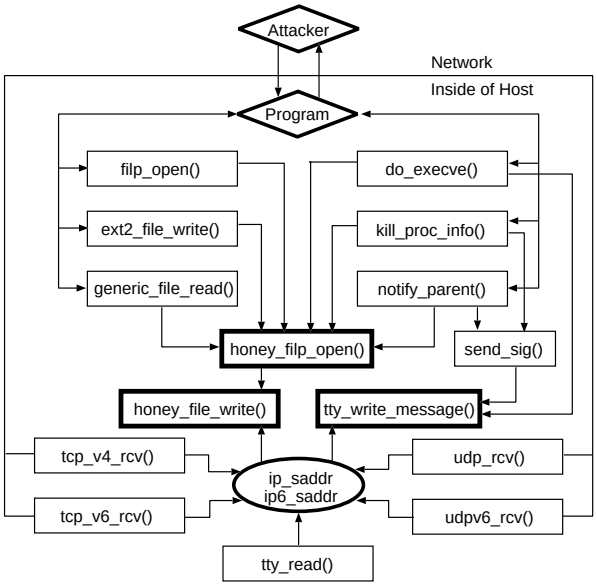

図 6: システムコールを監視する Honeypot の設計

```
/bin/ps -C httpd
  544 ?        S      0:00 httpd
  548 ?        S      0:00 httpd
（中略）
kill -KILL 544
```

図 7: http サーバを停止する不正アクセスの様子

5.3 評価

ここでは仮想的に攻撃を行い，ホストにおけるプロセスを停止したり，ファイルを改竄したりする例を示す．

図 7 に，侵入者が http サーバのプロセスを調査し，停止するまでの不正アクセスの様子を示す．この例では，侵入者は `ps` プログラムを用いて http サーバのプロセスを調査し，`kill` プログラムを用いて http サーバのプロセスを停止している．

図 8, 図 9 はそれぞれ exec システムコール，signal システムコールにおける監視内容をディスプレイに表示している様子を示す．exec システムコールはどのファイルを実行したのかを監視し，signal システムコールはどの

```
192.168.1.1,/bin/ps -C httpd
```

図 8: exec システムコールの監視内容

```
192.168.1.1,process=bash,pid=31030,sig=17
192.168.1.1,process=httpd,pid=544,sig=9
```

図 9: signal システムコールの監視内容

プロセスにどのような制御を行ったのかを監視する．図 9 における下線で示した部分は，行ったプロセスに対して送信した制御シグナルの種類である．この場合は，9 番シグナル (KILL) が送信され，プログラムを停止したことを示している．なお，kill プログラムはシェルビルトインコマンドであるため，ファイルの実行を行う exec システムコールでは監視されていない．

管理者は 図 8, 図 9 のようにディスプレイに表示された監視内容から，不正アクセスが行われたことを知ることができる．また，この監視内容はファイルにも記録されており，侵入者の行動分析を行う際に役立てることができる．

次に，図 10 に侵入者がユーザのアカウントを編集し，パスワードを改竄する不正アクセスの様子を示す．侵入者は passwd プログラムを用いてユーザ guest のパスワードを foobar に変更するよう試みている．

図 11, 図 12, 図 13 は，それぞれ open/read/write システムコールにおける監視内容を記録したファイルの内容である．悪意ある第三者が passwd プログラムがパスワードを編集するために用いた open/read/write システムコールにおいて，どのファイルを対象として読み書きを行ったのかを記録することができる．

以上，仮想ターミナル監視型 Honeypot が将来的普及した場合に想定されるバックドアに対する取り組みとして，システムコール監視型 Honeypot を実装し，侵入者のファイルの読み書きや実行，プロセスの制御などの不正アクセスを監視し，その内容を記録できることを示した．

6 考察

本研究では，カーネル空間において監視を行い，その内容を記録する設計を用いた．しかし LKM(Loadable Kernel Module) などを用いて，侵入者がカーネルの行動を調査できる可能性がある．この場合，侵入したホストが Honeypot であると看破されてしまう可能性がある．

```
/usr/bin/passwd guest
Changing password for user foobar
New UNIX password: foobar
passwd: all authentication tokens updated successfully
```

図 10: パスワードを改竄する不正アクセスの様子

```
3b9c879f,192.168.1.1,/etc/pam.d/passwd
3b9c879f,192.168.1.1,/lib/secruty/pam_pwdb.so
3b9c879f,192.168.1.1,/lib/secryty/pam_cracklib.so
3b9c879f,192.168.1.1,/etc/ld.so.cache
3b9c879f,192.168.1.1,/usr/lib/libcrack.so.2.7
3b9c879f,192.168.1.1,/etc/pam.d/other
3b9c879f,192.168.1.1,/lib/security/pam_deny.so
3b9c879f,192.168.1.1,/etc/pwdb.conf
3b9c879f,192.168.1.1,/etc/passwd
3b9c879f,192.168.1.1,/etc/shadow
（後略）
```

図 11: open システムコールの監視内容

したがって，LKM の処理ルーチンを改造し，侵入者が LKM から得られる情報を改竄したり，カーネルの行動に対する調査を無制限に許可しないよう抑制することが重要である．

また本研究では，侵入者が人間であることを仮定し，未知の脆弱性を用いた攻撃を行うケースを想定している．しかし不正アクセスは自動化される傾向にあり，侵入したホストから自動的に不正アクセスを行うツールが出現している．このようなツールは，プログラムに記述された不正アクセスしか行わない．このため本研究で実装した Honeypot が，自動的に行われる不正アクセスを分析するのに適しているかについて議論が必要である．

さらに本研究では Honeypot の実装と評価を行ったが，運用に当たっては解決すべき問題がある．本研究で実装した Honeypot は，侵入者の行動が危険であるかどうかの分析を行わない．したがって，侵入者の行動により，Honeypot を悪用し他のホストに対して不正アクセスを行うこともできる．本来 Honeypot はこの不正アクセスを監視する環境であるが，運用に当たっては Honeypot の表示する警告を注意深く調査し，他のホストへの不正アクセスを阻止する必要がある．

最後に本研究による Honeypot は，ホストに対する正当な権限を持つユーザと，不正アクセスにより侵入した攻撃者の区別を行わない．このため安易な行動分析を行うことで，ユーザのプライバシーを侵害する恐れがある．Honeypot を運用するには，予め倫理的な枠組みを定め，この枠組みの適応範囲で運用することが望ましい．

```
3b9c879f,192.168.1.1,/usr/bin/passwd
3b9c879f,192.168.1.1,/usr/bin/passwd
3b9c879f,192.168.1.1,/usr/bin/passwd
3b9c879f,192.168.1.1,/lib/ld-2.1.3.so
3b9c879f,192.168.1.1,/lib/ld-2.1.3.so
3b9c879f,192.168.1.1,/lib/libdl-2.1.3.so
3b9c879f,192.168.1.1,/lib/libpam.so.0.72
3b9c879f,192.168.1.1,/lib/libpam_misc.so.0.72
3b9c879f,192.168.1.1,/lib/libpwdb.so.0.61
（後略）
```

図 12: read システムコールの監視内容

```
3b9c879f,192.168.1.1,/etc/pwd.%d.31236
3b9c879f,192.168.1.1,/etc/passwd-
3b9c879f,192.168.1.1,/etc/passwd+
3b9c879f,192.168.1.1,/etc/spwd.31236
3b9c879f,192.168.1.1,/etc/shadow+
```

図 13: write システムコールの監視内容

7　おわりに

本研究は，効果的に未知の脆弱性を発見するための行動分析を行う環境として Honeypot に着目した．また，暗号化された通信路を提供するバックドアを用いて侵入が行われた場合も，仮想ターミナル監視型 Honeypot が侵入者の行動を監視することができ，その内容を記録できることを示した．さらに，仮想ターミナル監視型 Honeypot が普及した場合，将来的に行われるバックドアについても想定し，システムコール監視型 Honeypot を実装した．これを用いて，侵入者が Honeypot においてファイルの読み書きや実行，プロセスの制御を行った場合に，これらの行動を監視することができ，その内容を記録できることを示した．これより，侵入者の行動分析を行うことで，未知の脆弱性を効果的に発見し，新しい不正アクセス対策手法のための知見を得ることが期待できる．今後の課題として，悪意ある第三者の不正アクセスを収集した Honeypot が自動的に分析を行い，その結果を IDS へ通知するシステムの開発が挙げられる．

参考文献

[1] SSH Backdoors. http://ns2.crw.se/~tm.

[2] Andrew D. McDonald and Markus G. Kuhn. Stegfs: A steganographic file system for linux. In *Information Hiding*, pages 462–477, 1999.

[3] Kondara MNU/Linux. http://www.kondara.org.

[4] HonetNet Project. http://project.honeynet.org.