サーバ保護のための異常時切替えハニーポットの提案と試作

M2004MM036 大藤 純一
指導教員　長谷川 利治

1　はじめに

現在，コンピュータやネットワークの普及とともに不法侵入や情報の外部漏洩などの問題も増加し問題となっているため，セキュリティに関する様々な方法論や技術が提案，開発されている [7]．本研究では，セキュリティ分野でも特殊な存在であるハニーポットに注目する．

ハニーポットとはネットワーク上にわざと侵入しやすいホストを作り，ここに侵入者を誘き寄せて行動記録を調査することによって，攻撃者やウイルスの動向を探るものである [3]．

しかし，既存のハニーポットはそうした攻撃者の「監視」や「調査」は行うが、メインサーバの保護などは行わないため，ハニーポットを仕掛けたネットワークに対する不正アクセスを助長する，といった問題がある [4]．

そこで本研究では，通常はメインサーバとの通信を行い，異常が発生した場合にハニーポットサーバへ通信を切替えるシステムを提案し試作する．アプリケーション毎に通信状態の監視を行い，通常の状態とは異なる「ふるまい」を検出した場合に，別に構築した攻撃者を欺くための仮想ホストが応答を行い，メインサーバの保護，そして攻撃者のブラックリストの作成などを行う．

この方法ではハニーポットをアプリケーション毎に用意する必要があるため，共通の枠組みとしてサーバプログラムを正常時用，異常時用で動かし，正常時にはその両方がデータを受け取り応答するが，クライアントには正常サービスの応答のみを中継させる．そして異常時には，inetd のようなスーパーサーバのしくみを改造，セッションをハイジャックする，または divert socket を利用し，外部からの通信を横取りしメイン，サブ双方に送信，サーバからの返事を再び横取りし正常時はメインからの通信をクライアントに送信，異常発生時は切替えサブサーバからの通信をクライアントに送信する，といった状況に応じた方法を取り，仮想サーバへの切替えを行う．これにより既存のサーバプログラムを変更することなく，提案するシステムを組み込むことができ，それらを利用して警告出力のスコア表示，関連付けを行うことを目標とする．

2　システムの提案

この節では，ハニーポットの説明と問題点，提案するシステムについて述べる．

2.1　ハニーポットとは

ハニーポットとはクラッカーに対してあたかも本物のシステムであるかのように見せかけるおとりシステムであり，攻撃者をおびきよせるように設定されているサーバやネットワーク機器を指す．

ハニーポットには実際の OS を使用する方法と仮想 OS を使用する方法があり，実際の OS を使用する場合は詳細なデータを取ることが可能だが危険度が高く，仮想 OS を使用する方法は，限られたサービスを提供するので，セキュリティリスクを最小限に抑えることが可能となる．

しかしどちらの場合でも，ハニーポットは攻撃者を探ることが目的であり，メインサーバの保護などは行わない．

2.2　提案するシステム

本研究では，既存のハニーポットの問題点を解決するため，メインサーバの保護を行いつつ攻撃者の動向を探るための切替え型ハニーポットシステムを提案する．

本研究で提案するシステムは以下のような利点を持つ．

- 通常時はメインサーバがクライアントとのやりとりを行うことができる．
- 攻撃者が現れると，仮想サーバへの通信の切替えを行うので，メインサーバの保護が行える．
- すでにあるルータとなるホスト，もしくはメインサーバに組み込むことで，既存のサーバプログラムを改良することなく使用することが可能である．

2.3　既存の切替え方式

現在研究が進められているハニーポットツールにも切替え型ハニーポットがあり，主に以下の 2 つに分けられる．

- ネットワーク，ポートを見張る方式
- ブラックリストを作成する方式

アドレス，ポート監視型のハニーポットでは，指定したアドレスやポートへの通信は全てハニーポットへ転送するため，正規サービスの妨げやメインサーバの保護などに支障がある [5]．また，事前にブラックリストを作成する方法では，既知の攻撃の場合や悪意のあるユーザが以前攻撃を行った場合のみに反応し，また通信中に IDS からの警告があった場合も対処することができない [1]．

そこで本研究では応用層通信中での切替えを行い，既存のサーバプログラムを変更することなくシステムを構築する方法を提案する．以下にそれが実現可能と思われるいくつかの切替え方式を示す．

2.4　TCP 応用層通信途中の切替え方式

アプリケーションレベルで通信を見張り，セッション中に切替えを行う．攻撃者に切替えを悟られないため

に，仮想サーバは切替え直前の情報を引き継いで，反応を返す必要がある．これには，いくつかの方法が考えられる．

inetd のようなスーパーサーバのしくみを改造する

inetd とは，稼働しているホストが持つ全てのネットワークのインタフェースに対する接続の待ち受けを行うサーバプログラムである．同種のプログラムに指定したネットワークインタフェースに対するアクセスのみに処理を行う xnetd などがある．

inetd，xinetd などの INET スーパーサーバでは，socket はスーパーサーバで開いてそのつど稼働するサーバプロセスは標準出力にアプリケーション出力をする．しかし常時起動のデーモンには使えない，

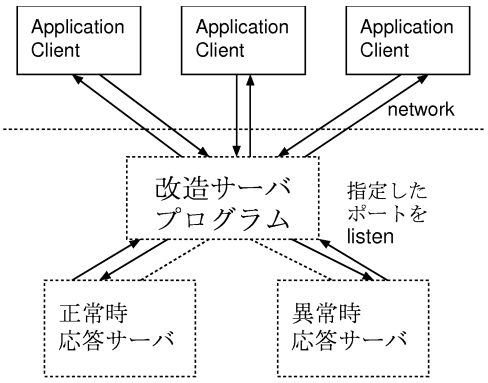

図 1　改造サーバによる切替え

指定 TCP ポートを listen して，接続後のサーバをメインサーバとハニーポットサーバの 2 つを起動し，TCP データを 2 つのサーバに標準出力で渡す．異常が生じるまではメインサーバの応答を使う．ハニーポットサーバにも同じデータを渡し応答は受け取るが異常が生じるまでは無視し，IDS からのアラームがあれば，ハニーポットサーバからの応答に切替える．その後の通信は，ハニーポットサーバが引き継ぎ，メインサーバは終了する．

TCP セッションをハイジャックする

セッションハイジャックとは，確立した通信路を横取りする行為のことを指し，なりすましに用いられる．

TCP には，信頼性と順番通りパケットを配信する機能がある．これを実現しているのは確認応答 (ACK) パケットと，シーケンス番号である [6].

まずホスト自体をハイジャックする．同一 LAN 内のメインサーバとは異なる IP アドレスを用意し，ARP 応答を偽り IP 通信を乗っ取る．そして，クライアントへメインサーバからの TCP 応答を推測して代わりに通信を行う．この時，シーケンス番号を予想する，もしくは途中経路のモニタを行う．

Divert Socket による切替え

IP パケット取扱いのためのソケットメカニズムに，Divert Socket がある．Divert Socket とは IP 層での IP

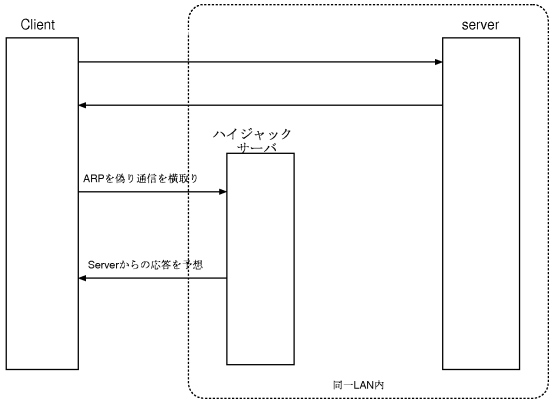

図 2　ハイジャックによる切替え

パケットの横取りや再注入を行うことができる socket であり，通常は FreeBSD カーネルに IPv4 のみに含まれている．一般的には NAT 機能をユーザプロセスで実現する natd などで使用されているが，Linux 用のカーネルも用意されている．

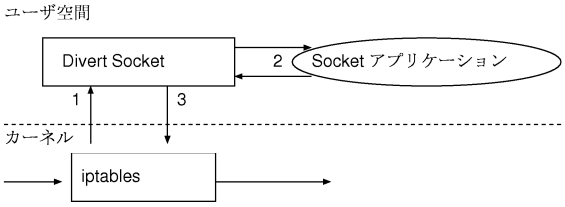

図 3　divert socket

Divert Socket の特徴は，ユーザプロセスにパケットを出し，再び受けられることである．この特徴によって，メインサーバとサブサーバへパケットを送り，また受け取ることができる．また，iptables で指定したルール毎に最大 65536 ポートの Divert Socket に振り分けることができるため，多数の異なる処理を宛先の異なるネットワーク，ホスト，プロトコル，ポートに指定できる．

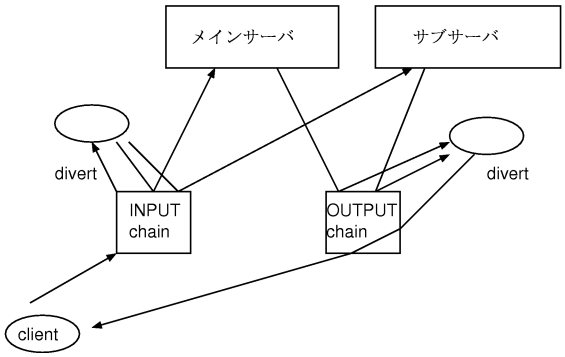

図 4　同一ホストの場合の Divert Socket による切替え

同一ホストで切替えシステムを実現した場合，正常時

は以下のような流れとなる．

1. 同一ホスト内で，同じサービスをポートのみを変えて 2 つ起動
2. INPUT 用，OUTPUT 用の Divert socket を用意する．
3. クライアントから通信があった場合，iptables の INPUT chain のルールに従い INPUT 用の divert socket に渡す
4. メインサーバへはそのまま送信，サブサーバへは宛先ポートを変更し送信する
5. メイン, サブからの通信があった場合, iptables の OUTPUT chain のルールに従い OUTPUT 用の divert socket に渡す
6. メインサーバからの通信のみをクライアントへ返す
7. 3-6 を通信が終了するまで繰り返す

異常が発生した場合は，外部に警告プログラムを作成しておき，切替えシステムへと通知する．通知があった場合は 4 をサブサーバのみへの通信へ切替え，6 をサブサーバからの通信をクライアントへ返すようにする．

2.5 メインサーバの通信の切断

切替えを行った後は，メインサーバへの通信のやりとりは行われないため，切替えを行った後はメインサーバはコネクションを切断する必要がある．本研究では切替えプログラムが切替え指示を受け取った場合にメインサーバへ切断の指示を出すようにプログラムする．

切断の方法には，RST パケットを送る方法がある．本研究のシステムの場合は切替えを行う直前のクライアントとメインサーバのコネクションを引き継ぎ，シーケンス番号を合わせてメインサーバへ RST パケットを送信し強制切断を行う．

2.6 異常検出と切替え指示

切替えシステムを実現するにあたって，異常を発見し切替え指示を出す必要がある．本研究では異常発見ログを見張り，異常があった場合に切替えシステムへ指示を出す．その方法に付いて説明する．

異常検出方法

通信状態の異常を知る方法として、ネットワーク型不正侵入検知システムがある．これはネットワークを流れるパケットを監視し，シグネチャ型と呼ばれる既知の攻撃パターンとのパターンマッチングにより異常を検知する方法，もしくはアノマリ型と呼ばれる正常な状態を定義しておいて，それと現在の状況を照らし合わせてある一定の値を越えた場合に異常と判断する方法がある [7]．

NIDS によって検出されたパケットは，アラートログとして書き出される．そのログファイルを監視し，異常と判断した場合は切替えプログラムに切替え要求を出す．

ネットワーク型不正侵入検知システムの中には，異常を検出した場合，他プロセスへ通知を行う事ができるものもあり，そのようなネットワーク型不正検出システムのひとつに snort がある．snort では次のような処理を行うフリーの NIDS である．

- パケットのデコード．
- ルールに記述されたシグネチャとパケットとの間でパターンマッチングを行う．
- 不正を検知した場合は警告を通知する．

snort はカスタマイズ性が非常に高く，個別のルールやルールカテゴリに優先度を割り当てることで，特定のアプリケーション毎に監視を行いリアルタイムで警告，またはなんらかのコマンドを実行することが可能となる [7]．

切替え指示

本研究では異常検出ログを見張り，異常発見時に切替え指示を切替えシステムに通知するプログラムの作成を目指す．動作中のプロセスに指示を与える方法に SIGNAL がある．送信側は

```
$ kill -HUP プロセス番号
```

で送信し，受信側は

```
signal(SIGHUP，ハンドラ関数)
```

で通知を受け取り，ハンドラ関数で切替え用の状態変数を変更する．

3 システムの試作

この節では、試作したシステムと，その使用について説明する．実際に作成したシステムの構成と通信の流れについて説明する．

3.1 同一ホスト内でシステムを実現

この節では，同一ホスト内でのシステムの実現方法について説明する．

divert socket, iptables の設定

この節では，divert socket，iptables の設定について示す．

Vine Linux で divert socket を利用するために，カーネルコンパイル (再構築) を行う．

iptables の設定は以下のように行う．

INPUT 用のルールは

```
iptables -A INPUT -p tcp --dport（正規サービスのポート）-j DIVERT（Divert IN 用のポート）
```

と設定する．

OUTPUT 用のルールは

```
iptables -A INPUT -p tcp --sport（正規サービスのポート）-j DIVERT（Divert OUT 用のポート）
iptables -A INPUT -p tcp --sport（仮想サービスのポート）-j DIVERT（Divert OUT 用のポート）
```

のように設定する．

3.2 システムの通信状態遷移

以下に示すのは本研究における異常時切替えを行った場合の通信の状態遷移である [6].

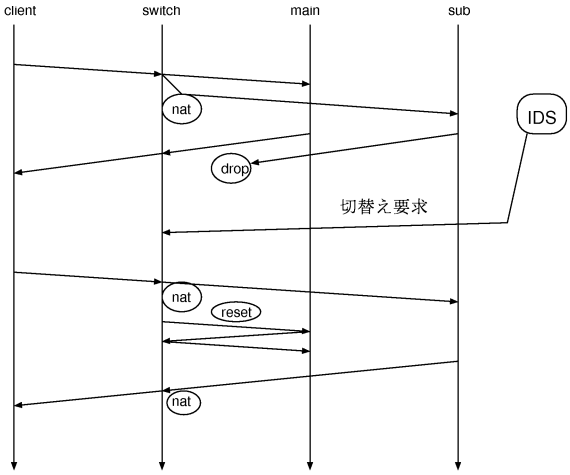

図 5　切替えシステムによるサーバ保護の状態遷移

メインサーバとサブサーバでポートを変え，同じサービスを起動する．iptables によって，外部から特定のサービスへアクセスがあった場合，INPUT 用の Divert Socket のポートへ変換する．Divert socket は通常時はクライアントからの通信を切替えシステムがメイン、サブ両方に転送する．そしてサーバからの返事は OUTPUT 用の Divert Socket を用い，通常時はメインからの返事のみをクライアントへ返し，サブサーバからの返事は破棄する．IDS 記録監視プログラムからの警告を切替えシステムが受け取った後は，クライアントから通信があった場合，サブサーバへのみクライアントからの通信を転送し，メインサーバへ切替えプログラムから TCP RST をシーケンス番号を合わせて送信し，接続を終了する．その後はサブサーバからの返事のみをクライアントへ返し，クライアントの動向を探る．通信が終了したら元に戻し，次の TCP セッションに備える．

3.3 複数サーバを用意した場合のシステムの実現

複数のサーバを用いた場合のシステムの構成について説明する．これには実際にサーバを用意する方法と仮想サーバを構築する方法がある．

システムの概要

複数サーバを仮想ホストで実現した場合のシステムの構成について説明する．

単一ホストで異なる IP アドレスをつけた仮想サーバを User Mode Linux によって実現し，メインサーバと同じサービスを模倣する．

User Mode Linux

User Mode Linux とは，Linux カーネルや Linux のアプリケーションを安全に使用することのできる，Linux 上で動く Linux である．本来の Linux マシンを危険にさらすことなく，バグの多いアプリケーションや開発途上の実験的なカーネルやシステムを実行することができる．

User Mode Linux は，実際のコンピュータコンピュータの上に仮想マシンを作り，実際のマシンが提供するものと同等のサービスを提供することができる．

User Mode Linux は，仮想ホストを作成するデーモンプロセスであり，複数の仮想ホストを作成することができる．User Mode Linux によって作成される仮想ホストの OS は実在する OS ではなく，User Mode Linux によってあたかもその OS であるように見せかけている実在しない仮想 OS である．また User Mode Linux は，IP アドレスを持たせることができる．

4　おわりに

本研究では，サーバ保護を行うと同時に攻撃者の動向を調べるための切替えプログラムを製作しハニーポットを実現した．いくつかの切替え方式を提案し，divert socket を用いたシステムの実現を図った．しかし実現できたのは divert socket で入力の通信を横取りしメイン，サブへの双方へ送信する，メイン，サブからの返事を受け取り片方を送信する，メインサーバを強制的に切断するに留まった．今後の課題としては，複数のプログラムを一つのプログラムにまとめ動作するように書き直す必要がある．また一つのサーバに対して複数コネクションがある場合，それらのうちどれを切替えるのか，外部の IDS 出力監視から通知を行うプログラムを作成する．また，実ネットワーク上で使用し実験する必要がある．

参考文献

[1] Bakos,G. : Tiny Honeypot, http://www.alpinista.org/thp/(参照 2005-8).

[2] Barrett,J.D., Silverman,E.R., Byrnes,G.R. : LINUX セキュリティクックブック, オライリージャパン.(2003)

[3] Kurtz,G., Schneier,B., Huger,A., Roesch,M., Granick,J., Spitzner,L. : The Honeynet Project, http://project.honeynet.org/(参照 2005-8).

[4] Preston,E., Mitchell,A., Wendland,M., Livingston,B., Bolin,R., Helft,M., Biever,C.: Project HoneyPot, http://www.projecthoneypot.org(参照 2005-8).

[5] SourceForge.net : Bait And Switch Honeypot System, http://sourceforge.net/projects/baitnswitch/(参照 2005-8).

[6] 笠野 秀松 : 通信プロトコル辞典, アスキー出版局 (2003).

[7] 渡辺 勝弘、飯原秀明 : 不正アクセス調査ガイド, オライリージャパン (2001).