仮想ホストを用いた攻撃パターンの収集と分析

安藤 純一　　壁屋 喜規　　後藤邦夫
南山大学数理情報学部情報通信学科
goto@it.nanzan-u.ac.jp

概要

本研究では、honeypot と呼ばれるセキュリティ管理が不十分な状態にみえる仮想ホストを設置し、攻撃者からの通信やコマンド操作を記録し、分析することを試みた。オープンソースの honeyd を利用して、2003 年 11 月 19 日から 2004 年 1 月 8 日までの約 50 日間、WindowsNT, Windows2000 Server, Linux の 3 つの仮想ホストを運用した。

典型的な例として HTTP, SMTP, TELNET の偽応用サービスプログラムを作成し、攻撃の詳しい挙動の調査を計画した。残念ながら、受けたアクセスはスキャンにとどまり、実際に被害を与えるためのアクセスはほとんど観測できなかった。したがって、既知の攻撃の詳細と新種の攻撃の情報は収集できなかった。

一方で、統計的情報は十分に得られた。各仮想ホストに 1 万回以上のアクセスがあり、仮想ホストが攻撃者に知られるまでの時間、攻撃の時間帯分布、アクセスが多いポート、年末年始の急増、攻撃を試みるホストが位置する国の割合などが明らかになった。

情報を多く収集するためには、より長期間の運用が必要ではあるが、honeypot 設置が攻撃に関する情報収集に役立つ見通しが得られた。

1　はじめに

インターネットの普及につれて、いたずらや業務妨害を目的とした不正アクセスが増加している。クラッキングツールを使用すれば、技術的知識は必要なく、子供でも容易にいたずらが可能である。また、悪意がなくても自分のコンピュータにワーム感染を許すと、他のホストへの加害行為に加担することになる。現在のおもなネットワークセキュリティ対策は、LAN への出入口または各ホストにおける通信の制限と監視で、それらは、それぞれ既知の攻撃に対する予防ならびに攻撃の試みの検出を目的とする。一方で未知の攻撃に備えるために日常的に攻撃パターンを調査する必要がある。

攻撃パターンとその攻撃が成功したさいに想定される被害、さらに攻撃者の挙動を調査するためには、攻撃者に対し、ある段階まで攻撃が成功したように見せかけなければならない。そのために故意にセキュリティ管理が不十分な状態で、攻撃者からの通信やコマンド操作を記録することを目的として設置されるホストは honeypot と呼ばれる。Honeypot の運用方法は大別して、実ホストと仮想ホスト利用の 2 つである。前者では実際に被害を受けるホストを設置するが、後者では仮想ホストで攻撃を受けて被害を受けたように見せかける。前者の実ホストを利用する方法は、危険が大きく、また常時監視も必要なので、運用が大変であるが、後者は比較的簡単に運用できる。

そこで本研究では、オープンソースの honeyd[1] を用い仮想ホストで honeypot を実験運用する。単一ホストで複数の OS の仮想ホストを模倣する honeyd の基本機能に加え、いくつかの偽応用サービスプログラムを作成し、各応用サービス固有の脆弱性を模倣する。初回の攻撃が失敗に終わった場合は、次に来る攻撃の情報は得られないので、偽応用サービスでは次の攻撃を誘うために、初回の攻撃を成功と見せかける。

得られた攻撃パターンを記録し分析することによって、既知の攻撃の第二段階の通信内容の把握、未知の攻撃を発見する。さらに、ホストの設置以後の攻撃数の増減から、脆弱性を持つホストが攻撃者に知られるまでの時間の推定と年末年始などの季節的変動も分析する。

本研究の成果は、攻撃者あるいは攻撃プログラムの挙動を予測し、対策することに利用できる。

次節では、実験システムの構成を述べる。第 3 節では、応用サービスの分類と模倣する応用サービスの例について説明する。第 4 節では、実験で得たアクセス記録を分析し、設定した 3 つの仮想ホストが受けた攻撃の、時間変化、パターン、攻撃をしたホストの国の分布を調べた。最後にまとめと今後の課題を述べる。

2 実験システムの構成

本節では実験システムの構成を述べる。

まず honey pot の種類を述べ、次に honeyd[1] のしくみと仮想 OS 機能を説明する。最後に実験環境を述べる。

2.1 Honeypot の種類

Honeypot とは、セキュリティの不十分な状態でホストを設置し、攻撃者から攻撃を受け、その行動をログとして記録するものである。由来は、はちみつ (honey) が入ったつぼ (pot) で、悪意のある攻撃者をおびきよせる ‘甘いわな’ である。

Honeypot には 2 つの種類がある。一つは、脆弱性を持つ本物のホストである。本物の応用サービスを利用する機会を攻撃者に与え、攻撃者の送ったすべての通信データやコマンド入力を記録することができる。この種の honey pot では、攻撃の踏み台にならないように厳重なセキュリティ対策を honeypot の外で施す必要がある。

他方は、安全なホストの中で稼働する仮想ホストである。この仮想ホストでは、攻撃者に対し、OS を偽り、アクセスが成功したとみせかけ、挙動を監視する。この種の HoneyPot では、本物の OS と応用サービスプログラムは使用しないので、実害を被る危険は少ない。

本研究では安全に実験するために後者の honeypot を設定する。仮想ホスト型 honeypot の市販製品では ManTrap[2]、SPECTER[3] などが知られているが、本研究では、以下の理由で honeyd を利用する。

- 調査した範囲で唯一のオープンソース honeypot である。
- 仮想 OS を用いるので、セキュリティ面においてリスクが小さい。
- 提供するサービスの拡張定義が容易である。

2.2 Honeyd の機能と応用サービスの起動

Honeyd[1] は、実ホストの OS のデータリンク層のデータ (フレーム) を直接読み書きする方法で複数の仮想ホストを実現するデーモンプロセスである。各仮想ホストには実ホストの IP アドレスと異なる IP アドレスを持たせ、代理 ARP で仮想ホスト宛のパケットを受信する。仮想ホスト宛のパケットは OS カーネルではなく honeyd プロセス内のプロトコル処理スタックで処理される。処理手順は以下の通りである (図 1)。

1. ネットワークから OS のデータリンク層を経て届いたフレームを、Packet Dispatcher で処理し、honeyd 内の ICMP, TCP, UDP 処理スタックに振り分ける。
2. TCP, UDP の場合はユーザが定義した応用サービスプログラムに渡す。
3. Personality Engine で異常なパケットに対する OS 固有の応答をまねて OS のデータリンク層を経由してフレームをネットワークに出力する。

本研究では、Linux の実ホストで稼働する honeyd で Windows2000Server(IIS5.0)、WindowsNT4.0(IIS4.0)、Linux の 3 つの仮想ホストを 1 つずつ設定し、代表的な応用サービスを模倣するプログラムを作成し honeyd から起動する。

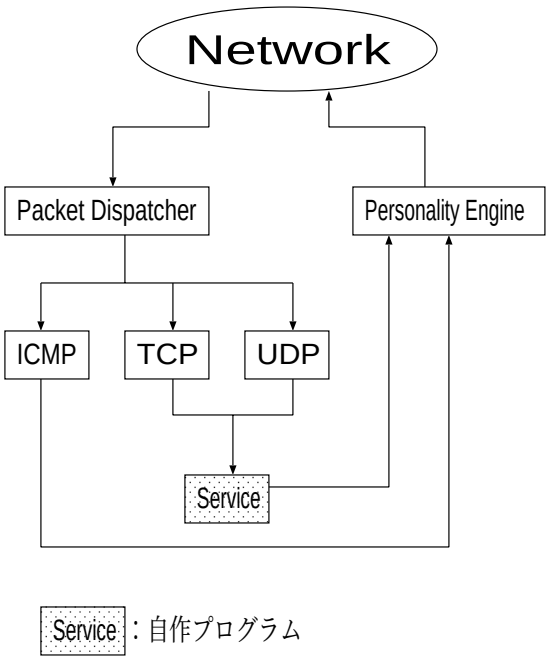


図 1: Honeyd のプロトコル処理

Windows 系 OS を模倣する理由は、多くのセキュリティホール (脆弱性) が指摘されており、各種のネットワークサービスが攻撃者からの攻撃を受けやすいことと、Web サーバソフトウェアとして利用が多い IIS (Internet Information Server) に既知のセキュリティホール (脆弱性) が多いことである。Linux の模倣は、攻撃される回数と攻撃パターンを Windows 系 OS の場合と比較するためである。

仮想ホストの OS は、不正あるいは異常なパケットに対する各 OS の応答の癖をまねることによって実現されている。各 OS の癖は、セキュリティスキャナ nmap[4] の OS fingerprinting 機能用データベースの情報を用いている。

2.3 実験環境

図 2 に示すように、大学のグローバル IP アドレスの実験用 LAN に Linux デスクトップ PC を設置し、honeyd を実行する。大学のインターネット接続ルータでは、TCP ポート、UDP ポート、ICMP について一部フィルタを設定しているので学外からは受けることができない通信がある。また、この honeyd ホストあるいは仮想ホストの IP アドレスは DNS に登録しないので、スキャンされなければ存在は知られない。さらに、パケットキャプチャ用ホスト (モニタリング用のホスト) を用意し、tcpdump を用いて、honeyd ホストの通信を全て記録する。監視を外部に知られないために、キャプチャ用ホストには IP アドレスをつけない。

この実験では 2 種類の記録をとる。

1. 仮想ホストが提供するサービスごとのすべての要求 (コマンド) と応答
2. 仮想ホスト (Windows200Server、WindowsNT4.0、Linux) のすべての通信

サービスごとの記録には、攻撃者が攻撃した要求 (コマンド) が記録されるので、攻撃者の攻撃方法やその目的を分析するときに用いる。後者は、仮想ホストが提供しないサービスに対するアクセスも含めた詳細な情報が必要になったときに参照する。

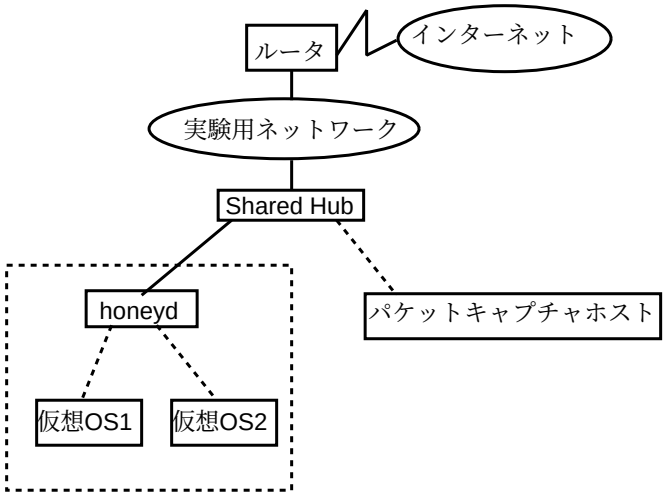


図 2: 実験環境

3 応用サービスの模倣

本節では、応用サービスへの攻撃を分類し、代表的な応用サービスを模倣する具体的な方法を述べる。

3.1 応用サービスへの攻撃の分類

実験システムで実現する偽の応用サービス例を検討するために、攻撃を通信形態とバナー情報利用の有無の 2 点で分類し図 3 に示す。

Web(HTTP) と TELNET は攻撃の目的別に複数の区分に示されている。区分 1 の Web(HTTP) は、1 回のアクセスで完了する攻撃、区分 3 では、複数回のアクセスで完了する攻撃を意味する。区分 2 の TELNET は別の攻撃のために、バナー情報から OS バージョンを調査する場合、区分 4 の TELNET はパスワード推測または他の方法でログインした後、コマンド操作等で攻撃者が目的とするプログラムを入れて起動する場合である。1 回のアクセスで完了する攻撃の典型的な例は、バッファオーバーフローによる管理権限の奪取である。Web(HTTP) の通信は基本的には一往復で完了するが、攻撃者は第一段階でコマンド実行要求を出し、それが成功した場合に次の段階のコマンド実行要求を出し、複数回の HTTP アクセスで攻撃が完了するという意味で区分 3 にも含めた。

区分 3 のおもなものは、Web サービス (ワームによる攻撃)、MSSQL(ワームによる攻撃) である [5]。ワームによる攻撃の場合、攻撃対象のバナー情報を見ることなく、無差別に攻撃をしてくることが多い。これらの応用サービスに対する攻撃方法の代表的な例を以下に示す [6][7]。

- Web サービスを利用するワームの例
  1. ランダムな IP アドレスに対して、ディレクトリ閲覧等のコマンド実行を含む HTTP 要求を試す。
  2. コマンド実行が可能と判明したら、ワームは、次以後の HTTP 要求でファイル転送コマンドを実行し、自分自身を攻撃対象ホストにコピーする。
  3. Web ページを閲覧しただけで、感染するように Web ページを書き換える。
- MSSQL サービスを利用するワームの例
  1. ネットワークを介して、パスワード設定が default 設定 (ID:sa、Passward:なしのもの) を探す。データベース管理者として login に成功した場合にはワームをいれる。
  2. 攻撃対象ホストのパスワードを変更する。
  3. 感染した PC のネットワーク設定を調べる。近隣の他のホストに感染を広げる。

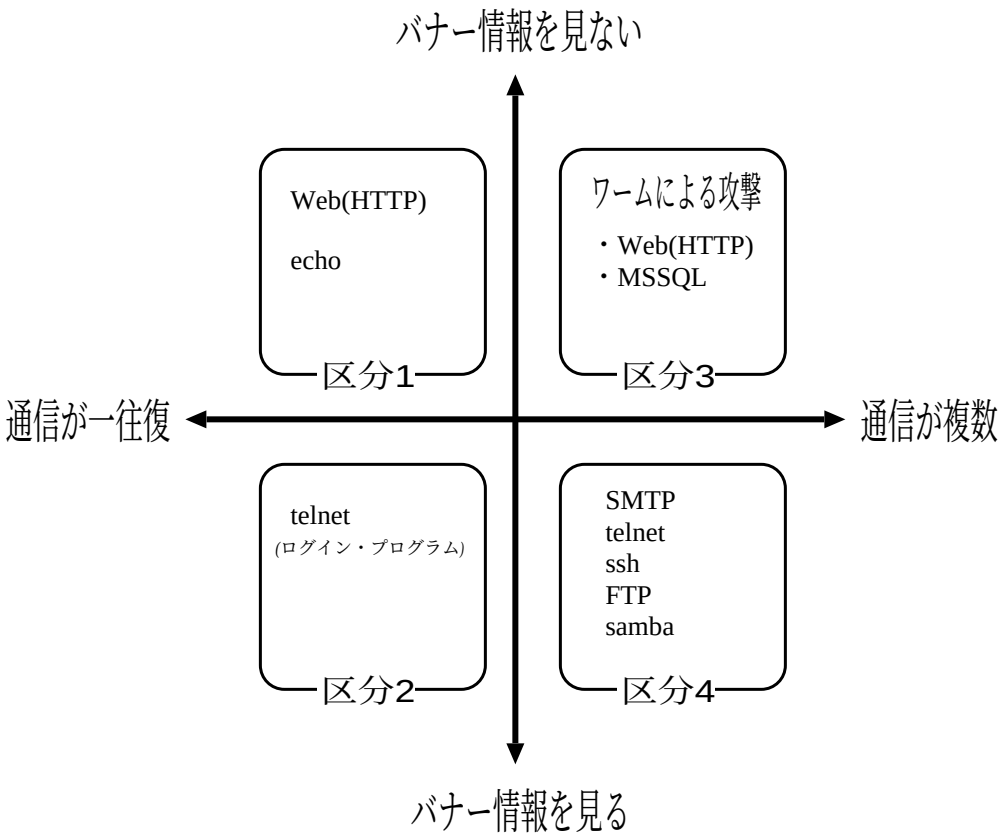


図 3: 応用サービスの分類

4. SQL サーバに接続してデータを盗む。

区分 4 の主なものは、SMTP, TELNET, SSH, FTP, SAMBA(ファイル共有サービス) である。攻撃者は、バナー情報から応用サービスのプログラムのバージョンを知り、バージョン固有の脆弱性を利用して、攻撃を続行する。バッファオーバーフローは、全ての応用サービスにおいて危険が大きい。機械語プログラムのデータを用いるので、脆弱性は CPU、OS、応用サービスプログラムのバージョンに依存するが、脆弱性が既知のものについては、手口が広範に周知されている可能性が高く、特に危険である。

以上の攻撃パターンをまとめて**図 4** に示す。

図 3 で示した区分 1 から 4 の応用サービスの各区分から 1 つずつ典型的なサービスを選び、それらの偽サービスプログラムを作成した。具体的には、区分 1 と 3 の Web サービス、区分 2 の TELNET サービス (login プログラム)、区分 4 の SMTP サービスとした。これら 3 つの偽サービス以外は実現しなかったが同様の手順で作成できる。

3.2 偽 Web サービスの実現

研究室の Unix Apache Web サーバへの不正アクセスを調べたところ、MS IIS に対する攻撃の前兆と見られるドライブ C のディレクトリ表示コマンドを含む HTTP 要求が多かった。研究室 Web サーバはこれに対し拒否応答を返すので、攻撃者は次の攻撃を試みない。したがって、次回以後の攻撃を含む HTTP 要求の情報が得られない。

本研究の目的は、セキュリティ対策に役立てるために、攻撃者からの一連の HTTP 要求を記録し、既知の攻撃と新種の攻撃を分析することなので、少なくとも初回の攻撃が成功したようにみせかけ、次回以後の攻撃が続くように応答に工夫する必要がある。

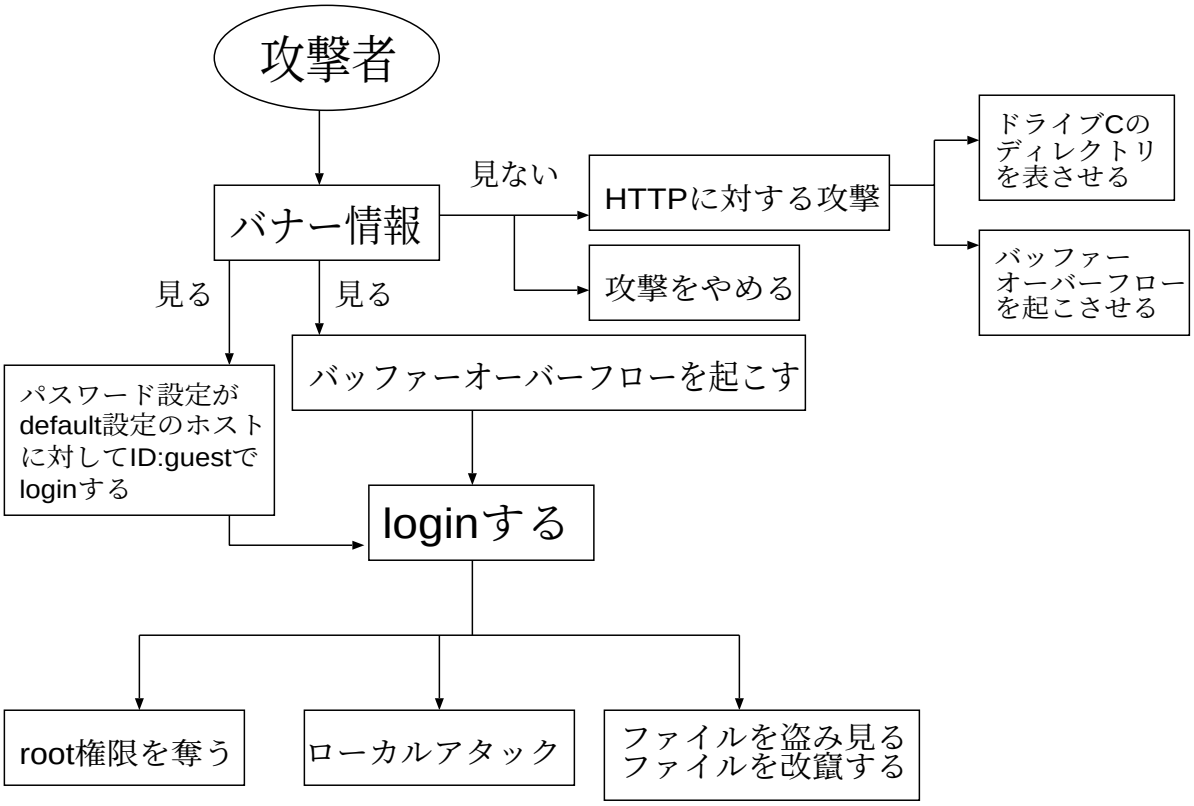


図 4: 攻撃者の攻撃パターン

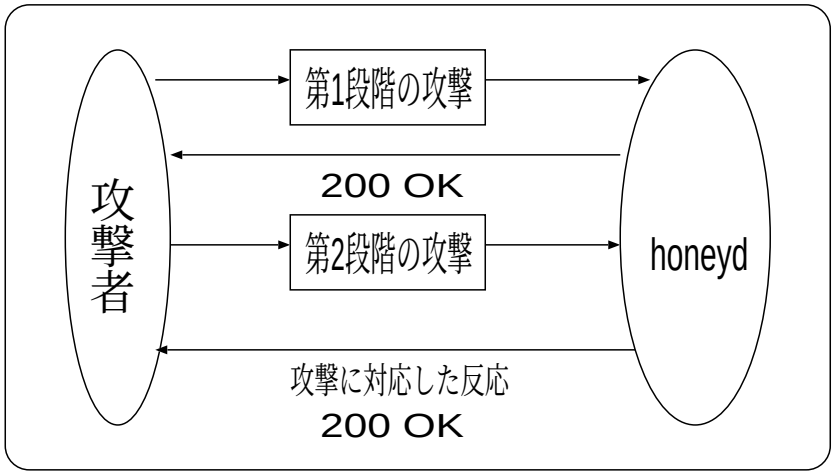


図 5: Web サーバの応答

実際に被害を被るにいたるまでの攻撃を許すように見せかけることが望ましいが、未知の攻撃に対し、応答に攻撃者が期待するデータまで含めることは困難である。そこで、少なくとも第 2 段階までの攻撃を誘うために、**図 5** に示すように、初回の攻撃を含む HTTP 要求に対し、成功を示す応答コード “200 OK” を返し、可能な場合はディレクトリ一覧などのデータも返す。2 回目以後もデータは返さないが “200 OK” を返すこととした。以下、実現の詳細について述べる。

3.2.1 攻撃と応答の具体例

Web サービスへの攻撃と作成したプログラムによる応答を NIMDA[8] を例に示す。

1. honeyd(WindowsNT4.0) の Web サービスに対して、dir コマンドが含まれる不正な HTTP 要求が送られる (第一段階)

   ```
   GET /sctipts/root.exe?/c+dir HTTP/1.0
   ```

2. これに対して、honeyd(WindowsNT4.0) は、応答コード “200 OK” と C のディレクトリ情報を HTTP 応答として返す。
3. 次に攻撃者は、以下の不正な HTTP 要求で、tftp コマンドを実行し、リモートホスト 133.XX.88.57 のファイル cool.dll(NIMDA) をローカルファイル httpodbc.dll にコピーする (第二段階)

   ```
   GET /scripts/root.exe?/c+tftp\%20-i\%20133.XX.88.57\%20（次行に続く）
                GET\%20cool.dll\%20httpodbc.dll HTTP/1.0
   ```

4. これに対し honeyd(WindowsNT4.0) は、応答コード “200 OK” を返す。

3.2.2 プログラム

Web サービスを模倣するためのプログラムは、シェルスクリプトで記述した。このプログラムは、honeyd の仮想ホスト (Windows2000Server、WindowsNT4.0、Linux) の 80 番ポートに通信がきた場合に honeyd から起動されるように設定した。

研究室のサーバのアクセス記録から攻撃の目的と攻撃者が求める応答を調査しておき、その結果に基づいて、以下の処理を行うプログラムを作成した。未知の不正な HTTP 要求を発見するたびに、手作業でプログラムを更新する。

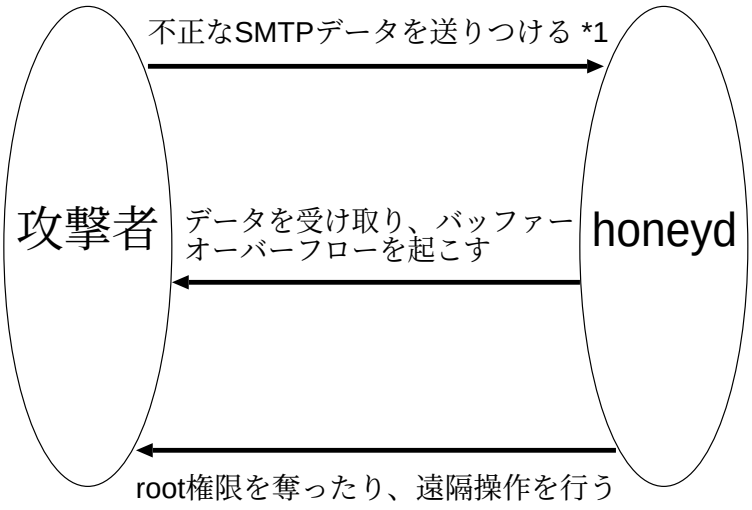
1. HTTP 要求 (コマンド) を記録する。
2. 既知の不正な HTTP 要求か判断する。
   - 既知の不正な要求であれば、
     (a) 攻撃が成功したように見せかけるために、攻撃者が求めるデータと応答コード “200 OK” を返す。
   - 未知の不正な要求であれば、
     (a) 攻撃に対する反応としては、“200 OK” の応答コードのみを返す。
     (b) 新しい不正な要求 (コマンド) について、攻撃目的と攻撃者が求める応答を推測する。
     (c) 新しい不正な要求 (コマンド) と応答を既知のリストに追加する。

3.3 偽 SMTP サービスの実現

メイルサーバへの攻撃のほとんどは、‘HELO‘、‘MAIL FROM‘、‘RCPT TO‘、‘EXPN‘などの SMTP コマンドでシェルコード (/bin/sh を起動する機械語プログラム) を含む長大なデータを送りつけてバッファオーバーフローを発生させることで、root 権限を奪うことを目的としている。図 6 に攻撃例を示す。

バッファオーバーフローを引き起こすためのデータは CPU、OS、SMTP サーバのバージョンに依存するので、SMTP サーバが出すバナー情報は攻撃者にヒントを与えることが多い。root 権限を奪うことに成功しなくても、サーバプロセスの停止に至ることが多い。バッファオーバーフローは SMTP コマンドの他に、電子

メイルメッセージ中の From:, To:, Cc:などのヘッダでも狙われることがあるが、本研究での偽サービスでは実際に電子メイルメッセージ (DATA コマンド) は受け取らないので、これらは含めない。



不正なSMTPデータ：
*1 : HELO XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....
: MAIL FROM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....
: RCPT TO XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....
: EXPN XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....

図 6: SMTP への攻撃と応答

3.3.1 プログラム

SMTP を模倣するために、以下の処理を行うプログラムをシェルスクリプトで作成した。攻撃の実験には、セキュリティスキャナ Nessus[9][10] を用い、アクセス記録から不正なコマンドと攻撃者が求める応答を調査した。更新は手作業で行った。

1. 仮想ホストの SMTP(25/tcp) に対する通信を記録する。
2. 既知の不正な SMTP コマンドか判断する。
   - 既知の不正なデータを含む SMTP コマンドであれば、攻撃が成功したように見せかけるために、SMTP 応答コード “250” と攻撃者が求めるデータを返す。
   - 未知の不正なデータを含む SMTP コマンドであれば、
     （a）SMTP 応答コード “250” だけを返す。
     （b）新しい不正な要求 (コマンド) について、攻撃目的と攻撃者が求める応答を推測する。
     （c）新しい不正なコマンドと応答を既知のリストに追加する。

3.4 偽 TELNET サービスの実現

TELNET サーバへの攻撃は、最も単純な推測したパスワードによるログインまたはバッファオーバーフロー等によって実質的にログインしたのちに、ファイルを改ざんする、ローカルアタックで root 権限を得る、任意のプログラムをいれて実行するなどの手口が多い。

攻撃者が、どのようにバッファオーバーフローを発生させるのか、ログイン後にとる行動 (コマンド操作) を記録するために、ログインに成功したように見せかける必要がある。

3.4.1 プログラム

TELNET サーバ (telnetd) には端末のタイプと制御方法など、ログイン手続き以前に複雑なプロトコル処理が含まれているので、すべてを模倣することは困難である。そこで、攻撃の目的となるログインのためのプログラムだけを模倣することにした。通常 telnetd からは/bin/login が起動されるが、そのかわりに偽 login プログラムを起動するように設定した。guest などの攻撃対象になりやすいアカウント名と簡単なパスワードを設定し、侵入を誘い、その挙動を記録することを期待する。

Perl スクリプトで以下の処理を行う login プログラムを作成した。未知の攻撃に対して攻撃者が求める応答を順次追加した。

1. 攻撃者が TELNET(23/tcp) に接続すると、ユーザ ID とパスワードを要求する。
2. 入力されたユーザ ID とパスワードを記録する。
3. ユーザ ID とパスワードをチェックし、あらかじめ設定した guest 等のパスワードと一致したら、侵入できたように見せかけるために、シェルプロンプト等を返す。
4. 相手の入力した文字を返し記録をとる。

telnet への攻撃の流れを**図 7** に示す。

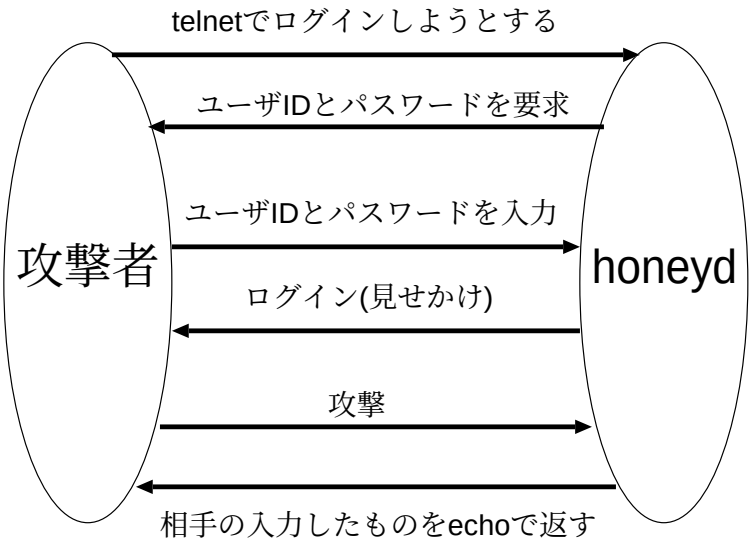


図 7: telnet への攻撃の流れ

4 アクセス記録の分析結果

2003 年 11 月 19 日午後 6 時に、学内の実験用のネットワークに honeyd を設置し、2004 年 1 月 8 日までのアクセス記録を分析した。

honeyd によって作成された仮想ホスト (WindowsNT4.0、Windows2000Server、Linux) はそれぞれ別の IP アドレスを持つ。当初各仮想ホストで受ける応用サービスは、3 節で述べた Web サービス、SMTP、自作ログインプログラムを起動するように設定した telnetd である。さらに TCP ポート 1433 への攻撃 (MSSQL の脆弱性をついた攻撃) が多いことを観測したので、12 月 10 日に MSSQL(1443/tcp) に対する簡単な偽サービスを追加した。2003 年 11 月 19 日午後 6 時以後のハニーポットへのアクセス記録を分析した結果を以下に述べる。ここで言う、アクセスとは、TCP の接続開始要求を意味し、接続が成立しないものも含めて 1 回と数える。ただし、偽サービスを用意している 23(TELNET)、25(SMTP)、80(HTTP)、1433(MSSQL、12 月

10 日以後) については接続が確立している。UDP については計数していない。攻撃の傾向、バックドアで使用されることが多いポートの情報は、他で提供されているセキュリティ関連情報 [11][12] で調査した。

4.1 一日あたりのアクセス数の変化

Honeypot は、DNS に登録していない。また、学外からの ICMP ECHO 要求は対外接続ルータで制限しているので、このホストの存在はポートスキャン以外の方法では知ることが困難である。そこで、攻撃者が honeypot の存在を知るまでの期間を推測するため、また年末年始に多数の攻撃があるという予想が正しいか確認するために、honeypot 設置以後の一日の総アクセス数の変化を調べた (図 8 参照)。

実験を開始から、徐々に honeypot へのアクセス数は増加している。ただし、11 月 21、25、30 日、12 月 1 日は honeypot を一時停止したのでアクセス数が極端に少ない。

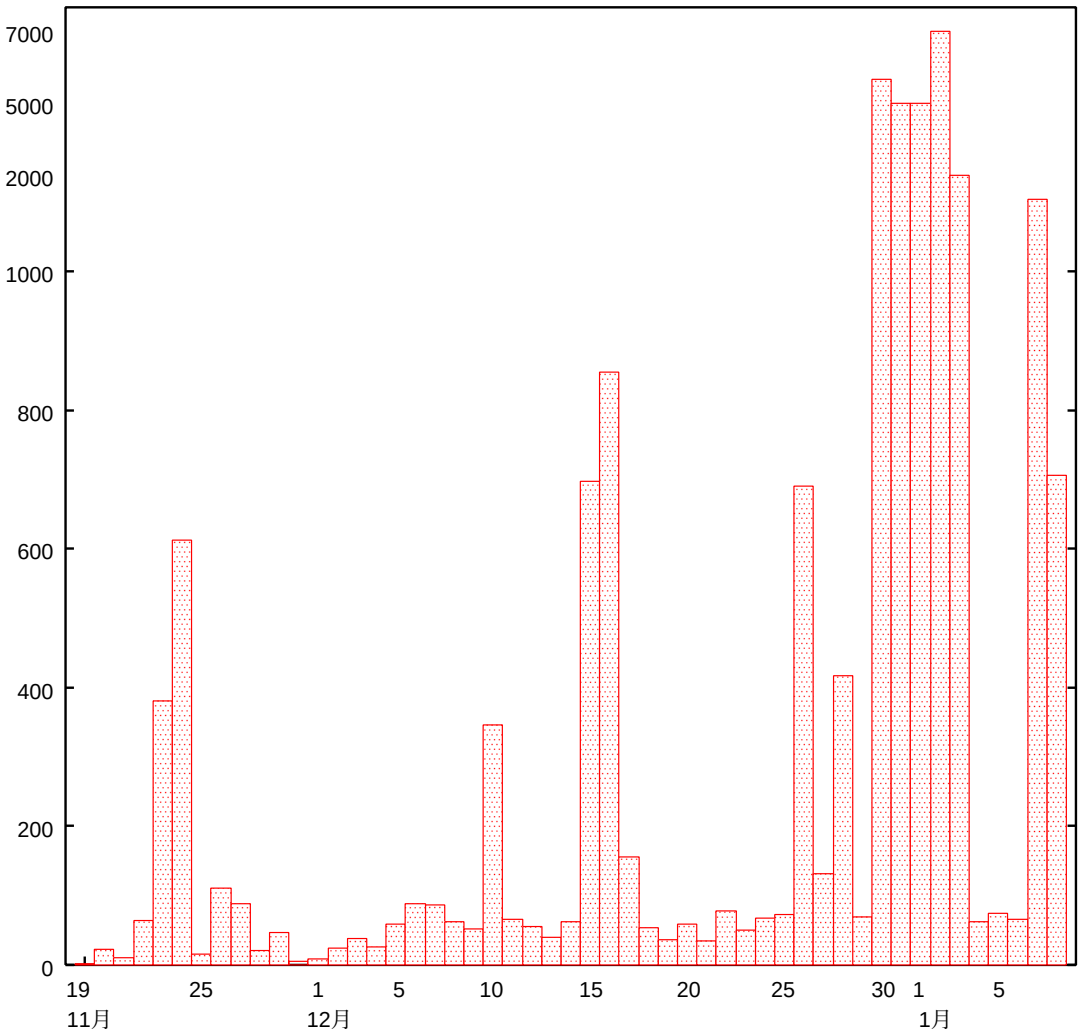


図 8: 日別アクセス数

設置後数日でアクセスが一日当たり 600 回程度にまで達したがその後減少し、12 月 15 から 16 日に 800 回程度まで上昇した。このことから、honeypot は、ネットワークに接続して数日以内に攻撃対象として認知されたといえる。12 月 25 日から 1 月 3 日までの年末年始には、予想通り、攻撃の試みと思われるアクセスが急増し、5,000 回を超える日も見られた。以下の特にアクセスが多かった日を除けば、一日あたりのアクセス数は 50 回から 150 回であった。

- 11 月 23 日 (日) と 24 日 (月) には、特にポート 10 に対してアクセスが集中した。
  ポート 10 に対応するサービスは IANA に未登録で、通常ホストはサービスを受け付けないので目的はわからない。
- 12 月 10 日 (水)・15 日 (月)・16 日 (火)・26 日 (金)・28 日 (日)・30 日 (火)・31 日 (水)、1 月 1 日 (木)・2 日 (金)・3 日 (土)・7 日 (水)・8 日 (木) はポート 1433(MSSQL) に対してアクセスが集中していた。

特に 12 月 10 日の 400 回のうちの 9 割は 1 ホストからのアクセスである。80 番ポートへポートスキャンをした後に、Windows 系の仮想ホストに対して、1433 番ポートに何度も攻撃を試みている。つまり、攻撃者は、honeypot に対してポートスキャンを行い、攻撃対象のポートが開いているかを確認して、攻撃を行ったと思われる。この行為をツールを使用した「人間」のアクセスか、「ワーム」による自動的なアクセスか区別することは難しい。しかし、DoS(Denial of Service) アタックを目的としない既知の「ワーム」による攻撃では、アクセスは多数のホストに分散することが多いので、400 回も連続したアクセスは「人間」の行為である可能性が高い。すなわち、攻撃者は、honeypot の存在を知り、集中的に honeypot へ攻撃してみたと推測できる。したがって、もし 12 月 10 日に攻撃してきた相手が「人間」であるならば、honeypot は、MSSQL(1433/tcp) の偽サービスを開始してすぐに攻撃対象として認知されたということになる。

4.2 時間帯別アクセス数

別総アクセス数の時間帯別分布を図 9 に示す。図の時刻は日本時間 (JST) である。

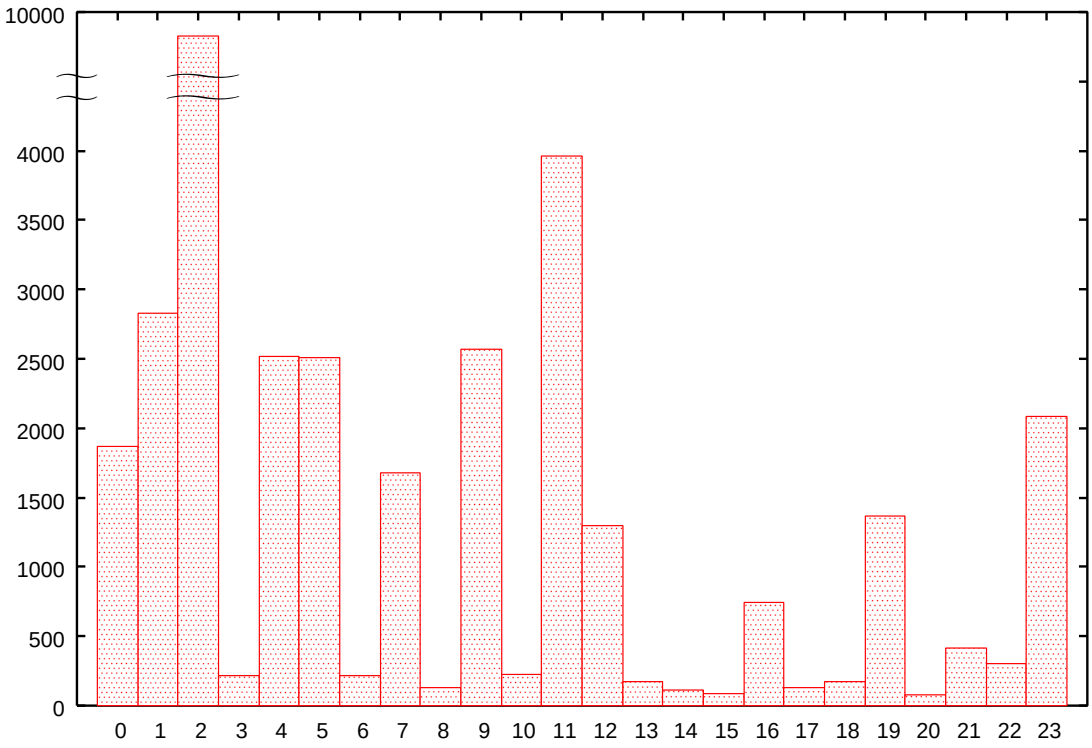


図 9: 時間帯別アクセス数

図 9 では次の特徴が見られる。

- 日本時間の深夜から昼にかけてアクセスが多い。
  深夜 2 時台が最も多く、合計 10,000 回程度のアクセスがある。2 時台には、複数のホストから攻撃を受けているが、一度に 2,700 回もアクセスしてきたホストが含まれる。この他のホストも一つのホストあたり 800 回前後のアクセスをしてきた。

- 昼から夜にかけてはアクセスが少ない。
  とくに 15 時と 20 時が少ない。

US からの攻撃が約 43%を占めるので、日本時間の午前中にアクセスが多い原因は、US で夜間に実行されるいたずら等であると思われる。

4.3 ポート別アクセス数

3 つの仮想ホスト (WindowsNT4.0、Windows2000Server、Linux) のポート別のアクセス数を比較する。

計測期間中の WindowsNT4.0、Linux、Windows2000Server の各仮想ホストへの総アクセス数は、それぞれ、10,870、10,046、12,917 で大差はなかった。

すべての仮想ホストにおいて多かったのは TCP ポート 1433, 10, 80 に対するアクセスで、Windows2000 Server 仮想ホストでだけ TCP ポート 16,409 に対するアクセスが多く記録された。他のポートに対するアクセスは 1%未満である。

1. TCP ポート 1433(MSSQL) – 41 から 47%(12 月 18 日以前)、91 から 93%(全期間)
2. TCP ポート 10(不明) – 30 から 33%(12 月 18 日以前)、2 から 3%(全期間)
3. TCP ポート 80(HTTP) – 12 から 14%(12 月 18 日以前)、1 から 2%(全期間)
4. (Windows2000 Server のみ)TCP ポート 16409(不明) – 10%(12 月 18 日以前)、3%(全期間)

一例として WindowsNT4.0 仮想ホストへのポート別アクセス数の割合を図 10 に示す。Linux 仮想ホストにおける割合はほぼ同じである。Windows 2000 Server 仮想ホストにおけるポート 16409 の分以外の割合もほぼ同じである。

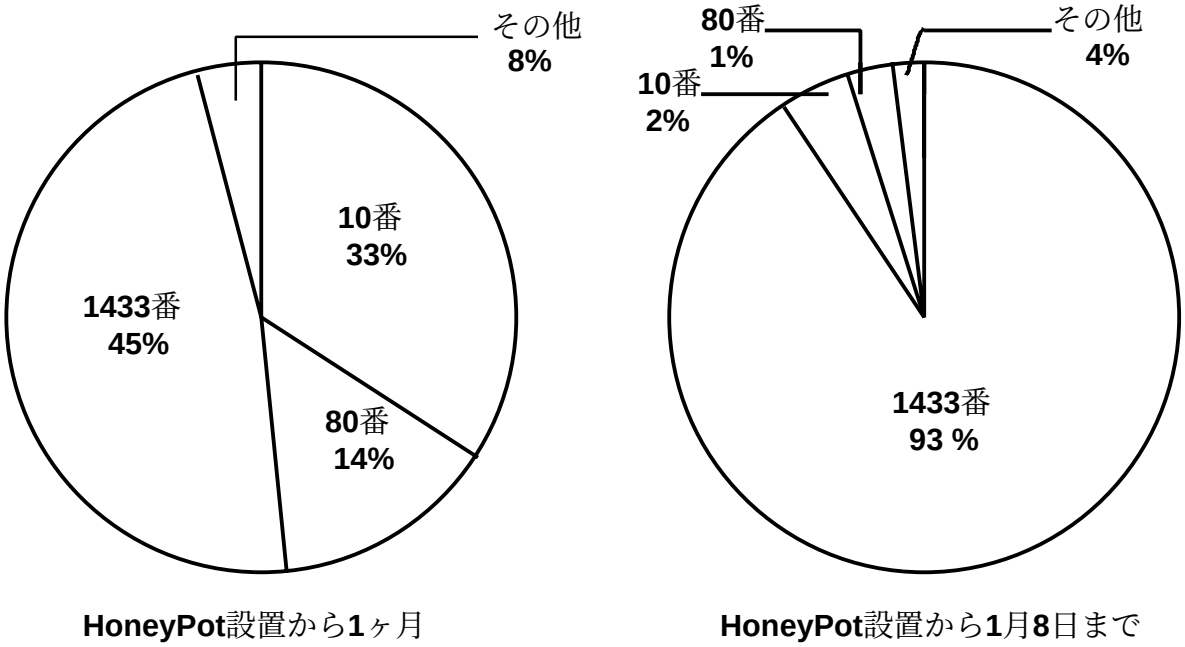


図 10: WindowsNT4.0 仮想ホストへのポート別アクセス数

以下にこれらのポートへのアクセスの傾向について述べる。

4.3.1 ポート 1433(MSSQL)

ポート 1433 は、Microsoft のデータベース・ソフトウェアである MSSQL で使用されるポートであり、データベース管理用アカウントへの侵入をねらった攻撃方法が知られている。第一段階の攻撃への応答として、送りつけられたコマンド文字列をそのまま返す偽応用サービスプログラムを用意して、第二段階の攻撃の記録取得を予定していたが、望む結果は得られなかった。Linux 仮想ホストにも Windows 仮想ホストとほぼ同数のアクセスがあったことから、OS 推定なしに、ワーム等で無差別に実行されている攻撃であると思われる。このポートへの攻撃は特にクリスマス以後急増し、すべての仮想ホストにおいて全期間の総アクセス数の 90%以上を占めた。

4.3.2 ポート 10(不明)

ポート 10 には、各仮想ホストにそれぞれ 269 回のアクセスがあった。このポートに対するアクセスは 11 月 23 日と 24 日に集中し以後はなかった。この未使用ポートは、既存のトロイの木馬が標準で使用するポートのリストにはないが、すでにしかけたバックドアの有無を確認し利用する試みで、一過性のものであると思われる。

4.3.3 ポート 80(HTTP)

ポート 80 ポートへのアクセスは、WindowsNT4.0, Linux, Windows2000 Server 仮想ホストに対しそれぞれ、181, 141, 178 回で、予想よりはるかに少なかった。Linux 仮想ホストでは、Apache をつかった実サービスを提供し、他の 2 つの仮想ホストでは、IIS に似せた偽 Web サービスを提供した。しかし、既知のワームによる攻撃がいくつか見られただけで、アクセスの大半はポート 80 への TCP 接続、すなわちポートスキャンにとどまり、新たな攻撃パターンは記録できなかった。

4.3.4 ポート 16409(不明)

このポートは、通常はポート 10 と同様に IANA に登録されていない未使用ポートである。ポート 16409 へのアクセスの目的は、ポート 10 と同様に、バックドア利用の試みと思われる。このアクセスが記録されたのは Windows2000 Server 仮想ホストだけなので、攻撃ツールには OS 推測の処理が組み込まれていると思われる。

4.3.5 他のポート

他のポートのアクセス傾向の例として Windows2000 Server 仮想ホストにおけるポート別アクセス回数を表 1 に示す。この表にないポートへのアクセスは 0 回である。表に含まれるが回数が 0 のポートに対するアクセスは、他の 2 つの仮想ホストでは 1 から 5 回記録されている。

偽応用サービスプログラムを用意した TCP ポート 23(TELNET) へのアクセスは、学外からは不可能なので記録できなかった。代わりにポート 22(SSH) へのアクセスは各仮想ホストで 5 回程度記録された。

4.4 国別アクセス数

地域や国の IP アドレス登録管理機関で提供されている whois データベースで、アクセスしたホストのソース IP アドレスが割り当てられている国を調べた。その結果得た国別アクセス数の割合を図 11 に示す。

表 1: Windows2000 Server 仮想ホストにおけるポート別アクセス数

ポート番号	アクセス数	ポート番号	アクセス数	ポート番号	アクセス数
10	269	22	4	25	4
80	178	280	0	443	11
554	10	901	6	1243	3
1257	6	1433	11713	1521	0
1526	1	1838	0	2277	1
3810	3	4000	38	4480	0
4898	1	4899	24	6112	1
6129	122	6192	1	7070	0
7100	2	9999	1	17300	26
27374	3	32771	1	34816	30
65439	0	406	1	1024	1
1182	1	1234	1	1490	1
3410	1	16409	447	32843	1
34817	3	40808	1		
ALL	12917				

Honeyd の仮想ホストに対してアクセスが多かった国は順に、US(アメリカ合衆国、43%)、NL(オランダ、29%)、AU(オーストラリア、21%)、UY(ウルグアイ、4%)、CA(カナダ、3%) である。インターネット大国である US からのアクセスが多いのは予想通りだが、NL からのアクセスが非常に多いことは特筆すべきである。他国からのアクセスもあったが、これら 3 つと比較すると無視できる程度であった。中国、韓国などアジアの他国からのアクセスも多いと予想していたが、ほとんどみられなかった。

アクセスのパターンは、二種類に分けられる。一つは、連続してアクセスをくり返すパターン (パターン 1)、他方は、間隔をあけてゆっくりアクセスするパターン (パターン 2) である。

US については、約 300 ホストからおもにパターン 1 のアクセスが記録された。一度に 100 回以上のアクセスを行ってきたホストは、7 つであった。一度に 1200 回のアクセスを行ってきたホストもあった。ポート 10 へのアクセスはすべて US からであった。

NL については、約 200 ホストからのパターン 1 とパターン 2 が混在するアクセスが記録された。1 ホストからの連続したアクセス回数は US より多く、一度に 100 回以上アクセスをしたホストは 21 もあった。一度で 2700 回もアクセスしたホストもあった。年末年始のアクセス数増加は、US、AU より極端でいたずらに熱心なインターネット利用者が多いと思われる。

AU からのアクセスはおもにパターン 2 であった。約 150 ホストからのアクセスが記録された。まれに、パターン 1 のアクセスがあったが、他国と比べると、パターン 1 のアクセスをしたホスト数が少ない。一度の攻撃で 100 回以上のアクセスをしてきたホストは 4 つで、一回の攻撃で 659 回のアクセスをしてきたホストがあった。年末年始のアクセスは US、NL と比較すると増加が少なかった。また、ポート 16409 へのアクセスは、全て AU からであった。

5 おわりに

本研究の目的は、仮想ホストで Honeypot を実験運用し、1) 偽の応用サービスプログラムを用いて、既知の攻撃や未知の攻撃の通信内容を記録すること、2) 攻撃数の増減から、脆弱性を持つホストが攻撃者に知られるまでの時間と年末年始などの季節的変動も分析することであった。

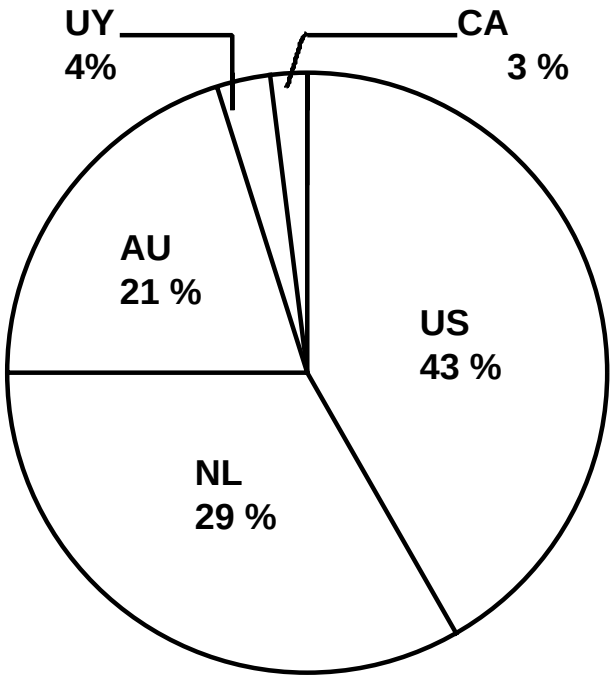


図 11: 国別アクセス数の割合

第 2 の目的については満足する結果が得られた。すなわち、仮想ホストに対する多くの攻撃の試みを観測することができた。実験で使用した HoneyHot は、DNS に登録していないので、アクセスが少ないと予想していたが、設置した初日からアクセスを観測できた。また、HoneyPot を設置するだけで短期間でも安全に多くの攻撃を観察できることが明らかになった。HoneyPot へのアクセスは全て攻撃と見なすことができるので、記録の分析が容易でいつ、どこから、どのポート (サービス) への攻撃が多いか、について興味深い結果が得られた。例えば、日本時間の午前 2 時頃に攻撃が多いこと、US、NL、AU の 3 ヶ国からの攻撃が多いこと、年末年始は予想通り攻撃が急増することがわかった。また、MSSQL に対する攻撃がいまだに多いことがわかった。

第 1 の目的については、攻撃を受けやすいサービスの脆弱性を模倣する偽サーバプログラムの作成方法は明らかになったが十分な結果が得られなかった。実験では、HTTP や SMTP サービスに対する攻撃を多数受けることを期待していたが、期待していた攻撃はほとんどなく、新しい攻撃の手口を記録することができなかった。

2003 年夏に猛威をふるった WindowsRPC の脆弱性への攻撃、netbios の脆弱性をついた攻撃、telnet を用いた不正ログインの試みについては、アクセスは 1 つも記録できなかった。大学の学外接続ルータで通信が制限されているので、これらについて、学外からのアクセスは不可能だが、学内ホストからは可能で、そのアクセスが記録されなかったことは学内ネットワークの管理上は望ましい。

受けることを期待していた攻撃の情報を収集できなかった問題は、Honeypot 自体のセキュリティを高め、攻撃にさらされやすい位置に設置し、必要あれば DNS に登録して、長期間運用することで解決できるだろう。

今後の課題には、まず、Honeypot の分析機能の向上がある。例えば、過去のアクセス記録をデータベースして、新たな攻撃と比較することで、新種の攻撃の発見や偽応用サービスプログラムの拡張が簡単になる。さらに、攻撃者の行動と意図を知るために侵入後の分析を行なうならば、本物の OS と応用サービスプログラムを使うほうが適していることは自明であるが、被害を被らないために安全策を施す必要があり、その安全を担保する手法の考案が必要である。

Honeypot の研究は、未開拓の分野であるが、攻撃に関わる情報収集の有用な手段であり、セキュリティ対策の重要な分野の一つとして十分研究するに値する題材である。

参考文献

[1] Niels Provos, “Honeyd-Network Rhapsody foy you,” http://www.citi.umich.edu/u/provos/honeyd/.

[2] (株) ディアイティ, “ManTrap,” http://www.dit.co.jp/symantec/mantrap.html.

[3] NetSec, “SPECTER Intrusion Detection System,” http://www.specter.com/.

[4] Insecure.Org, “Nmap Free Security Scanner, Tools & Hacking resoureces,” http://www.insecure.org/.

[5] セキュリティ アカデメイア, http://akademeia.info/.

[6] KAZU, **三分ハッキング**, 三才ブックス 2001.

[7] KAZU, **三分ハッキング２**, 三才ブックス 2002.

[8] 東京大学計算情報業務室, “セキュリティレベル high の弱点と対策方法,” http://www.ms.u-tokyo.ac.jp/security/shindan/2002-05-22/.

[9] Nessus, http://www.nessus.org/

[10] @police, http://www.cyberpolice.go.jp/

[11] Internet Security Systems, http://advice.isskk.co.jp/

[12] CERT, http://www.cert.org/