マルウエア検知

侵入検知技術の現状と将来

独立行政法人理化学研究所
渡辺　勝弘

侵入検知システムその後

昨年のおさらい

侵入検知システム

- Intrusion Detection System (IDS)
  - 侵入検知(IntrusionDetection)とは、コンピュータおよびネットワークに対するセキュリティ侵害の検出、通知、検出情報の管理に関する一連のプロセスを指し、侵入検知システムは、侵入検知を行うことを目的として設計されたシステムのことである

*ネットワーク侵入検知　武田圭史/磯貝宏著
ソフトバンクパブリッシング

侵入検知システム全般の問題点

- False PositiveとFalse Negative
  - 問題のないイベントを異常とみなしてしまう
  - 問題のあるイベントを見過ごしてしまう
- 侵入検知システム自身の機能、性能、精度にはあいかわらず問題があり、人手による分析作業等に依存しなければならず、コストの増大を招いてしまう

侵入検知システム全般の問題点

- 検知できる不正アクセス
  - ポートスキャン
  - Script Kiddies(ツール房)による幼稚な攻撃
  - 旧めのワーム、ウィルス
- でも私たちが知りたいのは
  - システムの弱点を突いたワンポイント攻撃
  - 未知の危険なワーム、ウィルス等に感染したPC

これら身近に迫っている真の危機

侵入検知システムの現状

- 昨年に引き続きいまだにシグネチャマッチングによる不正検知型のネットワーク侵入検知システムが主流
  - ネットワークを流れるパケットに含まれるデータパターンと、不正行為のデータパターン(シグネチャと呼ぶ)のマッチングを行うことで異常の検知を行う

シグネチャ型の問題点

- 未知の手法による不正アクセスには対処できない
- シグネチャはチューニングが必須
- 大量のアラートが発生してしまう
- 常にルール(シグネチャ)を更新しつづけなければならない
- ルールセットが肥大化してしまう

アノマリ型検知システムはどうか？

- プロトコルアノマリ
  - RFC等で規定されている正規の手順を踏まない通信を異常として検出する
- 行動ベース分析
  - システム、ユーザの振る舞いからプロファイルを作成し、通常でない振る舞いを異常として検知する
  - システム、ユーザの振る舞いに閾値を設定し、これを越える値を異常として検知する
  - 最近ではflowベースのアノマリ型検知システムが市場に現れてきている
  - SinkHoleのように動作して、LAN上のワーム等を検出する製品も現れている
    - 使われていないIPアドレス(DarkIPaddress)に対してスキャンや攻撃が行われたら、ワーム感染PCと見なして隔離する等

アノマリ型の現状

- アノマリ検知技術の一部は少しずつ実用化されだしている
- 侵入検知システムと少し方向性違っているかもしれない
  - WANでのDoS検知やLAN内でのマルウエア対策が目的
- これまでのシグネチャ型侵入検知に置き換わる物でなく、シグネチャ型と連携した侵入検知システムが現れるのではないか

侵入検知周辺の動向

- アプリケーションファイアーウォール
- IDPS（不正侵入防御システム）
- Target Based IDS
- UTM (統合脅威管理)
- SIM (統合セキュリティ管理ツール)
- エンドポイントセキュリティ
- トラフィック分析

アプリケーション ファイアウォール

- アプリケーションが正しい手順で通信を行っているか監視し、危険な通信を記録、無効化、遮断などする
- WEBアプリケーションファイアウォール(WAF)であれば、クライアントによるパラメータやCookieの改ざん、SQLインジェクションなどの不正なりクエストや、サーバーから送られるレスポンスに異常な値などがあった場合に通信を遮断する

不正侵入防御システム

- IDPS (Intrusion Detection & Prevention System)
  - 現時点での主流
  - 侵入検知システムに、自動フィルタリング機能が付いたもの
  - 既存の侵入検知システムがベースなので、その効果は推して知るべし
  - ワーム拡散防止などに活用している例はあるらしい
  - 何を遮断するかが鍵

TargetBasedIDS

- 監視対象のコンピュータのプロファイルを元に、アラートの重み付けを行うことで、検知精度を向上させる
  - たとえば、Linux+Apacheでサーバを運用しているなら、Nimdaによる悪性イベントが検出されても、必要以上の注意は無用であるから、アラートの危険度を下げる。逆にWindows+IISでサーバを運用しているなら、危険度を上げて管理者に注意を促す
  - いくつかの実装が存在し、市販されている
  - 実際にセキュリティ監視センターでは、計算機のペネトレーションテストの結果等を用いて検知精度を向上させている

MBSD 伊藤氏による解説 Snort-JP
http://www.snort.gr.jp/docs/N+I2005SnortBOF.pdf

統合脅威管理システム

- UTM (Unified Threat Management)
  - セキュリティ機能を単一のプラットフォーム上で提供するゲートウェイ型アプライアンス
    - ファイアーウォール、IDS/IDPS、ウィルス/スパム対策、コンテンツフィルタリング(HTTP、SMTP)等
  - 複数のセキュリティアプライアンスを一つの箱に実装することで省スペース、省コストを図る
  - 小規模IDS/IDPSはUTMに移行してしまった
  - ひとつにまとめただけのように見える
  - 検知精度が向上するわけではない
  - UTMにより不正アクセス管理が効率化するのかは、まだ良く分からない

SIM(統合セキュリティ管理ツール)

- SIM (Security Information Manager)
  - UTMとは逆のアプローチ
  - セキュリティアプライアンス等、さまざまな機器のアラート、ログ等をまとめて分析することで、不正行為、異常状態などを検知しようとする
  - 最近はSEM(SecurityEventManager)なる言葉も登場している
  - ふたぎさんのセッションで紹介

エンドポイントセキュリティ

- PFW(パーソナルファイアウォール)、アンチウイルスなど、クライアントコンピュータ上で驚異の検知と防御を行う
- 内部統制関係で、クライアントコンピュータの監視、管理ソリューションが普及しだした
- クライアントコンピュータにおける検知と防御は更に進化するだろう
- SIM等を用いて、他のセキュリティアプライアンス、ネットワーク機器等との連携することが求められる

トラフィック分析

- さまざまな通信の振る舞いを監視することで不正行為や異常を検知する
- 今年はトラフィック分析の技術を実装した製品が現れた
  - アノマリ型検知装置に近い形ではある
- セキュリティ監視よりはネットワーク監視の意味合いが強い
  - DoSやMassMail、ワームの繁殖等の検知はできるだろう
- 異常発生の参考データ程度にしかならない
- 他の監視手法と組み合わせる必要がある

侵入検知システムの今後

- シグネチャ型侵入検知システムは頭打ちになる
- アノマリ技術、トラフィック分析技術がさらに進化するだろう
- UTM、SIMにみられる統合化がいっそう進むだろう
- 侵入検知システムは特定通信の検知等、専用化が進むかもしれない
- ハードウェア化がいっそう進むだろう

マルウエア

昨年のおさらい

マルウエア (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マルウエア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

マルウエアとは

- マルウエアのふるまい
  - 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

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

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
- バックドアなどを発見しづらくするために用いられる
- 例
  - 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

『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

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)

- エンドノードに潜り込み、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)

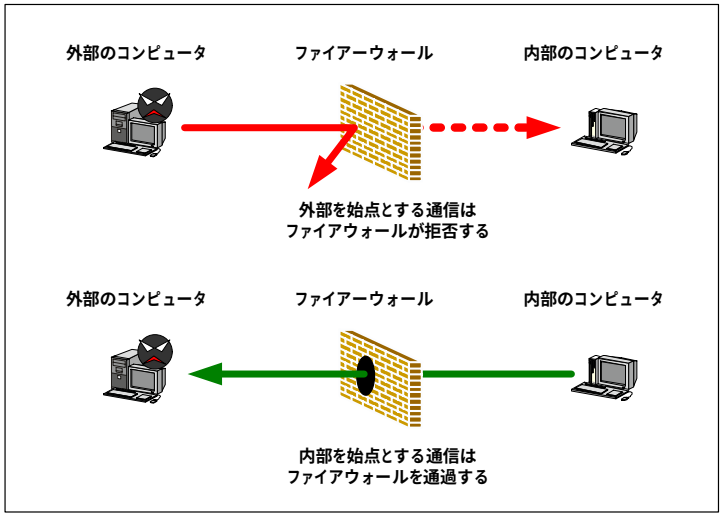
元はIRCの自動運転できるクライアントソフトで、発言に対して自動的に返答したり、ちょっとしたコマンドが実行できたりする

ファイアウォールは有効か

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

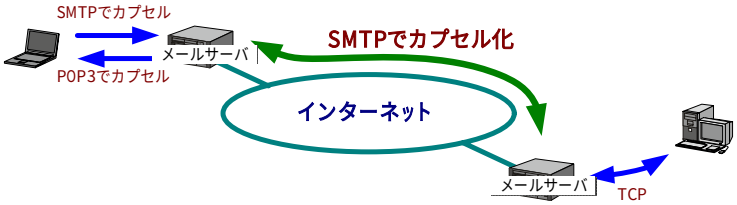
HTTPを用いたトンネル

- 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



メールが届く所ならトンネルを掘れる

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

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

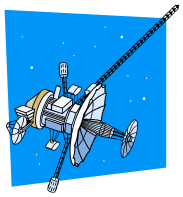
スパイウェアの例

- 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

ファイアウォールの有効性

- ファイアウォールを運用しているからって、安心しきれない
- ファイアウォールを迂回する技術とその実装が存在する
- プロクシ等を使って通信を厳しく制限していたとしても、なんらかの逃げ道が存在するはず



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マルウエアの検知

- 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、マルウエア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

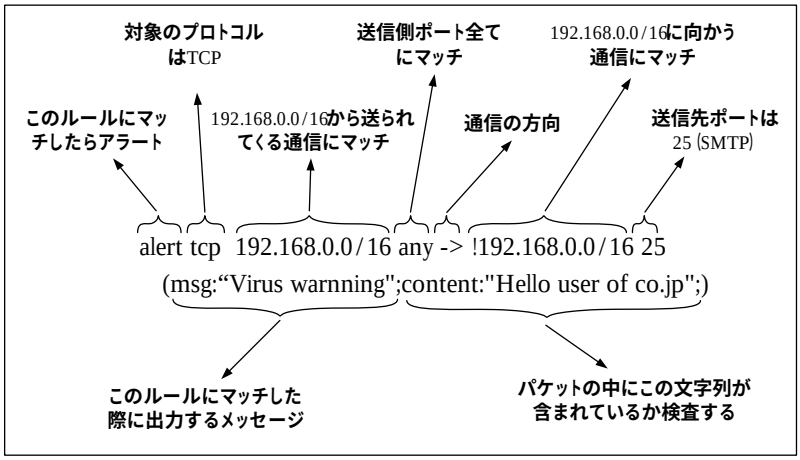
オープンソースの侵入検知システムSnortを用いて、いくつかの例を紹介しましょう

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-2.6.1が最新
  - 開発元のSourceFireはCheckPointに買収されなかった

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

シグネチャの構造



シグネチャの例

Bagle (ウィルスメール) を検知するルール

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any  (
msg:“Suspicious virus warnning v2 !!!”;
content:“Attached file";
content:"password";)

alert tcp $HOME_NET any -> $EXTERNAL_NET 25  (
msg:"Suspicious virus warnning v1 !!!";
content:"Hello user of Go.jp";)
```

BleedingEdgeルールセット

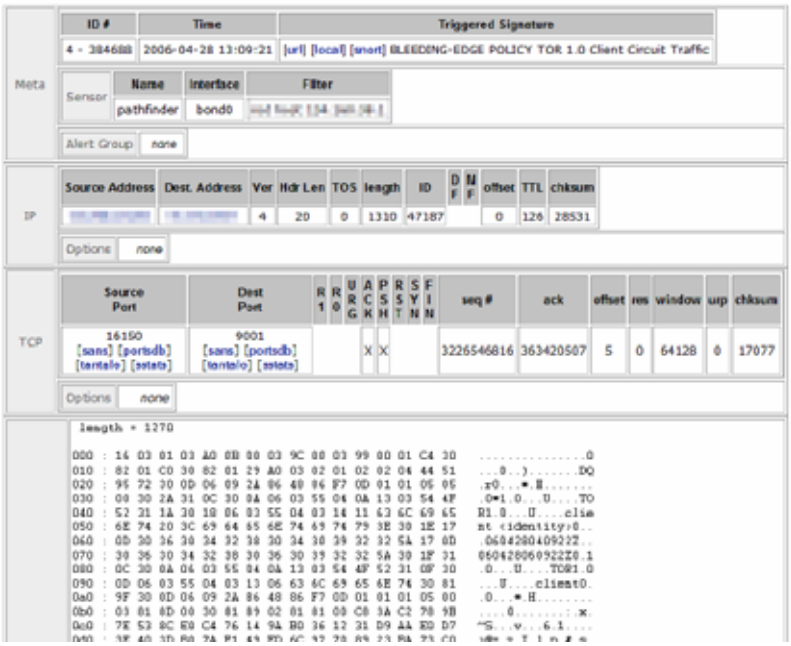
```
# IRC Trojan Reporting
#
# By Erik Fichtner
#
# Bleeding-Remix :: irc / ircbot detection state machine
# compiled from various sources.
# thanks to: Joe Stewart of LURHO, Joel Esler, Tomfi.

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;)
```

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

Covert Channelの検知

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

alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"BLEEDING-EDGE POLICY TOR 1.0 Client Circuit Traffic"; flow:established,to_server; content:"|54 4f 52|";content:"|63 6c 69 65 6e 74 20 3C 69 64 65 6E 74 69 74 79 3E|";distance:10; within:20; threshold:type both, track by_src, count 1, seconds 60; classtype:policy-violation; reference:url,tor.eff.org; sid:2001728; rev:3;)



ID #	Time	Triggered Signature
4 - 3846BB	2006-04-28 13:09:21	[url] [local] [snort] BLEEDING-EDGE POLICY TOR 1.0 Client Circuit Traffic

Sensor	Name	Interface	Filter
	pathfinder	bond0	

Alert Group: none

Source Address	Dest. Address	Ver	Hdr Len	TOS	length	ID	DF	MF	offset	TTL	chksum
		4	20	0	1310	47187			0	126	28531

Options: none

Source Port	Dest Port	R1	R0	URG	ACK	PSH	RST	SYN	FIN	seq #	ack	offset	res	window	urp	chksum
16150 [sans] [portsdb] [tantalo] [solats]	9001 [sans] [portsdb] [tantalo] [solats]				X	X				3226546816	363420507	5	0	64128	0	17077

Options: none

length = 1270

```
000 : 16 03 01 03 A0 0B 00 03 9C 00 03 99 00 01 C4 30   ...............0
010 : 82 01 C0 30 82 01 29 A0 03 02 01 02 02 04 44 51   ...0..).......DQ
020 : 95 72 30 0D 06 09 2A 86 48 86 F7 0D 01 01 05 05   .r0...*.H.......
030 : 00 30 2A 31 0C 30 0A 06 03 55 04 0A 13 03 54 4F   .0*1.0...U....TO
040 : 52 31 1A 30 18 06 03 55 04 03 14 11 63 6C 69 65   R1.0...U....clie
050 : 6E 74 20 3C 69 64 65 6E 74 69 74 79 3E 30 1E 17   nt <identity>0..
060 : 0D 30 36 30 34 32 38 30 34 30 39 32 32 5A 17 0D   .060428040922Z..
070 : 30 36 30 34 32 38 30 36 30 39 32 32 5A 30 1F 31   060428060922Z0.1
080 : 0C 30 0A 06 03 55 04 0A 13 03 54 4F 52 31 0F 30   .0...U....TOR1.0
090 : 0D 06 03 55 04 03 13 06 63 6C 69 65 6E 74 30 81   ...U....client0.
0a0 : 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01 05 00   .0...*.H........
0b0 : 03 81 8D 00 30 81 89 02 01 81 00 C8 3A C2 78 9B   ....0.......:.x.
0c0 : 7E 53 8C E0 C4 76 14 9A B0 36 12 31 D9 AA ED D7   ~S...v...6.1....
```

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.~..>...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.

SpyWareの検知

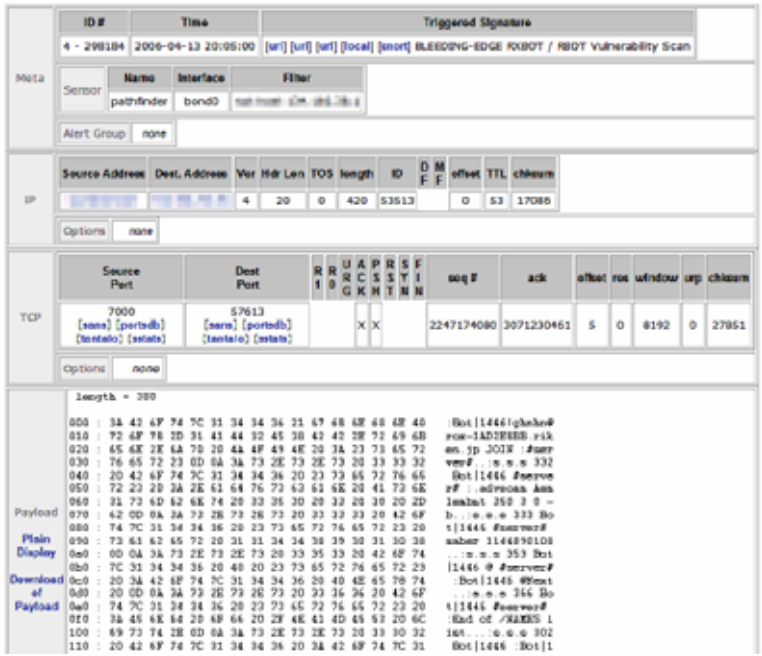
- Comet Cousorを検知するルール

```
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; )
```

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;)
```



ボットネットの検知

- 多くのボットは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
- 未知のプロトコルを使い外部と通信するボ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ているボットの検知は難しい
- 潜伏して外部と通信しないボットは検知が難しい
- もちろん未知のボットを検知することは難しい

侵入検知システムの有効性

- シグネチャベースの場合、シグネチャが用意されていないと、手も足もでない
- 潜伏しているマルウエアは検知できない
- 暗号化通信は苦手

- プロトコルアノマリで検知できるか疑問
- 行動ベース分析なら検知するかもしれない

進化するマルウエア

マルウエア対策を考えてみる

- SSLで暗号化通信を行ったり、P2Pネットワーク型のボットネットも出ています
- 派手に動き回らない潜伏型のマルウエアはやっかいです

- ファイアウォール等のインフラ側でセキュリティ対策できない
- IDS/IDPS等での検知が技術的に難しくなっている

マルウエア対策を考えてみる

- マルウエアの出現サイクルは短くなるいっぽう
- 特定組織を狙うウイルス等のスピア型攻撃も実際に発生しています

- エンドポイントのセキュリティ対策が効かない
  - アンチウイルスの対応が遅れる、またはできない
  - パーソナルファイアウォール等を回避する技術の実装
  - そもそも攻撃者はセキュリティレベルの低いコンピュータを狙う

マルウエア対策を考えてみる

- マルウエアを正確に検知し、防御できる技術は存在しない

なんらかの方法を考えないと