Network Telescopes

David Moore

October 29th, 2003 - USENIX LISA

dmoore@caida.org

www.caida.org

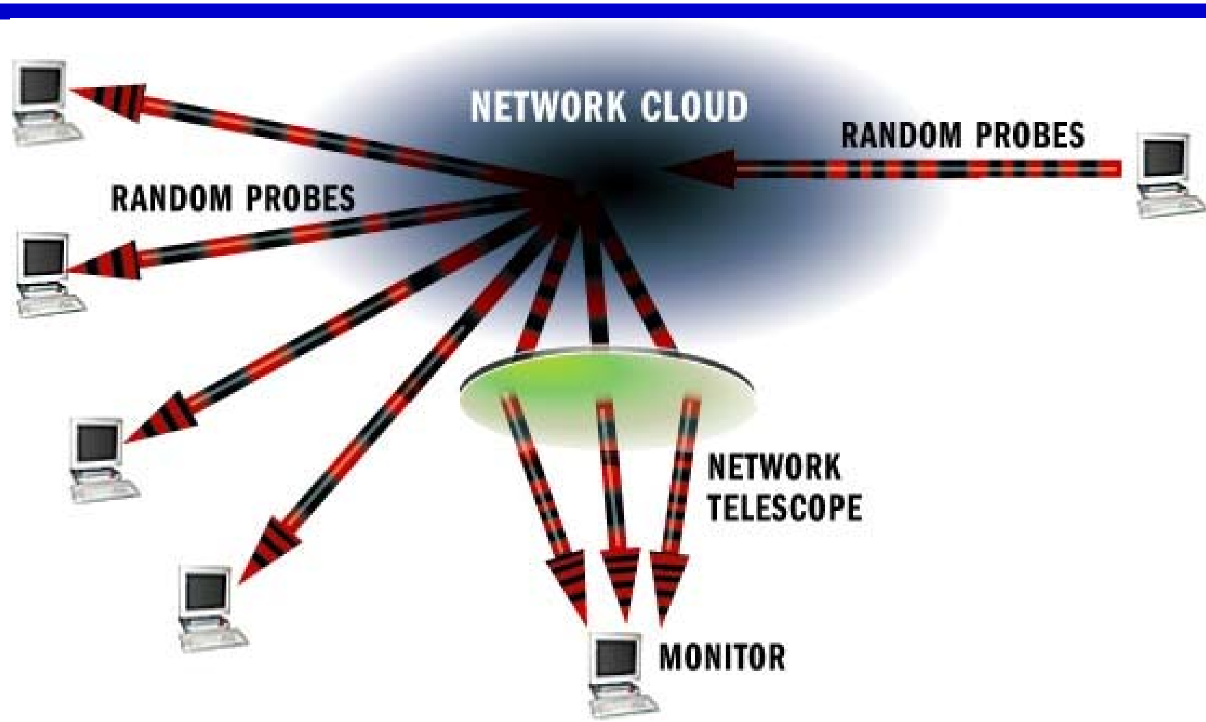
UCSD CSE

caida

What is a "Network Telescope"?

- A way of seeing remote security events, without being there.
- Can see:
  - victims of certain kinds of denial-of-service attacks
  - hosts infected by random-spread worms
  - port and host scanning
  - misconfiguration

Network Telescope: Basic Idea



If a computer sends packets to IP addresses *randomly*, we should see some of the packets if we monitor enough address space.

Network Telescope

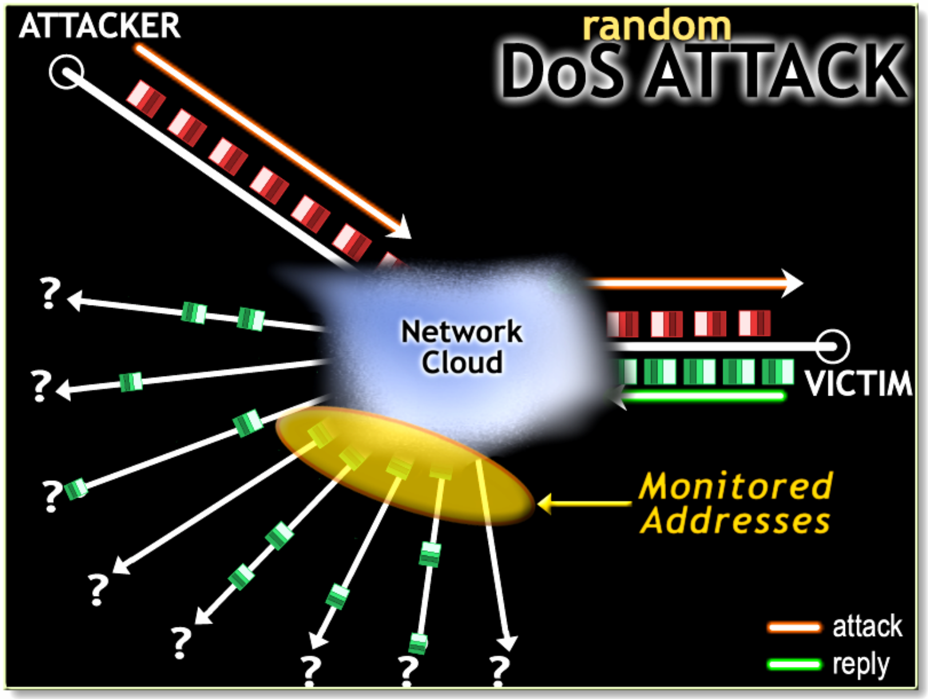
- Chunk of (globally) routed IP address space
- Little or no legitimate traffic (or easily filtered)
  - might be "holes" in a real production network
- Unexpected traffic arriving at the network telescope can imply remote network/security events
- Generally good for seeing explosions, not small events
- Depends on statistics/randomness working

Outline

- What is a network telescope?
- **Denial-of-Service Attacks**
- Internet Worms
- How to use your own telescope

Network Telescope: Denial-of-Service Attacks

- Attacker floods the victim with requests using random spoofed source IP addresses
- Victim believes requests are legitimate and responds to each spoofed address
- With a /8 ("class A"), one can observe 1/256th of all *victim responses* to spoofed addresses



Backscatter Analysis Technique

- Flooding-style DoS attacks
  - e.g. SYN flood, ICMP flood
- Attackers spoof source address **randomly**
  - True of many major attack tools
  - i.e. not SMURF or reflector attack
- Victims, in turn, respond to attack packets
- Unsolicited responses (*backscatter*) equally distributed across IP space
- Received backscatter is **evidence** of an attacker elsewhere

Backscatter Analysis

- Monitor block of *n* IP addresses
- Expected number of backscatter packets given an attack of *m* packets:

$$E(X) = \frac{nm}{2^{32}}$$

- Extrapolated attack rate R is a function of measured backscatter rate R':

$$R \geq R' \frac{2^{32}}{n}$$

Assumptions and Biases

- *Address uniformity*
  - Ingress filtering, reflectors, etc. cause us to **underestimate** number of attacks
  - Can bias rate estimation (can we test uniformity?)
- *Reliable delivery*
  - Packet losses, server overload & rate limiting cause us to **underestimate** attack rates/durations
- *Backscatter hypothesis*
  - Can be biased by purposeful unsolicited packets
    - Port scanning (minor factor at worst in practice)
  - Can we verify backscatter at multiple sites?

Identifying DoS Attacks

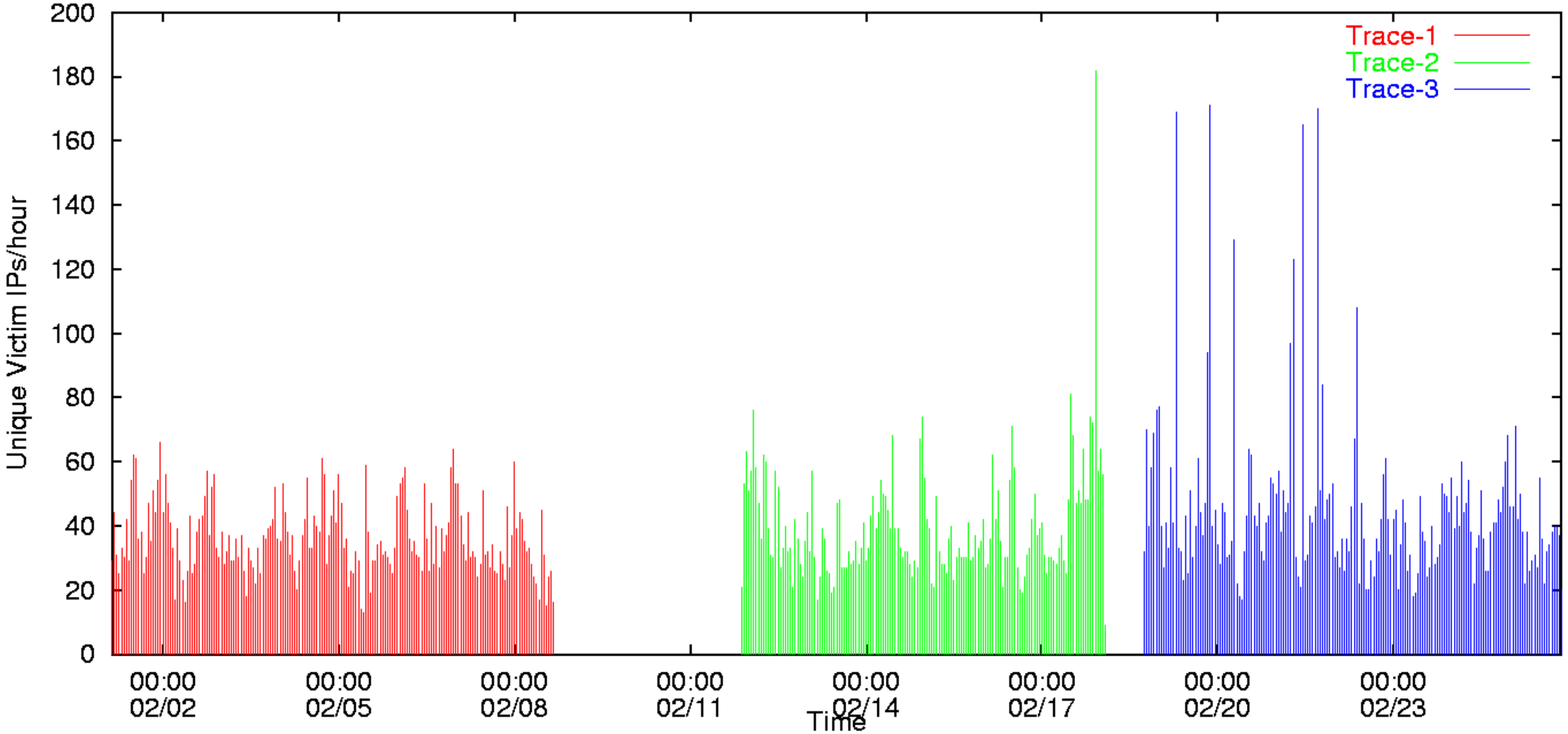
- Flow-based analysis (categorical)
  - Keyed on victim IP address and protocol
  - Flow duration defined by explicit parameters (min. threshold, timeout)

- Event-based analysis (intensity)
  - Attack event: backscatter packets from IP address in 1–minute window
  - No notion of attack duration or "kind"

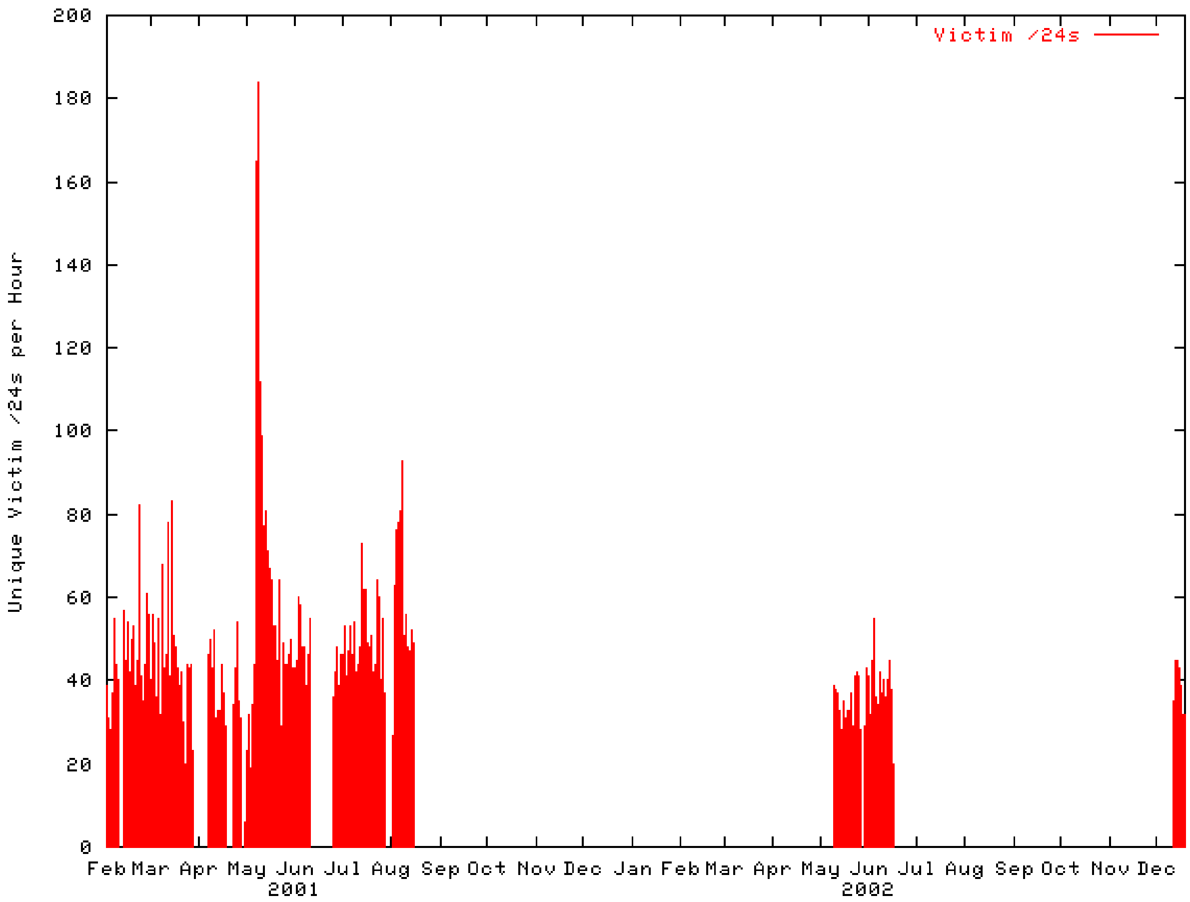
DoS Attack breakdown (three weeks in February 2001)

	Week1	Week2	Week3
Attacks	4173	3878	4754
Victim IPs	1942	1821	2385
Victim prefixes	1132	1085	1281
Victim ASes	585	575	677
Victim DNS domains	750	693	876
Victim DNS TLDs	60	62	71

DoS Attacks over time



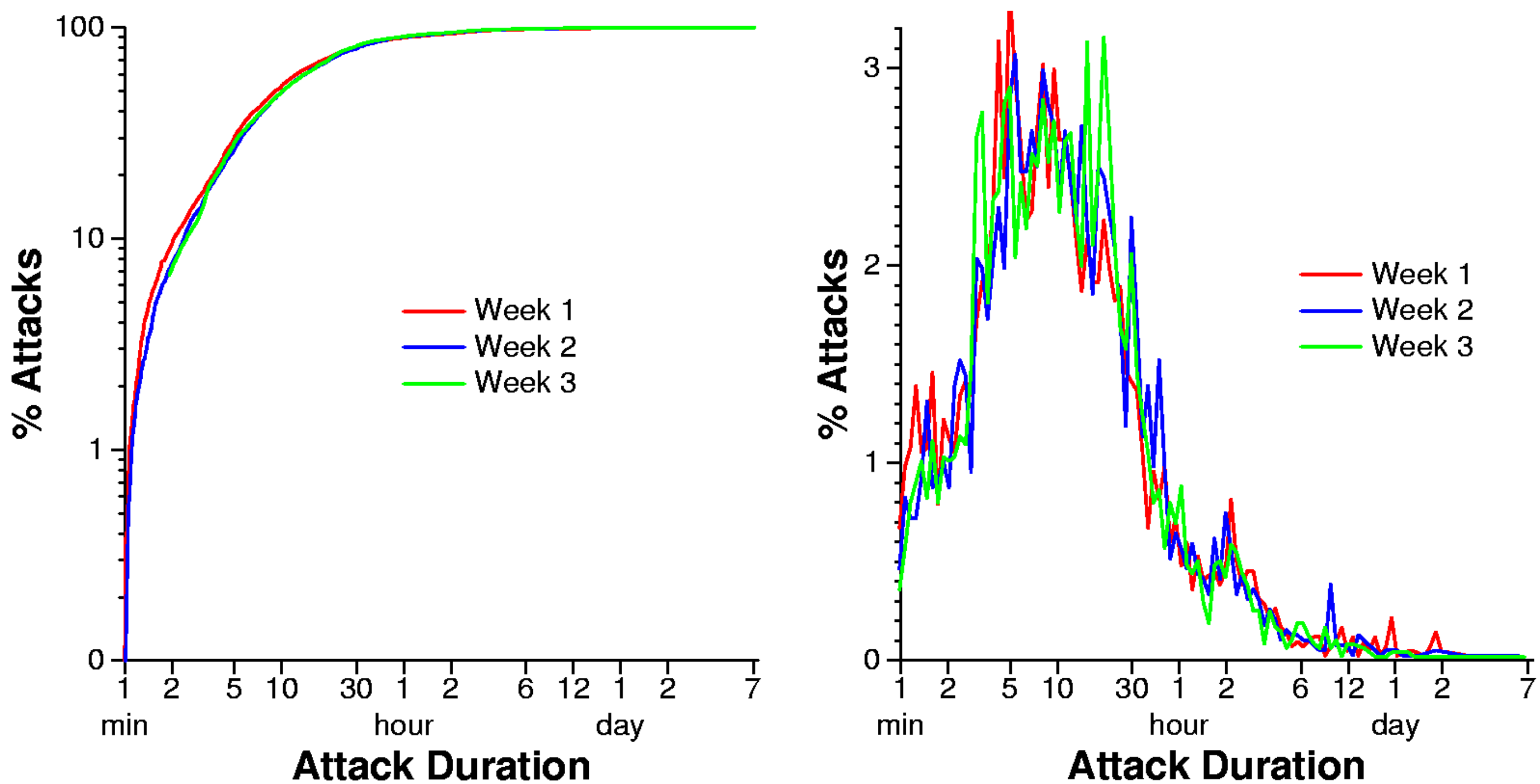
DoS Attacks over time



DoS Attack characterization

- Protocols
  - Mostly TCP (90-94% attacks), but a few large ICMP floods (up to 43% of packets)
  - Some evidence of ISP “blackholing” (ICMP host unreachable)
- Services
  - Most attacks on multiple ports (~80%)
  - A few services (HTTP, IRC) singled out

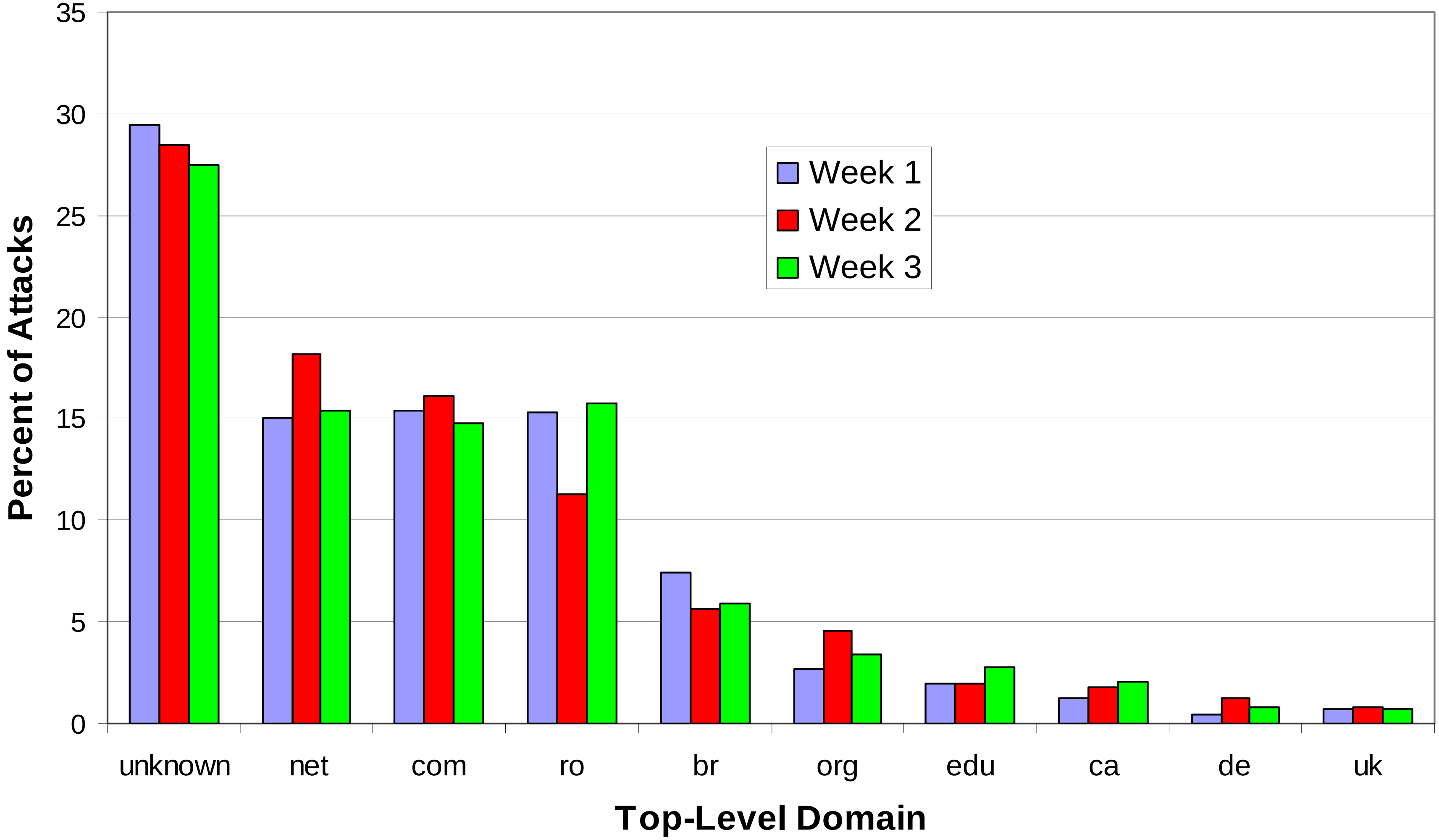
DoS Attack duration distribution



DoS Victim characterization

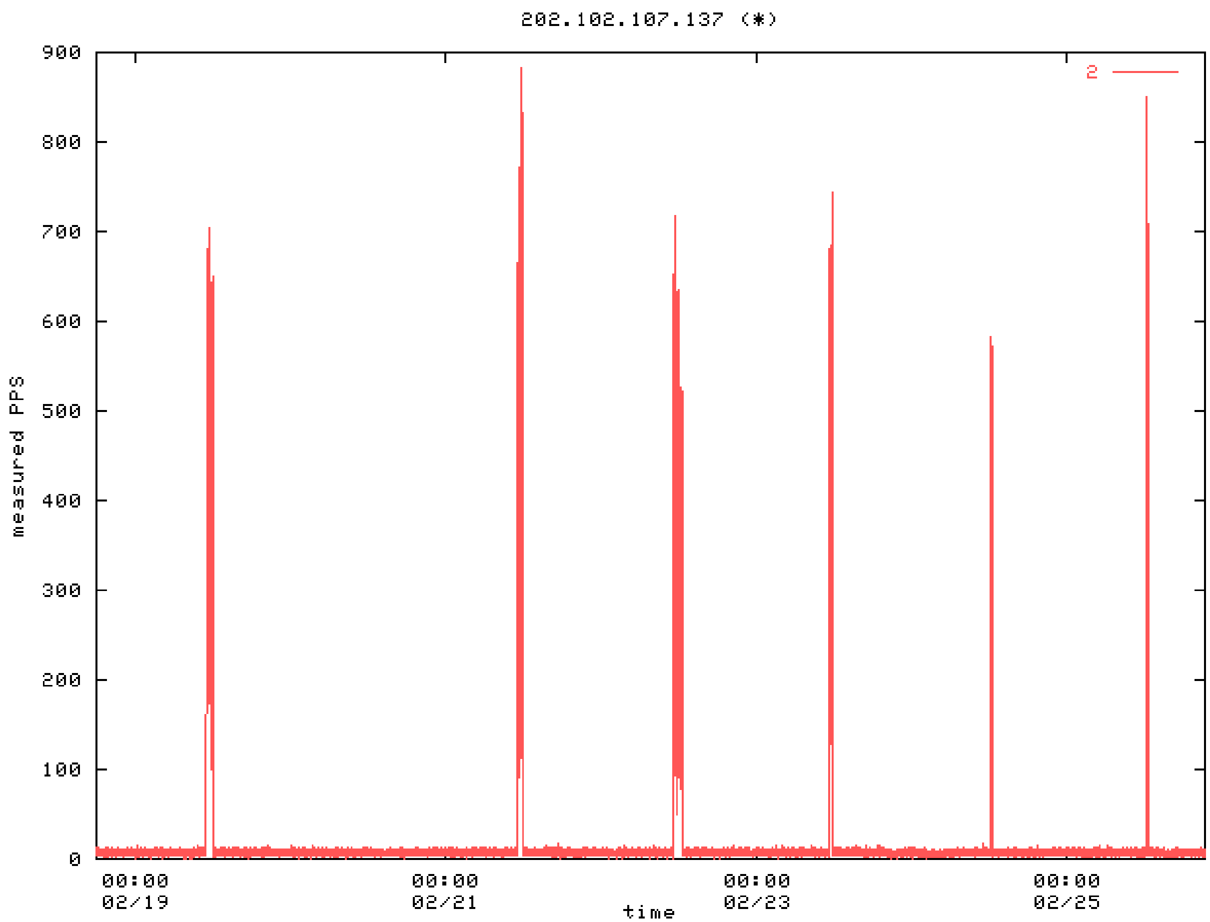
- Entire spectrum of commercial businesses
  - Yahoo, CNN, Amazon, etc and many smaller biz
- Evidence that minor DoS attacks used for personal vendettas
  - 10-20% of attacks to home machines
  - A few very large attacks against broadband
- 5% of attacks target infrastructure
  - Routers (e.g. core2-core1-oc48.paol.above.net)
  - Name servers (e.g. ns4.reliablehosting.com)

DoS Victim breakdown by TLD

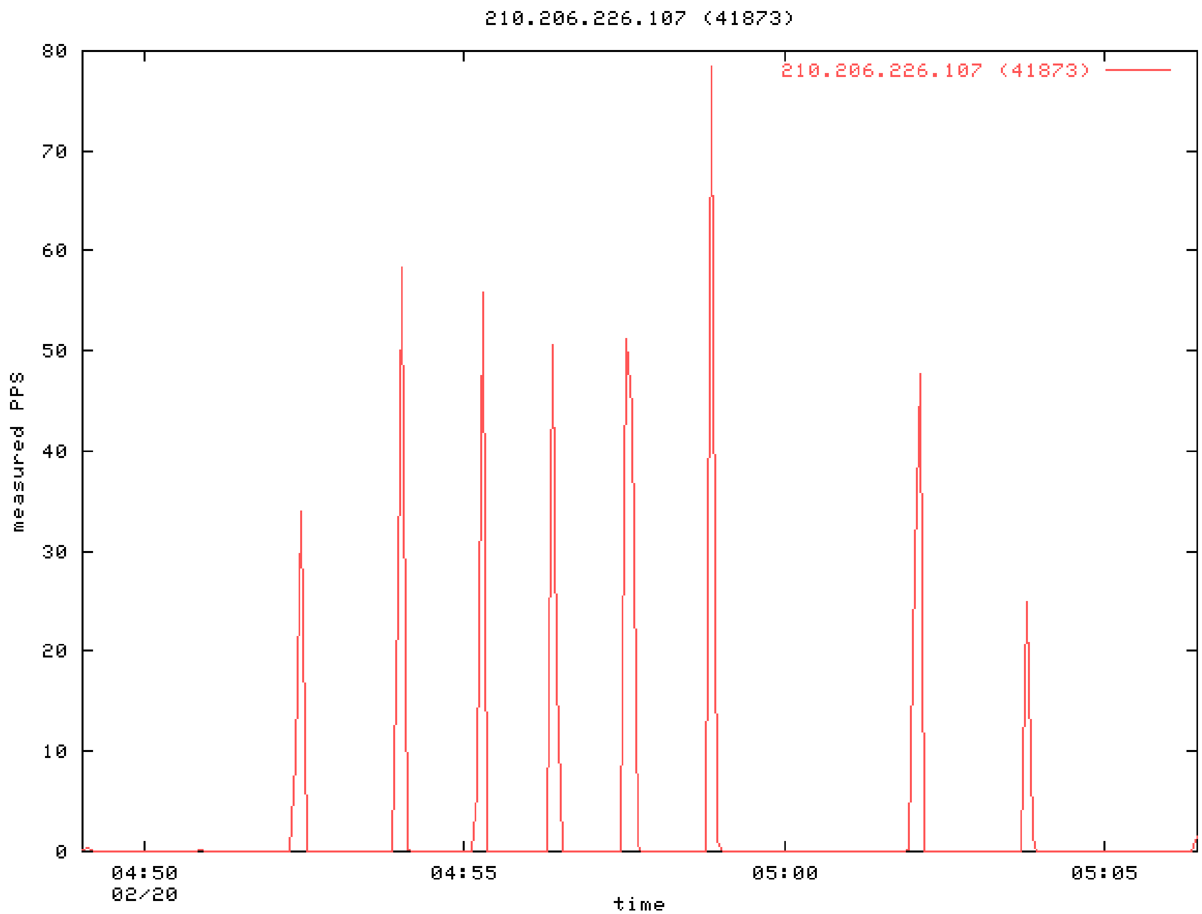


UC

Example 1:
Periodic attack (1hr per 24hrs)



Example 2:
Punctuated attack (1min interval)



Validation

- Backscatter not explained by port scanning
  - 98% of backscatter packets do not cause response
  - This may be changing
- Repeated experiment with independent monitor (3 /16's from Vern Paxson)
  - Only captured TCP SYN/ACK backscatter
  - 98% inclusion into larger dataset
- Matched to actual attacks detected by Asta Networks on large backbone network

Backscatter Conclusions

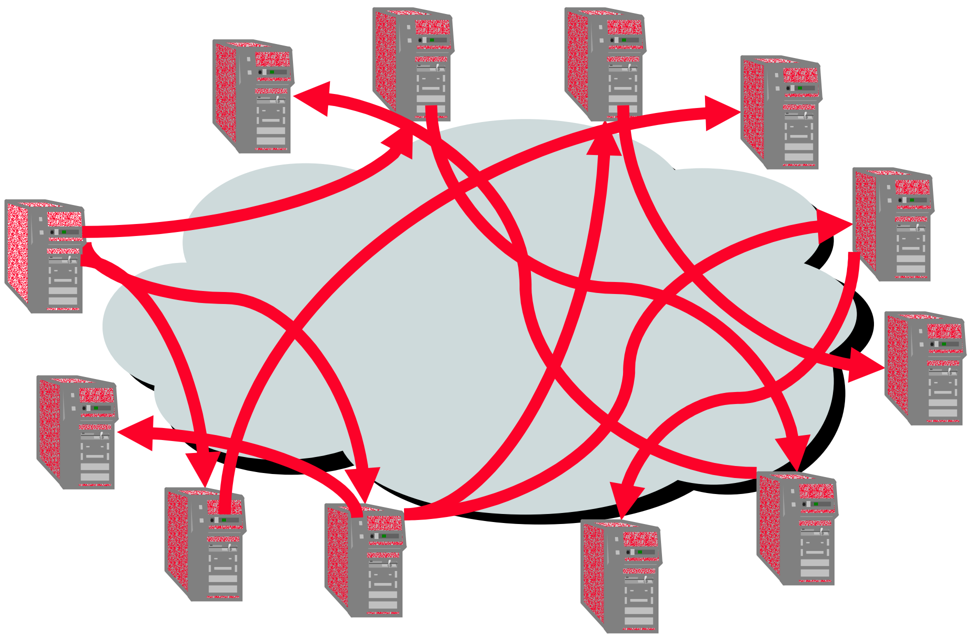
- Lots of attacks – some very large
  - **>12,000** attacks against **>5,000** targets
  - Most < **1,000** pps, but some over **600,000** pps
- Most attacks are short – some have long duration
  - a few victims were attacked continuously during the three week study
- Everyone is a potential target
  - Targets not dominated by any TLD or domain
    - Targets include large e-commerce sites, mid-sized business, ISPs, government, universities and end-users
    - Targets include routers and domain name servers
  - Something weird was happening in Romania

Outline

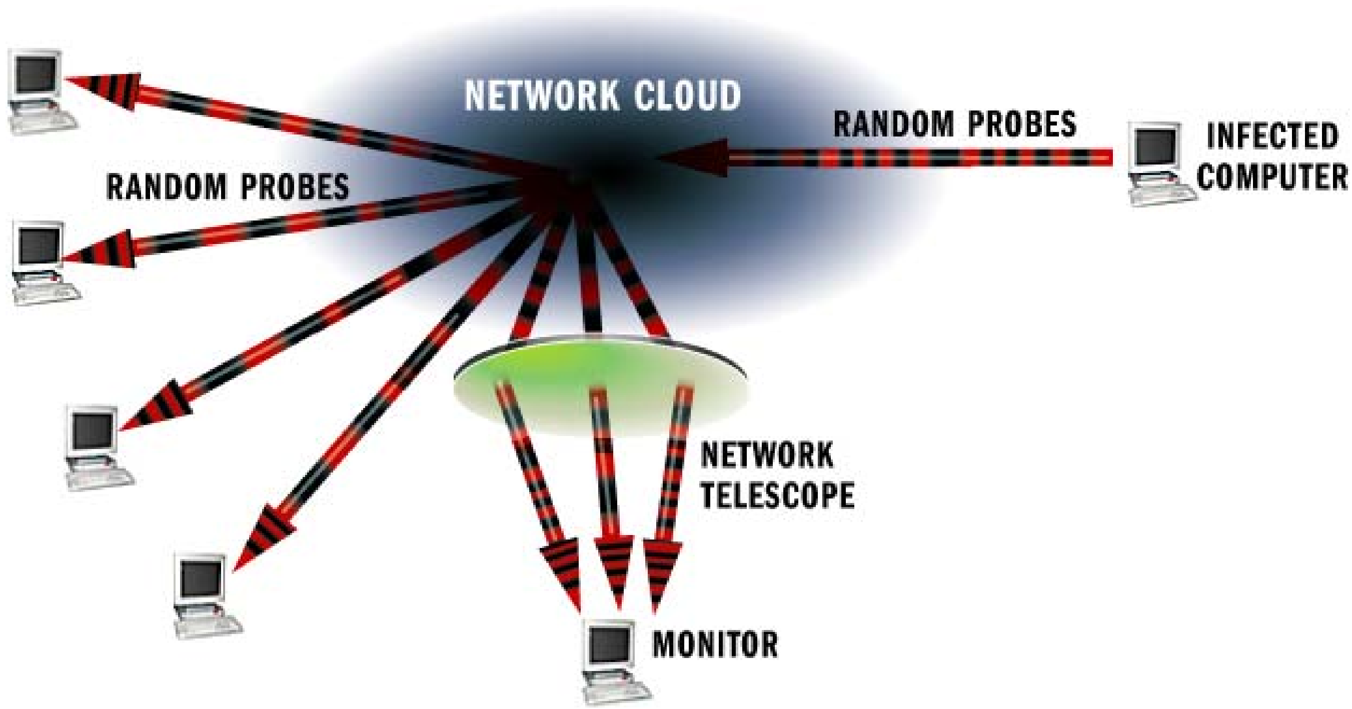
- What is a network telescope?
- Denial-of-Service Attacks
- **Internet Worms**
- How to use your own telescope

What is a Network Worm?

- Self-propagating self-replicating network program
  - Exploits some vulnerability to infect remote machines
    - No human intervention necessary
  - Infected machines continue propagating infection



Network Telescope: Worm Attacks



- Infected host scans for other vulnerable hosts by randomly generating IP addresses
- We monitor 1/256th of all IPv4 addresses
- We see 1/256th of all worm traffic of worms (when no bias or bugs)

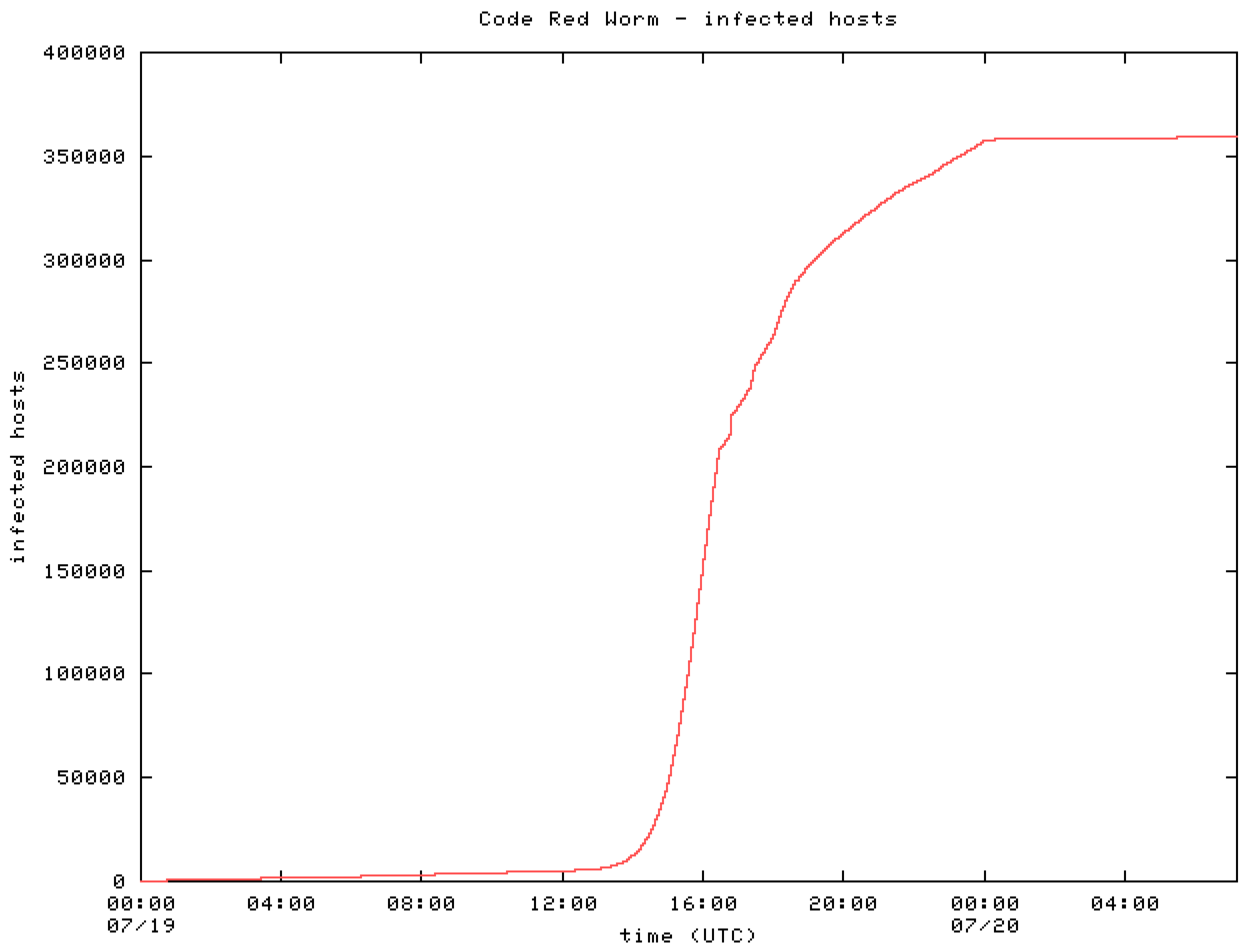
Code-Red worm – July 2001

- Exploits a vulnerability in Microsoft IIS
- Days 1-19 of each month
  - displays ‘hacked by Chinese’ message on English language servers
  - tries to open connections to infect randomly chosen machines using 100 threads
- Day 20-27
  - stops trying to spread
  - launches a denial-of-service attack on the IP address of www1.whitehouse.gov

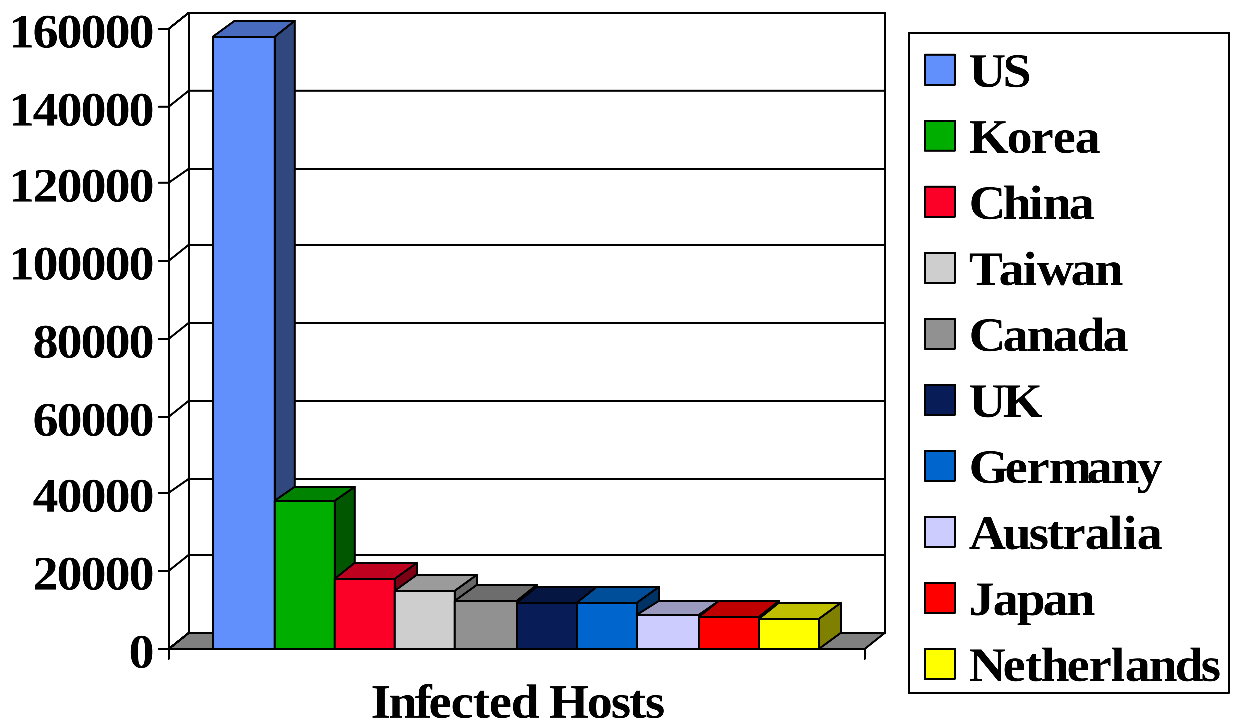
Code-Red Infection Rate

- 359,000 hosts infected in 24 hour period
- Between 11:00 and 16:00 UTC, the growth is exponential
- 2,000 hosts infected per minute at the peak of the infection rate (16:00 UTC)

Host Infection Rate



Host Characterization: Country of Origin



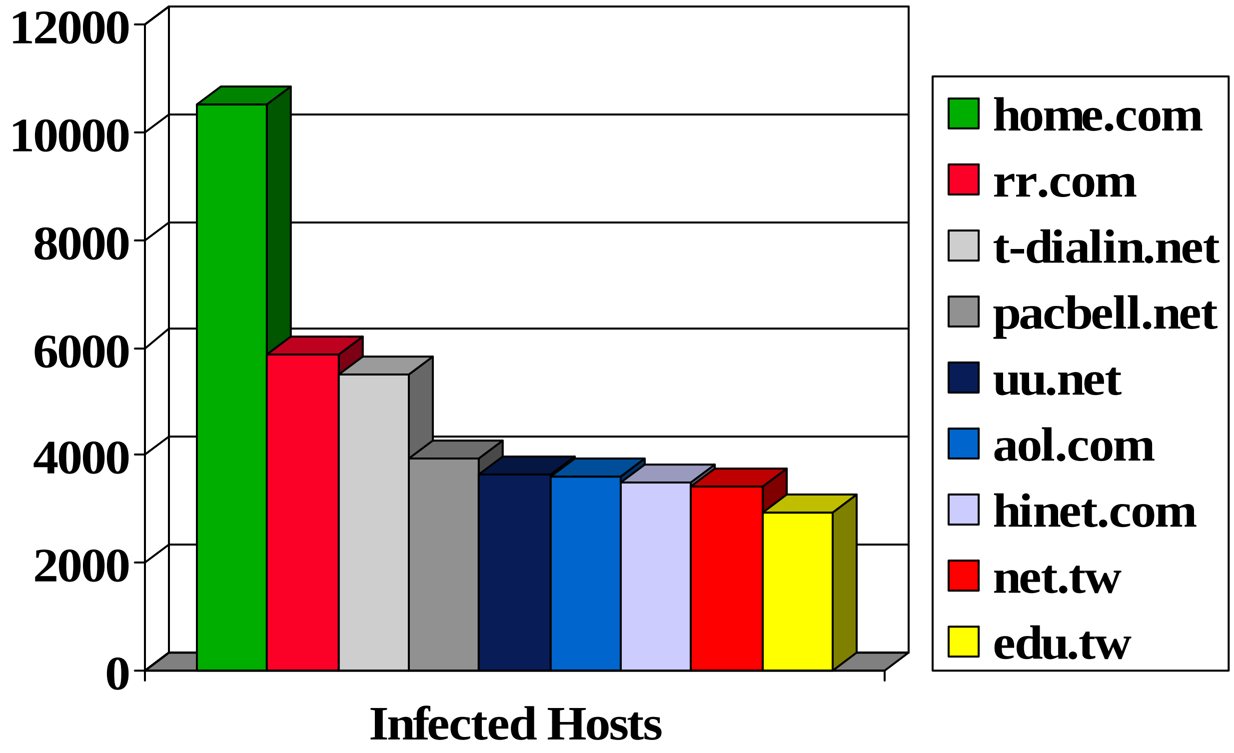
Host Characterization: Top-Level Domain (TLD)

- 47% of all infected hosts had no reverse DNS records, so we could not determine their TLDs
- .COM, .NET, and .EDU are all represented in proportions equivalent to their overall share of existing hosts
- 136 .MIL hosts and 213 .GOV hosts also infected
- 390 hosts on private networks (addresses in 10.0.0.0/8) infected, suggesting that private networks were vulnerable and many more private network hosts may be infected

Host Characterization: Domain

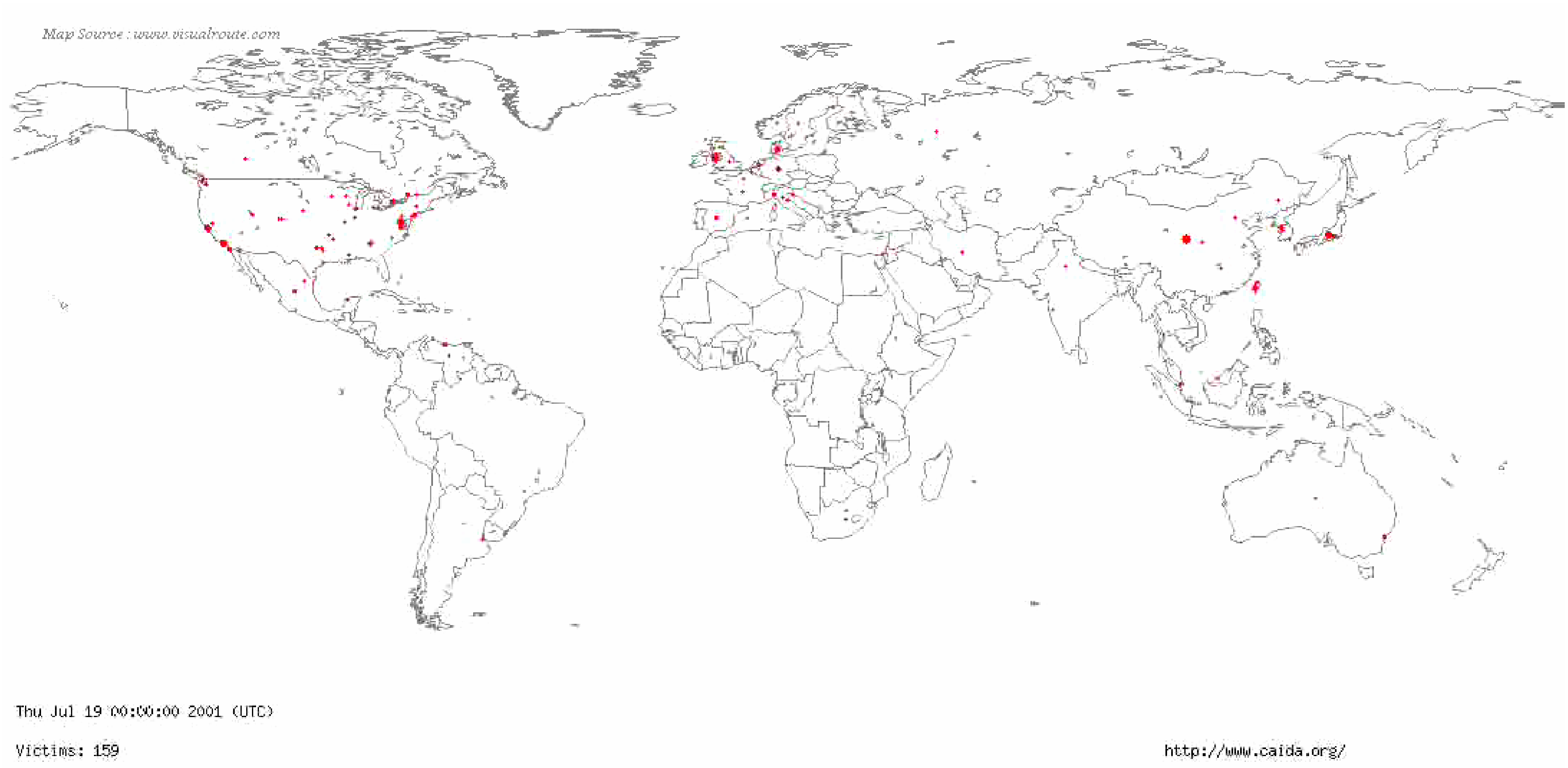
- ISPs providing connectivity to home and small-business users had the most infected hosts
- Machines maintained by home/small-business users (i.e. less likely to be maintained by a professional sysadmin) are an important aspect of global Internet health

Host Characterization: Domain



Internet Worm Attacks: Code-Red

(July 19, 2001)



University California, San Diego – Department of Computer Science

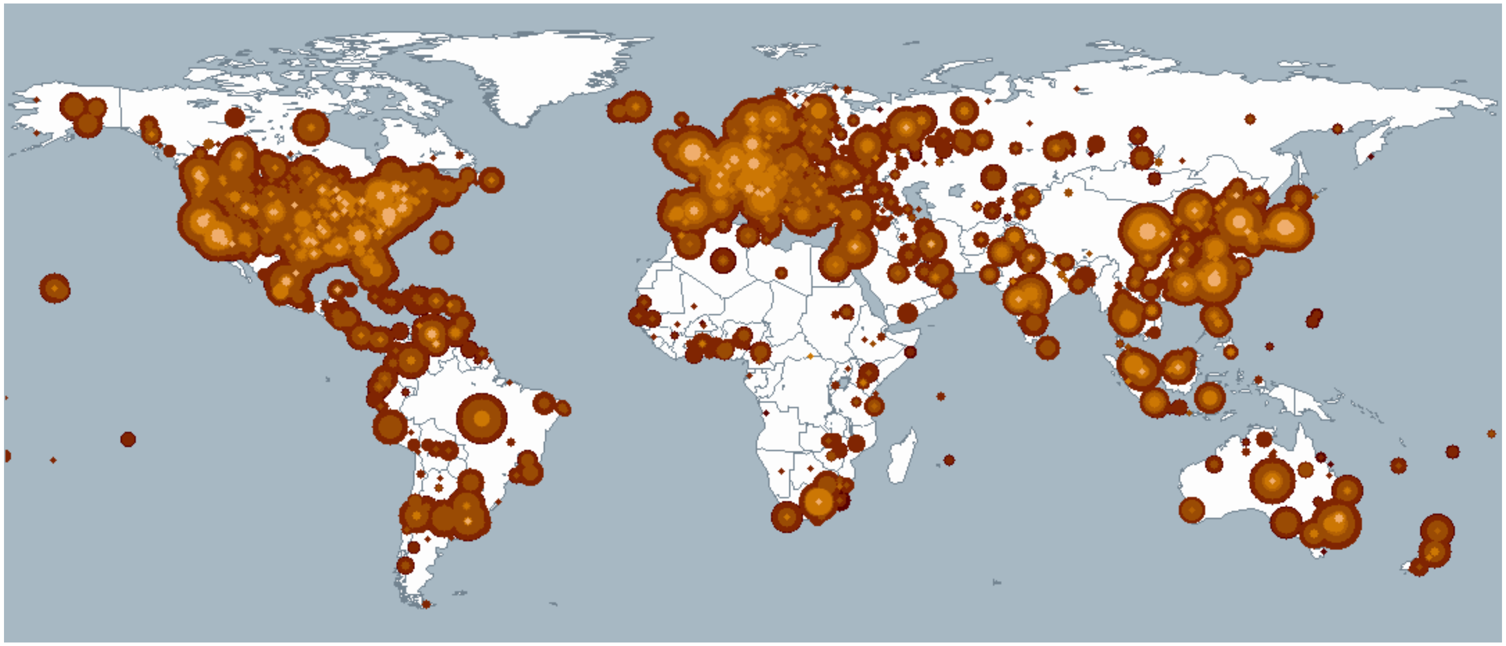
UCSD CSE

COOPERATIVE ASSOCIATION FOR INTERNET DATA ANALYSIS

caida

Internet Worm Attacks: Code-Red

(July 19, 2001)



- 360,000 hosts infected in *ten hours*
- No effective patching response
- More than $1.2 billion in economic damage in the first ten days
- Collateral damage: printers, routers, network traffic

Response to August 1st CodeRed

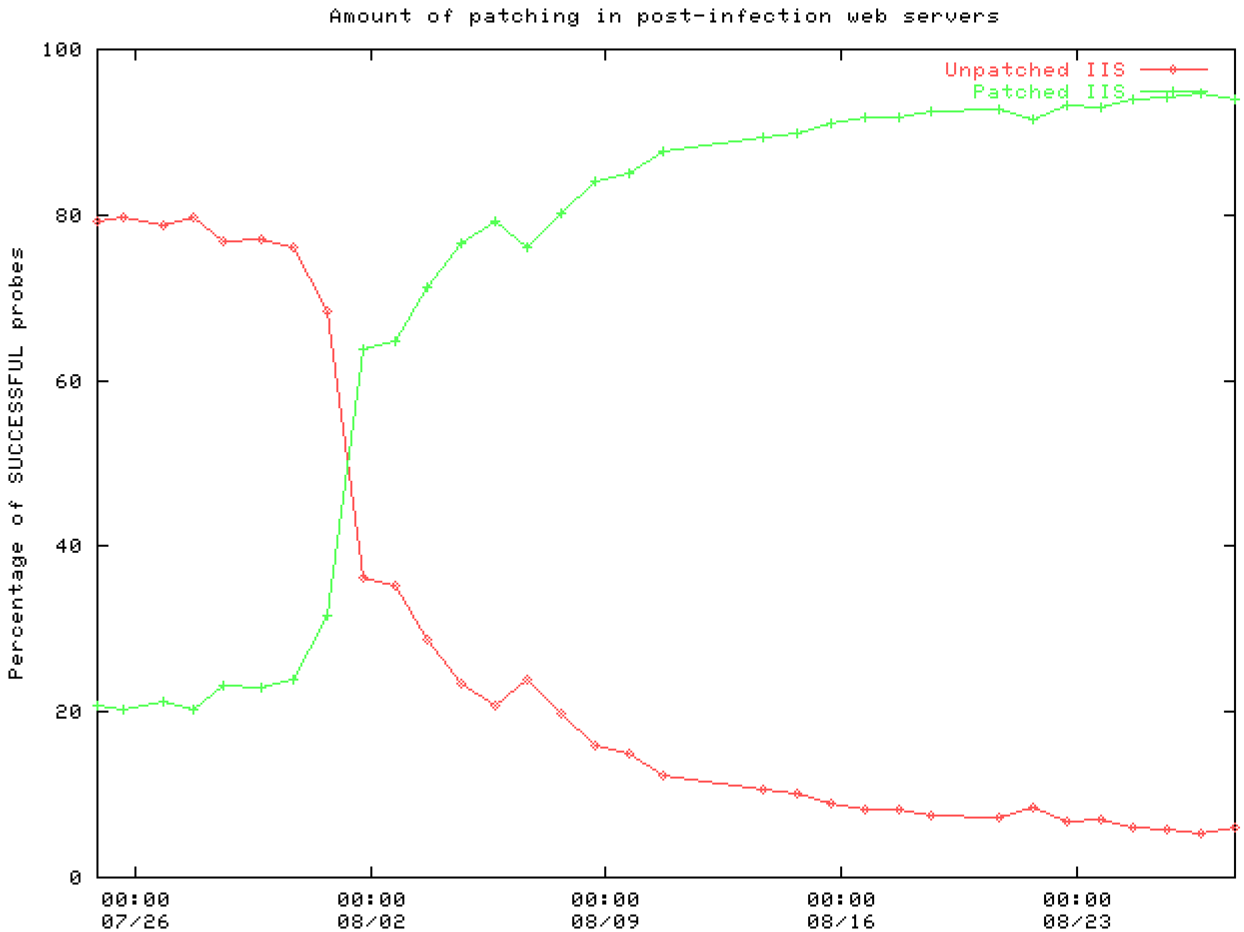
- CodeRed was programmed to deactivate on July 20th and begin spreading again on August 1st
- By July 30th and 31st, more news coverage than you can shake a stick at:
  - FBI/NIPC press release
  - Local ABC, CBS, NBC, FOX, WB, UPN coverage in many areas
  - National coverage on ABC, CBS, NBC, CNN
  - Printed/online news had been covering it since the 19th
- "Everyone" knew it was coming back on the 1st

- Best case for human response: known exploit with a viable patch and a known start date

Patching Survey

- How well did we respond to a best case scenario?
- Idea: randomly test subset of previously infected IP addresses to see if they have been patched or are still vulnerable
- 360,000 IP addresses in pool from initial July 19th infection
- 10,000 chosen randomly each day and surveyed between 9am and 5pm PDT

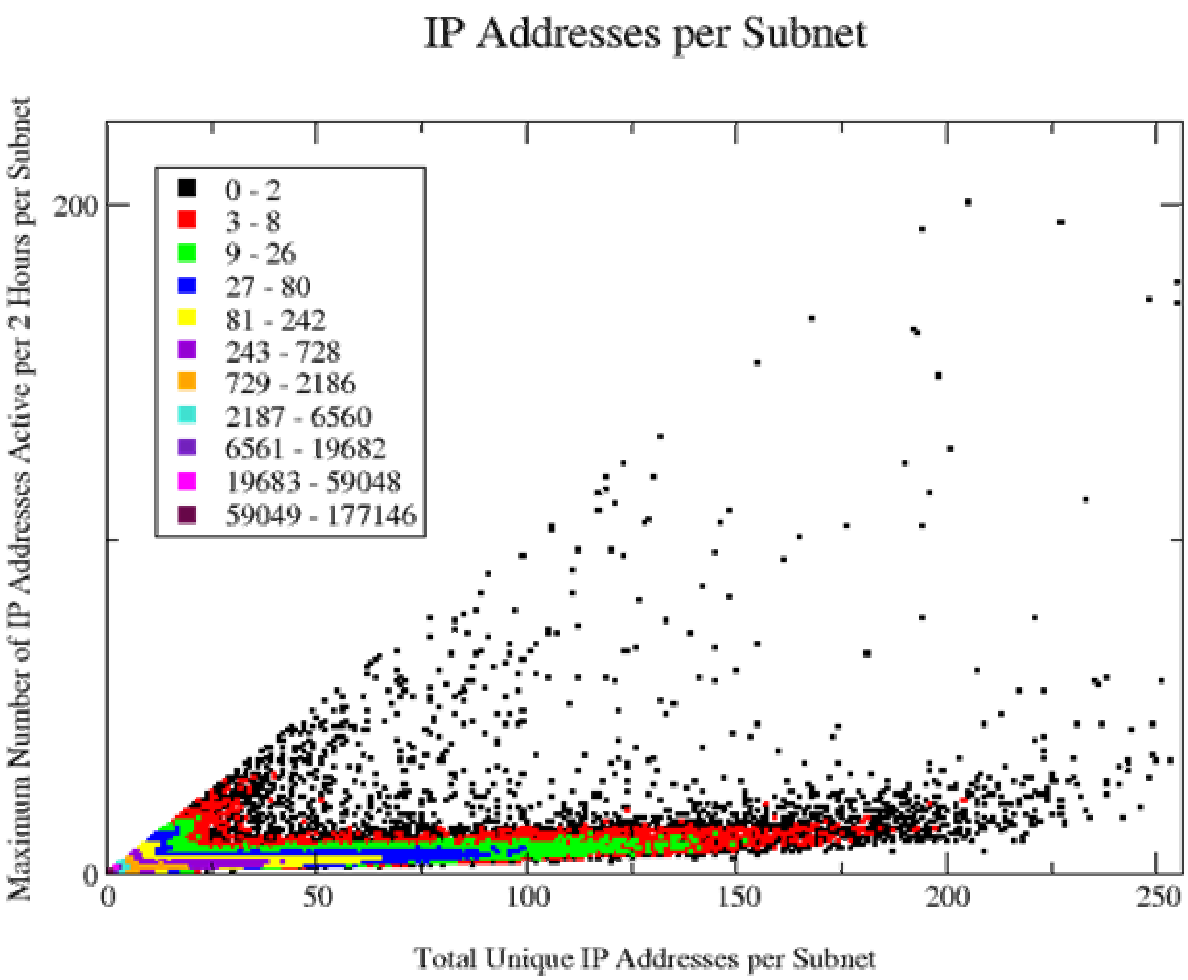
Patching Rate



Dynamic IP Addresses

- How can we tell how when an IP address represents an infected **computer**?
- Resurgence of CodeRed on Aug 1st: Max of ~180,000 unique IPs seen in any 2 hour period, but more than 2 million across ~a week.
- This ***DHCP effect*** can produce skewed statistics for certain measures, especially over long time periods

DHCP Effect seen in /24s



Summary of Recent Events

- **CodeRed** worm released in Summer 2001
  - Exploited buffer overflow in IIS
  - Uniform random target selection (after fixed bug in CRv1)
  - Infects 360,000 hosts in 10 hours (CRv2)
  - Still going…
- Starts **renaissance** in worm development
  - CodeRed II
  - Nimda
  - Scalper, Slapper, Cheese, etc.
- This year:
  - **Sapphire/Slammer** worm (Winter 2003)
  - Blaster, Welchia

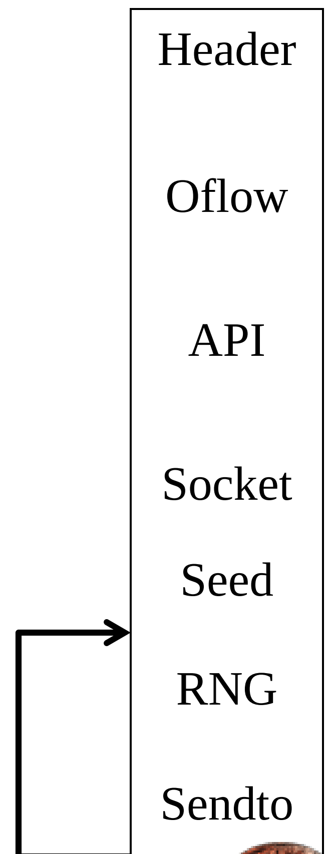
Inside the Sapphire/Slammer Worm

- Exploited bug in MSSQL 2000 and MSDE 2000
- Worm fit in a single UDP packet (404 bytes)

- Simple code structure
  - Cleanup from buffer overflow
  - Get API pointers

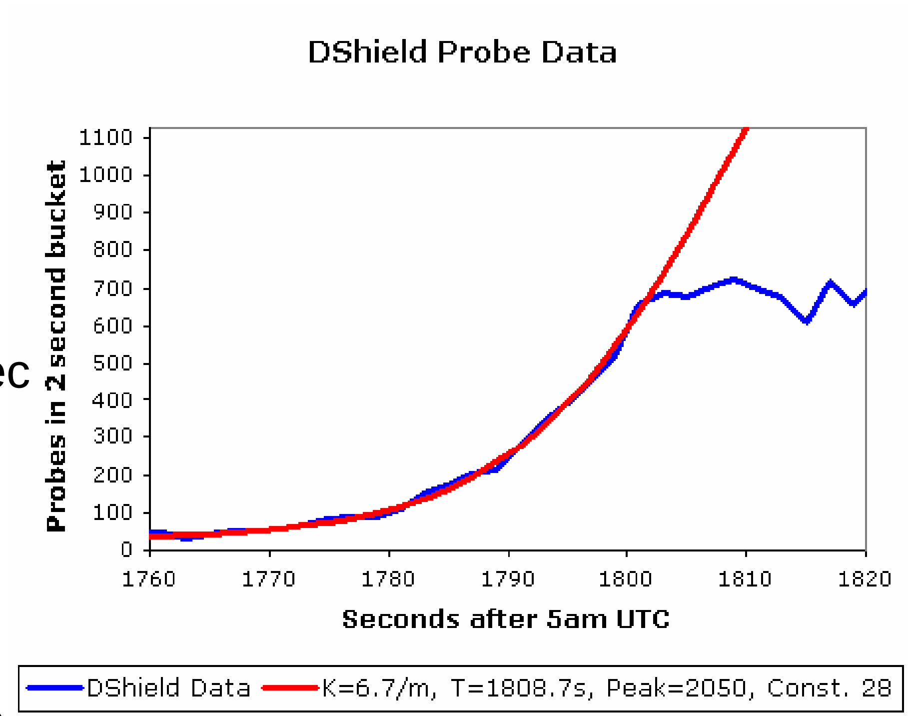
  } Code borrowed from published exploit

  - Create socket & packet
  - Seed RNG with `getTickCount()`
  - While (TRUE)
    - Increment RNG (mildly buggy)
    - Send packet to RNG address
- Key insight: non-blocking & stateless scanning (adaptable to TCP-based worms)

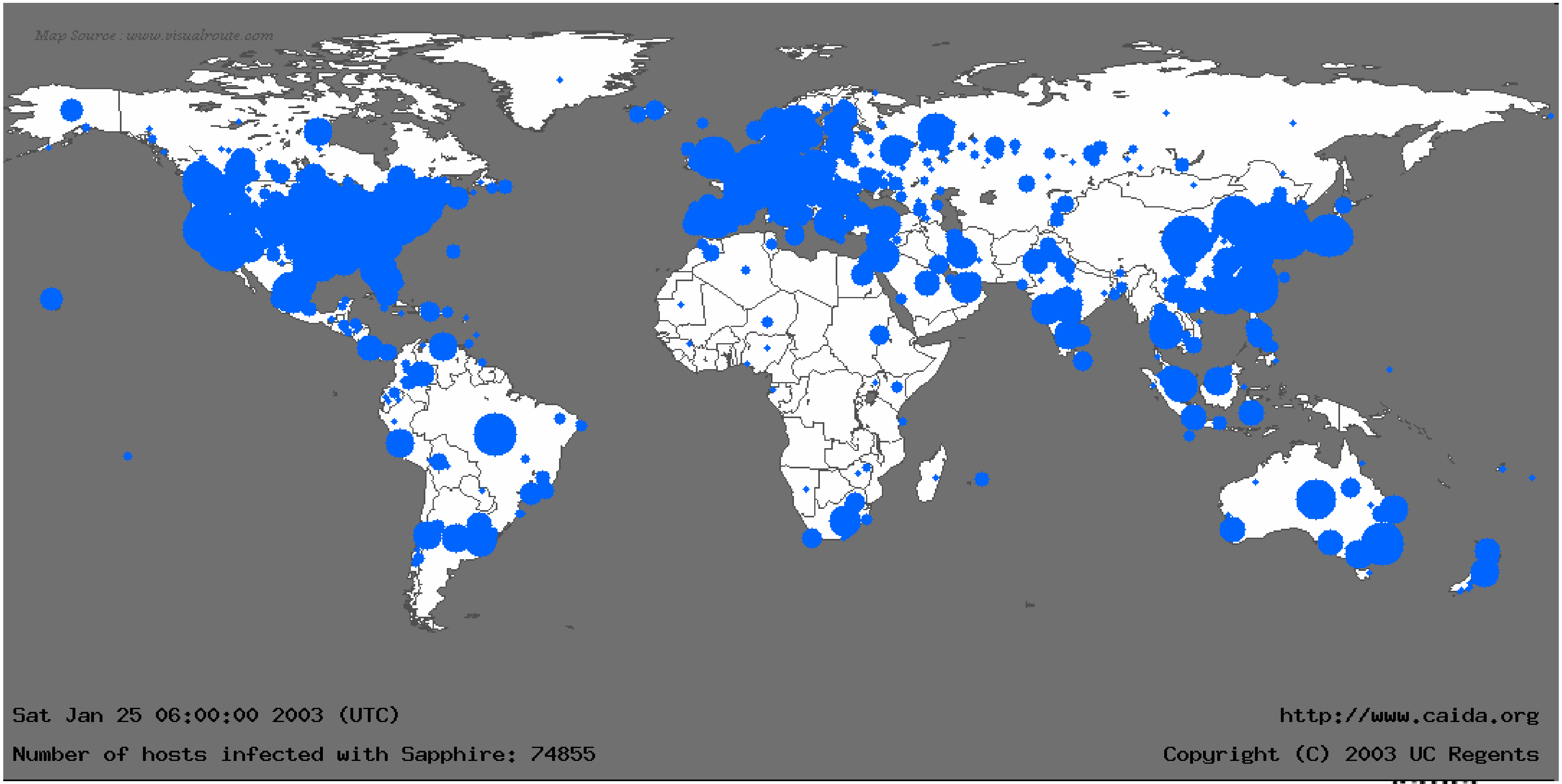


Sapphire growth

- First ~1min behaves like classic random scanning worm
  - Doubling time of ~8.5 seconds
  - Code Red doubled every 40mins
- >1min worm starts to saturate access bandwidth
  - Some hosts issue >20,000 scans/sec
  - Self-interfering
- Peaks at ~3min
  - 55million IP scans/sec
- 90% of Internet scanned in <10mins
  - Infected ~100k hosts (conservative due to PRNG errors)

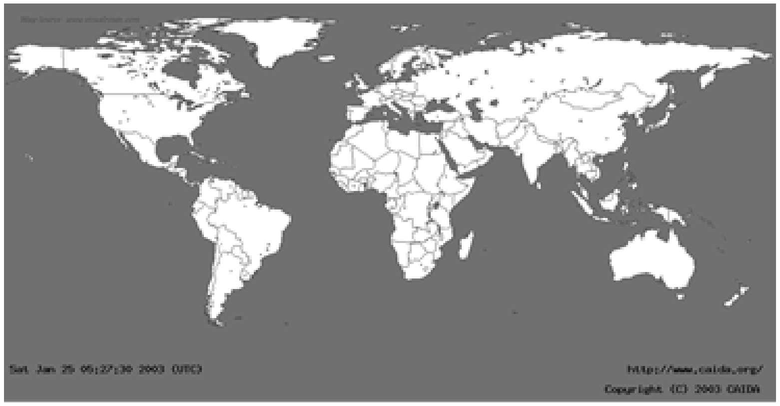


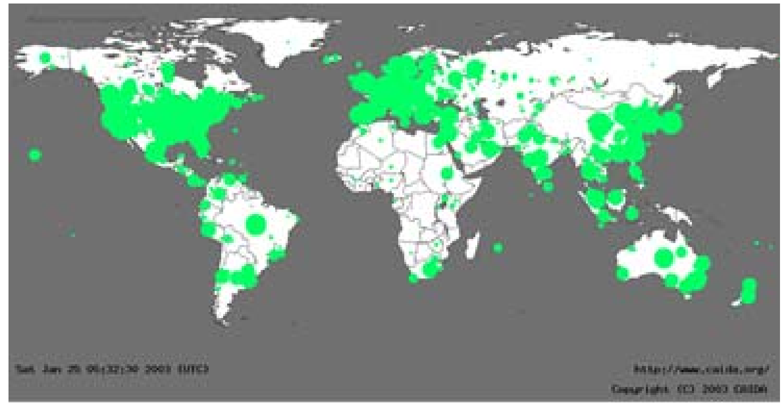
Sapphire Animation



Internet Worm Attacks: Sapphire

(aka SQL Slammer) – Jan 24, 2003





Before 9:30PM (PST) After 9:40PM (PST)

- ~100,000 hosts infected in *ten* ***minutes***
- Sent more than 55 million probes per second world wide
- Collateral damage: Bank of America ATMs, 911 disruptions, Continental Airlines cancelled flights
- Unstoppable; relatively benign to hosts

The Sky is Falling…

- **Worms are the worst Internet threat today**
  - Many *millions* of susceptible hosts
  - *Easy* to write worms
    - Worm payload separate from vulnerability exploit
    - Significant code reuse in practice
  - Possible to cause major damage
    - Lucky so far; existing worms have benign payload
    - Wipe disk; flash bios; modify data; reveal data; Internet DoS
- **We have no operational defense**
  - Good evidence that humans don't react fast enough
  - Defensive technology is nascent at best

What can we do?

- **Measurement**
  - What are worms doing?
  - What types of hosts are infected?
  - Are new defense mechanisms working?

- **Develop operational defense**
  - Can we build an automated system to stop worms?

Outline

- What is a network telescope?
- Denial-of-Service Attacks
- Internet Worms
- **How to use your own telescope**

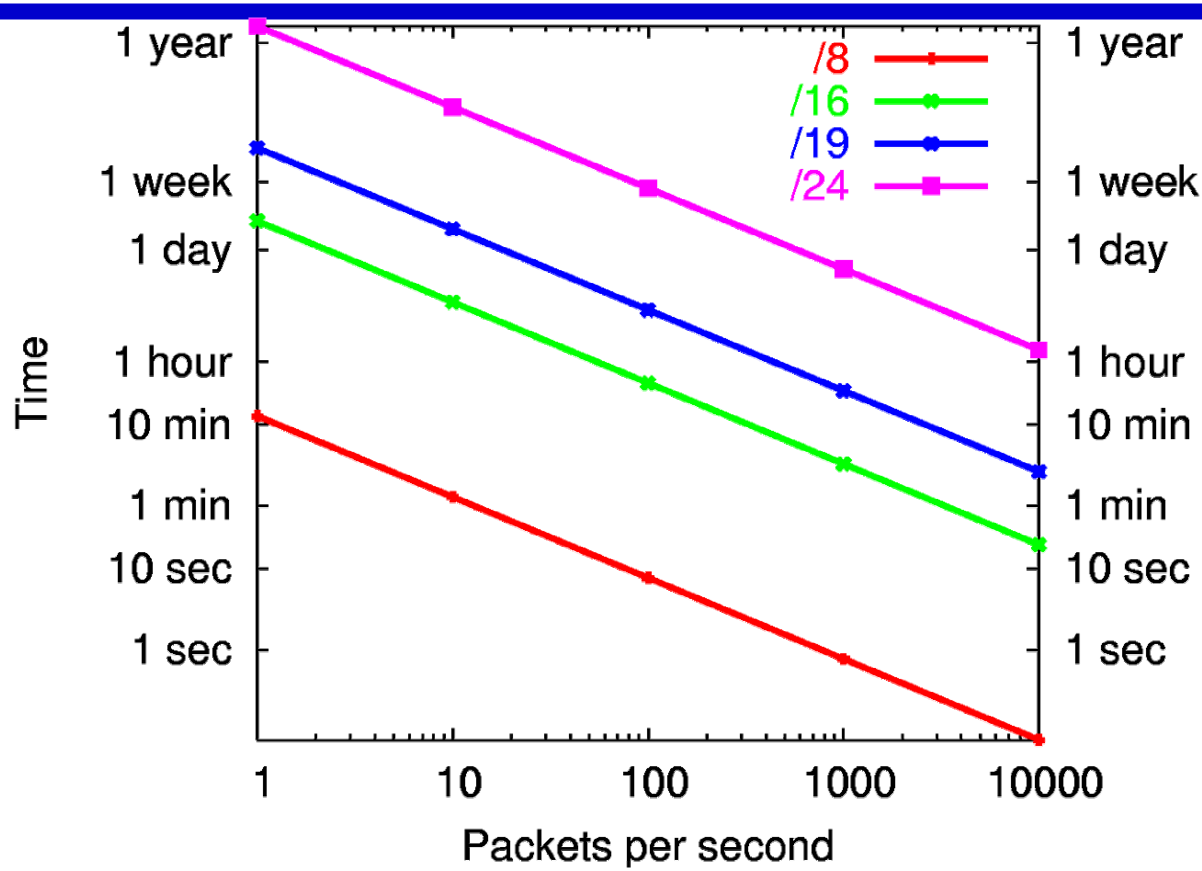
Using your own telescope: Effects of Size

- Larger telescopes are able to detect events that generate fewer packets, either because of short duration or low sending rate.

- Larger telescopes have better accuracy at determining the start and end times of an event.

- Using CIDR / notation on next few slides:
  - /8 = old class-A size, 16 million IP addresses
  - /16 = old class-B size, 65536 IP addresses

Detectable Events (95%)



Any event above and to the right of a line can be detected (at least one packet seen) with at least 95% probability.

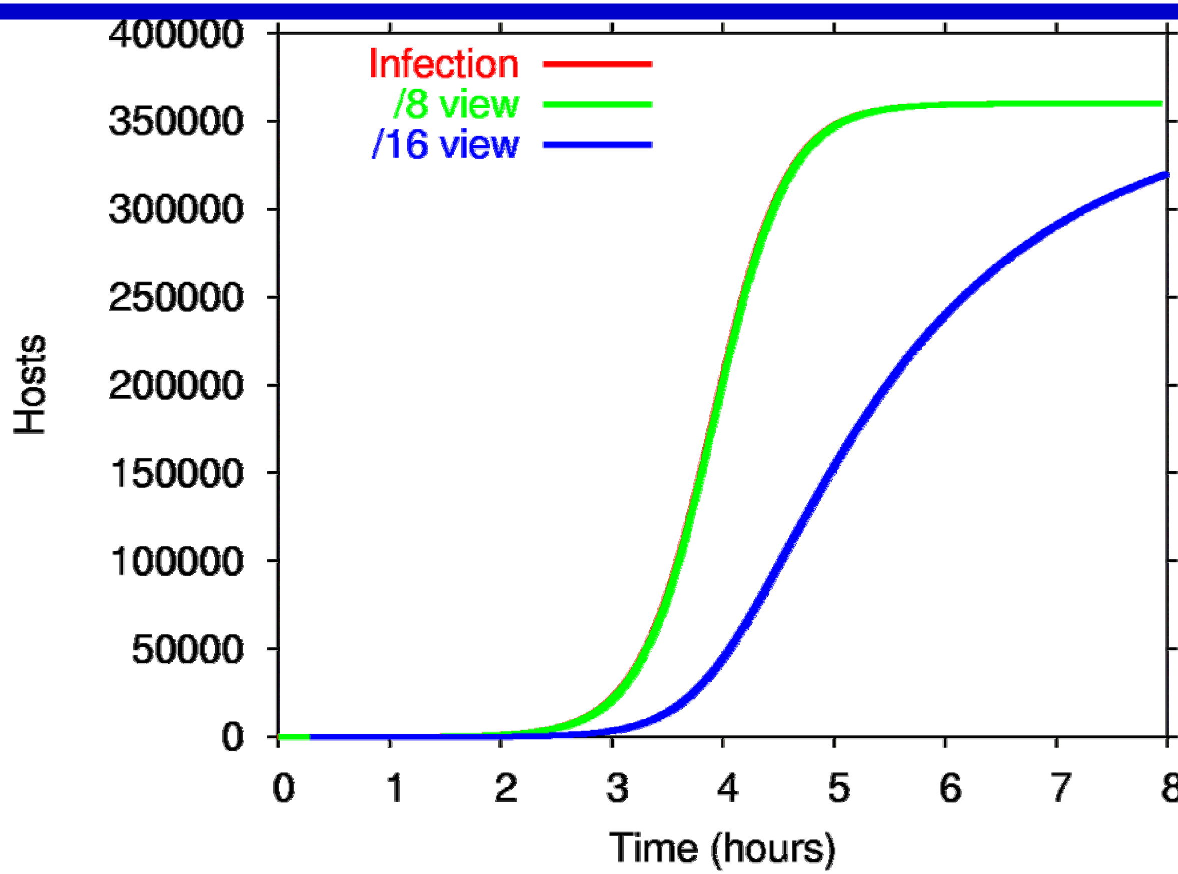
Detection Times - 10 pps events

(Code-Red approx. this rate)

Detection probability:	5%	50%	95%
/8	1.3 sec	18 sec	1.3 min
/14	1.4 min	19 min	1.4 hour
/15	3 min	38 min	2.7 hour
/16	6 min	1.3 hour	5.5 hour
/19	45 min	10 hour	1.8 day
/24	24 hours	14 day	58 day

Worm Spread – 10 probes/sec

(Code-Red approx. this rate)



- /8 telescope accurately tracks overall behavior of infection
- /16 telescope lags behind in time and shape is misleading

DoS Attack breakdown (/16 view) (three weeks in February 2001)

	Week1	Week2	Week3
Attacks /16 view	126	193	241
Attacks /8 view	4173	3878	4754

Organizational Telescopes

- Small telescopes may not be useful for observing external events
- However, setting up an internal facing telescope may help quickly identify internal problems
- With an internal facing telescope you can have /5 or better

Why have an internal telescope?

- Quickly detect internal machines infected with worms, certain kinds of misconfigurations, and potentially hacked machines.

- Capture data for hosts connecting to unallocated IP address space by:
  - if you use BGP (default-free) to all providers, you can point a default route at a monitor box
  - enable flow collection on your edge routers
  - announce a couple unallocated networks, but be careful if they ever get allocated by IANA (least desirable)

Extending it

- Combine a telescope watching traffic to unallocated IP addresses with monitoring all outbound traffic
  - you may notice anomalous behavior like a spam relay
  - verify that your firewall seems to be doing what you think
- Watch all *inbound* ICMP error messages, in particular HOST/NETWORK UNREACHABLE
  - evidence of scanning behavior
  - may show external connectivity & performance problems before users pick up the telephone

Tools to use

- Flow data (Cisco NetFlow, Juniper cflow, others):
  - FlowScan: http://net.doit.wisc.edu/~plonka/FlowScan
- Packet data
  - CoralReef report generator: http://www.caida.org/tools/
- Either
  - AutoFocus: http://ial.ucsd.edu/AutoFocus/
- Not an exhaustive list ☺

AutoFocus example

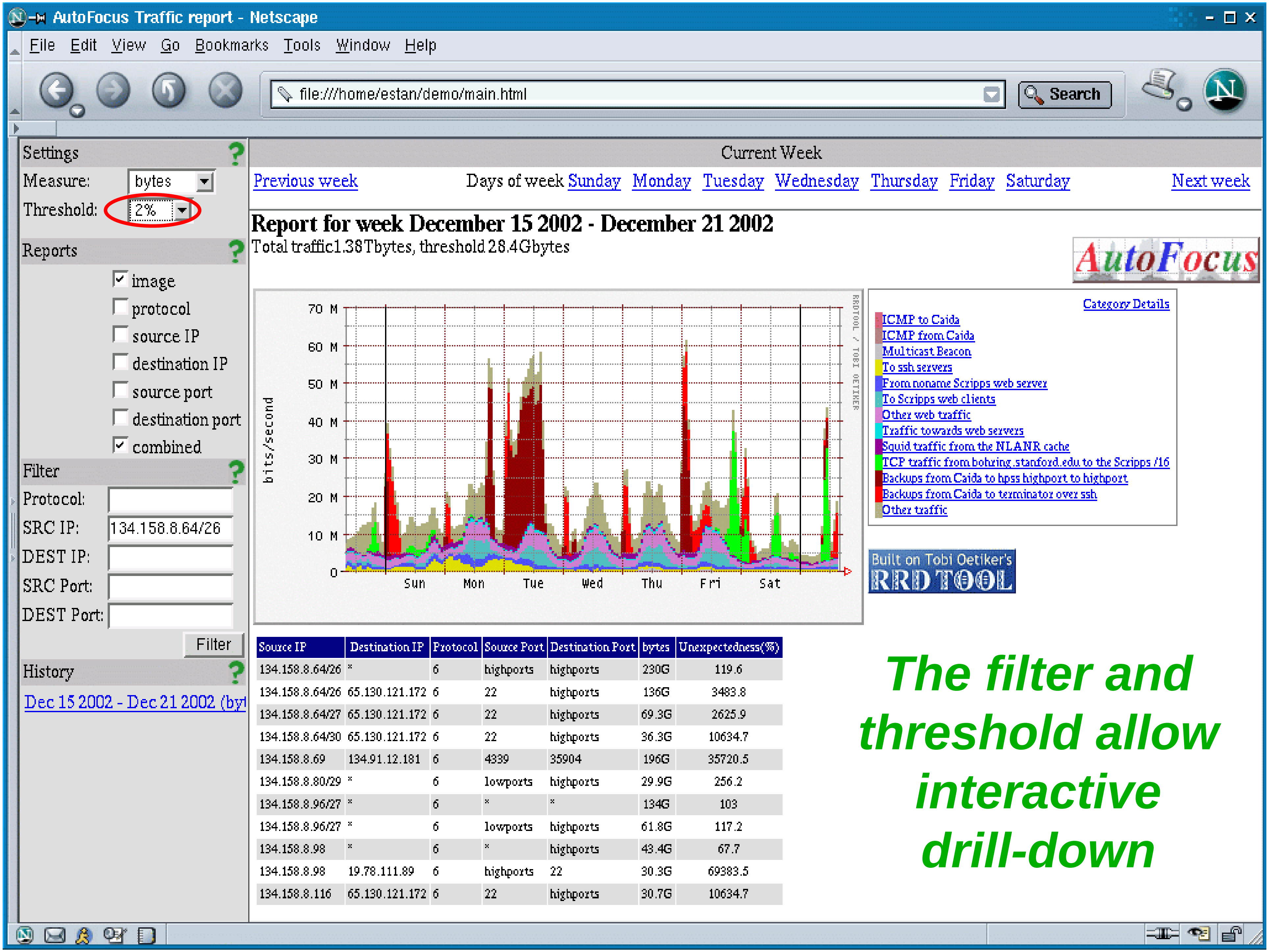
- Sapphire/SQL Slammer worm
  - Find worm port & proto automatically

Source IP	Destination IP	Protocol	Source Port	Destination Port	bytes	Unexpectedness(%)
*	*	6	highports	highports	827M	77.7
*	*	17	highports	1434	10.5G	112.6
*	152.249.0.0/16	*	*	*	604M	100
138.0.0.0/9	*	*	*	highports	3.66G	99.4
138.0.0.0/10	*	*	highports	*	3.68G	99.9
138.54.3.58	*	17	3341	1434	2.14G	672.5
138.54.11.4	*	17	7062	1434	950M	1551.3
152.249.56.0/22	*	*	highports	highports	723M	103.4
152.249.191.120	*	17	1959	1434	1.78G	810.0
152.249.191.121	96.0.0.0/8	17	1531	1434	645M	39523.7
152.249.210.3	*	17	4315	1434	2.36G	609.5
152.249.254.152	*	17	3787	1434	1.53G	941.8

AutoFocus example

- Sapphire/SQL Slammer worm
  - Can identify infected hosts

Source IP	Destination IP	Protocol	Source Port	Destination Port	bytes	Unexpectedness(%)
*	*	6	highports	highports	827M	77.7
*	*	17	highports	1434	10.5G	112.6
*	152.249.0.0/16	*	*	*	604M	100
138.0.0.0/9	*	*	*	highports	3.66G	99.4
138.0.0.0/10	*	*	highports	*	3.68G	99.9
138.54.3.58	*	17	3341	1434	2.14G	672.5
138.54.11.4	*	17	7062	1434	950M	1551.3
152.249.56.0/22	*	*	highports	highports	723M	103.4
152.249.191.120	*	17	1959	1434	1.78G	810.0
152.249.191.121	56.0.0.0/8	17	1531	1434	645M	39523.7
152.249.210.3	*	17	4315	1434	2.36G	609.5
152.249.254.152	*	17	3787	1434	1.53G	941.8



Source IP	Destination IP	Protocol	Source Port	Destination Port	bytes	Unexpectedness(%)
134.158.8.64/26	*	6	highports	highports	230G	119.6
134.158.8.64/26	65.130.121.172	6	22	highports	136G	3483.8
134.158.8.64/27	65.130.121.172	6	22	highports	69.3G	2625.9
134.158.8.64/30	65.130.121.172	6	22	highports	36.3G	10634.7
134.158.8.69	134.91.12.181	6	4339	35904	196G	35720.5
134.158.8.80/29	*	6	lowports	highports	29.9G	256.2
134.158.8.96/27	*	6	*	*	134G	103
134.158.8.96/27	*	6	lowports	highports	61.8G	117.2
134.158.8.98	*	6	*	highports	43.4G	67.7
134.158.8.98	19.78.111.89	6	highports	22	30.3G	69383.5
134.158.8.116	65.130.121.172	6	22	highports	30.7G	10634.7

The filter and threshold allow interactive drill-down

I'm a DoS Victim, help!!

- Different providers are *different*. While there is a trend towards bigger customers getting better service, the variability between ISPs is huge.

- Talk with your provider. Find out what they can do to help, before you are attacked. Make this part of your bidding and purchase process.

What can I buy?

- Several DoS products on the market. Many of them work better in your provider rather than at your access link.
- Understand if your threat is pipe-filling (lots of packets), server-loading (filling up SYN state on machine), or content-based (slow DB queries, SSL, etc).
- SYN cookies in many OSes and load balancing can help with server-loading.

Worms are after me

- VPNs and laptops are a leading cause of worm entry behind the firewall. Why do your users land behind your firewall? Why do you have a firewall at all?

- Some products out there. Best involve partitioning your network into multiple cells and detect worm-like behavior, not static signature filtering.

Conclusions

- Network telescopes provide insight into non-local network events
- Larger telescopes better capture the behavior of events and can see smaller events
- Build your own internal telescope – it's fun AND easy.

Related CAIDA/UCSD Papers

- Inferring Internet Denial-of-Service Activity [MSV01]
  - David Moore, Stefan Savage, Geoff Voelker
  - http://www.caida.org/outreach/papers/2001/BackScatter/
- Code-Red: A Case Study on the spread and victims of an Internet Worm [MSB02]
  - David Moore, Colleen Shannon, Jeffrey Brown
  - http://www.caida.org/outreach/papers/2002/codered/
- Internet Quarantine: Requirements for Containing Self-Propagating Code [MSVS03]
  - David Moore, Colleen Shannon, Geoff Voelker, Stefan Savage
  - http://www.caida.org/outreach/papers/2003/quarantine/
- The Spread of the Sapphire/Slammer Worm [MPS03]
  - David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, Nicholas Weaver
  - http://www.caida.org/outreach/papers/2003/sapphire/

Additional CAIDA/UCSD Information

- Code-Red v1, Code-Red v2, CodeRedII, Nimda
  - http://www.caida.org/analysis/security/code-red/
- Code-Red v2 In-depth analysis
  - http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml
- Spread of the Sapphire/SQL Slammer Worm
  - http://www.caida.org/analysis/security/sapphire/
- Network telescopes
  - http://www.caida.org/analysis/security/telescope/