The Spread of the Witty Worm

Colleen Shannon

David Moore

cshannon @ caida.org

dmoore @ caida.org

www.caida.org

UCSD Campus LISA, July 15, 2004

UCSD CSE

What is CAIDA?

- Cooperative Association for Internet Data Analysis
- Goals include measuring and understanding the global Internet.
- Develop measurement and analysis tools
- Collect and provide Internet data: topology, header traces, bandwidth testlab, network security, DNS
- Visualization of the network

Outline

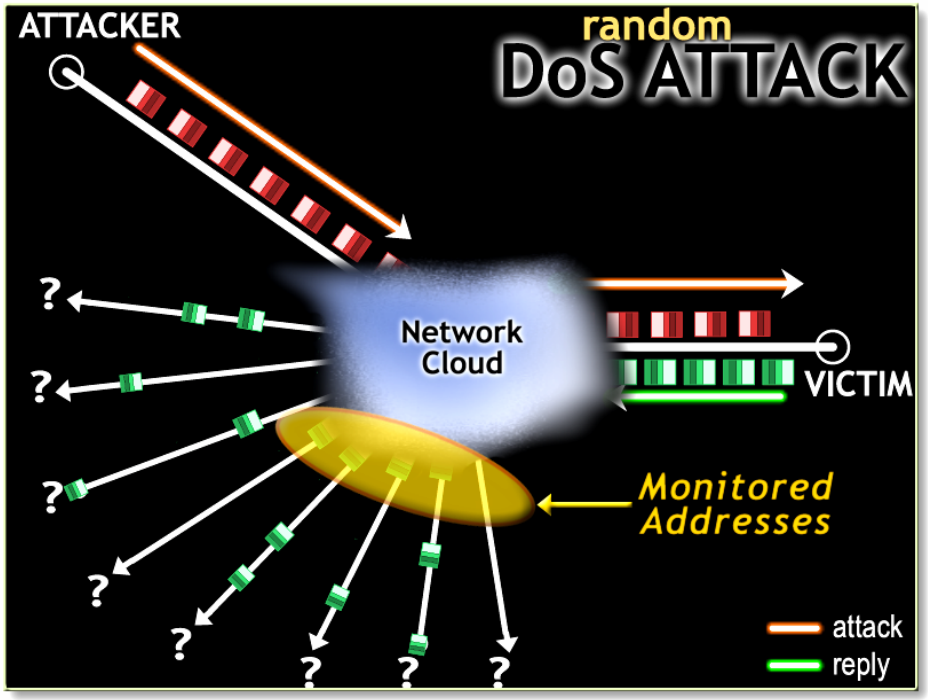
- What is a Network Telescope?
- The SCO Denial-of-Service Attack
- Past Internet Worms
  - Code-Red
  - SQL Slammer
- The Witty Internet Worm
  - Background
  - Witty Worm Spread
  - Witty Worm Victims
  - Conclusions

Network Telescope

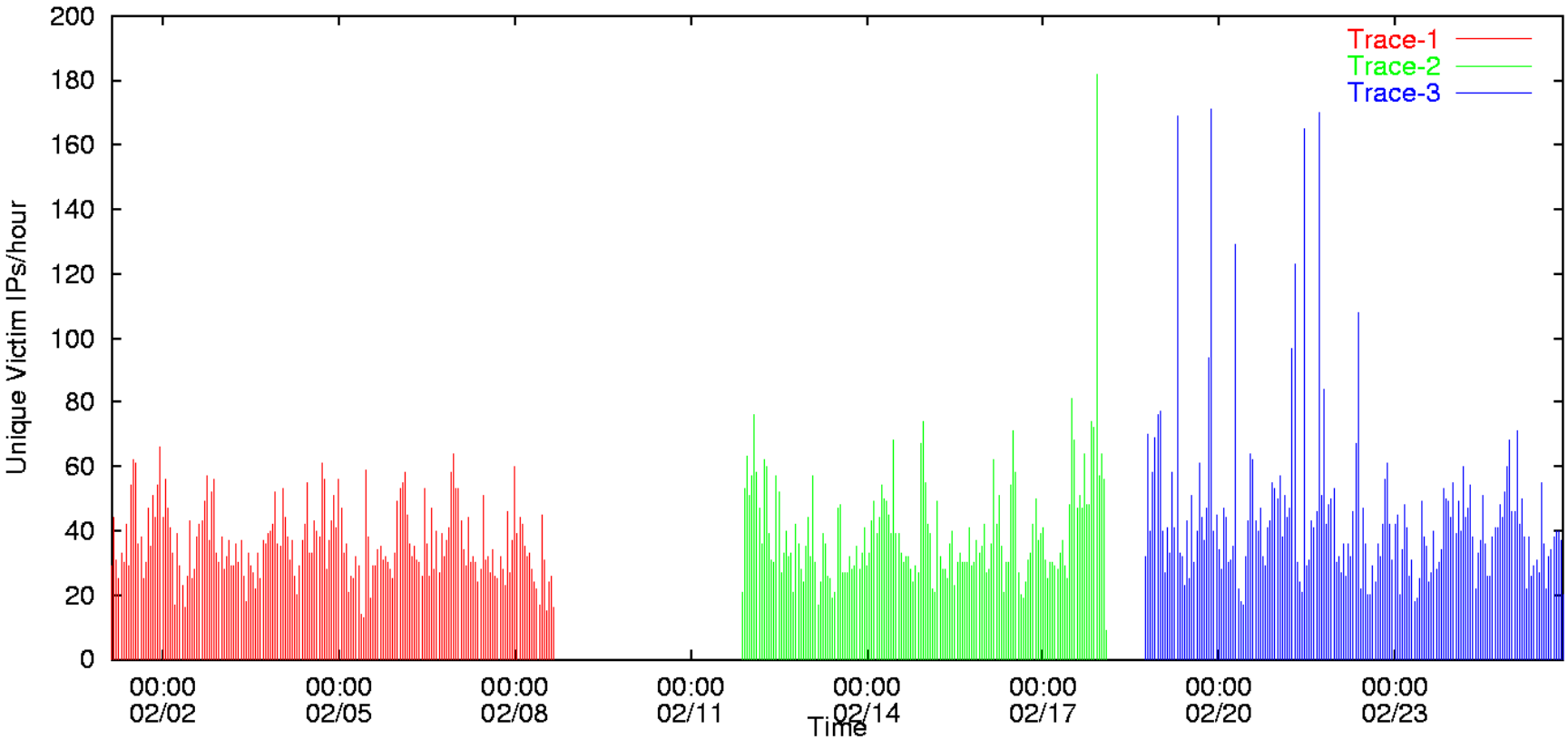
- Chunk of (globally) routed IP address space
  - 16 million IP addresses
- Little or no legitimate traffic (or easily filtered)
- Unexpected traffic arriving at the network telescope can imply remote network/security events
- Generally good for seeing explosions, not small events
- Depends on random component in spread

Network Telescope: Denial-of-Service Attacks

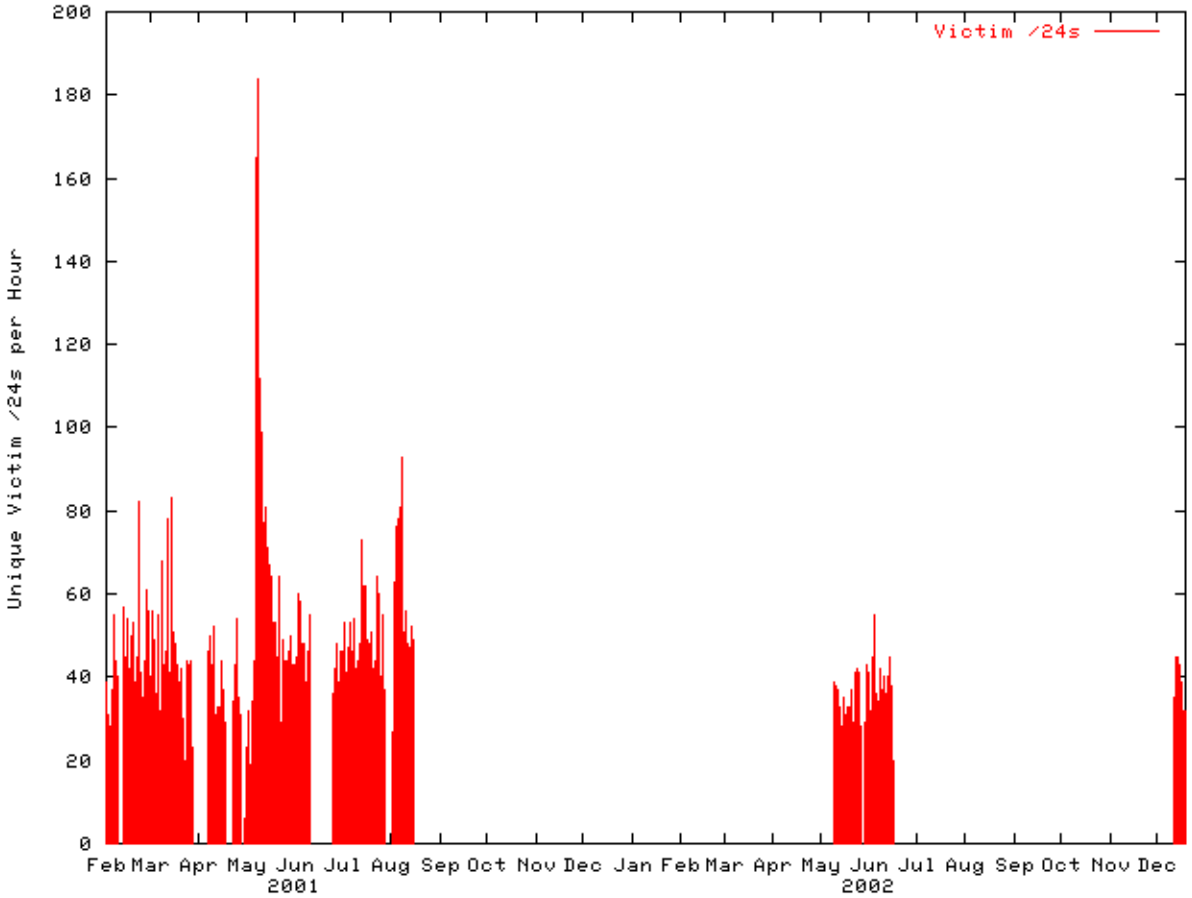
- Attacker floods the victim with requests using random spoofed source IP addresses
- Victim believes requests are legitimate and responds to each spoofed address
- We observe 1/256th of all *victim responses* to spoofed addresses [MSV01]



Denial-of-Service Attacks



DoS Attacks over time



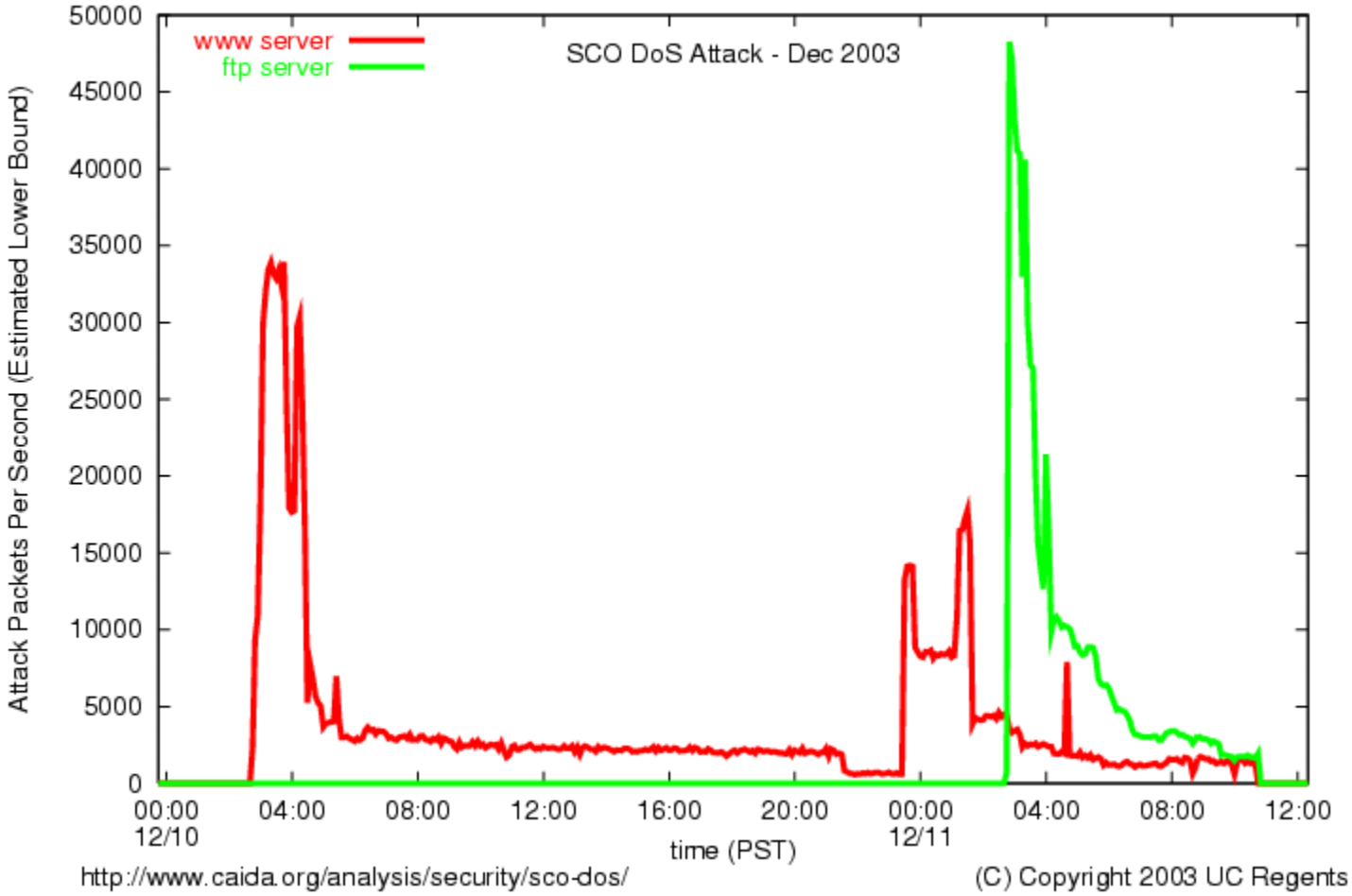
SCO Denial-of-Service Attack

- Who is SCO?
  - UNIX (linux) software company
  - Originally Santa Cruz Operations
  - Caldera bought Unix Server Division from Santa Cruz Operations in August of 2000
  - Caldera changed its name to "The SCO Group" in Aigist 2002
  - Sued IBM in March 2003 claiming that IBM misappropriated its UNIX operating system intellectual property (acquired from Novell)
  - Threatened lawsuits against many others

SCO Denial-of-Service Attack Timeline

- May 2003: SCO gets hit by its first major DoS Attack
- August 2003: SCO gets hit by its second major DoS Attack
  - random rumors that an internal network problem was publicized as a DoS attack
- December 10, 2003 3:20 AM: an ~340,000 MB/s SYN flood incapacitates SCO's web servers
- December 10, 2003 1:37 PM: groklaw.net blog "reports" on rumors that SCO is not being attacked; they are faking the whole thing to implicate the open source community
- December 11, 2003 2:50 AM: the SYN flood is expanded to target SCO's ftp server in addition to their webservers
- December 11, 3003 noon: SCO takes themselves off the 'net while pursuing upstream filters to block the attack

SCO Denial-of-Service Attack



SCO DoS Attack "Results"

- Security experts (us included) need to be careful what they say in the absence of details
  - Sure, technology exists to thwart SYN floods, but not at 340 MB/s inbound coming to a DS3 (45 MB/s duplex)
- It's no fun to be a SCO network admin
  - your own ISP won't admit they give you connectivity, let alone corroborate the attack reports
  - your CEO is quoting the aforementioned security experts who say any 5 year old could stop the attack
  - your only hope is upstream ISPs helping you, but your company is not popular with NOC employees

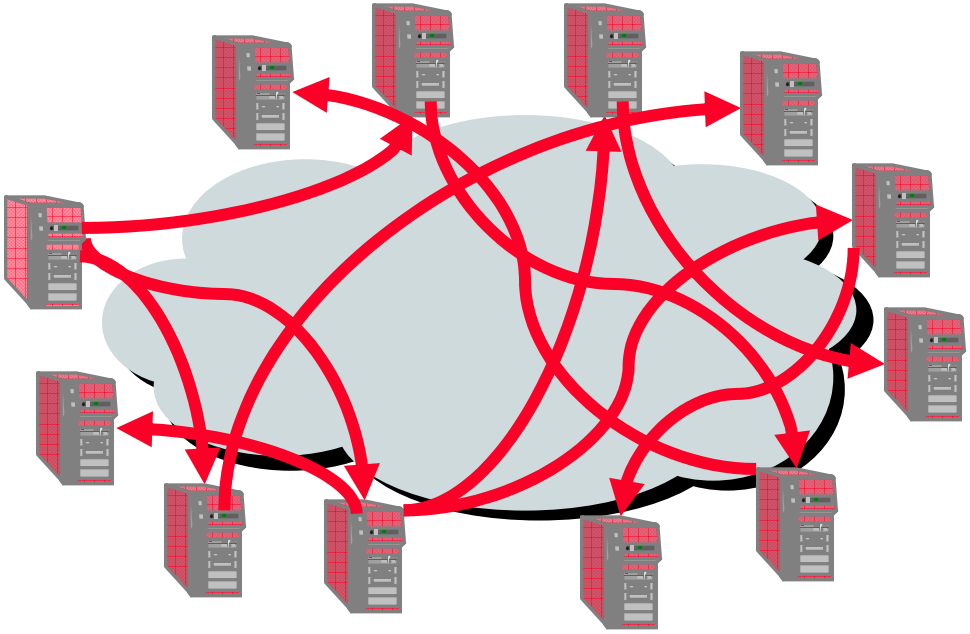
SCO DoS Attack Results (continued)

- Why did folks believe SCO was faking the attack?

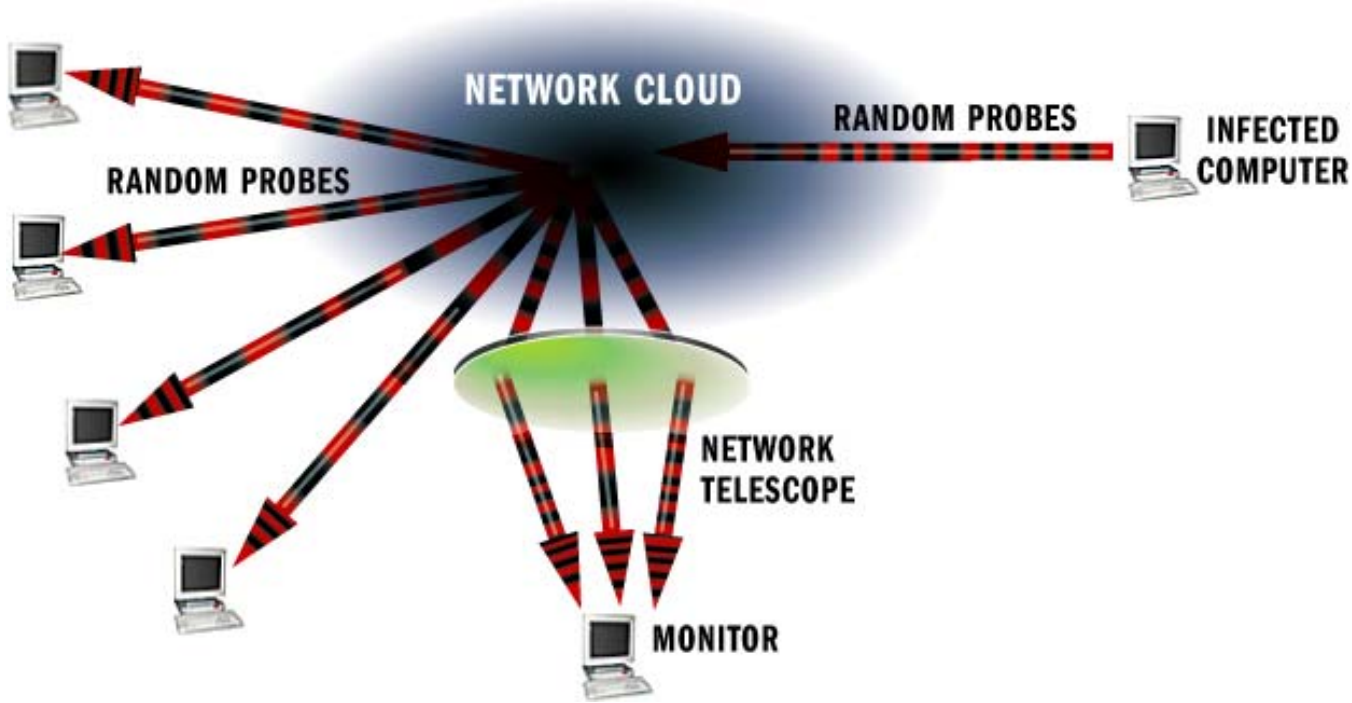
- People are paranoid and gullible (especially in large groups)
  - SCO is incredibly unpopular; why was it surprising that they were attacked
  - A real security problem: same reason phishing works…

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



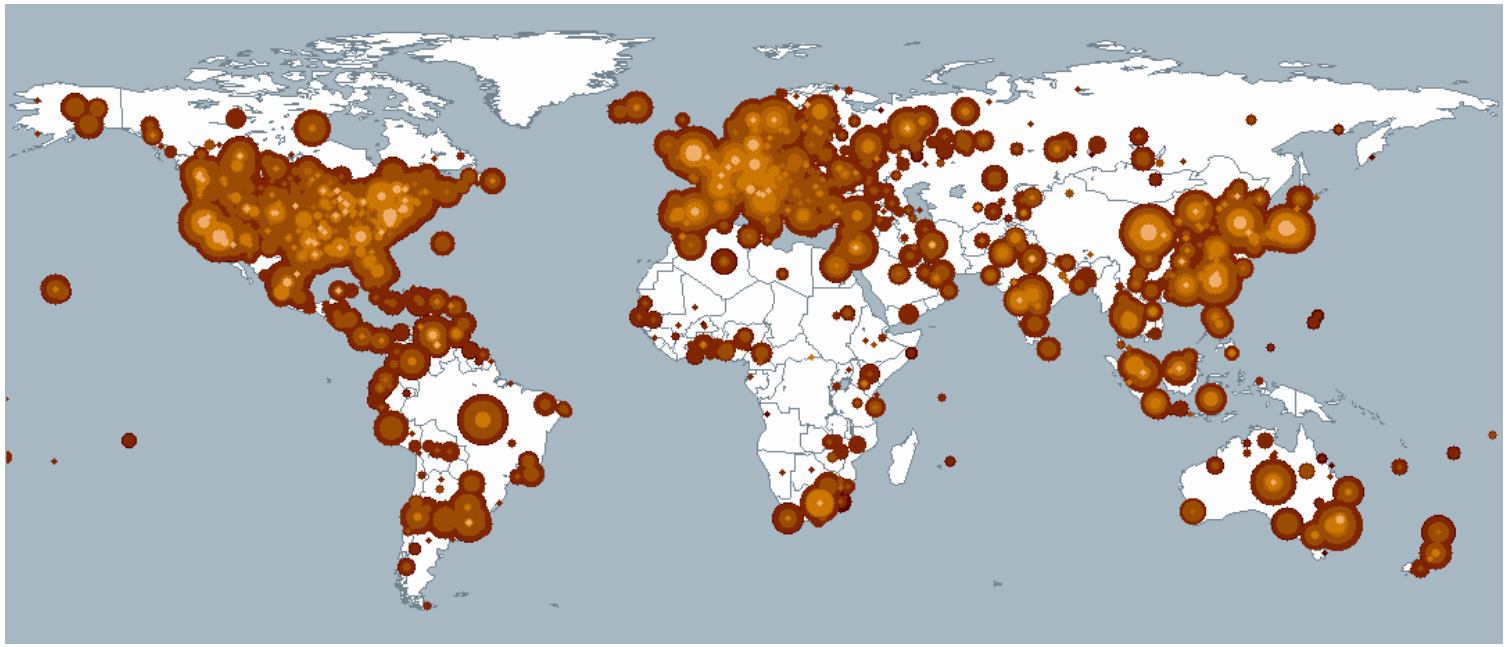
- Infected host scans for other vulnerable hosts by randomly generating IP addresses
- We monitor 1/256th of all IPv4 addresses
- We see 1/256th of all worm traffic of worms with no bias and no bugs

Internet Worm Attacks: Code-Red

(July 19, 2001)

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

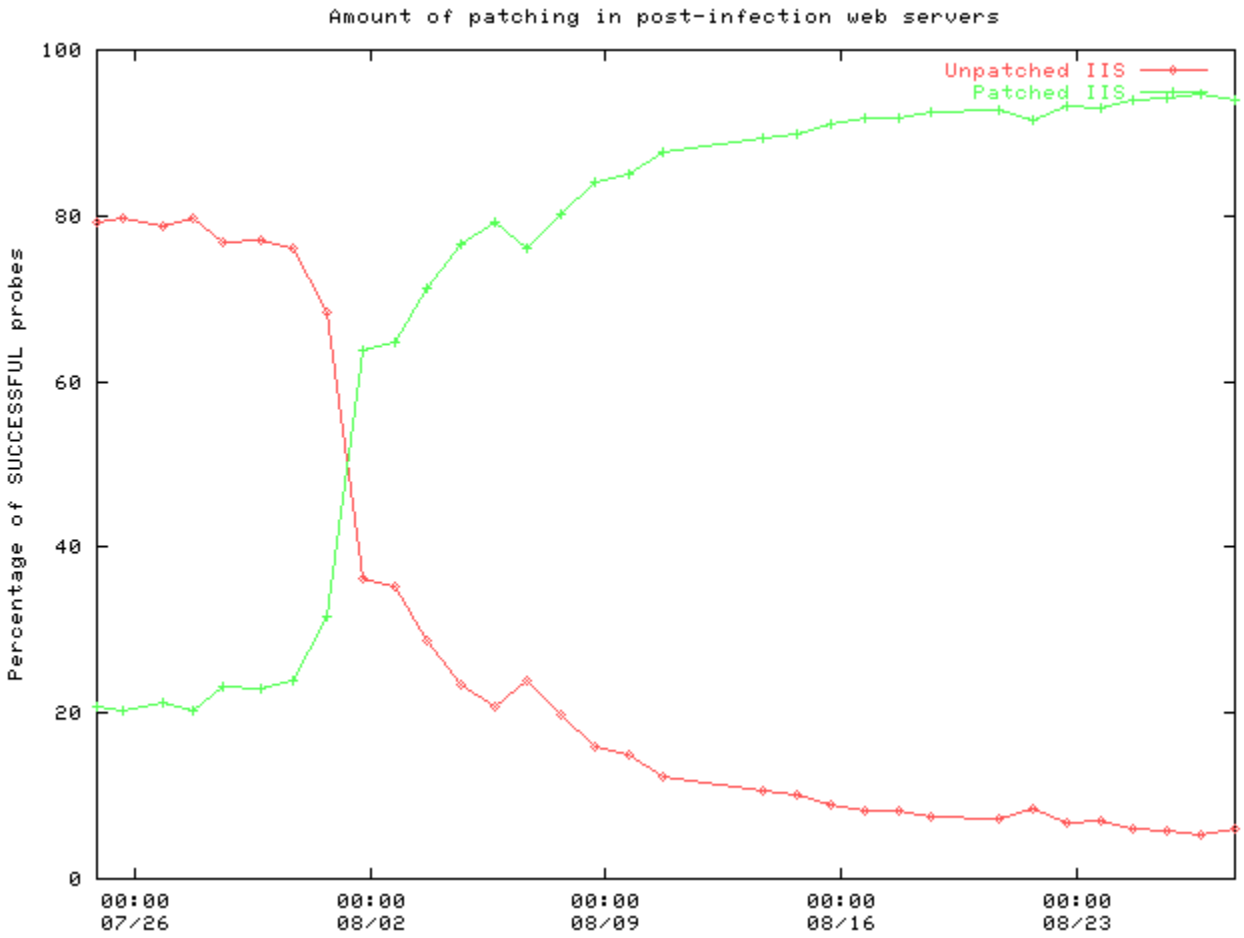
- CodeRed was programmed to deactivate on July 20th and begin spreading again on August 1st
- By July 30th and 31st, more news coverage than you can shake a stick at:
  - FBI/NIPC press release
  - Local ABC, CBS, NBC, FOX, WB, UPN coverage in many areas
  - National coverage on ABC, CBS, NBC, CNN
  - 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

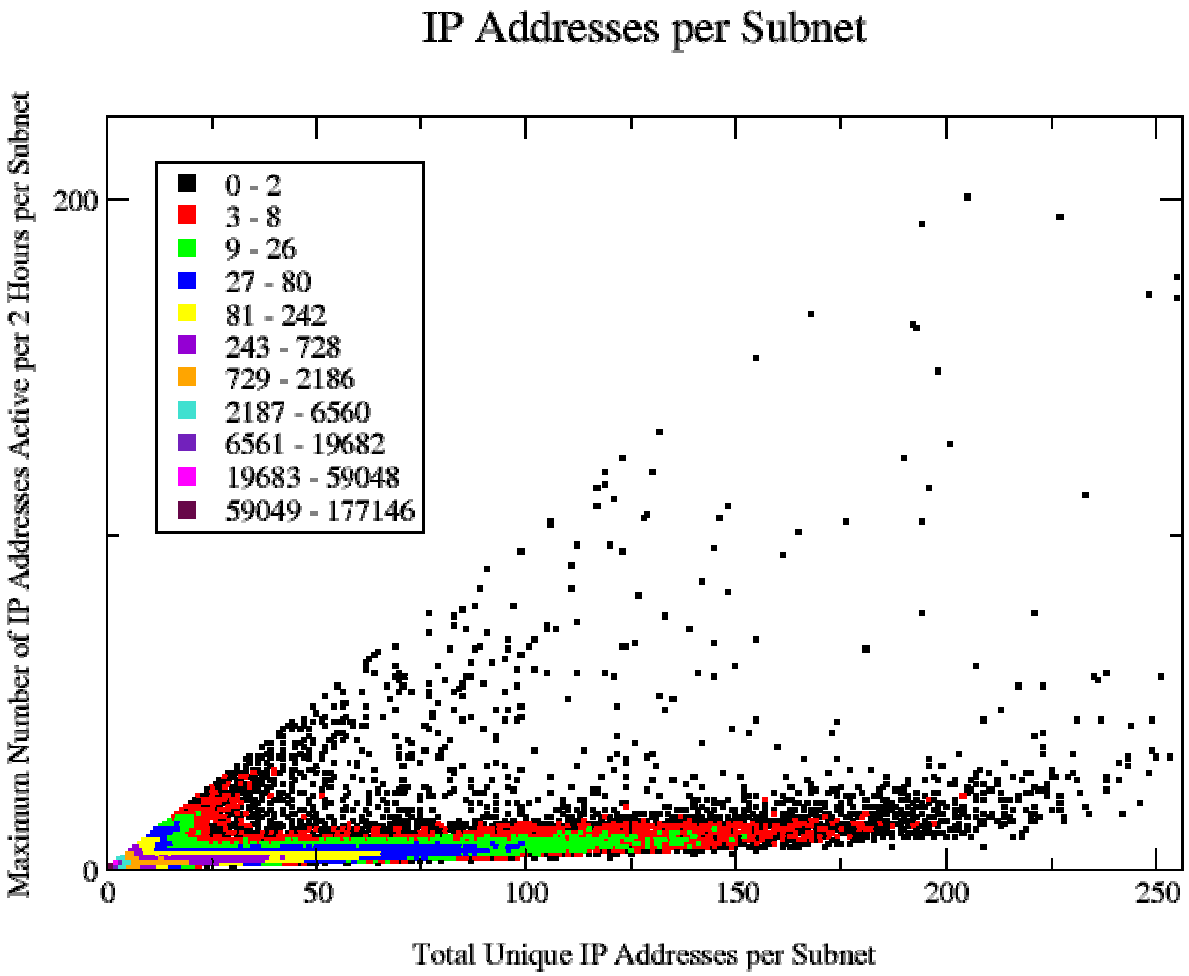
Patching Rate



Dynamic IP Addresses

- How can we tell how when an IP address represents an infected **computer**?
- Resurgence of CodeRed: 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 TCP worm development
  - CodeRed II
  - Nimda
  - Scalper, Slapper, Cheese, etc.
- Whoa dude, UDP can go fast!
  - **Sapphire/Slammer** worm (Winter 2003)
  - **Witty** worm (March 2004)

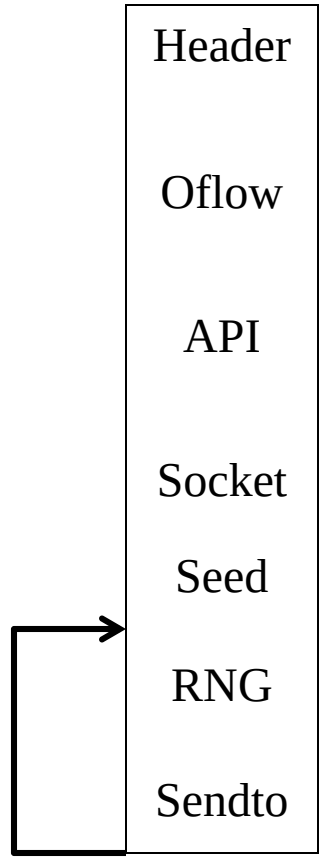
Inside the Sapphire/Slammer Worm

- 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

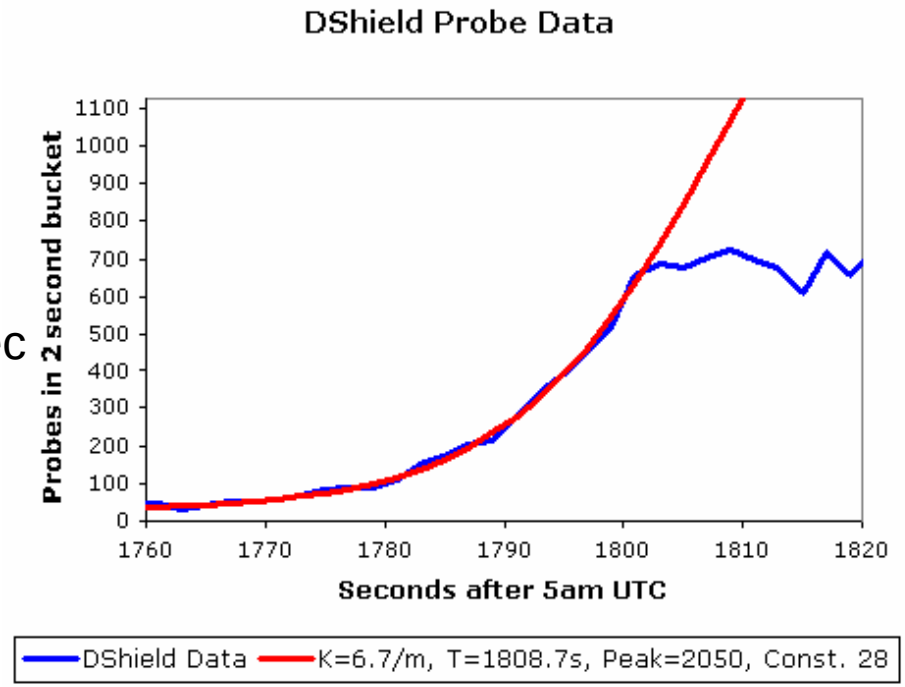
  Code borrowed from published exploit

  - 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)



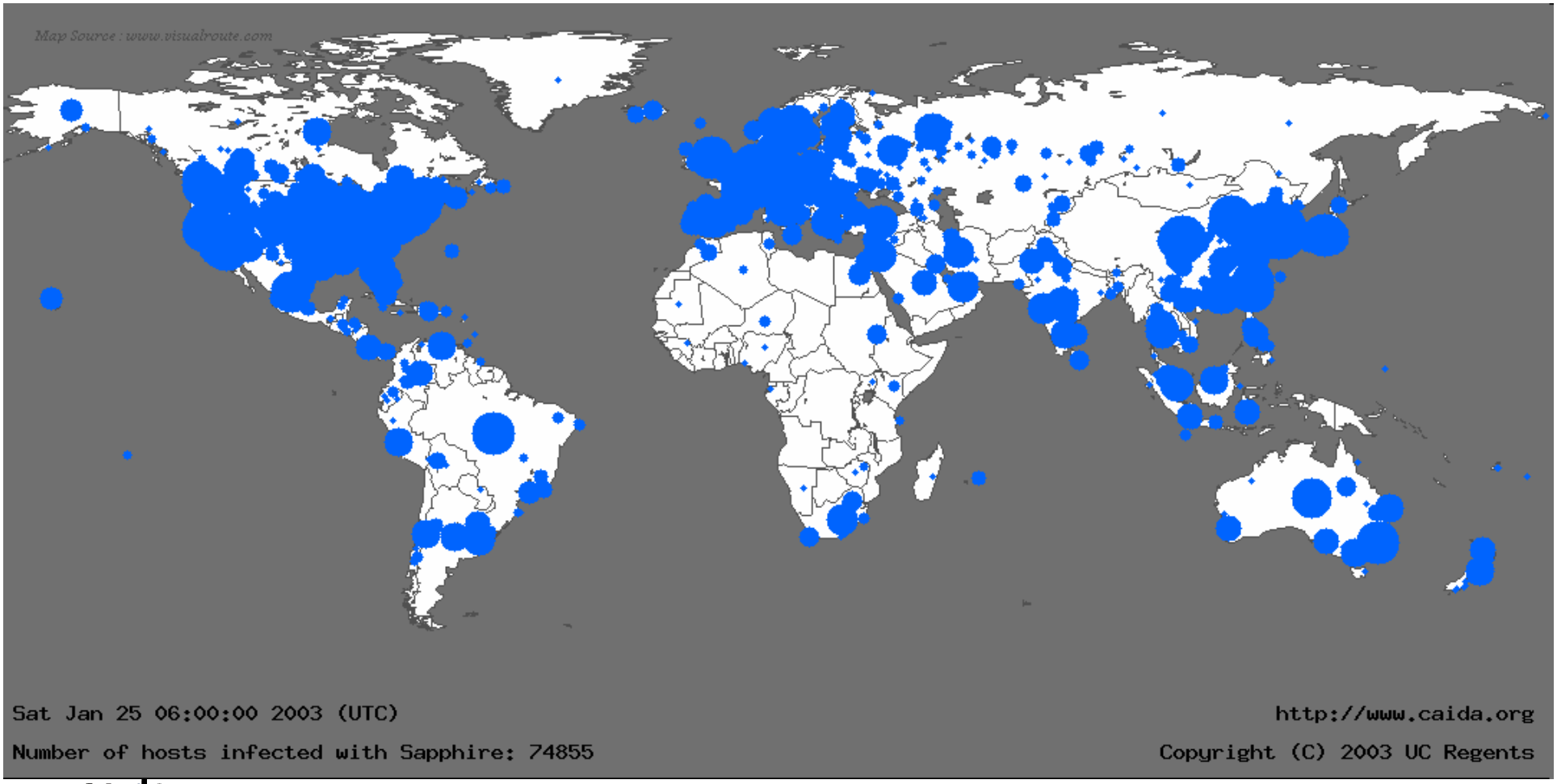
Sapphire growth

- 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



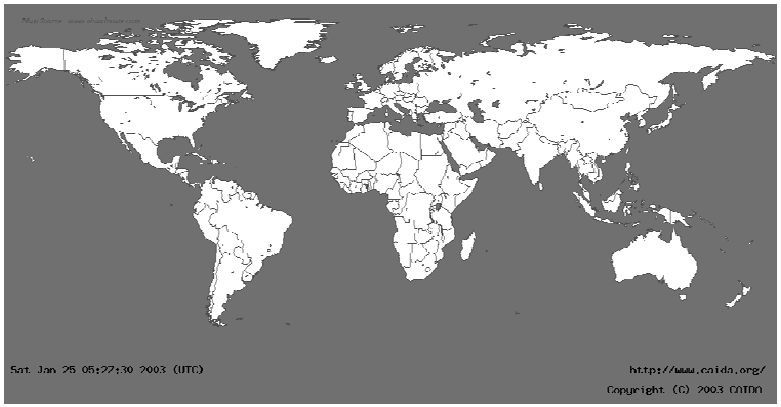
- 90% of Internet scanned in <10mins
  - Infected ~100k hosts (conservative due to PRNG errors)

Sapphire Animation

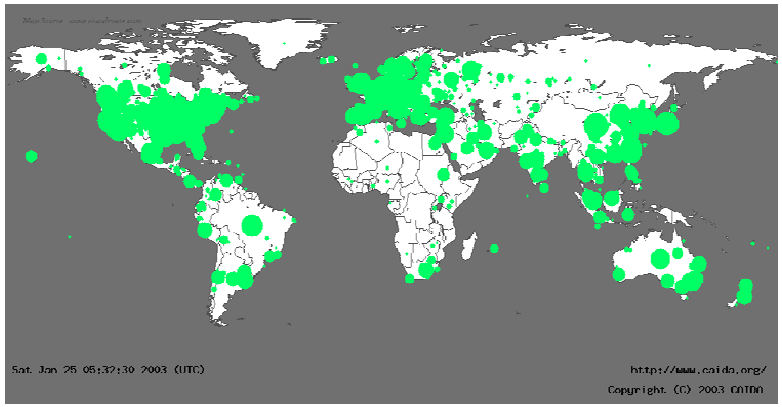


Internet Worm Attacks: Sapphire

(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 (could be worse…)

Witty Worm Background

March 19, 2004

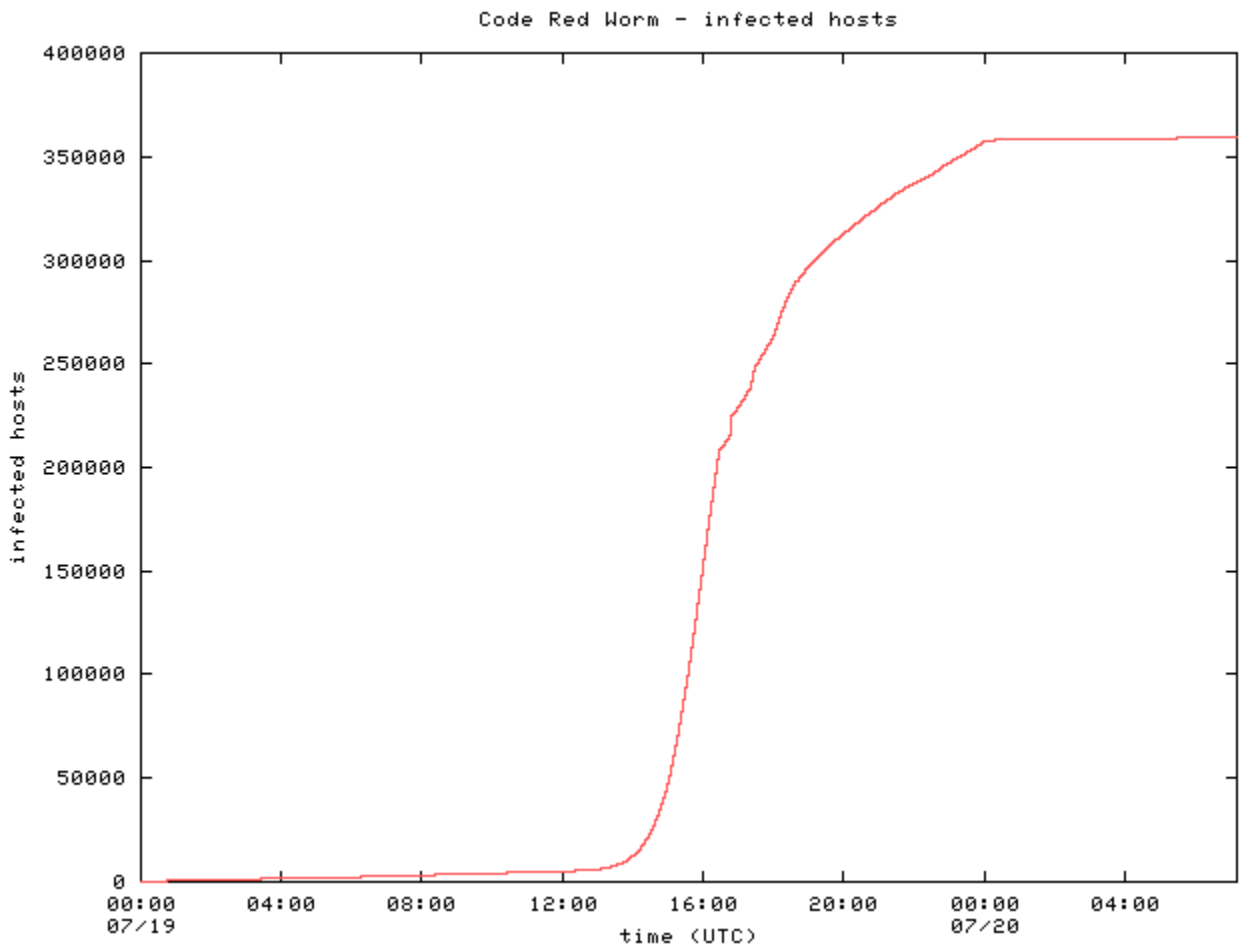
- ISS Vulnerability
  - A buffer overflow in a PAM (Protocol Analysis Module) in a Internet Security Systems firewall products
    - Version 3.6.16 of iss-pam1.dll
  - Analyzes ICQ traffic (inbound port 4000)
  - Discovered by eEye on March 8, 2004
  - Jointly announced March 18,2004 when “patch” available
    - Upgrade to the next version at customer cost…
- By far the closest to a zero-day exploit
  - Instead of 2-4 weeks after bug release, Witty appeared after *36 hours*

Witty Worm Structure

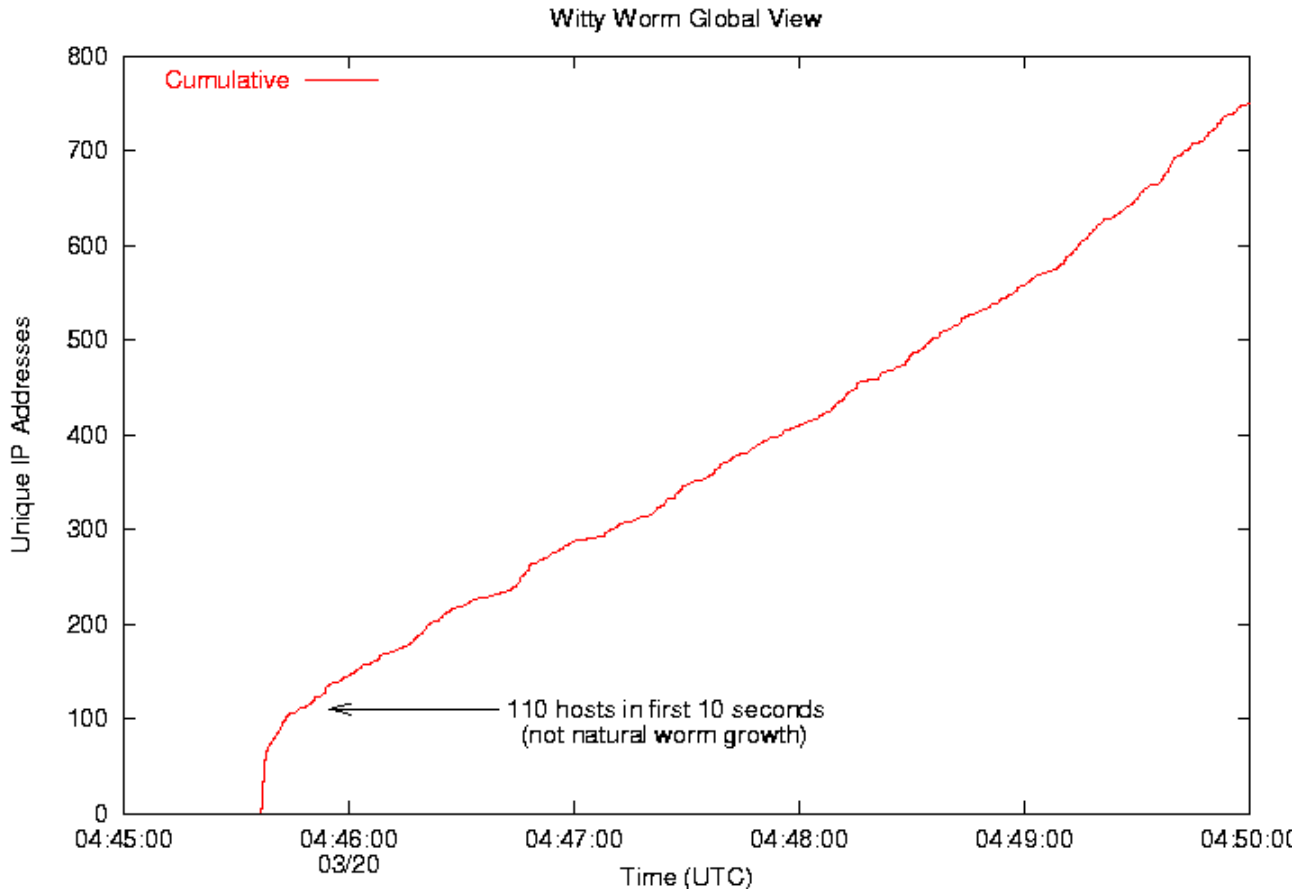
March 19, 2004

- Infects a host running an ISS firewall product
- Sends 20,000 UDP packets as quickly as possible:
  - to random source IP addresses
  - to random destination port
  - with random size between 796 and 1307 bytes
- Damage Victim:
  - select random physical device
  - seek to random point on that device
  - attempt to write over 65k of data with a copy of the beginning of the vulnerable dll
- Repeat until machine is rebooted or machine crashes irreparably

Typical (Code-Red) Host Infection Rate



Early Growth of Witty (5 minutes)

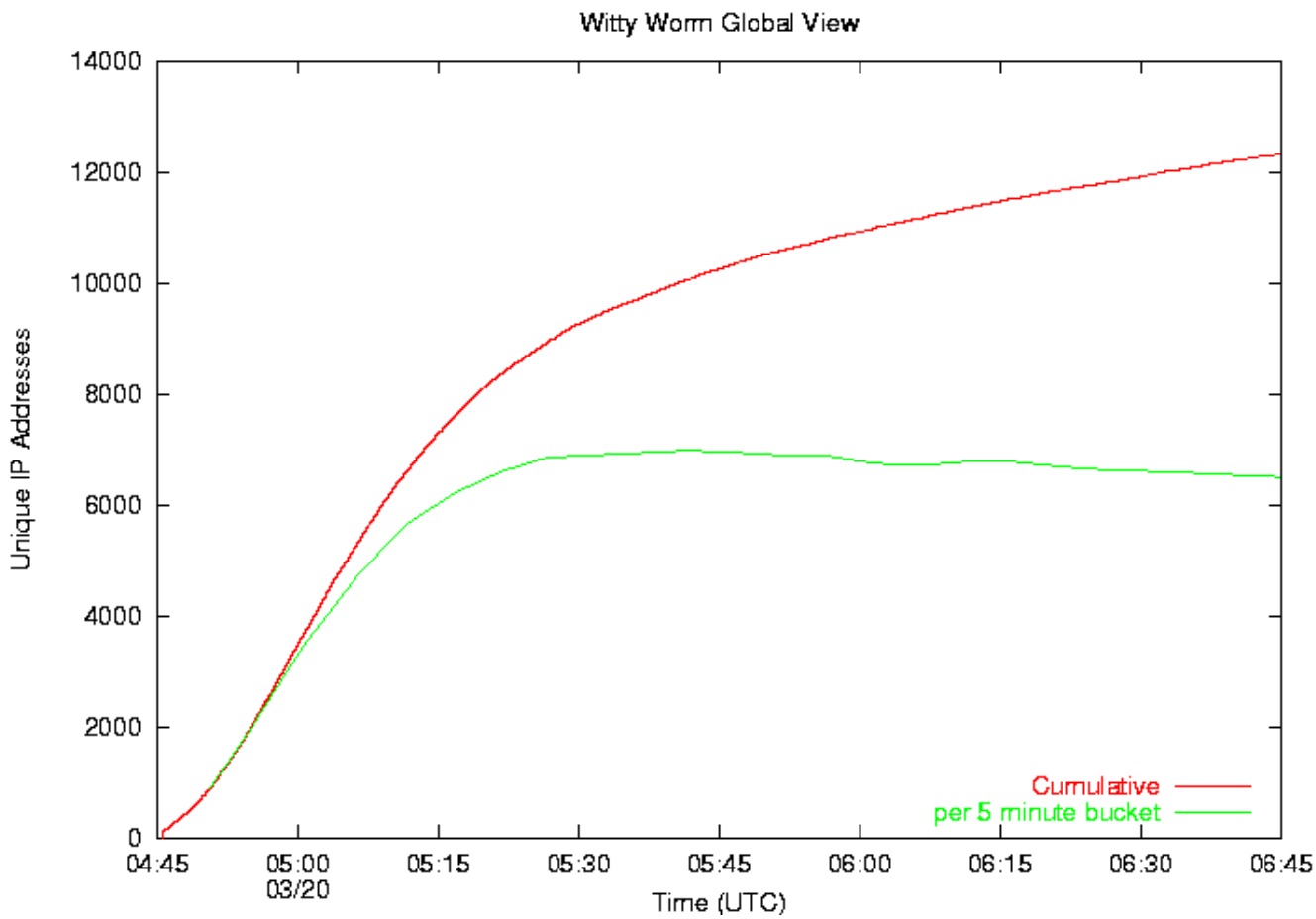


Witty Worm Spread

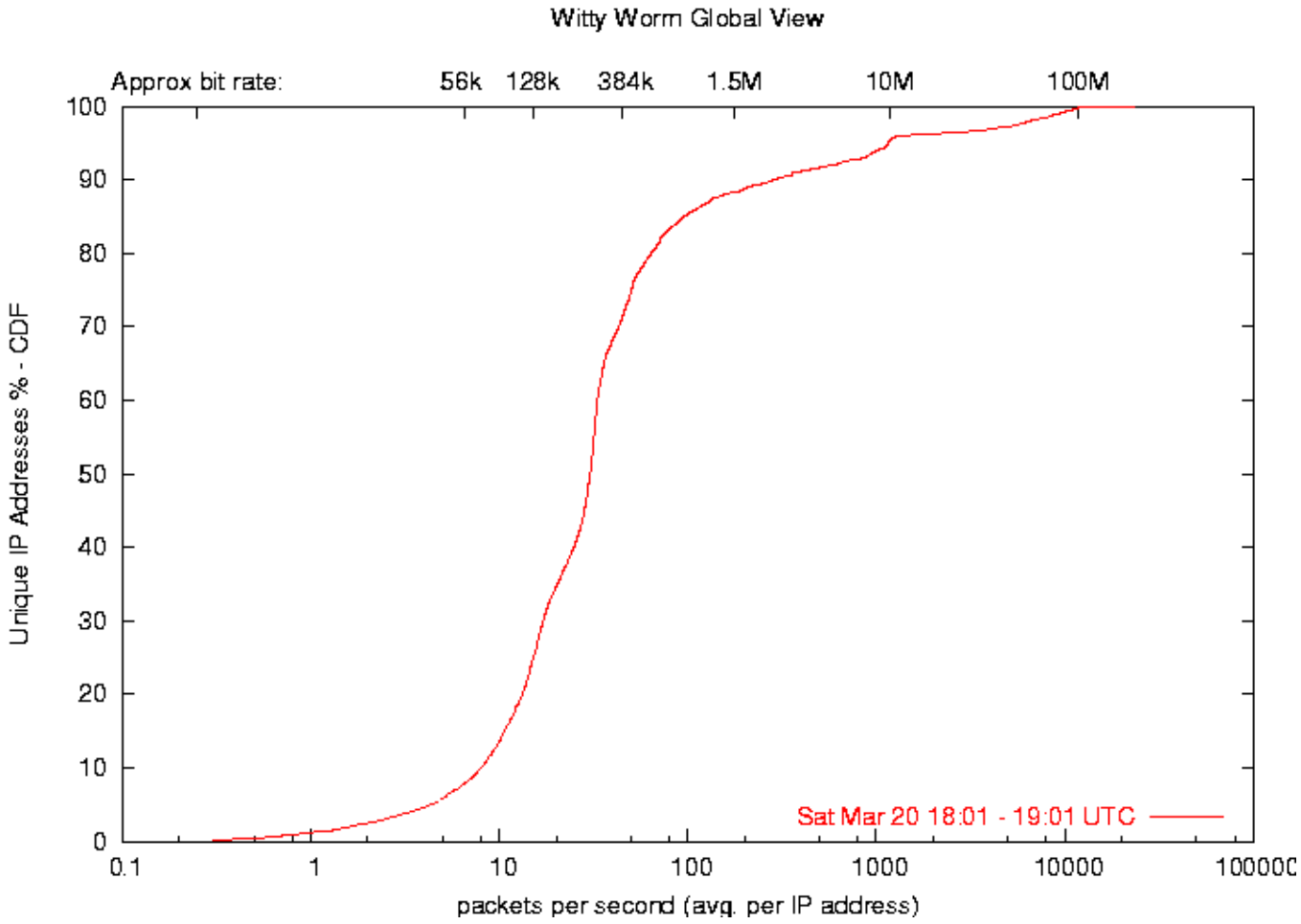
March 19, 2004

- Sharp rise via initial coordinated activity
- Peaked after approximately 45 minutes
  - Approximately 30 minutes later than the fastest worm we've seen so far (SQL Slammer)
  - Still far faster than any human response
  - At peak, Witty generated:
    - 90 GB/sec of network traffic
    - 11 million packets per second

Early Growth of Witty (2 hours)



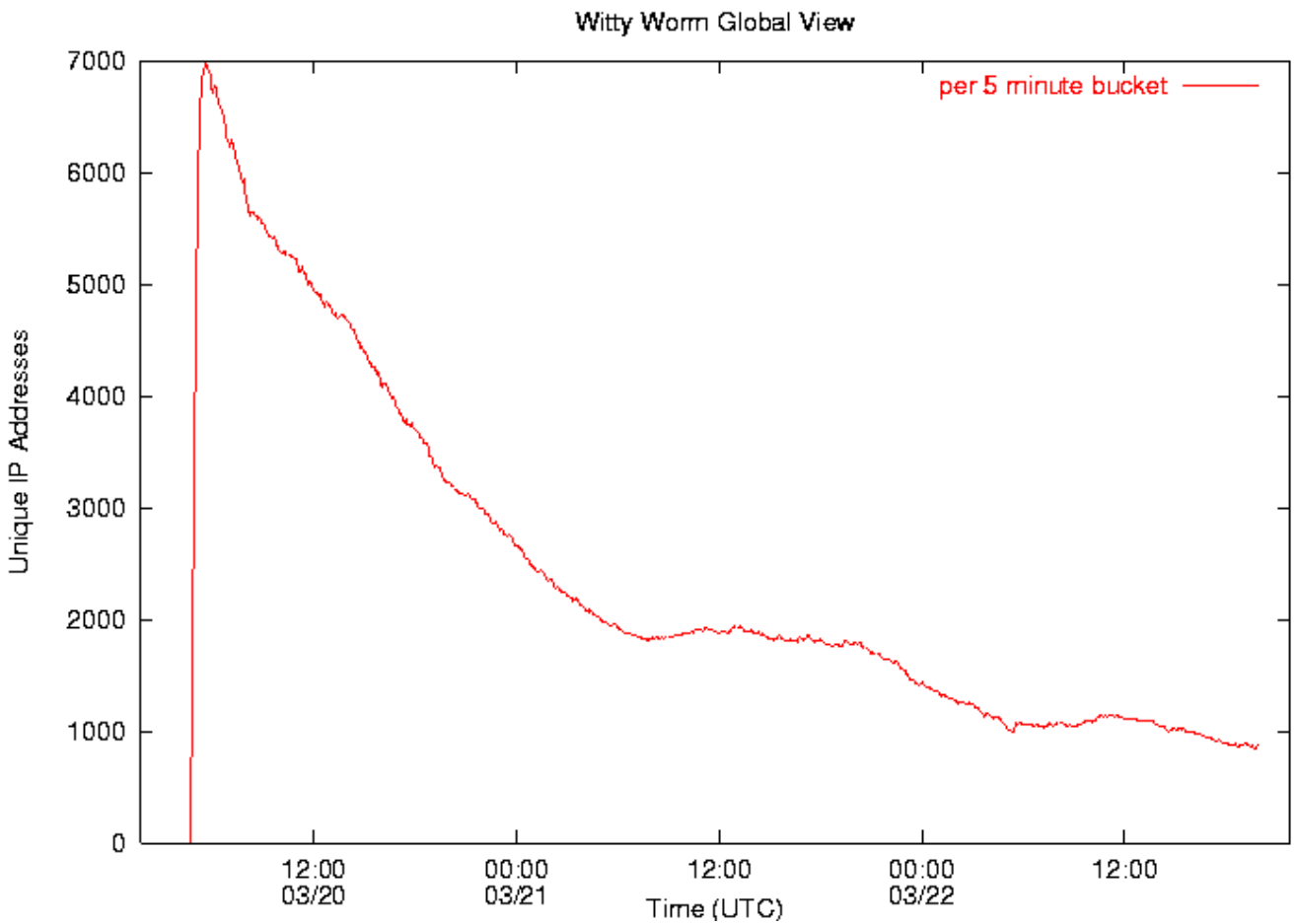
Witty Scan Rate



Witty Worm Scan Rate

- Like the earlier SQL Slammer worm, Witty hosts send UDP packets at line rate
- Wide variation in the scan rate of infected machines
  - From <1 pps to ~10,000 pps
  - From <14 kbps to >100 Mbps
  - 53% of hosts in range 128 – 512 kpbs (15-60 pps)
    - Cablemodem and DSL users
  - Overall average: 3 Mbps (357pps)
  - Average at peak scanning rate: 8 Mbps (970 pps)
  - Maximum scan rate: 23,500 pps sustained for more than an hour

Early Growth of Witty (3 days)

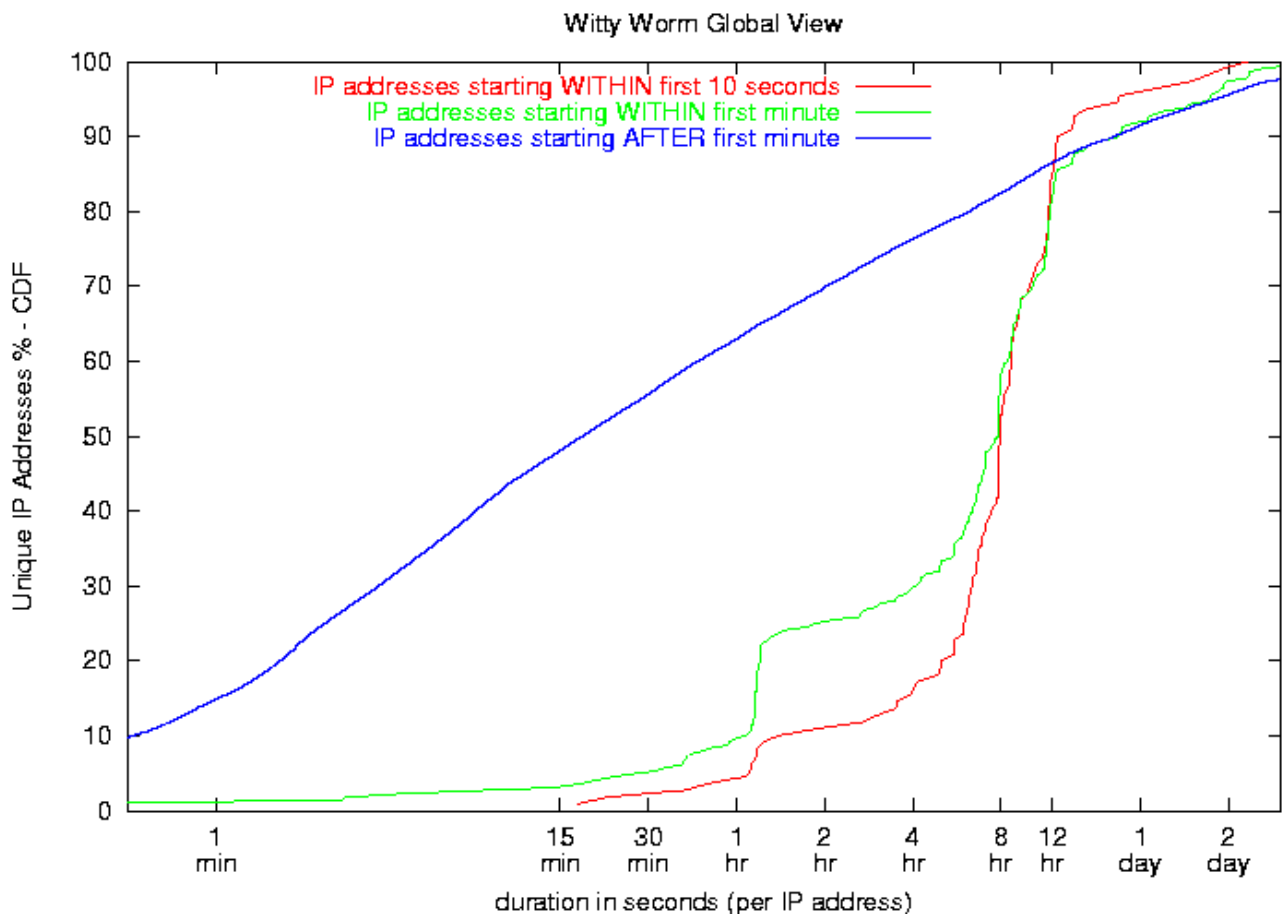


Witty Worm Decay

March 19, 2004

- 75% of hosts deactivated within 24 hours
  - Unprecedented response
    - Better coordination from network security and IT personnel
    - Majority of the impact results from destructive worm payload damaging to infected machines
  - Dynamic addressing limits the duration of many attacks
    - User perceptions (“my Internet is broken”, “my computer is slow”) can cause reboot and can result in a new IP address
    - NAT use also a significant factor (aggregates victims, rewrites packet headers
  - Traffic filtering artificially limits our view of infection duration (but we do accurately record the interval for which an infected machine is dangerous to others)

Witty Infection Durations



Witty Worm Victims

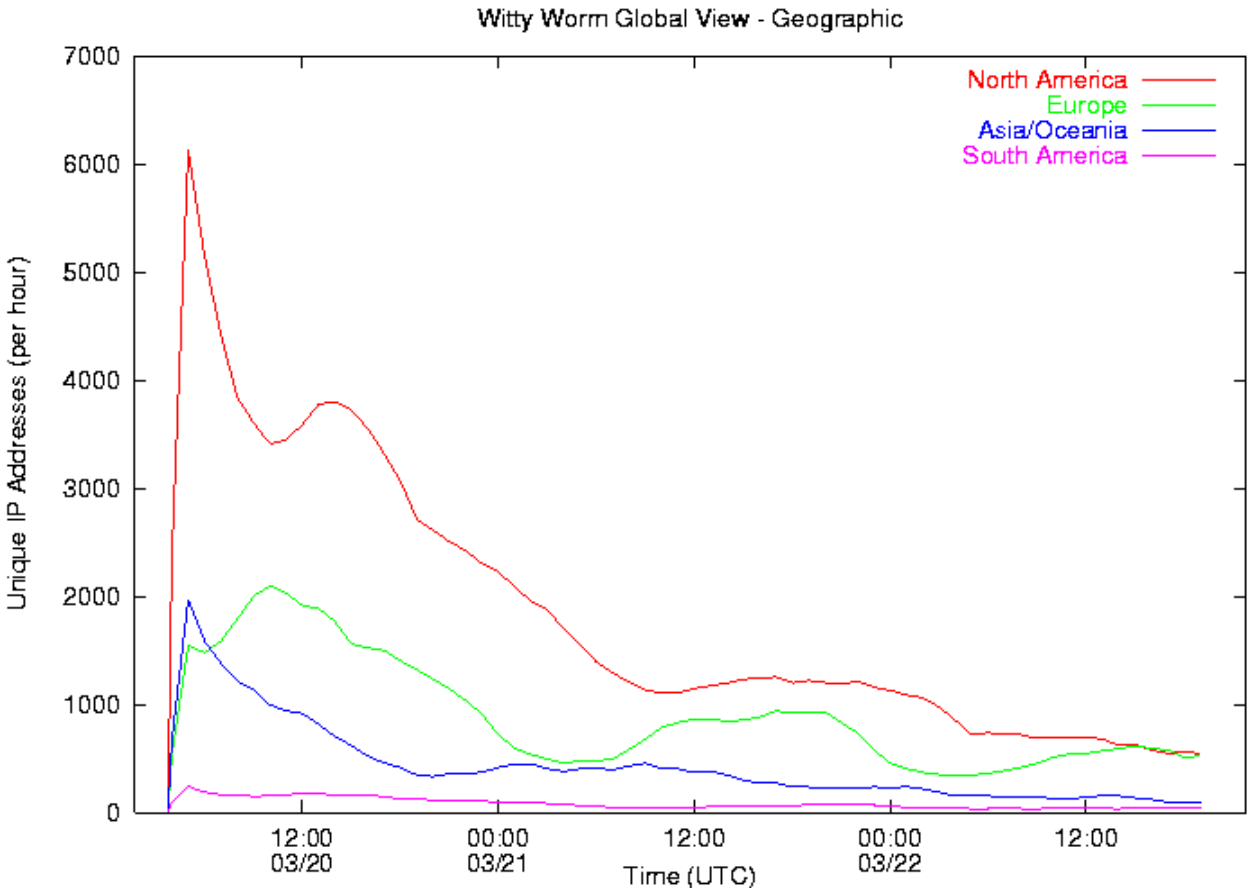
- Consistent with past worms:
  - Globally distributed
  - Majority high-bandwidth home/small business users
- Unique victim characteristics
  - 100% taking proactive security measures
  - Infected via software they ran purposefully

Witty Worm Victims

Country	Percent
United States	26.28
United Kingdom	7.27
Canada	3.46
China	3.36
France	2.94
Japan	2.17
Australia	1.83
Germany	1.82
Netherlands	1.36
Korea	1.21

TLD	Percent
com	33
net	20
no-DNS	15
fr	3
ca	2
jp	2
au	2
edu	1
nl	1
ar	1

Geographic Spread of Witty



Conclusions (1)

- Witty incorporates a number of novel and disturbing features:
  - Next day exploit for publicized bug
  - Wide-scale deployment
  - Successful exploit of small population (no more security through obscurity)
  - Future worms will continue to emulate botnets – increasing levels of stealth and flexibility
  - Infected a ***security*** product

Conclusions (2)

- Witty demonstrates conclusively that the patch model of networked device security has failed
  - You can't encourage people to sign on to the 'net with one click and then also expect them to be security experts
  - Running commercial firewall software at their own expense is the gold standard for end user behavior
    - Recognition that security is important
    - Recognition that they can't do it themselves

Conclusions (3)

- End-user behavior cannot solve current software security problems
- End-user behavior cannot effectively mitigate current software security problems
- We must:
  - Actively address prevention of software vulnerabilities
  - Turn our attention to developing large-scale, robust, reliable infrastructure that can mitigate current security problems without end-user intervention

Acknowledgements

- Technical support of Network Telescope at UCSD:
  - Brian Kantor, Jim Madden, and Pat Wilson
- Feedback on Witty research:
  - Cisco PSIRT Team, Wendy Garvin, Team Cymru, Nicholas Weaver, Vern Paxson, Mike Gannis, and Stefan Savage
- Support for this work was provided by: Cisco Systems, NSF, DARPA, DHS, and CAIDA members

More Information

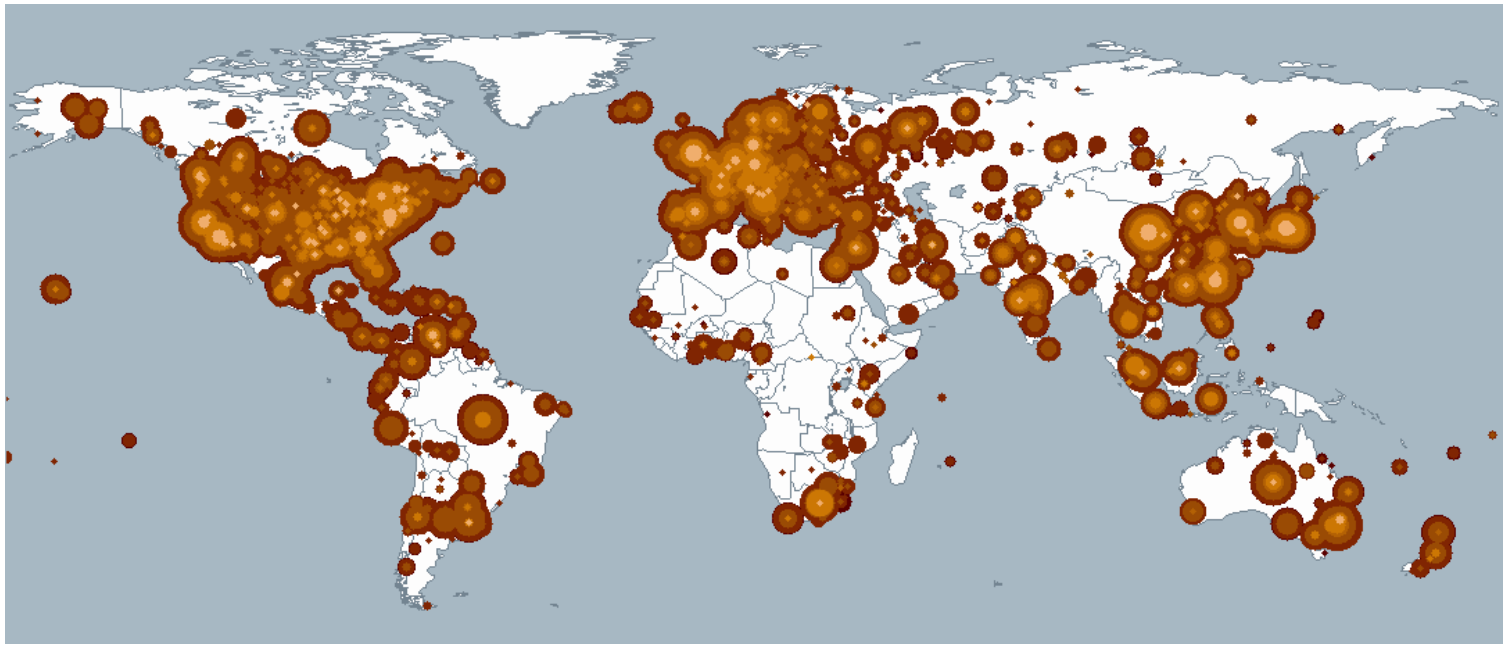
- Witty:
  - CAIDA report: http://www.caida.org/analysis/security/witty/
  - eEye vulnerability release: http://www.eeye.com/html/Research/Advisories/AD20040318.html
  - ISS vulnerability release: http://xforce.iss.net/xforce/alerts/id/166
  - Witty code analysis: http://www.lurhq.com/witty.html
  - Kostya Kortchinsky's Witty code disassembly (not CAIDA work): http://www.caida.org/analysis/security/witty/BlackIceWorm.html
- Other worm research:
  - Staniford, Paxson, Weaver: How to 0wn the Internet in Your Spare Time http://www.icir.org/vern/papers/cdc-usenix-sec02/
  - Moore, Shannon, Voelker, Savage: Internet Quarantine: Requirements for Containing Self-Propagating Code http://www.cs.ucsd.edu/users/savage/papers/Infocom03.pdf
  - CAIDA: The Spread of the Slammer Worm: http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml
  - Moore, Paxson, Savage, Shannon, Staniford, Weaver: http://www.caida.org/outreach/papers/2003/sapphire2/

Internet Worm Attacks: Code-Red

(July 19, 2001)

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

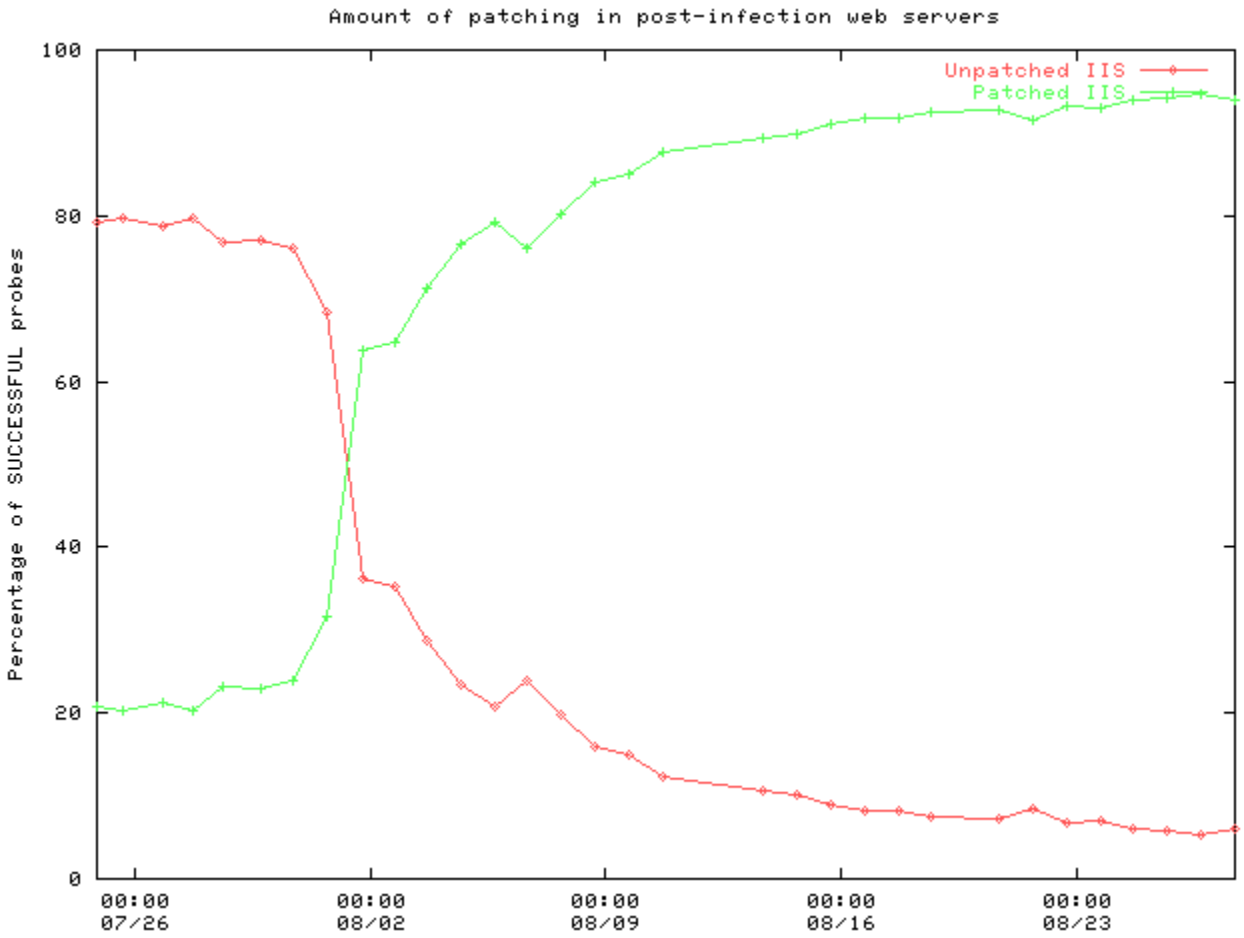
- CodeRed was programmed to deactivate on July 20th and begin spreading again on August 1st
- By July 30th and 31st, more news coverage than you can shake a stick at:
  - FBI/NIPC press release
  - Local ABC, CBS, NBC, FOX, WB, UPN coverage in many areas
  - National coverage on ABC, CBS, NBC, CNN
  - 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

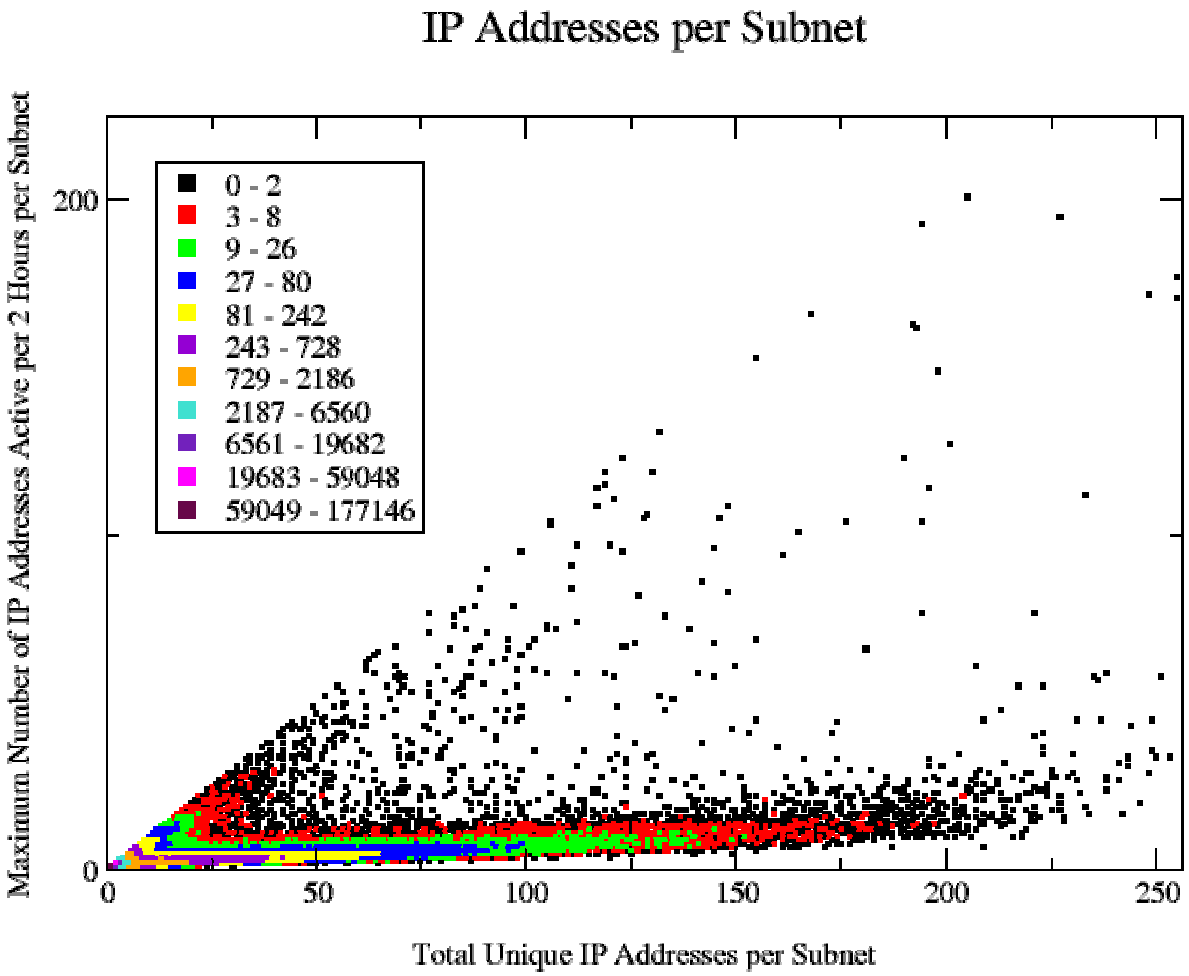
Patching Rate



Dynamic IP Addresses

- How can we tell how when an IP address represents an infected **computer**?
- Resurgence of CodeRed: 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.
- Culminating in **Sapphire/Slammer** worm (Winter 2003)

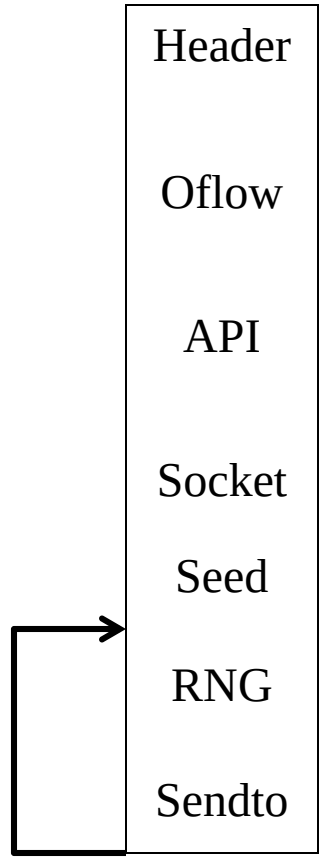
Inside the Sapphire/Slammer Worm

- 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

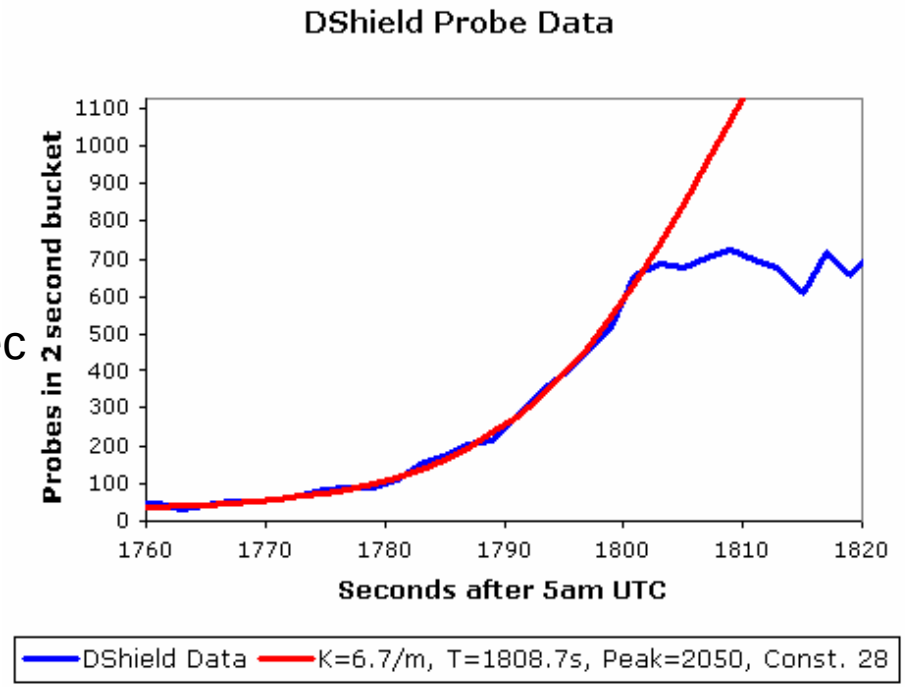
  Code borrowed from published exploit

  - 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)



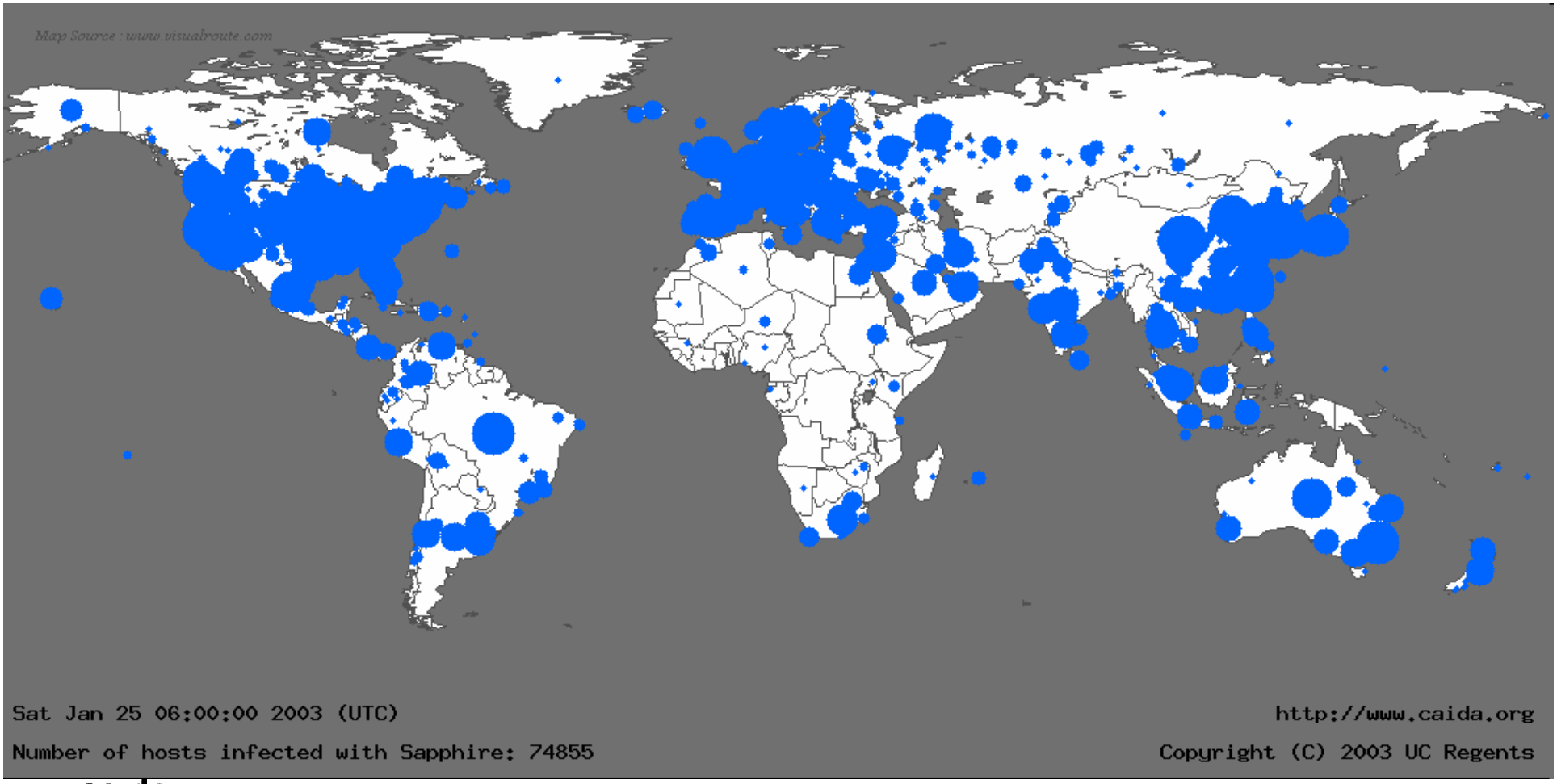
Sapphire growth

- 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



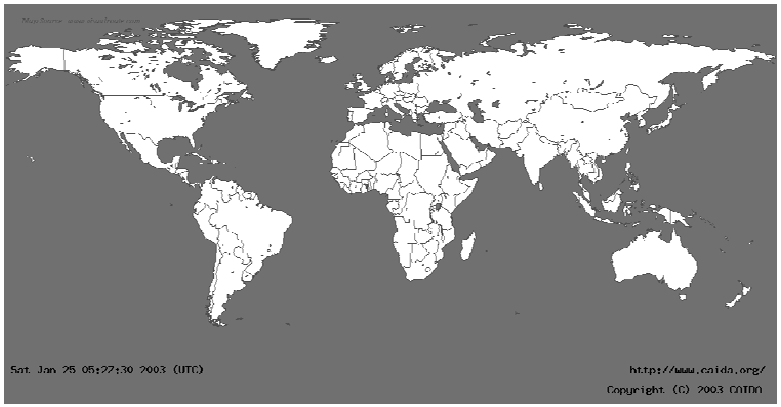
- 90% of Internet scanned in <10mins
  - Infected ~100k hosts (conservative due to PRNG errors)

Sapphire Animation

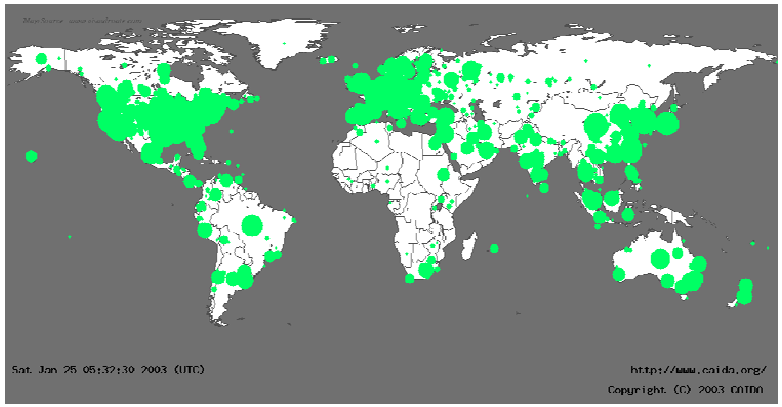


Internet Worm Attacks: Sapphire

(aka SQL Slammer) – January 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; most worms have had benign payload (not witty)
    - 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?

Open Research Questions for Measurement

- Denial-of-Service Attacks:
  - how much actual damage to victim
  - overall trends
- Internet Worms:
  - victim classification
  - early detection, automated filters
- Telescope Design:
  - distributed telescopes
  - making monitors which are robust under attack situations (millions of flows per second)

Acknowledgements

- Collaborators:
  - UCSD-CSE: Jeffrey Brown
  - ICSI/LBNL: Vern Paxson
  - Silicon Defense: Stuart Staniford, Nicholas Weaver
  - UCB-EECS: Nicholas Weaver

- Data Providers:
  - UCSD: Brian Kantor, Pat Wilson
  - UCB/LBNL: Vern Paxson
  - UWISC: Dave Plonka
  - Dshield: Johannes Ullrich
  - Compaq/WRL: Jeff Mogul
  - DOD CERT: Donald LaDieu, Matthew Swaar

- Funding:
  - Cisco University Research Program (URP)
  - DARPA
  - NSF
  - CAIDA Members

Related 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 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/