

# The Threats of Internet Worms

1

## What is a Worm?

- ❑ Self-replicating/self-propagating code.
- ❑ Spreads across a network by exploiting flaws in open services.
  - As opposed to viruses, which require user action to quicken/spread.
- ❑ Not new — Morris Worm, Nov. 1988
  - 6-10% of all Internet hosts infected
- ❑ Many more since, but for 13 years none on that scale, until ...

2

## Code Red

- ❑ Initial version released July 13, 2001.
- ❑ Exploited known bug in Microsoft IIS Web servers.
- ❑ Payload: Web site defacement
  - HELLO! Welcome to <http://www.worm.com!>
  - Hacked By Chinese!
  - Only done if language setting = English

3

## Code Red of July 13, con't

- ❑ 1st through 20th of each month: spread.
- ❑ 20th through end of each month: attack.
  - Flooding attack against 198.137.240.91 ...
  - ... i.e., [www.whitehouse.gov](http://www.whitehouse.gov)
- ❑ Spread: via random scanning of 32-bit IP address space.
  
- ❑ But: failure to seed random number generator  
⇒ linear growth.

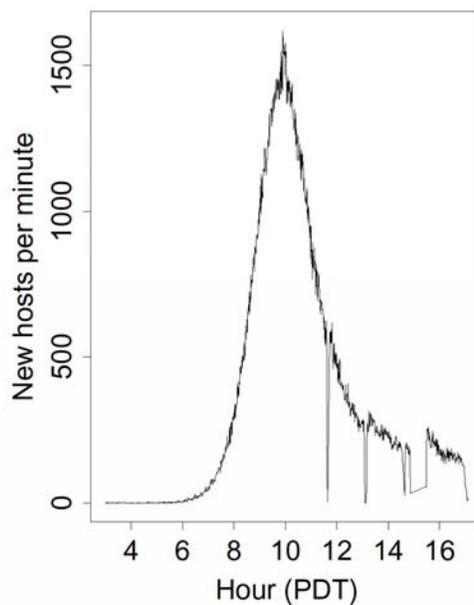
4

## Code Red, con't

- ❑ Revision released July 19, 2001.
- ❑ White House responds to threat of flooding attack by changing the address of `www.whitehouse.gov`
- ❑ Causes Code Red to die for date  $\geq$  20th of the month.
  
- ❑ But: This time random number generator correctly seeded. **Bingo!**

5

Growth of Code Red Worm



6

## Measuring Internet-Scale Activity: Network Telescopes

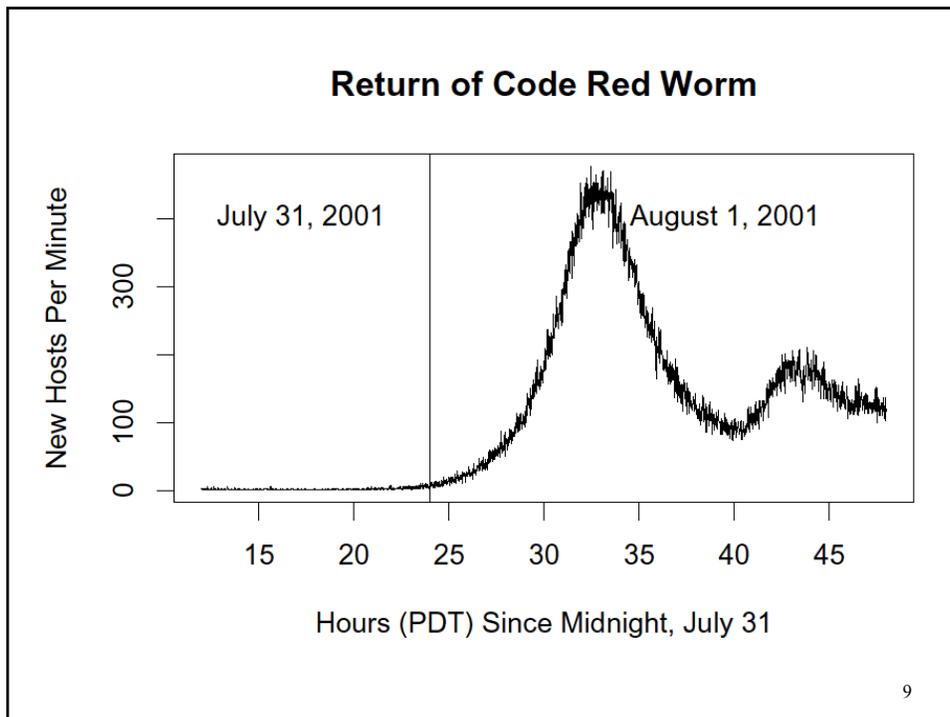
- Idea: Monitor a cross-section of Internet address space to measure network traffic involving wide range of addresses
  - "Backscatter" from DOS floods
  - Attackers probing blindly
  - Random scanning from worms

7

## Spread of Code Red

- Network telescopes estimate of # infected hosts: 360K.
- Note: larger the vulnerable population, faster the worm spreads.
  
- That night ( $\Rightarrow$  20th), worm dies ...
- ... except for hosts with inaccurate clocks!
- It just takes one of these to restart the worm on August 1st ...

8



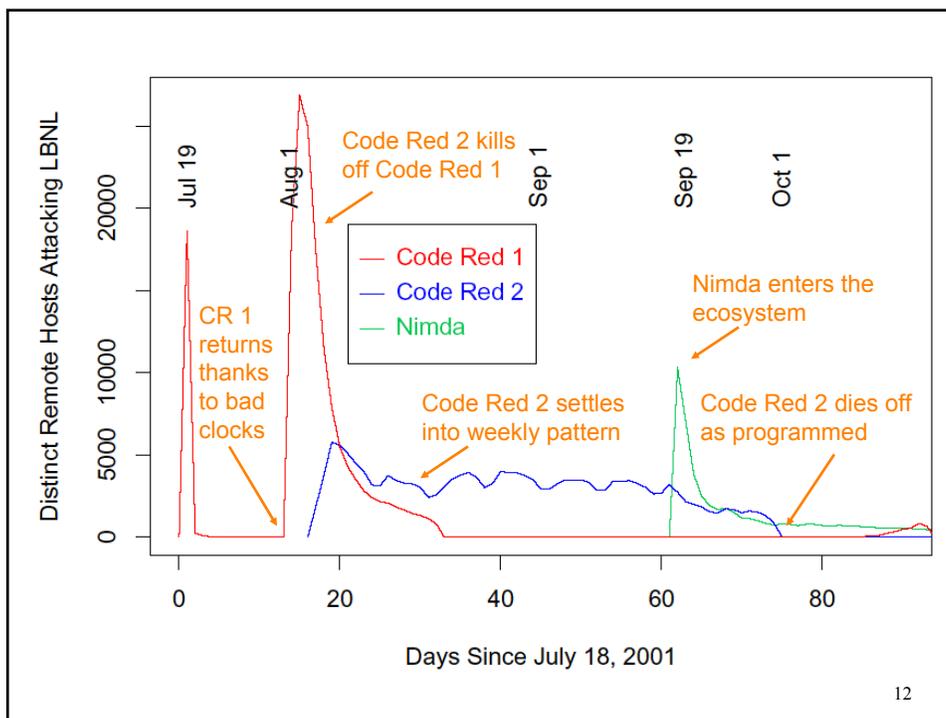
## Striving for Greater Virulence: Code Red 2

- ❑ Released August 4, 2001.
- ❑ Comment in code: "Code Red 2."
  - But in fact completely different code base.
- ❑ Payload: a root backdoor, resilient to reboots.
- ❑ Bug: crashes NT, only works on Windows 2000.
  
- ❑ Kills Code Red 1.
  
- ❑ Safety valve: programmed to die Oct 1, 2001.

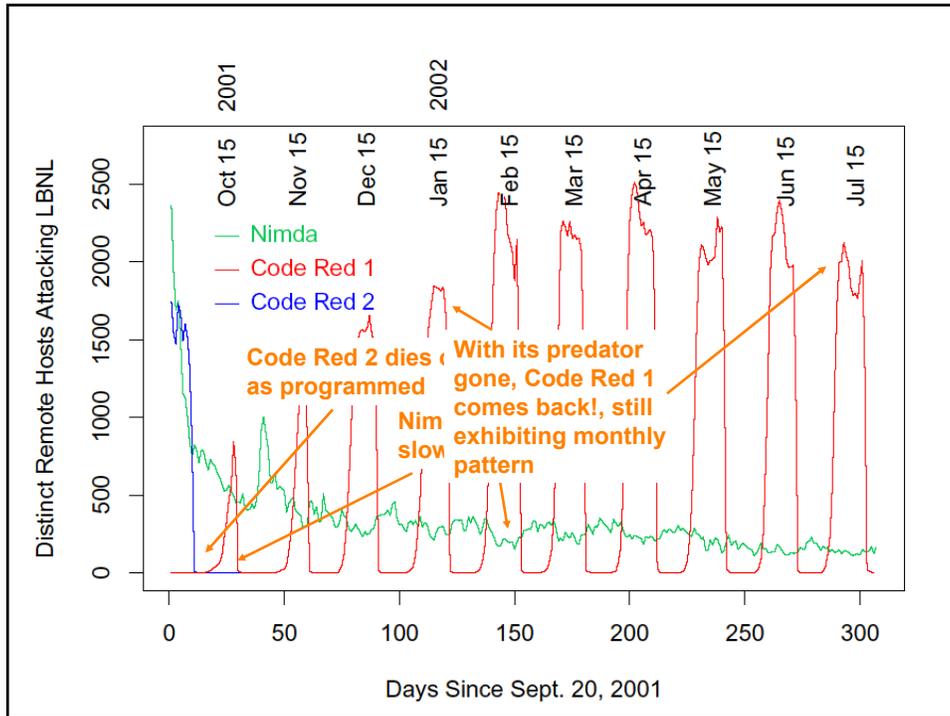
## Striving for Greater Virulence: Nimda

- ❑ Released September 18, 2001.
- ❑ Multi-mode spreading:
  - Attack IIS servers via infected clients
  - Email itself to address book as a virus
  - Copy itself across open network shares
  - Modifying Web pages on infected servers w/ client exploit
  - Scanning for Code Red II backdoors (!)
- ❑ Worms form an ecosystem!
- ❑ Leaped across firewalls.

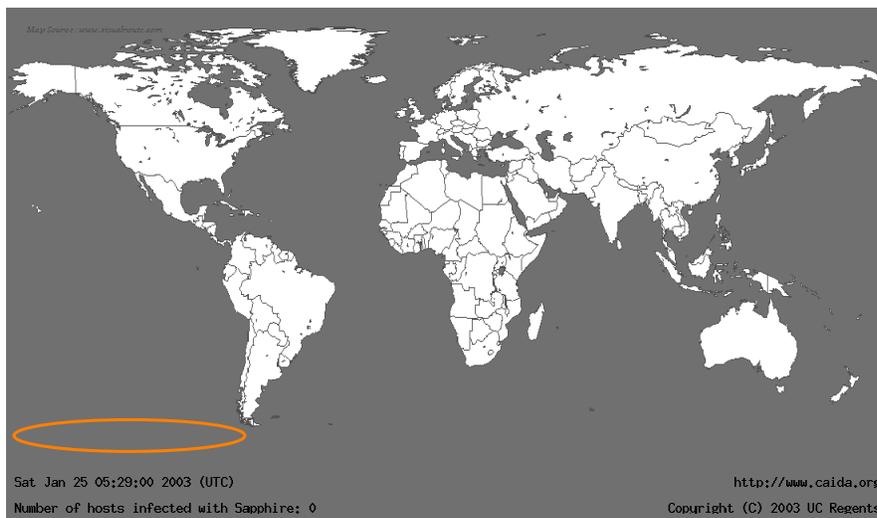
11



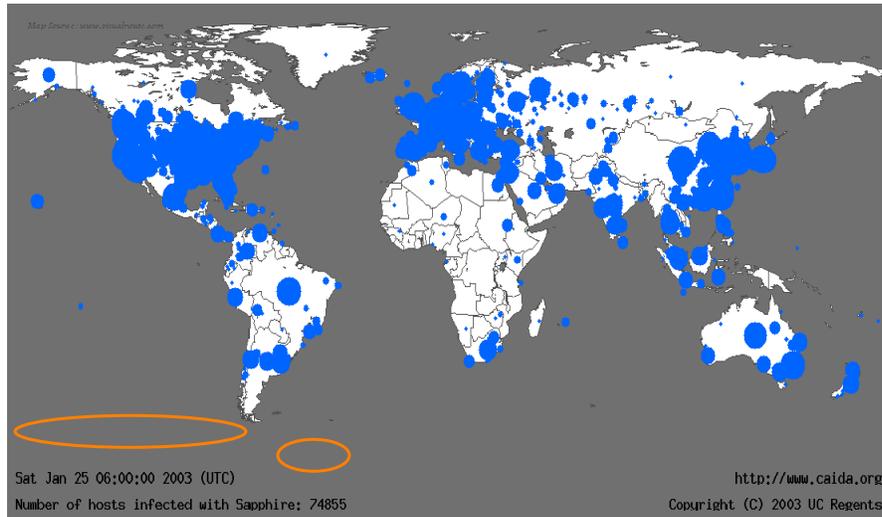
12



## Life Just Before Slammer



## Life Just After Slammer



15

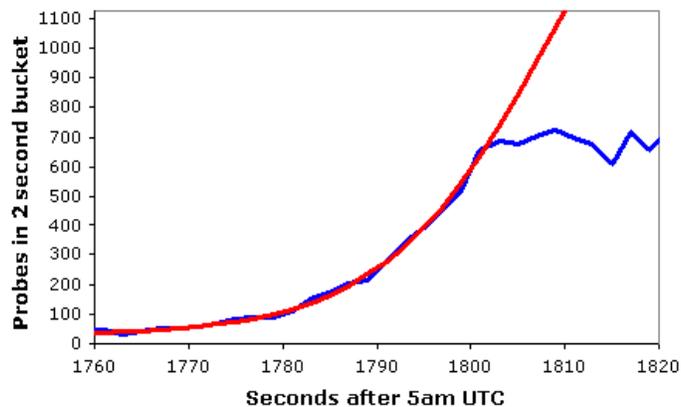
## A Lesson in Economy

- ❑ Slammer exploited a **connectionless UDP** service, rather than connection-oriented TCP.
- ❑ Entire worm fit in a single packet!
- ❑ When scanning, worm could **"fire and forget"**.
  
- ❑ Worm infected 75,000+ hosts in **10 minutes** (despite broken random number generator).
  - At its peak, **doubled** every **8.5 seconds**
- ❑ Progress limited by the Internet's bandwidth capacity!

16

## Slammer's Bandwidth-Limited Growth

DShield Probe Data



— DShield Data —  $K=6.7/m$ ,  $T=1808.7s$ , Peak=2050, Const. 28

17

## Blaster

- ❑ Released August 11, 2003.
- ❑ Exploits flaw in RPC service ubiquitous across Windows.
- ❑ Payload: attack Microsoft Windows Update.
- ❑ Despite flawed scanning and secondary infection strategy, rapidly propagates to (at least) 100K's of hosts.
- ❑ Actually, bulk of infections are really Nachia, a Blaster counter-worm.
- ❑ Key paradigm shift: firewalls don't help.

18

## What if Spreading Were Well-Designed?

- ❑ Observation:
  - Much of a worm's scanning is redundant.
- ❑ Ideas:
  - Accelerate later phase: coordinated scanning
  - Accelerate initial phase: use precomputed hit-list
- ❑ Greatly accelerates worm.

19

## Defenses

- ❑ Detect via **honey farms**: Collections of "honey pots" fed by a network telescope.
  - Any outbound connection from honey farm = worm.
  - Distill signature from inbound/outbound traffic.
- ❑ Thwart via scan suppressors: Network elements that block traffic from hosts that make failed connection attempts to too many other hosts.

20

## Defenses?

- ❑ Observation:  
Worms don't need to randomly scan
- ❑ Meta-server worm: Ask server for hosts to infect (e.g., Google for "powered by phpbb")
- ❑ Topological worm: Fuel the spread with local information from infected hosts (web server logs, email address books, config files, SSH "known hosts")
- ❑ No scanning signature; with rich inter-connection topology, potentially very fast.

21

## Defenses??

- ❑ Contagion worm: propagate parasitically along with normally initiated communication.
- ❑ E.g., using 2 exploits—Web browser & Web server—infect any vulnerable servers visited by browser, then any vulnerable browsers that come to those servers.
- ❑ E.g., using 1 BitTorrent exploit, glide along immense peer-to-peer network in days/hours.
- ❑ No unusual connection activity at all! :-)

22

## Some Cheery Thoughts

(Stefan Savage, UCSD/CCIED)

- ❑ Imagine the following species:
  - Poor genetic diversity; heavily inbred
  - Lives in “hot zone”; thriving ecosystem of infectious pathogens
  - Instantaneous transmission of disease
  - Immune response 10-1M times slower
  - Poor hygiene practices
- ❑ What would its long-term prognosis be?
- ❑ What if diseases were designed ...
  - Trivial to create a new disease
  - Highly profitable to do so

23

## Broader View of Defenses

- ❑ Prevention—make the monoculture hardier
  - Get the darn code right in the first place ...
    - ... or figure out what’s wrong with it and fix it
  - Lots of active research (static & dynamic methods)
  - Security reviews now taken seriously by industry
    - E.g., ~\$200M just to review Windows Server 2003
  - But very expensive
  - And very large Installed Base problem
- ❑ Prevention—diversify the monoculture
  - Via exploiting existing heterogeneity
  - Via creating artificial heterogeneity

24

## Broader View of Defenses, con't

- ❑ Prevention—keep vulnerabilities inaccessible
  - Cisco's Network Admission Control
    - Frisk hosts that try to connect, block if vulnerable
  - Microsoft's Shield ("Band-Aid")
    - Shim-layer blocks network traffic that fits known vulnerability (rather than known exploit)
- ❑ Detection—look for unusual repeated content
  - Can work on non-scanning worms
  - Key off many-to-many communication to avoid confusion w/ non-worm sources
  - EarlyBird, Autograph—distill signature
  - But: what about polymorphic worms?

25

## Once You Have A Live Worm, Then What?

- ❑ Containment
  - Use distilled signature to prevent further spread
- ❑ Would like to leverage detections by others
  - But how can you trust these?
  - What if it's an attacker lying to you to provoke a self-damaging response? (Or to hide a later actual attack)

26

## Once You Have A Live Worm, Then What?, con't

- ❑ Proof of infection
  - Idea: alerts come with a verifiable audit trail that demonstrates the exploit, ala' proof-carrying code
- ❑ Auto-patching
  - Techniques to derive (and test!) patches to fix vulnerabilities in real-time
    - (Excerpt from a review: "Not as crazy as it sounds")
- ❑ Auto-antiworm
  - Techniques to automatically derive a new worm from a propagating one, but with disinfectant payload
    - (This one, on the other hand, is as crazy as it sounds)

27

## Incidental Damage ... Today

- ❑ Today's worms have significant real-world impact:
  - Code Red disrupted routing
  - Slammer disrupted elections, ATMs, airline schedules, operations at an off-line nuclear power plant ...
  - Blaster possibly contributed to Great Blackout of Aug. 2003 ... ?
  - Plus major clean-up costs
- ❑ But today's worms are amateurish
  - Unimaginative payloads

28

## Where are the Nastier Worms??

- ❑ Botched propagation the norm
- ❑ Doesn't anyone read the literature?  
e.g., permutation scanning, flash worms,  
metaserver worms, topological, contagion
- ❑ Botched payloads the norm  
e.g., Flooding-attack fizzles
- ❑ Current worm authors are in it for kicks ...  
(... or testing) No arms race.

29

## Next-Generation Worm Authors

- ❑ Military.
- ❑ Crooks:
  - Denial-of-service, spamming for hire
  - "Access worms"
  - Very worrisome onset of blended threats
    - Worms + viruses + spamming + phishing + DOS-for-hire + botnets + spyware
- ❑ Money on the table ⇒ Arms race
  - (market price for spam proxies: 3-10¢/host/week)

30

## "Better" Payloads

- ❑ Wiping a disk costs \$550/\$2550\*
- ❑ "A well-designed version of Blaster could have infected 10M machines." (8M+ for sure!)
- ❑ The same service exploited by Blaster has other vulnerabilities ...
- ❑ Potentially a lot more \$\$\$: flashing BIOS, corrupting databases, spreadsheets ...
- ❑ Lower-bound estimate: \$50B if well-designed

31

## Attacks on Passive Monitoring

- ❑ Exploits for bugs in read-only analyzers!
- ❑ Suppose protocol analyzer has an error parsing unusual type of packet
  - E.g., tcpdump and malformed options
- ❑ Adversary crafts such a packet, overruns buffer, causes analyzer to execute arbitrary code

32

## Witty

- ❑ Released March 19, 2004.
- ❑ Single UDP packet exploits flaw in the passive analysis of Internet Security Systems products.
- ❑ "Bandwidth-limited" UDP worm ala' Slammer.
- ❑ Vulnerable pop. (12K) attained in 75 minutes.
- ❑ Payload: slowly corrupt random disk blocks.

33

## Witty, con't

- ❑ Flaw had been announced the previous day.
- ❑ Telescope analysis reveals:
  - Initial spread seeded via a hit-list.
  - In fact, targeted a U.S. military base.
  - Analysis also reveals "Patient Zero", a European retail ISP.
- ❑ Written by a Pro.

34