Worms

CS342, Handout 12

Tuesday, Oct. 17th , 2006
Wellesley College
Daniel Bilar

Goals today

- Appreciate the qualitative differences between Worm-Virus that autonomous mobility induces
- Evolution of Infection/Scan/Attack patterns

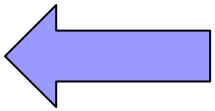
Worm: Definition

- A computer worm is a program that can run independently and can propagate a fully working version of itself to other machines

- Differences to computer viruses:
  - Worms are self-contained. Viruses attach themselves to a 'host' program.
  - Worms are self-activating. Viruses need user interaction to propagate (this is why I Love You and SoBig are viruses!)
- Worm on a system is also called worm node

Worm components

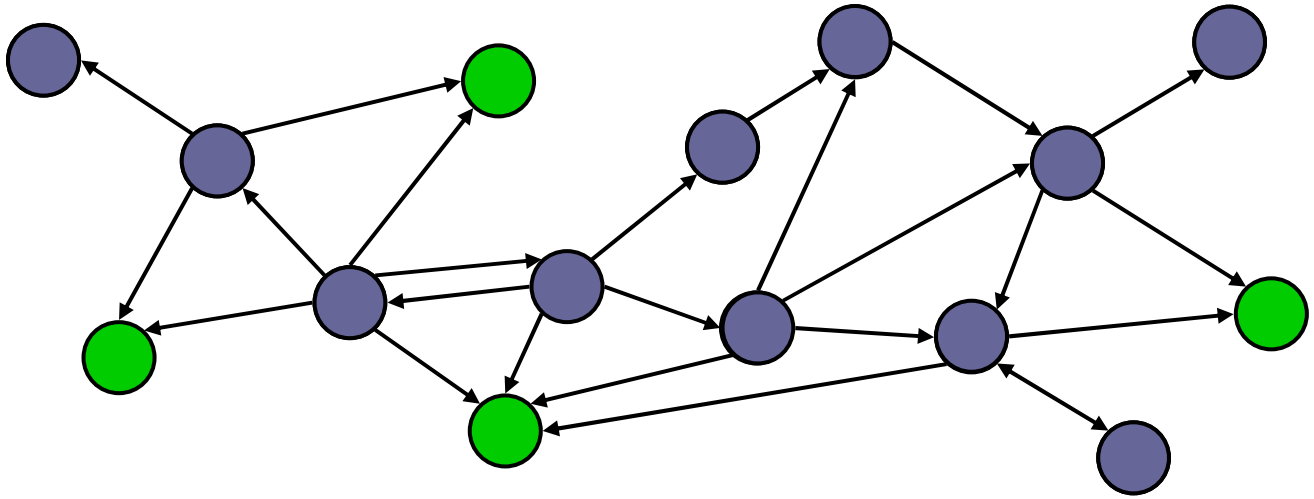
- Reconnaissance/Scanning
  - Discover susceptible hosts
- Attack
  - Penetrate the host
- Communication
  - Talk to other worm nodes
- Command
  - Control worm nodes
- Intelligence
  - Locate other worm nodes



Minimum worm components

The difference between virus and worms – independent mobility – seems small, but it has profound implications on the design of these components

Worms: Scanning

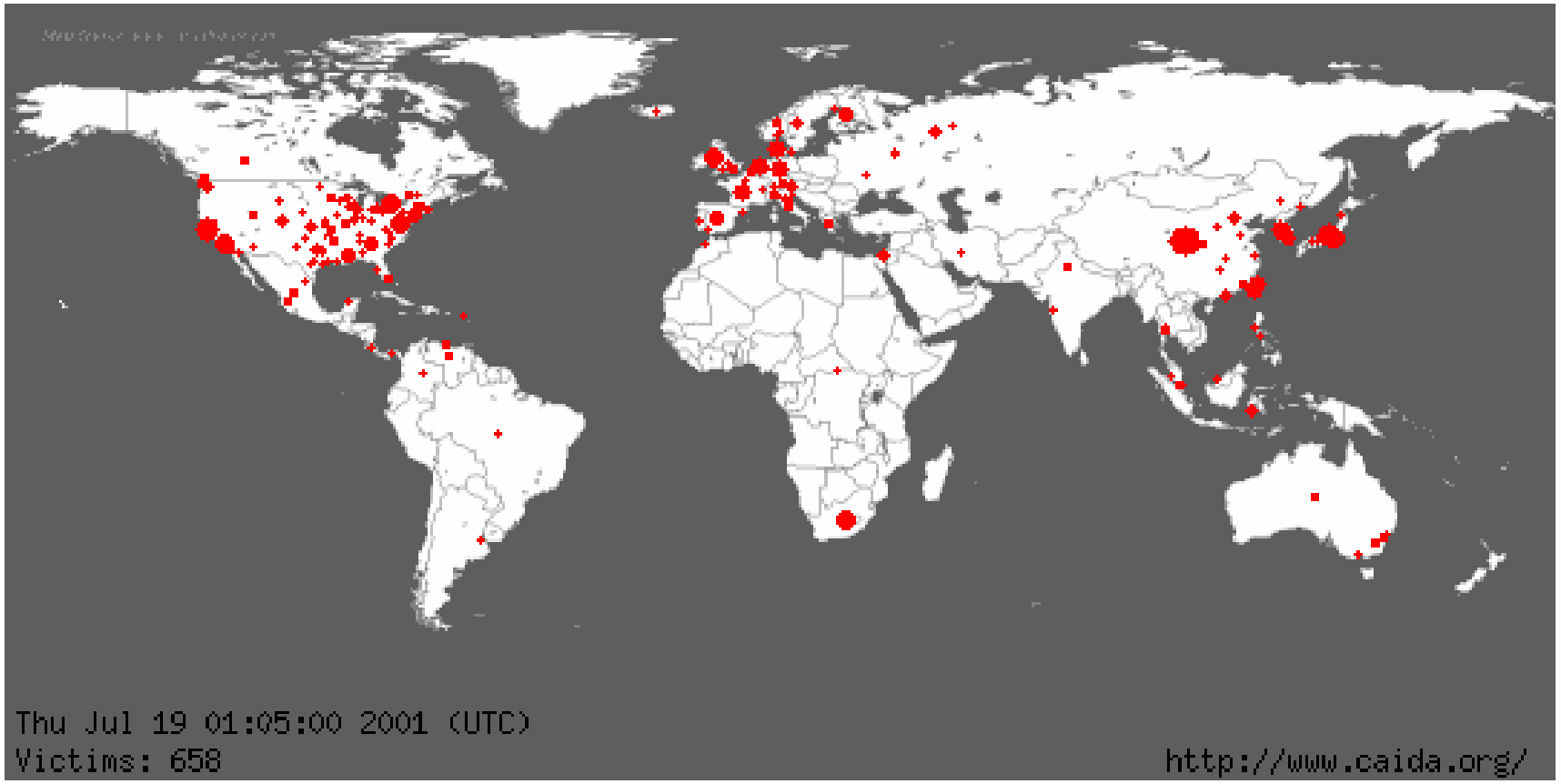


- Goal: Find new targets to attack
- Common techniques:
  - Topological: Use information on infected hosts, e.g. address book, .rhosts file, ...
  - Statistical: Scan 'random' IP addresses
- Avoid double infections!

Example modern worm: Code Red

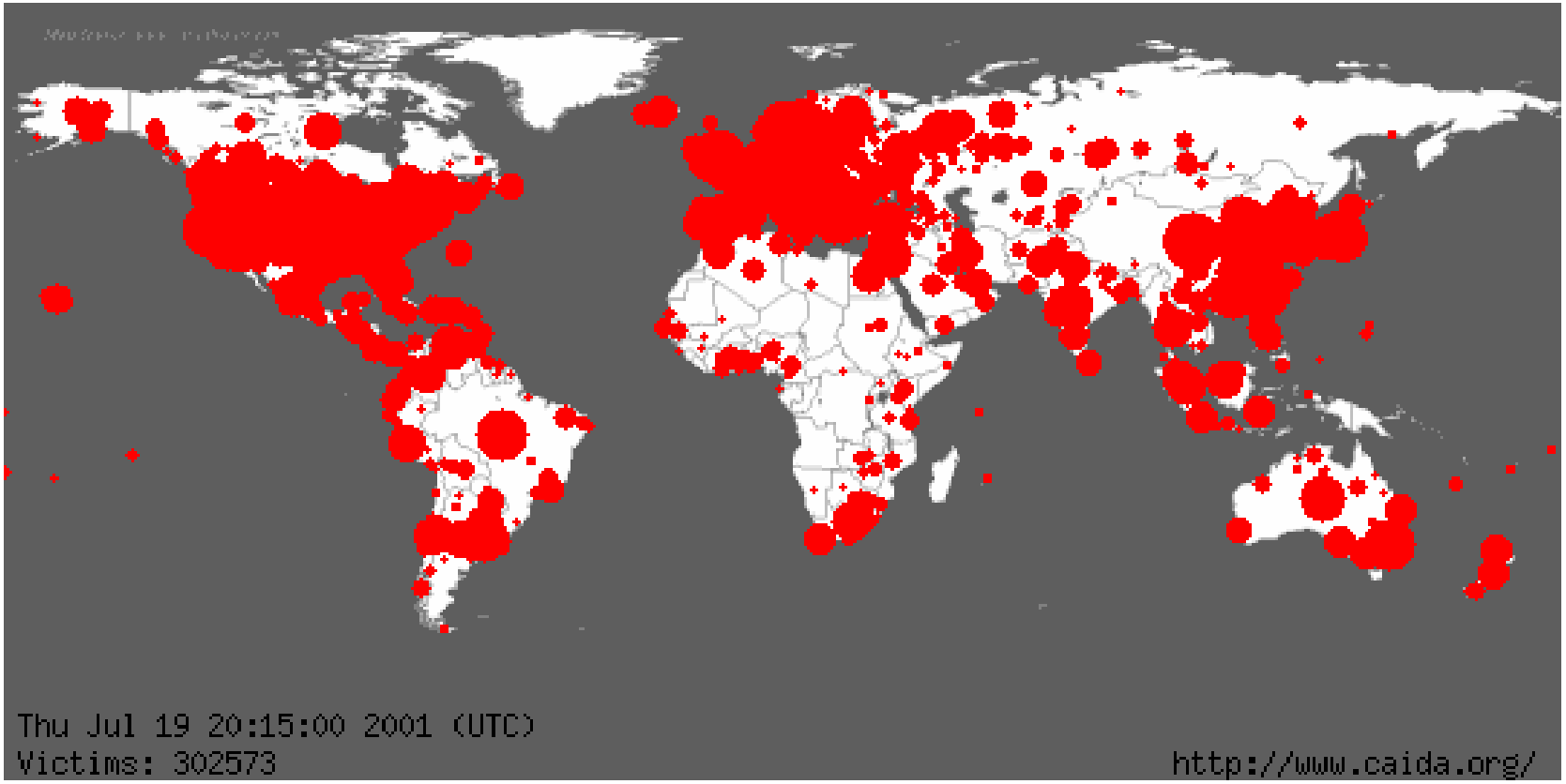
- Worm probes random IP addresses and infects web servers vulnerable to IIS exploit
- Defaces English websites hosted on server with message:
  `Welcome to http://www.worm.com! Hacked by Chinese!`
- On July 19 over 359,000 hosts infected in 13-hour period
  - over 2,000 hosts infected per minute at peak
  - at 5:00 pm, worm attempted DoS attack against 198.137.240.91 (www.whitehouse.gov)
  - David Moore – www.caida.org/analysis/security/code-red/index.xml
- Estimated 975,000 servers infected by end of August with losses of $2.4 billion – Computer Economics
- Shut down Japan Airline computer affecting ticketing & check-in, delaying 55 flights and 15,000 passengers 1-2 hours

Spread of Code Red Worm



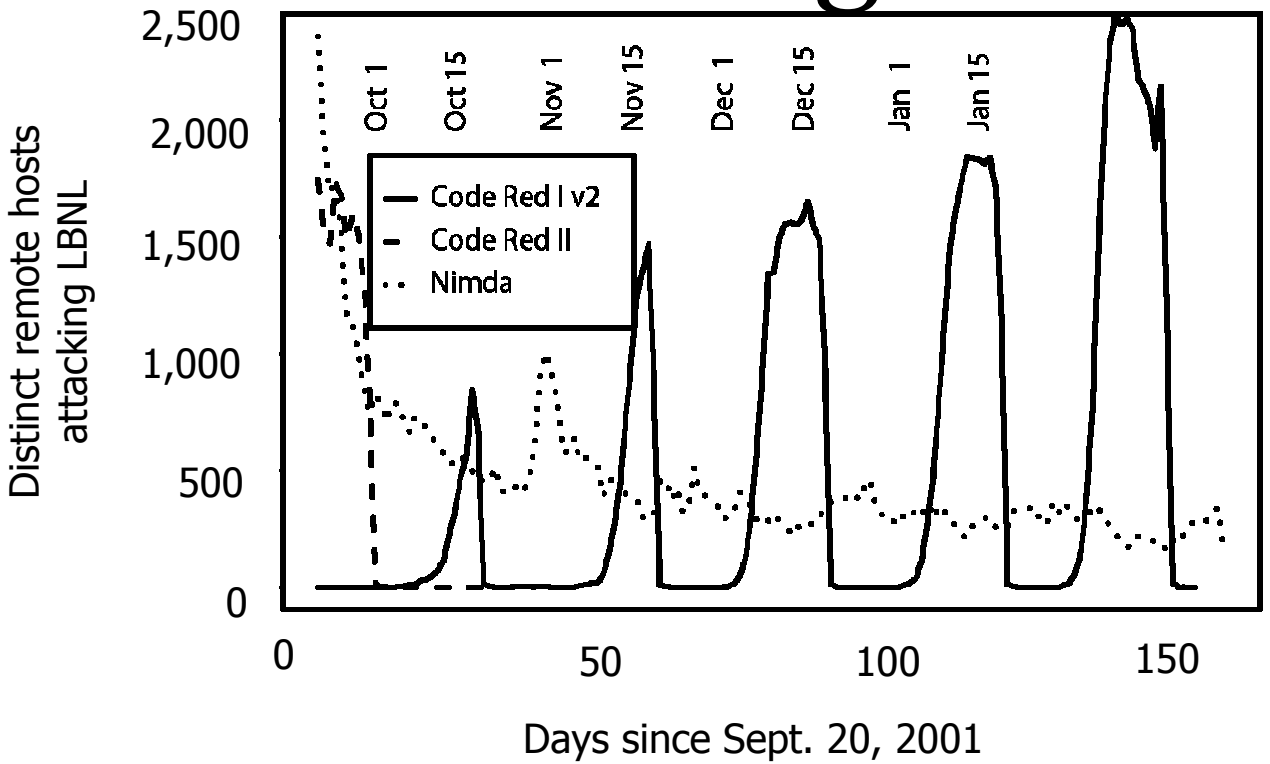
July 19 01:05:00 2001

19 Hours Later



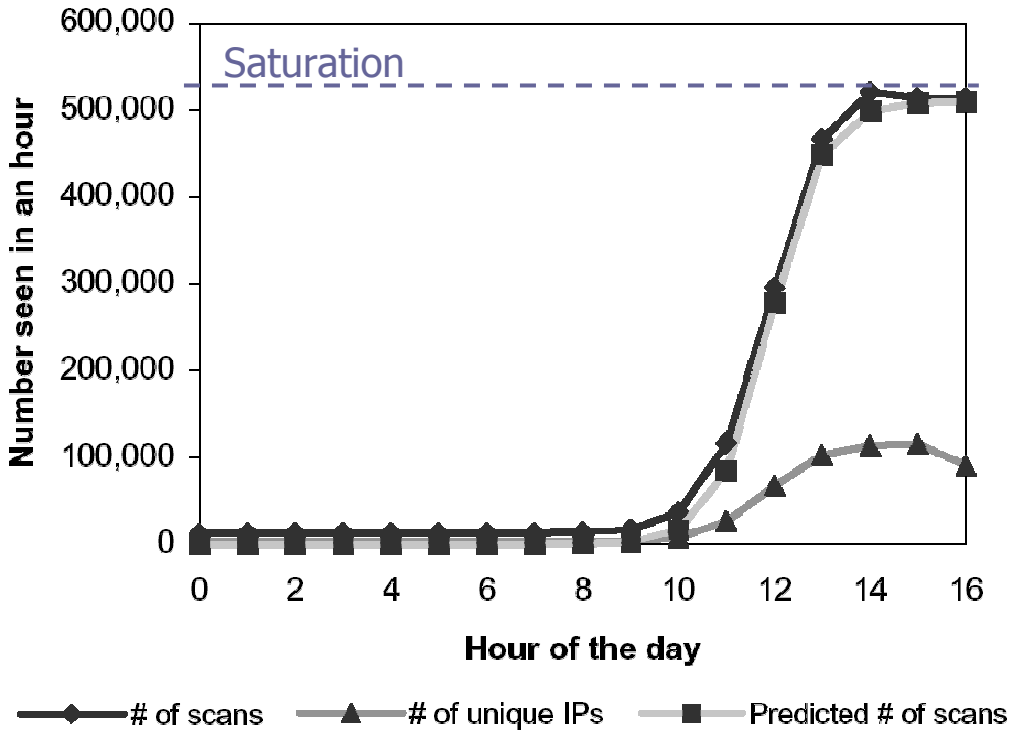
July 19 20:15:00 2001

Code Red: Scanning technique



- Code Red I: 99 threads scan for vulnerable IIS installations, using random number generator
- Worm deactivated itself after a few days, but was designed to reactivate every month

Code Red: Analytical model



Infected fraction

$$\frac{da}{dt} = K \cdot a \cdot (1-a) \qquad a = \frac{e^{K \cdot (t-T)}}{1+e^{K(t-T)}}$$

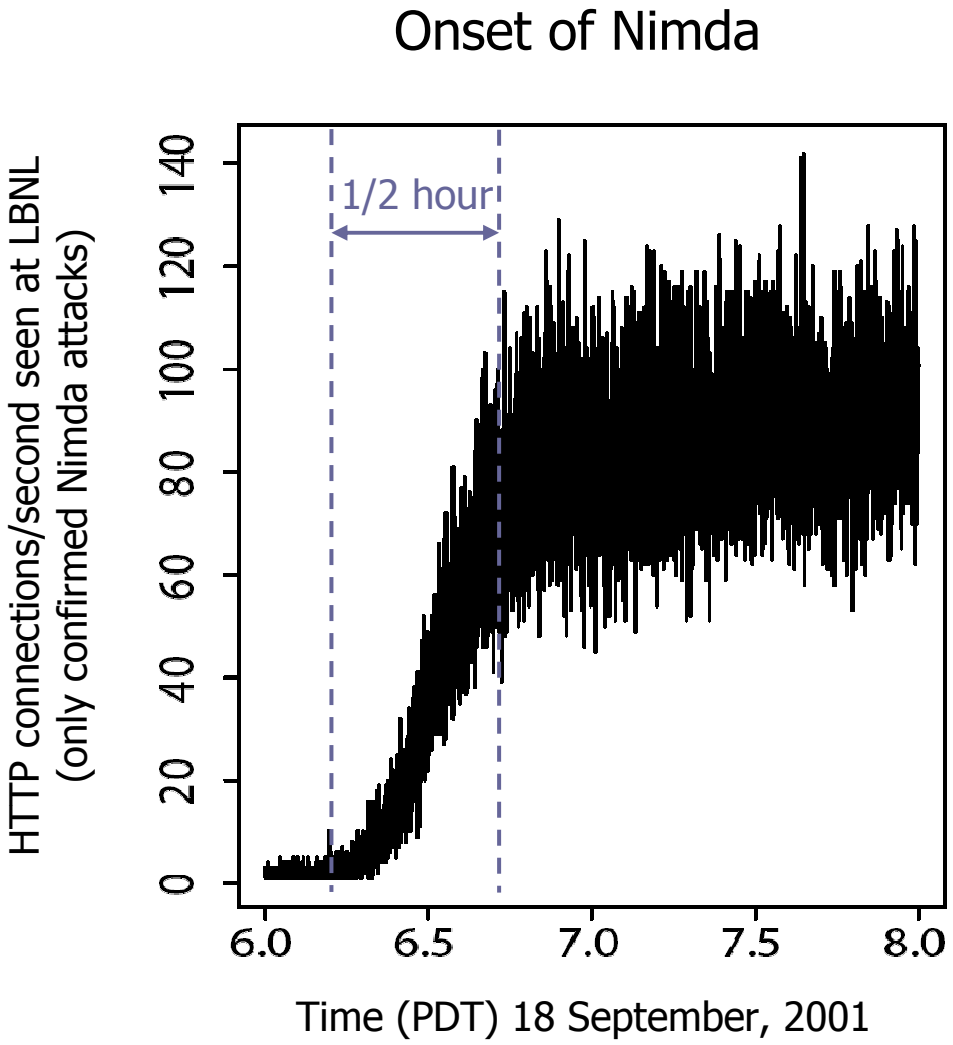
Initial compromise rate

- Simplifying assumptions:
  - No patching
  - No firewalls
  - No churn
- Infection rate is proportional to
  - # hosts already infected
  - # hosts not infected, but susceptible
- Result: Logistic equation
- Well known for epi-demics in finite systems

Improvements: Localized scanning

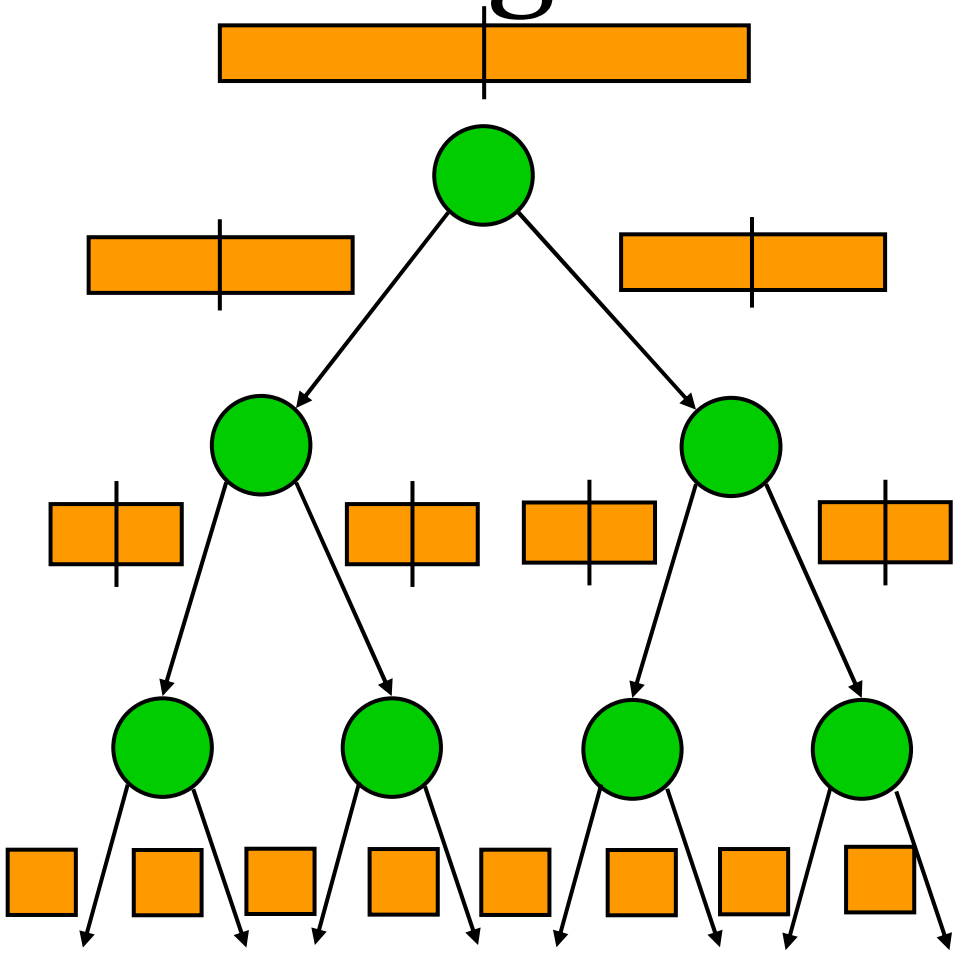
- Observation: Density of vulnerable hosts in IP address space is not uniform
- Idea: Bias scanning towards local network
- Used in CodeRed II
  - P=0.50: Choose address from local class-A network (/8)
  - P=0.38: Choose address from local class-B network (/16)
  - P=0.12: Choose random address
- Allows worm to spread more quickly

Improvements: Multi-vector



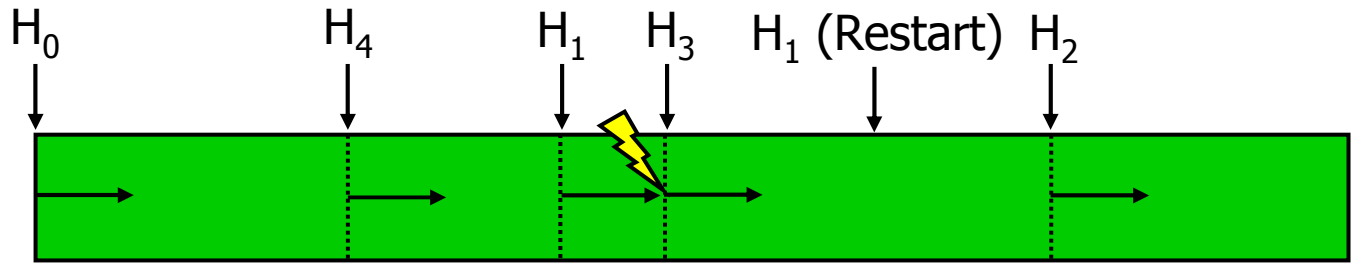
- Idea: Use multiple propagation methods simultaneously
- Example: Morris worm
  - fingerd attack
  - sendmail DEBUG cmd
  - rhosts files
  - Password cracking
- Example: Nimda
  - IIS vulnerability
  - Bulk e-mails
  - Open network shares
  - Defaced web pages
  - Code Red II backdoor

Improvements: Hit-list scanning



- Problem: Spread is slow during initial phase
- Idea: Collect a list of promising targets before worm is released
  - Low-profile 'stealthy' scan
  - Distributed scan
  - Spider/crawler
  - Surveys or databases
  - Attacks from other worms
- Low overhead, since list shrinks quickly

Improvements: Permutation scanning

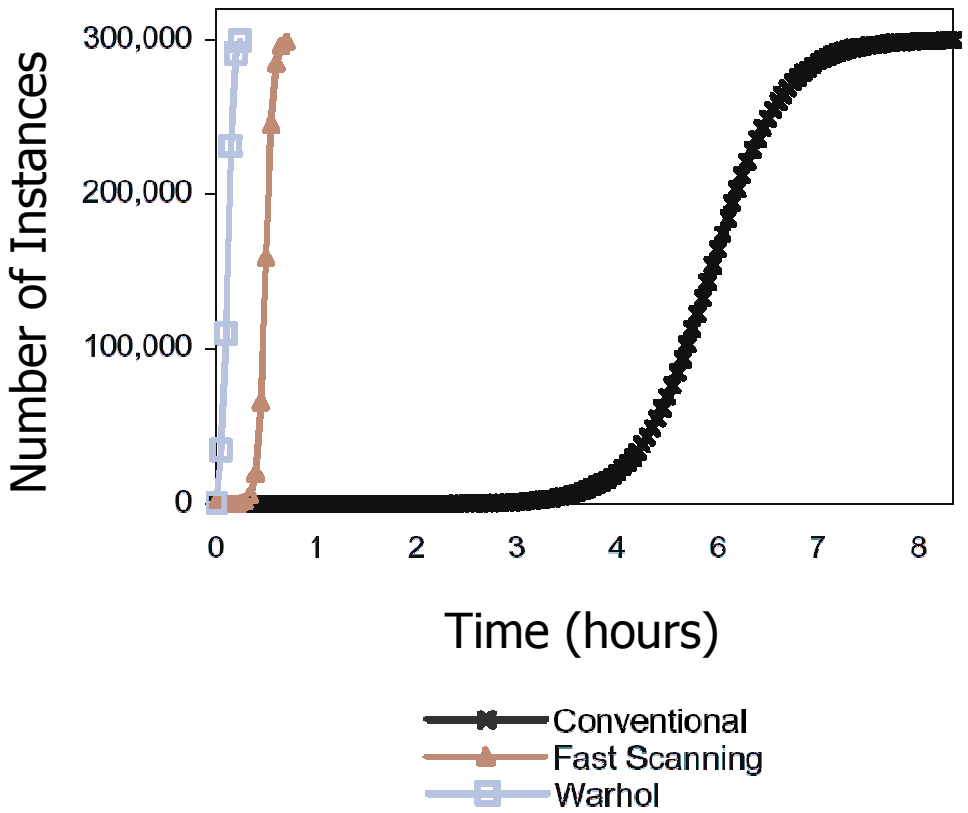


- Problem: Many addresses are scanned multiple times
- Idea: Generate random permutation of all IP addresses, scan in order
  - Hit-list hosts start at their own position in the permutation
  - When an infected host is found, restart at a random point
  - Can be combined with divide-and-conquer approach

Warhol worms

"In the future, everyone will have 15 minutes of fame"

-- Andy Warhol



- Worm using both hit-list and permutation scanning could infect most vulnerable targets in <1 hour
- Simulation: Compare
  - 10 scans/second (Code Red)
  - 100 scans/second
  - 100 scans/second plus 10,000 entry hit list (Warhol worm)
- First Warhol worm observed was SQLSlammer

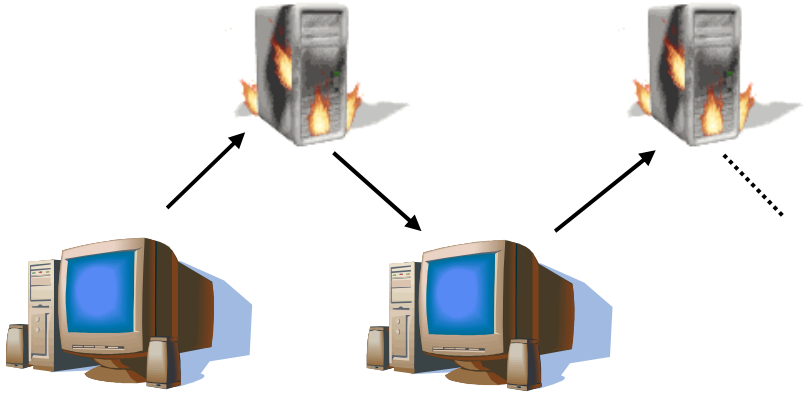
Flash worms

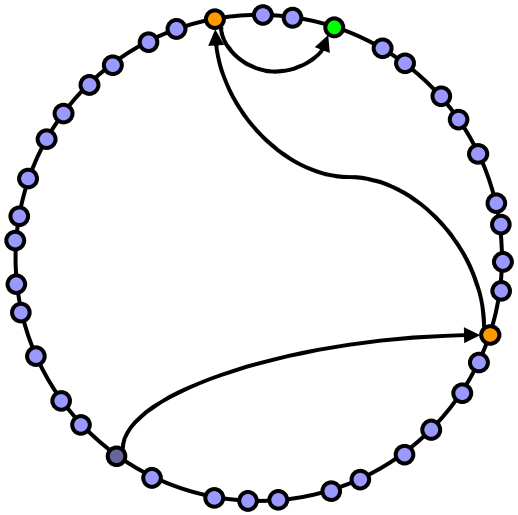
- A flash worm would start with a hit list that contains most/all vulnerable hosts
- Realistic scenario:
  - Complete scan takes 2h with an OC-12
  - Internet warfare?
- Problem: Size of the hit list
  - 9 million hosts ⇒ 36 MB
  - Compression works: 7.5MB
  - Can be sent over a 256kbps DSL link in 3 seconds
- Extremely fast:
  - Full infection in tens of seconds!

Coming soon to a network near you

- Warhol Worms
  - infect all vulnerable hosts in 15 minutes – 1 hour
  - optimized scanning
    - initial hit list of potentially vulnerable hosts
    - local subnet scanning
    - permutation scanning for complete, self-coordinated coverage
  - Slammer was first Warhol worm “in the wild”
- Flash Worms
  - infect all vulnerable hosts in 30 seconds
  - determine complete hit list of servers with relevant service open and include it with the worm
  - see paper by Stuart Staniford, Gary Grim, Roelof Jonkman, Silicon Defense

Surreptitious worms





- Idea: Hide worms in inconspicuous traffic to avoid detection
- Example: HTTP
- Leverage P2P systems?
  - High node degree
  - Lots of traffic to hide in
  - Proprietary protocols
  - Homogeneous software
  - Immense size (30,000,000 Kazaa downloads!)

Case study: Morris (or Cornell or Unix) Worm (1988)

- Robert T. Morris, Jr.
  - 23 years old, Cornell grad student, father worked at the NSA
  - He asked himself: “I wonder how large the Internet is?“
- Wrote a self-propagating program as a “test concept”
  - Exploited Unix vulnerabilities in sendmail and fingerd
  - Released at MIT
  - Bug in the worm caused it to go haywire – it was not planned to wreak havoc
- The first worm that propagated using the Internet
  - Internet was designed with *functionality in mind!*

How it entered

- sendmail (in debug mode, as released in SunOS)
- finger (VAX systems)
- r-services:
  - rexec
  - rsh

Who it attacked:

- accounts with obvious passwords:
  - none at all
  - the user name (once and appended to itself)
  - the “nickname”
  - last name (both spelled forwards and backwards)
- passwords from a 432 word included dictionary
- Used the words from `/usr/dict/words` as passwd
- trusted accounts through `.rhosts`

Systems affected

- SUN and VAX
- Gained hostnames and account names through:
  - `/etc/hosts.equiv`
  - `/.rhosts`
  - `.forward`
  - `.rhosts`
  - `routing tables`
  - `serial P2P links`
  - `randomly guessed first-hop addresses`

For further interest

- Read "With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988" at http://cs.wellesley.edu/~cs342/internet_worm1988.pdf
- Read Weaver "How to own the Internet in your spare time" at http://cs.wellesley.edu/~cs342/owninternetinsparetime.pdf