Outline

- Morning session (understanding)
  - The 10,000 foot issues
  - Overview and taxonomy
  - Worm history
  - **Epidemiological modeling**

- Afternoon session (defenses)
  - Overview
  - Detection
    - Signature-based
    - Behavioral
  - Mitigation

Why Model?

- Models frame the problem…and, therefore, the solutions
- Use models of worm propagation as a basis for
  - Systematic understanding of worm behavior
  - Systematic approach for developing defenses
- Use models to answer compelling questions
  - How does a larger vulnerable population affect worm propagation?
    - All Windows hosts vs. Windows 2000 w/ SQL?
  - How quickly does a 100x-faster worm propagate?
    - Code Red vs. Slammer
  - What are the practical limits on worm propagation?
    - What is the worst that we have to defend against?

Modeling Outline

1) Introduce basic model of worm propagation
   - Variations on model and their features

2) Use model as basis for understanding and evaluating more efficient worms
   - How do changes in worm design and behavior affect how they propagate?

- Defenses are next topic in the afternoon

Worms as Epidemics

- Worms well described as infectious epidemics
- Classic SI model (Susceptible → Infected)
  - N: population size (IP address space)
  - S(t): susceptible hosts at time t (MS IIS hosts)
  - I(t): infected hosts at time t (infected MS IIS hosts)
  - β: contact rate
  - i(t): I(t)/N
  - s(t): S(t)/N

$$\frac{dI}{dt} = \beta \frac{IS}{N}$$

$$\frac{dS}{dt} = -\beta \frac{IS}{N}$$

$$\Rightarrow \frac{di}{dt} = \beta i(1-i)$$

$$i(t) = \frac{e^{\beta(t-T)}}{1+e^{\beta(t-T)}}$$

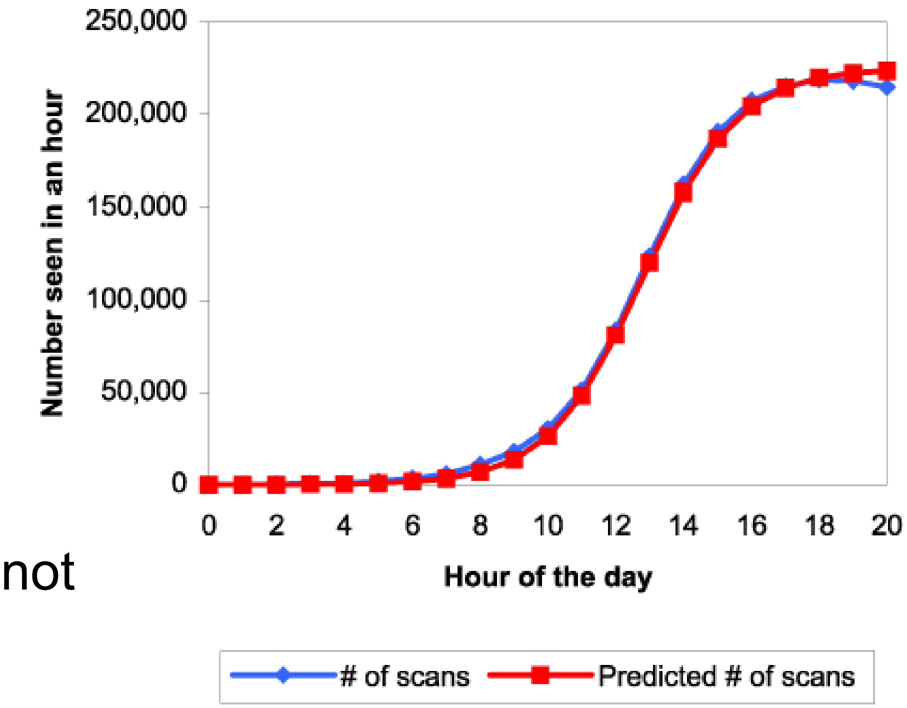
Moore, Shannon, Voelker, Savage, *Internet Quarantine: Requirements for Containing Self-Propagating Code*, Infocom 2003

Staniford, Paxson, Weaver, *How to Own the Internet in Your Spare Time*, USENIX Security 2002

UCSD CSE Computer Science and Engineering — ICSI International Computer Science Institute

Code Red Example [Staniford02]

- Early worm: Code Red v2
  - Uniform random scanning
  - August 1, 2001 outbreak
  - $N = 2^{32}$, $S = 225{,}000$, $\beta = 10/s$
- Early t, i(t) is exponential
  - Few S infected
  - Probe on S successfully infects
- Inflection in middle
  - S infected = S uninfected
  - Probe equally likely to infect or not
- Late t, i(t) → constant
  - All S become infected



Extending the Model

- SI model is very simple
  - Two host states, homogeneous behavior, complete network graph, etc.
- Situation more complex in practice
  - Host behavior: death, patching, vigilance
  - Worm behavior: delay, bias
  - Network constraints: congestion, bandwidth limits
  - Luck: early success → faster worm
- Require more complex analytic models
  - Continuous, discrete variants
- Network topology
  - Theoretical results on worm propagation

Host Behavior

- Hosts change state during outbreak
  - Epidemiological models capture different host behaviors
- Susceptible → Infected (SI) [Staniford02], [Moore03]
  - Once infected, host stays infected indefinitely
- Susceptible → Infected → Susceptible (SIS) [Zou02], [Chen03]
  - Reboot cleans host, but reverts back to susceptible
  - Cycle of infections (one virus to another)
  - Infection dies → model as death rate *d*
  - I decreases, no change in S
  - Worms slows down

UCSD CSE Computer Science and Engineering
ICSI INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Zou, Gong, Towsley, *Code Red Worm Propagation Modeling and Analysis*, CCS 2002

Chen, Gao, Kwiat, *Modeling the Spread of Active Worms*, INFOCOM 2003

Host Behavior (cont'd)

- Susceptible → Infected → Removed (SIR)
  - User patches, shuts down host, admin blocks traffic
    - Immunize infected (Kermack-McKendrick model)
    - Immunize susceptibles (real-time vaccination)
  - Model variants
    - Infection rate now a function of time: $\beta \rightarrow \beta(t)$ [Zou02]
    - Count hosts that change state according to patch rate *p* [Chen03]
  - Impact
    - Infected hosts I, susceptible hosts S decrease
    - Worms slow down more than SIS
    - Not all susceptibles infected (raises epidemic threshold)
- Susceptible → Infected → Immune → Susceptible (SIIS)
  - Clean reboots, cycle of infections…with a pause
  - User vigilance for viruses [Wang03]

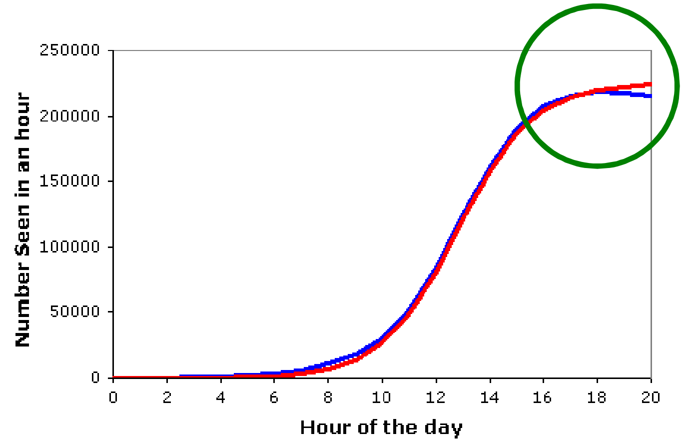
Wang, Wang, *Modelling the Effects of Timing Parameters on Virus Propagation*, WORM 2003

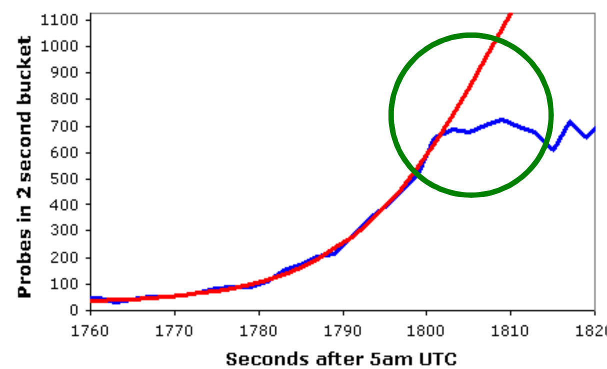
Worm Behavior

- Worms not necessarily continuous and uniform
- Delay [Chen03], [Wang03]
  - Worm infections are discrete events
  - Takes time to infect a machine, initiate new infections
  - Delay slows worm
    - 30 seconds to infect → worm 1/6th slower
- Biased scanning [Chen03]
  - Probe local addresses with higher probability
  - Interestingly, worm slows
  - Does not account for delay, density (more later)

Network Constraints

- Networks do not have unlimited resources
- Congestion [Zou02], [Serazzi03]
  - Code Red packets dropped at routers
  - Model as decreased β or failed links
  - Worm slows
- Bandwidth-limiting [Serazzi03], [Kesidis05]
  - Slammer rate β limited by access link
  - Enterprise networks as compartments
    - Different spreading rates within compartment, to Internet
    - Bandwidth constraint σ on links to Internet
  - Probe rate capped





Moore et al., *The Spread of the Sapphire/Slammer Worm*, CAIDA Tech Report, 2003

Serazzi, Zanero, *Computer Virus Propagation Models*, MASCOTS 2003

Kesidis, Hamadeh, Jiwasurat, *Coupled Kermack-McKendrick Models for Randomly Scanning and Bandwidth-Saturating Internet Worms*, QoS-IP 2005

UCSDCSE Computer Science and Engineering

ICSI INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Analytic Models

- Continuous
  - Random Constant Spread (RCS) [Staniford02]
    - Simple SI model
  - Two-Factor worm model [Zou02]
    - SIR model
  - Compartment-Based [Serazzi03], Coupled Kermack-McKendrick [Kesidis05]
    - SI model, internal/external B, constraints on B
- Discrete
  - Analytical Active Worm Propagation (AAWP) [Chen03]
    - $i_{t+1} = i_t + (N - i_t)[1 - (1 - 1/S)^{\beta i_t}]$ on average
- Worm Coverage Transitive Closure [Ellis03]
  - Framework, taxonomy for worm propagation

Ellis, *Worm Anatomy and Model*, WORM 2003

Network Topologies

- What is the impact of network topology on worms?
  - So far have assumed a fully connected graph
  - Worms can spread through application networks, too
- Consider connected, (un)directed graphs
  - Random (baseline)
  - Lattice, torus (spatial models)
  - Hierarchical (users exchanging programs)
  - Small-world (DHT)
  - Hypercube (DHT)
  - Power-law (AS connectivity)

Topology Model Framework

- Use SIS epidemiological model
  - Hosts only infect directly connected neighbors
  - Infection rate β, cure/recover/death rate δ
    - Note: Hosts can be infected repeatedly
  - Epidemic threshold ρ
    - $S(0) < \rho$ → infection dies out
    - $S(0) > \rho$ → infection persists
  - Duration of worm
    - Epidemic stops after finite amount of time…how long to stop?
- Early work simulated spread in simple graphs [Kephart91]
- Later work derives fundamental results and bounds [Garetto03], [Wang03], [Ganesh05]

Kephart, White, *Directed-Graph Epidemiological Models of Computer Viruses*, IEEE RSP 1991

Wang, Chakrabarti, Wang, Faloutsos, *Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint*, SRDS 2003

Garetto, Gong, Towsley, *Modeling Malware Spreading Dynamics*, Infocom 2003

Ganesh, Massoulié, Towsley, The Effect of Network Topology on the Spread of Epidemics, Infocom 2005

Epidemic Threshold [Wang03]

- How can we characterize the epidemic threshold for an arbitrary graph?
  - Knowing contact rate β and graph, can predict outcome
- Discrete Epidemic Threshold Model
  - Probabilistic model of transitions in SIS model
- Fundamental: Epidemic threshold τ of an arbitrary graph related to adjacency matrix
  - λ = largest eigenvalue
  - $\tau = 1/\lambda$
- Single graph parameter determines outcome
  - Cure rate $\delta < \beta * \lambda \rightarrow$ epidemic persists, otherwise dies
  - For arbitrary graphs

UCSD CSE Computer Science and Engineering

ICSI International Computer Science Institute

Wang, Chakrabarti, Wang, Faloutsos, *Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint*, SRDS 2003

Epidemic Duration [Ganesh05]

- How quickly does epidemic stop (network recover)?
  - Depends on topology
- Extend results of [Wang03]
  - Epidemic stops after finite amount of time
  - N hosts (all susceptible), T = epidemic duration
  - Fast epidemic: $E[T] = O(\log(N))$ (logarithmic)
  - Slow epidemic: $E[T] = \Omega(N^{\alpha})$ (exponential)
- Bounds on infection rate determine fast vs. slow
  - Connected: $\beta \rightarrow 1/N$
  - Hypercube: $\beta \rightarrow 1/\log_2 N$

UCSDCSE Computer Science and Engineering | ICSI International Computer Science Institute

Ganesh, Massoulié, Towsley, *The Effect of Network Topology on the Spread of Epidemics*, Infocom 2005

Worm Efficiency

- Use models as a basis for understanding more efficient worms
- All model variants aside, two key questions determine worm propagation:

1) How likely is it that a given infection attempt is successful?
   - Target selection
   - Vulnerability distribution (e.g., density – S(0)/N)

2) How frequently are infections attempted?
   - Determined by contact rate β

UCSDCSE Computer Science and Engineering
INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Target Selection Efficiency

- Recall random scanning worms
  - Generate random IP address, attempt to infect
  - Most attempts fail → Very inefficient!
- How might worms do better?
  - Improve likelihood that each attempt targets a susceptible host
  - Improve likelihood of targeting a susceptible & uninfected host
- Techniques
  - Local address scanning
  - Hit-list scanning
  - Permutation scanning
  - Warhol worm
  - Importance scanning
  - Topological scanning
  - Flash worms

Local Address Scanning

- Biased random address scanning
  - Target nearby hosts in IP address space
  - Targets likely exist

    → Improve vulnerability density
  - Targets likely have similar software

    → Improve probability of infection
  - Targets likely can communicate with each other

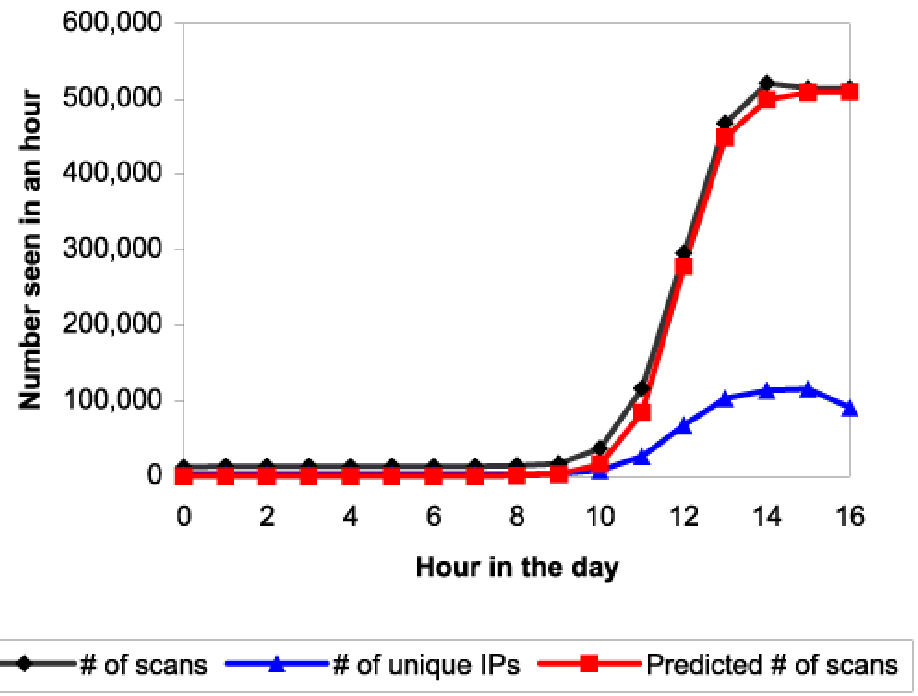
    → Improve firewall evasion
- Poster child: Code Red II worm
  - Pr(3/8): Choose IP from same class B (/16)
  - Pr(1/2): Choose IP from same class A (/8)
  - Pr(1/8): Choose random IP from 2^{32}
  - Empirically, appeared to work well

Hit-List Scanning

- Time to infect initial hosts dominates infection time

→ Use list of potentially vulnerable machines to seed worm

- Jumpstart phase
  - Divide list with child upon each infection
- Random scanning phase
  - When list ends
- Easy to gather list
  - Stealthy scans
  - Distributed scans
  - DNS searches
  - Spiders
  - Surveys
  - Passively listen
  - ?Inside Information?



Staniford, Paxson, Weaver, *How to Own the Internet in Your Spare Time*, USENIX Security 2002

UCSDCSE Computer Science and Engineering
ICSI INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Permutation Scanning

- Random scanning still has inefficiencies
  - Hosts probed multiple times
  - Do not know when all vulnerable machines infected
- Approach: Permutation scanning
  - Use same pseudo random permutation of IP addresses
  - Hosts start at different points of permutation (their IP)
  - Upon probing infected host, choose new random index
  - Newly infected hosts also choose new random index
  - Self-stop when probed multiple infected machines

Staniford, Paxson, Weaver, *How to Own the Internet in Your Spare Time*, USENIX Security 2002

Permutation Scanning (cont'd)

- Coordination
  - Probing already infected host → another host is already working the sequence and is further along
  - Random jump to new index reduces multiple scans
- Termination
  - Self-stop is local, independent decision (more later)
- Variant: Partitioned permutation scanning
  - Each infected host has a range of the permutation
  - Divide range with child upon infection
  - Fall back to permutation scanning when range is small
  - Further reduces redundant scans

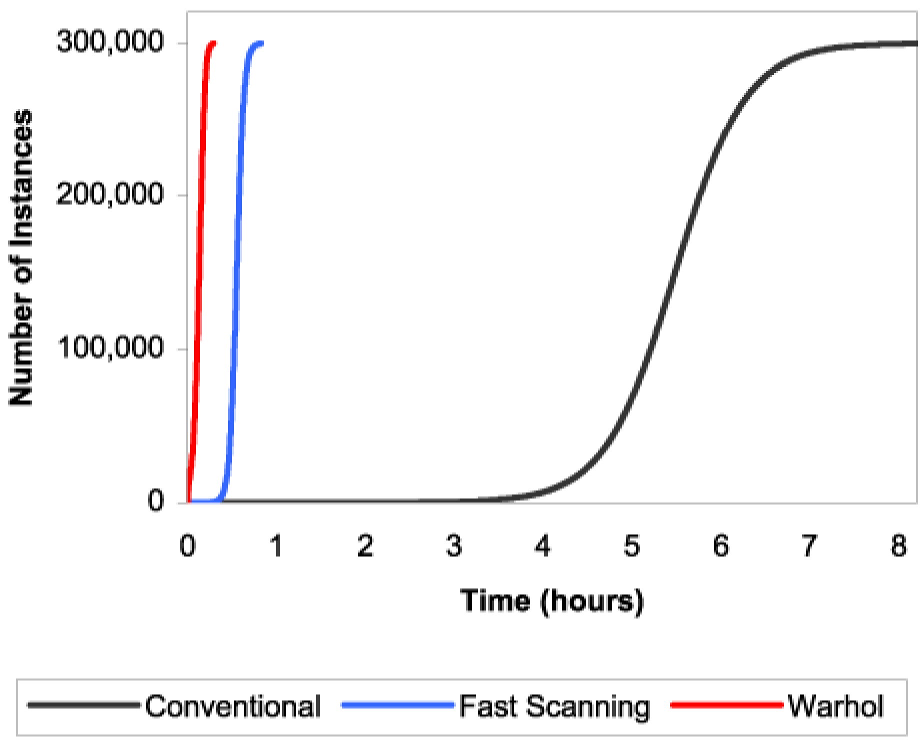
UCSDCSE Computer Science and Engineering

ICSI INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Staniford, Paxson, Weaver, *How to Own the Internet in Your Spare Time*, USENIX Security 2002

Warhol Worm

- Combine hit-list, permutation, and faster scanning
  - 100 scans/sec
- Infect 99.99% vulnerable hosts in 15 minutes



Staniford, Paxson, Weaver, *How to Own the Internet in Your Spare Time*, USENIX Security 2002

Importance Scanning [Chen05]

- Distribution of susceptible hosts varies across networks
  - Better to weight random scans by distributions
    - Wasted effort to probe empty networks
  - Exploited in an ad-hoc fashion by biased scanning
- Idea: Explicitly learn distributions as worm propagates
- Two stages
  - Learning: Infected hosts report IPs to a central server
    - Count N_i infected hosts for network *i* (e.g., all /8s)
    - Use server to obtain global estimate of population *N*
  - Importance: After receiving sufficient IPs, broadcast estimated distributions for each network
    - Scan with weight N_i/N
    - 10,000 IPs → distribution error 10^{-4}
- Speed: Importance > Permutation > Random

Chen, Ji, *A Self-Learning Worm Using Importance Scanning*, WORM 2005

Routing Worms [Zou06]

- Idea: Scan only routable IP address space
  - Reduce wasted effort scanning unroutable IPs
    - 33% of IPv4 address space is BGP routable
  - Enables selective attacks: company, country, ISP, AS
- Generate target address space based upon routing info
  - Overhead: need to disseminate network lists with scans
- /8 routing worm
  - Scan only assigned and routed /8 networks
  - May05: 132 /8s → only 132 bytes w/ each packet
- BGP routing worm
  - 78,000 prefixes → 200 KB
  - Reduce accuracy w/ aggregation → 30 KB
- Can combine with hit-lists, permutation scanning
  - Improvement proportional to reduced address space scanned

Zou, Towsley, Gong, Cai, *Advanced Routing Worm and Its Security Challenges*, TSMSI 2006

Topological Scanning

- Harvest new targets based upon information stored on infected host (dynamic hit-lists)
  - Address books: Email viruses
  - System logs, host files: Morris worm
  - URLs in cache, HTML content
  - Peer lists in P2P applications
- These "pointers" form a topology among hosts

1) Jumpstart using these pointers
  - Switch to permutation scanning after start

2) Spread entirely within application topology
  - DHT search time → DHT Infection time (Chord O(logN))

Staniford, Paxson, Weaver, *How to Own the Internet in Your Spare Time*, USENIX Security 2002

Flash Worms

- Variant of hit-list: Create entire list of vulnerable hosts, not just a seed list
  - Divide list into n blocks
  - Infect first address in each block
  - Delegate block to infected child, repeat
  - 3 million hosts, n = 10 → 7 levels deep → 30 seconds
  - Overlap blocks for redundancy
- Bottlenecks
  - Time to transfer initially long list
    - Seed using high-bandwidth hosts, use high-bandwidth list server
  - Latency to infect at each level
    - Depends on parallelism at each host

Staniford, Paxson, Weaver, *How to Own the Internet in Your Spare Time*, USENIX Security 2002

Contact Rate Efficiency

- Increase frequency of probe attempts (β)
- Three mechanisms
  - Reduce latency of each attempt
  - Maximize bandwidth utilization
  - Increase parallelism
- Techniques
  - Network transport
  - Local scanning
  - Even Flashier Flash worms

Network Transport

- Consider Code Red and Slammer
  - Code Red infects via HTTP on TCP
  - Slammer infects via single UDP packets
- Code Red probing limited by RTT and timeouts
  - TCP handshake, timeouts to non-existent hosts
  - Parallelism limited by # simultaneous TCP connections
  - 10 probes/sec, 14 hours to infect 360,000 hosts
- Slammer probing limited only by bandwidth
  - Maximum parallelism
  - 1000 probes/sec/worm, Internet scanned < 10 mins
- Worm not necessarily constrained by transport protocol semantics
  - TCP: Send SYNs at line rate → TCP worm can spread like slammer!
    - Needs a little more magic, however, to handle scan-induced congestion

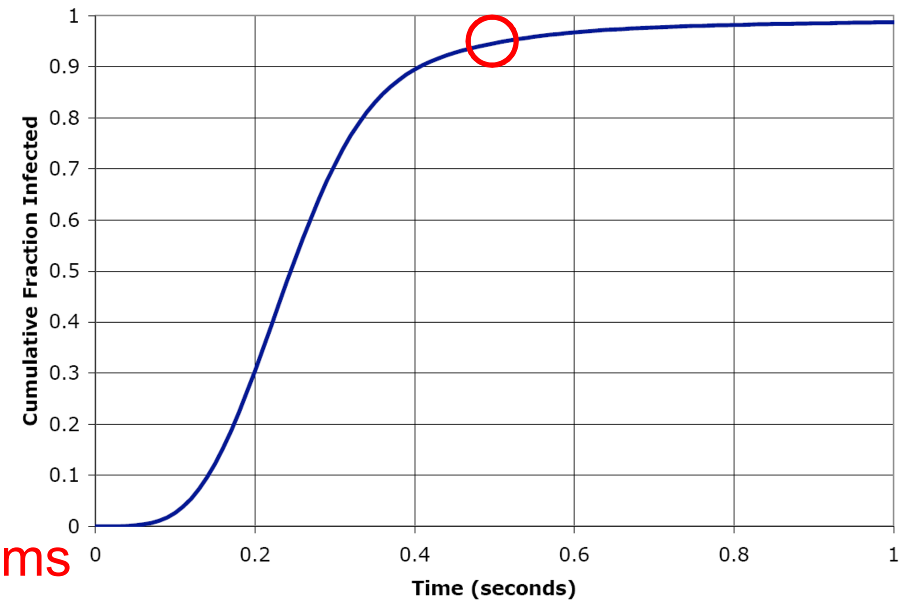
Staniford, Moore, Paxson, Weaver, *The Top Speed of Flash Worms*, WORM 2004

Local Address Scanning Redux

- Target hosts in same network have
  - Lower latency (<1 ms vs. 100 ms)
  - Higher bandwidth (100—1000 Mb/s vs. 0.1—10 Mb/s)
- Local address scanning naturally and conveniently takes advantage of both

Even Flashier Flash Worms

- Flash worm in 30 seconds? Bah. Child's play.
- What are the limits in terms of worm efficiency?
  - Infect w/ single UDP packet (Slammer)
  - Complete hit-list of known vulnerable hosts
  - High-bandwidth hosts for internal nodes of spread tree
    - 750 Mb/s for root node
    - 1 Mb/s for internal nodes
  - Latency analysis of Internet
    - 103ms between random hosts
  - Two-level tree
    - 10,000 first level
    - 100 second level
- Bottom line
  - 1 M hosts: 95% infected in 510ms



UCSDCSE Computer Science and Engineering — INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Staniford, Moore, Paxson, Weaver, *The Top Speed of Flash Worms*, WORM 2004

Modeling "Efficient" Worms

- Analytic models for "efficient" worms challenging
- Need model to capture variation in
  - Probe success rate due to, e.g., heterogeneity in vulnerability density (S(t)/N)
  - Probe frequency (β) due to network characteristics (latency, bandwidth)
    - Hosts fundamentally probe at different rates
  - Dependent behavior (need to do more than just count)
    - Tracking progress in permutation scanning
- Approaches
  - [Chen03] for attempt at local address scanning
  - [Zou06] for more advanced worm strategies (routing, hit-list, etc.)
- Motivates development of simulation models

UCSDCSE Computer Science and Engineering — ICSI INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Zou, Towsley, Gong, *On the performance of Internet worm scanning strategies,* Performance Evaluation 63 (2006)

Simulation Models

- Simulate actions of worm as it infects hosts in network
- Incorporate realistic
  - Network topologies: AS graphs, router graphs
  - Network characteristics: latencies, bandwidths
  - Victim vulnerability distributions
    - Victims of Code Red v2, Code Red II, Witty worms
- Challenge: Trading off scale and accuracy ("simulating the Internet")
- Simulation techniques
  - [Wagner03], [Staniford04], [Vogt04] for simulating worm propagation
  - [Weaver03] evaluates scale-down to tradeoff accuracy & scale
  - [Liljenstam03], [Moore03] for examples of rich simulation models to evaluate worm defenses
- See www.datcat.org for data sets (Code Red, Witty)

Weaver, Hamadeh, Kesidis, Paxson, *Preliminary Results Using Scale-Down to Explore Worm Dynamics*, WORM 2004

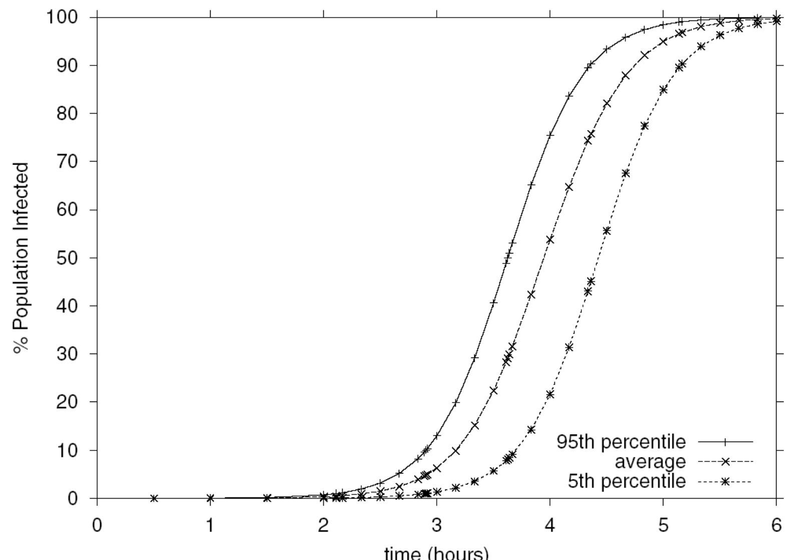
Liljenstam, Nicol, Berk, Grey, Simulating Realistic Network Worm Traffic for Worm Warning System Design and Testing, WORM 2003

Vogt, *Simulating and optimizing worm propagation algorithms*, http://www.lemuria.org/security/WormPropagation.pdf, 2004

Wagner, Dubendorfer, Plattner, Hiestand, *Experiences with Worm Propagation Simulations*, WORM 2003

On Average

- Worm spreading is very sensitive to early variability
  - Lucky random scanning worm infects quickly
    - Extreme: Probe a vulnerable host on each random roll
  - Unlucky worm takes more time to snowball
    - Takes many time steps to randomly find vulnerable host
  - 4 hours: 55% infected on average, 80% for 95th percentile



Moore, Shannon, Voelker, Savage, *Internet Quarantine: Requirements for Containing Self-Propagating Code*, Infocom 2003

On Average (cont'd)

- Natural to compute average behavior
  - Analytic models usually assume average case
  - Might average multiple simulation runs
- But may not want average case
  - Do you want to defend against an average worm, or against most of the possible worm outcomes?
- Difficult to capture using analytic models, much easier using simulation models
  - Behavior in 95/100 simulation runs

Summary

- Random scanning worms well modeled as epidemics
  - Susceptible→Infectives (S,I) in population N, contact rate β
- Variants tune for different conditions
  - Delay, patching, death, topology
- Efficiency determined by key two factors
  - Likelihood that a probe infects
    - Reduce population N, improve density S/N
  - Frequency of probe attempts
    - Contact rate β
- Many ways that a worm can improve efficiency
  - Target selection: local bias, hit-list, permutation, topological, …
  - Contact rate: latency, bandwidth, parallelism

Model Papers

- Chen, Gao, Kwiat, *Modeling the Spread of Active Worms*, INFOCOM 2003
- Ellis, *Worm Anatomy and Model*, WORM 2003
- Kesidis, Hamadeh, Jiwasurat, *Coupled Kermack-McKendrick Models for Randomly Scanning and Bandwidth-Saturating Internet Worms*, QoS-IP 2005
- Moore, Shannon, Voelker, Savage, *Internet Quarantine: Requirements for Containing Self-Propagating Code*, Infocom 2003
- Moore et al., *The Spread of the Sapphire/Slammer Worm*, CAIDA Tech Report, 2003
- Serazzi, Zanero, *Computer Virus Propagation Models*, MASCOTS 2003
- Staniford, Paxson, Weaver, *How to 0wn the Internet in Your Spare Time*, USENIX Security 2002
- Wang, Wang, *Modeling the Effects of Timing Parameters on Virus Propagation*, WORM 2003
- Zou, Gong, Towsley, *Code Red Worm Propagation Modeling and Analysis*, CCS 2002

Topology Papers

- Kephart, White, *Directed-Graph Epidemiological Models of Computer Viruses*, IEEE RSP 1991
- Ganesh, Massoulie, Towsley, *The Effect of Network Topology on the Spread of Epidemics*, Infocom 2005
- Garetto, Gong, Towsley, *Modeling Malware Spreading Dynamics*, Infocom 2003
- Wang, Chakrabarti, Wang, Faloutsos, *Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint*, SRDS 2003

Efficiency Papers

- Chen, Ji, *A Self-Learning Worm Using Importance Scanning* , WORM 2005
- Staniford, Moore, Paxson, Weaver, *The Top Speed of Flash Worms*, WORM 2004
- Staniford, Paxson, Weaver, *How to 0wn the Internet in Your Spare Time*, USENIX Security 2002
- Zou, Towsley, Gong, *On the performance of Internet worm scanning strategies*, Performance Evaluation 63 (2006)
- Zou, Towsley, Gong, Cai, *Advanced Routing Worm and Its Security Challenges*, TSMSI 2006

Simulation Papers

- Liljenstam, Nicol, Berk, Grey, *Simulating Realistic Network Worm Traffic for Worm Warning System Design and Testing*, WORM 2003
- Moore, Shannon, Voelker, Savage, *Internet Quarantine: Requirements for Containing Self-Propagating Code*, Infocom 2003
- Staniford, Paxson, Weaver, *How to 0wn the Internet in Your Spare Time*, USENIX Security 2002
- Vogt, *Simulating and optimizing worm propagation algorithms*, http://www.lemuria.org/security/WormPropagation.pdf, 2004
- Wagner, Dubendorfer, Plattner, Hiestand, *Experiences with Worm Propagation Simulations*, WORM 2003
- Weaver, Hamadeh, Kesidis, Paxson, *Preliminary Results Using Scale-Down to Explore Worm Dynamics*, WORM 2004