

How to Own the Internet in Your Spare Time

Stuart Staniford (*Silicon Defense*)

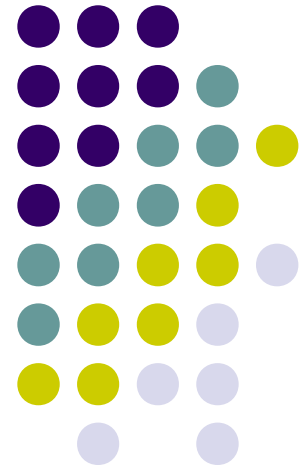
Vern Paxson (*ICSI Center for Internet Research*)

Nicholas Weaver (*UC Berkeley*)

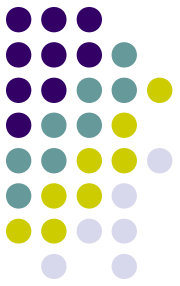
11th USENIX Security Symposium 2002

ligouras@csd.uoc.gr

CS455

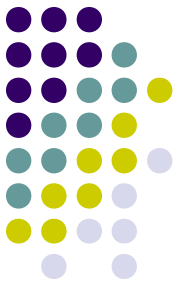


Introduction

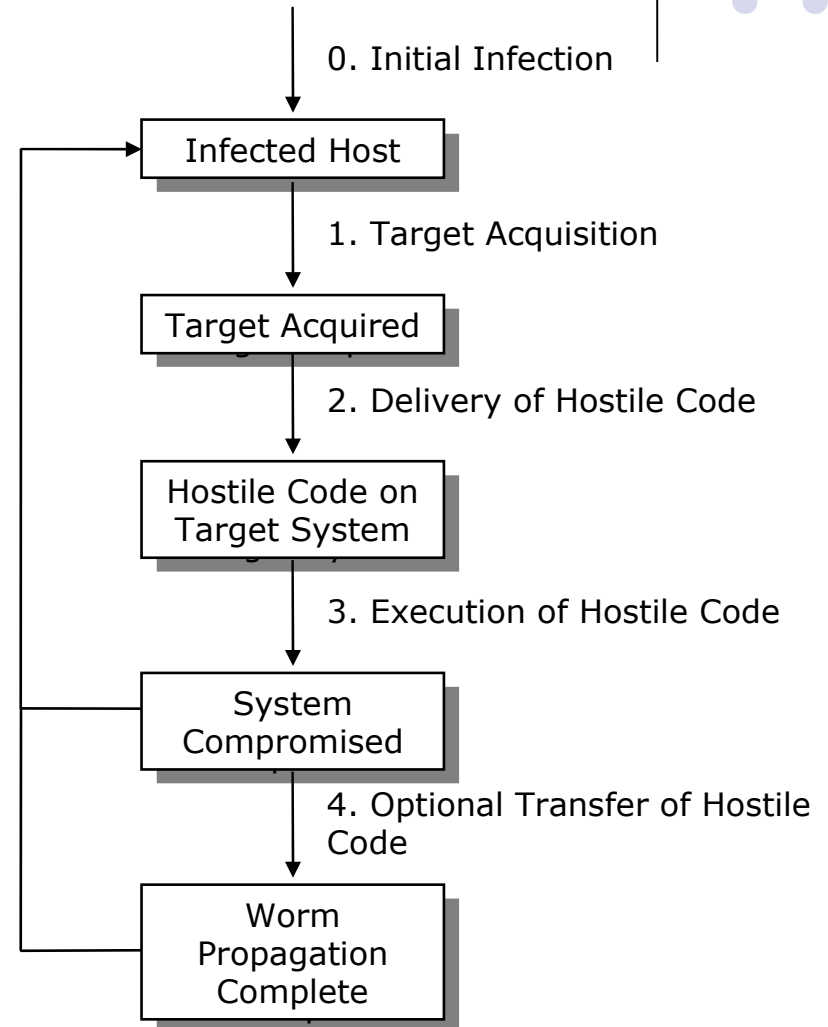


- The threat: An attacker controlling high numbers (millions) of hosts on the Internet could cause immense damage
 - DDoS attacks: bring down as much of the Internet as possible
 - Access sensitive information (passwords, credit card numbers, archived mail etc.)
 - Corrupt information, spread false information
- The reason: Worms
 - Programs that self-propagate across the Internet by exploiting security flaws in widely-used services

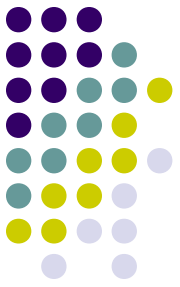
So, what do worms do exactly?



- Target Selection
 - Attack random hosts
 - Target specific subnets
- Exploit
 - Vulnerable services
 - Unpatched hosts
- Deployment Tactics
 - Aggressive
 - Subtle (harder to detect)

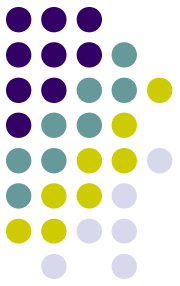


The Internet worm: A brief history

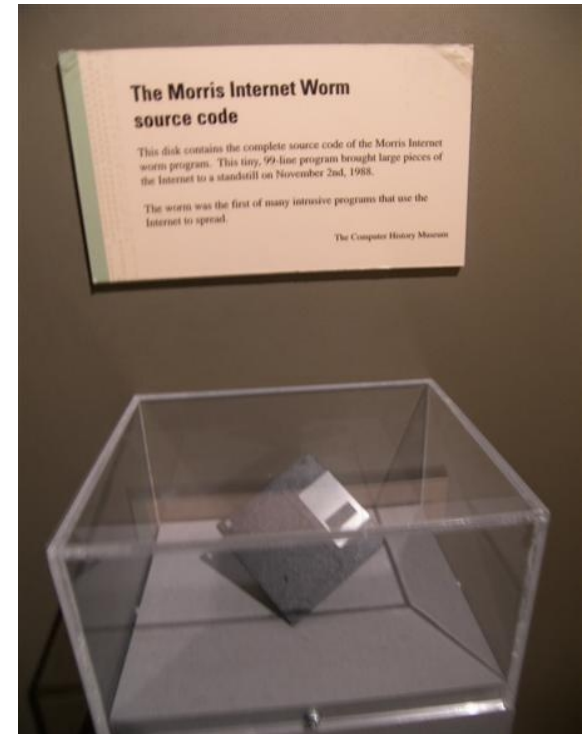


- 1988: Morris Worm
- 1999: Happy99, Melissa
- 2000: ILOVEYOU
- July 13, 2001: Code Red I v1
- July 19, 2001: Code Red I v2
- Aug. 4, 2001: Code Red II
- Sept. 18, 2001: Nimda
- 2003: Blaster, SQL Slammer (aka *Sapphire*)
- 2004: MyDoom, Witty, Sasser
- 2007: Storm Worm
- April 1, 2009: Conficker (aka *Downadun*)

Morris Worm (1988)

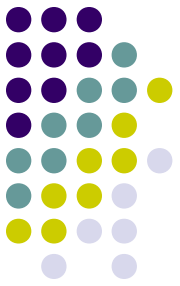


- Multi-vectored like Nimda
 - rexec, rsh
 - fingerd via buffer overflow that worked on VAX and caused core dump on Suns
 - UNIX sendmail debug mode
- Morris worm infected 6,000 of 60,000 hosts (5-10%)
 - Very large percentage compared to today's worms



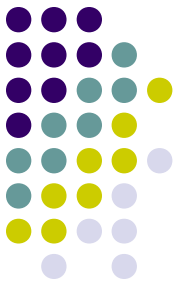
Disk containing the source code for the Morris Worm held at the [Boston Museum of Science](#)

Code Red (CRv1 July 13, 2001)



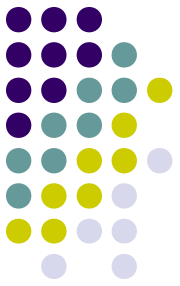
- Used a Microsoft IIS vulnerability to perform website defacement
- Spread by random scanning IP addresses
 - Launched 99 threads which generated the hitlist
 - A 100th thread did the defacement
 - Bug in random number generator seed hindered its spread

Code Red I (CRv2 July19, 2001)



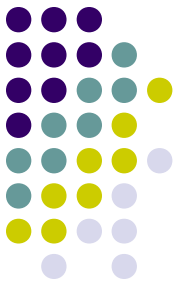
- Same codebase as CRv1, fixed the bug with random number generation
- Found to affect additional devices with web interfaces, such as routers, switches, and printers.
- After successful infection, the worm would check the date of the system.
 - Between 1st and the 20th: Generate a random list of IP addresses & try to infect them
 - Between the 20th and the 28th: Launch a DoS attack against www.whitehouse.gov
- Stopped spreading July 20, 00:00 UTC

Spread of Code Red

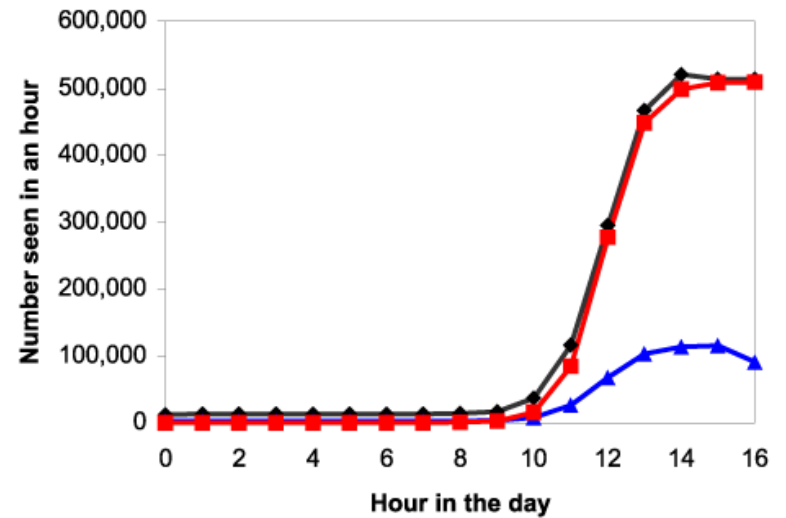


- Analytic model:
 - N = total number of vulnerable hosts
 - K = compromise rate, new hosts/host/hour
 - $a(t)$ = proportion of vulnerable machines compromised at time t
- Then:
$$N da = (Na) K (1-a) dt \qquad a(t) = \frac{e^{K(t-T)}}{1 + e^{K(t-T)}}$$
- Logistic equation
 - The rate of growth of epidemics in finite systems when all entities are equally likely to infect any other entity

Spread of Code Red con't

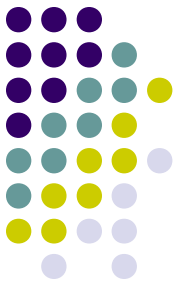


- Cas.org probe data
- Analytic model:
 - $K = 1.8$
 - $T = 11.9$
 - Maximum probe rate
510,000 scans/hour

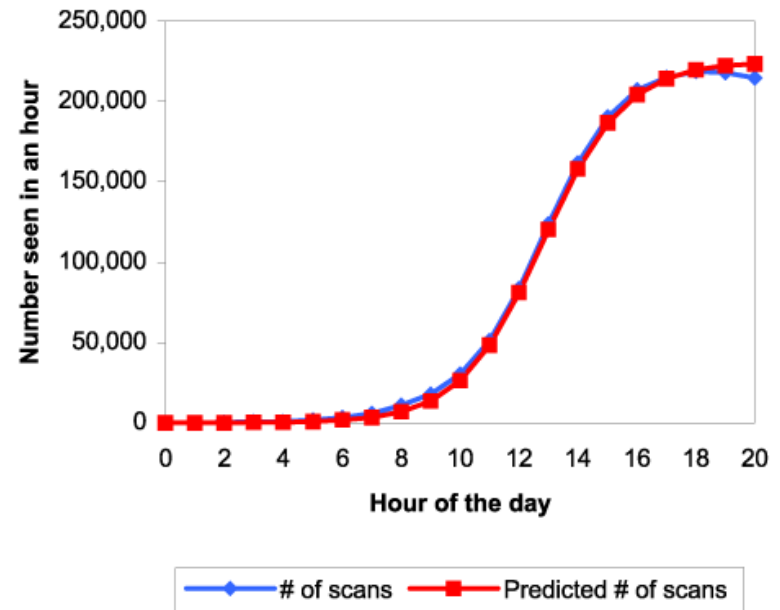


—◆— # of scans —▲— # of unique IPs —■— Predicted # of scans

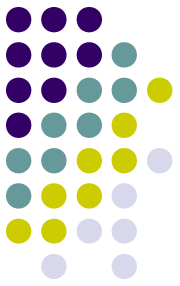
Spread of Code Red con't



- Reemergence
 - July 31 / August, 2001
 - Infected hosts with misconfigured clocks to blame
- Analytic model:
 - $K = 0.7$
 - Vulnerable hosts were less than 40%

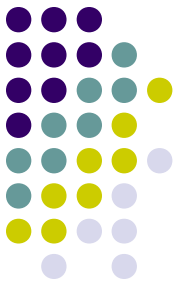


Code Red II (August 4, 2001)

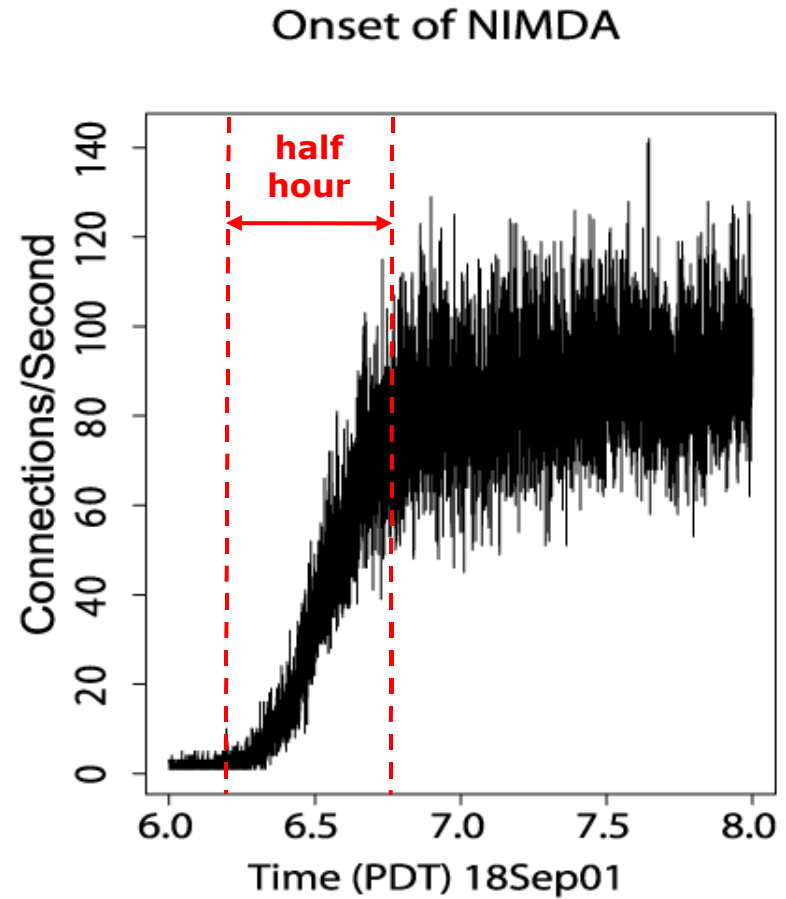


- Used same IIS vulnerability as Code Red I but installed root backdoor instead
- Fixed random IP generator
- Localized scanning
 - Class B address space 3/8 probability
 - Class A address space 1/2 probability
 - Whole Internet address space 1/8 probability
- Quicker infection
 - Similar IP addresses are close together
 - Bypass external firewall

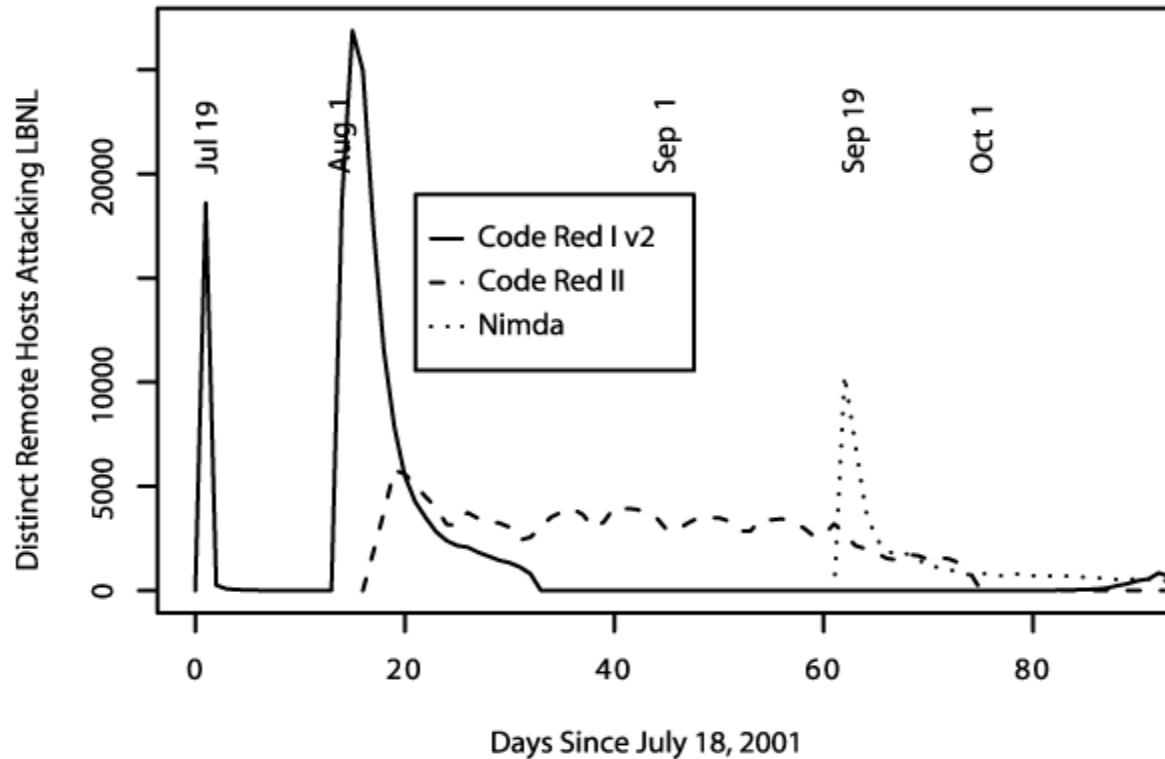
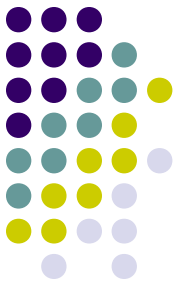
Nimda (September 18, 2001)



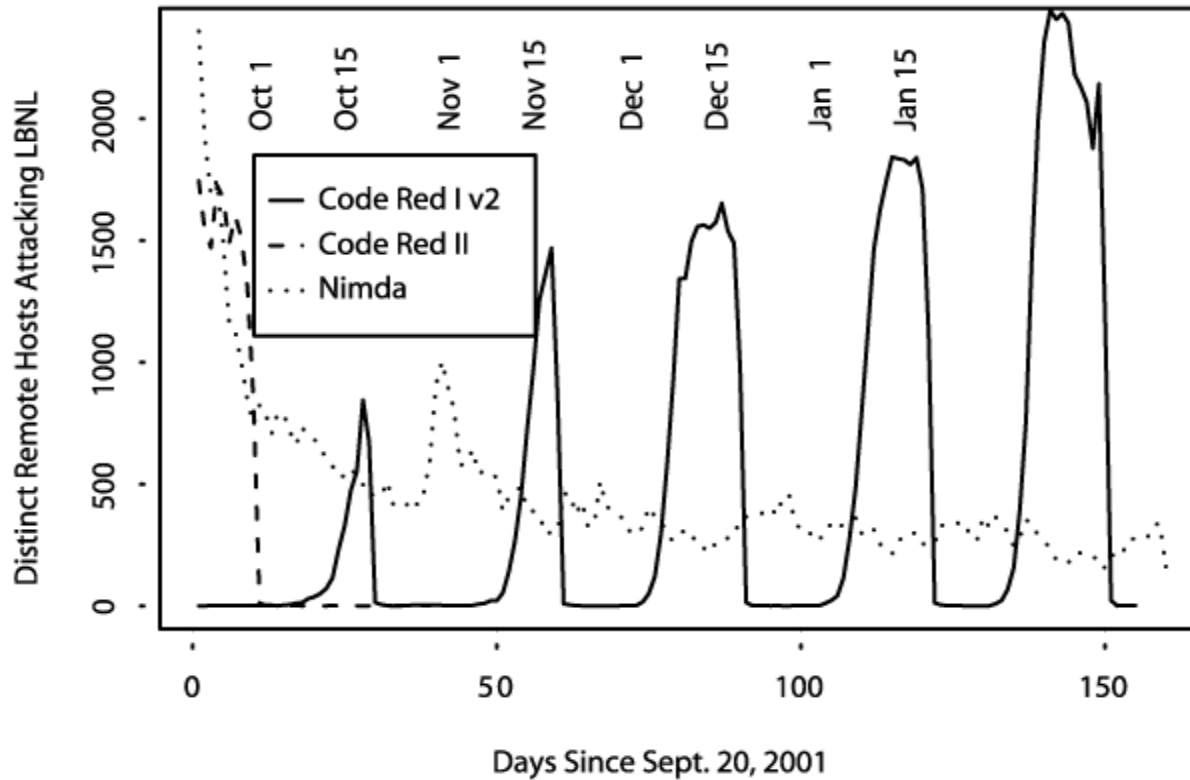
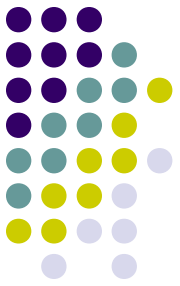
- Multi-vector worm
 - IIS vulnerability
 - Email
 - Network shares
 - Infect webpages
 - Scan for Code Red and Sadmind backdoors
- Almost no probing to 100 probes/sec in half an hour



Onset of Code Red I v2, Code Red II, and Nimda



The endemic nature of Internet worms

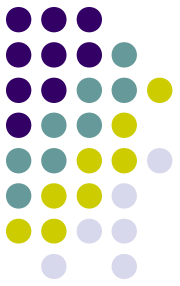


How to spread faster



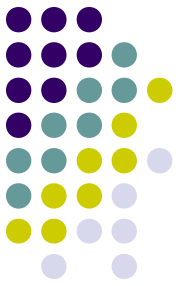
- Hit list scanning
 - Faster startup
- Permutation scanning
 - Limit redundant scans
- Topological scanning
- Internet scale hit-lists

Hit-lists



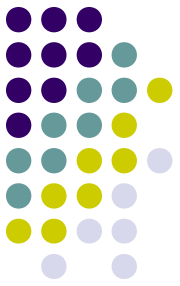
- Collect a list of potentially vulnerable machines
 - 10,000 to 50,000, ideally with good network conditions
 - Spread, splitting the list on each infection
- How to make the list
 - Stealthy scans
 - Distributed scanning
 - DNS searches
 - Spiders
 - Public surveys

Permutation scanning

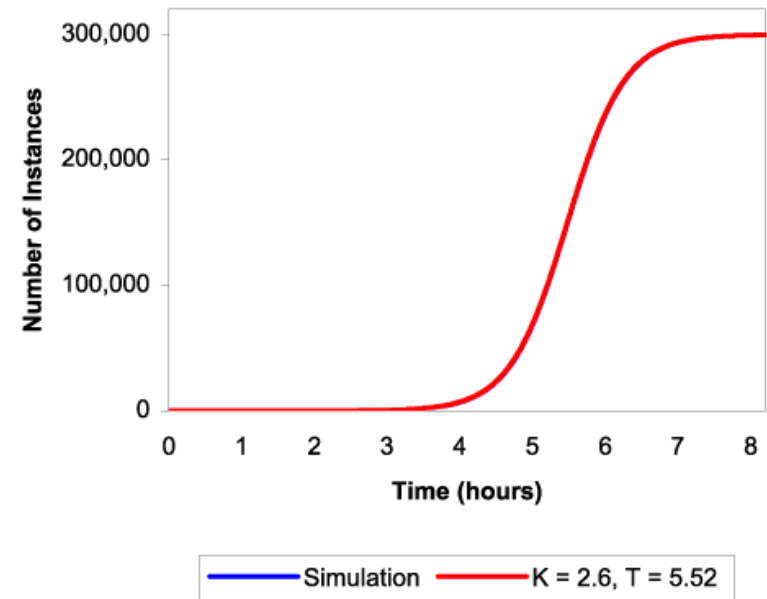


- Eliminate redundant scanning by partitioning searches
- All worms share a common pseudo random permutation of the IP address space
- Start scanning from your point in permutation
 - If machine in sequence is infected, randomly choose new point to scan and increment counter
 - Else infect computer and then scan
- Worms infected by permutation scanning would stats at a random point
- Stop scanning when counter == SCAN_LIMIT

Warhol Worm

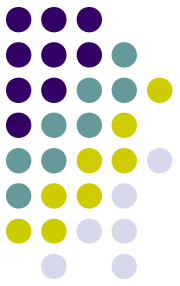


- A combination of Hit-list and Permutation Scanning
 - Hit-list improves initial speed, quickly achieving a set base
 - Permutation scanning keep the worm's infection rate high for a longer period
- Achieves 99% infection in around 15mins
 - 10 scans/sec, 300,000 hosts

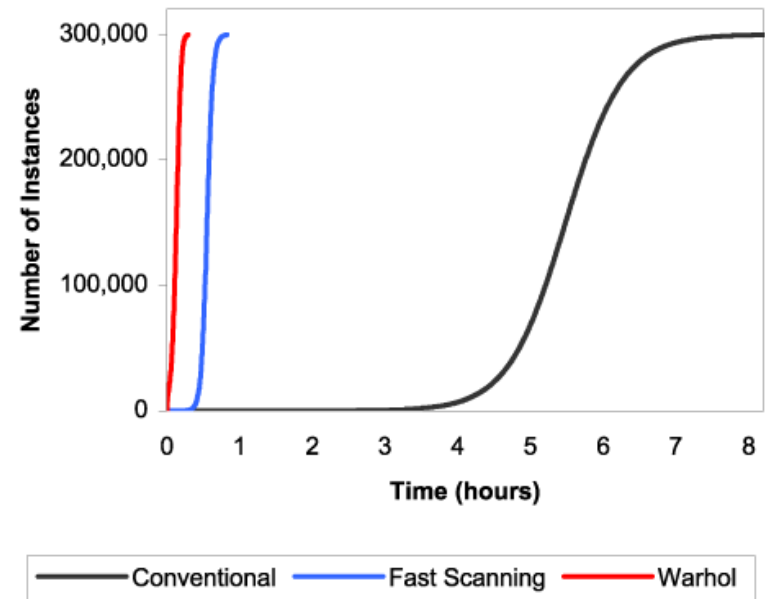


"In the future, everyone will be world-famous for 15 minutes."
-- Andy Warhol

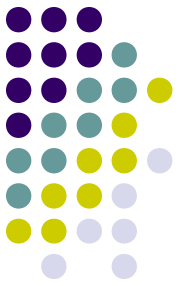
Warhol Worm con't



- Conventional
 - 10 scans/sec
- Fast scanning
 - 100 scans/sec
- Warhol
 - 100 scans/sec
 - 10,000 entry hit-list
 - Permutation scanning
 - Gives up when count = 2
- SQLSlammer (2003)

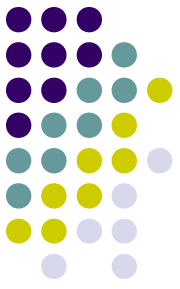


Topological scanning



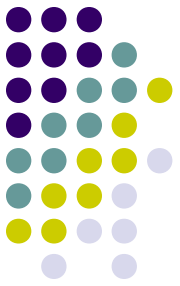
- Use information contained on the victim machine to find new targets
- Email worms
 - MyDoom used Google, Yahoo, Altavista and Lycos
- Internet cache for URLs
- P2P systems for peers IP addresses

Flash Worm

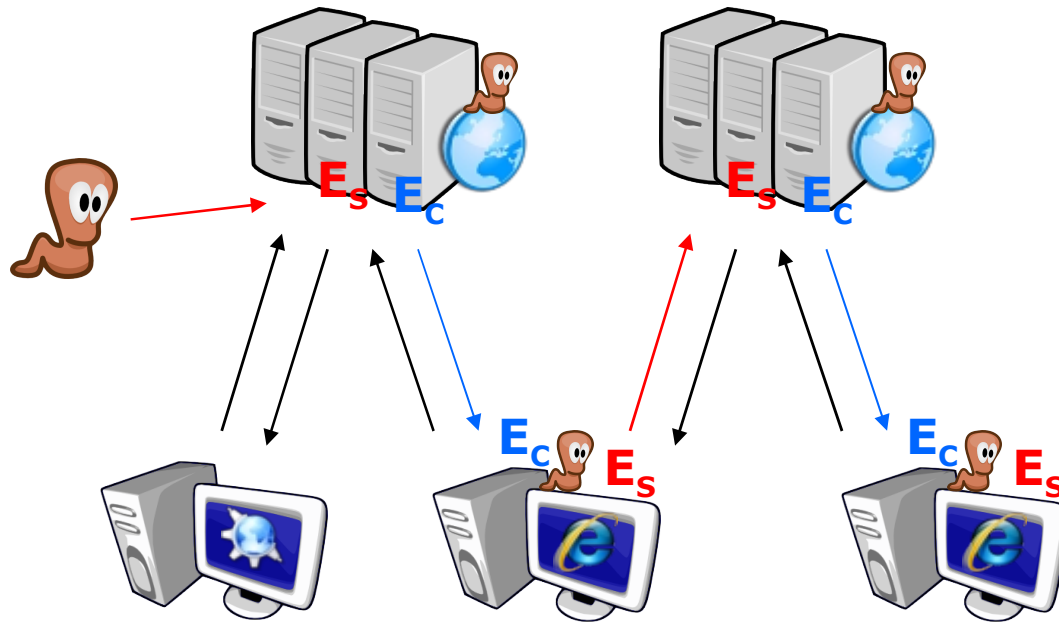


- Idea: start with a hit list that contains all vulnerable hosts
 - Brute-force scanning of the entire address space
 - 2hr with OC-12 / STM-4x (622Mbps)
- Need high bandwidth link
 - 9 million addresses, sorted, differenced and gzipped = 13MB
 - Initial copies of the worm have hit-list
 - Hit-lists could be divided into chunks and distributed on known high bandwidth servers
- Capable of infecting most vulnerable servers in less than 30secs

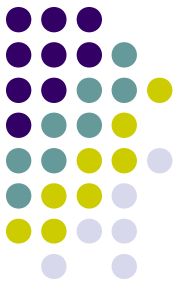
Contagion Worms



- Slowly spreading worm to avoid detection
- No change in communication patterns

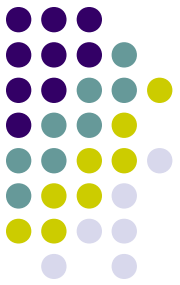


Exploiting P2P networks



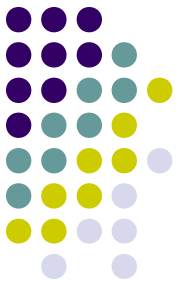
- Likely only need one exploit, not a pair
- Often, peers running identical software...
- Tend to have rich interconnection patterns to piggyback on
- Often used to transfer large files
- Not mainstream - less vulnerability assessment, monitoring
- “Grey” content – users less likely to mention unusual activity

Updating Worms



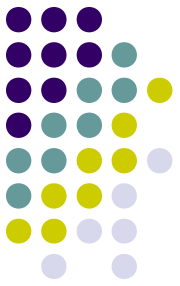
- Distributed Control
 - Each worm could have a subset of infected hosts
 - Each command can be signed and then sent to other copies of worm
 - Received commands can be verified and then forwarded
- Programmable Updates
 - Dynamic code loading
 - Flexible language combined with a small interpreter

Cyber Center for Disease Control



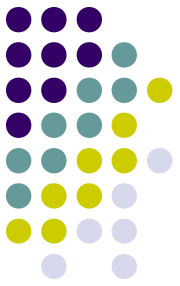
- Identify outbreaks
- Rapid pathogen analysis
- Fight infections
- Anticipate new vectors
- Proactively devise and deploy detectors
- Resist future threats

Observations



- Infection from a new exploit (0-day) can happen fast! (or even an old exploit)
- A well-written virus/worm without any “large” errors could do some real damage
- Some potential “solutions”...
 - Distributed Firewalls
 - Traditional IDS
 - Honeypots

Lots of things to work on



- Buffer Overflows still prevalent
- Passwords still poorly chosen
- People with a lot less skill than Robert Morris have done much more damage
- Misconfigured policies
- Complexity is an evil to security
 - Morris used a sendmail vulnerability
- People don't keep up with patches (even on servers)
 - Security Holes ... Who Cares?

[USENIX security 2003, <http://www.usenix.org/events/sec03/tech/rescorla.html>]

The End

- Questions?

