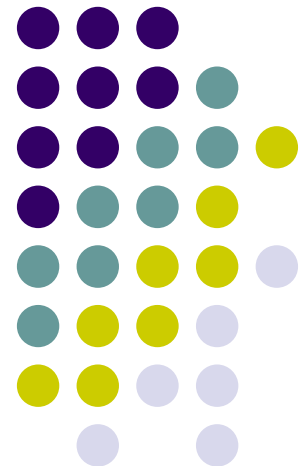
How to 0wn the Internet in Your Spare Time

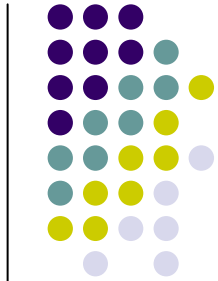
S.Staniford and V. Paxson and N. Weaver, in Proceedings of the 11th USENIX Security Symposium, pages 149-167, San Francisco, CA, August 2002.



Presented By:

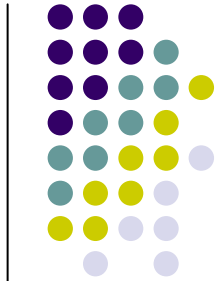
Mohamed Hassan

Background



- What is a Computer Worm?
- is a self-replicating computer program. It uses a network to send copies of itself to other systems and it may do so without any user intervention. Unlike a virus, it does not need to attach itself to an existing program.

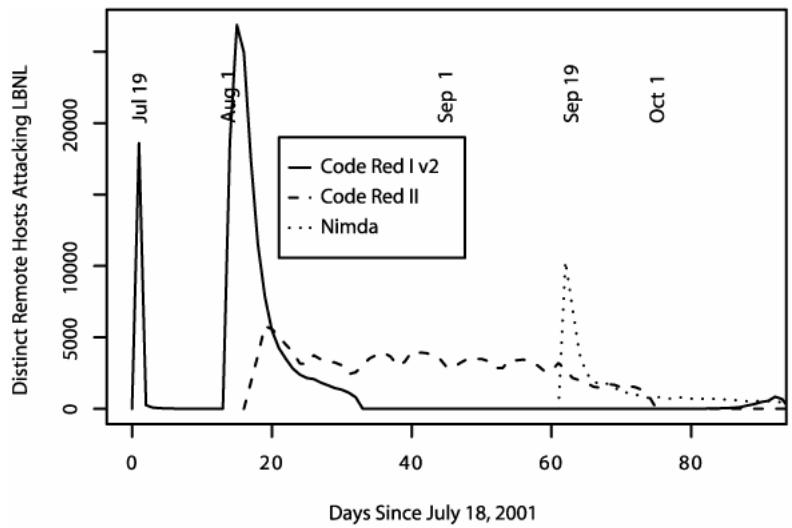
Background



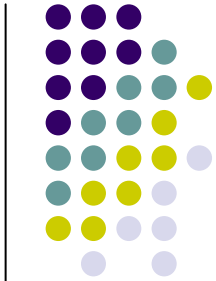
- History
- 1978: Shoch and Hupp originally designed the worm to find idle processors on the network and assign them tasks improving the 'CPU cycle use efficiency'
- November 2, 1988 : Morris worm, the first internet worm, exploited a hole in sendmal and infecting several thousand computers.
- 2001: Ramen worm (RedHat), Code Red worm I (IIS), Code Red worm II, Nimda
- 2003: SQL Slammer(MS SQ L), Blaster worm (Windows RPC), Sobig worm (Email)
- 2004: MyDoom (Fastest spreading via Email), Sasser worm (Windows)
- 2005, 2006: Samy worm (MySpace profile defacement), W32 Nyxem (mass mailing worm that stops security related applications)

Background

- What a worm can do
  - A worm can compromise million hosts in few hours
  - Launch untraceable DDOS
  - access any sensitive information on these hosts
  - spread confusion and disruption by corrupting information or sending out false or classified information
  - Can be used in warfare between nations or in the service of terrorism

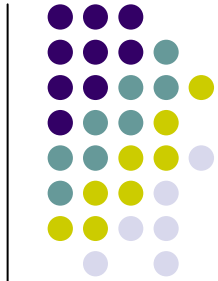


Related Work



- Code Red I
  - On June 18, 2001 eEye released information about a buffer-overflow vulnerability in Microsoft's IIS webservers
  - Version 1 first observed on July 13th, 2001 by Ryan Permeh and Marc Maiffret of Eeye Digital Security. The first version had a bug in the implementation
  - Version 2 outbreak on July 19 th, 2001. Same code base but the bug was fixed

Related Work



- How it works?
  - Send an HTTP request to the victim host and exploit the .ida IIS vulnerability which allows the worm code to execute
  - Start 100 threads of the worm
  - The first 99 threads are used to spread the worm by attacking randomly generated IPs (Bug in ver1!)
  - Thread 100 checks if the running system language is US English
  - If yes, the default homepage will will be changed to a message that says “Welcome to http://www.worm.com !, Hacked By Chinese!”
  - Each worm thread checks for c:\notworm. If this file exists stop the attack
  - Each thread checks If the time is between 20:00 UTC and 23:59 UTC then will proceed to attack www.whitehouse.gov

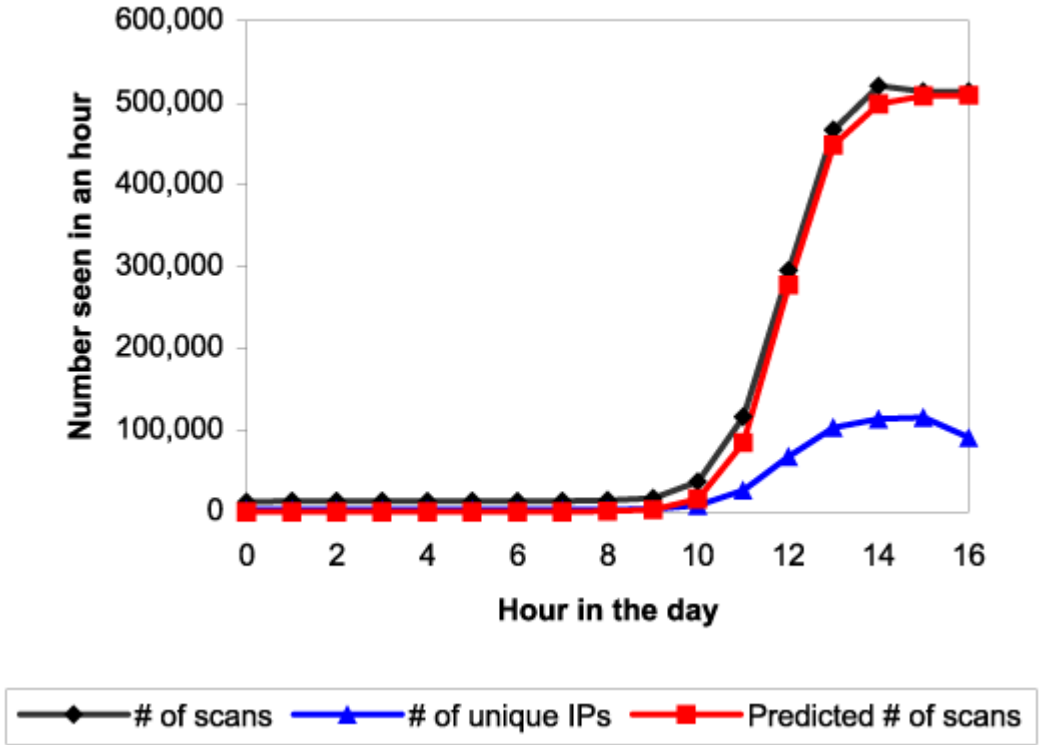
Related Work

- Code Red I Analysis
  - The worm uses a Random Constant Spread (RCS) model.
  - Each worm generates random IPs of the host it will attack next.
  - The Model
    - Given N number of vulnerable machines interconnected hosts
    - a the is the proportion of vulnerable machines N which have been compromised.
    - t is the time in hours
    - T is a time which fixes when the incident happens.
    - K the number of vulnerable hosts which an infected host can find and compromise per hour at the start of the incident
    - Then

$$a = \frac{e^{K(t-T)}}{1 + e^{K(t-T)}},$$

Related Work

- Comparing the model to real data

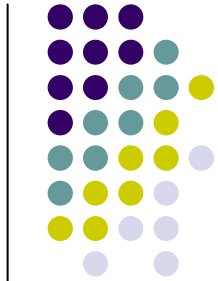


K=1.8

T=11.9

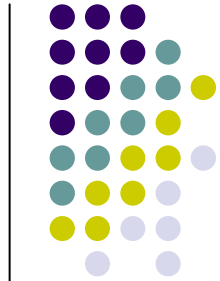
Hourly probe rate data for inbound port 80 at the Chemical Abstracts Service during the initial outbreak of Code Red I on July 19th, 2001.

Related Work



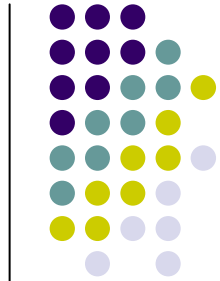
- Localized Scanning Worms
- Code Red II (Aug 4, 2001)
  - Used same vulnerability as code red I
  - installed a root backdoor allowing unrestricted remote access to the infected host
  - It used local scanning by generating random IPs with probability 3/8 from class B (/16) addresses and ½ from its own network class A addresses and 1/8 from the whole internet
  - Better practice because servers with similar IPs are close together in network topology
  -

Related Work



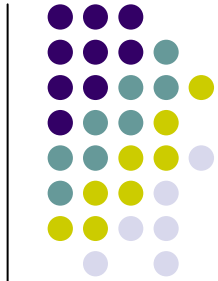
- Multi-vector worms
- Nimda (Sep 18, 2001)
  - used at least five different methods to spread itself
    - Attacking same vulnerability in IIS servers
    - Bulk mailing to email addresses found the infected machines
    - Copying itself across open network shares
    - Adding exploit code to webpages on the server
    - Looking for backdoors left behind Code Red II
  - Was able to go through firewalls as it used email messages which is not usually scanned by firewalls.

Methodology



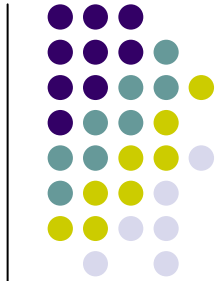
- Problem with previous models
  - The worm takes a long time to “get of the ground”. It takes few hours to infect 10,000 hosts
  - inefficiency of random scanning: many addresses are probed multiple times
- Faster worm models
  - Warhol worm using hit-list and permutation scanning
  - Flash worm
  - Contagion worm

Methodology



- Hit-list scanning
  - The worm author collect a list of 10,000 to 20,000 potentially vulnerable hosts
  - When the worm infects a host it divides the list into halves and send one half to the recipient worm
  - How to build a hit-list?
    - Stealthy scan for months. (low profile hard to detect)
    - Distributed scanning using already compromised hosts
    - DNS searches (For Example, search MX records for mail servers)
    - Spiders that use web crawlers similar to search engines
    - Public surveys from http://www.netcraft.com/survey/
    - Just list for broadcasts (some worms broadcast infected hosts)

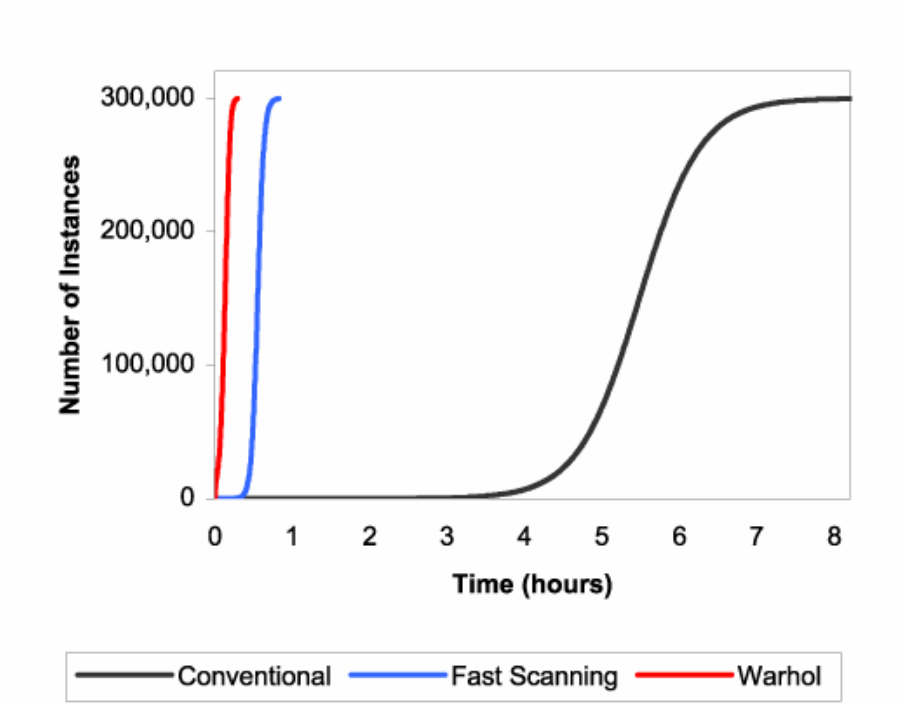
Methodology



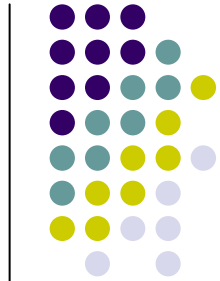
- Permutation scanning
  - Used to generate victim IPs
  - All worms share a pseudo random number
  - Any worms infects a machine in a list (machine x) it start scanning just after its point in permutation (machine x+1)
  - If a worm W scans a host in the list and find that host infected it means that W' is already working on that sequence of the permutation. So, W start on a different sequence
  - This model Self-coordination eliminating duplicate scanning. Therefore, it increases the worm efficiency and the rate of infecting distinct host in the address space
  - If a worm finishes a certain sequence and can sleep for a while and wake up trying a different list to infect more servers.

Methodology

- Warhol worm
  - A Warhol warm can employ the 2 previous models to attack 300,000 in less than 15 minuets

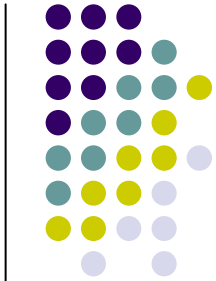


Methodology



- Flash worm
  - The worm contains a list of all web servers on the internet (12.6 million servers=48 MB address list)
  - The list is divided to n blocks and each block is sorted by servers bandwidth
  - The worm attacks the top servers of each block
  - If it succeed it distribute the list block on child worms and this process continues like a tree
  - To avoid a dead branch of the tree the list block has cross listed IPs
  - To infect 300 million hosts with n=10 the tree will be 7 layers in depth

Methodology

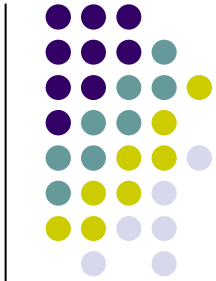


- Stealth worms-Contagion
  - Like human viruses, they spread slowly so they are hard to detect but they surreptitiously attack their hosts
  - Each worm has 2 exploits : Ec and Es.
  - The worm starts from the Server or client
  - When A client surfs the server Ec and Es, the worm tries the Ec exploit on that client if it works it transfers both Ec and Es to the client machine.
  - When an infected client visits a server which is vulnerable to Es. The worm transfer Ec and Es to the server.
  - It can work very well in P2P applications because a P2P application is both client and server so no need to find 2 exploits
  - It takes longer time but hard to detect because it is hidden in the client traffic or P2P trafic

Methodology

- Update and Control
  - Some worms install remote control code on the hosts to allow for a remote DDOS commands
  - Worms can download updates from WebPages
  - Worms can update each other in a distributed model by doing permutation probing similar to permutation scanning and communicate the updates using encrypted channels.

Results



- **Center for Disease Control**
- **Tasks**
  - Identifying outbreaks
    - Gathering field information from Internet, cellular, private (Privacy !!!)
    - Detecting worms based on traffic patterns
  - Rapidly analyzing pathogens
    - Develop state of the art program analysis tool
    - Provide laboratories stocked with VMs for simulating worms
  - Fighting infections
    - Develop a mechanism to propagate signatures that can detect worms.
    - Building agents that utilizes these signatures to terminate or isolate worms
  - Anticipating new vectors
    - analyze the threat potential of new application
    - foster the development of application analysis module

Conclusion

- The authors have introduced models of worms that can spread in few minuets or less.
- This models can create a threat for the safety of the internet.
- There is an urgent need for control centers that fight the spreading of worms.

Any Questions

- Ask Bill