It Is Time To Open Emulab Account
- Go to http://www.emulab.net
- Sign up with existing project CIS859
- Go to Documentation link and read instructions how to use Emulab
- If you don’t know yet what is your project topic come and talk to me before the deadline

How to Own Internet in Your Spare Time
Stuart Staniford, Vern Paxson and Nicholas Weaver
Presented by: Jelena Mirkovic

What is This Paper About?
- Self-propagating malicious code (worms and E-mail viruses)
  - Discusses their characteristics – mostly how they spread
  - And what bad things they can do
  - Also suggests what we can do to defend against them

What Can You Do with 1,000,000 Hosts?
- Perform a very distributed, large-scale DDoS attack
  - Why is this much worse than an attack with 1,000 hosts?
- Access sensitive information
  - Credit card numbers, E-mail messages, addressbooks, corporate information
  - You could even find information by blindly searching around – such as latest Windows code or secret military information
- Damage hosts
- Publish sensitive data or simply create confusion

Additional Weight on the Problem
- Internet has become an essential part of many critical services
  - It is certainly undesirable that anyone can take control over it

Can You Own 1,000,000 Hosts?
- Yes, easily with a worm that exploits a vulnerability in a popular service
  - There must be more than 1,000,000 hosts running Microsoft Outlook
- And with a worm that spreads fast
  - So they don’t manage to patch
- E-mail virus wouldn’t spread as fast but it would be able to go behind many firewalls
Code Red (CRv1)
- Released July 13th, 2001
- Attacked Microsoft IIS Web servers
- Each copy of the worm forks 100 threads to spread to 99 other hosts (1 thread defaces a Web page)
- Original code had a bug
  - Each of 99 threads would try a particular (always the same) IP address sequence – overall 99 sequences tried
  - This resulted in linear spread

Code Red I (CRv2)
- 6 days later the bug was fixed and another version of Code Red released
- Now all 100 threads were spreading the code much faster
- Instead of Web page defacement, code was carrying DDoS tool targeting the IP address of www.whitehouse.gov
- This also had a bug – IP address was hardcoded
- What can we learn from this change of goal?
- What can we learn from the presence of bugs?

Modelling Random Constant Spread
- If currently we have \( a \) compromised machines, how many will we have at next time interval?
  \[ N_{da} = (Na)K(1-a)dt \]
- This gives a solution:
  \[ a = \frac{e^{K(t-T)}}{1 + e^{K(t-T)}} \]
- Exponential for small \( t \), 1 for large \( t \)
Modelling Random Constant Spread

- On the average one copy was infecting 1.8 other servers per hour
- Not so much
- Still it took less than a day to infect all vulnerable machines
- When worm resurfaced 11 days later, infection rate was down to 0.7
- Some machines got patched

New Advances in Code Red II

- Released August 20th, 2001 (roughly a month after CRv1)
- It used the same vulnerability as Code Red
  - Installed a backdoor on the machine
- It used localized scanning – giving preference to addresses on local subnet
  - Quicker – vulnerable hosts are close
  - Many hosts have the same vulnerability
  - Avoid firewalls

New Advances in Nimda

- Released September 18th, 2001 (roughly 2 months after CRv1)
- It used five different spreading techniques (multi-vector)
  - Infecting vulnerable IIS servers
  - Through E-mail as an attachment
  - Copying itself across network shares
  - Adding exploit code to Web pages
  - Scanning for open backdoors left by previous worms
- It is still not known what was the goal of Nimda

How to Make Worms Spread Faster

- Hypothetical new techniques:
  - Hit-list scanning
  - Permutation scanning
  - Topologically aware worms
  - Internet scale hit-lists

Hit-list Scanning

- Time to infect first 10,000 hosts dominates the infection
- If we feed the worm a list of 10,000 vulnerable machines it will take off fast
Permutation Scanning
- All copies share the same IP address space permutation
- Essentially the same sequence containing all IP addresses
- Machines infected through non-permutation scan start scanning from their own index in the sequence
- Machines infected through permutation scan start from a random point in the sequence
- This also do machines that encounter an infected machine
- Another optimization – partition sequence similar to hit-list scanning – partitioned permutation scan
- Even resilient to fake “infected” replies

Warhol Worm
- Artificial worm
- Uses permutation and hit-list scanning
- Infects all vulnerable hosts within an hour

Topological Scanning
- Gather a list of IP addresses off an infected machine
- E-mail viruses use this strategy
- Peer-to-peer applications
- Web servers
- Silent spread, alive and mostly vulnerable machines

Flash Worms
- Assembling an Internet-size hit-list of all vulnerable machines
- Perform an Internet-wide scan – could be completed in 2 hours
- A large list, but could be initially divided into $n$ blocks and handed off to $n$ children to propagate further
- Overall, the worm would infect vulnerable population in 30 seconds or less
- Implication:
  - We need very, very, very fast automatic defenses

Stealth Worms
- Spread slowly and carefully so as not to attract attention
- Assume that the attacker has two exploits: Ec that infects a Web client and Es that infects a Web server
- Clients visiting infected server will carry worm with them
- So will servers visited by an infected client
- Simple, non-suspicious infection pattern
Stealth Worms
- Similar could be done with peer-to-peer services
  - Uniform population
  - Highly interconnected
  - Usually residing on desktops not servers
  - Lot of binary/suspicious content already circulating
  - Huge population
    - 5.10 million established connections at a given university per day!
    - 9 million distinct hosts per month

Possible Advances in Worm Design
- Distributed control
  - So far, a few masters or an IRC channel
  - Each copy knows of a few other copies (like P2P network) and can talk to them through an encrypted channel
  - Author sends signed commands to one copy, they then spread further and are executed
- Programmatic updates
  - So far, being downloaded from Web pages
  - New updates could spread through worm P2P network
  - If worm uses an interpreter, a source code could be sent

Center for Disease Control
- Similar to CDC for biological viruses
  - Identify outbreaks
  - Rapidly analyze pathogens
  - Fight infections
  - Anticipate new vectors
  - Proactively design detectors for new vectors
  - Resist future threats

Identifying Outbreaks
- So far, through a few mailing lists
  - Monitor and detect worm spread
    - Possibly within the core – this raises privacy issues
  - Quickly spread the word through multiple channels
    - For fast-spreading worms this may not be good enough

Rapidly Analyzing Pathogens
- Understand
  - How worm spreads
  - How to detect/stop it
  - What else it does in addition to spreading
    (e.g., plant DDoS code)
- Spreading is easy, but the rest requires code analysis and code may be obfuscated in weird ways
  - Need a lot of machines with diverse applications to plant the code and see what it does

Fighting Worms
- Quickly develop and propagate signatures
  - We would need a P2P network of widely distributed agents to efficiently spread the signature
Anticipating New Vectors
- Track new/popular applications
- Analyze their code
- Analyze how homogeneous are servers/clients
- Analyze native application communication patterns

Proactively Devising Detectors
- Analyze identified new vectors
- Their communication patterns
- Their code characteristics
- With a goal to devising fast detectors

Resisting Future Threats
- Devising protection mechanisms so that Internet is not so susceptible to infections
- Good programming practices
- Sandboxing of applications
- Diversity
- Limit traffic/connection rate

Information Sharing
- Should there be a Web site for information sharing during an outbreak?
  - Currently companies compete to analyze the code first
  - Privacy concerns
  - Security concerns
  - Aiding an attacker
  - Needs human supervision
- How about an international information sharing
  - Definitely useful but needs human supervision

Let’s Rate this Paper!
- How important is this research?
- How good are their ideas?
- How well is the paper written?
  - Presentation/organization
  - Writing style
  - Appearance
- Is there something missing?

Any Ideas You Might Have