Worm Defenses

Advanced Topics in Network Security
CS239

Everett Anderson
May 3, 2004

What can attackers do?

- Install back doors and execute arbitrary code
  - Launch DDoS attacks
- Gather sensitive information

...at Internet scale!

Means of Attack

- E-mail and other user level network applications
  - Attachments, executable file extensions
    - Good for perimeter penetration
- Network shares/services
  - Default accounts
  - Unrestricted local network access
  - Buffer overflows

Examples

Name	Exploits
Code Red	IIS buffer overflow
Nimda	IIS Unicode “dotdot” attack, net shares, e-mail (and browser download), other worm backdoors
Slammer	Buffer overflow in MS SQL Server 2000
Blaster	DCOM RPC buffer overflow

Complicating Factors

- Infection rate
  - easy to scan for hosts, coverage depends only on rate
- Hackers quickly incorporate new ideas
  - vulnerabilities, countermeasures
- Patch lag
- Multiple attack methods
- Worm updates
- Homogeneity

Increasing Scan Rate

- Localized scanning
- Topological scanning
- Hit lists
  - Partial
  - Full (“Flash Worm”)
- Permutation scanning
- Contagion

Findings from “Internet Quarantine...”

- Must have automated containment system (minute level reaction time)
- Content filtering can contain more aggressive worms than blacklisting
- Effective containment requires nearly full deployment
  - Ideal system of total deployment contains Code Red-like 100 probes/sec worm to 1% infected in 24 hours using content filtering and 18 minute reaction time
  - If only 100 top ISPs, infection is at least 18% at 24 hours even with reaction times of less than 1 second

Solutions?

- Prevention
  - Usual: Up to date systems, DMZs, user education
  - Openness of implementations to find exploits
- Treatment
  - Usual: Scanners
- Containment
  - Only real solution?

Honeyd Framework

- Fewer false positives since honeypots should have no legitimate incoming traffic
- Automatic signature detection
- Disinfect connecting hosts
    - Prepare disinfection/immunization code at time of patch release

Microsoft Shield Project

- Users don't want to apply patches due to possible instability (or are unaware of problems)
- Install automated “shields” until patches are installed, blocking exploits

Discussions

- If it's true that most attacks are based on vulnerabilities revealed when patches are released, how should a company respond to a newly found hole?

Discussions

- Imagine a worm defense system with the following properties:
  - Full deployment
  - Ingress and egress filtering of known worm traffic
  - Hierarchical alerts from a “CDC”
  - Active immunization
- Is that enough? How do we determine what's enough?
- Zero-day exploits? Automatic signature generation?

Discussions

- In “Cooperative Response Strategies for Large Scale Attack Mitigation,” they proposed a back-off mechanism to allow good traffic to resume and an alert threshold before taking action.
- Does back-off make sense?
- Wouldn't an alert threshold ensure infection?