Intelligent Malware Defense for Insider Threats in Mobile Networks

Errin W. Fulp and Joseph V. Antrosio

Department of Computer Science

nsg.cs.wfu.edu

April 28, 2005

Malware

- Interconnected networks offer many benefits
  - Also enabled an increasing number of security threats
- Malware is unwanted software that exploits flaws
  - Worm is the most prevalent and dangerous
  - No human interaction is required
- *Why are worms successful?*
  - Homogeneous software and high-speed networks
- Worm stages
  - Target selection
  - Exploitation
  - Infection

Mobile Networks

- Security threats are more difficult to defend
  - Users can easily bypass standard security devices
  - Internal threat becomes more important
  - Heterogeneous environment, which is difficult to control
  - Personal firewalls are not feasible
- User authentication offers no protection
  - Does not authenticate the security of the machine
  - Authenticated user can compromise the internal network
- Administrator has limited control
  - Cannot patch vulnerable systems
  - Cannot enforce compliance

Mitigation Approaches

1. Prevention
   - Prevent vulnerabilities via better engineering
2. Treatment
   - Fix vulnerabilities, patch software
   - Time to develop patch may be too long
   - *What do you do in the interim?*
3. Confinement
   - Contain malware via software/infrastructure
   - Allows time for proper patching

While prevention and treatment are important, they are not sufficient. Confinement is the most promising

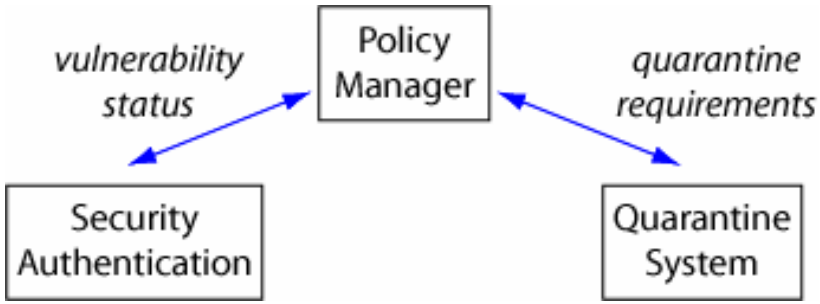
Confinement Strategies

- Content Filtering (malware oriented solution)
  - Database of signatures applied to all traffic
  - If fingerprint matches then packet is dropped
  - Scalable since individual hosts are not identified
  - *What if fingerprints change?*
  - *Requires constant filtering of all traffic*
- Containment (host oriented solution)
  - Identify infected hosts
  - Drop traffic associated with infected ports
  - Does not rely on fingerprints
  - All malware associated with the vulnerability is managed

Desired System Requirements

- Detection characteristics
  - Preventative not reactive
  - Not dependent on signatures
  - Not dependent on infected machines
  - No client software
- Containment
  - Immediate containment
  - Suitable for mobile and high-speed networks
  - Defend before acquisition phase
  - Internal defense

Adaptive Malware Containment



- Contain vulnerable and infected machines
  - Focus on vulnerabilities, W*hy?* Addresses malware variants and other exploits
  - Provide secure access and maximize system utility
  - Internal and external defense with no host software
- System consists of three parts
  - Security authentication
  - Containment system
  - Policy manager

Security Authentication

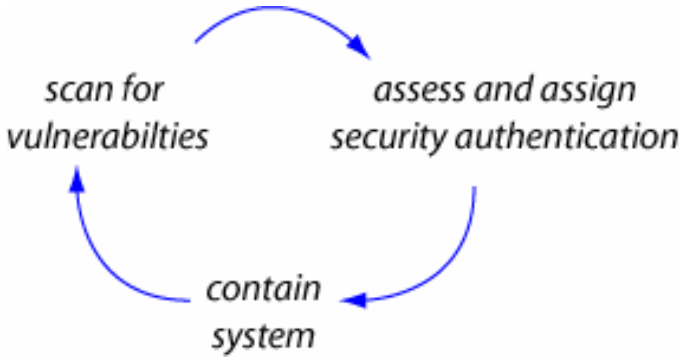
- Remotely detects system vulnerabilities
  - Scans for services and possibly perform mock exploit
  - Returns vulnerability status (security authentication)
    - Want to differentiate vulnerable from infected
  - User authentication can augment security authentication
- Done periodically since system status changes
  - Results are given to the policy manager
  - Authentication determines appropriate containment
  - *Contain individually, one group, or as multiple groups?*

Containment System

- Prevents infection and/or infecting others
  - Containment done before acquisition stage
  - Does not require a successful attack
- Isolation using only the network infrastructure
  - Performed using OSI layers 2 and 3 (MAC and network)
  - No host software, applies to heterogeneous networks
- Provide containment and maximize utility
  - Protect the vulnerable and disable the infected
  - Machines operate safely until patched/updated
  - Can be used to safeguard defense system components

Policy Manger

- Directs system components, performs three tasks



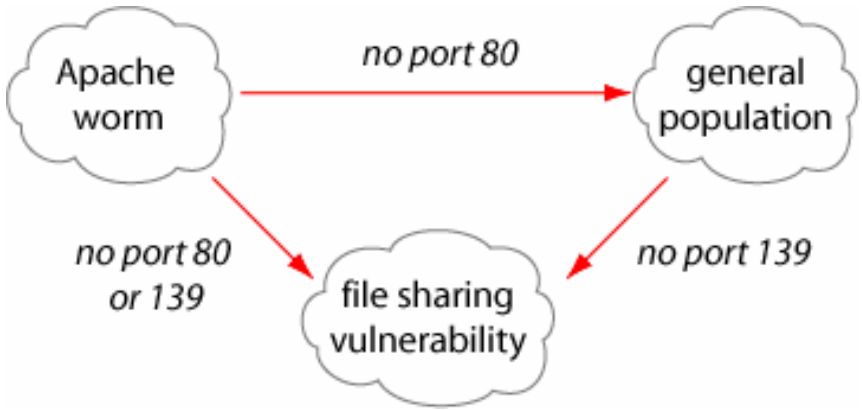
- Consider a new machine entering the network
  - Machine placed in highly restrictive containment
  - Scanning performed, determine security authentication
  - Policy manager assigns appropriate security group
  - Done periodically, allows for changing systems
  - Depends on the **security policy**

Security Policies

- A simple policy has only two access types
  - Full access (no vulnerabilities or malware present)
  - Severely restricted access (to patch servers)
- Blacklisting
  - Every vulnerable/infected machine protected individually
  - As a result, security group of one (not scalable)
- Security Groups
  - Group machines with same security authentication
  - Scalable approach compared to blacklists
  - *Not as secure since group can be at risk…*

Security Groups

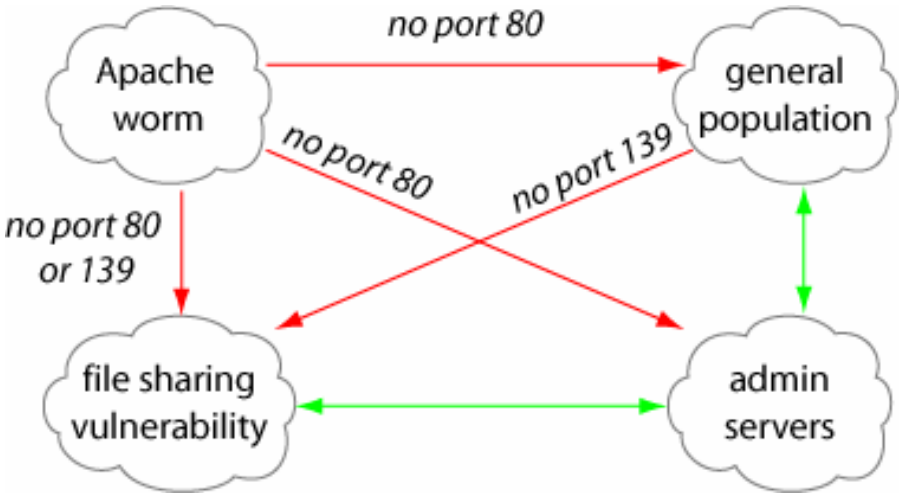
- Security group is a collection of similar machines
  - Systems with same security authentication
  - More scalable, for example requires fewer ACL entries
- Consider the following groups
  - Apache worm carriers, file sharing vulnerable, and general population



- Can groups can be too conservative

User and Security Authentication

- Combine security and user authentication
  - Allows more groups interaction (trust associations)
- Consider dividing general population



- Management becomes more difficult

Security by Contract

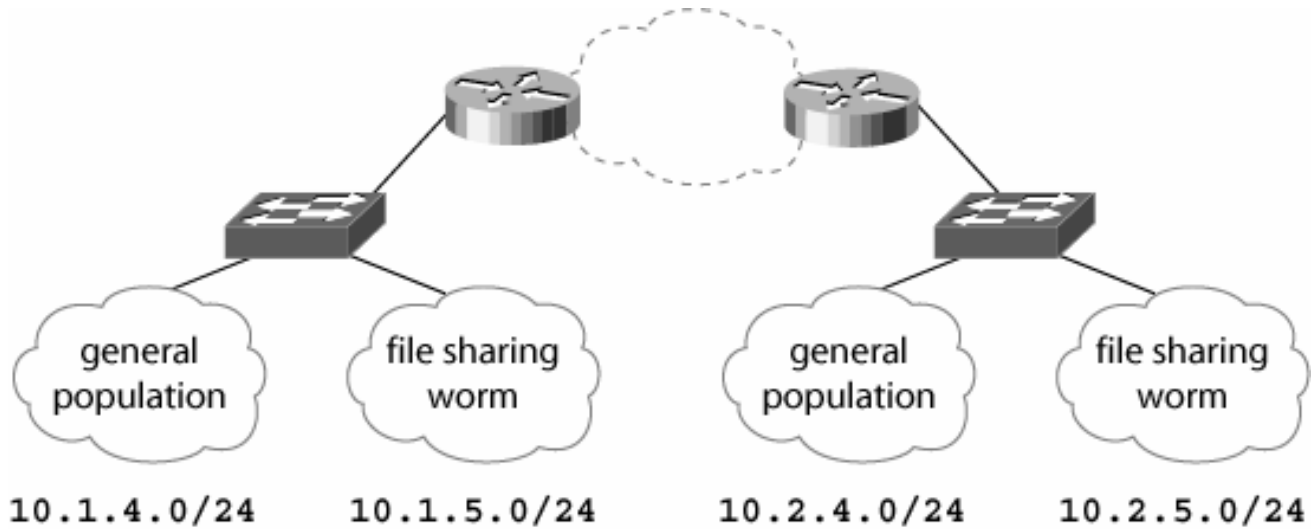
- Allow negotiation of security contract with host
  - Describe the services available and protected
- System has a more detail description of services
  - Can have a more directed scan
  - Reduces the burden on security authentication
  - *Does require client software to negotiate contract*

System Implementation

- Vulnerability detector, evaluates vulnerabilities
  - Nmap is quick, but Nessus provides more detail
  - Include IDS to detect slow worms
- Containment restricts network connectivity
  - Use packet filters (ACL) for layer 3 containment
  - Can market packets or use network addressing
  - VLAN's provide layer 2 protection, but still vulnerable
- Policy manager
  - Daemon process interacting with vulnerability detector and containment system

Security Group Management

- Should scale to a variety of networks



- Groups are virtual
  - located across network (subnet boundaries)
  - Management issues become more difficult

Policy Evaluation

- Evaluate policy types using following criteria
  - Complexity - Implementation and management
  - Correctness - Internal safeguard ability
  - Risk - Risk associated of mischaracterization
  - Utility - Usefulness regardless of state

Policy	Complex	Correct	Risk	Utility
User authentication	Low	Low	High	High
2 groups	Low	High	Low	Low
n groups	Medium	High	Medium	High
blacklist	High	High	Low	High

Conclusions

- Internal malware threats continue to increase
  - Securing mobile networks is more challenging
- Malware defense system
  - Focus on vulnerabilities
  - Provide controlled access to the network
  - No signatures, client software, good for mobile networks
  - Can implement using open source software
- Future directions
  - Scalability across large networks
  - Wireless containment
  - Security group management
  - Integration with more proactive detection methods