Security Alert Systems and Handling Compromised Hosts
CS 239
Advanced Topics in Network Security
Peter Reiher
May 22, 2006

Today’s Topics

• Bad things happen on the net
  – Many affect almost everyone
  – How do we get the word out?
• Many machines attached to the net are compromised
  – Attackers can do arbitrary bad things with them
  – How do we deal with that situation?

Security Alert Systems

• Most network problems threaten lots of people
  – Thousands to millions
• Often there are things one can do to protect oneself
• How do we tell people what they should do?

Difficult Issues for Alert Systems

• Scale
• Delay
• Security
• Trust
• Cost
• Deployment

Scale and the Alert Problem

• Many threats against Windows
• Tens of millions of Windows machines
• Increasingly, usually attached to the network
• Security alerts must work at that scale

Delay and the Alert Problem

• Worms can spread in a few minutes
• Unprotected machines attached to the net are compromised in minutes
  – Sometimes seconds
• Shrinking delays between flaws being revealed and being exploited
• How fast do you need to get alerts out?
Security and the Alert Problem

- Alert systems make recipients do something
- Actions can have wide-ranging effects
  - Close a port?
  - Refuse some types of email?
  - Refuse to talk to certain machines?
- What if attackers generate an alert?
- Can they make large numbers of nodes behave badly?
- How do we stop that from happening?

Trust and Alert Systems

- Assuming we have security,
- Only trusted parties send these powerful alerts
- But who should we trust?
- The “obvious suspects” have done things that suggest trusting them might not be good
  - Not alerting to all problems
  - Not sending alerts very quickly
  - Actions taken for their own motives

Cost and Alert Systems

- Mostly meant to protect us from hazards
- Hazards don’t come up that often
- How much should we pay to get alerts?
  - At alert time?
  - At all times?
  - And where?
- If there’s a monetary cost, who pays?
  - And why would they pay?

Deployment and Alert Systems

- Where do alert components need to be deployed?
- Why would owners of deployment points cooperate?
- How well does the system work if they don’t?
  - Never?
  - At critical moments?

Basic Approaches to Security Alerts

- Centralized alert servers
  - CERT, Symantec, Microsoft, etc.
- Distributed alert servers
  - A la DNS servers
- Peer systems
  - Revere
  - DHT-based
  - Unstructured

What Gets Alerted?

- Worm spread
- Malware signatures
- Sharing of spam signatures
- Blacklists of sites
- Patches
  - SW and firewall rules
- What about use for non-cyber alerts?
  - Tornadoes, hurricanes, tsunami?
- One alert system for everything?
  - Or several different systems?
A Different Approach

- Don’t rely on SW infrastructure to spread alerts
- Rely on people
  - Send out to mailing lists
  - Post on web sites
  - Assume people will tell their friends
- How is this better (or worse)?
- How can we “bundle” this effect with pure SW distribution systems?

Handling Compromised Machines

- Many machines are compromised
  - Hundreds of thousands at least
  - Maybe millions
- Many of them are never fixed
- Many of them are permanently attached to Internet
- What problems does this cause?
- What can we do about them?

Typical Actions of Compromised Machines

- “Manually” compromising other machines
- Spam agents
- Phishing agents
- Storing copyrighted material (warez)
- DDoS agents
- Harvesting of sensitive material
- Password cracking
- Other unforeseen uses

Botnets

- A recent development
  - Three or four years old, mostly
- Attempts by attackers to organize sets of compromised machines
  - Using distributed system techniques
- Eases management
- Allows automated control of large numbers of machines

Typical Botnet Organization

Specifics of Typical Botnet Organization

- An overlay network
  - Generally not optimized for underlying network
- Usually hierarchically arranged
  - Generally not deep hierarchachy
- Often little resiliency
  - Bots know about 1 master
  - But usually mechanisms to change that
Typical Botnet Organization

- Communication generally over IRC
- Usually master/slave communications
  - Usually one master machine
  - Master usually identified by password/key
    - Not by node identity
  - Big botnets might require hierarchy

Approaches to Handling Compromised Machines

- Force cleanup
  - How?
  - “Attack-back” or forced patching legally questionable
- Identify and blacklist compromised nodes
  - Or treat them with caution
- Live with it

Blacklist Approaches

- Someone makes list of compromised nodes
- Notifies other participants in system
- All participants refuse traffic from blacklisted sites
  - Or treat it cautiously

Challenges for Blacklists

- How do you identify compromised nodes?
- How do you alert others?
- What if you’re wrong?
- How do cleaned-up nodes get off the blacklist?
- Boils down to some typical distributed systems/data replication questions

Handling Botnets

- How do you figure out which nodes are in a botnet?
- How do you interfere with a botnet’s operations?
- How do you trace from botnet to attacker’s machine?

Basic Network Tracing

But how do we get back to the hacker’s machine?
More Advanced Network Tracing

How, if we don’t own the botnet master machine?

Watch for commands to botnet master

Watch at Internet ingress point

Stepping Stones

These machines are called stepping stones

What if the hacker doesn’t want to be traced?

Complexities of Dealing With Stepping Stones

• Requires tracing at multiple points
  – Can you get to all of them?
• Complicates tracing by confounding traffic
  – May be hard to pull command traffic out of all stepping stone traces
• Makes it hard to figure out how far you have to go
• Takes longer