Plan

- **DDoS:**
  - Can't prevent malicious traffic reaching you

- **Worms:**
  - Huge number of mostly-identical, poorly managed hosts
  - Cost/effort of timely updating SW is prohibitive
  - Automatic updating is not realistic

- **Today:**
  - What can you do when your machine is compromised? How do you find out?
  - What do we have available to us?
  - What’s the best you can do?
What is an intrusion?

- Attempt to:
  - Compromise confidentiality, integrity, availability
  - Bypass computer or network security mechanisms

- Results from:
  - internet attacks
  - authorized users misusing their privileges
  - authorized users escalating their privileges

Why are intrusions so prevalent?

- Larger # of machines accessible
  - 1988: Morris worm infected 6000 (10%)

- Greater connectivity
  - higher amplification factor, increased range

- Larger # of attackers

- Attack tools more accessible
  - Rise of the script kiddies

- Poorly administered, homogeneous OS & services

- More money in the Internet
How to limit intrusions?

- Prevention
  - Design / Implementation
  - Scan for vulnerabilities (SATAN, ISS)
    * Have to know what to look for
  - Use firewalls to contain / monitor avenues of attack
    * Leaves fewer number of gaping holes

- Preemption? (join the hackers)

How to limit intrusions?

- Deterrence
  - Dress down to unimpress; be boring
  - Scream loud about security measures
  - Add obstacles to increase cost / benefit
    * Must be convincing. Won’t deter all.

- Deflection
  - Contain/lure intruders to “safe” playpens
    - Jails, wrappers, honey pots
    - Dummy accounts with no privileges
    * Questionable legality & effectiveness.
  * Cost / resource expensive.
How to limit intrusions?

- **Detection**
  - Host vs. Network IDS
  - Anomaly vs. misuse detection

- **Countermeasures**
  - Alert sys admin
  - Require reauthentication of user
  - Terminate connection / usage
  - *Could backfire painfully.*

What is an IDS?

- **HW/SW which:**
  - Automatically processes events on computers and networks
  - Analyzes event for security compromise
  - Associates malicious activities with responsible party
  - Responds to attacks
Properties of ideal IDS

- Detects attacks in progress
  - Zero false positives / negatives
- Immediate notification
- Diagnoses attack
- Provides “fixes” for blocking the attack

Host-based IDS

- Analyzes audit trails and system logs for inappropriate sequences
  - Accessing protected files
  - Attempting to increase privilege
  - Illegal set of system calls
- Determines which process or user is attacking
- Determines outcome of attack
Host-based IDS

- Advantages:
  - Encrypted n/w traffic not a problem
  - Sees attacks not evidenced in n/w traffic
    - File access / replacement
    - Software modification (trojans)

- Disadvantages:
  - Don’t see multi-host attacks (scans)
  - Performance overhead
  - If OS compromised:
    - Audit & log trails can be manipulated
    - IDS may be disabled

Recent Host-based IDS research

- pH IDS
  - Create DB of “safe” system call traces
  - Watch system calls for anomalies
  - Susceptible to mimicry attacks

- LiveWire
  - Pull Host IDS out of host for protection
  - Monitor VMM interaction, use to “guess” software state of host

- ReVirt
  - Checkpoint using VMs, log nondeterministic events
  - Analyze by “playing back” attack
  - Doesn’t prevent, but detects and analyzes
Network-based IDS

- Interprets raw n/w packets
- Looks for malicious packets or sequence of packets

Why we like them:
- Can monitor entire n/w at once
- Passive, so no n/w mods required
- OS independent
- Can see multi-host attacks (scans)

Why they could be better:
- Because passive, fail-open
- Difficult keeping up w/ line speed
- Can’t analyze encrypted packets
- Can’t determine affect of individual packets
- Can’t determine success of attack
Anomaly Detection

- Compares expected activity with observed activity
  - Requires the construction of a model for known, expected behavior
  - Works well for predictable usage, but not widely varying, constantly changing usage

Anomaly Detection Usefulness

- Can result in a lot of false positives
  - Wasted sysadm time

- **BUT** is capable of detecting new attacks
  - Necessary to keep up w/ increasing # of attacks
  - Essential if attacks are intrusions are not easily detected
    - Scariest intrusions do hard-to-detect damage
Misuse Detection

- Looks for known “bad” activity
  - Signatures detects specific sequences of attack behavior
  - Invalid state transitions
    - Initial access, misuse privilege or escalate privileges

Misuse Detection Usefulness

- Why we use them:
  - Very effective at detecting known attacks
  - Compensates for long software upgrade delay
    - Could this be eliminated with upgrade distribution system?
  - Creates a safer environment for buggy, but necessary, services
**Misuse Detection Usefulness**

*HOWEVER:*
- does not detect new attacks or attack variations
- writing signatures is difficult
- there is no system for writing signatures and distributing them quickly
  - Could you build one?

**Bro: A System for Detecting Network Intruders in Real-Time**

*Where does Bro fit in?*
- Misuse-detection Network IDS
- “network grep” signature based system
- Searches both packet headers and contents for malicious content

*High-speed network monitor*
- Real-time event notification
- Separate policy & mechanism
- Extensible
Bro architecture

- Libpcap
  - Whittle pkt stream
- Event engine
  - Pkt hdr integrity
  - Maintains conn state
  - Dispatch to handler
  - Generate events
- Policy script interpreter
  - Process events as specified in policy scripts

Types of attacks detected by signature-based NIDSs

- Exploits:
  - CGI scripts (phf)
  - Web server attacks (////////)
  - Buffer overflows
- Reconnaissance
  - TCP scan, UDP scan, OS ID, account scans
- DoS:
  - Ping-of-death, SYN flood
Attacking Bro

- Monitor is passive (fail-open)
- Overload monitor & sneak by
  - Lots of packets to process
  - Packets that generate events
  - Packets that are logged
- Crash monitor by exhausting resources
  - Fragmentation, data reconstruction
- Subterfuge: desynchronize monitor state and endpoint state
  - TCP checksum, IP TTL, IP DF

Contributions

- Refocuses attention on security measures in light of increasing attacks
- Open development of NIDS
  - At the time, lots of commercial development
  - Previous military work, combined anomaly detection & event rules
- Enumerates possible attacks on monitor
  - Insertion, evasion, exhaustion
- Bro itself (in use at Berkeley?)
Discussion

- What percentage of attacks are known (stopped by misuse-detection)? How long are they needed?
  - How important are misuse vs. anomaly detection systems?

- Can we change the incentive from protection to prevention? Do we want to?
  - ISPs become responsible for preventing outgoing attacks

Discussion

- Can remote software administration (faster bug fixes) eliminate misuse detectors?

- Worms spread at an incredible rate. Could you quickly distribute signatures to prevent spread?

- If auto software distribution won’t work, how about distribution of IDS signatures?
Colourful Context

- 1975: Wozniak & Jobs create blue boxes to hack phone system; thanks Captain Crunch
- 1983: War Games
- 1986: Computer Fraud & Abuse Act, Electronic Communications Privacy Act
- 1987: Brain – first MSDOS computer virus?
- 1988: Morris Worm; CERT created
- 1991: Evening with Berford (Cheswick)
- 1993: Poulsen tries to win a Porsche
- 1995: Levin, Mitnick
- 1997: AOHell
- 2000: DDoS attacks on Yahoo, CNN, Amazon, Ebay
  - “I Love You” virus
- 2001: Code Red & Nimda worms