Intrusion Detection and Protection (IDS/IPS)

- Overview
- Host vs Network based
- Signature vs Anomoly based
- Examples

What it is

Intrusion detection and protection systems at their core do the following:

- monitor systems or networks for policy violations
- report/log all violations
- perform/execute some action based on violations (IPS)

Host Intrusion Detection Systems (HIDS)

- Operates on a host system
- Monitors network traffic in and out of system
- Monitors logs and files on the system
  - If critical files change, error/report
  - If logs show patterns known as attack, error/report

/var/log/auth.log

```
Nov 18 05:14:01 pop-os sshd[12489]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=::1  user=root
Nov 18 05:14:03 pop-os sshd[12489]: Failed password for root from ::1 port 60376 ssh2
Nov 18 05:14:10 pop-os sshd[12489]: message repeated 2 times: [ Failed password for root from ::1 port 60376 ssh2]
Nov 18 05:14:10 pop-os sshd[12489]: Connection closed by authenticating user root ::1 port 60376 [preauth]
Nov 18 05:14:10 pop-os sshd[12489]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=::1  user=root
Nov 18 05:14:33 pop-os sshd[12501]: Invalid user admin from ::1 port 60378
Nov 18 05:14:35 pop-os sshd[12501]: pam_unix(sshd:auth): check pass; user unknown
Nov 18 05:14:35 pop-os sshd[12501]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=::1
Nov 18 05:14:37 pop-os sshd[12501]: Failed password for invalid user admin from ::1 port 60378 ssh2
Nov 18 05:14:39 pop-os sshd[12501]: pam_unix(sshd:auth): check pass; user unknown
Nov 18 05:14:41 pop-os sshd[12501]: Failed password for invalid user admin from ::1 port 60378 ssh2
Nov 18 05:14:44 pop-os sshd[12501]: pam_unix(sshd:auth): check pass; user unknown
Nov 18 05:14:45 pop-os sshd[12501]: Failed password for invalid user admin from ::1 port 60378 ssh2
Nov 18 05:14:47 pop-os sshd[12501]: Connection closed by invalid user admin ::1 port 60378 [preauth]
Nov 18 05:14:47 pop-os sshd[12501]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=::1
Nov 18 05:14:56 pop-os sshd[12519]: Invalid user administrator from ::1 port 60380
Nov 18 05:14:58 pop-os sshd[12519]: pam_unix(sshd:auth): check pass; user unknown
Nov 18 05:14:58 pop-os sshd[12519]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=::1
Nov 18 05:15:00 pop-os sshd[12519]: Failed password for invalid user administrator from ::1 port 60380 ssh2
Nov 18 05:15:02 pop-os sshd[12519]: pam_unix(sshd:auth): check pass; user unknown
Nov 18 05:15:04 pop-os sshd[12519]: Failed password for invalid user administrator from ::1 port 60380 ssh2
Nov 18 05:15:07 pop-os sshd[12519]: pam_unix(sshd:auth): check pass; user unknown
Nov 18 05:15:08 pop-os sshd[12519]: Failed password for invalid user administrator from ::1 port 60380 ssh2
Nov 18 05:15:10 pop-os sshd[12519]: Connection closed by invalid user administrator ::1 port 60380 [preauth]
Nov 18 05:15:10 pop-os sshd[12519]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=::1
```

Figure 1: ssh-log

How to move from detect to protect

Figure 2: fail2ban

- detect (multiple failed login attempts)
- report (email admin/ log in central location)
- protect (use iptables to block all communication from that IP address)

Works with almost anything that has an authentication log file

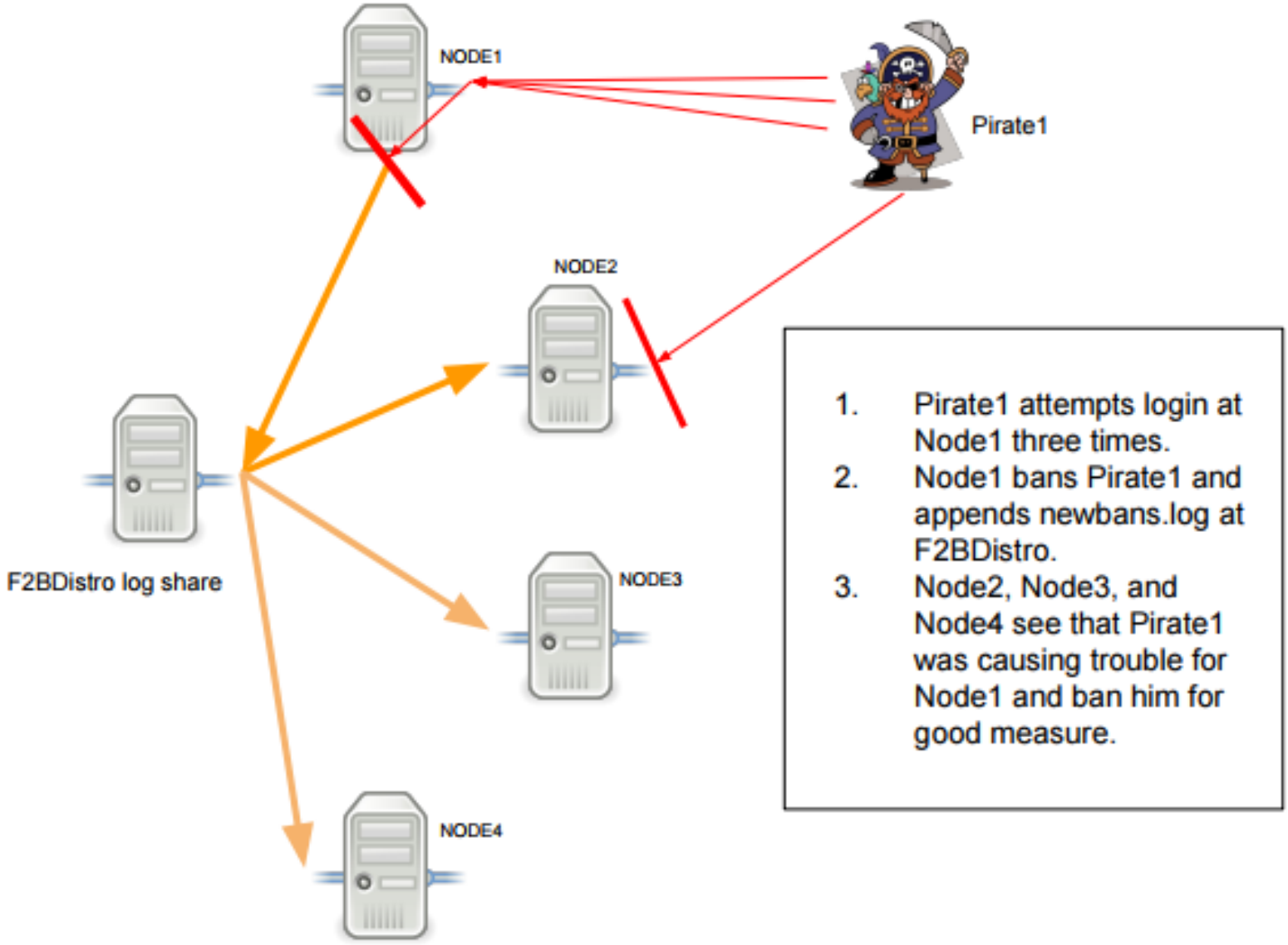

Figure 3: distributed

HIDS beyond logs

Can detect more intrusions than just inspecting logs.

- File integrity monitoring (tripwire)
  - `/sbin`, `/bin`, `/usr/bin`, everything except `/home`
  - `C:\Windows`, `C:\Program Files`, everything except `C:\Users`
  - If files change in these directories, thsi could be considered an anomoly/outside the norm
- Binary analysis
  - Look for already known signatures/hashes/patterns of malicious activity or files

Signature vs Anomoly detection

Signature based detection (binary analysis and others) * assumes that we know what we are looking for * assumes we (or whoever made our rules) has seen this before * we need a good rule set/ signature database * and it will only grow...

Anomoly based detection (we called this statistical previously) * assumes we know what should NOT change * assumes we have a baseline for normal behavior * frequently uses machine learning to model good behaviour * compares new behavior against the model to determine if it is malicious

Network Intrusion Detection System (NIDS)

- Operates at the network level
- Monitors traffic to ALL hosts on network
- Location Location Location...
  - Typically just behind the firewall
  - Has access to all outbound traffic
  - Has access to all inbound traffic that is NOT dropped by firewall
  - Can be in multiple places / distributed across the network
  - Ideally would have access to all internal traffic

Signature vs Anomoly again

The same ideas that we applied in HIDS can be applied here.

Snort (now owned by Cisco) is a powerful NIDS

Signature based * snort blacklist of bad acting IP addresses * snort signature database of known malicious attack patterns (from previously seen network attacks)

Anomoly based detection * typically needs to be application aware * needs to know which ports a given application is running on

Examples

- Antivirus/anti-malware (can be IDS and IPS)
- fail2ban (IPS)
- tripwire (IDS but could protect with better rules)
- OSSEC
- snort (IDS and IPS when allowed to alter rules based on detection)
- selinux (not IDS per se but can detect and log intrusions)
- Kismet (wireless IDS)

Links

Wikipedia IDS Linux IDS Kismet Snort Snort blacklist Snort conf Snort rules