Inside the Perimeter

6 Steps to Improve Your Security Monitoring

Chris Fry, CCSP
Martin G. Nystrom, CISSP-ISSAP
Cisco CSIRT
Cisco TelePresence
Next-generation IP video conferencing
Monitoring Architecture
Cisco IDS, Netflow, and CS-MARS

Netflow anomaly detection

CS-MARS

IDS events signature detection

Cisco IDS

Data Center

CCM 424

CCM 424

CSIRT monitoring
False Positive Traffic Example: 
*SSH sync between CM’s*

False Positive: 
normal sync traffic between call managers
### Security Event Example: Infected host attacking call managers

IDS and MARS detecting hosts attacking call managers

### IDS and MARS
- Detecting hosts attacking call managers

### Atting host was blackholed and submitted for remediation
6 steps to improve your security monitoring

1. Know your policy
2. Know the network
3. Select targets
4. Choose event sources
5. Feed and tune
6. Troubleshoot
What We Assume About Our Audience

- You’ve got an incident response team
- You have experience deploying tools and monitoring
- Focus on discussing deploying monitoring solutions
Step 1. Build and understand your policy
Monitor Against Defined Policies

- Which policies to monitor?
  Be concrete, precise
  Which will management enforce?

- Types of policies
  Compliance with regulations or standards
  - SOX – monitor financial apps and databases
  - HIPAA – monitor healthcare apps and databases
  - ISO 17799 - best practices for information security

  Employee policies
  - Rogue devices – laptops, wireless, DC devices, honeypots, etc.
  - Employees using shared accounts
  - Hardened DMZ devices – services running that should not be?
  - Direct login with privileged accounts (root, DBA, etc.)
  - Tunneled traffic – P2P, etc.
Policy Monitoring Examples

- **Policy: COBIT DS9.4: Configuration Control**
  
  Monitor changes to network devices, reconcile against approved change lists

- **Policy: No direct privileged logins**
  
  Monitor IDS, SSH logs for successful root logins

- **Policy: Use strong passwords**
  
  Vulnerability scan for routers with cisco/cisco credentials

- **Policy: No internet access from production servers**
  
  Monitor for accepted connections to Internet initiated from servers

- **Policy: No protocol tunneling**
  
  Monitor IDS alerts for protocols tunneled over DNS to/from non-DNS servers
Example: FTP Root Login

Caught successful FTP Administrator login via IDS

Caught successful FTP Administrator login via IDS
Example: SSH root login message

Mar 28 16:19:01 xianshield sshd[pam_unix][1398]:
session opened for user root by (uid=0)

Caught direct root login via syslog
Step 2: Know Your Network
Do You Have a **Self Defeating** Network?

- Unknown
- Unmonitored
- Uncontrolled
- Unmanned
- Trusted

Source: Richard Beijtlich
What Is Meant by ‘Telemetry’?

Te·lem·e·try — a technology that allows the remote measurement and reporting of information of interest to the system designer or operator. The word is derived from Greek roots *tele* = remote, and *metron* = measure.
Network Telemetry - What’s it Do For Me?

- Historically used for capacity planning
- Detects attacks
  With analysis tools, can detect anomalies
- Supports investigations
  Tools can collect, trend, and correlate activity
- Well supported
  Arbor PeakFlow
  CS-MARS
  NetQoS
  OSU FlowTools
- Simple to understand
Network Telemetry — Time Synchronization

- Without it, can’t correlate different sources
- Enable Network Time Protocol (NTP) everywhere supported by routers, switches, firewalls, hosts, and other network-attached devices
- Use UTC for time zones
What is NetFlow?

- NetFlow is a form of *telemetry* pushed from the network devices.
- NetFlow is best used in combination with other technologies: IPS, vulnerability scanners, and full traffic capture.

  Traffic capture is like a *wiretap*
  NetFlow is like a *phone bill*

- We can learn a lot from studying the network phone bill!

  *Who’s talking to whom? And when?*
  Over *what protocols & ports?*
  *How much data was transferred?*
  At *what speed?*
  For *what duration?*
Netflow is our #1 tool

- Packet Count
- Byte Count
- Start sysUpTime
- End sysUpTime
- Input ifIndex
- Output ifIndex
- Type of Service
- TCP Flags
- Protocol
- Source IP Address
- Destination IP Address
- Source TCP/UDP Port
- Destination TCP/UDP Port
- Next Hop Address
- Source AS Number
- Dest. AS Number
- Source Prefix Mask
- Dest. Prefix Mask
- From/To
- Application
- Routing and Peering

Ingress i/f

Data Flow

Egress i/f

Data Flow
Netflow Setup

- Don’t have a copy of netflow data b/c IT won’t share?
  
  Many products have the ability to copy flow data off to other destinations.

Regionalized collection to minimize WAN impact

Export netflow data to OSU Flowtools Collector

Netflow data copied to other destinations with flow-fanout
NetFlow Collection at Cisco

- DMZ Netflow Collection (4 servers)
- Data Center Netflow Collection (20+ servers)
- Query/Reporting tools (OSU Flowtools, DFlow, Netflow Report Generator)

200K pps
3 ISP gateways
600GB ~ 3 months
OSU Flowtools - Netflow Collector Setup

- Tool: OSU FlowTools
  - Free
  - Developed by Ohio State University

- Examples of capabilities
  - Did 192.168.15.40 talk to 216.213.22.14?
  - What hosts and ports did 192.168.15.40 talk to?
  - Who’s connecting to port TCP/6667?
  - Did anyone transfer data > 500MB to an external host?
OSU Flowtools Example - Who’s Talking?

- Scenario - New botnet, variant undetected
  
  Goal: identify all systems that ‘talked’ to the botnet C&C

  Be glad: you have netflow collection at all your PoPs

flow.acl file uses familiar ACL syntax. create a list named ‘bot’

concatenate all files from Feb 12, 2007 then filter for src or dest of ‘bot’ acl

host in the botnet!
NetFlow Report Generator – Query by IP

Netflow Report Generator

click on any of the links above the forms for help, or visit the FAQ.

Source IP: 64.102.53.34
Use File for Source
Upload File

Source Port: Destination IP: Destination Port:

Time: 1 day

Report: Netbios Resolve: Uniq:
html

DNS Resolve: both

Netflow Collector:
all
charybdis (San Jose)
rtp-nfc
ams-nfc
syd-nfc

Email address
mnystrom@cisco.com
Run Report

<table>
<thead>
<tr>
<th>DESTINATION</th>
<th>(HOSTNAME:DOMAIN:USER)</th>
<th>PACKETS</th>
<th>TIMESTAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>60.190.23.153[unknown]:7000</td>
<td>1</td>
<td>1205.21:35:59</td>
<td></td>
</tr>
<tr>
<td>61.150.119.94[unknown]:7000</td>
<td>1</td>
<td>1206.00:16:04</td>
<td></td>
</tr>
<tr>
<td>61.152.107.59[unknown]:7000</td>
<td>1</td>
<td>1206.00:23:00</td>
<td></td>
</tr>
<tr>
<td>60.190.23.153[unknown]:7000</td>
<td>1</td>
<td>1206.03:20:57</td>
<td></td>
</tr>
<tr>
<td>61.152.107.59[unknown]:7000</td>
<td>1</td>
<td>1206.11:15:58</td>
<td></td>
</tr>
<tr>
<td>60.190.23.153[unknown]:7000</td>
<td>1</td>
<td>1206.12:42:48</td>
<td></td>
</tr>
<tr>
<td>60.190.23.153[unknown]:7000</td>
<td>1</td>
<td>1206.12:58:27</td>
<td></td>
</tr>
</tbody>
</table>
IP Address Data

- Critical to understanding a given incident involving 10.2.3.5
  Is 10.2.3.5 in your DMZ? lab? remote access? desktop? data center?

- Make the data queryable
  Commercial & open source products available

- Build the data into your security devices
  SIMS - netForensics asset groups
  SIMS - CS-MARS network groups
  IDS - Cisco network locale variables

```plaintext
variables DC_NETWORKS address 10.2.121.0-10.2.121.255,10.3.120.0-10.3.127.255,10.4.8.0-10.4.15.255
variables DMZ_PROD_NETWORKS address 198.133.219.0-198.133.219.255
variables DMZ_LAB_NETWORKS 172.16.10.0-172.16.11.255
```

```plaintext
eventId=116846837254753459 eventType=evIdsAlert hostId=xxx-dc-nms-4appName=sensorApp
appInstanceId=6718 tmTime=1178426525155 severity=1 vLan=700 Interface=ge2_1 Protocol=tcp
riskRatingValue=26 sigId=11245 sigDetails=NICK...USER" src=10.2.121.10 srcDir=DC_NETWORKS
dcport=40266 dst=208.71.169.36 dstDir=OUT
dstport=6665
```
Network Telemetry - MRTG/RRDTool

- Not just netflow, can also use SNMP to grab telemetry
- Shows data volumes between endpoints

You must understand your network traffic volume!
Step 3. Select Your Targets
1. Determine Which Assets to Monitor

- Face it: you can’t monitor everything equally

- How to prioritize?
  - Revenue impact?
  - Regulatory compliance/legal obligation?
  - Expense reduction?
  - At risk?
    - Systems that can’t be patched
    - Most attractive targets to hackers?
  - Sensitive data?
  - Visibility to upper management?
  - Manageable event rates?

- Hopefully, someone else figured this out for you
  - Disaster planning teams

- Which incidents can be mitigated?
Recommendation: Best Targets

1. Accesses sensitive data
   - Legal compliance
   - Intellectual property
   - Customer sensitive data

2. Risky
   Fewer controls (ACL’s, poor configs, etc.)
   Hard to patch (limited patch windows, high uptime requirements, custom vendor code, etc.)

3. Generates revenue

4. Produces actionable events
   - Why monitor if you can’t mitigate?
2. Determine Components to Monitor

- What assets are associated with the target?
  - host names
  - databases
  - applications
  - network devices

- Example: Monitor ERP system
  - List assets associated with system
    - 10 clustered Linux servers
    - 5 clustered database servers
    - 4 “logical” application names
    - 1 LDAP server
  - Policy: Database should only be accessed from app server
  - Monitor for:
    - Outbound connections from db
    - Access to DB on non SQL ports (SSH, terminal services, etc.)
Step 4. Choose Event Sources
Choosing Event Sources: What to Consider

- How will you use it?
  - For monitoring
  - For incident response
  - For investigations

- How will you collect it?
  - Pushed from device (syslog, netflow, etc.)
  - Pulled from device (SDEE, SNMP, Windows logs, etc.)
  - Detected with special equipment (IDS, etc.)

- Performance: what will it do to the sending device?
  - Can you get sufficient detail?
  - Will the support staff give it to you?
Choosing Event Sources: What to Consider (cont.)

- How much storage do you have?
- What tools will you use to read it?
  - SIM, log analyzer, etc.
- Application specific
  - Can you recognize “false positive” patterns and tune them out?
  - Will you get enough information to act on it without a full packet-capture?
  - Can you identify specific incidents and how you’d see it with your event source?
  - Do you know what you’d do with it if there’s really an incident?
Three Best Event Sources

- **Netflow**
  - Collect at chokepoints (data center gateways)
  - Cheap to collect: SJC stores 3 ISP gateways, 200k pps, 600GB storage, can query back 3 months
  - Free tools to collect, relay, query
    - OSU FlowTools, nfdump/nfsen, etc.

- **Network IDS**
  - Collect at chokepoints (data center gateways)
  - No agents or feeds taxing end systems

- **Host logs**
  - Unix: syslog
  - Collect common services via syslog (web servers, mail servers, etc.)
  - Collect with syslog relay/collector
    - syslog-ng, splunk, etc.
  - Collect Windows logs into same infra with Snare agents
Searching Through Logs w/Splunk

![Splunk Interface](image)

- **Events by Time**: Click any bar to set a time range. Shift-click another to extend the range.

- **Events (≥10,000)**

  - **Event Types (≥59)**
  - **Source Types (≥1)**
  - **Hosts (≥2)**
  - **Sources (≥3)**

- **Events 1-10 of at least 10,000, sorted by time.**

  - **Page**: 1 2 3 4 5 6 ...
  - **Results per page**: 10

- **Event 1**: 171.70.168.141 - emantest1 [16/Mar/2007:00:06:49 -0700] "GET /cgi-bin/front.x/ReturnsTool?AppName=RMA_Status&MODE=QUERY&gtype=POS&num=92889&..."
  - **Event Type**: 10
  - **Time**: 03/16/2007, 03:06:49
  - **Similar** | **Related**
  - **Source Type**: access_log_u1
  - **Host**: chris-frys-computer.local
  - **Source**: /applications/splunk/var/spool/splunk/cco-sj-2-access_log
  - **Show source**
  - **Look up event** | **Share event**

- **Event 2**: 210.227.114.167 - - [16/Mar/2007:00:06:43 -0700] "GET /pcgi-bin/imagemap/guestbar.jsp?34,4 HTTP/1.1" 302 309
  - **Event Type**: 77
  - **Time**: 03/16/2007, 03:06:43
  - **Similar** | **Related**
  - **Source Type**: access_log_u1
  - **Host**: chris-frys-computer.local
  - **Source**: /applications/splunk/var/spool/splunk/cco-sj-2-access_log
  - **Show source**
  - **Look up event** | **Share event**

- **Event 3**: 203.141.92.14 - yokotak [16/Mar/2007:00:06:43 -0700] "GET /cgi-bin/order/ipc_entrance.pl?ipc_store=CISCO_NETWORKING PRODUCTS_NEW_OT HTTP/1.1"
  - **Event Type**: 75
  - **Time**: 03/16/2007, 03:06:43
  - **Similar** | **Related**
  - **Source Type**: access_log_u1
  - **Host**: chris-frys-computer.local
  - **Source**: /applications/splunk/var/spool/splunk/cco-sj-2-access_log
  - **Show source**
  - **Look up event** | **Share event**

- **Event 4**: 10.76.191.80 - rgudipot [16/Mar/2007:00:06:35 -0700] "GET /cgi-bin/front.x/src/servlet/SRCICITServlet?ApplicationName=SRC&ServiceName=SRCCor..."
  - **Event Type**: 16
  - **Time**: 03/16/2007, 03:06:35
  - **Similar** | **Related**
  - **Source Type**: access_log_u1
  - **Host**: chris-frys-computer.local
  - **Source**: /applications/splunk/var/spool/splunk/cco-sj-2-access_log
  - **Show source**
  - **Look up event** | **Share event**
Searching Through Logs w/Sawmill

![Sawmill Interface]

<table>
<thead>
<tr>
<th>Session ID</th>
<th>User ID</th>
<th>User Name</th>
<th>Events</th>
<th>Events (%)</th>
<th>Start Time</th>
<th>End Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>209.47.179.250-2007-03-15:05:05:06</td>
<td>209.47.179.250</td>
<td>88</td>
<td>0.1%</td>
<td>15 Mar 2007 05:05:06</td>
<td>15 Mar 2007 06:40:45</td>
</tr>
<tr>
<td>6</td>
<td>152.102.1.107-2007-03-15:00:13:01</td>
<td>152.102.1.107</td>
<td>80</td>
<td>0.1%</td>
<td>15 Mar 2007 00:13:01</td>
<td>15 Mar 2007 01:04:54</td>
</tr>
</tbody>
</table>
Step 5. Feed and Tune
IDS/IPS Refresher

- **IDS** - Intrusion Detection System
  - passive network traffic monitoring
  - limited actions, mostly for alerting

- **IPS** - Intrusion Prevention System
  - *inline* network traffic monitoring
  - alerting + ability to drop packets

![Diagram of IDS/IPS traffic flow and decision process](image)
IDS - basic deployment steps

Analyze

Design

Deploy

Tune

Manage

Tuning and management are ongoing
Setup IDS

- Avoid asymmetry in your traffic view!
- Minimize the number of platforms and designs

Two different designs: small vs. large data centers

Distribution layer router uplink traffic ideal

BB uplinks mirrored to sensor

BB uplinks mirrored to load balancer
Feed Netflow to SIMs and Other Tools

- Feed Netflow to every tool that will use it
  - MARS, PeakFlow, etc.
- Regionalize deployment
  - Minimize sending over network

```
SFHJPO
SIM
Network analyzer
Netflow collector
Netflow collector
```

region 1

region 2

SIM
Network analyzer
Host Syslog

- Capture, store, and relay with syslog-ng
- For monitoring, be sure your SIM can parse events
- Deploy standard template (syslog.conf)
- Key events to log
  - authentication logs
  - authorization logs (sudo, su, etc.)
  - daemon status logs (know when they stop/start)
  - security application logs (tcpwrappers, portsentry, etc.)
- Windows logging
  - Agents can relay events via syslog
  - Very noisy, grab only important events

<table>
<thead>
<tr>
<th>EventID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>528</td>
<td>User Logon</td>
</tr>
<tr>
<td>529 - 537</td>
<td>Logon Failure</td>
</tr>
<tr>
<td>538</td>
<td>User Logoff</td>
</tr>
<tr>
<td>612</td>
<td>Audit Policy Change</td>
</tr>
<tr>
<td>517</td>
<td>Audit Log Cleared</td>
</tr>
</tbody>
</table>
Other Logs

- **Web server logs**
  - Can verify and elaborate attacks
  - Use HTTP status codes to determine if IDS alert really worked
  - Can provide URL details during attack
  - **Apache**
    - Send as syslog via httpd.conf setting
  - **IIS**
    - Send as syslog via MonitorWare Agent

- **App server logs**
  - Find way to relay as syslog
  - Send via SNMP events
  - Pull via SQL queries

- **Oracle logs**
  - Pull logs from AUD$ table via SQL
What’s the difference?
Many more false positives sources

Tuning more complex

A good relationship with IT application and service owners is key
Enterprise Datacenter Monitoring
Complications / Difficulties

- **Traffic**: 100+ Gbps globally vs. 4 Gbps outside
- **Protocols**: Higher number of services/protocols increases variety and complexity of tuning
- **Alerts**: Untuned sensor in large datacenter generates > 100 million alerts/day
Enterprise Datacenter Monitoring
Complications / Difficulties (continued)

- **Higher availability expectations**
  
  Enterprise data centers have very high availability requirements
  
  Inline “IPS” a hard sell, most hardware not properly redundant
  
  We don’t use inline IPS

- **False positives**
  
  Difficult and time consuming to identify
  
  Key: good relationship with IT application and service owners

- **Relatively new technology**
  
  Not well understood by IDS & SIMs yet
  
  Limited signature base
  
  Most signatures based on Internet attacks
False Positives - Examples

- SigID 3320 - ADMIN$ access
- SigID 3337 - Windows RPC Race Condition
- SigID 5722 Google Appliance ProxyStyleSheet Cmd Exec
- SigID 3653 Multiple Rapid SSH Connections

Each of these required that we contact the IT application or system owners to verify false positive.
Step 6. Maintain & Troubleshoot
Maintain Documented Commitments

- Document agreements with IT
  - Fixed timelines
  - Expectations (SLAs, OS patching, etc)
  - Refresh commitments every year

- Review assets regularly
  - Look for new assets, new feeds, replaced hosts, etc.
  - Check for feeds/hosts that may have changed/disappeared
  - Check for ownership changes due to re-orgs
Maintain IDS Feeds

- Monitor your IDS sensor uplinks
- Watch for spikes/drops in sensor alert volume
- Have monitoring staff monitor feeds

This sensor is no longer receiving network traffic!
Verify Feeds

- **Syslog feed verification**
  
  Script `awk` to grab hostnames of systems that syslog daily and do a diff
  
  Ask IT to use a daily cron to re-set syslog.conf on servers

- **Netflow feed verification**
  
  `tcpdump -i eth0 port 2060 -c 1000 | grep gw | awk '{print $2}' | sort | uniq

```
May 16 07:57:40 flanders-mac com.apple.SecurityServer: Succeeded authorizing right system.preferences/System Preferences.app for authorization created by /Applications/System Preferences.app
May 16 07:57:41 flanders-mac com.apple.SecurityServer: Succeeded authorizing right system.preferences/System Preferences.app for authorization created by /Applications/System Preferences.app
May 16 09:51:40 flanders-mac com.apple.SecurityServer: Succeeded authorizing right system.preferences/System Preferences.app for authorization created by /Applications/System Preferences.app
May 16 09:51:40 flanders-mac com.apple.SecurityServer: Succeeded authorizing right system.preferences/System Preferences.app for authorization created by /Applications/System Preferences.app
May 16 15:50:07 flanders-mac com.apple.SecurityServer: Succeeded authorizing right system.login System/Library/CoreServices/loginwindow.app for authorization created by /System/Library/CoreServices/loginwindow.app for authorization created by /System/Library/CoreServices/loginwindow.app for authorization created by /System/Library/CoreServices/loginwindow.app for authorization created by /System/Library/CoreServices/loginwindow.app
May 16 17:22:36 flanders-mac sshd[17844]: Could not write ident string to UNKNOWN
```
Lessons Learned

- Start small
  - Too many events at once is overwhelming
  - Understand/tune each source before adding more
  - Understand “normal” traffic thoroughly before moving on
    - Avoid alerting on false-positives

- Use a SIM
  - Event correlation, false positive reduction

- Choose carefully what you want to monitor
  - …or you’ll waste your time chasing false positives

- Use defined playbooks, escalation procedures

- Have allies in the IT support teams
  - Network support, DBA’s, webmasters, etc.
  - They can explain/remediate issues you find
6 steps to improve your security monitoring

1. Know your policy
2. Know the network
3. Select targets
4. Choose event sources
5. Feed and tune
6. Troubleshoot