Monitoring Data Fusion for Intrusion Tolerance

Atul Bohara
P.I.: W. H. Sanders
ACC Seminar, Oct 7, 2015
Overview

• Protect a real-world networked system against malicious activities
  – E.g., Enterprise network / campus network / cloud data center
  – Prevention techniques are not sufficient
  – Need to rely on security monitoring and detection

• Questions
  – How to make *overall sense* out of the data generated by these monitors?
  – How to support intrusion detection and automatic response?
Motivation

Types of cyber crime attacks experienced by companies in the United States as of June 2014

- Viruses, worms, trojans: 100
- Malware: 97
- Botnets: 76
- Web-based attacks: 61
- Malicious code: 46
- Phishing and social engineering: 44
- Malicious insiders: 41
- Stolen devices: 37
- Denial of service: 34

Quellen:
Pokémon Institute, Hewlett-Packard (HP Enterprise Security)
© Statista 2015

Additional Information:
United States: Pokémon Institute; June 2014; 59 U.S. organizations
Motivation

- Security Breaches are increasing
- Security data is large and unmanageable
Problem

- System sizes are growing
  - New attack types
  - Variety of tools and techniques required to protect the system

- Overwhelmingly big and heterogeneous security data

- Difficult to have a holistic view of system
Can we do something?

<table>
<thead>
<tr>
<th>Process Name</th>
<th>Sent Bytes</th>
<th>Recvd By...</th>
<th>Sent Packets</th>
<th>Recvd Packets</th>
<th>PID</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dropbox</td>
<td>552 KB</td>
<td>12.6 MB</td>
<td>1,313</td>
<td>53,783</td>
<td>312</td>
<td>absdr</td>
</tr>
<tr>
<td>mDNSResponder</td>
<td>148 KB</td>
<td>11.4 MB</td>
<td>384</td>
<td>96,873</td>
<td>87</td>
<td>mdnsrespond</td>
</tr>
<tr>
<td>networkd</td>
<td>120 KB</td>
<td>6.8 MB</td>
<td>1,606</td>
<td>78,515</td>
<td>376</td>
<td>_networkd</td>
</tr>
<tr>
<td>SpotlightNetHelper</td>
<td>14 KB</td>
<td>50 KB</td>
<td>29</td>
<td>47</td>
<td>338</td>
<td>absdr</td>
</tr>
<tr>
<td>Box Sync</td>
<td>12 KB</td>
<td>42 KB</td>
<td>30</td>
<td>66</td>
<td>327</td>
<td>absdr</td>
</tr>
<tr>
<td>apad</td>
<td>9 KB</td>
<td>7 KB</td>
<td>48</td>
<td>46</td>
<td>73</td>
<td>root</td>
</tr>
<tr>
<td>MATLAB</td>
<td>8 KB</td>
<td>27 KB</td>
<td>113</td>
<td>112</td>
<td>836</td>
<td>absdr</td>
</tr>
<tr>
<td>syslogd</td>
<td>2 KB</td>
<td>0 bytes</td>
<td>11</td>
<td>0</td>
<td>42</td>
<td>root</td>
</tr>
<tr>
<td>ntpd</td>
<td>1,008 bytes</td>
<td>37 KB</td>
<td>21</td>
<td>509</td>
<td>173</td>
<td>root</td>
</tr>
<tr>
<td>screensharingd</td>
<td>852 bytes</td>
<td>472 bytes</td>
<td>65</td>
<td>50</td>
<td>1069</td>
<td>root</td>
</tr>
<tr>
<td>config</td>
<td>300 bytes</td>
<td>88 KB</td>
<td>1</td>
<td>274</td>
<td>50</td>
<td>root</td>
</tr>
<tr>
<td>SystemUIServer</td>
<td>64 bytes</td>
<td>0 bytes</td>
<td>16</td>
<td>0</td>
<td>255</td>
<td>absdr</td>
</tr>
<tr>
<td>kernel_task</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>root</td>
</tr>
<tr>
<td>appleeventd</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>49</td>
<td>_appleevents</td>
</tr>
<tr>
<td>nsurlstoringd</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>242</td>
<td>absdr</td>
</tr>
<tr>
<td>ScopedsBookmarkAgent</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>386</td>
<td>absdr</td>
</tr>
<tr>
<td>Mail</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>436</td>
<td>absdr</td>
</tr>
<tr>
<td>iccd</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>245</td>
<td>absdr</td>
</tr>
<tr>
<td>mdflagwriter</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>369</td>
<td>absdr</td>
</tr>
<tr>
<td>com.apple.hiservices-xpcserver</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>628</td>
<td>absdr</td>
</tr>
<tr>
<td>AppleIDAuthAgent</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>294</td>
<td>absdr</td>
</tr>
<tr>
<td>bash</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>821</td>
<td>absdr</td>
</tr>
<tr>
<td>com.apple.CodeSigningHelper</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>199</td>
<td>root</td>
</tr>
<tr>
<td>pbs</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>343</td>
<td>absdr</td>
</tr>
<tr>
<td>warmd</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>58</td>
<td>root</td>
</tr>
<tr>
<td>mdworker</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>441</td>
<td>_spotlight</td>
</tr>
<tr>
<td>bash</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>824</td>
<td>absdr</td>
</tr>
<tr>
<td>com.apple.audio.SandboxHelper</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>634</td>
<td>root</td>
</tr>
<tr>
<td>networkd</td>
<td>0 bytes</td>
<td>0 bytes</td>
<td>0</td>
<td>0</td>
<td>156</td>
<td>_networkd</td>
</tr>
</tbody>
</table>

Packets in: 487,087
Packets out: 46,677
Packets in/sec: 54
Packets out/sec: 7

Date received: 57.5 MB
Data sent: 8.5 MB
Data received/sec: 6.98 KB
Data sent/sec: 389 bytes
Can we do something?

Can we build an activity monitor for a real-world networked system?
Our Approach

• Learn profile of the system over time
  – Define important categories of information
  – Host, network, users, application
  – Define multiple views of the system
• Generate and maintain these views in real-time
• Ability to drill-down and roll-up

Monitoring Fusion: create, maintain and present higher-level system views.
Detect anomalies, policy violations; act upon them
System Views

Global views

View₁ ... Viewₙ

Fusion

View₁ ... Viewₙ

System state

Kernelₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜₜ
System Views

Global views

View \(_1\) ... View \(_n\)

Fusion

View \(_1\) ... View \(_n\)

represented as

System state

Kernel\(_{\text{HostA}}\) Kernel\(_{\text{HostN}}\) Switch

represented as

Legend

Unobservable state

View
System Views

Global views

View_1  ...  View_n

Fusion

View_1  ...  View_n

System state

Kernel_{HostA}  Kernel_{HostN}  Switch

Legend

- Unobservable state
- View
Outline

1. Generating System Views
   1. Handling Big and heterogeneous data

2. Issues
   1. Conflicting evidences
   2. Independence of data sources
   3. Ensure security of the procedure
Outline

1. Generating System Views
   1. Handling Big and heterogeneous data

2. Issues
   1. Conflicting evidences
   2. Independence of data sources
   3. Ensure security of the procedure
Generating System Views

• Define a catalog of views

• Combine data from multiple sources intelligently
  – Monitor selection and data extraction
  – Data reduction techniques
  – Hierarchical and on-demand fusion

• Group parts of system intelligently
  – Topological grouping
  – Grouping based on behavior
Example 1: Process-Traffic View

Number of processes, network bytes, and network packets

Combined profile of a group of two hosts
Drilling-down in Process-Traffic View

Host 1

Number of processes

Number of packets

Number of bytes

Host 2

Number of processes

Number of packets

Number of bytes
Example 2: Network-Resources View

Local resource usage on hosts linked to every network activity

java (pid 31482)

sshd (pid 1377)

chrome (pid 7616)

thunderbird (pid 25127)

dropbox (pid 9115)

79.27.13.215

146.48.98.155

108.160.163.108
Raw Monitoring Data

Problems:
- Large volumes
- Difficult to analyze
Network-Resources View Data

<table>
<thead>
<tr>
<th>Monitors</th>
<th>Number of records</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw data</td>
<td>top</td>
<td>20,219</td>
</tr>
<tr>
<td></td>
<td>lsof</td>
<td>14,930</td>
</tr>
<tr>
<td>Fused data</td>
<td>top+lsof</td>
<td>254</td>
</tr>
</tbody>
</table>

**Fusion benefits**
- Reduction in volume
- Easy visualization
Fusion System Architecture

Monitoring data

- top output
- lsof output
- Sysdig output
- Switch logs

Data Streaming and Parsing
Apache Kafka

- Kafka producers
- Kafka broker
- Kafka consumers

Kafka topics

Fusion and Visualization

Monitoring fusion

System Views
Outline

1. Generating System Views
   1. Handling Big and heterogeneous data

2. Issues
   1. Conflicting evidences
   2. Independence of data sources
   3. Ensure security of the procedure
Confidence in Fused Information

• Hypothesis: it’s harder for an intrusion to deflect the output of multiple independent monitors

• Our approach
  – Utilize multiple independent monitors to generate same view
  – Accept the data if consistent otherwise report an alarm
Independence of Monitors

• Following classes of monitors are considered independent
  – Deployed at different physical locations
    • Host v/s network
  – Working at different access levels
    • User v/s kernel
Conclusion

• **Monitoring fusion** allows us to generate system views

• **System views**: convert monitoring data to useful information
  – Provide more useful and concise information
  – Ability to drill-down as required
  – Improve efficiency of decision making processes
  – Sophisticated algorithms can be built on top

• **Set of views** ➔ **System Activity Monitor**
Future Work

• Build a catalog of useful views
• Data collection from multiple hosts and network devices
• Streaming + offline analytics
• Ensuring security of the process
Previous Related Literature

Fusion techniques for information security

• Data to information
  – Aggregation, data reduction techniques [Valdes’01, Ma’08, Czejdo’12]

• Information to knowledge
  – Alert correlation, multiple classifier combination [Cuppens’00, Debar’01, Ning’02, Ning’04, Totel’04, Zhu’06, Li’07, Almgren’08, Li’10, Roschke’10, Zhou’11, Kumar’14]

Data-driven approaches in intrusion detection

• Data mining and machine learning techniques in intrusion detection [Lee’08, Lan’10, Zhang’12, Yen’13, Kiss’14, Miller’14]

Online clustering algorithms

• K-means based [Aggarwal’03, Ackermann’12]
• Density based [Cao’06, Chen’07, Ding’15]
Previous Related Literature

Distributed System Management and Monitoring

• Bro Network Security Monitor
• Nagios IT infrastructure monitoring
• Distributed system resource managers
  – Mesos
  – YARN
  – OpenStack
Backup Slides
References


References

References


References


References


