Attribution and Aggregation of Network Flows for Security Analysis

Annarita Giani
Ian De Souza
Vincent Berk
George Cybenko

Institute for Security Technology Studies
Thayer School of Engineering
Dartmouth College
Hanover, NH

FloCon 2006, Portland, OR
Why flow data

The context in which we are interested in flow analysis is the following.

• We believe that **automated correlation** is hard to do.

• The world consists of **processes** so our approach to correlation is process-based.

• Introduction, in 2003, of generic process-based correlation engine concept and implementation, **Process Query System** (PQS).
  • Integration of multiple existing and new sensor types and attacks models
  • Flow aggregation and correlations between flow data with security events
  • Implementation of a **PQS based process detection for Cyber Situational Awareness**.

• **Need for flow data.**
Process Query System

Observable events coming from sensors

Models

Model $M_1$

Model $M_2$

... 

Model $M_k$

Likelihood $L_1$

Likelihood $L_2$

Likelihood $L_k$

PQS ENGINE

Implemented for:
Vehicle Tracking
Computer Security
Social Network
Plume Tracking
Cyber Situational Awareness

An Environment

Multiple Processes
\( \lambda_1 = \text{router failure} \)
\( \lambda_2 = \text{worm} \)
\( \lambda_3 = \text{scan} \)

Events

Real World

Indicators and Warnings

129.170.46.3 is at high risk
129.170.46.33 is a stepping stone

Hypotheses

Hypothesis 1
Hypothesis 2

Track 1
Track 2
Track 3

Track Scores

FORWARD PROBLEM

INVERSE PROBLEM

that are used for control

that detect complex attacks and anticipate the next steps

that PQS resolves into

Sample Console
PQS in Computer Security

Internet

DMZ

BRIDGE

WWW  Mail

WS

WinXP  LINUX

DIB:s  BGP  IPTables  Snort  Flow

observations

Flow

PQS ENGINE

Models

Worm  Exfiltration  Phishing
# Sensors and Models

<table>
<thead>
<tr>
<th>Sensors</th>
<th>Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIB:s</td>
<td>Noisy Internet Worm Propagation – fast scanning</td>
</tr>
<tr>
<td>Snort, Dragon</td>
<td>Email Virus Propagation – hosts aggressively send emails</td>
</tr>
<tr>
<td>IPtables</td>
<td>Low&amp;Slow Stealthy Scans – of our entire network</td>
</tr>
<tr>
<td>Samba</td>
<td>Unauthorized Insider Document Access – insider information theft</td>
</tr>
<tr>
<td>Flow sensor</td>
<td>Multistage Attack – several penetrations, inside our network</td>
</tr>
<tr>
<td>ClamAV</td>
<td>DATA movement</td>
</tr>
<tr>
<td>Tripwire</td>
<td>TIER 2 models</td>
</tr>
</tbody>
</table>

#### Sensors
- **DIB:s**: Dartmouth ICMP-T3 Bcc: System
- **Snort, Dragon**: Signature Matching IDS
- **IPtables**: Linux Netfilter firewall, log based
- **Samba**: SMB server - file access reporting
- **Flow sensor**: Network analysis
- **ClamAV**: Virus scanner
- **Tripwire**: Host filesystem integrity checker
Multi Stage Attack Example: Phishing

Step 1: as usual browses the web and visits a web page.
Step 2: inserts username and password.
Step 3: (the same used to access his machine)
Step 4: accesses user machine using username and password
Step 5: uploads some code
Step 6: downloads some data
Step 7: attacks the victim
Step 8: records username and password

Attacker: 51.251.22.183
Victim: 100.10.20.9
Stepping stone: 100.20.3.127
Web page, Madame X: 165.17.8.126
Phishing Attack Observables

1. **RECON**
   - Source: SNORT: KICKASS_PORN DRAGON: PORN HARDCORE
   - Destination: 100.20.3.127
   - Time: Sept 29 11:17:09

2. **ATTACK**
   - SSH (Policy Violation), Non-Standard-Protocol
   - Source: 51.251.22.183
   - Destination: 100.20.3.127
   - Time: Sept 29 11:23:56

3. **DATA UPLOAD**
   - Flow Sensor
   - Source: 51.251.22.183
   - Destination: 100.20.3.127
   - Time: Sept 29 11:23:56

4. **ATTACK RESPONSE**
   - Source: User Name: password
   - Destination: 165.17.8.126
   - Time: Sept 29 11:24:06

5. **DATA DOWNLOAD**
   - Flow Sensor
   - Source: 100.10.20.9
   - Destination: 100.20.3.127
   - Time: Sept 29 11:24:07

Web Server used: Madame X

Victim

Atacker
Flow Sensor

Based on the *libpcap* interface for packet capturing. Packets with the same source IP, destination IP, source port, destination port, protocol are aggregated into the same flow.

- Timestamp of the last packet
- # packets from Source to Destination
- # packets from Destination to Source
- # bytes from Source to Destination
- # bytes from Destination to Source
- Array containing delays in microseconds between packets in the flow
Two Models Based on the Flow Sensor

### Low and Slow UPLOAD

<table>
<thead>
<tr>
<th>Volume</th>
<th>Packets</th>
<th>Duration</th>
<th>Balance</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiny: 1-128b</td>
<td>4: 10-99</td>
<td>4: 1000-10000 s</td>
<td>Out</td>
<td>&gt;80</td>
</tr>
<tr>
<td>Small: 128b-1Kb</td>
<td>5: 100-999</td>
<td>5: 10000-100000 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6: &gt; 1000</td>
<td>6: &gt; 100000 s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### UPLOAD

<table>
<thead>
<tr>
<th>Volume</th>
<th>Packets</th>
<th>Duration</th>
<th>Balance</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiny: 1-128b</td>
<td>1: one packet</td>
<td>0: &lt; 1 s</td>
<td>Out</td>
<td>&gt;80</td>
</tr>
<tr>
<td>Small: 128b-1Kb</td>
<td>2: two pkts</td>
<td>1: 1-10 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium: 1Kb-100Kb</td>
<td>3: 3-9</td>
<td>2: 10-100 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Large: &gt; 100Kb</td>
<td>4: 10-99</td>
<td>3: 100-1000 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5: 100-999</td>
<td>4: 1000-10000 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6: &gt; 1000</td>
<td>5: 10000-100000 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6: &gt; 100000 s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Aggregation

Flow aggregation.
Recognizing that different flows, apparently totally unrelated, nevertheless belong to the same broader event (activity).
Flows are aggregated from captured network packets.
We aggregate flows into activities.
Example:
User requests a webpage (all DNS and HTTP flows aggregated)

Activity aggregation.
Recognizing that similar activities occur regularly at the same time, or dissimilar activities occur regularly in the same sequence.
We correlate activities into activity groups, patterns.
Examples:
• Nightly backups to all servers (each backup is an activity)
• User requests a sequence of web-pages every morning.

Packet = Aggregated Bytes
Flow = Correlated Packets
Activity = Correlated Flows
Pattern = Correlated Activities
Web Surfing in Detail

1. The browser communicates with a name server to translate the server name "www.dartmouth.edu" into an IP Address, which it uses to connect to the server machine.

2. The browser forms a connection to the web server at that IP address on port 80.

3. Following the HTTP protocol, the browser sends a GET request to the server, asking for the file "http://www.dartmouth.edu/index.html."

4. The web server sends the HTML text for the Web page to the browser.

5. The browser reads the HTML tags and formatted the page onto your screen.

6. Browser possibly initiates more DNS requests for media such as images and video.

7. Browser initiates more HTTP and/or FTP requests for media.
Resulting Flows and Activity

Flows in the activity

Activity

Stacked Activities in Black, UDP Flows in Blue

Seconds

x 10^5
Activities and Flows

- UDP Flow
- TCP Flow

Horizontally Stacked Activities in Black with Associated Flows (Red=TCP, Blue=UDP)

Activity

Long Flow
Complex Activities ....

Correlated Network Flows Within a LAN

- TCP portscan
- Regular UDP broadcasts (NTP)
- System upgrade
- Regular browsing/download behavior
- UDP portscan
Packets in a flow triggered IDS alerts

PQS instantiates models based on observation coming from flow and snort sensor.

Snort rule 1560 generates an alert when an attempt is made to exploit a known vulnerability in a web server or a web application.

Snort rule 1852 generates an alert when an attempt is made to access the 'robots.txt' file directly.

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Sensor</th>
<th>src IP</th>
<th>dst IP</th>
<th>Proto</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul 09 16:28:32</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 09 16:29:35</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 09 16:44:44</td>
<td>S1850</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 09 18:26:08</td>
<td>S1850</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 09 21:05:03</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 09 22:31:08</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 09 22:31:08</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 10 02:45:19</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 10 02:45:23</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 10 09:21:15</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 10 14:33:43</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 10 17:54:54</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 10 22:07:02</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 11 01:38:09</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 11 04:05:54</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 11 04:20:00</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 11 11:07:12</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 11 11:56:12</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>Jul 11 17:16:59</td>
<td>S1852</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>S Jul 10 02:30:27</td>
<td>F</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
<tr>
<td>E Jul 10 23:55:56</td>
<td>F</td>
<td>65.54.188.140</td>
<td>208.233.154.195</td>
<td>TCP</td>
</tr>
</tbody>
</table>

Table 2: A sample track of correlated IDS and Flow events

The flow can be characterized as malicious and further investigation must be done.
Future Direction

Theoretical approach for clustering aggregated flows.

Flow = As defined
Activity = Aggregated flows
Pattern = Correlated Activities

Approach: Graph theory (flows are the nodes and the edges are between correlated nodes).

We are thinking about defining a metric that captures the closeness between two different activities to allow grouping into patterns.

Activity 1.

Activity 2.

Can they be grouped in one pattern?
Notion of distance between activities.
www.pqsnet.net
agiani@ists.dartmouth.edu
Student and researcher use this network to browse the web, print documents, send upload and download files...
Web Surfing

208.253.154.210  host name
208.253.154.195  dns.pqsnet.net
129.170.16.4      ns.dartmouth.edu

1. ns.pqsnet.net requests www.nytimes.com ip address to ns.dartmouth.edu
2. ns.dartmouth.edu returns the ip address – 199.239.136.245
3. TCP three-way handshake between the host machine and the web server.
4. HTTP GET request to 199.239.136.245
5. TCP ACK from the web server
6. Other packets exchanges between the web server and the host

All these network connections are related to the same host activity.