Computer Network Defense: Compromise Detection Prototype

Carl Beisel, Jim Jones, Christian Iivari
The Problem

- **Zero Day Attacks:**
  - How and when does a novel, previously unknown attack first get discovered? Can that attack be detected and stopped before affected systems are compromised and exploited?

- **Problem:**
  - Signature based detection patterns are based on having discovered, evaluated and defined patterns for the attack. Behavior based detection has high false positives.

- **Approach:**
  - Non-signature, non-behavior based detection
  - Attack Modeling: reason over observables (indicators, anomalies, second-order effects, etc.)
Technical Approach

- **Perform Deep Packet Inspection** of network traffic and capture of packets of interest matching one of our 16+ observable rules
  - Observables represent pieces of evidence relevant to the activities an attacker may perform during an attack as represented by the following transition states:
    - Vulnerability Research
    - Exploit Development and Testing
    - Reconnaissance
    - Exploit Execution
    - Cleaning Activities
    - Back Door Installation
  - Outputs observables for analysis by the reasoning model

- **Assess the likelihood of an attack** using HyReM
  - Use Bayesian Network model to accumulate and assess evidence and calculate the likelihood of a successful attack (i.e., a compromise).

- **Provide graphical output** to a user indicating the likelihood of an attack.
  - Graphical depiction of analysis and calculated ‘Likelihood of System Compromise’
  - Can be enhanced to interface with widely used network monitoring and alert tools.
Zero Day Attack Identification and Cyber Defense
Experimental Environment

Virtual Machine Configuration: Four VMs to monitor and analyze network traffic, attacker and victim.

- **SNORT Monitor (Backtrack 5.0 on Ubuntu Linux):**
  - Snort IDS (Live monitoring)
- **Zero Day Attack Monitor (Ubuntu Linux):**
  - Capture Tool (LibPCAP)
  - HyReM – Zero Day Attack Analysis/GUI (Demo)
- **Attacker (Backtrack 5.0 on Ubuntu Linux):**
  - Attacks (Metasploit, Minishare)
- **Victim (Win XP SP1 a, has numerous vulnerabilities):**
  - Clean snapshot re-instated after each experiment
## Experiment Results

One result per test.

<table>
<thead>
<tr>
<th>Test #</th>
<th>Pcap File</th>
<th>Pcap Packet Cnt</th>
<th>Description</th>
<th>Obs File</th>
<th>Total Obs Cnt</th>
<th>Likelihood of Compromise</th>
<th>Snort Alerts</th>
<th>Priority 1</th>
<th>Priority 2</th>
<th>Priority 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.pcap</td>
<td>2179</td>
<td>Metasploit exploit ms04-011</td>
<td>1_obs.pcap</td>
<td>89</td>
<td>0.9740</td>
<td>13</td>
<td>2</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2.pcap</td>
<td>2588</td>
<td>Metasploit exploit ms03-026</td>
<td>2_obs.pcap</td>
<td>105</td>
<td>0.9905</td>
<td>12</td>
<td>2</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3.pcap</td>
<td>2420</td>
<td>Metasploit exploit ms08-067</td>
<td>3_obs.pcap</td>
<td>102</td>
<td>0.9742</td>
<td>12</td>
<td>2</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4.pcap</td>
<td>1669</td>
<td>Minishare - noisy</td>
<td>4_obs.pcap</td>
<td>90</td>
<td>0.9753</td>
<td>13</td>
<td>2</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5.pcap</td>
<td>1642</td>
<td>Minishare - moderate</td>
<td>5_obs.pcap</td>
<td>103</td>
<td>0.9752</td>
<td>10</td>
<td>2</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>87.pcap</td>
<td>311+5339</td>
<td>Minishare - quiet</td>
<td>8_obs.pcap</td>
<td>6</td>
<td><strong>0.7803</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>6.pcap</td>
<td>325</td>
<td>Clean 1</td>
<td>6_obs.pcap</td>
<td>2</td>
<td>0.0141</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>7.pcap</td>
<td>5339</td>
<td>Clean 2</td>
<td>7_obs.pcap</td>
<td>0</td>
<td>0.0100</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>14.pcap</td>
<td>3848</td>
<td>Chronological merge of files 1 and 4</td>
<td>14_obs.pcap</td>
<td>161</td>
<td>0.9970</td>
<td>23</td>
<td>4</td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>26.pcap</td>
<td>2913</td>
<td>Chronological merge of files 2 and 6</td>
<td>26_obs.pcap</td>
<td>108</td>
<td>0.9915</td>
<td>12</td>
<td>2</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>46.pcap</td>
<td>1994</td>
<td>Chronological merge of files 4 and 6</td>
<td>46_obs.pcap</td>
<td>92</td>
<td>0.9792</td>
<td>13</td>
<td>2</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>57.pcap</td>
<td>6981</td>
<td>Chronological merge of files 5 and 7</td>
<td>57_obs.pcap</td>
<td>128</td>
<td>0.9752</td>
<td>15</td>
<td>2</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>347.pcap</td>
<td>9428</td>
<td>Chronological merge of files 3, 4, and 7</td>
<td>347_obs.pcap</td>
<td>219</td>
<td>0.9970</td>
<td>31</td>
<td>4</td>
<td>21</td>
<td>6</td>
</tr>
</tbody>
</table>

Our approach found an attack that the standard toolset missed.
Test Environment – Optimal Configuration

1a/b – Document Capture & HyReM Results
2 – Compare with Snort Alerting
3 – Inject Cyber Attacks
4 – Inject Live/Simulated Network Data Flow
Compromise Model and Observables

S1: No Vulnerability
T1: Vulnerability Research
(obsables)
unprocessed data to service (e.g., 80/443 pkts w/o HTML tags)

S2: Vulnerability Known
T2: Exploit Development and Testing
(obsables)
crashed services, unexpected payloads

S3: Exploit Known
T3: Reconnaissance
(obsables)
up (ping, ½ SYN), service (handshake w/o data, banner), service
(incomplete data, anomalous data, reset connections)

S4: Target Identified
T4: Exploit Execution
(obsables)
incomplete data, anomalous data, shell code, new service

S5: Target Compromised
T5: Cleaning Activities
(obsables)
communication on shell port, rm/del not on port 23

S6: Artifacts Cleaned
T6: Back Door Installation
(obsables)
exe file transfer, new port, new RDP listener

S7: Back Door Active
(obsables)
traffic on new port
Observable Modeling