Implementing Snort into SURFids

Sander Keemink and Michael van Kleij

February 6, 2008
Implementing Snort into SURFids

1. SURFids
2. Snort
3. Assignment
4. Experiments and results
5. Integrating Snort
6. Conclusion
7. Future work
Intrusion Detection System

- Detects unwanted activity
- Host based or Network based
Implementing Snort into SURFids

SURFids

Diagram showing network components and connections:
- Internet
- 1st VPN Tunnel
- Client LAN
- Sensor
- Loggingserver
- Webserver
- Observer
- Public Server LAN
- Tunnelsender + load balancing
- Private Server LAN
- Argos
- Nepenthes

SURFids
Implementing Snort into SURFids

Honeypots

**Nepenthes**
- Low interaction honeypot
- Simulates known vulnerabilities

**Argos**
- High interaction honeypot
- Analyses the operating system
## Nepenthes information

<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
<th>IP Address</th>
<th>Port</th>
<th>Protocol</th>
<th>Command</th>
<th>Antivirus</th>
</tr>
</thead>
<tbody>
<tr>
<td>30-01-2008 12:17:23</td>
<td>Malicious attack - Nepenthes</td>
<td>71.83.121.44</td>
<td>2323</td>
<td>TEST</td>
<td>Symantec AV</td>
<td></td>
</tr>
<tr>
<td>30-01-2008 12:51:44</td>
<td>Malicious attack - Nepenthes</td>
<td>83.206.104.118</td>
<td>5757</td>
<td>TEST</td>
<td>NetDDE</td>
<td></td>
</tr>
<tr>
<td>30-01-2008 13:00:24</td>
<td>Malicious attack - Nepenthes</td>
<td>71.147.32.143</td>
<td>57019</td>
<td>TEST</td>
<td>AGN1</td>
<td></td>
</tr>
<tr>
<td>30-01-2008 13:18:04</td>
<td>Malicious attack - Nepenthes</td>
<td>91.163.215.158</td>
<td>2958</td>
<td>TEST</td>
<td>Symantec AV</td>
<td></td>
</tr>
<tr>
<td>30-01-2008 13:19:11</td>
<td>Malicious attack - Nepenthes</td>
<td>91.171.126.127</td>
<td>4695</td>
<td>TEST</td>
<td>DCOM</td>
<td></td>
</tr>
<tr>
<td>30-01-2008 13:00:24</td>
<td>Possible malicious attack</td>
<td>71.147.32.143</td>
<td>57011</td>
<td>TEST</td>
<td>Symantec AV</td>
<td></td>
</tr>
<tr>
<td>30-01-2008 13:18:04</td>
<td>Possible malicious attack</td>
<td>91.163.215.158</td>
<td>2963</td>
<td>TEST</td>
<td>8555</td>
<td></td>
</tr>
<tr>
<td>30-01-2008 13:18:04</td>
<td>Possible malicious attack</td>
<td>91.163.215.158</td>
<td>2954</td>
<td>TEST</td>
<td>2967</td>
<td></td>
</tr>
<tr>
<td>30-01-2008 13:19:10</td>
<td>Possible malicious attack</td>
<td>91.171.126.127</td>
<td>4644</td>
<td>TEST</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>30-01-2008 13:19:11</td>
<td>Possible malicious attack</td>
<td>91.171.126.127</td>
<td>4655</td>
<td>TEST</td>
<td>135</td>
<td></td>
</tr>
</tbody>
</table>
## Argos information

### Details of attack ID: 487473

<table>
<thead>
<tr>
<th>Type</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argos ID</td>
<td>1713960475</td>
</tr>
<tr>
<td>Process ID</td>
<td>376</td>
</tr>
<tr>
<td>OS</td>
<td>win2k</td>
</tr>
<tr>
<td>Imagename</td>
<td>win2k.img</td>
</tr>
<tr>
<td>Module</td>
<td>svchost.exe</td>
</tr>
<tr>
<td>TCP Port</td>
<td>135</td>
</tr>
<tr>
<td>TCP Port</td>
<td>8721</td>
</tr>
<tr>
<td>TCP Port</td>
<td>1027</td>
</tr>
<tr>
<td>UDP Port</td>
<td>135</td>
</tr>
</tbody>
</table>
Implementing Snort into SURFids

Snort

Network Intrusion Detection System

Rule and anomaly based
Implementing Snort into SURFids

Assignment

Definition

"Which implementation of Snort into SURFids gives the most added value to the customer while not degrading performance in a noticable way."

Research questions

- Added value of Snort?
- Where to place Snort?
- How can Snort output be integrated?
Performance

**SURFids**
- 3 Mbits constant
- 30 Mbits max peaks

**Snort**
- 125 Mbits without packet loss
Experiments

1. Snort before Argos
2. Snort besides Argos and Nepenthes
3. Snort on the tunnel server
Experiment 1
Results experiment 1

Results

- Over 90% of the attacks registered by Argos were detected by Snort
- Other attacks also recognized
- Timeskew, Multiple entries per attack
Results experiment 2

Not conducted due to time and hardware limitations
Experiment 3
Results experiment 3

Over 90% of the attacks registered by Nepenthes were detected by Snort

Identification of 10% of the possible malicious attacks
Integrating Snort

Barnyard, a Snort output processor

- Offloads Snort
- Supports multiple output formats
- Database aware
Integrating Snort

Develop a database output plugin
- Shortest path
- IP packet payload information

Parse Comma Seperated Value output
- Relative easy to develop
- No IP packet payload information
Conclusion

Snort provides added value to SURFids

Nepenthes possible malicious attacks can be discarded
Future work

Develop a program that deals with Snort output