From Students…
…to Professionals

MICHIGAN STATE UNIVERSITY

Project Plan
Enhanced Network Anomaly Detection Suite
The Capstone Experience

Team Rook Security
Cam Gibson
Brian Harazim
Grant Levene
Zach Rosenthal
Andrew Werner

Department of Computer Science and Engineering
Michigan State University
Fall 2016
Functional Specifications

Monitors highly-virtualized networks to detect behavior-based attacks

• Optimize Windows agent performance
• Improve analysis engine with machine learning
• Develop agent management console GUI
• Create Linux and OS X agent versions
• Add encryption for all communications
• Add encrypted local database to the agents
Design Specifications

Web Management Console Features

• Agent health and directory
• Host health and directory
• Anomaly alerts via email, dashboard, and push notifications
• Network statistics
• Remote agent management
Screen Mockup: Home Page

ROOK SECURITY
NETWORK ANOMALY DETECTION SUITE

- 42% CPU usage
- 384 / 511 agents online
- 7 detections today

Notable Agents

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Platform</th>
<th>Health</th>
<th>Last Update</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPA57F7</td>
<td>Windows 7</td>
<td>15%</td>
<td>3h</td>
<td></td>
</tr>
<tr>
<td>DX507</td>
<td>Unix (Arch)</td>
<td>17%</td>
<td>1d 7h</td>
<td></td>
</tr>
<tr>
<td>AT789</td>
<td>Windows 7</td>
<td>14%</td>
<td>4h</td>
<td></td>
</tr>
<tr>
<td>TRT219</td>
<td>Windows 10</td>
<td>77%</td>
<td>10d 8h</td>
<td></td>
</tr>
<tr>
<td>TRT731</td>
<td>Unix (Ubuntu)</td>
<td>56%</td>
<td>1d 18h</td>
<td></td>
</tr>
<tr>
<td>V0007B</td>
<td>Windows 7</td>
<td>66%</td>
<td>7h</td>
<td></td>
</tr>
<tr>
<td>V0006</td>
<td>Windows 7</td>
<td>80%</td>
<td>8h</td>
<td></td>
</tr>
<tr>
<td>LPC73</td>
<td>Windows 7</td>
<td>26%</td>
<td>4d</td>
<td></td>
</tr>
</tbody>
</table>
Screen Mockup: Data Visualization
Screen Mockup: Agent Management

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Platform</th>
<th>Health</th>
<th>Last Update</th>
<th>Version</th>
<th>Department</th>
<th>User</th>
<th>Status</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPA077</td>
<td>Windows 7</td>
<td>55%</td>
<td>3h</td>
<td>12.3</td>
<td>Human Resources</td>
<td>zhe0080</td>
<td>Healthy</td>
<td></td>
</tr>
<tr>
<td>LS507</td>
<td>Unix (Arch)</td>
<td>17%</td>
<td>1d 7h</td>
<td>12.1</td>
<td>Quality Assurance</td>
<td>hdp3368</td>
<td>Needs Update</td>
<td></td>
</tr>
<tr>
<td>AT879</td>
<td>Windows 7</td>
<td>14%</td>
<td>4h</td>
<td>12.3</td>
<td>Human Resources</td>
<td>jdh7712</td>
<td>Healthy</td>
<td></td>
</tr>
<tr>
<td>TRT210</td>
<td>Windows 10</td>
<td>77%</td>
<td>100d 8h</td>
<td>12.3</td>
<td>Mobile Techno..</td>
<td>bkr8791</td>
<td>Healthy</td>
<td></td>
</tr>
<tr>
<td>TRT713</td>
<td>Unix (Ubuntu)</td>
<td>56%</td>
<td>1d 16h</td>
<td>12.2</td>
<td>Systems</td>
<td>ldd8668</td>
<td>Anomaly Detected</td>
<td></td>
</tr>
<tr>
<td>VCD273</td>
<td>Windows 7</td>
<td>68%</td>
<td>7h</td>
<td>11.0</td>
<td>Database Admin..</td>
<td>lbd0665</td>
<td>Needs Update</td>
<td></td>
</tr>
<tr>
<td>VCD266</td>
<td>Windows 7</td>
<td>80%</td>
<td>8h</td>
<td>10.1</td>
<td>Human Resources</td>
<td>znh0085</td>
<td>Needs Update</td>
<td></td>
</tr>
<tr>
<td>LPC73</td>
<td>Windows 7</td>
<td>28%</td>
<td>4d</td>
<td>12.3</td>
<td>Quality Assurance</td>
<td>yzc9434</td>
<td>Anomalies Detected</td>
<td></td>
</tr>
</tbody>
</table>
Technical Specifications

• Management Console
  ▪ Frontend: ReactJS
  ▪ Backend: Django Rest Framework
  ▪ Message Queue: Apache Kafka

• Machine Learning
  ▪ Octave
  ▪ Clustering Libraries: Graphlab-create, HDBScan
  ▪ NumPy

• Environment
  ▪ Containerization with Docker Compose
System Components

- Hardware Platforms
  - Rack Servers
  - Network Clients

- Software Platforms
  - Windows
  - Linux / Unix
  - OS X

Software Technologies
- Docker / Docker Compose
- C
- Python (Django)
- Daphne
- Apache Kafka
- ReactJS
- HTML / CSS
- MaterialUI
- OpenSSL
- Graphlab-create
- HDBScan
- NumPy
Testing

- Frontend: Jest.js
- Backend: Django Test Framework
- API Endpoints: Postman
- Unit and Integration tests
Risks

• Limited knowledge of technologies
  ▪ Django, Apache Kafka, Daphne, and Windows development
  ▪ Write simple prototypes using these technologies

• Getting enough traffic to do testing
  ▪ Software requires a high volume of traffic to gather data
  ▪ Simulate different attacks to try and catch

• Secure code and keeping software secure
  ▪ Writing secure code and protecting the company’s software
  ▪ Learn what secure code is, and locking our computers

• Machine learning getting baseline dataset
  ▪ Realistic dataset for machine learning algorithms to “learn” from
  ▪ Understand machine learning and simulate normal network traffic