Alpha Presentation
Force Platform Ingestion Tool

The Capstone Experience

Team Rook
Roy Barnes
Matt Hammerly
Will McGee
Chiyu Song
Mark Velez

Department of Computer Science and Engineering
Michigan State University
Spring 2017
Project Overview

• Integration into Rook’s new Force platform
• Enhance analyst efficiency in daily work
• Provide an easy way to integrate new clients
• Machine learning to improve alert correlation
Login Page

FORCE PIT

Username or Email

Password

Login
Login Page - Error Message
## Alerts Page

![Force PIT Alerts](image)

<table>
<thead>
<tr>
<th>Company</th>
<th>IP Address</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company 1</td>
<td>62.14.233.98</td>
<td>2016-08-15 11:14</td>
</tr>
<tr>
<td>Company 2</td>
<td>test ip</td>
<td>TEST TIME</td>
</tr>
<tr>
<td>Company 3</td>
<td>test ip</td>
<td>TEST TIME</td>
</tr>
<tr>
<td>Company 4</td>
<td>test ip</td>
<td>TEST TIME</td>
</tr>
<tr>
<td>Company 5</td>
<td>test ip</td>
<td>TEST TIME</td>
</tr>
<tr>
<td>Company 6</td>
<td>test ip</td>
<td>TEST TIME</td>
</tr>
<tr>
<td>Company 7</td>
<td>test ip</td>
<td>TEST TIME</td>
</tr>
<tr>
<td>Company 8</td>
<td>test ip</td>
<td>TEST TIME</td>
</tr>
<tr>
<td>Company 9</td>
<td>test ip</td>
<td>TEST TIME</td>
</tr>
<tr>
<td>Company 10</td>
<td>test ip</td>
<td>TEST TIME</td>
</tr>
<tr>
<td>Company 11</td>
<td>test ip</td>
<td>TEST TIME</td>
</tr>
</tbody>
</table>
Alerts Page - Selected Alert
Configuring an API

### Configured APIs

<table>
<thead>
<tr>
<th>Configure a new API</th>
<th>Name</th>
<th>URL</th>
<th>Interval (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No APIs have been configured yet. Try configuring one!</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Configuring an API
Configuring an API

New API configuration

Connection details

Name: Random Alert Job

Url: 

Run every mins: 

Data type: JSON

Http method: GET

Add HTTP header Remove HTTP header

Name Value

Test connection
Configuring an API

New API configuration

Connection details

- Name: Random Alert Job
- URL: /datasource/getmerok
- Run every mins: 1
- Data type: JSON
- HTTP method: GET

Test connection
Configuring an API

New API configuration

Connection details
Name: Random Alert Job
Url: 0/dotparser/getmeraki
Run every mins: 1
Data type: JSON
Http method: POST

Post data:

```
{"random":"true",
"number":"1"
}
```

Test connection
Configuring an API
Configuring an API
Configuring an API
Configuring an API

The Capstone Experience
Team Rook Alpha Presentation
Configuring an API
Configuring an API

Configured APIs

<table>
<thead>
<tr>
<th>Name</th>
<th>URL</th>
<th>Interval (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Alert Job</td>
<td><a href="http://127.0.0.1:8000/dataparser/getmeraki/">http://127.0.0.1:8000/dataparser/getmeraki/</a></td>
<td>1</td>
</tr>
</tbody>
</table>
Configuring an API

Data pulled by Random Alert Job (latest 0)

Normalizing data from http://127.0.0.1:8000/dataparser/getmeraki/ every 1 minute(s).

Alert mapping:
"Description": "key('\_source\').key('Message')", "AlertID": "key('\_id')", "HostID": "key('\_source\').key('host')"

Time Fetched | Raw | Normalized
This job has not been run yet.
Configuring an API

Data pulled by Random Alert Job (latest 1)

Normalizing data from http://127.0.0.1:8000/dataparser/getmeraki/ every 1 minute(s).

Alert mapping:

```
{"Description": "key":{"_source"}", "key":{"Message"}, "AlertID": "key":{"_id"}, "HostIP": "key":{"_source"}.key{"host"})
```

<table>
<thead>
<tr>
<th>Time</th>
<th>Raw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb. 20, 2017, 9:38 p.m.</td>
<td></td>
</tr>
</tbody>
</table>

```
{
    "id": "AVnxugLdBSPinseey4Q2",
    "index": "logstash-example_customer-2017.01.30",
    "_score": 1,
    "_source": {
        "@timestamp": "2017-01-30T23:33:35.263Z",
        "version": "1",
        "data": {
            "host": "10.231.112.60",
            "source": "example_customer",
            "src_ip": "10.231.112.60",
            "src_port": 63718,
            "timestamp": "2017-01-30T23:33:33.848Z",
            "type": "meraki"
        }
    }
}
```
Configuring an API
ML Clustering into Cases

Scikit-Learn Clustering on Test Data
What’s left to do?

• Make UI design cohesive, get Rook feedback

• Use Django “Channels” library to update React/Redux UI in real time

• Finish ML, append to data normalization flow

• Build out support for as many APIs as we can
Questions?