Applying Data Mining Techniques to Identify Malicious Actors

Techniques For Turning Data Into Action

Balaji Balakrishnan
Disclaimer:
Opinions expressed in this presentation are my own.

I am speaking for myself, not for my employer, anyone or anything else.
Agenda

Introduction

Threat hunting platform

Data mining techniques

Key takeaways

Conclusion
Introduction
Main () {
    printf("I’m Balaji, and I have more than 16 years of experience working in Information Technology and Information Security (security operations and incident response), primarily in the financial services domain");
}
Agenda

Introduction

**Threat hunting platform**

Data mining techniques

Key takeaways

Conclusion
Why Threat Hunting?

Benefits of Threat Hunting

- 52% say threat hunting found previously undetected threats on their enterprise
- 74% of those implementing threat hunting have reduced attack surfaces
- 59% enhanced speed and accuracy of response by using threat hunting

Source: SANS

Maturity Model

Figure 15. Maturity of Threat-Hunting Techniques

- Basic Search: 85%
- Statistical Analysis: 55%
- Visualization Techniques: 50%
- Simple Aggregations: 48%
- Machine Learning: 21%
- Bayesian Probability: 11%

Source: SANS
Threat Hunting Platform

- Threat Hunting Platform (Big Data Analytics platform)
  - SIEM+TIP+OSINT+Other data sources (context)

- Capabilities
  - Basic search + Pivoting + Graph + Visualization – Context for SOC Analyst
    - Create tools for SOC analysts for exploration, visualization, orchestration for real time contextual analysis
    - Useful in IOC search, rule based and adversary TTP based pivoting

- Integration with Python/R scripting - Advanced analytics/ML for Security data analytics team
  - Advanced analytics/machine learning algorithms
Example: OpenSOC - Stitching Things Together

**Source Systems**
- Passive Tap
- Traffic Replicator

**Data Collection**
- PCAP

**Telemetry Sources**
- Syslog
- HTTP
- File System
- Other

**Messaging System**
- Kafka
- PCAP Topic
- DPI Topic
- A Topic
- B Topic
- N Topic

**Real Time Processing**
- Storm
  - PCAP Topology
  - DPI Topology
  - A Topology
  - B Topology
  - N Topology

**Storage**
- Hive
  - Raw Data
  - ORC

**Elastic Search**
- Index

**Access**
- Analytic Tools
  - R / Python
  - Power Pivot
  - Tableau
- Web Services
  - Search
  - PCAP Reconstruction

**Flume**
- Flume
- Agent A
- Agent B
- Agent N
Data Collection - Host

• What?
  • ShimCache, AmCache, Scheduled tasks, Process list, Services, Drivers, Autoruns
  • Prefetch, Browser history
  • Hash of running processes, downloaded files
  • Event logs
  • Command line history
  • AV, HIDS, HIPS logs

• How?
  • GRR, PSRECON, irCRpull(CrowdResponse)
  • FCIV
  • Carbonblack
Data Collection - Network

• What?
  • Netflow
  • DNS, passive DNS
  • PCAP
  • Firewall, NIPS, NIDS

• How?
  • OpenSOC, ONI

Data Collection - Other Sources

• What?
  • Access logs, authentication, authorization and audit logs; Application logs; HR data; Physical access logs; OSINT; Open Source Intel Feeds; E-Mails / PDF parsing; Sandbox analysis; Honeypot; Vulnerability scanning data; Incident data

• How?
  • API integration, Syslog, Python
Threat Hunting using Data Science (1 of 2)

• Threat Hunting Process
  • Generate use cases
  • Develop analytical techniques
  • Move it to production for SOC analyst and other teams to use
  • Continuous feedback and improvement of analytical techniques based on usage

• Triggers
  • Cyber Intel Feeds – IOCs
  • Threat actors TTPs
  • Correlation alerts
  • Security events
  • Security Distributed Alerting
  • Purple team exercises
  • Post incident analysis
  • Honeynet/Honeytoken
Threat Hunting using Data Science (2 of 2)

• What are the enablers?
• Threat Hunting Platform examples
  • OpenSOC, ELK, RITA, Hadoop, ONI, Splunk, Sqrrl
• Data Collection (host, network, application, contextual)
• Threat Intelligence Sharing Automation (STIX/TAXII)
• Cloud technologies rapidly evolving
• Big data analytics technologies rapidly evolving

• Key Takeaways
  • Build Threat Hunting Platform integrating all data sources
  • Create Threat Hunting process integrating all teams (SOC, TVM, admins etc)
Agenda

Introduction

Threat hunting platform

Data mining techniques

Key takeaways

Conclusion
Data Science Techniques

• Approaches
  • Exploration & Visualization
    • Graph
    • Parallel coordinates
  • Statistical Analysis
    • Top talkers & Long tail analysis
    • Using Baselines
    • Risk Scoring
    • Natural Language Processing
    • Time series analysis
  • Machine Learning
    • Supervised learning
      • Classification
      • Regression
    • Unsupervised learning
      • Clustering

….and Skills
Exploration and Visualization - Graph example – neo4j
Parallel Coordinates - Multidimensional search example
Statistical Analysis - Risk Scoring Methodology

- Step 1 – Identify anomalous events based on baseline or threshold

- Step 2 – Assign risk scores for each user/identity for each anomalous event

- Step 3 - Aggregate all the risk scores per day to identify top user/identity that requires further investigation to determine the threat activity involved.
Risk Scores – Splunk Example

• **Step 1 – Identify anomalous events based on baseline or threshold**
• **Step 2 – Assign risk scores for each user/identity for each anomalous event**

```splunk
index=loginduration
| eval dhour=duratio...on/3600
| eval Risk_Score=0
| eval Risk_Score=if((dhour>8),Risk_Score+20,Risk_Score+0)
| table _time,user,Risk_Score
| collect index=userriskscore
```

```splunk
index=dcount
| eventstats avg(count) as avgcount, stdev(count) as stdevc
| where (count > avgcount + 2 * stdevc) or (count < avgcount – 2 * stdevc)
| eval Risk_Score=0
| eval Risk_Score=Risk_Score+20
| table _time,user,Risk_Score
| collect index=userriskscore
```
Risk Scores – Splunk Example

• Step 3- Aggregate all the risk scores per day to identify top user/identity that requires further investigation to determine the threat activity involved.

index=userriskscore
| stats sum(Risk_Score) by _time user
| rename sum(Risk_Score) as Total_Risk_Score
| sort---Risk_Score
Risk Score Model using Statistical Deviations

• Simple User/Network Behavior Analytics - A complete statistical model can be applied to the daily user/network activity to calculate anomalous events:
  
  • Calculate the Average and the Standard Deviation for each User/network behavior value on Daily, Weekly, and Monthly time windows
  • Daily comparison of the User/network behavior for each activity on that assessed Day, the prior Week from the current day, and previous Month from the current day
  • All calculated values that are sufficiently different from the average via standard deviation comparison are to be identified as anomalous and assigned a risk score
  • The User/network behaviors on a daily basis with the highest risk scores across the Daily, Weekly, and Monthly measurements are to be identified as highest potential risk
Statistical Analysis – Time Series Analysis

• Methods
  • Baseline
  • Simple Moving Average
  • Exponential Moving Average
  • Weighted Moving Average
  • FFT
  • Timeline Analysis - Plaso

• Use cases
  • Beaconing
  • Login failure ratio
  • SSH Bruteforce
  • Scanning
  • File creation times
  • Denial of Service
Baseline (DDOS) Example

- Baseline was created based on the average traffic to these targets Hourly/Daily/Weekly/Monthly.
- Based on the Baseline, the live traffic was compared.
- In case of deviation, alarm is set to trigger.
- Very simple but very powerful.
Simple Moving Average

sourcetype=access*
| timechart avg(bytes) as avg_bytes
| trendline sma5(avg_bytes) as moving_avg_bytes
| eval spike=if(avg_bytes > 2 * moving_avg_bytes, 10000, 0)
likely_beacons (FFT)--Customer: example
Src: 192.168.0.3 Dest: 192.168.0.4 Proto: DstPort: 10

Detecting A Two Second Beacon
Machine Learning Steps

• Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data.

Supervised Learning

• Target values known
• Training data labeled with target values
• Train model to map data object to target value
• Classification Algorithms
  • Neural Networks
  • Random Forests
  • Support Vector Machines (SVM)
  • Decision Trees
  • Logistic Regression
  • Naive Bayes
Unsupervised Learning

• Trying to find hidden structure in unlabeled data
• No error or reward signal to evaluate a potential solution
• Common techniques: K-Means clustering, Hierarchical clustering, hidden Markov models, etc.

K-Means Clustering

• Process of partitioning data points into similarity clusters
• Unsupervised technique
• Only works for numeric data
Text Classification

Source: https://gallery.cortanaintelligence.com/Experiment/f43e79f47d8a4219bf8613d271ea2c45

Train and Evaluate Model
**N-grams TF Feature Extraction**

Reader
- read the preprocessed train... text from BLOB Storage

Metadata Editor
- set the text column type as non-categorical string

Feature Hashing
- extract numeric features: get the occurrence frequency of n-grams in text

Filter Based Feature Selection
- select the N most relevant features

**Deploy**

Reader
- read the single instance dataset from BLOB Storage

Reader
- read the sentiment stop word list from BLOB Storage

Execute R Script
- text preprocessing

Feature Hashing
- extract numeric features: get the occurrence frequency of n-grams in text

Score Model

Project Columns
- select the score columns

Web service output
STEP 2: THE **FIRST STAGE** MALWARE IS **EXECUTED**

Once the **Upatre malware** is executed, its sole purpose is to **download** **Dyre**.

This is completed in a few **stages**.

It’s important to note that this **stage** of the process is **completely dynamic**.

**URLs** and **payloads** are **constantly shifting** in order to **evade detection**.

The **Upatre malware** itself **constantly evolves** and remains **obfuscated** allowing it to **evade antivirus measures** as well.

1) **Upatre** contacts **checkip.dyndns.org** in order to determine the **public IP address** of the machine it is on.

This **website** replies with a simple message: ‘**Current IP Address: x.x.x.x**’.

The malware uses this **information** to understand who it has **infected**.

2) Next, a **STUN** (**Session Traversal Utilities for NAT**) server is contacted to determine the **public IP address** and the type of **NAT** (**Network Address Translation**). Service it’s **sitting behind**.

3) Internet **connectivity** is checked to determine if a **proxy** is being utilized by contacting google.com.

4) **Upatre** makes its initial contact with the **Command & Control (C&C)** server.

5) **Upatre** downloads **Dyre** from a varied list of domains as well as changing filenames.

For example, `metflex(.)uk(.)com` hosted a file named `Æœť İmage.jpg`, which is the **Dyre** malware.
Advantages of building data science capabilities – Different use cases...

Once developed, data science tools can be used to solve many use cases

- Threat hunting
- Threat Intelligence aggregation from various sources
- Incident response/Forensics
- Vulnerability remediation prioritization
- Risk management
- Security Automation/Orchestration
- User/Network Behavior Analytics
- Fraud detection
- Automating CIS Critical Security Controls
- Cloud Access Security Monitoring

...that include the prevention of cyber crime and the actions of bad actors

Conclusion
Key Takeaways & Conclusion

• Need lot of patience, difficult to get immediate results
• Integration of the different components is challenge
• Quick wins
  • Post Incident Analytics
  • Purple team exercises
  • Working sessions with SOC, TVM and operations/admins
• Long way to go, the journey forward seems exciting
Questions