# SHERLOCK

ADVANCED PERSISTENT THREAT DETECTION SYSTEM

*RONY XAVIER*

## PROJECT DESCRIPTION

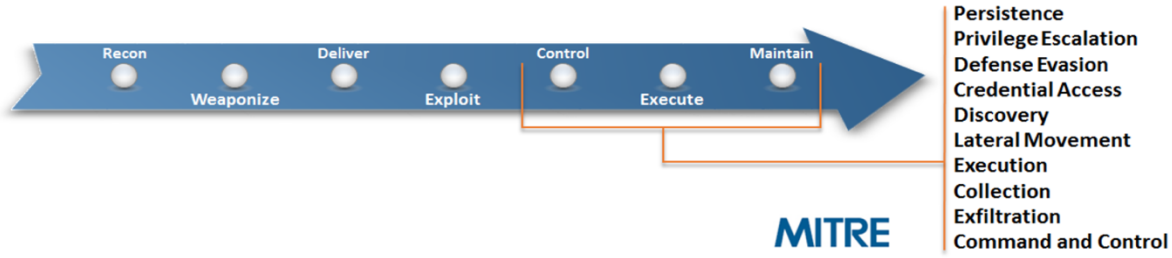
### ADVANCED PERSISTANT THREAT

An advanced persistent threat (APT) is a network attack in which an unauthorized person gains access to a network and stays there undetected for a long period of time. The intention of an APT attack is to steal data rather than to cause damage to the network or organization. [1] Since APTs are usually targeted, it is not easily detected by traditional methods such as hash signature or yara rules.

### CYBER KILL CHAIN & MITRE ATT&CK

Developed by Lockheed Martin, the Cyber Kill Chain® framework is part of the Intelligence Driven Defense model for identification and prevention of cyber intrusions activity. The model identifies what the adversaries must complete in order to achieve their objective. [2]

MITRE's Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK™) is a curated knowledge base and model for cyber adversary behavior, reflecting the various phases of an adversary's lifecycle and the platforms they are known to target [3]. ATT&CK expands on the Cyber Kill Chain stages after 'Exploit'.




### MITRE ATT&CK MATRIX

The ATT&CK Matrix for Enterprise provides a visual representation of the adversarial techniques described in the ATT&CK for Enterprise threat model.

Tactic categories are listed on the top row, individual techniques as cells underneath each tactic to denote that technique can be used to accomplish that particular tactic. Techniques can span multiple tactic categories signifying that they can be used for more than one purpose. [4]

| Persistence | Privilege Escalation | Defense Evasion | Credential Access | Discovery | Lateral Movement | Execution | Collection | Exfiltration | Command and Control |
|---|---|---|---|---|---|---|---|---|---|
| Accessibility Features | Accessibility Features | Binary Padding | Brute Force | Account Discovery | Application Deployment Software | Command-Line Interface | Audio Capture | Automated Exfiltration | Commonly Used Port |
| AppInit DLLs | AppInit DLLs | Bypass User Account Control | Credential Dumping | Application Window Discovery | Exploitation of Vulnerability | Execution through API | Automated Collection | Data Compressed | Communication Through Removable Media |
| Authentication Package | Bypass User Account Control | Code Signing | Credential Manipulation | File and Directory Discovery | Logon Scripts | Execution through Module Load | Clipboard Data | Data Encrypted | Connection Proxy |
| Basic Input/Output System | DLL Injection | Component Firmware | Credentials in Files | Local Network Configuration Discovery | Pass the Hash | Graphical User Interface | Data Staged | Data Transfer Size Limits | Custom Command and Control Protocol |
| Bootkit | DLL Search Order Hijacking | Component Object Model Hijacking | Exploitation of Vulnerability | Local Network Connections Discovery | Pass the Ticket | InstallUtil | Data from Local System | Exfiltration Over Alternative Protocol | Custom Cryptographic Protocol |
| Change Default File Association | Exploitation of Vulnerability | DLL Injection | Input Capture | Network Service Scanning | Remote Desktop Protocol | MSBuild | Data from Network Shared Drive | Exfiltration Over Command and Control Channel | Data Encoding |

## CYBER ANALYTIC REPOSITORY

The Cyber Analytics Repository (CAR) is a knowledge base of analytics developed by MITRE based on the Adversary Tactics, Techniques, and Common Knowledge (ATT&CK™) adversary model.

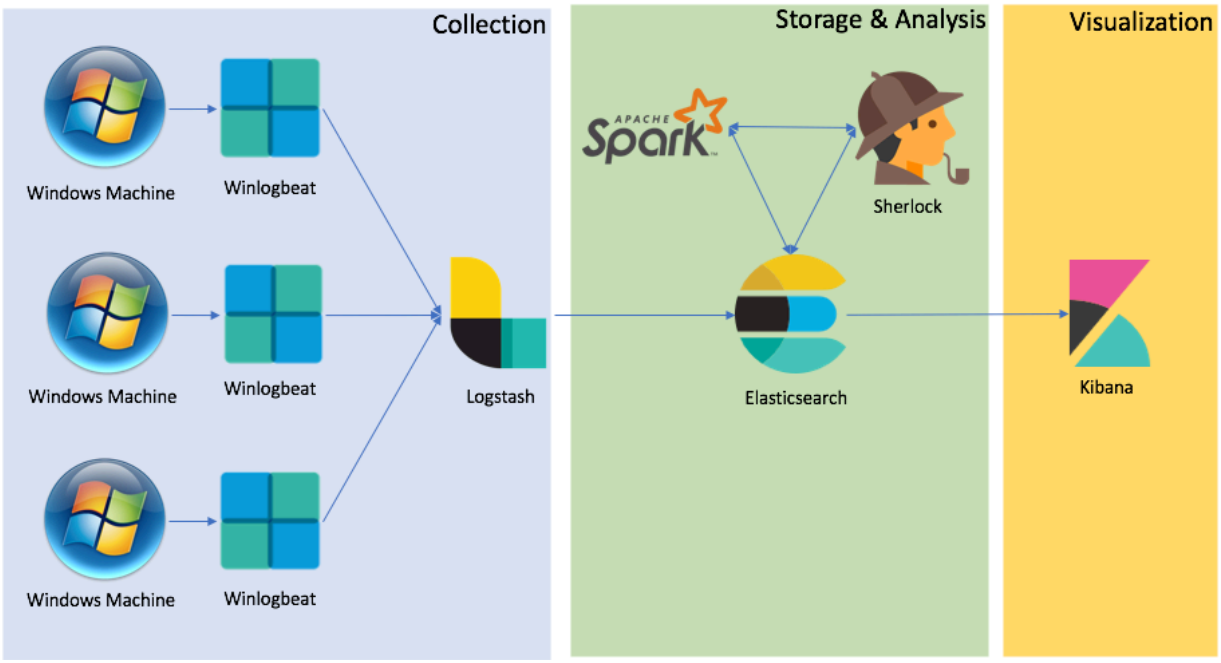
Analytics stored in CAR contain the following information, a hypothesis which explains the idea behind the analytic, the information domain or the primary domain the analytic is designed to operate within (e.g. host, network, process, external) references to ATT&CK Techniques and Tactics that the analytic detects the type of analytic.

## PURPOSE

Sherlock is designed to correlate possible adversary tactics detected through the analysis, to stages in the ATT&CK Frame will expose systems that are at a higher probability of infection. Sherlock visualizes its findings and let the user see the patterns and make connections. It is not designed to replace and anti-virus tool since with Sherlock the user makes the decisions and remediation and not software.

# PROJECT DESIGN

## DESIGN DIAGRAM




## TECHNOLOGIES

| Tool | Role | Location |
|---|---|---|
| **Windows 2012** | Test Host | VMware Fusion |
| **Windows 7** | Test Host | VMware Fusion |
| **Sysmon** | Log Source | Windows Host (VMware Fusion) |
| **Windows Event Log** | Log Source | Windows Host (VMware Fusion) |
| **WinlogBeat 5.6.3** | Collection | Windows Host (VMware Fusion) |
| **Logstash 5.6.3** | Collection | Ubuntu (VMware Fusion) |
| **Elasticsearch 5.6.3** | Storage | Ubuntu (VMware Fusion) |
| **Kibana 5.6.3** | Visualization | Ubuntu (VMware Fusion) |

| | | |
|---|---|---|
| **Python 3.4.2** | Analysis | Zeppelin Debian GNU/Linux 8 (Docker) |
| **Spark 2.2.0** | Analysis | Zeppelin Debian GNU/Linux 8 (Docker) |
| **DataFrames** | Analysis | Spark 2.2.0 |
| **Elasticsearch-Hadoop 6.0.1** | Analysis | Zeppelin Debian GNU/Linux 8 (Docker) |
| **Python 3.4.2** | Analysis | Zeppelin Debian GNU/Linux 8 (Docker) |
| **Sherlock 1.0** | Analysis | Zeppelin Debian GNU/Linux 8 (Docker) |
| **macOS High Sierra** | WorkStation | MacBook Pro (13-inch, Mid 2012) |

## COLLECTION

Log sources for the analysis are the following sensors:

- Sysmon
  - Process Events Logs
  - Network Events Logs
- Windows Event Logs
  - System Event Logs
  - Security Event Logs
  - Application Event Logs

All logs are collected using WinlogBeat and shipped to Logstash. Multiple hosts ship logs to the instance of Logstash which is shipped to the instance of Elasticsearch.

## STORAGE

Logstash ships logs into the instance of Elasticsearch at index: ‘winlogbeat*’.

The same instance of Elasticsearch also saves the analyzed logs from ‘Sherlock’ at index: ‘sherlock’.

The mapping for the index ‘sherlock’ is defined in ‘[SHERLOCK_HOME]/es_mapping/ sherlock_mapping.json’

## ANALYSIS

Sherlock collects logs from the Elasticsearch index ‘winlogbeat*’ using ‘Elastic-Hadoop’ plugin. The collected logs are analyzed using CAR analytic codes and shipped back into Elasticsearch at index ‘sherlock’ for review.

## CODE STRUCTURE

Sherlock is designed to be a live analysis tool; however, it can be started with a DAYS_OFFET parameter to start evaluation from past logs. Once the past logs have been evaluated it switches to live analysis.

Sherlock collects and instantiates all analytic code from the '[SHERLOCK_HOME]/CAR_FILES' folder automatically hence additional analytic can be added to the folder following the correct code style and Sherlock will be able to analyze them.

Sherlock creates time-slices of event logs and runs the analytic code against it. Since the analytic code could require different length of time-slices, Sherlock creates time-slices based on the 'duration' parameter defined within the analytic.

If the required timeframe for the analytic is available, it will be executed, if not it will move on to the next analytic or wait for the required timeframe to be available.

Below is an example of an analytic describing the required style.

CAR-2013-02-003: Processes Spawning cmd.exe

```
TECHNIQUES = ['Command-Line Interface']
TACTICS = ['Execution']
DURATION_MINS = 30

from pyspark.sql.functions import *

class CAR_2013_02_003():
    def __init__(self):
        self.time = 0
        self.duration = DURATION_MINS
        self.tactics = TACTICS
        self.techniques = TECHNIQUES
        self.df = 0

    def analyze(self):
        sysmon_df = self.df.where(col('log_name') == 'Microsoft-Windows-Sysmon/Operational')
        process_create_events = sysmon_df.where(col('event_id') == 1)
        events = process_create_events.where(col('event_data.Image').rlike("cmd.exe"))
        events = events.where(col('event_data.ParentImage').rlike("explorer.exe") == False)
        return events
```

The file name must match the class name. The appropriate TECHNIQUES and TACTICS and DURATION must be defined. The analytic code is defined within the analyze method.

## ANALYTIC CODE

Following is the list of CAR Analytics implemented in Sherlock.

| CAR Index | Description |
|---|---|
| **CAR-2013-02-003** | Processes Spawning cmd.exe |
| **CAR-2013-03-001** | Reg.exe called from Command Shell |
| **CAR-2013-05-002** | Suspicious Run Locations |
| **CAR-2013-05-003** | SMB Write Request |
| **CAR-2013-05-004** | Execution with AT |
| **CAR-2013-07-002** | RDP Connection Detection |
| **CAR-2013-07-005** | Command Line Usage of Archiving Software |
| **CAR-2013-08-001** | Execution with schtasks |
| **CAR-2014-03-006** | RunDLL32.exe monitoring |
| **CAR-2014-04-003** | Powershell Execution |
| **CAR-2014-05-001** | RPC Activity |
| **CAR-2014-05-002** | Services launching Cmd |
| **CAR-2014-07-001** | Service Search Path Interception |
| **CAR-2014-11-003** | Debuggers for Accessibility Applications |
| **CAR-2014-11-004** | Remote PowerShell Sessions |
| **CAR-2014-11-006** | Windows Remote Management (WinRM) |
| **CAR-2014-11-008** | Command Launched from WinLogon |
| **CAR-2016-03-001** | Host Discovery Commands |
| **CAR-2016-03-002** | Create Remote Process via WMIC |
| **CAR-2016-04-002** | User Activity from Clearing Event Logs |
| **CAR-2016-04-003** | User Activity from Stopping Windows Defensive Services |
| **CAR-2016-04-004** | Successful Local Account Login |
| **CAR-2016-04-005** | Remote Desktop Logon |

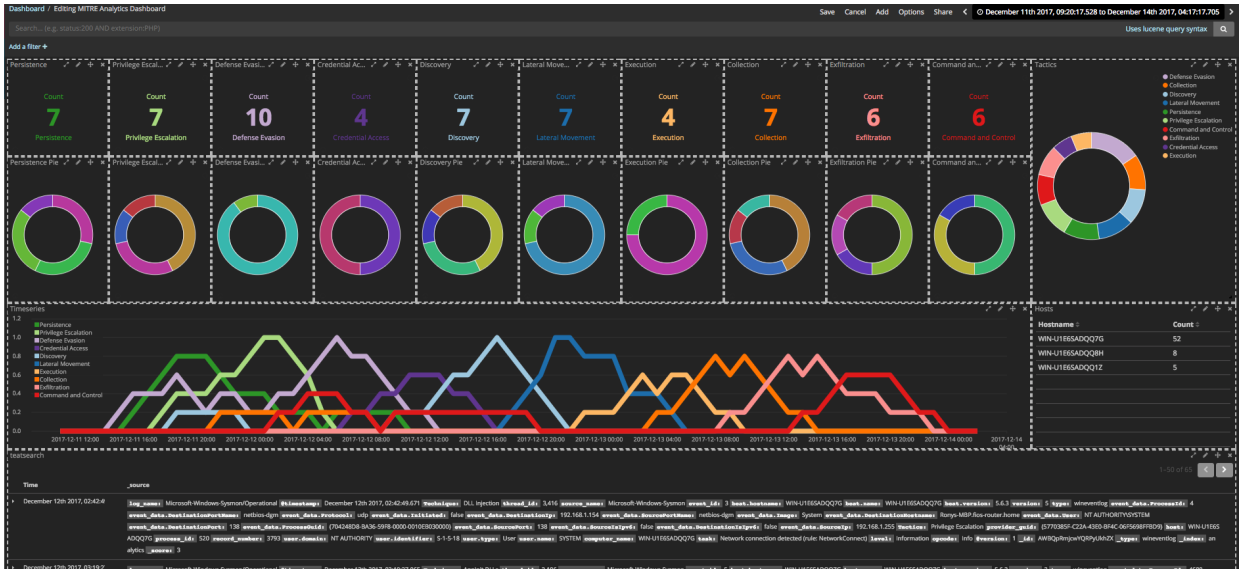
## EXECUTION


Run Command:

spark-submit --jars elasticsearch-hadoop-6.0.1.jar sherlock.py

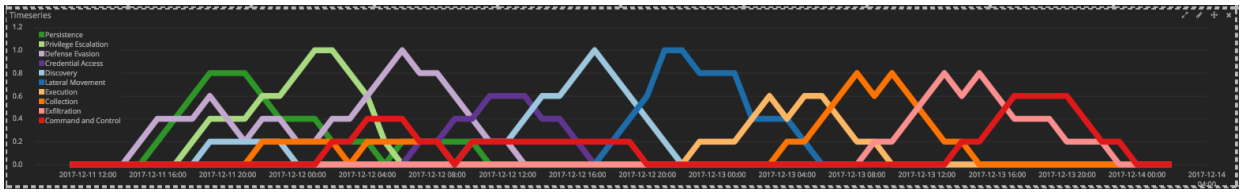
## VISUALIZATION

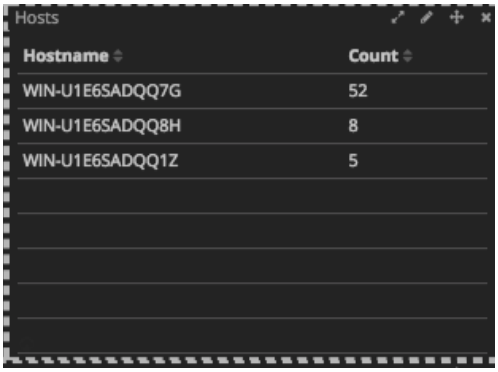
The findings from Sherlock analysis is visualized using Kibana Dashboard, image below shows the complete Sherlock Dashboard.



Below is a zoom in of the ‘Tactics’ and ‘Techniques’ part of the dashboard. The count of indicators of each tactic is shown in the top row. Row below shows a pie chart of ‘Techniques’ within each ‘Tactic’. Pie chart on the right shows ‘Tactics’. The Pie Charts can be used to filter data.

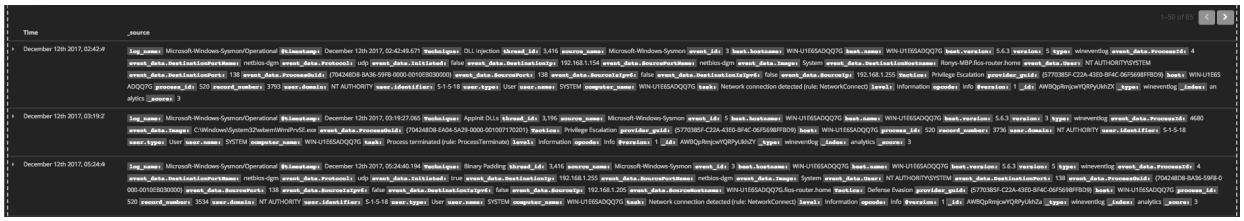

Below is the zoom in of the ‘Timeseries’ part of the dashboard which can help review indicators of ‘Tactics’ in a time-based view. MITRE ATT&CK claims that the adversary will naturally progress the ‘Tactics’ in sequence, hence ‘Timeseries’ can be useful to view the progression of a possible adversary. Data can be filtered using the ‘Timeseries’ chart as well.




On the left is a zoom in of the Host section showing a list of analyzed hosts. Hosts can be filtered here to review individual systems.

Below is a zoom in of the Log view of the Sherlock Dashboard where individual logs showing indicators can be reviewed.

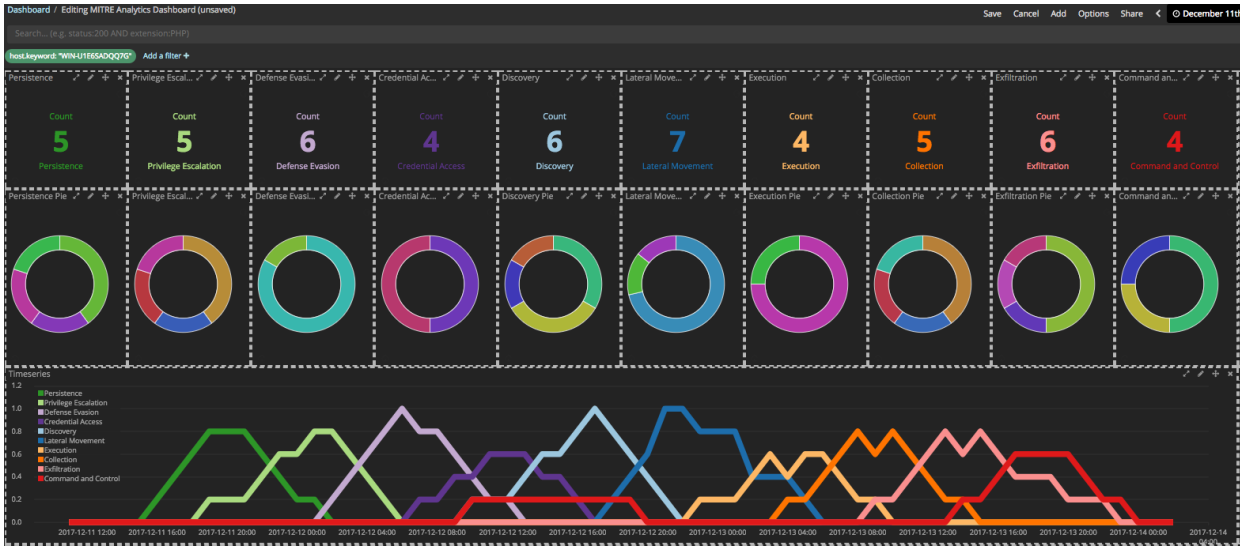


## TEST & RESULTS

For testing we have two Windows hosts where commands have been run to trigger indicators of ‘Tactics’. Following are two scenarios:

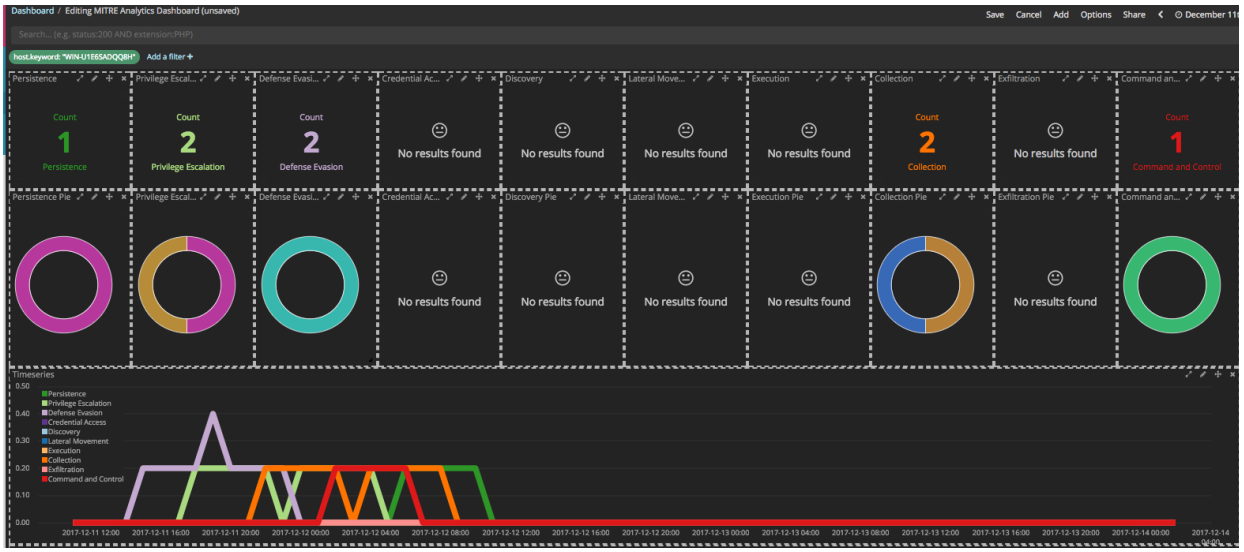
### ADVERSARY PRESENCE LIKELY

Dashboard below filtered with Host1 shows a likely presence of an adversary. Indicators of all the ‘Tactics’ are present and ‘Timeseries’ shows that the events happened in relatively the right sequence.




### ADVERSARY PRESENCE UNLIKELY

Dashboard below filtered with Host2 showing an unlikely presence of an adversary. Not all Indicators of all the ‘Tactics’ are present and ‘Timeseries’ shows that the events did not happen in the right sequence.



## CONCLUSION

In conclusion, Sherlock could be a valuable tool in detecting the presence of APTs in a system where traditional methods of detection might fail. Visualizations are significantly more effective than going through log text. Sherlock analytics can enable even novice Cyber Security Analysts make expert level analysis. In future, I hope to enhance Sherlock analytics with machine learning. The current version could analyze a power user such as a SysAdmin to be an adversary, machine learning will alleviate that error. Sherlock currently include only the publically released version of MITRE CAR Analytics, as a future enhancement I hope to include to enhance it with its private repository as well.

## BIBLIOGRAPHY

[1] "Advanced Persistent Threat (APT)," [Online]. Available: http://searchsecurity.techtarget.com/definition/advanced-persistent-threat-APT.

[2] "Cyber Kill Chain," [Online]. Available: https://www.lockheedmartin.com/us/what-we-do/aerospace-defense/cyber/cyber-kill-chain.html.

[3] "MITRE ATT&CK," [Online]. Available: https://attack.mitre.org/wiki/Main_Page .

[4] A. Matrix. [Online]. Available: https://attack.mitre.org/wiki/ATT%26CK_Matrix.