بِسم الله الرَحمن الرَحيم

Palestine
Polytechnic
University

College of IT and Computer Engineering
Department of Computer System Engineering
Graduation Project

Next-Generation Firewall, Deep Learning Endpoint Protection and Intelligent SIEM Integration

Group members

Sara Iseed

Afnan Madhoun

Ibrahim Abusamrah

Academic Supervisor

Dr. Liana Tamimi

Industry Supervisor

Mr. Abdelraheem Tamimi

Principal Security Consultant

SAFEDENY for Secure Technologies

January 2021

Acknowledgements

First and foremost, we would like to thank Allah for giving us the strength to complete this work. We also wish to express our sincere gratitude to our industry supervisor, Mr. Abdelraheem Tamimi, and academic supervisor, Dr. Liana Tamimi for their effective guidance and comments that greatly improved this work.

Also, we would like to express our gratitude to our family, friends and classmates for the unceasing encouragement, support and attention.

Abstract

In our daily lives, we heavily depend on technology and using the Internet has become an important part of our daily life. This greatly exposes us to cyber-attacks; we need systems and devices to maintain the security and confidentiality of information and data. Among the companies that are interested in the field of information security and data is Sophos, which has many systems and devices that work to maintain data protection and keep from being stolen and attacked.

Moreover, IBM is interested in manufacturing and producing information security products such as IBM QRadar which collects the logs and events from real-time network monitoring, so it can predict the presence of risks or vulnerabilities on the devices and network.

We have integrated the AI of Sophos Next-Generation (NGFW) firewall and Sophos Intercept X Deep Learning with IBM QRadar appliance SIEM solution through collecting and analyzing the data generated from Sophos Central and Next Generation Firewall.

Integrating Sophos Central and Sophos NGFW with IBM QRadar appliance offers a comprehensive insight into the IT infrastructure to collect enough data about the other systems inside the network and this gives the possibility to detect advanced attacks. Furthermore, Increase the performance of network real-time monitoring in IT infrastructure that has a Sophos Next-Generation firewall and Intercept X endpoint.

As for the result of this project, we have developed a framework that integrates Sophos NGFW and Intercept X with IBM QRadar based on the integration methodology that we developed. Moreover, we have augmented Sophos NGFW and Intercept X Deep learning detections into the QRadar AI engine which decreased false-positives and attacks detection time

Table of Contents

List of Figures

List of Tables

Abbreviations

SIEM	Security Information and Event Management
NGFW	Next-Generation Firewall
API	Application Program Interface
REST	Representational state transfer
AI	Artificial Intelligence
IT	Information Technology
SGS	Symantec Gateway Security
LEEF	Log Event Extended Format
UDP	User Datagram Protocol
CEF	Common Event Format
RegEx	Regular Expression
IPS	Intrusion Prevention Systems
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
GUI	Graphical User Interface
QID	QRadar Identifier
MVC	Model-view-controller
SSH	Secure Shell
SQL	Structured Query Language
CRE	Custom Rule Engine
SSL	Secure Sockets Layers
XML	extensible markup language
VPN	Virtual Private Network
LAN	Local Area Network
JWT	JSON Web Token

CHAPTER 1: INTRODUCTION

1.1 Overview

This chapter introduces our project description, key performance indicators, problem statement, and final project results.

1.2 Project description

This project aims to reduce effects of cyber-attacks and threats against companies through live network monitoring within the company organization which makes the detection and response to attacks faster. There are two main parts to this project, the first part is Sophos company that has Sophos Next-Generation firewall which is a system that offers network protection against advanced threats. However, it can detect attacks on the network side only and has no enough data about the other systems inside the network. Sophos Intercept X is an endpoint protection appliance. The second part is IBM company which has IBM QRadar platform designed to automatically identify and analyze threats earlier within analyzed logs and flow data across multiple environments to detect suspicious events in real-time using its artificial intelligent engine.

We have developed a methodology that is summarized as follows, we have sent Sophos NGFW logs and built a SAFEDENY-Orchestrator framework to send Sophos Central data to QRadar. After that, we have normalized the logs and data to be readable for the QRadar system. The second level is AI enhancement to be contained in QRadar rules.

1.3 Project key performance indicators (KPIs)

Our project helps in:

- Increasing the visibility on the network.
- Providing immediate detection to attacks.
- Reducing attacks detection and response time.
- Reducing human capacity.

1.4 Problem statement

1.4.1 Problem analysis

QRadar is an AI system that acts as a central logging system and can correlate the events, logs and assets from many devices (Windows, Linux, Firewalls, Routers, Web servers, DB servers, and others) to detect attacks in real-time using its AI engine. Sophos NGFW can detect attacks on the network side but has no enough data about the other systems inside the network to detect advanced attacks. Sophos Intercept X deep learning is an AI system that can detect the attacks on the PCs and servers but without enough knowledge about the other systems like the network, mobile devices, and other systems where Intercept X is not installed.

So, we have integrated the AI of Sophos NGFW and Intercept X Deep Learning with IBM QRadar SIEM solution.

1.4.2 Purpose

Our main purpose is integrating the AI of Sophos NGFW and Intercept X deep learning with IBM QRadar SIEM solution to get faster and less false-positive detections, and immediate response to network cyber-attacks.

1.4.3 Methodology and requirements

After we made a literature review and studied the options, we developed our methodology to achieve our requirements. The system requirements are summarized as:

- Developing a framework for fetching Sophos Central data to QRadar.
- Fetching Sophos NGFW logs to QRadar.
- Making Sophos Central data and Sophos NGFW logs readable from QRadar.
- Mapping event properties to the correct QRadar data collections.
- Checking the effect of the built-in rule on the Sophos Central data and Sophos NGFW logs to update or create the rules accordingly.

1.4.4 Motivations

Zero-day attacks[1] are increasing every moment and becoming more advanced. Security specialists face challenges in detecting advanced and zero-day attacks or at least detection time is too long which causes more loss.

We are motivated to decrease attack detection time, and false-positives in an IT infrastructure that has Intercept X, Sophos NGFW, and QRadar. Moreover, we are motivated to develop a methodology to integrate other devices in the future.

1.5 Results

We are proud to mention what we succeeded to achieve:

- Integration methodology: we have developed a methodology that can be used to integrate other devices in the future.
- Integration framework: we have developed a framework that integrates Sophos NGFW and Intercept X with IBM QRadar based on the integration methodology mentioned above.
- Decrease attack detection and response time: we have augmented Sophos NGFW and Intercept X Deep learning detections into the QRadar AI engine.

[1] zero-day attack is an attack that exploits a potentially serious software security weakness that the vendor or developer may be unaware of.

CHAPTER 2: BACKGROUND

2.1 Overview

This chapter introduces the theoretical background related to our project, describes the systems that have been used in the project, and discusses the specifications and constraints we have faced in the project.

2.2 Theoretical background

Our project focuses on three systems, Firewall, Endpoint protection, and SIEM. Firewalls can be effective means of protecting a local system or network of systems from network-based security threats while at the same time affording access to the outside world via wide area networks and the Internet [1]. Next-Generation Firewall is a network security device that provides capabilities beyond a traditional, stateful firewall [2] while a traditional firewall typically provides stateful inspection of incoming and outgoing network traffic. It also includes additional features like application control, integrated Intrusion Prevention Systems (IPS), and cloud-delivered threat intelligence [3].

The Endpoints serve as points of access to an enterprise network and create points of entry that can be exploited by malicious actors. So, Endpoint protection software protects these points of entry from the risky activity and malicious attack [4].

Finally, SIEM is software that gives enterprise security professionals both insight into and a track record of the activities within their IT environment. Its combined Security Event Management (SEM) which analyzes log and event data in real-time to provide threat monitoring, event correlation, and incident response with Security Information Management (SIM) which collects, analyzes, and reports on log data [5].

2.3 Literature review

This section reviews some of the previous solutions for the integration of IBM QRadar with firewalls and endpoint protection. Through research and review of several sources related to IBM QRadar integration, we have come across different companies that make integration with IBM QRadar.

2.3.1 Barracuda

IBM QRadar SIEM integrated with Barracuda Spam & Virus Firewall [6]. IBM QRadar accepts both mail Syslog events and web Syslog events from Barracuda Spam & Virus Firewall appliances. Web Syslog that records information on user activity, and configure changes occur on Barracuda Spam & Virus Firewall. Mail Syslog events contain the event and action that is taken when the firewall processes email. Syslog sends logs to IBM QRadar using UDP port 514 to make traffic between two systems.

2.3.2 Fortinet

IBM QRadar SIEM made integration with Fortinet FortiGate Security Gateway that collects events from Fortinet FortiGate Security Gateway and Fortinet FortiAnalyzer products using Syslog. Syslog Redirect protocol to send logs file to IBM QRadar.

2.3.3 PaloAlto

IBM QRadar integrated with PaloAlto endpoint Security Manager (Traps) [7]. IBM QRadar collects the event in Log Event Extended Format (LEEF), and Common Event Format (CEF) from the device. The logs are sent to IBM QRadar in Syslog protocol and it’s automatically discovered.

2.3.4 Symantec

IBM QRadar integrated with Symantec Gateway Security (SGS) Appliance and Symantec Endpoint Protection [8]. IBM QRadar collects events from SGS and Symantec Critical System by using Syslog protocol, this event is automatically discovered.

2.3.5 Cisco

The Cisco Advanced Cisco Experience firewall integrated with IBM QRadar [9] which accepts events that are forwarded from Cisco ACE Firewalls by using Syslog and displays it on the Log Activity tab of QRadar.

After studying the previous integration projects with IBM QRadar appliance, we found that different companies make integration by using Syslog protocol to forward events to QRadar. In our project, we have integrated Sophos NGFW with QRadar by using a Syslog protocol to send logs. Also, we have integrated Sophos Central with QRadar by building an API service (SAFEDENY-Orchestrator) that collects events, alerts and endpoint data.

2.4 Systems characteristics

This section introduces the systems characteristics that we have used in our project which are Sophos Next Generation Firewall, Intercept X Endpoint Protection, and IBM QRadar Security Information and Event Management.

2.4.1 Sophos Next-Generation Firewall

Sophos XG Firewall includes a built-in reporting engine. Furthermore, it has automatic threat response that instantly identifies and isolates compromised systems on a network and stops threats from spreading. Also, it analyzes incoming and outgoing network traffic and detects sophisticated attacks by using protection technologies. These include:

- Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain, learn from large amounts of data [9]. Deep learning added another layer of protection within the sandbox which protects against the latest unseen advanced threats like ransomware, bots, worms, and APTs without using signatures.
- Sandboxing is named Sandstorm in Sophos NGFW which uses NG sandbox technology with integrated deep learning, giving the organization an extra layer of security against ransomware and targeted attacks. It defends against the latest payload-based malware lurking in phishing attacks, spam, and file downloads.
- Intrusion prevention system (IPS) protects systems against vulnerabilities and exploits kits. IPS monitors network traffic as it passes through Sophos NGFW for malicious activity. It logs the activity and attempts to block and prevent the infection and then reports the activity.
- Advanced threat protection monitors global outgoing traffic. It blocks outgoing network traffic attempting to contact command and control servers. This prevents remote access trojans from reporting back to their malicious servers.
- Web protection has dual anti-virus (AV), and SSL inspection. Sophos NGFW protects the network by scanning HTTP and HTTPS traffic for unwanted content or malware.

Sophos NGFW provides overall logging capabilities for traffic on systems and networks. Detailed log information provides a live analysis of network activity, which helps identify security issues and reduce network abuse. Sophos NGFW can store logs locally or forward logs to any external Syslog servers, which are configured in the device.

2.4.2 Intercept X

Intercept X is endpoint protection for computers and servers, which stops the endpoint threats using several modern AI techniques, such as deep learning neural networks that detect unknown malware. Using deep learning allows Intercept X to detect threats without relying on signatures. Moreover, deep learning increases the number of detection and prediction of Zero-Day malware [10]. Moreover, it will report any detections to Sophos Central which is a console for managing Sophos products, so we can see the logs in Sophos Central Dashboard.

There are many log types in Sophos Central and we will investigate in events logs. Events have many categories such as run time detections, malware, and web control.

Sophos provides APIs to get Sophos Central data such as events, alerts and endpoints data which will be used in this project.

2.4.3 IBM QRadar SIEM

IBM QRadar Security Information and Event Management (SIEM) assists security teams to accurately identify and prioritize threats and attacks across the organization and provides intelligent insights that enable teams to respond in a short time to accidents which will reduce its effect [11]. It collects processes, aggregates, and stores network data in real time. It uses that data to manage network security by providing real-time information and monitoring, alerts and offenses, and responses to network threats [12].

The following figure 2.1 [12] shows the QRadar architecture. The three layers below data collection, data processing and data searches are the main functionality for any QRadar.

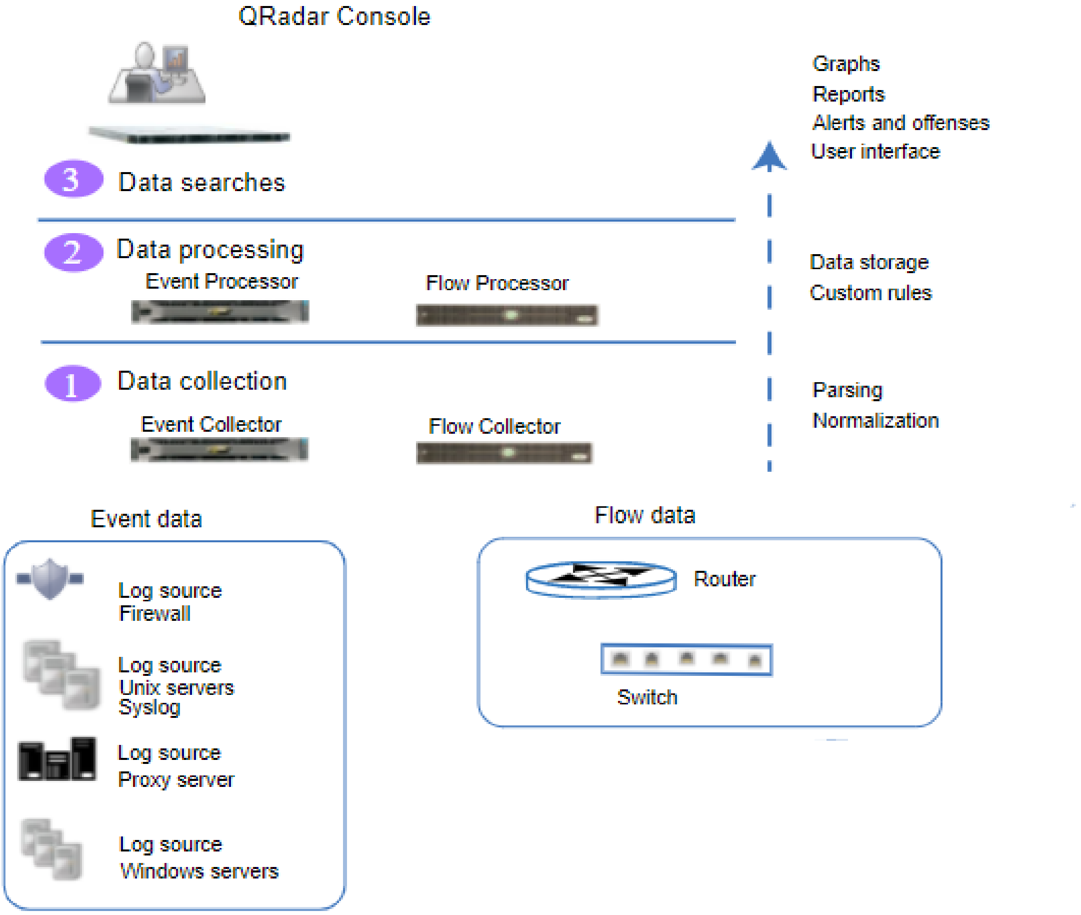


Figure 2.1: QRadar architecture.

Data collection is the first layer, where data is collected from the network. The data collection process can be done directly from the network or use collectors. The data is normalized and normalized to present it in a structured and usable format before it passes to the processing layer. Event data is events that occur in the user's environment such as logins and VPN connections. Flow data is network activity information. The second layer is data processing where event data and flow data are run through the Custom Rules Engine (CRE), which generates offenses and alerts, and then the data is written to storage. The third layer is data searches that contain data which is collected and processed by QRadar and available to users for searches, analysis, reporting, and alerts or offense investigation [12].

2.5 System constraints

This section looks for limitations the project encounters.

2.5.1 Technical constraints

- QRadar is a resource intensive system, it must have 64 GB and eight CPU cores at minimum.
- There are a huge number of logs, so we will handle parts of them because of time limitations.
- There is no existing integration methodology for these systems.

2.5.2 Compliance constraints

- Customer privacy: customers do not want their logs and data to appear because it contains sensitive information.

Chapter 3: Integration Options

3.1 Overview

This chapter introduces the integration options, discusses and determines the options that we have used in our project based on our researches and experiments.

3.2 API

API stands for Application Programming Interface. It is a set of protocols and rules which is a software intermediate layer that allows two products or services to talk with each other without needing to know how they are implemented. API provides a layer of security because when using API, you share just necessary data [13].

Rest/Restful API it stands for Representational State Transfer. It is an architectural style for distributed hypermedia systems. It has a set of rules that developers follow when they create their API.

3.2.1 IBM QRadar API

IBM QRadar appliance provides a RESTful API which can be used by sending HTTP requests. Each request contains authentication information and parameters that modify the request. The action is indicated by the HTTP methods of the request. Most resources format this response as JavaScript Object Notation (JSON). There are many REST API interfaces such as analytics, ariel, asset_modle, reference data and others.

We made an experiment on QRadar API and sent a GET request on reference data API to get IP blacklist reference set and this was the response:

```
{
  "time_to_live": "0 years 0 mons 7 days 0 hours 0 mins 0.00 secs",
  "timeout_type": "LAST_SEEN",
  "number_of_elements": 3,
  "data": [
    {
      "last_seen": 1587847396776,
      "first_seen": 1587847396776,
      "source": "admin",
     "value": "10.100.100.1"
  },
    {
     "last_seen": 1587847410605,
     "first_seen": 1587847410605,
     "source": "admin",
     "value": "**.100.100.90"
    },
    {
     "last_seen": 1587847420635,
     "first_seen": 1587847420635,
     "source": "admin",
     "value": "**.200.200.200"
    }
  ],
  "creation_time": 1389035969294,
  "name": "Asset Reconciliation IPv4 Blacklist",
  "element_type": "IP"
}
```

We have found that the QRadar API can be used to retrieve reference sets and assets data, so we will use it for the management process in future, such as adding or deleting IPs from lists.

3.2.2 Sophos XG Firewall API

Sophos NGFW supports RESTful API calls which allow services or software to access or change system resources using a predefined set of operations. The data is only sent and received from API requests in XML format.

SET Request on XG Firewall through API direct calling using python

```
.
.
username = *******
password = *******
URL = *******
verify = *******
ListName = *******
ip_list = *******
PARAMS = {'reqxml': '<Request><Login><Username>'+ username +'</Username><Password>'+ password
+'</Password></Login>' '<Set
operation="update"><IPHost><Name>'+ListName+'</Name><IPFamily>IPv4</IPFamily><HostType>IPList</Ho
stType>"<ListOfIPAddresses>'+ip_list'</ListOfIPAddresses>' '</IPHost></Set></Request>'}
r =requests.get (url=URL, params=PARAMS, verify=verify)
```

From the above request, we set an IP in the firewall list and this was successfully done. We can make specifications for this IP like blacklist IP, whitelist IP, and other options.

Get Request on XG Firewall through API direct calling using python

```
.
.
username = *******
password = *******
URL = *******
verify = *******
ListName = *******
PARAMS = {'reqxml': '<Request><Login><Username>'+ username +'</Username><Password>'+ password
+'</Password></Login><Get>"<IPHost><Filter>'
'<key name="Name" criteria="=">'+ListName+'</key>"</Filter></IP
Host></Get></Request>'}
r =requests.get (url=URL, params=PARAMS, verify=verify)
```

From this request, we retrieved the list that was found on the firewall. The following code gives an example from the response that we got.

```
<Response APIVersion="1800.1"IPS_CAT_VER="1">
<Login><status>Authentication Successful</status></Login>
<IPHosttransactionid="">
<Name>QRadar-BlockedIPs</Name>
<IPFamily>IPv4</IPFamily>
<HostType>IPList</HostType>
<ListOfIPAddresses>**.9.8.7, **.66.33.22</ListOfIPAddresses>
</IPHost></Response>
```

In this option, it has been found that the use of API for Sophos NGFW does not retrieve logs, alerts, or events. We have used the API for the management process, such as adding or deleting IPs form list.

3.2.3 Sophos Central API

Sophos Central provides a RESTful API and all communication is over HTTPS. APIs allow the retrieval of events, alerts and endpoint data from Sophos Central, for the use in other systems.

Sophos Central has two API versions that can be summarized as:

1. API token: is an old API version which allows the retrieval of events and alerts from Sophos Central. So, it can be used to send endpoints events to QRadar.

   There is a SEIM script developed by Sophos that uses API token to make integration between Sophos Central and SEIM solutions. However, QRadar and Sophos Central are not compatible with each other which means we need to normalize the events then pass them to QRadar one by one which makes it more complex and our aim to make it easier and less complicated.

   We used API token to make a request to get events from Sophos central and this was the response:

```
{
  "appCerts": null,
  "appSha256": null,
  "origin": null,
  "endpoint_type": "computer",
  "endpoint_id": "'165bcfd5-4af5-4934-****************'",
  "customer_id": "********",
  "severity": "low",
  "created_at": "2020-11-13T18:39:44.838Z",
  "source_info": {
    "ip": "192.168.****"
  },
  "threat": null,
  "user_id": "*********",
  "when": "2020-11-13T18:39:44.819Z",
  "core_remedy_items": null,
  "name": "Peripheral allowed: Intel(R) Centrino(R) Advanced-N 6235",
  "location": "********",
  "id": "*********",
  "type": "Event::Endpoint::Device::AlertedOnly",
  "source": "*******\\sara_",
  "group": "PERIPHERALS"
}
```

2. API credential: is a new API version which allows the retrieval of events, alerts, and endpoints data from Sophos Central. So, it can be used to send endpoints events to QRadar.

Furthermore, endpoints data API introduces details about the endpoints data and this will help us in asset models on the QRadar.

We used endpoint API credentials to make a request to get endpoints data from Sophos Central through curl and this was the response:

```
"items":[
{
"id":"********",
"type":"computer",
"tenant":{
    "id":"********"
    },
"hostname":"*******s",
"health":{
    "overall":"bad",
    "threats":{
    "status":"suspicious"
    },
"services":{
    "status":"bad",
    "serviceDetails":[{
    "name":"HitmanPro.Alert service",
    "status":"running"},
    {"name":"Sophos Anti-Virus",
    "status":"running"},
    {"name":"Sophos Anti-Virus Status Reporter",
    "status":"running"},
    {"name":"SophosAutoUpdate Service",
    "status":"running"},
    {"name":"Sophos Clean Service",
    "status":"running"},
    {"name":"Sophos Device Control Service",
    "status":"running"},
    {"name":"Sophos Endpoint Defense",
    "status":"running"},
    {"name":"Sophos Endpoint Defense Service",
    "status":"running"},
    {"name":"Sophos File Scanne
    "status":"running"},
    {"name":"Sophos File Scanner Service",
    "status":"running"},
    {"name":"Sophos MCS Agent",
    "status":"running"},
    {"name":"Sophos MCS Client",
    "status":"running"},
    {"name":"Sophos Network Threat Protection",
    "status":"stopped"},
    {"name":"SophosSafestore Service",
    "status":"running"},
    {"name":"Sophos System Protection Service",
    "status":"running"},
    {"name":"Sophos Web Control Service",
    "status":"running"},
    {"name":"Sophos Web Intelligence Filter Service",
    "status":"running"},
    {"name":"Sophos Web Intelligence Service",
    "status":"running"}]}},
"os":{"isServer":false,
"platform":"windows",
"name":"Windows 10 Pro",
"majorVersion":10,
"minorVersion":0,
"build":18362},
"ipv4Addresses":[
    "*********",
    "*********"],
"ipv6Addresses":[
    "*********",
    "*********"],
"macAddresses":[
    "*********",
    "*********",
    "*********"],

"associatedPerson":{
"name":"*********",
"viaLogin":"**********"},
"tamperProtectionEnabled":true,
"assignedProducts":[{
"code":"endpointProtection","version":"10.8.5","status":"installed"},
{"code":"deviceEncryption","version":"2.0.70","status":"notInstalled"},
{"code":"interceptX","version":"2.0.16","status":"installed"},
{"code":"coreAgent","version":"2.5.2","status":"installed"}],
"lastSeenAt":"2019-12-14T11:26:09.673Z"},
```

Furthermore, we made a request on alerts API and got alerts from Sophos Central and this was the response:

```
{
  "id":"*******",
  "allowedActions":[
    "authPua",
    "cleanPua",
    "clearThreat"
  ],
  "category":"pua",
  "description":"PUA detected: 'Generic PUA IE' at 'C:\\Program*******'",
  "groupKey":"********",
  "managedAgent":{
    "id":"********",
    "type":"computer"
  },
  "person":{
    "id":"********"
  },
  "product":"endpoint",
  "raisedAt":"2020-11-04T10:10:19.272Z",
  "severity":"medium",
  "tenant":{
    "id":"********",
    "name":"SOC Lab"
  },
  "type":"Event::Endpoint::Threat::PuaDetected"
}
```

We have used Node.js programming language to fetch events, alerts and endpoints data form Sophos Central APIs.

The following table 3.1 summarizes all systems APIs:

Table 3.1: Comparison between Systems API.

	Sophos central		XG firewall	IBM QRadar
API name	Token API	Credential API	XG firewall API	QRadar API
Access to	Alerts and events	Alerts and endpoints	Access or change system resources	Everything

3.3 Syslog protocol

A protocol that provides a transport to allow a device to send event notification messages across IP networks to event message collectors, also known as Syslog servers. Syslog uses the User Datagram Protocol (UDP) for communication. Sophos NGFW supports Syslog protocol [14].

The only way to get logs from Sophos NGFW and send it to QRadar is Syslog protocol. We have configured Syslog on Sophos NGFW with specific instructions which include Syslog parameters.

3.4 SAFEDENY-Orchestrator framework

This system is an integration framework between QRadar and Sophos Central. It fetches data from Sophos Central API and sends them to QRadar.

3.5 IBM QRadar app editor

The IBM QRadar GUI Application Framework is used to develop new application modules that integrate with QRadar and provide new capabilities. Each app has its own allocated memory and allocated amount of CPU resources. The main web language that is used in its development is Python. Moreover, we can use the Flask framework to develop the application. Every app runs on its own unique server; each server runs within a secure Linux container.

We have used this editor in building an application that connects with the SAFEDENY-Orchestrator framework.

3.6 QRadar rules

Rules in QRadar have some conditions which are used to test the actions to search and detect suspicious activities in the network. Once all conditions are met, the rule will apply and generate responses. Rules categories are classified into high-level categories and low-level categories. The high-level categories are used to group incoming events to be processed by QRadar. Furthermore, every high-level category contains several low-level categories.

The anomaly detection rules test the results of saved flow or events searches to detect when unusual traffic patterns occur in the network. As we mentioned before when the rule applies, the responses will generate. There are a lot of rule response types, we will present the most important types:

- Dispatch New Event.
- Notify.
- Reference Set.
- Custom Actions.

We have used the rules to test the integration through different attack scenarios.

3.6.1 Reference set

The reference set is a collection of unique values and it is derived from events or flows on a network. It is used to compare a property value against a list, such as IP addresses or usernames.

We have used it in rule test conditions, and rule responses. We have put admins usernames in a reference set and used it in rule test condition. The following figure 3.1 shows an example of a reference set as a rule response.

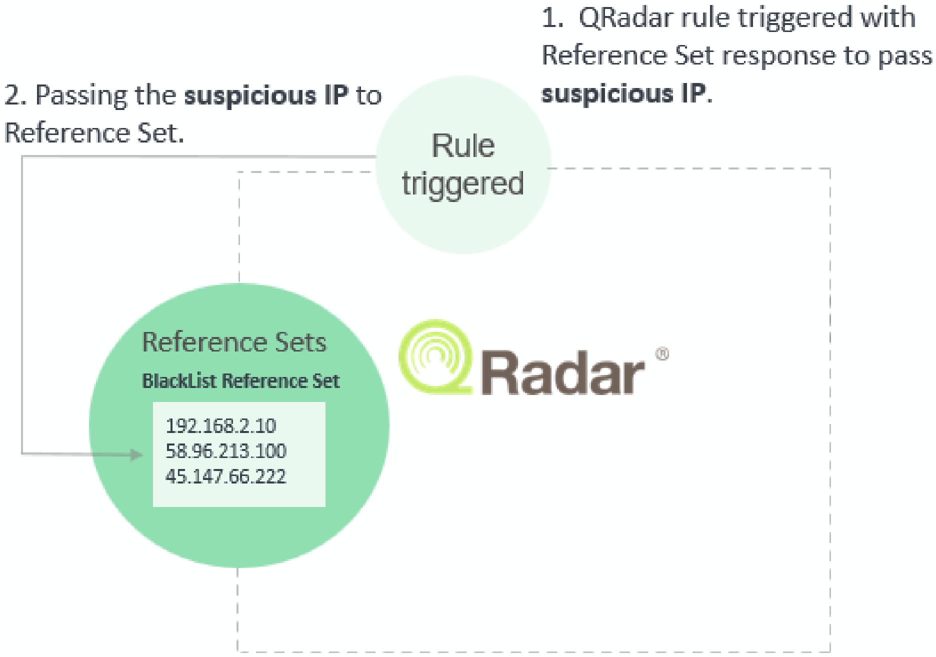

1. QRadar rule triggered with Reference Set response to pass **suspicious IP**.
2. Passing the **suspicious IP** to Reference Set.
Rule triggered
Reference Sets
BlackList Reference Set
192.168.2.10
58.96.213.100
45.147.66.222
QRadar ®

Figure 3.1: Reference set as a rule response diagram.

3.6.2 Custom actions

In QRadar we can attach a script to rules to do specific actions as rule response by using custom action technology. So custom action using to define a parameter and passing it to the script, by using custom action can pass parameters outside QRadar such as server, Sophos XG firewall, and databases.

There are two types of custom action parameters, fixed property is predefined value and network event property is a dynamic value that is generated by events.

So, we have used this option to pass parameters to the Sophos NGFW to do some actions on the firewall such as passing suspect IP to block it on the firewall.

The following figure 3.2 shows examples of how we can use this option for management.

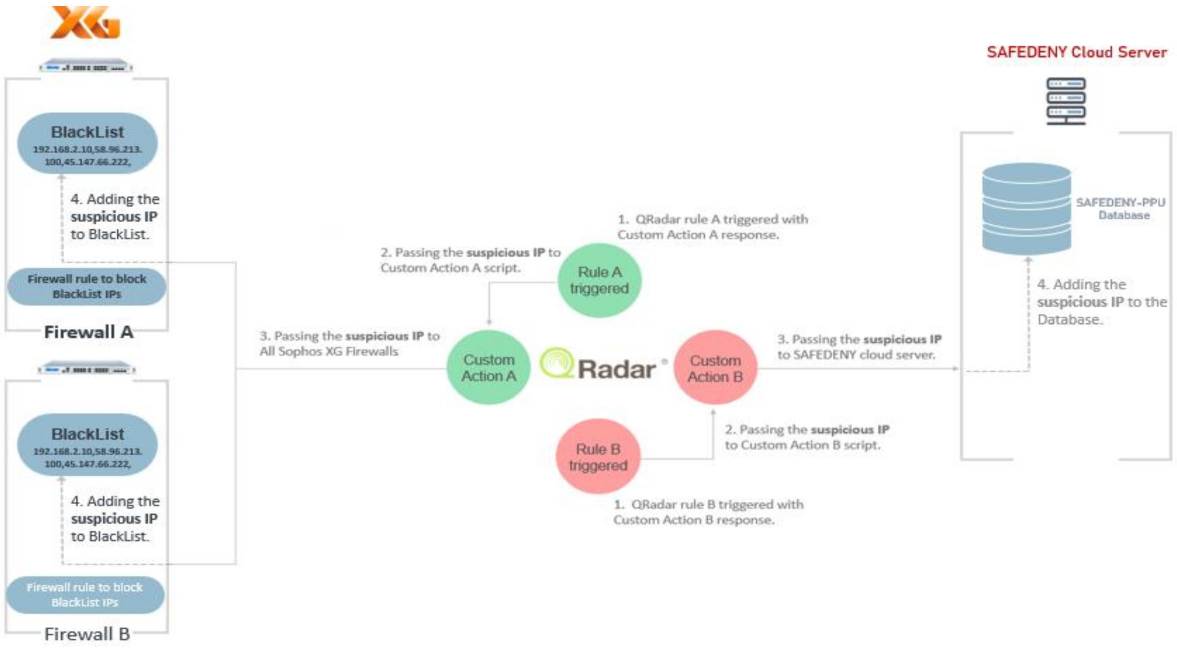

XG
BlackList
192.168.2.10,58.96.213.100,45.147.66.222,
4. Adding the suspicious IP to BlackList.
Firewall rule to block BlackList IPs
Firewall A
BlackList
192.168.2.10,58.96.213.100,45.147.66.222,
4. Adding the suspicious IP to BlackList.
Firewall rule to block BlackList IPs
Firewall B
3. Passing the suspicious IP to All Sophos XG Firewalls
2. Passing the suspicious IP to Custom Action A script.
1. QRadar rule A triggered with Custom Action A response.
Rule A triggered
Custom Action A
QRadar
Custom Action B
Rule B triggered
2. Passing the suspicious IP to Custom Action B script.
1. QRadar rule B triggered with Custom Action B response.
3. Passing the suspicious IP to SAFEDENY cloud server.
SAFEDENY Cloud Server
SAFEDENY-PPU Database
4. Adding the suspicious IP to the Database.

Figure 3.2: QRadar Custom Actions diagram for passing suspicious IP to Sophos NGFW and database.

3.2.3 Discussion

The following table 3.2 shows what we have used to make the integration framework for every system in our project.

Table 3.2: Integration framework.

Service Name	Integration Options	Will use or NOT	Usage
Sophos Central	Siem.py script	No	
	API Token	Yes	Get events.
	API Credential	Yes	Get alerts and endpoints.
Sophos XG Firewall	Forward Syslog	Yes	Get events.
	XML API	Yes	Get commands from QRadar for firewall management.
QRadar	API	No	
	QRadar Apps	Yes	Take events and data from Sophos Central API and Sophos XG Firewall.
	Custom Actions	Yes	Sends commands to Sophos XG Firewall for management.

The following diagram 3.3 shows the integration framework for our project. For Sophos Central, we have built SAFEDENY-Orchestrator service contain two parts. The first part is fetching alerts, events, and endpoints data from Sophos Central. The second part is the API service to send Sophos Central data to QRadar through SAFEDENY-Orchestrator app.

For Sophos NGFW, we have used the Syslog protocol to send Sophos NGFW logs to QRadar. Moreover, there is two-way integration which means we also send data from QRadar to Sophos NGFW for management through script execution (Custom actions).

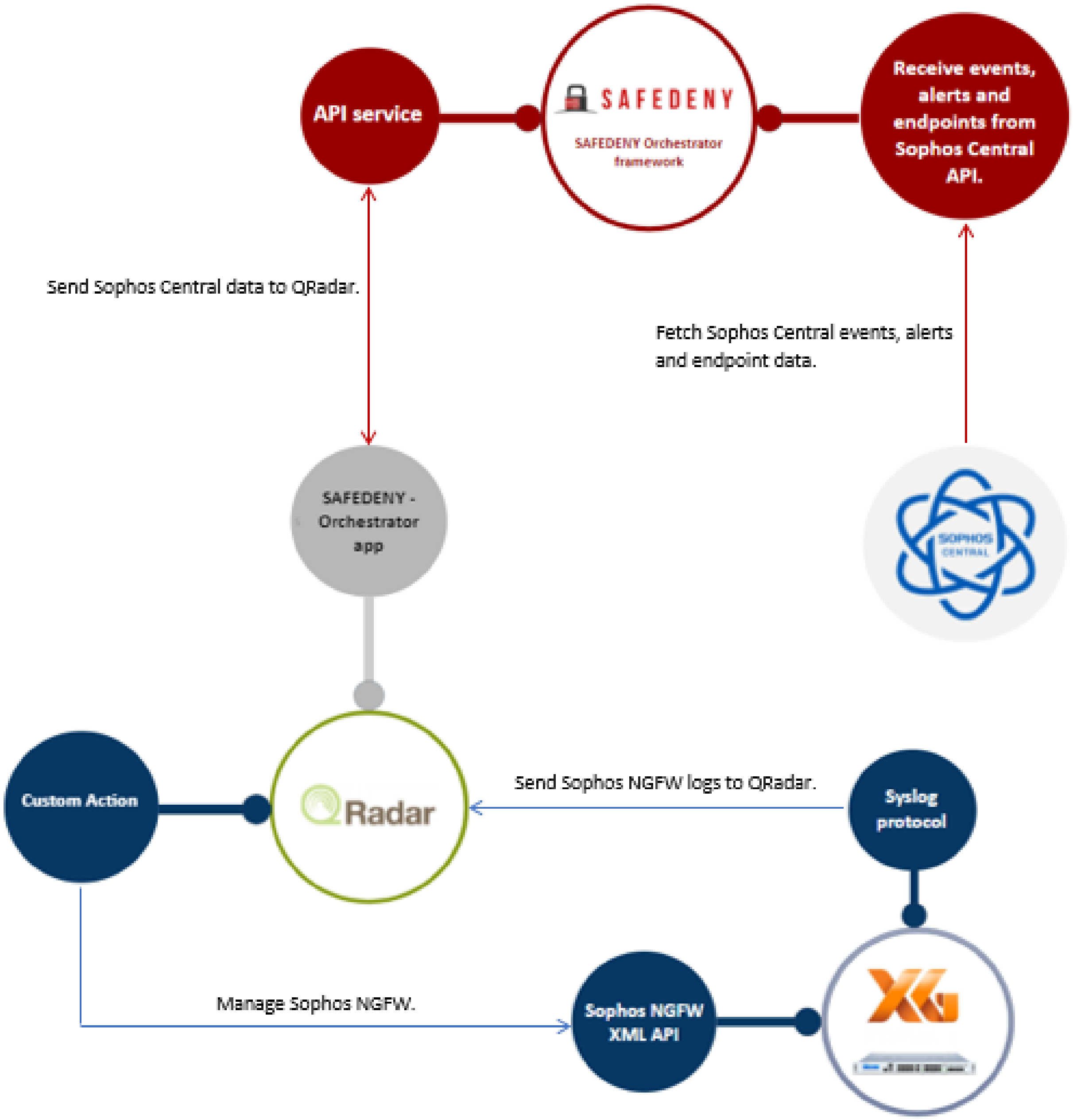

API service
SAFEDENY
SAFEDENY Orchestrator framework
Receive events, alerts and endpoints from Sophos Central API.
Send Sophos Central data to QRadar.
Fetch Sophos Central events, alerts and endpoint data.
SAFEDENY - Orchestrator app
SOPHOS CENTRAL
Custom Action
Radar
Send Sophos NGFW logs to QRadar.
Syslog protocol
Manage Sophos NGFW.
Sophos NGFW XML API
XG

Figure 3.3: Integration framework diagram.

Chapter 4: Implementation

4.1 Overview

This chapter introduces the implementation of integration for Sophos Central and Sophos NGFW with QRadar.

4.2 Sophos Central integration with QRadar

This section introduces the implementation of integration for Sophos Central with QRadar.

4.2.1 SAFEDENY-Orchestrator development

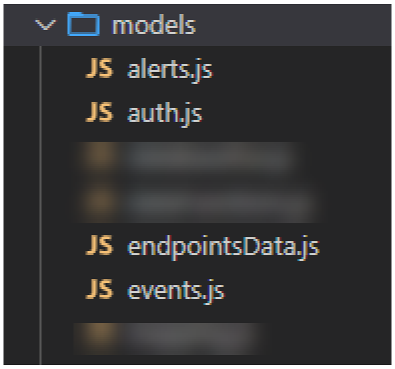
We have used the NodeJS and express environment that runs code written in JavaScript to develop the service. Furthermore, we have used an MVC architecture pattern in development.

4.2.1.1 Fetch data from Sophos Central API

We have developed the code that fetches events, alerts, and endpoints data from Sophos Central API. The code runs in the background on the cloud Linux server and services two types of Sophos customers which are the service provider (Partner Customer), and the customer who does not have a service provider (Tenant Customer). The code fetches the data and stores it in the database.

To fetch data from Sophos API we need to get Sophos credential so we have developed a web service with basic functions to get the credential from customers. The web service interfaces are in Appendix A.

The following shot 4.1 shows the four main files in our code that contain fetch functions.

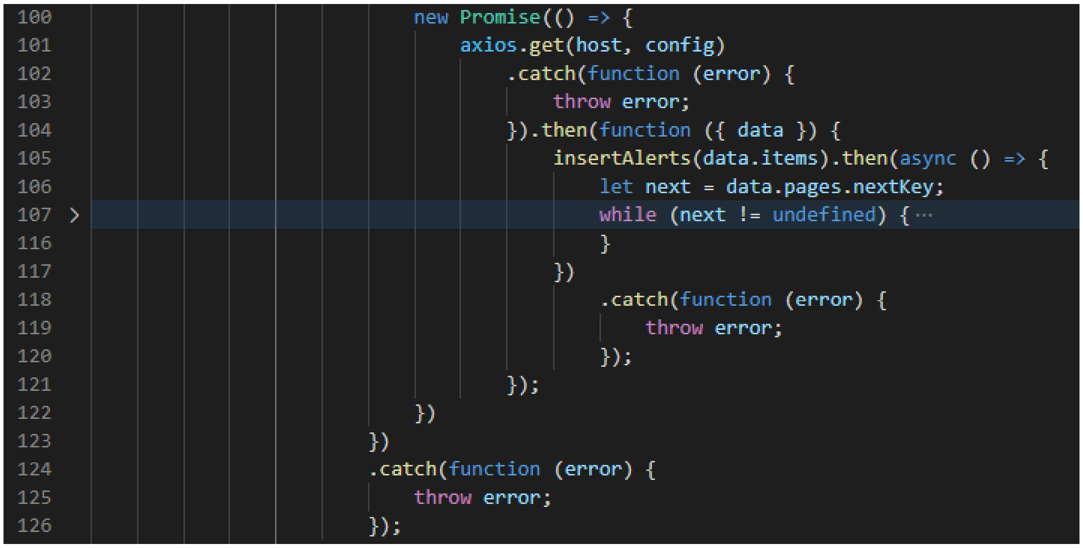


```
models
    JS alerts.js
    JS auth.js
    [redacted]
    [redacted]
    JS endpointsData.js
    JS events.js
    [redacted]
```

Figure 4.1: The models files for fetch Sophos Central data.

auth.js file: it contains the code that connects with Sophos authentication API to validate the customer credential data and get a token to use it for fetch events, alerts, and endpoints data.

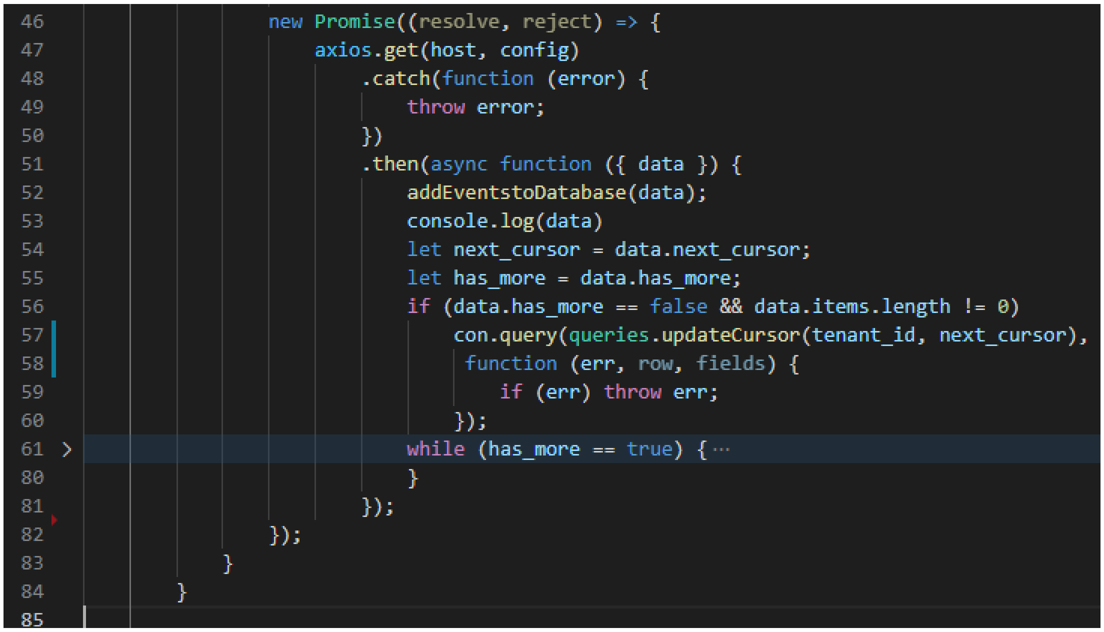
alerts.js file: it contains the code that connects with Sophos alerts API to fetch customers alerts and store it in the database. The following shot 4.2 shows a part of the fetch alerts function.



```
    new Promise(() => {
        axios.get(host, config)
            .catch(function (error) {
                throw error;
            }).then(function ({ data }) {
                insertAlerts(data.items).then(async () => {
                    let next = data.pages.nextKey;
                    while (next != undefined) {···
                    }
                })
                    .catch(function (error) {
                        throw error;
                    });
            });
    })
})
.catch(function (error) {
    throw error;
});
```

Figure 4.2: Part of the fetch alerts function.

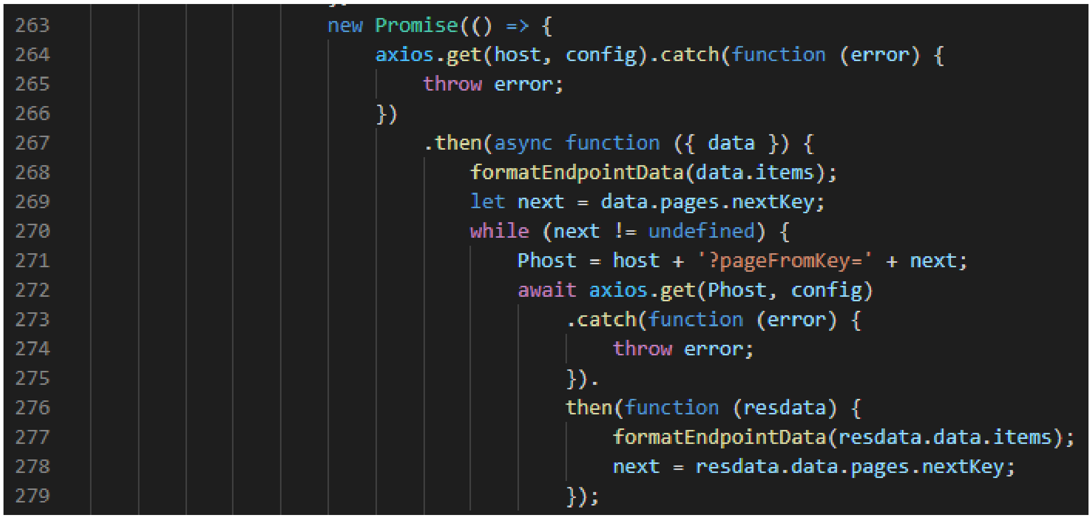
events.js file: it contains the code that connects with Sophos events API to fetch customers events and store it in the database. The following shot 4.3 shows a part of the fetch events function.



```
        new Promise((resolve, reject) => {
            axios.get(host, config)
                .catch(function (error) {
                    throw error;
                })
                .then(async function ({ data }) {
                    addEventstoDatabase(data);
                    console.log(data)
                    let next_cursor = data.next_cursor;
                    let has_more = data.has_more;
                    if (data.has_more == false && data.items.length != 0)
                        con.query(queries.updateCursor(tenant_id, next_cursor),
                         function (err, row, fields) {
                            if (err) throw err;
                        });
                    while (has_more == true) {···
                    }
                });
        });
    }
}
```

Figure 4.3: Part of fetch events function.

endpointsData.js file: it contains the code that connects with Sophos endpoints API to fetch customers endpoints data and store it in the database. The following shot 4.4 shows a part of the fetch endpoints data function.

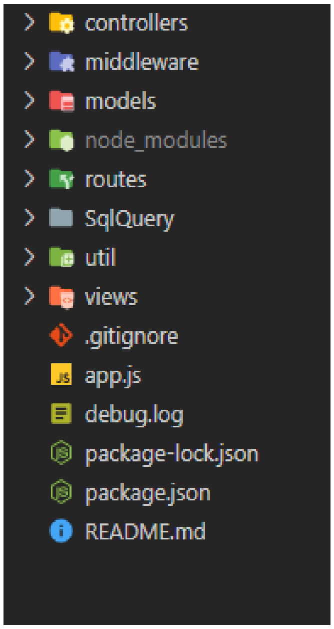


```
new Promise(() => {
    axios.get(host, config).catch(function (error) {
        throw error;
    })
        .then(async function ({ data }) {
            formatEndpointData(data.items);
            let next = data.pages.nextKey;
            while (next != undefined) {
                Phost = host + '?pageFromKey=' + next;
                await axios.get(Phost, config)
                    .catch(function (error) {
                        throw error;
                    }).
                    then(function (resdata) {
                        formatEndpointData(resdata.data.items);
                        next = resdata.data.pages.nextKey;
                    });
```

Figure 4.4: Part of fetch endpoints data function.

4.2.1.2 API service development

To access the Orchestrator API, an API key is needed so developed the API Key generator using JSON Web Token (JWT). Every customer has his own API Key. The following shot 4.5 shows the Node JS files for the APIs.



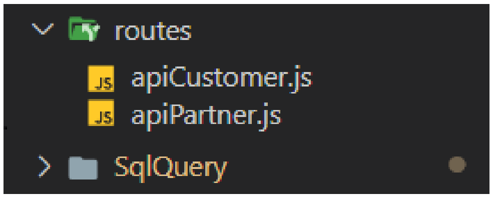
```
> controllers
> middleware
> models
> node_modules
> routes
> SqlQuery
> util
> views
  .gitignore
  app.js
  debug.log
  package-lock.json
  package.json
  README.md
```

Figure 4.5: Node JS Files for API service.

In the following we illustrate the APIs files:

app.js file: is the listener that receives the requests. In this file we have verified the API Key. If it expired or not correct then the server will deny the access. On the other hand, the listener checks which the correct route will convert the request to.

There are two types of routers which are apiCustomer, and apiPartner routes. The following shot 4.6 shows the routes folder.



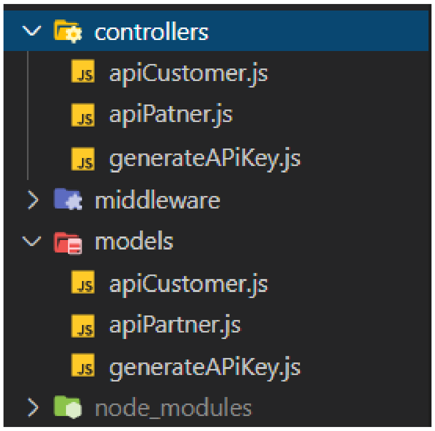
```
v routes
    apiCustomer.js
    apiPartner.js
> SqlQuery
```

Figure 4.6: API routes folder.

apiCustomers.js file: contains a set of APIs requests that customers can request information about their endpoints.

apiPartners.js file: We have built this folder for future works which we will use to give the partner more control over the data of their customers.

The following shot 4.7 shows the models folder in the API service.



controllers
apiCustomer.js
apiPatner.js
generateAPiKey.js
middleware
models
apiCustomer.js
apiPartner.js
generateAPiKey.js
node_modules

Figure 4.7: Models folder in the API service.

generateAPiKey file: generates the API key for the customer that passes the id to it.

database.js file: contains the connection data of the MySQL database.

The following table 4.1 summarize the most used API services in the SAFEDENY-Orchestrator.

Table 4.1: Most used APIs in the SAFEDENY-Orchestrator.

Number	API Details				
1	Method	URL			
	GET	/api/customer/v1.0/allEvents			
	Authentication	Key	api key	Value	<value>
	Param	-	-	-	-

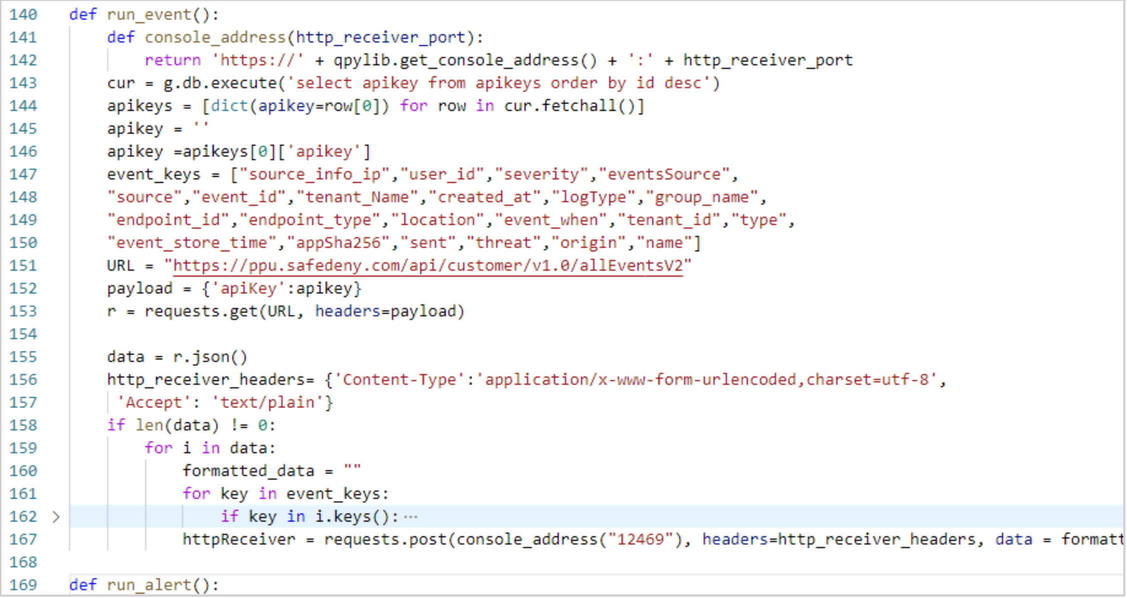
	Description				
	It returns all events for all endpoints for a specific customer from the database.				
2	Method	URL			
	GET	/api/customer/v1.0/allAlertsV2			
	Authentication	Key	api key	Value	<value>
	Param	-	-	-	-
	Description				
	It returns all alerts for all endpoints for a specific customer which are not sent to the Qradar app from the database.				
3	Method	URL			
	GET	/api/customer/v1.0/authentication			
	Authentication	Key	api key	Value	<value>
	Param	-	-	-	-
	Description				
	It returns the validate API key.				

The rest of the API that we have implemented is in Appendix B.

4.2.1.3 QRadar-SAFEDENY-Orchestrator plugin development

We have built an application in the QRadar to fetch Sophos Central events and alerts from the API service that we have built. We have used the QRadar app editor to develop the application using a flask framework and used the SQLit database in the QRadar to store the API key that was added by the customer. Also, the application formats the logs as HTTP data and sends it to Sophos Central log source (HTTP receiver).

The following shot 4.8 shows part of the application code that fetch events from the service and send it to the QRadar. Moreover, the SAFEDENY-Orchestrator app interfaces are in Appendix A.

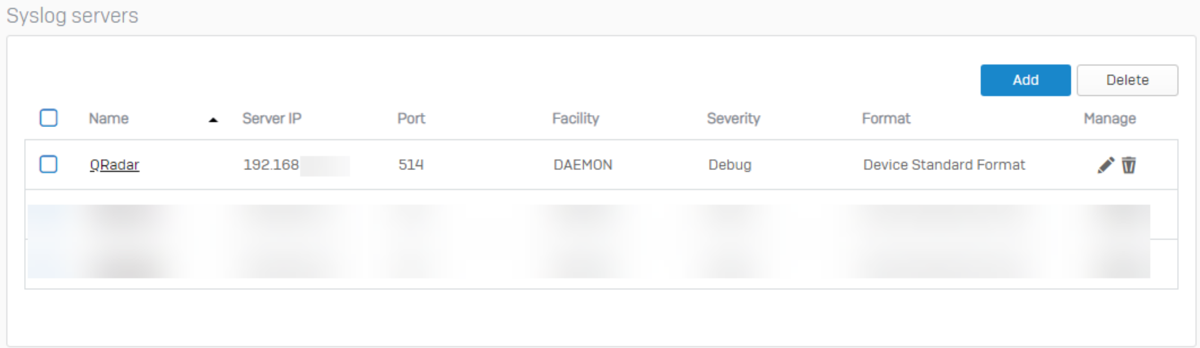


```python
def run_event():
    def console_address(http_receiver_port):
        return 'https://' + qpylib.get_console_address() + ':' + http_receiver_port
    cur = g.db.execute('select apikey from apikeys order by id desc')
    apikeys = [dict(apikey=row[0]) for row in cur.fetchall()]
    apikey = ''
    apikey =apikeys[0]['apikey']
    event_keys = ["source_info_ip","user_id","severity","eventsSource",
    "source","event_id","tenant_Name","created_at","logType","group_name",
    "endpoint_id","endpoint_type","location","event_when","tenant_id","type",
    "event_store_time","appSha256","sent","threat","origin","name"]
    URL = "https://ppu.safedeny.com/api/customer/v1.0/allEventsV2"
    payload = {'apiKey':apikey}
    r = requests.get(URL, headers=payload)

    data = r.json()
    http_receiver_headers= {'Content-Type':'application/x-www-form-urlencoded,charset=utf-8',
     'Accept': 'text/plain'}
    if len(data) != 0:
        for i in data:
            formatted_data = ""
            for key in event_keys:
                if key in i.keys():···
            httpReceiver = requests.post(console_address("12469"), headers=http_receiver_headers, data = formatt

def run_alert():
```

Figure 4.8: Fetch event function in the SAFEDENY-Orchestrator app.

4.2.2 Define a Syslog server for Sophos NGFW

We have created a Syslog server which has the Syslog port and the IP for QRadar. The following shot 4.9 shows the Syslog server that we created.



Syslog servers

□	Name ▲	Server IP	Port	Facility	Severity	Format	Manage
□	QRadar	192.168 [redacted]	514	DAEMON	Debug	Device Standard Format	✎ 🗑

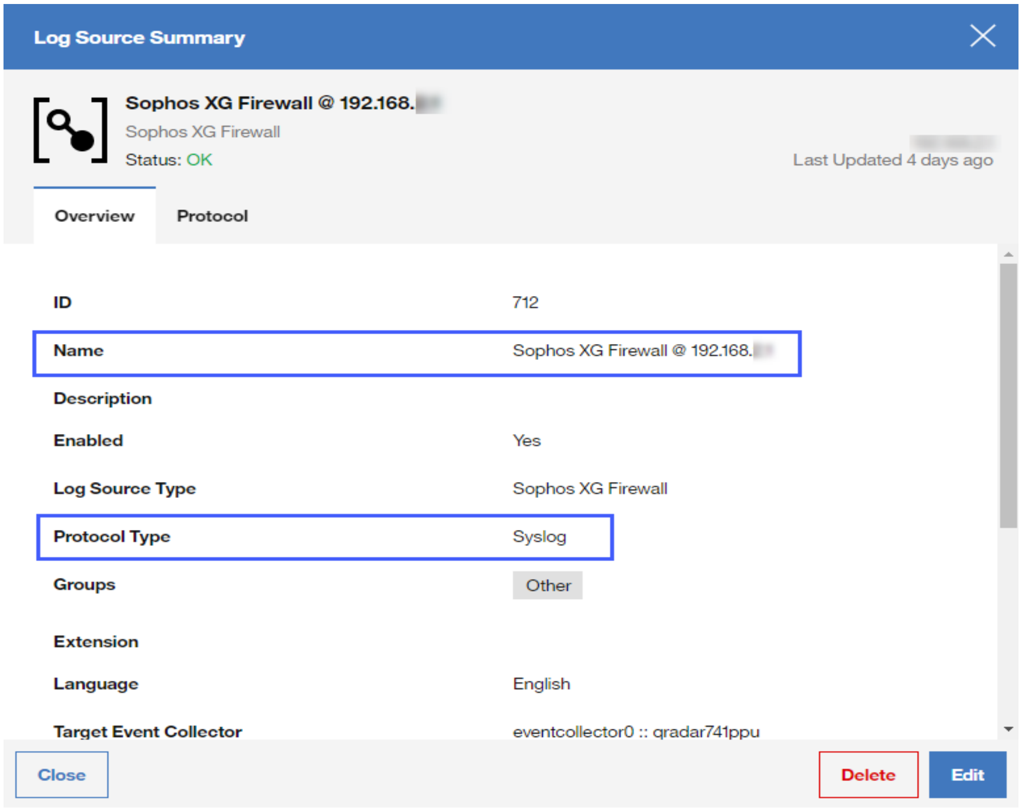
[redacted]

Figure 4.9: Syslog server on Sophos NGFW.

We have chosen the log types that we want to send through Syslog protocol such as Firewall, IPS, Content filtering, Events, Web server protection, and Advanced threat protection.

4.2.2.1 Define a log source for Sophos NGFW logs on QRadar

We have used the DSM editor and created a log source type named (Sophos XG Firewall) to associate the incoming firewall logs and extra content with just that log source type. Then we have created the Sophos firewall log source as a Syslog server in the QRadar to receive the logs from it. The following shots 4.10 show the setting of the log source which we have created for Sophos NGFW logs.



Log Source Summary

Sophos XG Firewall @ 192.168.[redacted]
Sophos XG Firewall
Status: OK

[redacted]
Last Updated 4 days ago

ID	712
Name	Sophos XG Firewall @ 192.168.[redacted]
Description	
Enabled	Yes
Log Source Type	Sophos XG Firewall
Protocol Type	Syslog
Groups	Other
Extension	
Language	English
Target Event Collector	eventcollector0 :: qradar741ppu

Figure 4.10: Sophos NGFW log source setting.

4.3 Sophos Central and Sophos NGFW data normalization

All received data was in a different format from QRadar data format. We have normalized Sophos Central alerts, events, and Sophos NGFW logs properties using RegEx expression and data manipulation. Furthermore, we have added some important additional properties and normalized it such as threat name, and web category. The number of properties that we have normalized above fifty properties for both Sophos Central and Sophos NGFW.

The following table 4.2 shows some of the Sophos Central and Sophos NGFW properties and their expression which we have normalized.

Table 4.2: Sophos Central and Sophos NGFW normalized properties.

	Properties	Expression type	Expression	Delimiter
Sophos Central	Event Category	Regex	&eventsSource=([^"]+):([^"]+)&source([^"]+)&type=([^"]+)::([^"]+)::([^"]+)::([^"]+)&event_store_time=	$1 - $2 - $4 - $5 - $6 - $7
			&eventsSource=([^"]+):([^"]+)&source([^"]+)&type=([^"]+)::([^"]+)::([^"]+)&event_store_time=	$1 - $2 - $4 - $5 - $6
			&eventsSource=([^"]+):([^"]+)&tenant_Name([^"]+)&type=([^"]+)::([^"]+)::([^"]+)::([^"]+)::([^"]+)&endpoint_type	$1 - $2 - $4 - $5 - $6 - $7 - $8
	Event ID	Regex	&type=([^"]+)::Firewall::([^"]+)&event_store	$2
			&group_name=DATA_LOSS_PREVENTION&([^"]+)&type=([^"]+)::([^"]+)::([^"]+)&event_store	$4
			&type=([^"]+)::([^"]+)::([^"]+)::([^"&]+)&event_store	$3 - $4
			&name=([^"]+):	$1
			^<.*\spriority=.*user_name="([^"]+)"\ssrc_ip=.*	

Sophos NGFW	Username	Regex		$1
			^<.*user_name="([^"]+)"\susergroupname.*	
			^<.*user_name="([^"]+)"\suser_gp.*	
	Event Category	Regex	log_type="([^"]+)"\slog_component="([^"]+)" log_subtype="([^"]+)" status="([^"]+)"	$1 - $2 - $3 - $4
			log_type="([^"]+)"\slog_component="([^"]+)" log_subtype="([^"]+)"	$1 - $2 - $3 - NA

Other properties of Sophos Central and Sophos NGFW and their expressions are in appendix C.

4.4 Sophos Central and Sophos NGFW data AI enhancement

For data AI normalization we have done the following steps:

1. We have analyzed events at both Sophos Central and Sophos NGFW. Through analysis, we have determined a lot of things such as why events occurred, events origin, and search for any related other events. It helped us in the AI normalization.
   Here is an example, a malware detection event in Sophos Central. This event is generated when the endpoint detects malware in the endpoint based on signature or behavior also this event is generated by Sophos Anti-virus component. We found that it has related events which are cleaned up successfully when the system can delete it, and failed to clean up when it needed manual clean.
2. Mapping event properties to the correct QRadar data collections
3. We have checked the built-in rules effect on this event and update the rules accordingly.
4. We have created the new rules when needed.

The number of event types that we have mapped above 180 types for both Sophos Central and Sophos NGFW. The following table 4.3 shows some of the Sophos Central and Sophos NGFW properties and their expression which we have normalized.

Table 4.3: Sophos Central and Sophos NGFW mapped events.

	Event Category	QID Name	Low Level Category	QID Severity
Sophos Central	SAFEDENY Orc. - Sophos Central - Event - Endpoint - CoreDetection	Malicious File Detected	Malicious Software	6
	SAFEDENY Orc. - Sophos Central - Alert - Event - Endpoint - Threat - CleanupFailed	Manual cleanup required	Remove Failed	8
	SAFEDENY Orc. - Sophos Central - Event - Endpoint – CoreClean	Malware detected and successfully deleted	Remove Successful	3
Sophos NGFW	Firewall - Appliance Access - Denied - Deny	Firewall Deny	Firewall Deny	4
	Firewall - SSL VPN - Denied - Deny	VPN Login Failed	Remote Access Login Failed	3

Other mapped events for Sophos Central and Sophos NGFW are in appendix C.

4.5 Custom action implementation

We have implemented custom action to pass suspicious IP to specific lists in Sophos NGFW. The following code shows a python custom action script that was added as a response in QRadar rule.

Passing parameter to Sophos NGFW script:

```
.
.
ipsource = sys.argv[1];
URL = sys.argv[2];
username = sys.argv[3];
password = sys.argv[4];
ListName = sys.argv[5];
PARAMS = {‘reqxml': '<Request><Login><Username>'+ username
+'</Username><Password>'+ password +'</Password></Login>' '<Set
operation="update"><IPHost><Name>'+ ListName
+'</Name><IPFamily>IPv4</IPFamily><HostType>IPList</HostType>'
'<ListOfIPAddresses>' + ip_list + '</ListOfIPAddresses>'
'</IPHost></Set></Request> ‘}
r = requests.get (url=URL, params=PARAMS, verify=verify)
```

4.6 Rules implementation

We have created some policies in Sophos Central and rules in Sophos NGFW that have helped us in the rules implementation such as Sophos Central data loss prevention policy and blocked black list IPs firewall rule. We have implemented rules in QRadar for testing issues and used this rule in our scenarios. In the following, we introduce a pseudo-code for some of these rules.

1. **Rule name: Login to VPN**

```
IF (
        High level category is Authentication,
        AND low-level category is User Login Success,
        AND the event payload contains auth_client="IPSec",
        AND username is in (VPN Users) reference set
   )
        THEN add the source IP of the event in (Monitoring VPN Users) reference set;
ENDIF
```

2. **Rule name: Suspicious Login to Unauthorized Account**

```
IF (
        NOT source IP is related to any username in (IP-Username Matched) reference map,
        AND high-level category is Access,
        AND low-level category is Session Opened,
        AND the event(s) detected by (WinServer @ SAFEDENY-Server) log source,
        AND source IP are contained is in (Monitoring Current VPN Users) reference set
        )
        THEN create new event named "Suspicious Login to Unauthorized Account via authorized VPN User",
            AND add source IP and username to the (Suspicious Login to
            Unauthorized Account) reference map of sets,
            AND execute custom action to disable VPN user on Sophos NGFW,
            AND Add annotate offense named "Suspicious Login to Unauthorized
        Account via Authorized VPN User";
    ENDIF
```

3. **Rule name: Malware Malicious Software**

```
IF (
        High level category is Malware,
        AND low-level category is Malicious Software,
        AND the event detected from (Sophos Central @ orchestrator) log source;
  )
ENDIF
```

4. **Rule name: Category Definition: Superuser Accounts**

```
IF (
         The event username matches the following admin, superuser, root, toor,
        init, Admin, Administrator, ADMINISTRATOR, ADMIN, ROOT, SYS,
        SYSTEM
  )
ENDIF
```

5. **Rule name: Category Definition: Sensitive data**

```
IF (
        the Event Payload contains .xlsx
   )
ENDIF
```

6. **Rule name: Suspicious Malware Detected after Unauthorized User Login**

```
IF (
        "Malware Malicious Software" rule occurred at least one time in 50 minutes
        after "Suspicious Login to Unauthorized Account" rule occurred with same
        destination IP
   )
        THEN Create new event named "Suspicious Login to Unauthorized Account via
        Authorized VPN User",
                AND Add annotate Offense named "Suspicious Malware Detected
                after Unauthorized User Login";
ENDIF
```

7. **Rule name: Exploit: Administrator Social Engineering Account Added**

```
IF (
        Event matches with "Superuser Accounts",
        AND high-level category is Authentication,
        AND low-level category is User Account Added
   )
```

```
        THEN create new event named “Administrator creates an account”,
        AND add annotate Offense named “Administrator creates an account”,
        AND add created username to (New Created User) reference set;
ENDIF
```

8. Rule name: Exploit: New User Accesses Sensitive Data

```
IF (
        Username is in (New Created User) reference set,
        AND high-level category is Risk,
        AND low-level category is Data Loss Possible,
        AND event matches with Sensitive data,
        AND the event(s) detected by (Sophos Central @ orchestrator) log source;
    )
        THEN create new event named “A recently created account used to access
                sensitive Data”,
                AND add annotate “A recently created account used to access sensitive
        data”;
ENDIF
```

9. Rule name: Exploit: Administrator Social Engineering Account Deleted

```
IF (
        Event matches with “Superuser Accounts”,
        AND high-level category is Authentication,
        AND low-level category is User Account Removed,
        AND Target User Name are contained is in (New Created User) reference set
    )
        THEN create new event named “Administrator creates an account”,
            AND add username to the (Administrator delete recent user) reference sets,
            AND Add annotate offense named “Administrator creates an account”;
ENDIF
```

10. Rule name: Exploit: Social Engineering Used to Access Sensitive Data

```
IF (
        “Exploit: Administrator Social Engineering Account Deleted” rule occurred
        at least 1 time in 5 days after “Exploit: New user accesses sensitive data”,
        “Exploit: Administrator Social Engineering Account Added” rule occurred
    )
        THEN create new event named “An account created then used to access
            sensitive data then the account removed”,
        AND add annotate Offense named “An account created then used to access
        sensitive data then the account removed”;
ENDIF
```

11. Rule name: Suspicious VPN Login

```
IF (
        Username is in (New Created User) reference set,
        AND high-level category is Authentication,
        AND low-level category is User Login Success,
        AND Event Payload contains auth_client="IPSec"
        AND username is contained in (VPN Users) reference set;
        )
        THEN create new event named “Suspicious Login to VPN”,
                AND add sourceIP as the key AND Add the username in Suspicious VPN Login reference map of set;
ENDIF
```

CHAPTER 5: INTEGRATION TEST

5.1 Overview

This chapter introduces the testing results to guarantee we achieved project main purpose and project requirements.

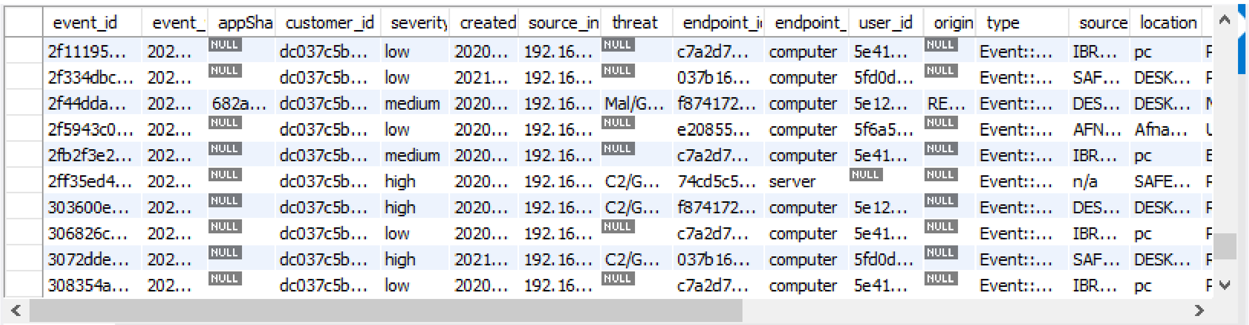
5.2 SAFEDENY-Orchestrator framework test

This section introduces the integration testing for Sophos Central framework with QRadar.

5.2.1 Fetch data from Sophos Central API test

The following are the results of the code for fetch events, alerts, and endpoints. The following shots (5.1, 5.2, and 5.3) show the data after stored in the database tables.

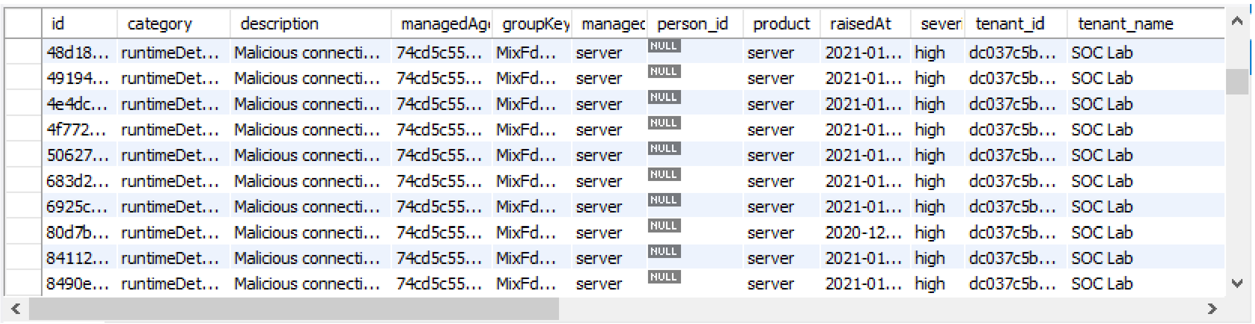
1. Fetch events test. The following shot 5.1 shows part of the events for SAFEDENY SOC Lab in the events table after they are fetched.



event_id	event_	appSha	customer_id	severity	created	source_in	threat	endpoint_i	endpoint_	user_id	origin	type	source	location	
2f11195...	202...	NULL	dc037c5b...	low	2020...	192.16...	NULL	c7a2d7...	computer	5e41...	NULL	Event::...	IBR...	pc	F
2f334dbc...	202...	NULL	dc037c5b...	low	2021...	192.16...	NULL	037b16...	computer	5fd0d...	NULL	Event::...	SAF...	DESK...	F
2f44dda...	202...	682a...	dc037c5b...	medium	2020...	192.16...	Mal/G...	f874172...	computer	5e12...	RE...	Event::...	DES...	DESK...	M
2f5943c0...	202...	NULL	dc037c5b...	low	2020...	192.16...	NULL	e20855...	computer	5f6a5...	NULL	Event::...	AFN...	Afna...	U
2fb2f3e2...	202...	NULL	dc037c5b...	medium	2020...	192.16...	NULL	c7a2d7...	computer	5e41...	NULL	Event::...	IBR...	pc	E
2ff35ed4...	202...	NULL	dc037c5b...	high	2020...	192.16...	C2/G...	74cd5c5...	server	NULL	NULL	Event::...	n/a	SAFE...	F
303600e...	202...	NULL	dc037c5b...	high	2020...	192.16...	C2/G...	f874172...	computer	5e12...	NULL	Event::...	DES...	DESK...	F
306826c...	202...	NULL	dc037c5b...	low	2020...	192.16...	NULL	c7a2d7...	computer	5e41...	NULL	Event::...	IBR...	pc	F
3072dde...	202...	NULL	dc037c5b...	high	2021...	192.16...	C2/G...	037b16...	computer	5fd0d...	NULL	Event::...	SAF...	DESK...	F
308354a...	202...	NULL	dc037c5b...	low	2020...	192.16...	NULL	c7a2d7...	computer	5e41...	NULL	Event::...	IBR...	pc	F

Figure 5.1: Part of the SOC Lab events in the database.

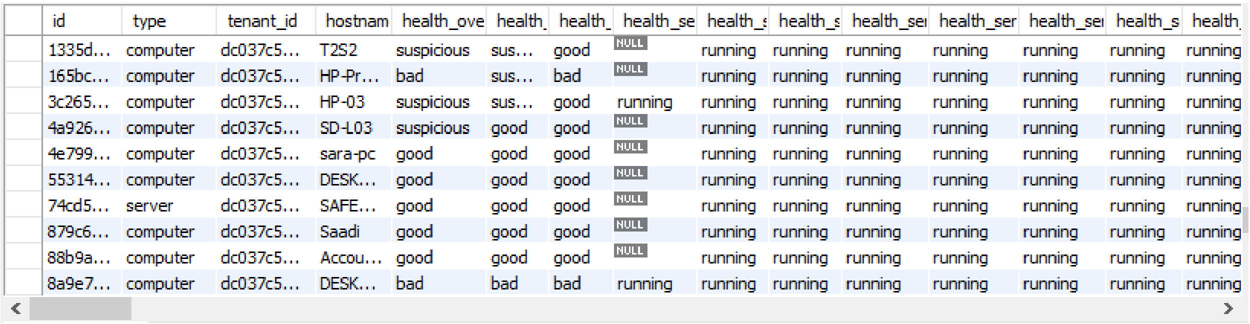
2. Fetch alerts test. The following shot 5.2 shows part of the alerts for SAFEDENY SOC Lab in the alerts table after they are fetched.



id	category	description	managedAg	groupKey	managec	person_id	product	raisedAt	severi	tenant_id	tenant_name
48d18...	runtimeDet...	Malicious connecti...	74cd5c55...	MixFd...	server	NULL	server	2021-01...	high	dc037c5b...	SOC Lab
49194...	runtimeDet...	Malicious connecti...	74cd5c55...	MixFd...	server	NULL	server	2021-01...	high	dc037c5b...	SOC Lab
4e4dc...	runtimeDet...	Malicious connecti...	74cd5c55...	MixFd...	server	NULL	server	2021-01...	high	dc037c5b...	SOC Lab
4f772...	runtimeDet...	Malicious connecti...	74cd5c55...	MixFd...	server	NULL	server	2021-01...	high	dc037c5b...	SOC Lab
50627...	runtimeDet...	Malicious connecti...	74cd5c55...	MixFd...	server	NULL	server	2021-01...	high	dc037c5b...	SOC Lab
683d2...	runtimeDet...	Malicious connecti...	74cd5c55...	MixFd...	server	NULL	server	2021-01...	high	dc037c5b...	SOC Lab
6925c...	runtimeDet...	Malicious connecti...	74cd5c55...	MixFd...	server	NULL	server	2021-01...	high	dc037c5b...	SOC Lab
80d7b...	runtimeDet...	Malicious connecti...	74cd5c55...	MixFd...	server	NULL	server	2020-12...	high	dc037c5b...	SOC Lab
84112...	runtimeDet...	Malicious connecti...	74cd5c55...	MixFd...	server	NULL	server	2021-01...	high	dc037c5b...	SOC Lab
8490e...	runtimeDet...	Malicious connecti...	74cd5c55...	MixFd...	server	NULL	server	2021-01...	high	dc037c5b...	SOC Lab

Figure 5.2: Part of the SOC Lab alerts in the database.

3. Fetch endpoints data test. The following shot 5.3 shows part of the endpoints data for the SAFEDENY SOC Lab in the endpointData table after they are fetched.



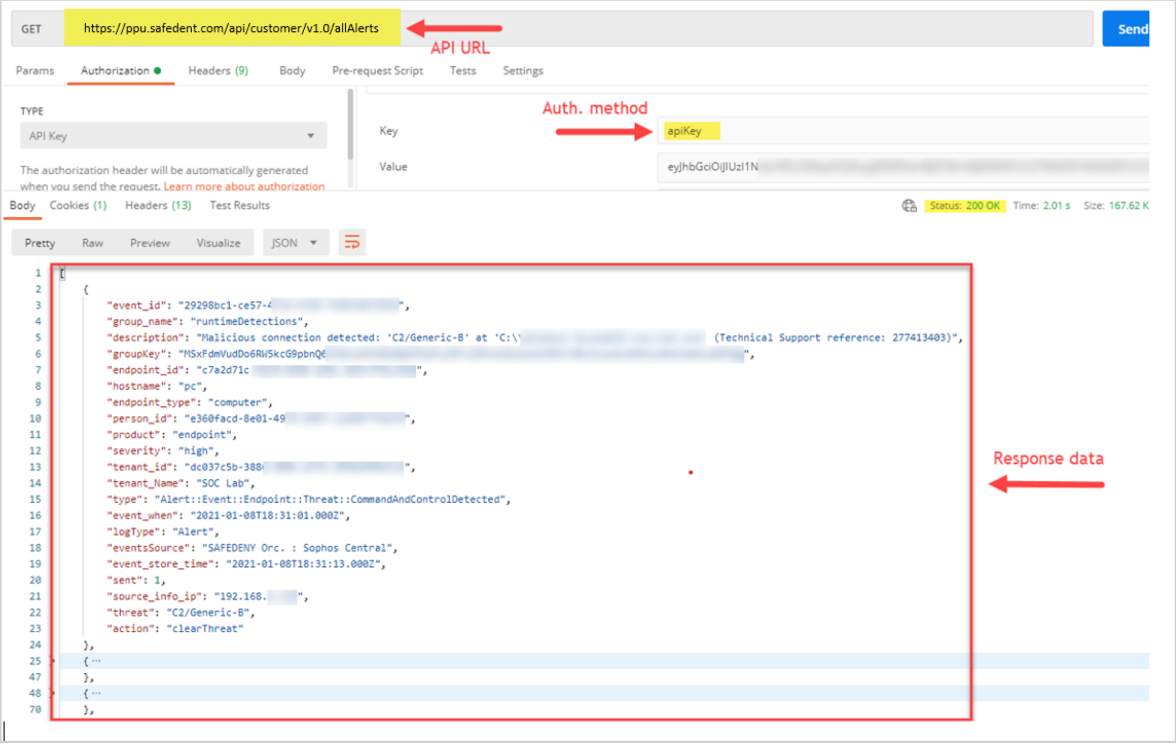
id	type	tenant_id	hostnam	health_ove	health_	health_	health_se	health_s	health_s	health_sei	health_ser	health_sei	health_s	health_
1335d...	computer	dc037c5...	T2S2	suspicious	sus...	good	NULL	running	running	running	running	running	running	running
165bc...	computer	dc037c5...	HP-Pr...	bad	sus...	bad	NULL	running	running	running	running	running	running	running
3c265...	computer	dc037c5...	HP-03	suspicious	sus...	good	running	running	running	running	running	running	running	running
4a926...	computer	dc037c5...	SD-L03	suspicious	good	good	NULL	running	running	running	running	running	running	running
4e799...	computer	dc037c5...	sara-pc	good	good	good	NULL	running	running	running	running	running	running	running
55314...	computer	dc037c5...	DESK...	good	good	good	NULL	running	running	running	running	running	running	running
74cd5...	server	dc037c5...	SAFE...	good	good	good	NULL	running	running	running	running	running	running	running
879c6...	computer	dc037c5...	Saadi	good	good	good	NULL	running	running	running	running	running	running	running
88b9a...	computer	dc037c5...	Accou...	good	good	good	NULL	running	running	running	running	running	running	running
8a9e7...	computer	dc037c5...	DESK...	bad	bad	bad	running	running	running	running	running	running	running	running

Figure 5.3: Part of the endpointsData table in the database.

5.2.2 SAFEDENY-Orchestrator API service test

The following are the results of some APIs requests using the postman tool.

1. (/api/customer/v1.0/allAlerts) API test. The following shot 5.4 shows the result of the allAlerts API request for SAFEDENY SOC Lab using postman tool.



GET https://ppu.safedent.com/api/customer/v1.0/allAlerts

TYPE: API Key

The authorization header will be automatically generated when you send the request. Learn more about authorization

Key: apiKey

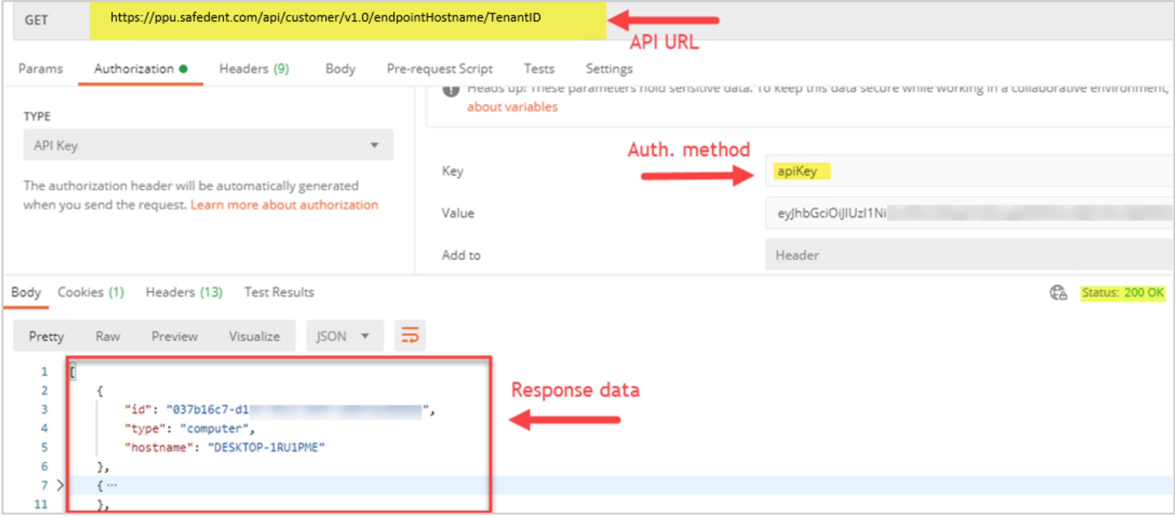
Value: eyJhbGciOiJIUzI1N[redacted]

Status: 200 OK Time: 2.01 s Size: 167.62 K

```
[
    {
        "event_id": "29298bc1-ce57-4[redacted]",
        "group_name": "runtimeDetections",
        "description": "Malicious connection detected: 'C2/Generic-B' at 'C:\[redacted] (Technical Support reference: 277413403)",
        "groupKey": "MSxFdmVudDo6RW5kcG9pbnQ6[redacted]",
        "endpoint_id": "c7a2d71c[redacted]",
        "hostname": "pc",
        "endpoint_type": "computer",
        "person_id": "e360facd-8e01-49[redacted]",
        "product": "endpoint",
        "severity": "high",
        "tenant_id": "dc037c5b-388[redacted]",
        "tenant_Name": "SOC Lab",
        "type": "Alert::Event::Endpoint::Threat::CommandAndControlDetected",
        "event_when": "2021-01-08T18:31:01.000Z",
        "logType": "Alert",
        "eventsSource": "SAFEDENY Orc. : Sophos Central",
        "event_store_time": "2021-01-08T18:31:13.000Z",
        "sent": 1,
        "source_info_ip": "192.168.[redacted]",
        "threat": "C2/Generic-B",
        "action": "clearThreat"
    },
    {…
    },
    {…
    },
```

Figure 5.4: GET request for SOC Lab alerts.

2. (/api/customer/v1.0/endpointHostname/TenantID) API test. The following shot 5.5 shows the result of the endpointHostname API request for SAFEDENY SOC Lab using postman tool.



GET https://ppu.safedent.com/api/customer/v1.0/endpointHostname/TenantID

API URL

TYPE: API Key

The authorization header will be automatically generated when you send the request. Learn more about authorization

Heads up: These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, about variables

Auth. method

Key: apiKey

Value: eyJhbGciOiJIUzI1Ni[redacted]

Add to: Header

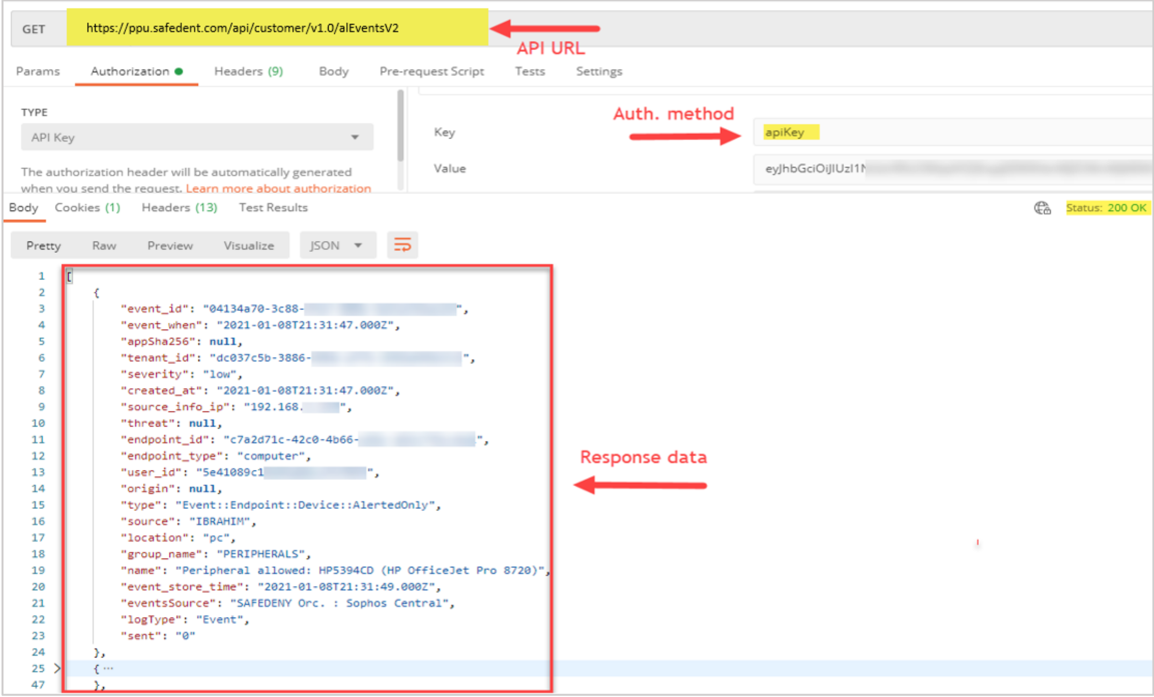
Status: 200 OK

```
[
    {
        "id": "037b16c7-d1[redacted]",
        "type": "computer",
        "hostname": "DESKTOP-1RU1PME"
    },
    {…
    },
```

Response data

Figure 5.5: GET request for SOC Lab endpoints hostname.

3. (/api/customer/v1.0/alEventsV2) API test. The following shot 5.6 shows the result of the allEventsV2 API request for SAFEDENY SOC Lab using postman tool.



GET https://ppu.safedent.com/api/customer/v1.0/alEventsV2

API URL

TYPE: API Key

The authorization header will be automatically generated when you send the request. Learn more about authorization

Auth. method

Key: apiKey

Value: eyJhbGciOiJIUzI1N[redacted]

Status: 200 OK

```
[
    {
        "event_id": "04134a70-3c88-[redacted]",
        "event_when": "2021-01-08T21:31:47.000Z",
        "appSha256": null,
        "tenant_id": "dc037c5b-3886-[redacted]",
        "severity": "low",
        "created_at": "2021-01-08T21:31:47.000Z",
        "source_info_ip": "192.168.[redacted]",
        "threat": null,
        "endpoint_id": "c7a2d71c-42c0-4b66-[redacted]",
        "endpoint_type": "computer",
        "user_id": "5e41089c1[redacted]",
        "origin": null,
        "type": "Event::Endpoint::Device::AlertedOnly",
        "source": "IBRAHIM",
        "location": "pc",
        "group_name": "PERIPHERALS",
        "name": "Peripheral allowed: HP5394CD (HP OfficeJet Pro 8720)",
        "event_store_time": "2021-01-08T21:31:49.000Z",
        "eventsSource": "SAFEDENY Orc. : Sophos Central",
        "logType": "Event",
        "sent": "0"
    },
    {…
    },
```

Response data

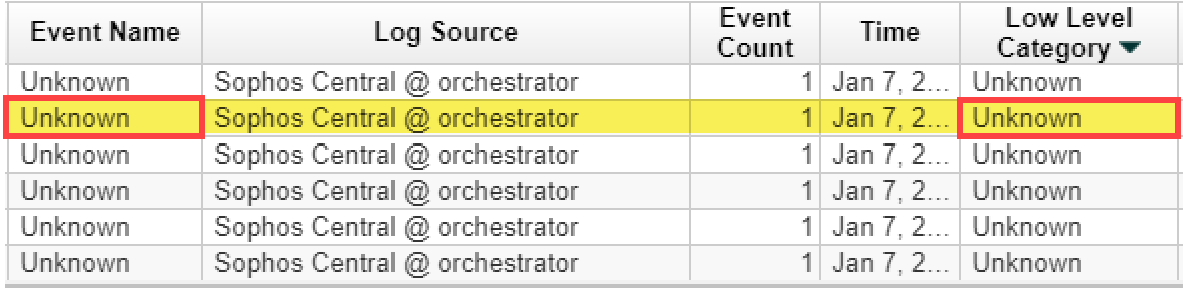
Figure 5.6: GET request for SOC Lab events.

5.3 Testing from QRadar side

This section introduces the testing results from the QRadar aspect.

5.3.1 QRadar application test

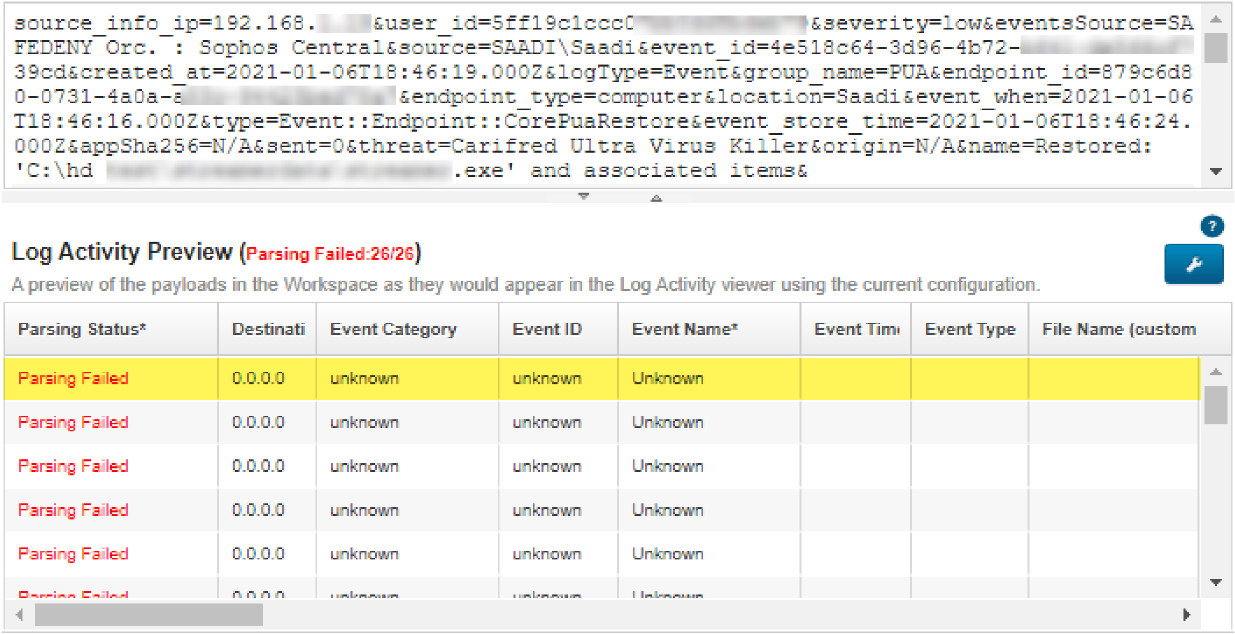
The following shot 5.7 shows the Sophos Central events and alerts that were received in (Sophos Central @ orchestrator) log source at QRadar through HTTP receiver in SAFEDENY-Orchestrator App (QRadar App).



Event Name	Log Source	Event Count	Time	Low Level Category ▼
Unknown	Sophos Central @ orchestrator	1	Jan 7, 2...	Unknown
Unknown	Sophos Central @ orchestrator	1	Jan 7, 2...	Unknown
Unknown	Sophos Central @ orchestrator	1	Jan 7, 2...	Unknown
Unknown	Sophos Central @ orchestrator	1	Jan 7, 2...	Unknown
Unknown	Sophos Central @ orchestrator	1	Jan 7, 2...	Unknown
Unknown	Sophos Central @ orchestrator	1	Jan 7, 2...	Unknown

Figure 5.7: Sophos Central data in QRadar before the normalization.

The following shot 5.8 shows an example of the Sophos Central event payload when it arrived at QRadar and the parsing status.



```
source_info_ip=192.168.[redacted]&user_id=5ff19c1ccc0[redacted]&severity=low&eventsSource=SA
FEDENY Orc. : Sophos Central&source=SAADI\Saadi&event_id=4e518c64-3d96-4b72-[redacted]
39cd&created_at=2021-01-06T18:46:19.000Z&logType=Event&group_name=PUA&endpoint_id=879c6d8
0-0731-4a0a-a[redacted]&endpoint_type=computer&location=Saadi&event_when=2021-01-06
T18:46:16.000Z&type=Event::Endpoint::CorePuaRestore&event_store_time=2021-01-06T18:46:24.
000Z&appSha256=N/A&sent=0&threat=Carifred Ultra Virus Killer&origin=N/A&name=Restored:
'C:\hd [redacted].exe' and associated items&
```

Log Activity Preview (Parsing Failed:26/26)

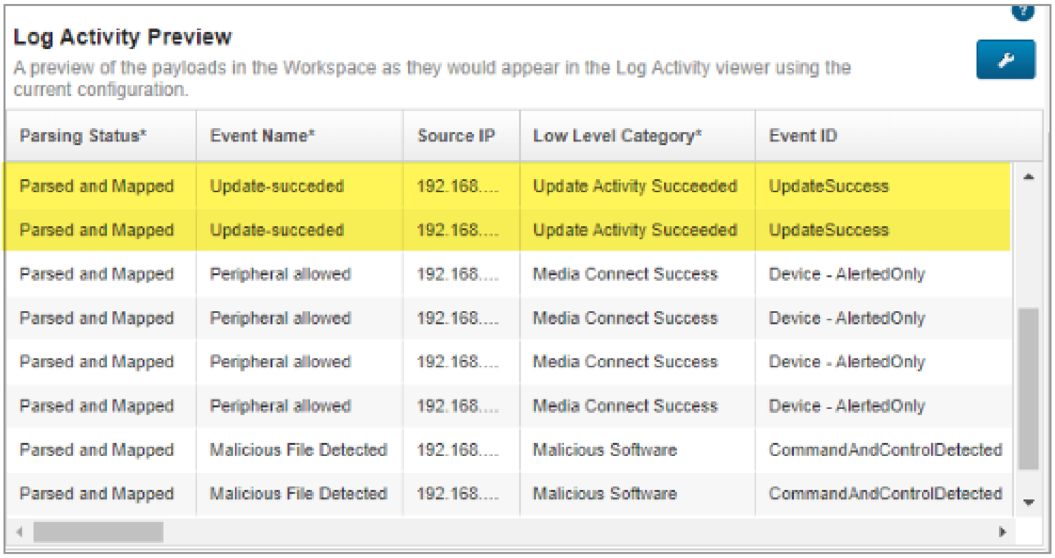
A preview of the payloads in the Workspace as they would appear in the Log Activity viewer using the current configuration.

Parsing Status*	Destinati	Event Category	Event ID	Event Name*	Event Tim	Event Type	File Name (custom
Parsing Failed	0.0.0.0	unknown	unknown	Unknown			
Parsing Failed	0.0.0.0	unknown	unknown	Unknown			
Parsing Failed	0.0.0.0	unknown	unknown	Unknown			
Parsing Failed	0.0.0.0	unknown	unknown	Unknown			
Parsing Failed	0.0.0.0	unknown	unknown	Unknown			
Parsing Failed	0.0.0.0	unknown	unknown	Unknown			

Figure 5.8: Sophos Central event payload before the normalization.

5.3.2 Sophos Central logs mapping and normalization test

When events and alerts reached QRadar, it reached a different log source from what we have defined because the mapping and normalizing process is not ready yet. The following shot 5.9 shows the events and alerts received in the correct log source (Sophos Central @ orchestrator) and normalized and mapped in the correct way.



Log Activity Preview

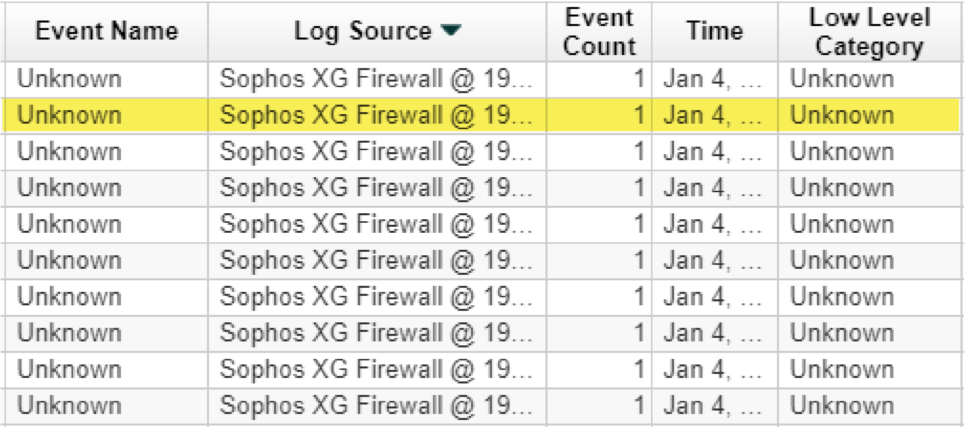
A preview of the payloads in the Workspace as they would appear in the Log Activity viewer using the current configuration.

Parsing Status*	Event Name*	Source IP	Low Level Category*	Event ID
Parsed and Mapped	Update-succeded	192.168....	Update Activity Succeeded	UpdateSuccess
Parsed and Mapped	Update-succeded	192.168....	Update Activity Succeeded	UpdateSuccess
Parsed and Mapped	Peripheral allowed	192.168....	Media Connect Success	Device - AlertedOnly
Parsed and Mapped	Peripheral allowed	192.168....	Media Connect Success	Device - AlertedOnly
Parsed and Mapped	Peripheral allowed	192.168....	Media Connect Success	Device - AlertedOnly
Parsed and Mapped	Peripheral allowed	192.168....	Media Connect Success	Device - AlertedOnly
Parsed and Mapped	Malicious File Detected	192.168....	Malicious Software	CommandAndControlDetected
Parsed and Mapped	Malicious File Detected	192.168....	Malicious Software	CommandAndControlDetected

Figure 5.9: Parsed and mapped Sophos Central events and alerts.

5.3.3 Sophos NGFW Syslog test

The following shot 5.10 shows the Sophos NGFW logs that were received in (Sophos XG Firewall @ 192.168.[redacted]) log source at QRadar through Syslog protocol.

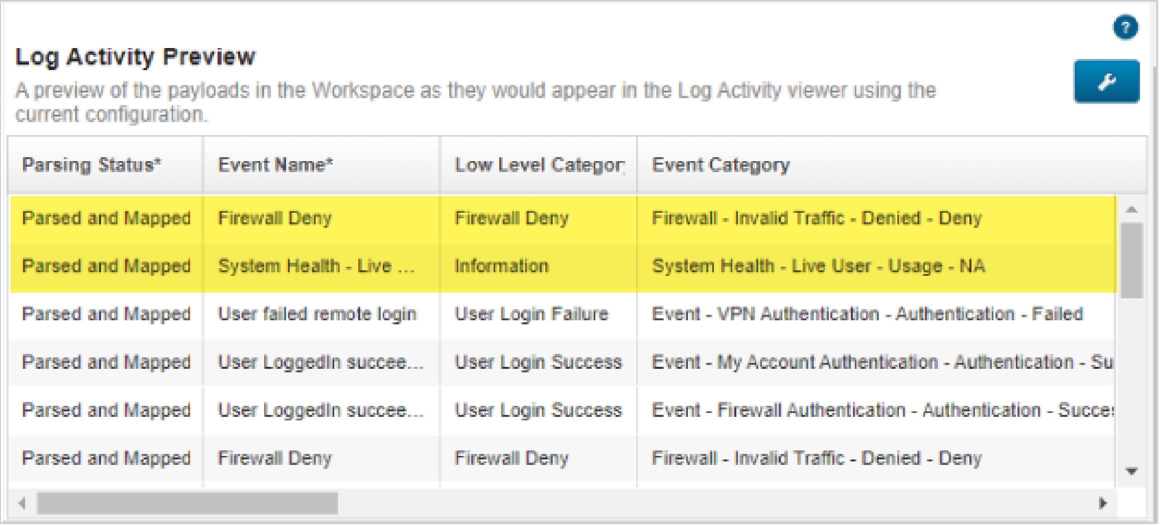


Event Name	Log Source ▼	Event Count	Time	Low Level Category
Unknown	Sophos XG Firewall @ 19...	1	Jan 4, ...	Unknown
Unknown	Sophos XG Firewall @ 19...	1	Jan 4, ...	Unknown
Unknown	Sophos XG Firewall @ 19...	1	Jan 4, ...	Unknown
Unknown	Sophos XG Firewall @ 19...	1	Jan 4, ...	Unknown
Unknown	Sophos XG Firewall @ 19...	1	Jan 4, ...	Unknown
Unknown	Sophos XG Firewall @ 19...	1	Jan 4, ...	Unknown
Unknown	Sophos XG Firewall @ 19...	1	Jan 4, ...	Unknown
Unknown	Sophos XG Firewall @ 19...	1	Jan 4, ...	Unknown
Unknown	Sophos XG Firewall @ 19...	1	Jan 4, ...	Unknown
Unknown	Sophos XG Firewall @ 19...	1	Jan 4, ...	Unknown

Figure 5.10: Sophos NGFW logs in QRadar before the normalization.

5.3.4 Sophos NGFW logs mapping and normalization test

The following shot 5.11 shows Sophos NGFW logs after we have mapped and normalized it.



Log Activity Preview

A preview of the payloads in the Workspace as they would appear in the Log Activity viewer using the current configuration.

Parsing Status*	Event Name*	Low Level Categor	Event Category
Parsed and Mapped	Firewall Deny	Firewall Deny	Firewall - Invalid Traffic - Denied - Deny
Parsed and Mapped	System Health - Live ...	Information	System Health - Live User - Usage - NA
Parsed and Mapped	User failed remote login	User Login Failure	Event - VPN Authentication - Authentication - Failed
Parsed and Mapped	User LoggedIn succee...	User Login Success	Event - My Account Authentication - Authentication - Su
Parsed and Mapped	User LoggedIn succee...	User Login Success	Event - Firewall Authentication - Authentication - Succe
Parsed and Mapped	Firewall Deny	Firewall Deny	Firewall - Invalid Traffic - Denied - Deny

Figure 5.11: Parsed and mapped Sophos NGFW logs.

5.4 Integration test

In this section, we have tested the integration through rules testing. We have applied several scenarios to make sure the rules are working properly and in the following sections, we introduce the scenarios and the response of the rule.

5.4.1 QRadar Rules test

The following scenarios have used to test the rules:

The first scenario: We have executed social engineering attack[2] scenario. In this attack, the attacker is a former employee in X company and his VPN account was disabled after his resignation. Because of one of the human errors, the network administrator has forgotten to disable his account on the domain. He knew of that and exploited one of the employees who has an active VPN account to do malicious things. The following figure 5.12 shows the sequence of this social engineering attack.

[2] Social engineering attack is psychological manipulation of people into performing actions or divulging confidential information[15].

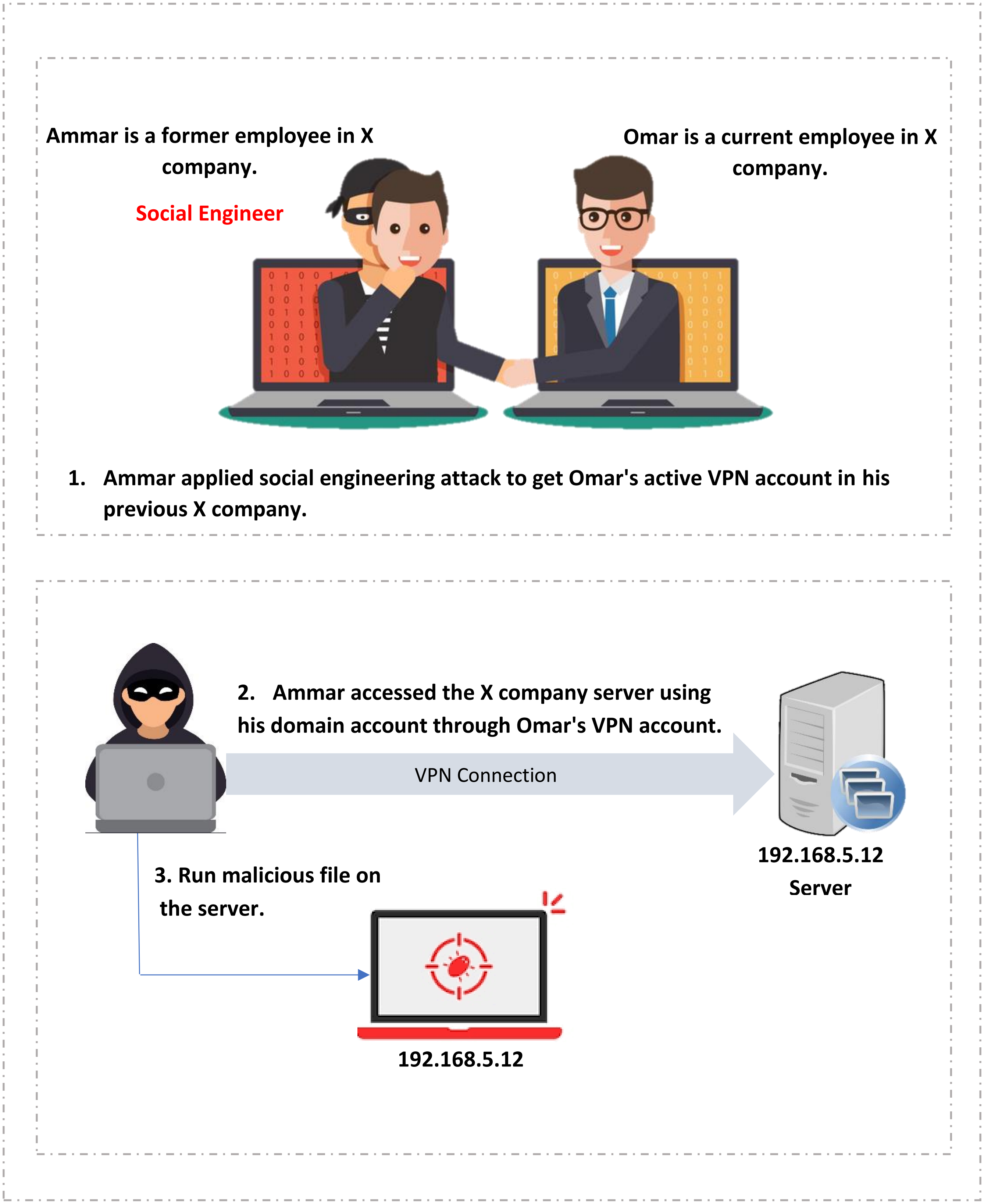
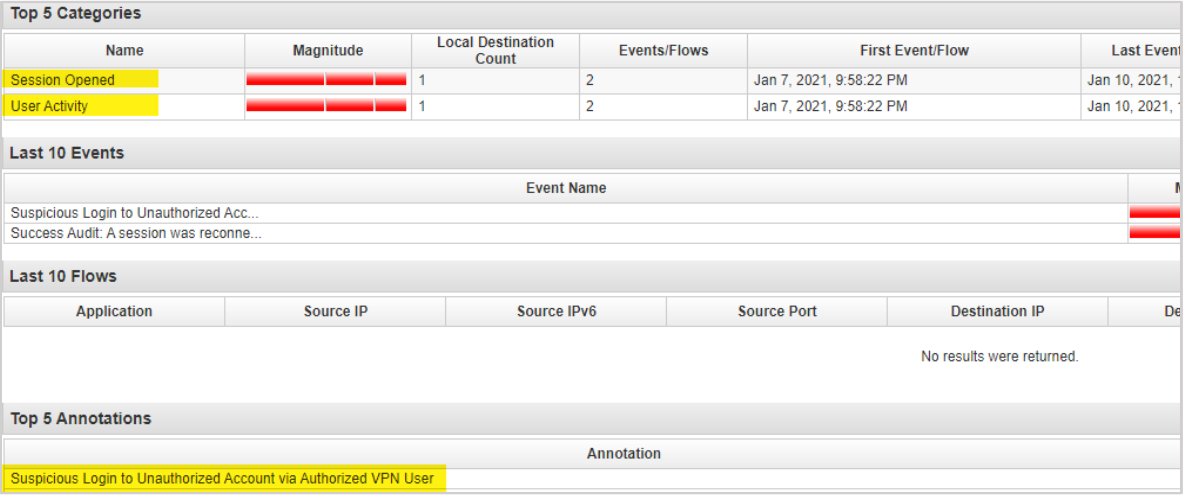

Ammar is a former employee in X company.
Social Engineer
Omar is a current employee in X company.
1. Ammar applied social engineering attack to get Omar's active VPN account in his previous X company.
2. Ammar accessed the X company server using his domain account through Omar's VPN account.
VPN Connection
192.168.5.12
Server
3. Run malicious file on the server.
192.168.5.12

Figure 5.12: Social engineering attack diagram.

The following shot 5.13 shows the “Suspicious Login to Unauthorized Account via Authorized VPN User” rule response. This rule occurred in the previous scene when Ammar login to his domain account which unauthorized to Omar through Omar's VPN account.



Top 5 Categories

Name	Magnitude	Local Destination Count	Events/Flows	First Event/Flow	Last Event
Session Opened	🟥	1	2	Jan 7, 2021, 9:58:22 PM	Jan 10, 2021,
User Activity	🟥	1	2	Jan 7, 2021, 9:58:22 PM	Jan 10, 2021,

Last 10 Events

Event Name	M
Suspicious Login to Unauthorized Acc...	🟥
Success Audit: A session was reconne...	🟥

Last 10 Flows

Application	Source IP	Source IPv6	Source Port	Destination IP	De

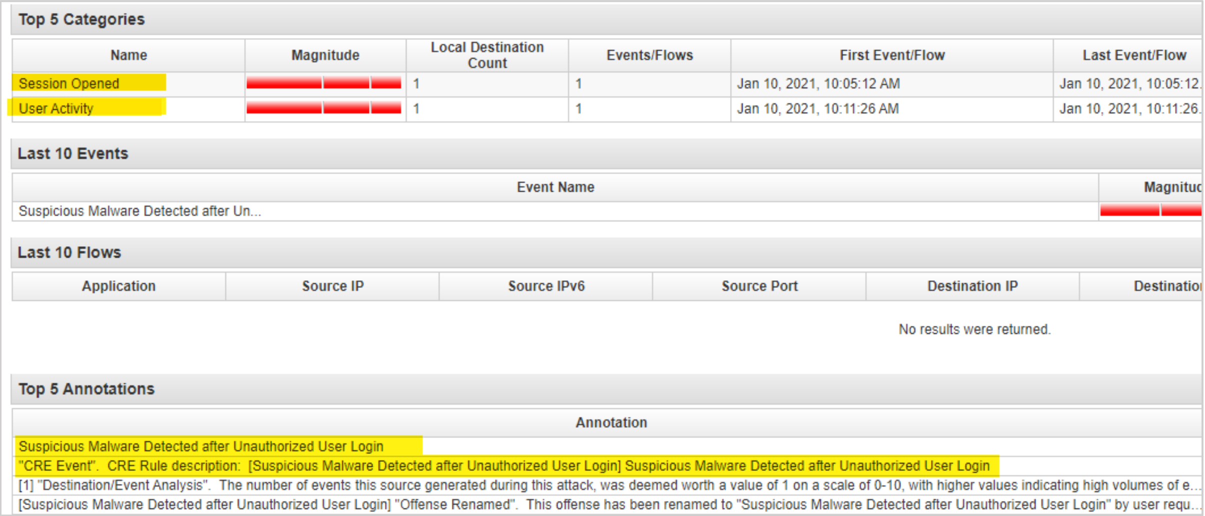
No results were returned.

Top 5 Annotations

Annotation
Suspicious Login to Unauthorized Account via Authorized VPN User

Figure 5.13: “Suspicious Login to Unauthorized Account” rule response.

The following shot 5.14 shows the” Suspicious Malware Detected after Unauthorized User login” rule response. This rule occurred in the previous scene when Ammar run a malicious file after login to the server using his domain account through Omar's VPN account.



Top 5 Categories

Name	Magnitude	Local Destination Count	Events/Flows	First Event/Flow	Last Event/Flow
Session Opened	🟥	1	1	Jan 10, 2021, 10:05:12 AM	Jan 10, 2021, 10:05:12
User Activity	🟥	1	1	Jan 10, 2021, 10:11:26 AM	Jan 10, 2021, 10:11:26

Last 10 Events

Event Name	Magnituc
Suspicious Malware Detected after Un...	🟥

Last 10 Flows

Application	Source IP	Source IPv6	Source Port	Destination IP	Destinatio

No results were returned.

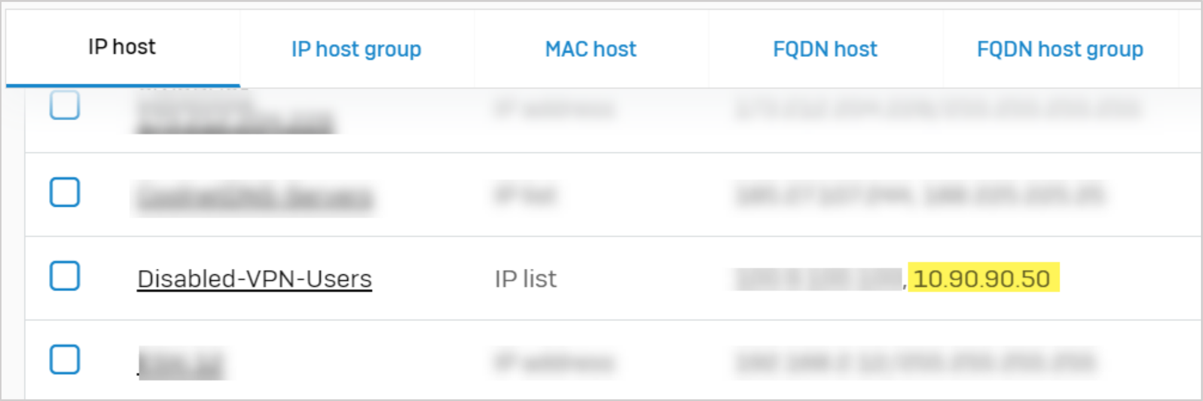
Top 5 Annotations

Annotation
Suspicious Malware Detected after Unauthorized User Login
"CRE Event". CRE Rule description: [Suspicious Malware Detected after Unauthorized User Login] Suspicious Malware Detected after Unauthorized User Login
[1] "Destination/Event Analysis". The number of events this source generated during this attack, was deemed worth a value of 1 on a scale of 0-10, with higher values indicating high volumes of e...
[Suspicious Malware Detected after Unauthorized User Login] "Offense Renamed". This offense has been renamed to "Suspicious Malware Detected after Unauthorized User Login" by user requ...

Figure 5.14: “Suspicious Malware Detected after Unauthorized User login” rule response.

The following shot 5.15 shows the disable VPN user list on Sophos NGFW. This list updated in the previous scene when the “Suspicious Login to Unauthorized Account via Authorized VPN

User” rule occurred and execute the custom action script which sent Omar's VPN IP to Sophos NGFW and added his VPN IP to Disabled-VPN-Users for block access to LAN.



□	[redacted]	[redacted]	[redacted]
□	[redacted]	[redacted]	[redacted]
□	Disabled-VPN-Users	IP list	[redacted], 10.90.90.50
□	[redacted]	[redacted]	[redacted]

Figure 5.15: Disable-VPN-Users list in Sophos NGFW.

The second scenario: We have executed an internal attack[3] scenario. In this attack, the attacker is a network administrator in X company. He wants to theft sensitive data from the company to which the financial group users have access to it but he doesn’t want to do it through his admin account. The following figure 5.16 shows the sequence of this internal attack.

[3] Internal attack is occurring when an individual or a group within an organization seeks to disrupt operations or exploit organizational assets[16].

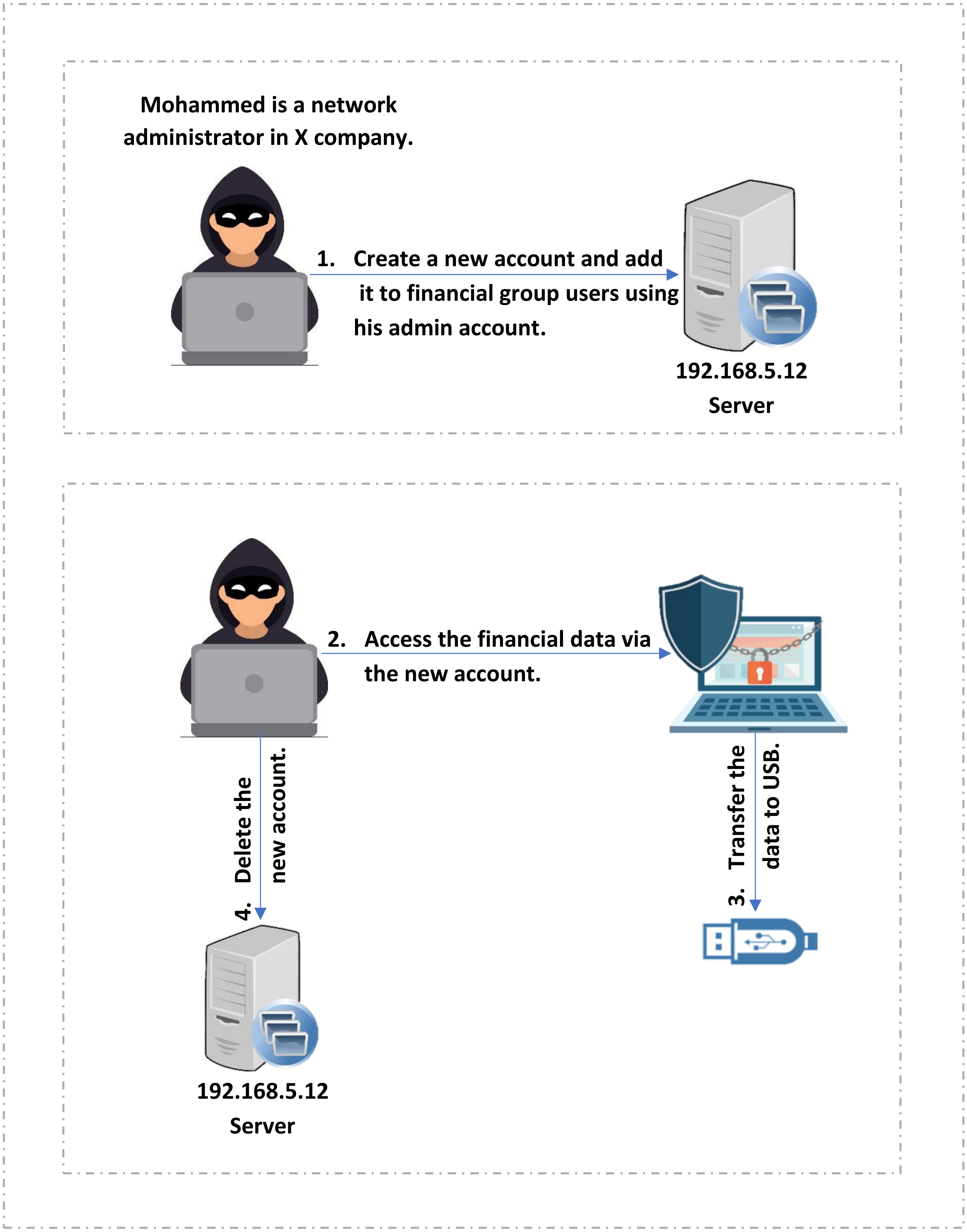
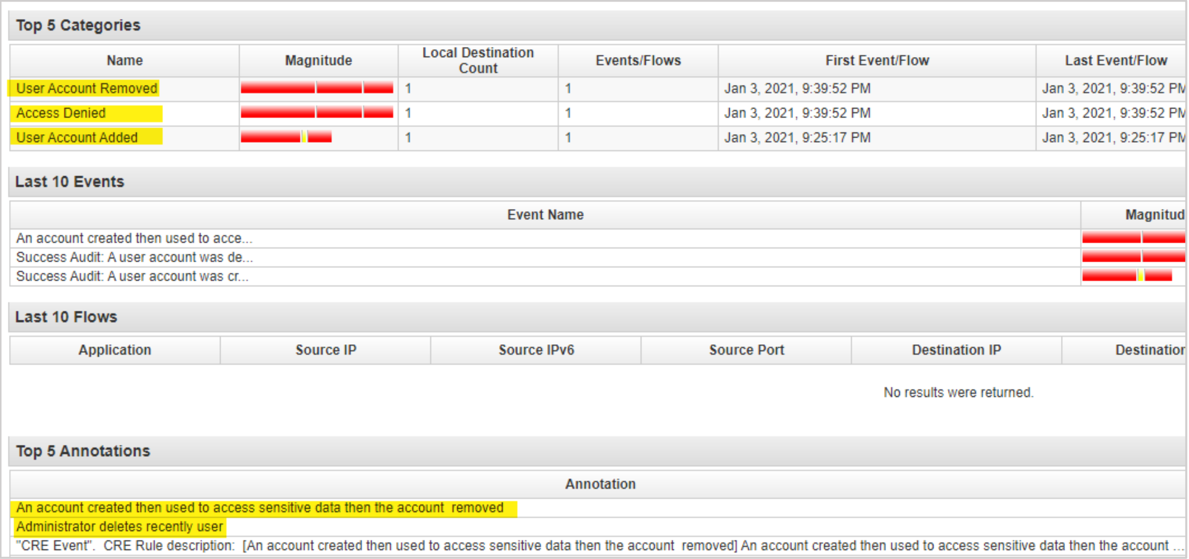

Mohammed is a network administrator in X company.
1. Create a new account and add it to financial group users using his admin account.
192.168.5.12
Server
2. Access the financial data via the new account.
3. Transfer the data to USB.
4. Delete the new account.
192.168.5.12
Server

Figure 5.16: Internal attack diagram.

The following figure 5.17 shows "An account created then used to access sensitive data then the account removed" rule response. This rule occurred in the previous scene when Mohammed created a new account and use it to accessed the sensitive data and tried to transfer the data to a USB, then deleted the account.



Top 5 Categories

Name	Magnitude	Local Destination Count	Events/Flows	First Event/Flow	Last Event/Flow
User Account Removed	🟥	1	1	Jan 3, 2021, 9:39:52 PM	Jan 3, 2021, 9:39:52 PM
Access Denied	🟥	1	1	Jan 3, 2021, 9:39:52 PM	Jan 3, 2021, 9:39:52 PM
User Account Added	🟥	1	1	Jan 3, 2021, 9:25:17 PM	Jan 3, 2021, 9:25:17 PM

Last 10 Events

Event Name	Magnitud
An account created then used to acce...	🟥
Success Audit: A user account was de...	🟥
Success Audit: A user account was cr...	🟥

Last 10 Flows

Application	Source IP	Source IPv6	Source Port	Destination IP	Destinatior

No results were returned.

Top 5 Annotations

Annotation
An account created then used to access sensitive data then the account removed
Administrator deletes recently user
"CRE Event". CRE Rule description: [An account created then used to access sensitive data then the account removed] An account created then used to access sensitive data then the account ...

Figure 5.17: "An account created then used to access sensitive data then the account removed" rule response.

CHAPTER 6: RESULTS AND FUTURE WORK

6.1 Overview

This chapter introduces the challenges which were faced us during the project, contribution results, future works, and conclusion.

6.2 Project challenges

We faced various challenges in our project which are summarized as:

- We worked on new systems that we did not deal with it before.
- Customers refused to let us use their logs and events and this caused the reduction in log types we worked on.
- QRadar is resource-intensive so we need servers with specific characteristics and these servers took a long time to get.
- Quarantine prevented us from working in the company which enforced us to work remotely with poor internet infrastructure.

6.3 Contribution results

After we integrated the three systems, Firstly, we successfully increased the visibility of the network in the company that has Intercept X, Sophos NGFW, and QRadar when Sophos Central and Sophos NGFW was able to send their logs to QRadar and QRadar was able to include these into its AI engine which gave a comprehensive view into this IT infrastructure. We allowed the AI engine in the three systems to work together and analyze all events and this what we will see next.

Moreover, we succeed to decrease attack detection and response time, we achieved it by included Sophos Central and Sophos NGFW logs into the QRadar AI engine and rules. The Automatic analysis and detection for all logs together was decreased the attack detection time and gave immediate detection. The immediate detection appeared as an offense on the top in QRadar with high severity which is increased through the rules.

On the other hand, in the best case when an admin security saw the logs every day, the self-analyzing process will take a lot of time because of a huge number of devices and logs.

Furthermore, we implemented two-way integration successfully which means we successes sent data from QRadar to Sophos NGFW, this point shows how we reduced attack response time. According to the social engineering attack scenario which we applied and shows in figure 5.12. We applied this attack ten times and through block VPN IP to access the LAN we recorded the time that QRadar needs to send the IP to Sophos NGFW list every time. The following table 6.1 shows the results for the time needed to block the VPN user after suspicious activity and the number of repeats for this time.

Table 6.1: Results for time needed to block VPN users after suspicious activity.

Time needed to block VPN user (Seconds)	0	2	5	6	9	60
Number of repeats	2	4	1	1	1	1

As we noticed in the previous table, we got an immediate response, and the most repeated time is four seconds. We noticed there are zero seconds and we got it two times. In this, the attacker was blocked when clicked login directly and this is the best case where no internet issues or lags and QRadar got the logs immediately. Also, there is one case that took one minute, and this because of the lags in the internet and the QRadar receive logs.

Finally, through automatic detection and response, we reduced the human capacity, and when the true offense severity increased and appeared on the top, we reduced the human capacity and false-positive.

6.4 Future works

Our suggestions for future work are summarized as:

- Working on assets that provide endpoint information in the network.
- Working on flow data which provides network activity information that will improve the rule AI engine in QRadar.
- Making two-way integration with Sophos Central.

- Creating new API filters that allow the customer to search and generate reports on his endpoints.
- Creating a new API version that allows a partner to see all his customers data and make control it.

6.5 Conclusion

We have succeeded in developing an integration methodology after studied all options and we achieved the main purpose of our project to integrate the AI of Sophos NGFW and Intercept X deep learning with AI of IBM QRadar SIEM solution to get faster detections and responses to network cyberattacks. Our methodology can be used for any other systems.

References

[1] W.Stallings, L. Brown, Computer Security Principles and Practice. New York: Pearson Education, 2018.

[2] "*Cisco ACE Firewall,"* Accessed on: March. 13, 2020. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SS42VS_DSM/com.ibm.dsm.doc/t_DSM_guide_Cisco_ACE_cfg.html#t_dsm_guide_cisco_ace_cfg

[3] *"Extreme Next-Gen Firewall,"* Accessed on: April.20 2020. [Online]. Available: https://www.sophos.com/en-us/products/next-gen-firewall/enterprise-protection.aspx

[4] "Cyber *Edu, what is Endpoint Security,"* Accessed on: April. 17, 2020. [Online]. Available: https://www.forcepoint.com/cyber-edu/endpoint-security

[5] "*What is SIEM software,"* Accessed on: April. 17, 2020. [Online]. Available: https://www.csoonline.com/article/2124604/what-is-siem-software-how-it-works-and-how-to-choose-the-right-tool.html

[6] "*Barracuda Spam & Virus Firewall,"* Accessed on: March. 12, 2020. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SS42VS_DSM/com.ibm.dsm.doc/c_DSM_guide_Barracuda_firewall.html?view=embed

[7] "*Palo Alto Endpoint Security Manager,"* Accessed on: March. 13, 2020. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SS42VS_DSM/com.ibm.dsm.doc/c_dsm_guide_Palo_Alto_Endpoint_Security_Mgr_overview.html

[8] "*Symantec SGS,"* Accessed on: March. 13, 2020. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SS42VS_DSM/com.ibm.dsm.doc/c_dsm_guide_overview_symantec_sgs.html?view=embed

[9] *"What Is Deep Learning AI? A Simple Guide With 8 Practical Examples,"* Accessed on: May. 7, 2020. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2018/10/01/what-is-deep-learning-ai-a-simple-guide-with-8-practical-examples/#54f243778d4b

[10] *"Sophos, Intercept X Endpoint,"* Accessed on: April. 17 2020. [Online]. Available: https://www.sophos.com/en-us/products/endpoint-antivirus.aspx

[11] "*IBM QRadar SIEM –Overview,* "Accessed on: April. 13, 2020. [Online]. Available: https://www.ibm.com/products/qradar-siem

[12] "QRadar *architecture overview,"* Accessed on: May. 7, 2020. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SS42VS_7.4/com.ibm.qradar.doc/c_qradar_deployment_guide_arch.html

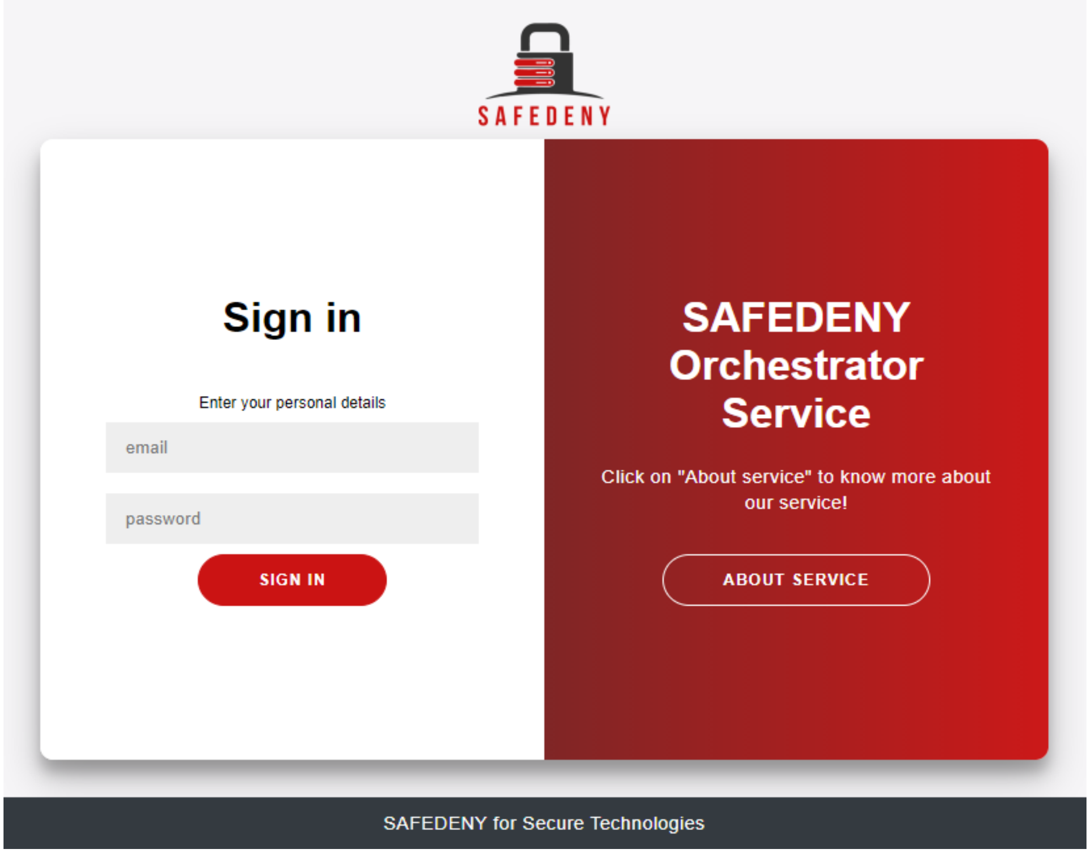
[13] "What *is an API*," Accessed on: May. 7, 2020. [Online]. Available: https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces

[14] "*Sophos XG Firewall: How To add a Syslog Server,* "Accessed on: May. 15, 2020. [Online]. Available: https://community.sophos.com/kb/en-us/123184

[15] *"Social engineering (security),"* Accessed on: Feb. 1, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Social_engineering_(security)#cite_note-1

[16] *"Internal Attack,"* Accessed on: Feb. 1, 2021. [Online]. Available: https://www.techopedia.com/definition/26218/internal-attack

Appendix A



SAFEDENY

Sign in

Enter your personal details

email

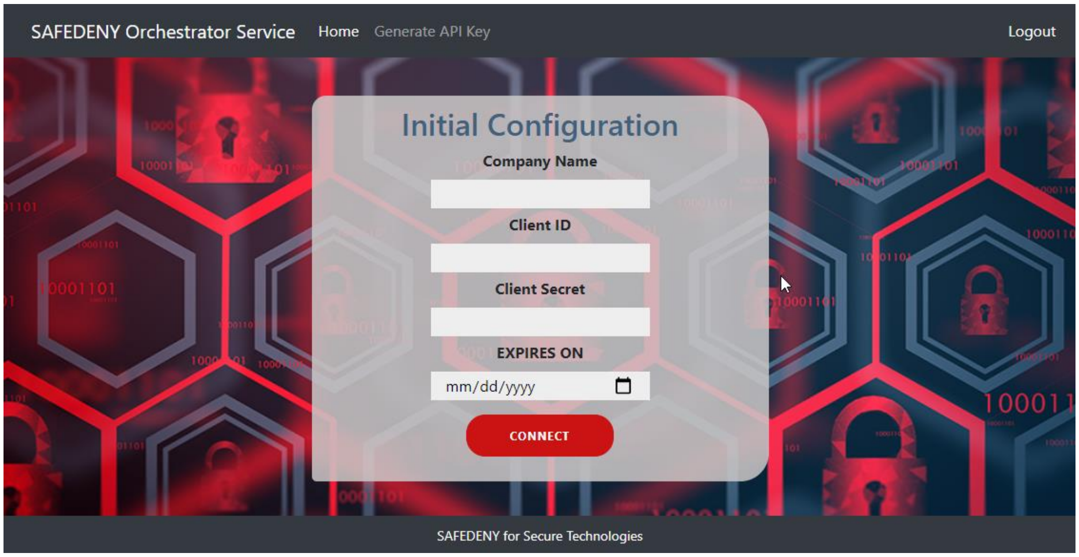
password

SAFEDENY Orchestrator Service

Click on "About service" to know more about our service!

SAFEDENY for Secure Technologies

Figure A.1: Web service login page.



Initial Configuration

Company Name

Client ID

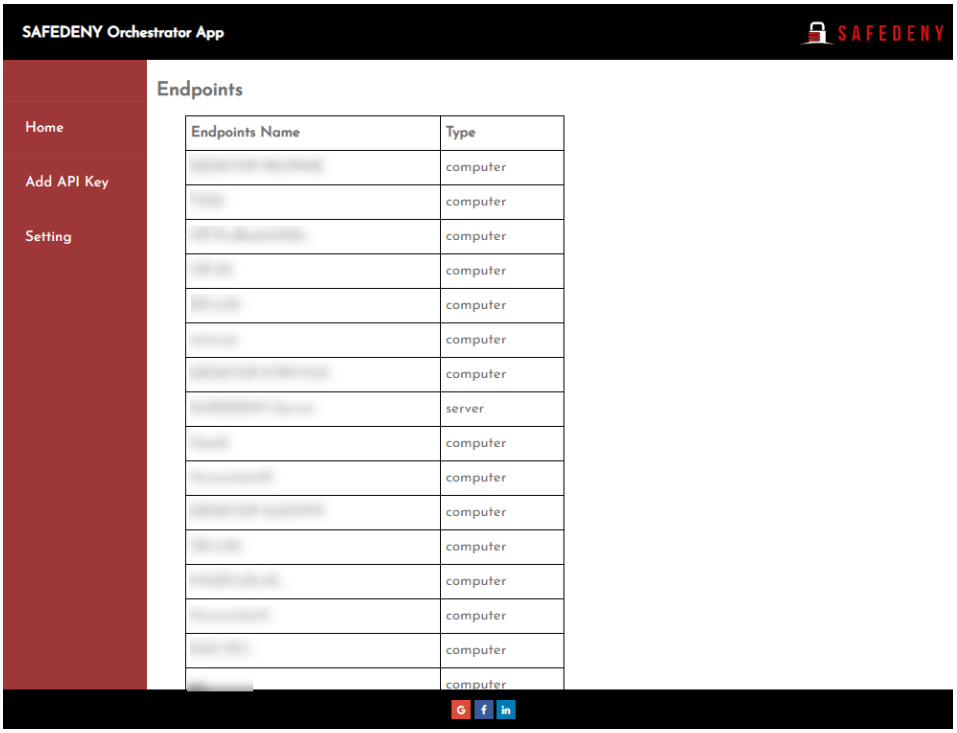
Client Secret

EXPIRES ON

mm/dd/yyyy

SAFEDENY for Secure Technologies

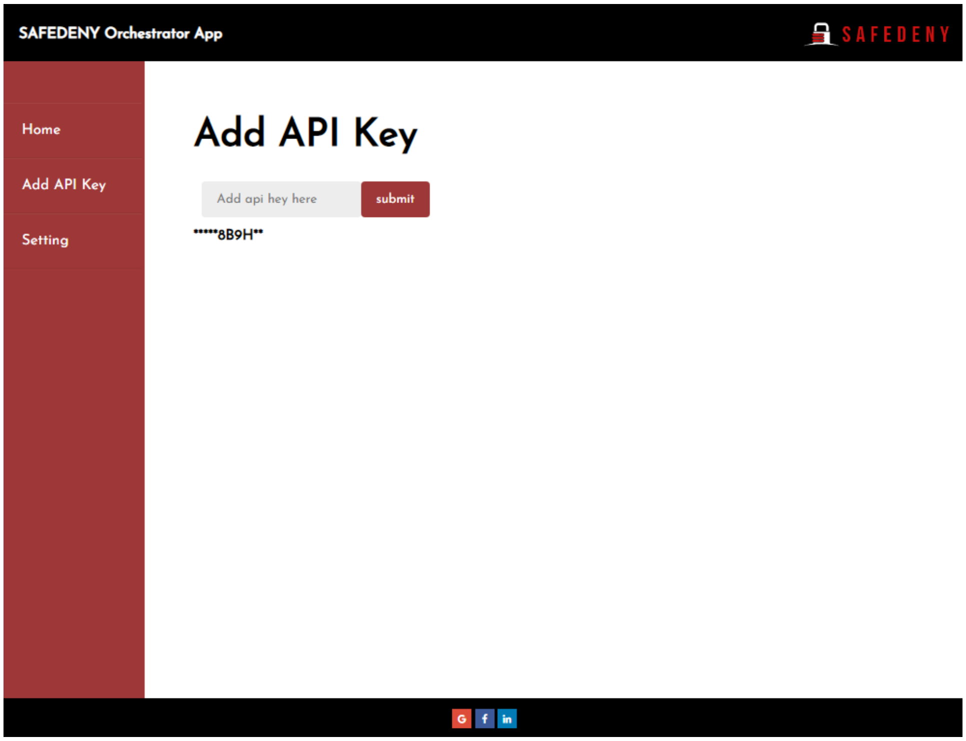
Figure A.2: Web service initial configuration page.



Endpoints

Endpoints Name	Type
[redacted]	computer
[redacted]	computer
[redacted]	computer
[redacted]	computer
[redacted]	computer
[redacted]	computer
[redacted]	computer
[redacted]	server
[redacted]	computer
[redacted]	computer
[redacted]	computer
[redacted]	computer
[redacted]	computer
[redacted]	computer
[redacted]	computer
[redacted]	computer

Figure A.3: SAFEDENY-Orchestrator Home page.



Add API Key

*****8B9H**

Figure A.4: SAFEDENY-Orchestrator Add API Key page.

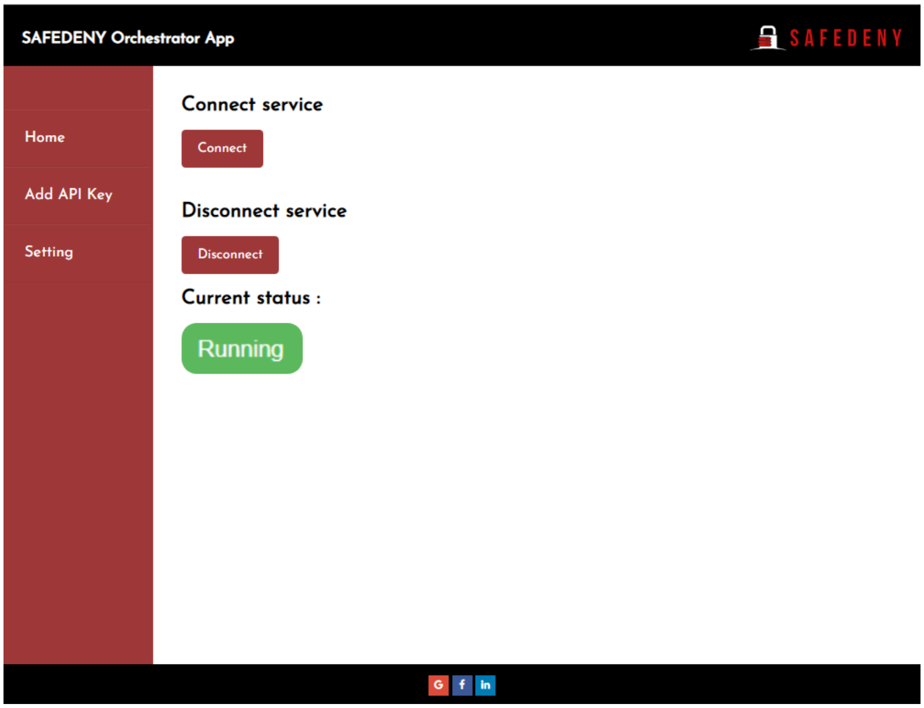


Figure A.5: SAFEDENY-Orchestrator Setting page one.

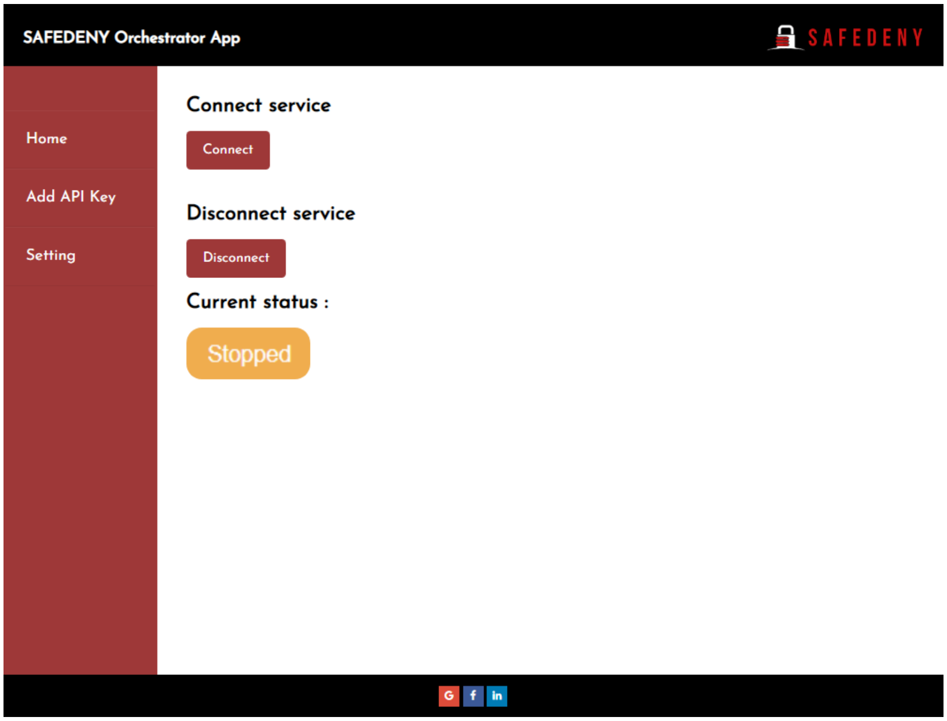


Figure A.6: SAFEDENY-Orchestrator Setting page two.

Appendix B

Table B.1: Rest APIs in the SAFEDENY-Orchestrator.

<table>
<tr><th>Number</th><th colspan="6">API Details</th></tr>
<tr><td rowspan="6">1</td><td>Method</td><td colspan="5">URL</td></tr>
<tr><td>GET</td><td colspan="5">/api/customer/v1.0/TenantID/DeviceID</td></tr>
<tr><td colspan="2">Authentication</td><td>Key</td><td>api key</td><td>Value</td><td><value></td></tr>
<tr><td colspan="2">Param</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="6">Description</td></tr>
<tr><td colspan="6">It returns all endpoint information for any customer from the database and the header must include the apiKey for authentication.</td></tr>
<tr><td rowspan="6">2</td><td>Method</td><td colspan="5">URL</td></tr>
<tr><td>GET</td><td colspan="5">/api/customer/v1.0/TenantID</td></tr>
<tr><td colspan="2">Authentication</td><td>Key</td><td>api key</td><td>Value</td><td><value></td></tr>
<tr><td colspan="2">Param</td><td>deviceId</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="6">Description</td></tr>
<tr><td colspan="6">It returns a specific endpoint information for a specific customer from the database.</td></tr>
<tr><td rowspan="4">3</td><td>Method</td><td colspan="5">URL</td></tr>
<tr><td>GET</td><td colspan="5">/TamperProtectionEnabledState</td></tr>
<tr><td colspan="2">Authentication</td><td>Key</td><td>api key</td><td>Value</td><td><value></td></tr>
<tr><td colspan="2">Param</td><td>deviceId</td><td>-</td><td>-</td><td>-</td></tr>
</table>

	Description				
	It returns the tamper protection status for a specific endpoint from the database.				
4	**Method**	**URL**			
	GET	/api/customer/v1.0/ThreatStatus/TenantID			
	Authentication	Key	api key	Value	<value>
	Param	state	-	-	-
	Description				
	It returns the threat status for all endpoints from the database.				
5	**Method**	**URL**			
	GET	/api/customer/v1.0/ThreatStatus/TenantID/DeviceID			
	Authentication	Key	api key	Value	<value>
	Param	deviceId	-	-	-
	Description				
	It returns the threat status for the endpoint from the database.				
6	**Method**	**URL**			
	GET	/api/customer/v1.0/HealthOverall/TenantID			
	Authentication	Key	api key	Value	<value>
	Param	state	-	-	-

	Description				
	It returns the health status for all endpoints from the database.				
7	Method	URL			
	GET	/api/customer/v1.0/HealthOverall/TenantID/DeviceID			
	Authentication	Key	api key	Value	<value>
	Param	deviceId	-	-	-
	Description				
	It returns the health status for a specific endpoint from the database.				
8	Method	URL			
	GET	/api/customer/v1.0/ServiceDetailse/TenantID			
	Authentication	Key	api key	Value	<value>
	Param	state	-	-	-
	Description				
	It returns the service status for a specific endpoint from the database.				
9	Method	URL			
	GET	/api/customer/v1.0/ServiceDetailse/TenantID/DeviceID			
	Authentication	Key	api key	Value	<value>
	Param	deviceId	-	-	-

	Description				
	It returns the service details for a specific endpoint from the database.				
10	**Method**	**URL**			
	GET	/api/customer/v1.0/endpointHostname/TenantID			
	Authentication	Key	api key	Value	<value>
	Param	-	-	-	-
	Description				
	It returns the hostname for all endpoints for a specific customer from the database.				

Appendix C

Table C.1: Rest of Sophos Central normalized properties.

Properties	Expression type	Expression	Value Delimiter/ Format String	Delimiter
Application (custom)	Regex	&name=Application ([^"]+) was	1	
		APPLICATION_CONTROL([^"]+)&name=([^"]+): ([^"]+)&	3	
Destination IP	NAME VALUE PAIR	source_info_ip	=	&
Detection Component (custom)	NAME VALUE PAIR	origin	=	&
Event Category	Regex	&eventsSource=([^"]+) : ([^"]+)&source([^"]+)&type=([^"]+)::([^"]+)::([^"]+)::([^"]+)&event_store_time=	$1 - $2 - $4 - $5 - $6 - $7	
		&eventsSource=([^"]+) : ([^"]+)&source([^"]+)&type=([^"]+)::([^"]+)::([^"]+)&event_store_time=	$1 - $2 - $4 - $5 - $6	
		&eventsSource=([^"]+) : ([^"]+)&tenant_Name([^"]+)&type=([^"]+	$1 - $2 - $4 - $5 - $6 - $7 - $8	

		)::([^"]+)::([^"]+)::([^"]+)::([^"]+)&endpoint_type		
		&eventsSource=([^"]+) : ([^"]+)&tenant_Name([^"]+)&type=([^"]+)::([^"]+)::([^"]+)::([^"]+)&endpoint_type	$1 - $2 - $4 - $5 - $6 - $7	
		&eventsSource=([^"]+) : ([^"]+)&tenant_Name([^"]+)&type=([^"]+)::([^"]+)::([^"]+)&endpoint_type	$1 - $2 - $4 - $5 - $6	
Event Created Time (custom)	NAME VALUE PAIR	created_at	=	&
Event ID	Regex	&type=([^"]+)::Firewall::([^"]+)&event_store	$2	
		&group_name=DATA_LOSS_PREVENTION&([^"]+)&type=([^"]+)::([^"]+)::([^"]+)&event_store	$4	
		&type=([^"]+)::([^"]+)::([^"]+)::([^"&]+)&event_store	$3 - $4	
		&name=([^"]+):	$1	
		&type=([^"]+)::([^"]+)::([^"]+)&event_store	$3	

		&type=([^"]+)::([^"]+)::([^"]+)&endpoint_type	$3	
		&type=([^"]+)::([^"]+)::([^"]+)::([^"]+)::([^"]+)&action	$4 - $5	
		&type=([^"]+)::([^"]+)::([^"]+)::([^"]+)&action	$4	
Event Time	NAME VALUE PAIR	event_when	=	&c
Event Type	NAME VALUE PAIR	group_name	=	&
Host ID (custom)	NAME VALUE PAIR	endpoint_id	=	&
Hostname	NAME VALUE PAIR	location	=	&
Host Type (custom)	NAME VALUE PAIR	endpoint_type	=	&
Log Type (custom)	NAME VALUE PAIR	logType	=	&
Peripheral Name (custom)	Regex	&name=Peripheral ([^"]+): ([^"]+)&	2	
Source IP	NAME VALUE PAIR	source_info_ip	=	&
Tenant ID (custom)	NAME VALUE PAIR	tenant_id	=	&

Tenant Name (custom)	Regex	tenant_Name=([^"]+)&group_name	=	
Threat Location (custom)	Regex	&name=([^"]+) at '([^"]+)'	2	
		&name=([^"]+) at '([^"]+)	2	
Threat Name (custom)	NAME VALUE PAIR	threat	=	&
User ID (custom)	NAME VALUE PAIR	user_id	=	&
Web Category (custom)	Regex	&name=([^"]+) due to category '([^"]+)'&	2	
Username	Regex	source=([^"]+)\\([^"]+)&event_id	$2	
	NAME VALUE PAIR	source	=	&
URL (custom)	Regex	WebControlViolation([^"]+)&name='([^"]+) '	2	

Table C.2: Rest of Sophos Central mapped events.

Event Category	QID Name	Low Level Category	QID Severity
SAFEDENY Orc. - Sophos Central - Event - Endpoint - Application - Blocked	Application - Blocked	Access Denied	4
SAFEDENY Orc. - Sophos Central - Alert - Event - Endpoint - Threat - CleanupFailed	Manual cleanup required	Remove Failed	8
SAFEDENY Orc. Sophos Central - Event - Endpoint - CoreClean	Malware detected and successfully deleted	Remove Successful	3
SAFEDENY Orc. - Sophos Central - Event - Endpoint - CoreDetection	Malicious File Detected	Malicious Software	6
SAFEDENY Orc. - Sophos Central - Event - Endpoint - Device - AlertedOnly	Peripheral allowed	Media Connect Success	1
SAFEDENY Orc. - Sophos Central - Alert - Event - Firewall - FirewallAdvancedThreatProtection	Suspected botnet detected	Botnet Address	7
SAFEDENY Orc. - Sophos Central - Alert - Event - Other - FirewallFirmwareUpdateSuccessfullyFinished	FirewallFirmware UpdateSuccessfullyFinished	Update Activity Succeeded	1

SAFEDENY Orc. - Sophos Central - Alert - Event - Firewall - FirewallHAStateDegraded	Firewall HA State Degraded	Warning	5
SAFEDENY Orc. - Sophos Central - Alert - Event - Firewall - FirewallGatewayDown	FirewallGatewayDown	Gateway Status	7
SAFEDENY Orc. - Sophos Central - Alert - Event - Firewall - FirewallGatewayUp	FirewallGatewayUp	Gateway Status	2
SAFEDENY Orc. - Sophos Central - Alert - Event - Firewall - FirewallHAStateRestored	FirewallHAStateRestored	Information	2
SAFEDENY Orc. - Sophos Central - Alert - Event - Firewall - FirewallVPNTunnelDown	FirewallVPNTunnelDown	Session Terminated	4
SAFEDENY Orc. - Sophos Central - Event - Firewall - FirewallVPNTunnelUp	FirewallVPNTunnelUp	Session Opened	1
SAFEDENY Orc. - Sophos Central - Event - Endpoint - HmpaExploitPrevented	Blocked Exploit	Quarantine Successful	3

Table C.3: Rest of Sophos NGFW normalized properties.

Properties	Expression type	Expression	Value Delimiter/ Format String	Delimiter
Destination IP	NAME VALUE PAIR	dst_ip	=	/s
Destination Port	NAME VALUE PAIR	dst_port	=	/s
Event Category	Regex	log_type="([^"]+)"\slog_component="([^"]+)" log_subtype="([^"]+)" status="([^"]+)"	$1 - $2 - $3 - $4	
	Regex	log_type="([^"]+)"\slog_component="([^"]+)" log_subtype="([^"]+)"	$1 - $2 - $3 - NA	
Event ID	NAME VALUE PAIR	log_id	=	/s
Source IP	NAME VALUE PAIR	src_ip	=	/s
Source Port	NAME VALUE PAIR	src_port	=	/s
Username	Regex	^<.*user_name="([^"]+)"\suser_gp.*	$1	

	Regex	^<.*\spriority=.*user_name="([^"]+)"\ssrc_ip=.*	$1	
	Regex	^<.*user_name="([^"]+)"\susergroupname.*	$1	
Destination Zone (custom)	NAME VALUE PAIR	dstzone	=	/s
Source Zone (custom)	NAME VALUE PAIR	srczone	=	/s
Message	Regex	message="([^"]+)"	1	

Table C.4: Rest of Sophos NGFW mapped events.

Event Category	QID Name	Low Level Category	QID Severity
Firewall - Firewall Rule - Allowed - Allow	Firewall Permit	Firewall Permit	0
Firewall - Firewall Rule - Denied - Deny	Firewall Deny	Firewall Deny	4
Firewall - Invalid Traffic - Denied - Deny	Firewall Deny	Firewall Deny	4
Firewall - Appliance Access - Denied - Deny	Firewall Deny	Firewall Deny	4

Firewall - SSL VPN - Denied - Deny	VPN Login Failed	Remote Access Login Failed	3
Firewall - ICMP ERROR MESSAGE - Allowed - Allow	ICMP	ICMP	1
Anti-Spam - SMTP - Allowed - NA	VPN Client: Established	VPN Opened	1
Content Filtering - HTTP - Allowed - NA	Webfilter Allowed	Firewall Permit	0
Content Filtering - HTTP - Denied - NA	Webfilter Blocked	Firewall Deny	4
Event - CLI - Admin - Failed	Admin Failed Login	Admin Login Failure	3
Content Filtering - Application - Denied - NA	Application Control Block	Firewall Deny	4
Event - AD SSO - Authentication - Failed	Channel Authentication Failed	General Authentication Failed	3
Event - IPSec - System - Expire	Connection terminated for peer	IPSec Session Ended	1
Event - DHCP Server - System - Renew	DHCP Renew Lease	DHCP Success	1

Event - DHCP Server - System - Expire	Expire	DHCP Session Closed	1
Event - IPSec - System -Expire	IKE SA lifetime expired.	IKE Error	1
Event - IPSec - System - Failed	IPSec Authentication Failed	IPSec Authentication Failed	4
System Health - Disk - Usage - NA	System Health - Disk - Usage	Information	2
System Health - Interface - Usage - NA	System Health - Interface - Usage	Information	2
System Health - Live User - Usage - NA	System Health - Live User - Usage	Information	2
System Health - Memory - Usage - NA	System Health - Memory - Usage	Information	2
Event - Anti-Virus - System - NA	Update-succeeded	Update Activity Succeeded	1
Event - SSL VPN - System - NA	VPN: Session Started	VPN Opened	1