SOC Automation Project- Asagba Jeremiah

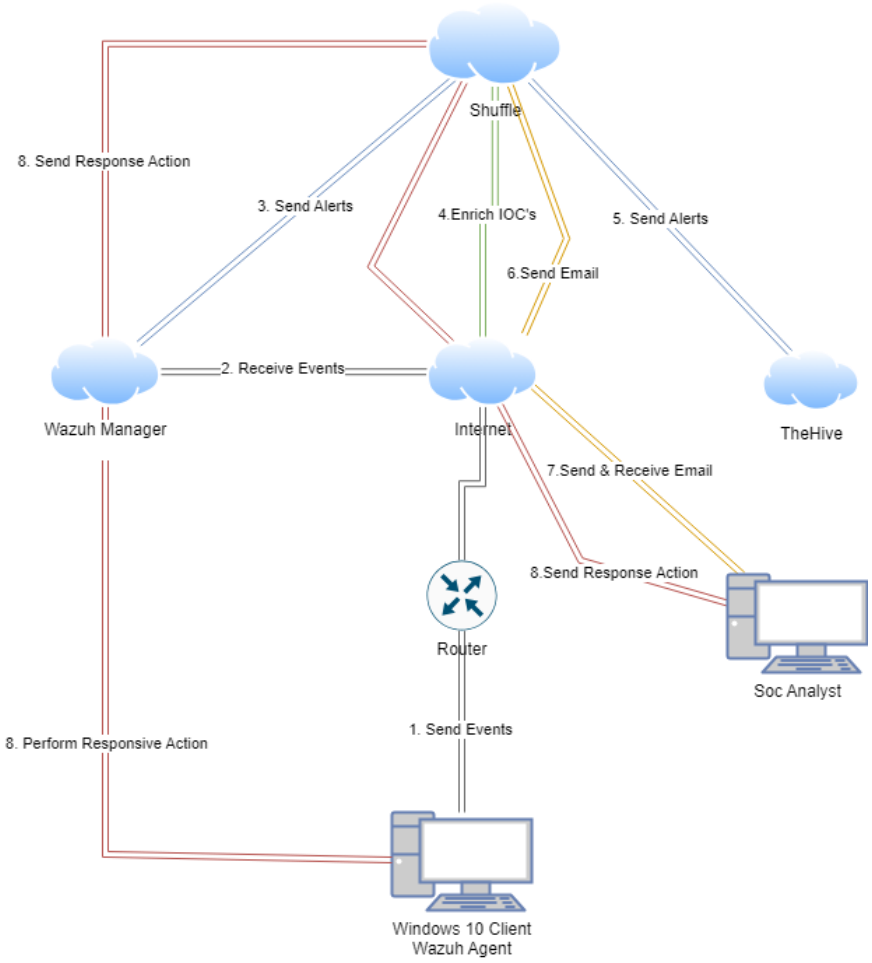
Objective: The SOC Automation Project aimed to create a robust and automated SOC environment capable of effectively detecting, analyzing, and responding to security incidents The primary focus remains on detecting Mimikatz, utilizing Wazuh for SIEM,Windows 10 agents, TheHive for case management, and now incorporating Shuffle with SOAR workflows

Tools/Machines:

Draw.io-Design environment
Wazuh Manager- Extended detection and response (XDR) and Security information and event management (SIEM)-Ubuntu 22.04
TheHive-Open-source Security Incident Response Platform-Ubuntu 22.04
Shuffle->Security orchestration, automation and response (SOAR)
Windows 10 Wazuh agent/client
Sysmon-System Monitor is a Windows system service and device driver

Design:

Draw.io was used to design the logical structure of data flow.
Draw.io is an open-source diagram maker. This design visual can help me understand where and how the data flow will be implemented in the project. It also points out different connections between tools and machines. Below is my Draw.io diagram that I referenced during the project. Screenshot



Credits: https://www.youtube.com/@MyDFIR

Setup Explanation:

1. A Windows 10 client with Wazuh agent will send events through the internet to the Wazuh manager that will host a Wazuh dashboard which I will showcase further in the project.
2. The Wazuh manager/dashboard will send the alert through the Shuffle-SOAR platform and add it under case management in Thehive.
3. Shuffle not only aids in enriching the Indicators of Compromise (IOCs) but also facilitates the exchange of emails between Shuffle and the Security Operations Center (SOC) Analyst.
    a. I am in the process of establishing a lab that involves the utilization of Mimikatz, a program designed for extracting passwords, hashes, PINs, and Kerberos tickets from Windows memory. In the event that the sysmon on the Windows 10 client detects the presence of the Mimikatz application, the Wazuh client will promptly send an alert to Shuffle.
    b. Shuffle will extract the SHA 256 hash to check reputation score with Virustotal-Analyze suspicious files, domains, IPs and URLs to detect malware and other breaches, and automatically share them with the security community.

c. Shuffle will then send updated information to the Thehive for case management.
d. Lastly, send an email to an SOC analyst.

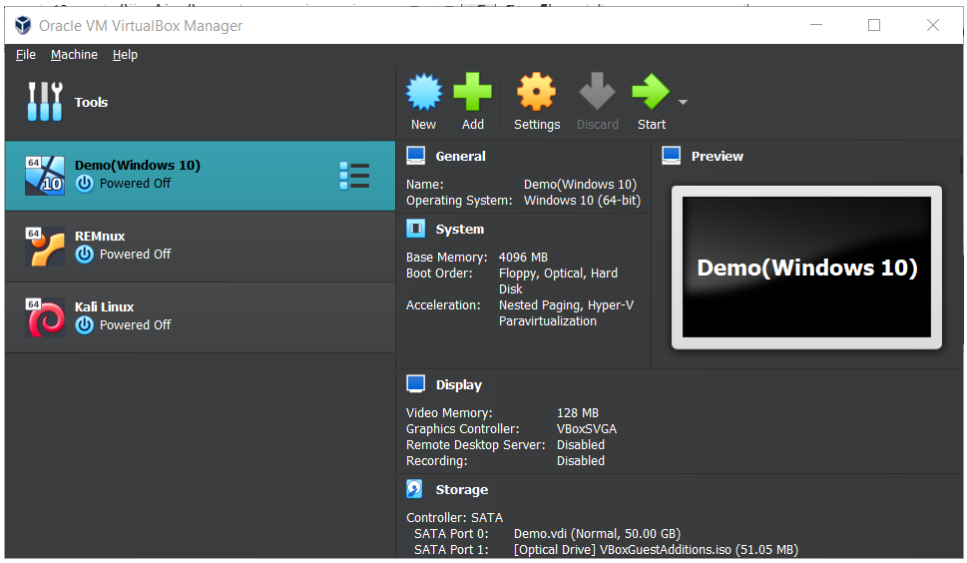
4. The SOC analyst will receive an email containing an option to respond to the alert, which will follow a path through Shuffle, the dashboard, and ultimately reach the Windows 10 client.

Machine configurations:

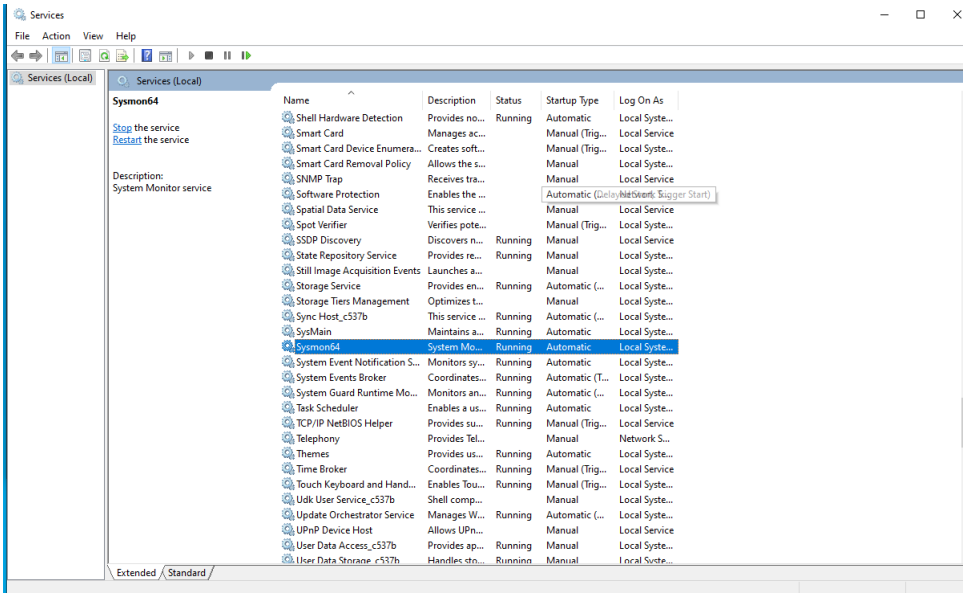
Windows 10 and sysmon setup: [My GitHub]

instruction: - github.com/Fynnesse/SOC-Automation-Lab

A. I utilized VirtualBox where I set up Windows 10 ISO from Microsoft:



B. Sysmon on windows 10:

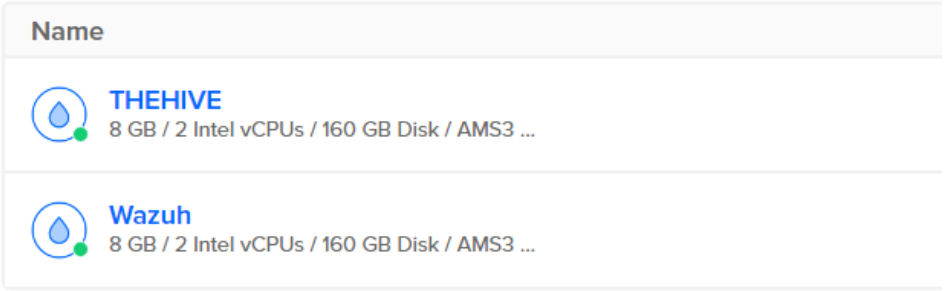


C.

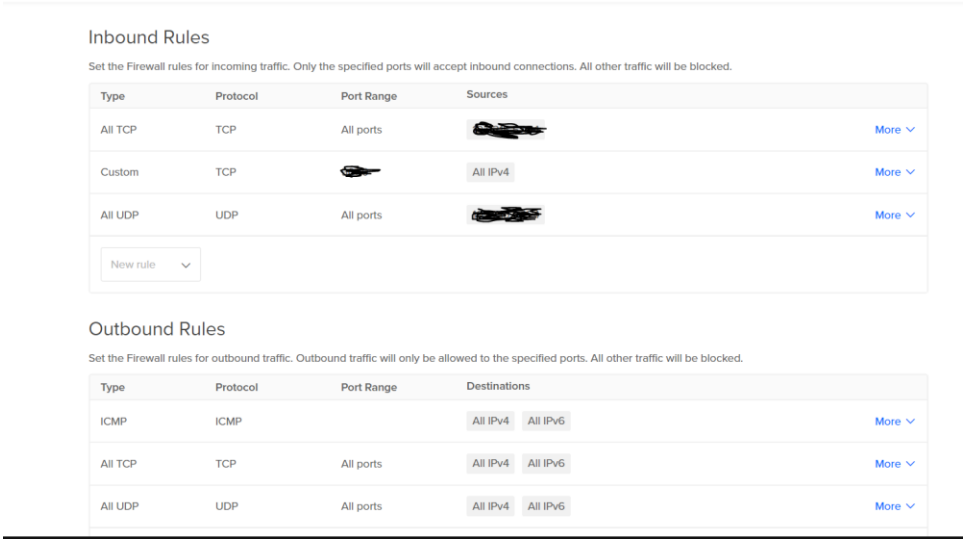
WAZUH manager server and thehive Setup: https://youtu.be/YxpUx0czgx4?t=917

I used a digitalocean(https://www.digitalocean.com/)-online cloud provider and set up 2 Ubuntu 22.04 machines: 1 for the Wazuh manager that will serve the dashboard and the other for Thehive for case management. B. Specifications:

CPU: 2 CPU Cores
RAM: 8GB+
HDD: 50GB+
OS: Ubuntu 22.04 LTS
Make sure to write down passwords for both machines.



C. Setup firewall for both machines:
   a. Click on the networking tab and set up the firewall with the rules below and assign firewall to both machines.



D. Run the droplet and run updates: apt-get update && apt-get upgrade
   a. Specifications
   b. RAM: 8GB+
   c. HDD: 50GB+
   d. OS: Ubuntu 22.04 LTS

   e. Install Wazuh 4.7
   f. curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh -a

g. Make sure to write down username and password at the end of the command.
h. Extract Wazuh Credentials
i. sudo tar -xvf wazuh-install-files.tar

E. Use https://WAZUH-Droplet-Address to access dashboard

Thehive: https://youtu.be/YxpUx0czgx4?t=1258 -Mydfir

A. Setup the droplet with the same Specifications as the Wazuh manager and use the same configured firewall
B. Open Thehive and install the prerequisites : Java, Cassandra, Elastic search and thehive; all the command are in instruction file in github: TheHive Instructions
C. When everything is installed use systemctl status to check if they are active and running.

Configuration of thehive and Wazuh manager:
https://youtu.be/VuSKMPRXN1M?t=37

Make sure to restart after each config; as this is important to apply necessary changes. Systemctl restart.

Thehive config:

A. Cassandra config: Nano into /etc/cassandra/cassandra.yaml and change cluster name to whatever you would like and change listen, RPC, and seed provider address to the public ip address of the thehive server. Remove any old files: rm -rf /var/lib/cassandra/*
B. Elasticsearch config:
    a. nano /etc/elasticsearch/elasticsearch.yml.
    b. Uncomment cluster name and put thehive.
    c. Uncomment node-name
    d. Uncomment network host and put in public ip of thehive and remember that default port is 9000
    e. Uncomment cluster.initial_master_nodes:[“node-1”]-here we can setup to scale the elasticsearch

C. Thehive
    a. Ls -la /opt/thp
    b. Chown -R thehive:thehive /opt/thp—————>> changing ownership of user/group to run the hive
    c. Nano /etc/thehive/application.conf
    d. Change database and index config ip address to the thhive public address
    e. Under service config: Application.baseurl should have thehive public address on port 9000 so you can access the dashboard.

Default Credentials on port 9000
Credentials are 'admin@thehive.local' with a password of 'secret'

ERROR: if thehive is giving errors or not starting up then its most likely elasticsearch, if that is the case then add more computing power to the droplet.
Make sure to restart after each config; as this is important to apply necessary changes. Systemctl restart.

Wazuh Config and dashboard:

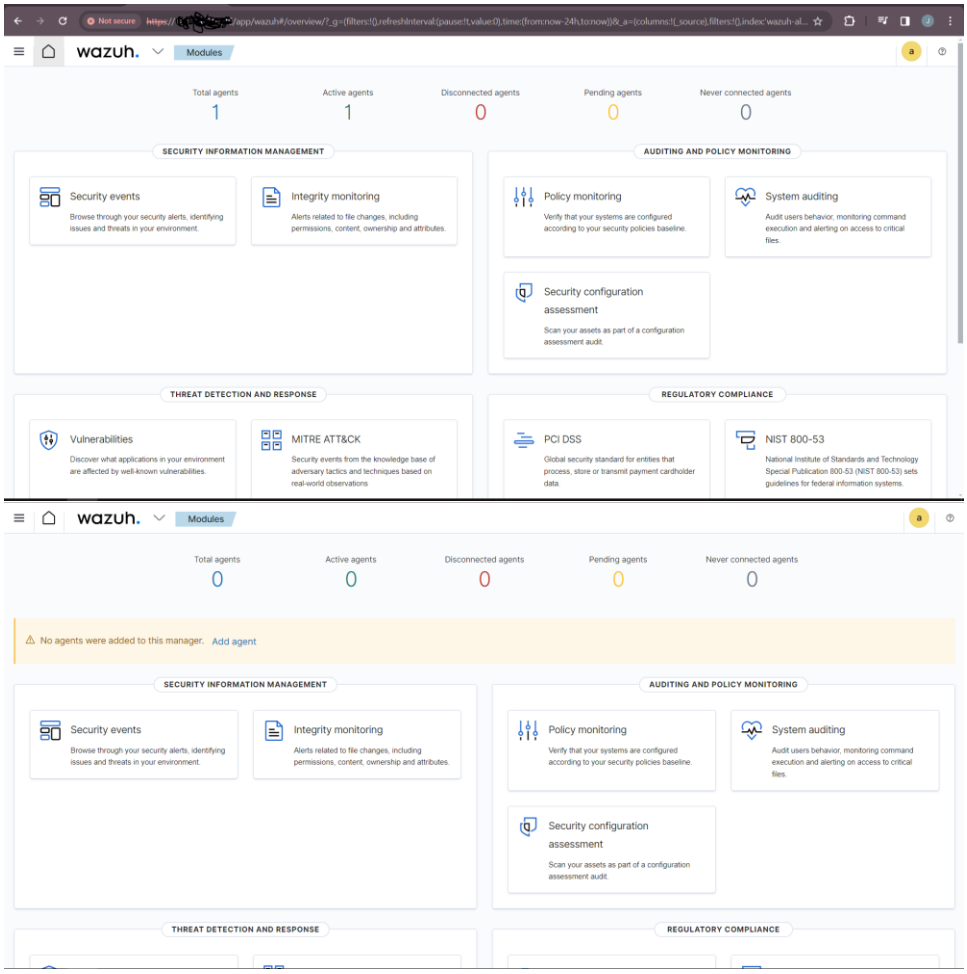
A. Extract Wazuh Credentials: sudo tar -xvf wazuh-install-files.tar
B. We need to extract wazuh-passwords.txt: the file should be in /wazuh-install-files—--------->> take a note of user API username and pass(this will be used with shuffle for automation)and the admin username and pass

C. Using the admin username and pass, the dashboard can be accessed.

Make sure to restart after each config; as this is important to apply necessary changes. Systemctl restart

Wazuh dashboard:

A. Click add agent on the dashboard and select windows 10 option that should be MSI 32/64 bits.
B. Add Wazuh public address in the server address box
C. Add name for the windows 10 agent and select default group
D. Open admin powershell and copy and run command; Start Wazuh agent
E. Double check in services if the Sysmon and Wazuh services are running(net start wazuhsvc).
F. Now there should be active agents on the dashboard:
G. End goal is to detect Mimikatz on the client machine

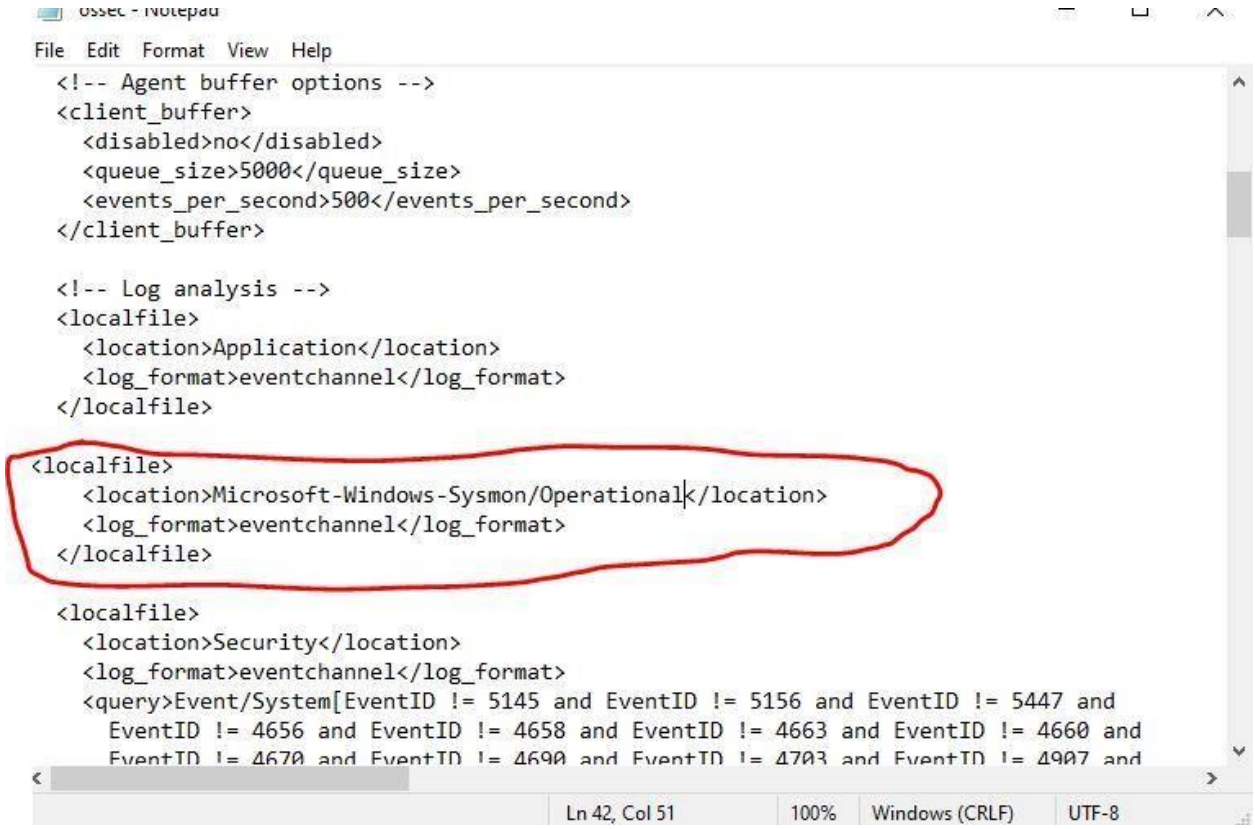


Windows 10 Telemetry:

A. This telemetry is needed for sysmon and wazuh dashboard to communicate with each other.

B. Go to local disk>program files x86 > ossec agent:---open ossec conf file using admin privileges notepad

C. Under log analysis add the red circled text in the ossec conf file(show in the pic below)



D. Save file and restart wazuh service

E. Open the wazuh dashboard and notice the agent that you named.

F. In security events tab search for sysmon and wazuh should detect the usage of sysmon.

Mimikatz detection:

Mimikatz github link–https://github.com/gentilkiwi/mimikatz/releases/tag/2.2.0-20220919

A. Unzip mimikatz zip folder and using admin privileges; run Mimikatz on the agent .\mimikatz.exe

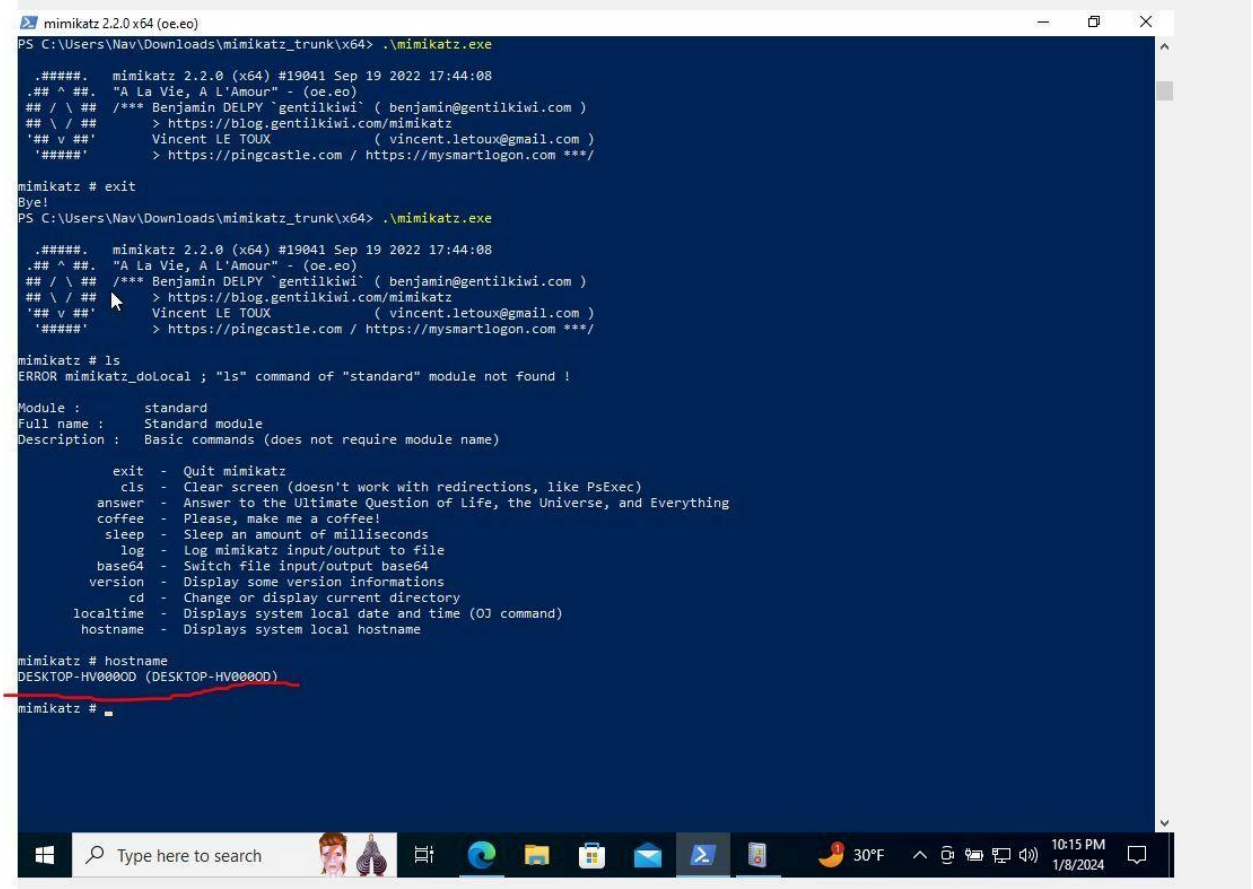
B. Return to wazuh dashboard and search for security events mimikatz.

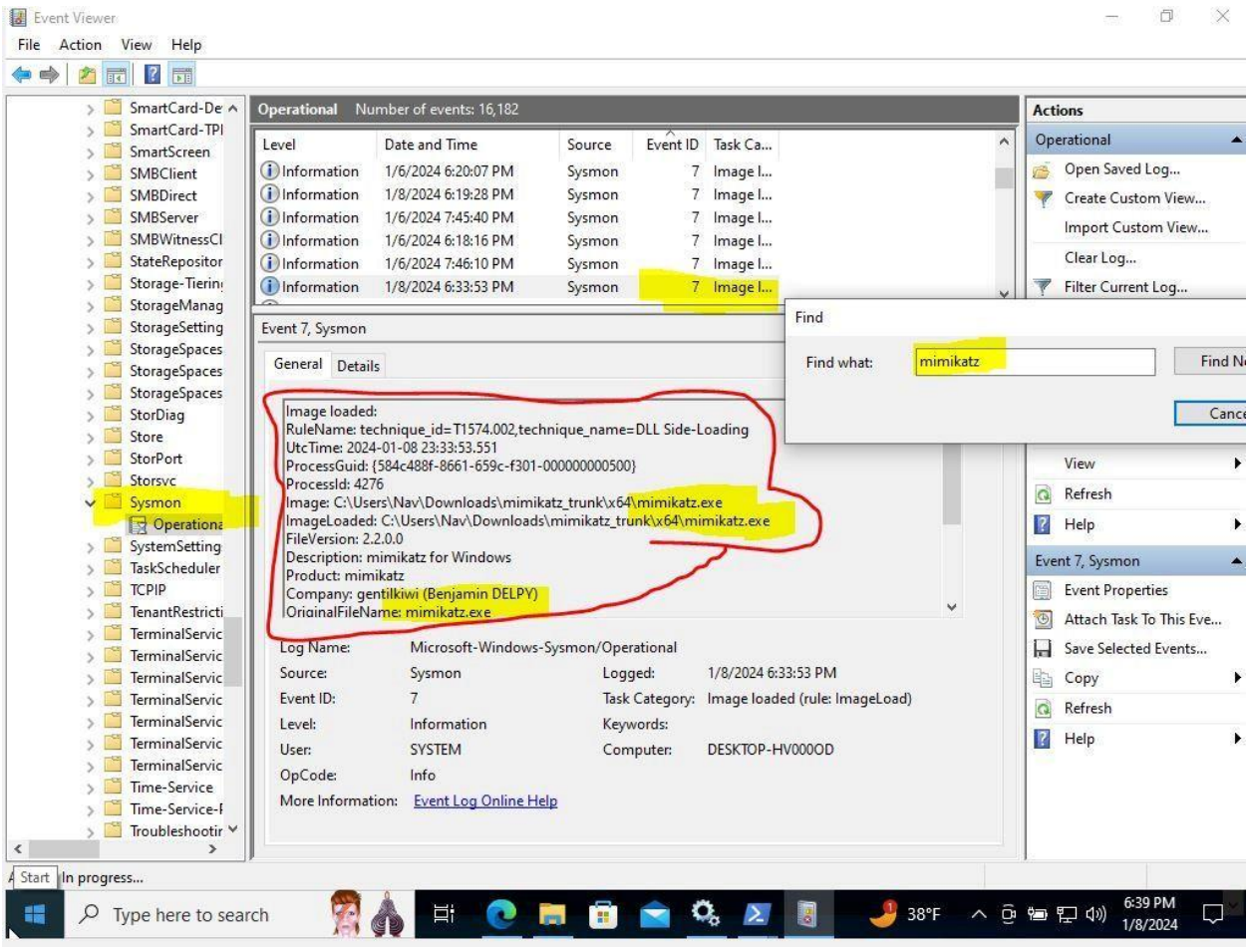
C. If mimikatz is not detected, then head over to wazuh manager and nano into /var/ossec/etc/ossec.conf and change ossec.conf setting of logall and logall_json to yes and save the file. Additionally this can be done through the dashboard using the rule editor.

D. In the directory /var/ossec/logs/archives: nano into /etc/filebeat/filebeat.yml—-and under filebeat.modules change archives enabled to true.

Make sure to restart after each config; as this is important to apply necessary changes. Systemctl restart

E. In the dashboard, under stack management click create index pattern and name it wazuh-archives-* and click next and select timestamp, and finally create an index pattern.
Screenshots below:


mimikatz 2.2.0 x64 (oe.eo)
PS C:\Users\Nav\Downloads\mimikatz_trunk\x64> .\mimikatz.exe
.#####. mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
/ \ ## /*** Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
\ / ## > https://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX (vincent.letoux@gmail.com)
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/
mimikatz # exit
Bye!
PS C:\Users\Nav\Downloads\mimikatz_trunk\x64> .\mimikatz.exe
.#####. mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
/ \ ## /*** Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
\ / ## > https://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX (vincent.letoux@gmail.com)
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/
mimikatz # ls
ERROR mimikatz_doLocal ; "ls" command of "standard" module not found !
Module : standard
Full name : Standard module
Description : Basic commands (does not require module name)
exit - Quit mimikatz
cls - Clear screen (doesn't work with redirections, like PsExec)
answer - Answer to the Ultimate Question of Life, the Universe, and Everything
coffee - Please, make me a coffee!
sleep - Sleep an amount of milliseconds
log - Log mimikatz input/output to file
base64 - Switch file input/output base64
version - Display some version informations
cd - Change or display current directory
localtime - Displays system local date and time (OJ command)
hostname - Displays system local hostname
mimikatz # hostname
DESKTOP-HV000OD (DESKTOP-HV000OD)
mimikatz #
Type here to search
30°F
10:15 PM
1/8/2024


Event Viewer
File Action View Help
Operational Number of events: 16,182
Level Date and Time Source Event ID Task Ca...
Information 1/6/2024 6:20:07 PM Sysmon 7 Image l...
Information 1/8/2024 6:19:28 PM Sysmon 7 Image l...
Information 1/6/2024 7:45:40 PM Sysmon 7 Image l...
Information 1/6/2024 6:18:16 PM Sysmon 7 Image l...
Information 1/6/2024 7:46:10 PM Sysmon 7 Image l...
Information 1/8/2024 6:33:53 PM Sysmon 7 Image l...
Event 7, Sysmon
General Details
Image loaded:
RuleName: technique_id=T1574.002,technique_name=DLL Side-Loading
UtcTime: 2024-01-08 23:33:53.551
ProcessGuid: {584c488f-8661-659c-f301-000000000500}
ProcessId: 4276
Image: C:\Users\Nav\Downloads\mimikatz_trunk\x64\mimikatz.exe
ImageLoaded: C:\Users\Nav\Downloads\mimikatz_trunk\x64\mimikatz.exe
FileVersion: 2.2.0.0
Description: mimikatz for Windows
Product: mimikatz
Company: gentilkiwi (Benjamin DELPY)
OriginalFileName: mimikatz.exe
Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon Logged: 1/8/2024 6:33:53 PM
Event ID: 7 Task Category: Image loaded (rule: ImageLoad)
Level: Information Keywords:
User: SYSTEM Computer: DESKTOP-HV000OD
OpCode: Info
More Information: Event Log Online Help
Find
Find what: mimikatz
Actions
Operational
Open Saved Log...
Create Custom View...
Import Custom View...
Clear Log...
Filter Current Log...
View
Refresh
Help
Event 7, Sysmon
Event Properties
Attach Task To This Eve...
Save Selected Events...
Copy
Refresh
Help
Start In progress...
Type here to search
38°F
6:39 PM
1/8/2024

›	Mar 21, 2024 @ 13:38:28.348	013	Mr-Fynnesse	T1003	Credential Access	Mimikatz Usage Detected	15	100002

Rule Creation for detecting sysmon event ID 1:

A. Head over to wazuh management on the dashboard and click rules and then custom rules.

B. Edit local_rules.xml and type in exactly the rule that is below in the screenshot.

```
<!-- Local rules -->

<!-- Modify it at your will. -->
<!-- Copyright (C) 2015, Wazuh Inc. -->

<!-- Example -->
<group name="local,syslog,sshd,">

  <!--
  Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
  -->
  <rule id="100001" level="5">
    <if_sid>5716</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>sshd: authentication failed from IP 1.1.1.1.</description>
    <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
  </rule>

  <rule id="100002" level="15">
    <if_group>sysmon_event1</if_group>
    <field name="win.eventdata.originalFileName" type="pcre2">(?i)mimikatz\.exe</field>
    <description>Mimikatz Usage Detected</description>
    <mitre>
      <id>T1003</id>
    </mitre>
  </rule>
```

C. There are more rules if you just google search or github search or you are able to configure your own rules. So with this rule creation, even if the attacker changes the file name, we will still know based on event ID and sysmon logs. D. Restart and run mimikatz again and test the rule.

SOAR(Security orchestration, automation and response)and Shuffle automation setup:
Objective: Shuffle website–https://shuffler.io/

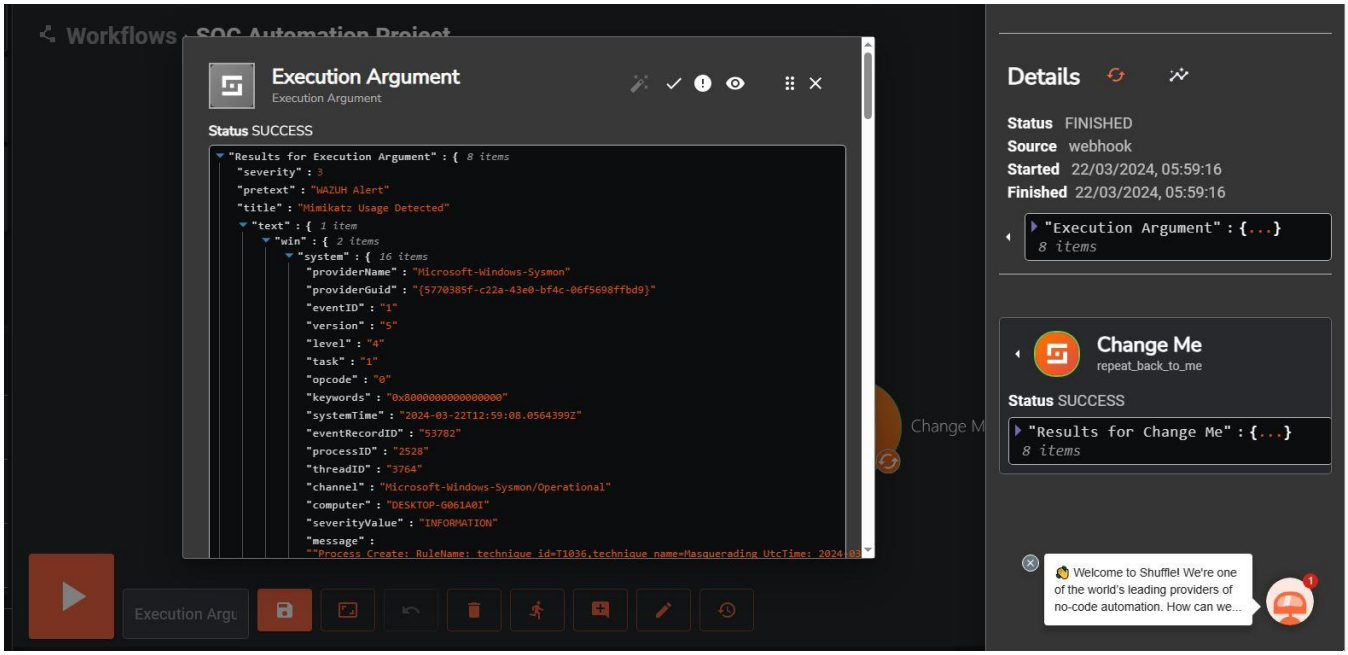
- Connect SOAR(Shuffle)
- Send alert to thehive
- Send alert via email to SOC analyst

**** I have summarized my instructions for this portion of the lab because there are many little bits and pieces that need to connect together and activate all the capabilities. It is better to watch mydfir's videos for a more perfect guide- https://www.youtube.com/watch?v=GNXK00QapjQ****

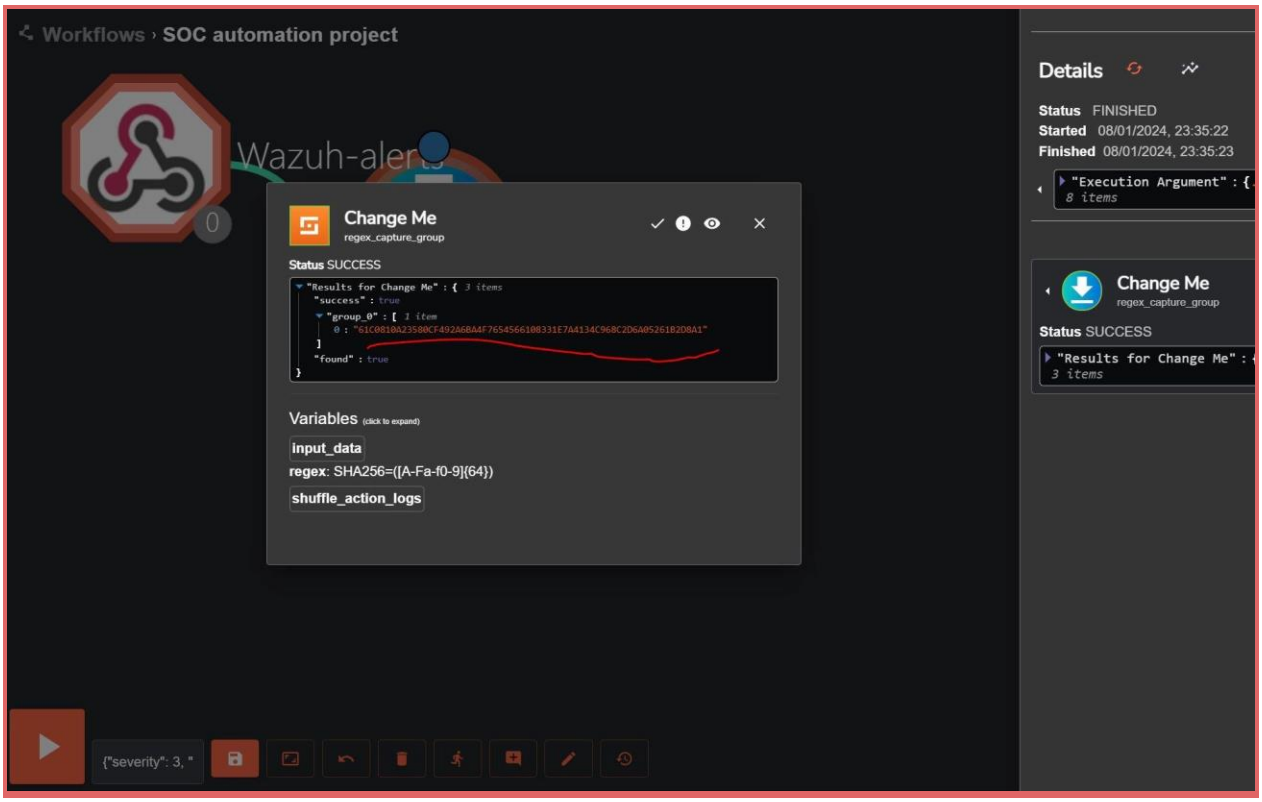
A. Create a new workflow and add all apps that are show in my workflow pic below

a. Wazuh alerts(requires api or authentication)
b. Get API-uses Wazuh api(requires api or authentication)
c. Wazuh- active response(requires api or authentication)
d. User input
e. Sha-256 Regex
f. Email
g. TheHive(requires api or authentication)
h. Virustotal-requires authentication or API key Workflow:

1. Mimikatz alert to shuffle via wazuh alerts
2. Shuffle will extract sha 256 hash
3. Virustotal will be used to check the reputation score
4. Shuffle will send details from wazuh alert and virustotal to TheHive to open a new case and insert all details
5. Shuffle will send email to SOC analyst to alert the analyst and also ask for response to block based on event ID.

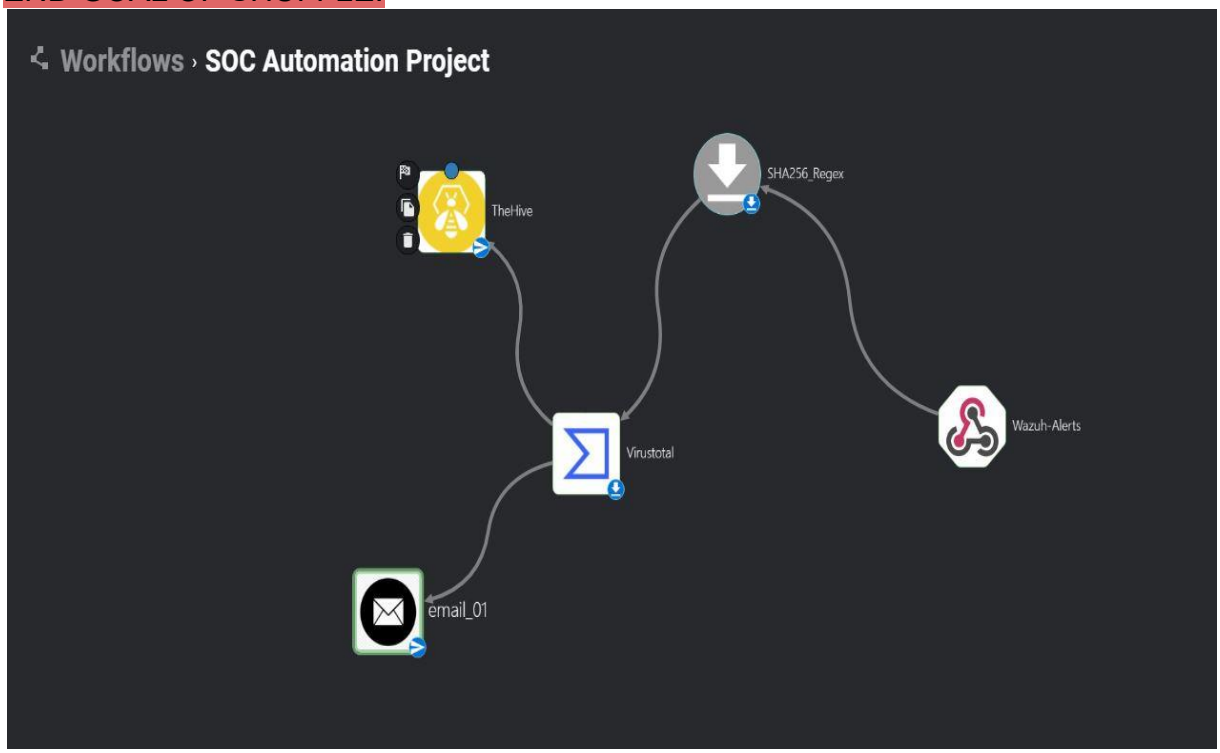


This screenshot shows alert on wazuh alert on shuffle ↑↑



This screenshot shows Extracted sha-256 hash which later can be add to virustotal for reputation score↑↑↑↑

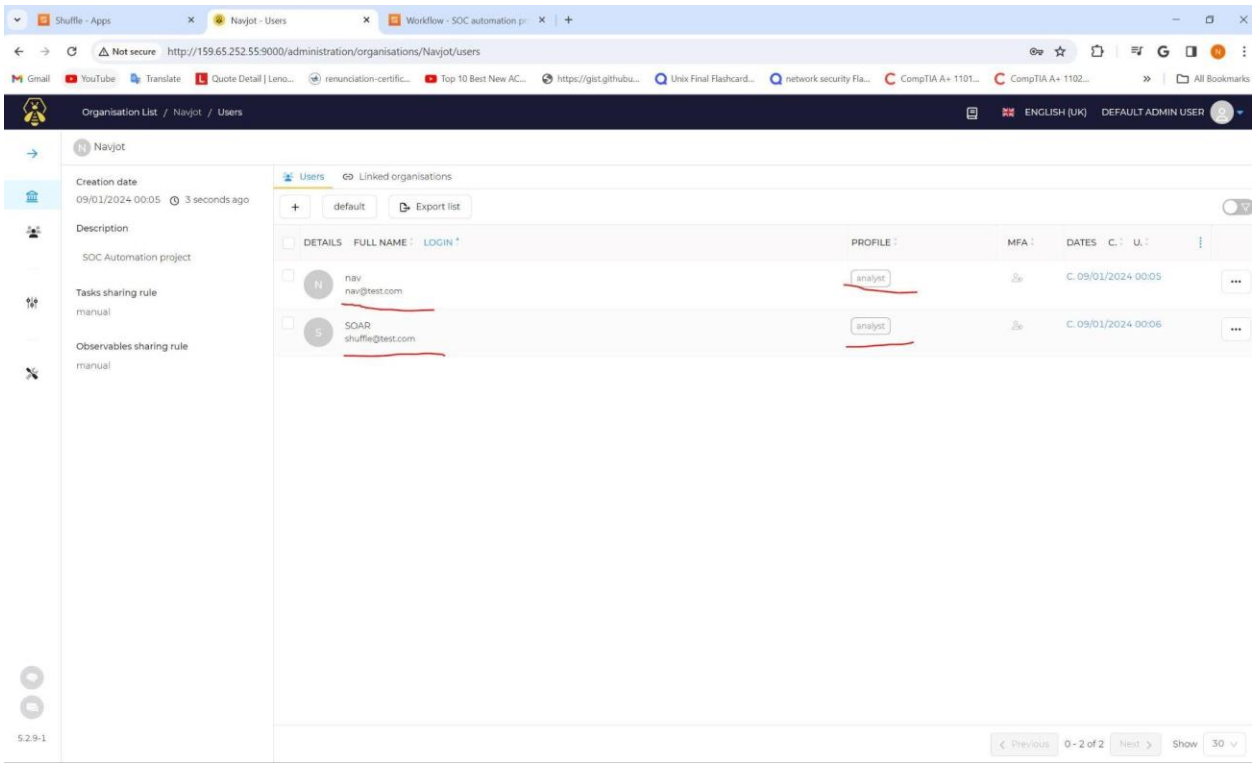
END GOAL OF SHUFFLE:

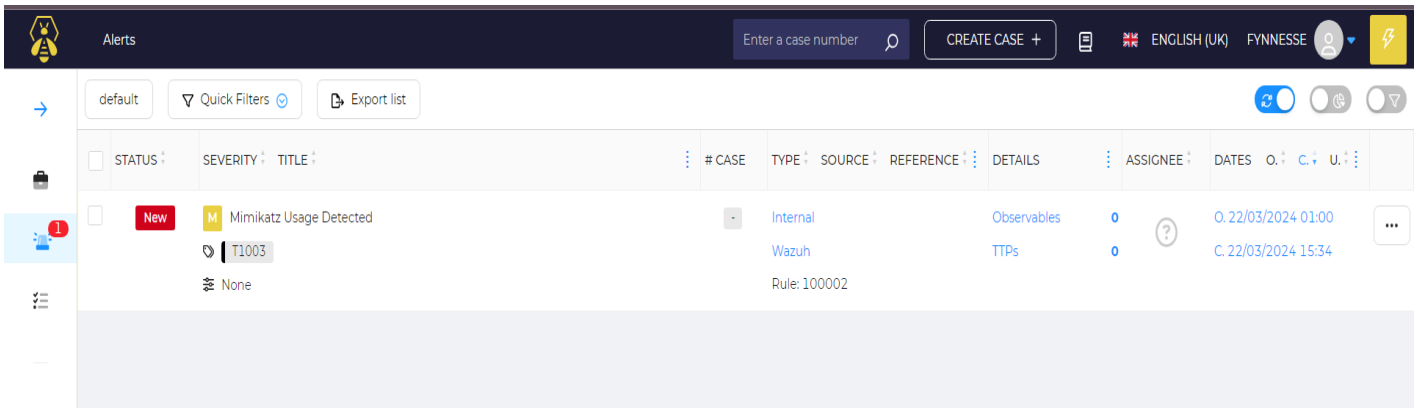


TheHive:

Log into thehive using: admin@thehive.local and password-secret

A. Create and name organizations, select normal type at the top and add login information like for me I added nav@test.com and select analyst profile
B. Add another: Create and name SOAR organizations, select service type at the top and add login information like for me I added shuffle@test.com and select analyst profile—Make sure that this account is following the principle of least privilege such as read-only.
C. Click on the soar account and add a password, make to save this or you can also create an API key and add that on thehive in shuffle.


Shuffle - Apps
Navjot - Users
Workflow - SOC automation pr
Not secure
http://159.65.252.55:9000/administration/organisations/Navjot/users
Organisation List / Navjot / Users
ENGLISH (UK)
DEFAULT ADMIN USER
Navjot
Creation date
09/01/2024 00:05 3 seconds ago
Description
SOC Automation project
Tasks sharing rule
manual
Observables sharing rule
manual
Users
Linked organisations
default
Export list
DETAILS
FULL NAME
LOGIN
PROFILE
MFA
DATES C. U.
nav
nav@test.com
analyst
C. 09/01/2024 00:05
SOAR
shuffle@test.com
analyst
C. 09/01/2024 00:06
Previous
0 - 2 of 2
Next
Show
30
5.2.9-1



Email: For this setup I used Square X which is a free browser extension that can generate email and has an inbox.

A. All you need to do is add the Square X email and add subject and body.

B. User Input will send a separate email for response that will go all the way back to window 10 client through shuffle.

Throughout this project, I learned alot about SOC workflows, tool integrations, and the importance of systematic configurations. It was a hands-on journey into the world of security incident handling, and I'm pretty proud of the setup. Many humble thanks to Mydfir(https://www.youtube.com/@MyDFIR) for setting a good example of the project. –Thank you so much

If you have any questions or need more details on a specific aspect, feel free to ask! 😄