

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ź D I N

Fran Sviličić

Automatizacija procesa u SOC-u

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE

Sadržaj

1. Uvod	1
2. Metode i tehnike rada	2
3. Alati	3
3.1 Sysmon	4
3.2 Firewall.....	5
3.3 Wazuh.....	6
3.3.1 Instalacija wazuha	6
3.3.2 Wazuh agent	6
3.4 TheHive	7
3.4.1 Konfiguracija TheHive.....	9
3.5 Konfiguracija windows 10 virtualne mašine	12
3.6 Instalacija Mimikatz	13
3.7 Konfiguracija logova za Wazuh	13
3.8 Kreiranje pravila	15
3.9 Shuffle	17
3.9.1 Dodavanje webhook okidača.....	17
3.9.2 Provjera reputation score sa virustotal	17
3.9.3 Slanje alerta na TheHive	19
4. Zaključak	21
5. Popis slika.....	22

1. Uvod

Tema ovog rada je implementacija sustava za automatizaciju procesa u Security Operations Centru (SOC) koristeći alate Wazuh, TheHive i Shuffle. Cilj ovog rada je detaljno opisati korake povezivanja ovih alata putem clouda te objasniti kako se kroz njih realizira obrada sigurnosnih događaja.

Sustav je postavljen na platformi Digital Ocean, gdje su kreirane virtualne mašine za alate i simulaciju napada. Projekt je zamišljen kao primjer modernog pristupa upravljanju sigurnosnim incidentima korištenjem open-source tehnologija.

2. Metode i tehnike rada

Rad uključuje:

1. Postavljanje okruženja:

- Korištenje platforme Digital Ocean za kreiranje virtualnih mašina.
- Instalacija alata Wazuh za praćenje sigurnosnih događaja.
- Postavljanje TheHive platforme za upravljanje incidentima.
- Integracija Shuffle alata za automatizaciju.

2. Simulacija napada:

- Instalacija alata poput Mimikatz i Sysmon na virtualne mašine radi simulacije sigurnosnih prijetnji.
- Generiranje sigurnosnih događaja koje detektira Wazuh.

3. Automatizacija procesa:

- Konfiguracija Shuffle alata za prosljeđivanje detektiranih događaja prema TheHive platformi.
- Slanje obavijesti putem e-maila sigurnosnim analitičarima.

4. Analiza i evaluacija:

- Provjera reakcija sustava na simulirane napade.
- Praćenje funkcionalnosti svakog alata u sustavu.

3. Alati

Sustav je podijeljen na tri glavna segmenta:

1. Detekcija:

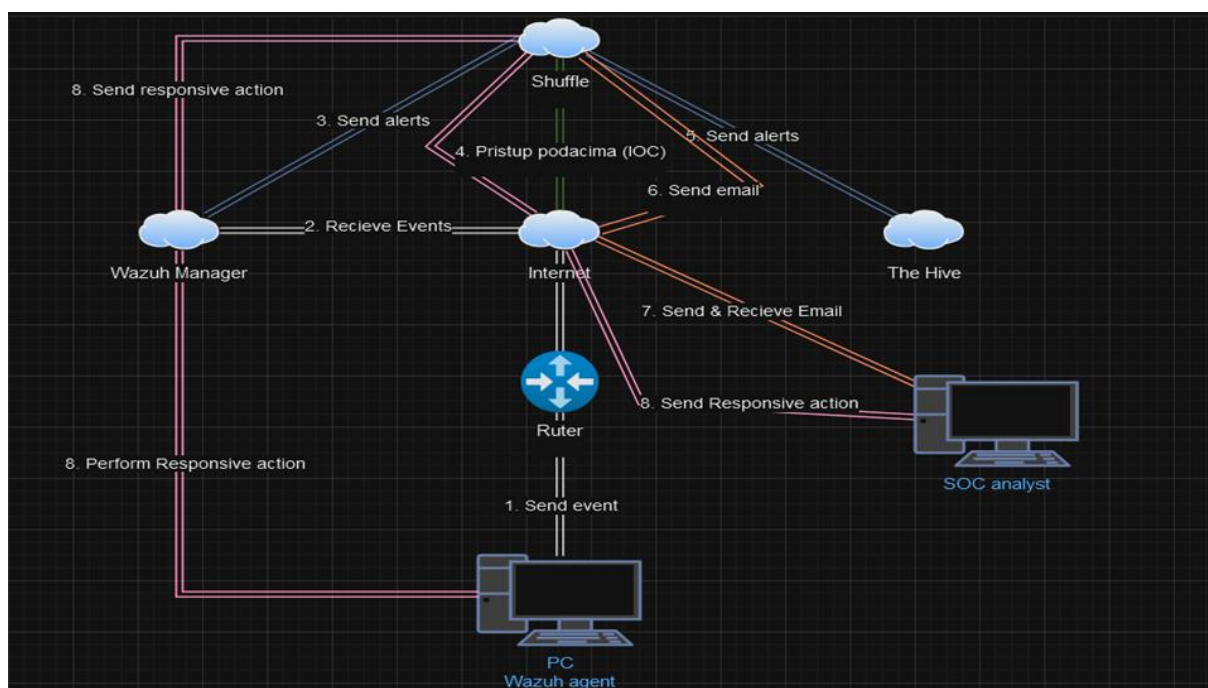
- Wazuh prati događaje na virtualnoj mašini te generira alarme u slučaju otkrivanja prijetnji.

2. Automatizacija:

- Shuffle preuzima alarme iz Wazuha i proslijeđuje ih prema TheHive platformi te putem e-maila obavještava analitičare.

3. Upravljanje incidentima:

- TheHive omogućuje sigurnosnim analitičarima detaljan pregled i obradu prijavljenih incidenata.



Slika 1: Vizualizacija sustava

Slika prikazuje tijek informacija kroz sustav:

1. **PC (Wazuh agent):** prikuplja podatke o aktivnostima i šalje događaje prema Wazuh Manageru.
2. **Wazuh Manager:** prima događaje i generira alarme u slučaju prijetnji.

3. **Shuffle:** automatizira proces slanjem alarma prema TheHive-u i e-mail obavijestima analitičarima.
4. **TheHive:** omogućuje upravljanje incidentima na temelju alarma.
5. **SOC analitičar:** prima informacije i poduzima odgovarajuće akcije.
6. **Povratne akcije:** akcije koje analitičar definira vraćaju se kroz sustav prema izvoru problema.

3.1 Sysmon

Prije instalacije wazuha, na virtualnu mašinu instaliran je sysmon. Sysmon je koristan alat za detaljno praćenje aktivnosti na sustavu, poput izvršavanja procesa, mrežne komunikacije i pristupa datotekama. Instalirali smo sysmon jer omogućuje generiranje podataka o potencijalno zlonamjernim aktivnostima, što je bitno za testiranje i validaciju funkcionalnosti wazuha.

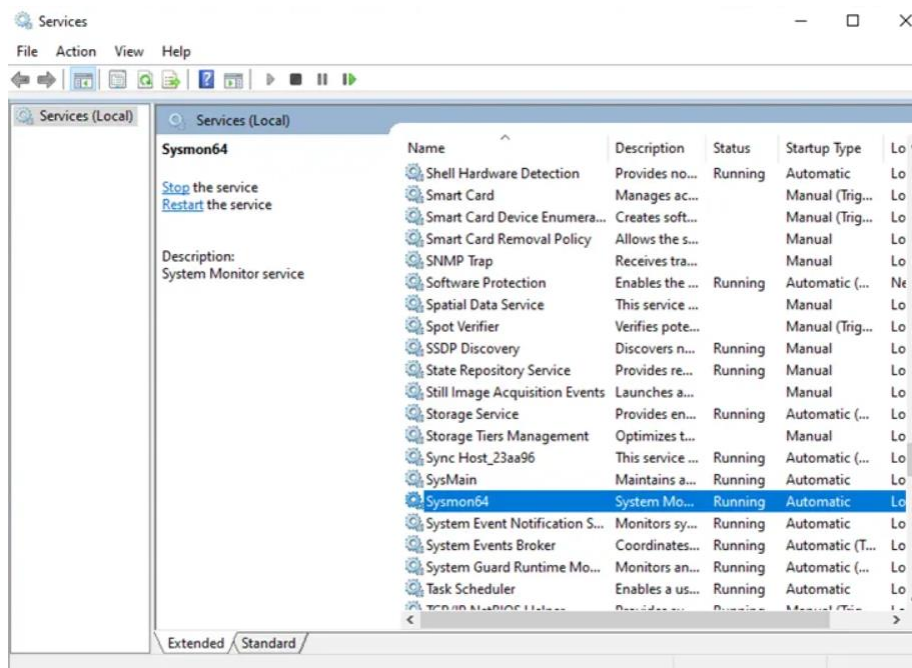
```
PS C:\Windows\system32> cd ..\..\Users\Windows10VM\Downloads\Sysmon\
PS C:\Users\Windows10VM\Downloads\Sysmon> ls

Directory: C:\Users\Windows10VM\Downloads\Sysmon

Mode                LastWriteTime         Length Name
----                -
-a----          20/05/2024   13:53             7490 Eula.txt
-a----          20/05/2024   13:53          8447792 Sysmon.exe
-a----          20/05/2024   13:53          4545344 Sysmon64.exe
-a----          20/05/2024   13:53          4999984 Sysmon64a.exe
-a----          20/05/2024   13:53           253169 sysmonconfig.xml

PS C:\Users\Windows10VM\Downloads\Sysmon> .\Sysmon64.exe.
PS C:\Users\Windows10VM\Downloads\Sysmon> .\Sysmon64.exe. -i .\sysmonconfig.xml
```

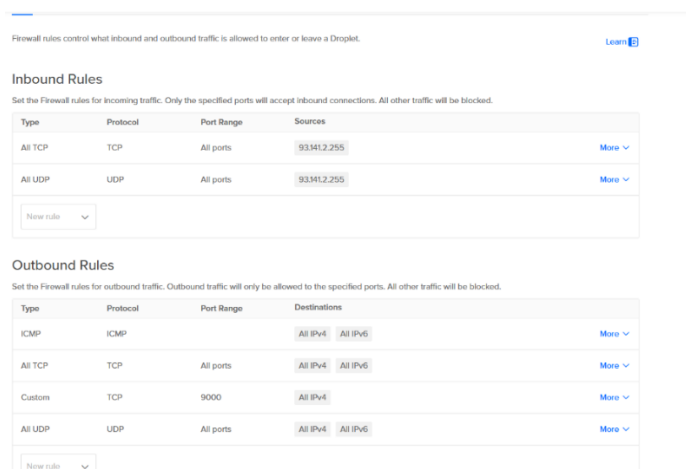
Slika 2: Sysmon instalacija



Slika 3: Status sysmona u prozoru „Services“

3.2 Firewall

Prije instalacije wazuha i TheHive-a konfiguriran je firewall kako bi se blokirao promet od vanjskih skenera. Inbound rules znači dopušten je dolazni promet samo s odabranih izvora (moja IP adresa) za sve TCP i UDP portove. Outbound rules znači dopušten je odlazni promet prema svim IPv4 i IPv6 destinacijama, uključujući specifične TCP portove, poput porta 9000 koji će bit potreban kasnije za TheHive.



Slika 4: Firewall

3.3 Wazuh

3.3.1 Instalacija wazuha

Instalacija je provedena prema dokumentaciji alata, u konzoli od DigitalOcean-a izvršena je naredba za instalaciju:

```
root@wazuh:/# curl -s0 https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh -a_
```

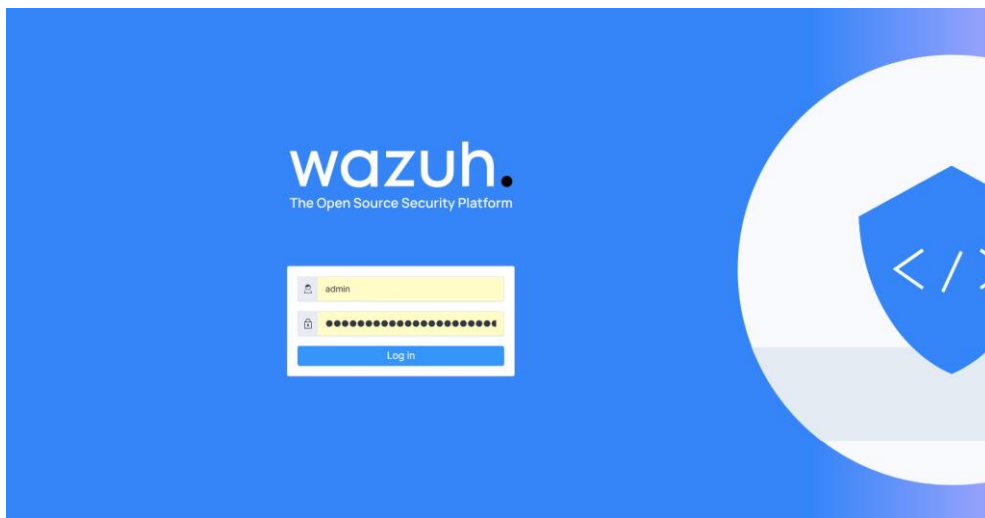
Slika 5: Naredba za instalaciju Wazuh

I naredba za uzimanje korisničkih podataka:

```
root@wazuh:/# sudo tar -xvf wazuh-install-files.tar
```

Slika 6: Naredba za uzimanje korisničkih podataka

Nakon toga dobijemo korisničke podatke i pristupamo wazuh-u tako da upišemo njegovu ipv4 adresu u preglednik.



Slika 7: Wazuh interface

3.3.2 Wazuh agent

Wazuh agent mora biti instaliran na našem krajnjem uređaju kako bi se logovi prebacili na naš poslužitelj. Pomoću dolje navedene powershell naredbe možemo instalirati uslugu na krajnjem uređaju sa sustavom Windows 10.


```
PS C:\Windows\system32> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.7.3-1.msi -Outfile $
{env.tmp}\wazuh-agent; msixexec.exe /i ${env.tmp}\wazuh-agent /q WAZUH_MANAGER='Your IP address here' WAZUH_AGENT_NAME='Y
our cluster name here' WAZUH_REGISTRATION_SERVER='Your ip address here'
>>
>> net start wazuhsvc
```

Slika 8: Instalacija Wazuh agenta

3.4 TheHive

Na isti način kreiramo na DigitalOcean virtual machine za TheHive i povezujemo ga sa prijašnje kreiranim firewallom. Za instalaciju koristili smo naredbe iz njihove dokumentacije i upisali ih u konzolu. Za TheHive moramo izračunati 4 komponente: Java, Cassandra, ElasticSearch i TheHive

- Java

```
wget -q0- https://apt.corretto.aws/corretto.key | sudo gpg --dearmor -o /usr/share/keyrings/corretto.gpg
echo "deb [signed-by=/usr/share/keyrings/corretto.gpg] https://apt.corretto.aws stable main" | sudo tee -a /etc
sudo apt update
sudo apt install java-common java-11-amazon-corretto-jdk
echo JAVA_HOME="/usr/lib/jvm/java-11-amazon-corretto" | sudo tee -a /etc/environment
export JAVA_HOME="/usr/lib/jvm/java-11-amazon-corretto"
```

Slika 9: Instalacija Java

- Cassandra

1. Add Apache Cassandra repository references

- Download Apache Cassandra repository keys using the following command:

```
wget -q0 - https://downloads.apache.org/cassandra/KEYS | sudo gpg --dearmor -o /usr/share/keyrings/cassandra-
```

- Add the repository to your system by appending the following line to the `/etc/apt/sources.list.d/cassandra.sources.list` file. This file may not exist, and you may need to create it.

```
echo "deb [signed-by=/usr/share/keyrings/cassandra-archive.gpg] https://deb.debian.org/debian/cassandra-40x main" |
```

2. Install the package

- Once the repository references are added, update your package index and install Cassandra using the following commands:

```
sudo apt update
sudo apt install cassandra
```

Slika 10: Instalacija Cassandra

- ElasticSearch

1. Add Elasticsearch repository references

- To add Elasticsearch repository keys, execute the following command:

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elas  
sudo apt-get install apt-transport-https
```

- Add the repository to your system by appending the following line to the `/etc/apt/sources.list.d/elastic-7.x.list` file. This file may not exist, and you may need to create it

```
echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/7.x/a
```

2. Install the package

- Once the repository references are added, update your package index and install Elasticsearch using the following commands:

```
sudo apt update  
sudo apt install elasticsearch
```

Slika 11: Instalacija ElasticSearcha

- TheHive

For Debian systems, use the following commands:

```
wget -O- https://raw.githubusercontent.com/StrangeBeeCorp/Security/main/PGP%20keys/packages.key | sudo gpg --dearmor
```

Install TheHive package by using the following commands:

DEB RPM Other Installation Methods

```
echo 'deb [arch=all signed-by=/usr/share/keyrings/strangebee-archive-keyring.gpg] https://deb.strangebee.com thehive  
sudo apt-get update  
sudo apt-get install -y thehive
```

Slika 12: Instalacija TheHivea

3.4.1 Konfiguracija TheHive

U .yaml datoteci za Cassandra potrebno je konfigurirati seeds, listen address i RPC address kako bi odražavali TheHive droplet, odnosno mijenjamo default postavke sa ipv4 adresom TheHive-a, yaml datoteci pristupamo sa naredbom nano/etc/cassandra/cassandra.yaml i dobijemo sljedeći prikaz:

```
# any class that implements the SeedProvider interface and has a
# constructor that takes a Map<String, String> of parameters will do.
seed_provider:
  # Addresses of hosts that are deemed contact points.
  # Cassandra nodes use this list of hosts to find each other and learn
  # the topology of the ring. You must change this if you are running
  # multiple nodes!
  - class_name: org.apache.cassandra.locator.SimpleSeedProvider
    parameters:
      # seeds is actually a comma-delimited list of addresses.
      # Ex: "<ip1>,<ip2>,<ip3>"
      - seeds: "167.99.241.153"

# For workloads with more data than can fit in memory, Cassandra's
# bottleneck will be reads that need to fetch data from
# disk. "concurrent_reads" should be set to (16 * number_of_drives) in
# order to allow the operations to enqueue low enough in the stack
# that the OS and drives can reorder them. Same applies to
# "concurrent_counter_writes", since counter writes read the current
# values before incrementing and writing them back.
#
```

Slika 13: .yaml file (Cassandra)

```
# It will fall back to InetAddress.getLoopbackAddress(), which is wrong for producti
#
# Setting listen_address to 0.0.0.0 is always wrong.
#
listen_address: 167.99.241.153

# Set listen_address OR listen_interface, not both. Interfaces must correspond
# to a single address, IP aliasing is not supported.
# listen_interface: eth0
```

Slika 14: .yaml file (Cassandra)

```
#
# Note that unlike listen_address, you can specify 0.0.0.0, but you must also
# set broadcast_rpc_address to a value other than 0.0.0.0.
#
# For security reasons, you should not expose this port to the internet. Firewall it if needed.
rpc_address: 167.99.241.153

# Set rpc_address OR rpc_interface, not both. Interfaces must correspond
# to a single address, IP aliasing is not supported.
# rpc_interface: eth1

# If you choose to specify the interface by name and the interface has an ipv4 and an ipv6 address
# you can specify which should be chosen using rpc_interface_prefer_ipv6. If false the first ipv4
# address will be used. If true the first ipv6 address will be used. Defaults to false preferring
# ipv4. If there is only one address it will be selected regardless of ipv4/ipv6.
# rpc_interface_prefer_ipv6: false
```

Slika 15: .yaml file (Cassandra)

Kada završimo sa konfiguracijom pokrećemo cassandru sa naredbom `systemctl start cassandra`. Tako pokrećemo i ostale komponente samo mijenjamo naziv ovisno o komponenti. Sljedeće konfiguriramo yaml datoteku od ElasticSearch, gdje mijenjamo cluster name i network host, ulazimo sa naredbom `nano/etc/elasticsearch/elasticsearch.yaml`:

```
#
# Use a descriptive name for your cluster:
#
cluster.name: thehive
#
# ----- Node -----
#
# Use a descriptive name for the node:
#
node.name: node-1
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
```

Slika 16: .yaml file (ElasticSearch)

```
# address here to expose this node on the network:
#
network.host: 167.99.241.153
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# ----- Discovery -----
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "::1"]
#
discovery.seed_hosts: ["host1", "host2"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
cluster.initial_master_nodes: ["node-1"]
#
# For more information, consult the discovery and cluster formation module documentation.
#
```

Slika 17: .yaml file (ElasticSearch)

Sada se moramo pobrinuti da TheHive user ima pristup određenim putanjama do datotekama, ovako izgleda trenutna putanja:

```
root@thehive:~# ls -la /opt/thp
total 12
drwxr-xr-x 3 root root 4096 Dec 10 20:31 .
drwxr-xr-x 5 root root 4096 Dec 10 20:31 ..
drwxr-xr-x 5 root root 4096 Dec 10 20:31 thehive
```

Slika 18: Putanja

sa naredbom `chown -R thehive:thehive /opt/thp` možemo promjeniti vlasnika TheHive korisnika i putanja bi onda trebala ovako izgledati:

```
root@thehive:~# ls -la /opt/thp
total 12
drwxr-xr-x 3 thehive thehive 4096 Jan  8 16:14 .
drwxr-xr-x 5 root    root    4096 Jan  8 16:14 ..
drwxr-xr-x 5 thehive thehive 4096 Jan  8 16:14 thehive
```

Slika 19: Promjena vlasnika

Sada krećemo sa konfiguracijom samog TheHive-a. Datoteci pristupamo naredbom `nano /etc/thehive/application.conf` i mijenjamo database index, storage location on the server, the server IP i file ownership.

```
# Database and index configuration
# By default, TheHive is configured to connect to local Cassandra 4.x and a
# local Elasticsearch services without authentication.
db.janusgraph {
  storage {
    backend = cql
    hostname = ["167.99.241.153"]
    # Cassandra authentication (if configured)
    # username = "thehive"
    # password = "password"
    cql {
      cluster-name = Test Cluster
      keyspace = thehive
    }
  }
  index.search {
    backend = elasticsearch
    hostname = ["167.99.241.153"]
    index-name = thehive
  }
}
```

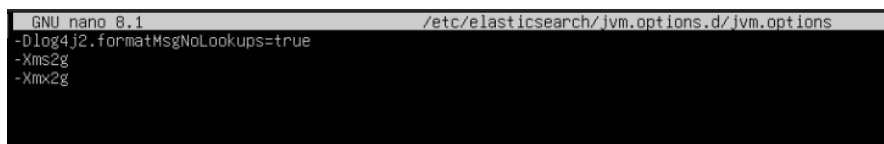
Slika 20: Konfiguracija TheHivea

```
# Service configuration
application.baseUrl = "http://167.99.241.153:9000"
play.http.context = "/"

# Additional modules
#
# TheHive is strongly integrated with Cortex and MISP.
# Both modules are enabled by default. If not used, each one can be disabled by
# commenting the configuration line.
scalligraph.modules += org.thp.thehive.connector.cortex.CortexModule
scalligraph.modules += org.thp.thehive.connector.misp.MispModule
```

Slika 21: Konfiguracija TheHivea

Sada bi trebali moći pokrenuti TheHive sa korisničkim podacima: username: admin@thehive.local, password: secret, ali javlja se greška prilikom prijave. Uzrok greške je Elasticsearch koji je iz nekog razloga prestao raditi, kako bi popravili ovu grešku moramo napraviti jvm custom file. Pristupamo novoj datoteci sa naredbom nano /etc/elasticsearch/jvm.options.d/jvm.options i unutar te datoteke upisujemo sljedeći sadržaj



```
GNU nano 8.1 /etc/elasticsearch/jvm.options.d/jvm.options
-Dlog4j2.formatMsgNoLookups=true
-Xms2g
-Xmx2g
```

Slika 22: Potrošnja memorije

To smo upisali kako bi javi ograničili potrošnju memorije na 2 gigabajta pošto naš virtual machine ima 8 gigabajta morali smo smanjiti potrošnju memorije.

3.5 Konfiguracija windows 10 virtualne mašine

Na virtualnoj mašini otvoren je ossec tekstualna datoteka za prilagodbu postavki kojoj se može pristupiti kada se instalira wazuh menager. Prva stavka konfiguracije uključivala je definiranje pravila za detekciju aktivnosti alata mimikatz putem sysmona. Dodana su specifična pravila kako bi se generirali alarmi kada sysmon detektira izvršenje procesa povezanih s mimikatzom. Kada upisujemo <location></location> moramo upisati puno ime sysmona koje se nalazi u event vieweru.



```
ossec-agent
File Edit Format View Help
<manager_address>159.203.17.</manager_address>
<agent_name>mydfir</agent_name>
</enrollment>
</client>

<!-- Agent buffer options -->
<client_buffer>
  <disabled>no</disabled>
  <queue_size>500</queue_size>
  <events_per_second>500</events_per_second>
</client_buffer>

<!-- Log analysis -->
<localfile>
  <location>Application</location>
  <log_format>eventchannel</log_format>
</localfile>

<localfile>
  <location>Microsoft-Windows-Sysmon/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>

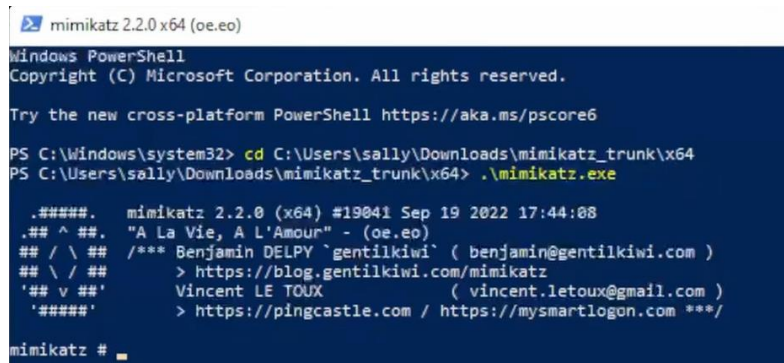
<localfile>
  <location>Security</location>
  <log_format>eventchannel</log_format>
  <query>Event/System[EventID != 5145 and EventID != 5156 and EventID != 5447 and
    EventID != 4656 and EventID != 4658 and EventID != 4663 and EventID != 4668 and
    EventID != 4670 and EventID != 4690 and EventID != 4703 and EventID != 4907 and
    EventID != 5152 and EventID != 5157]</query>
</localfile>

<localfile>
```

Slika 23: Konfiguracija ossec

3.6 Instalacija Mimikatz

Za instalaciju mimikatz moramo onemogućiti virus & threat protection i unutar njega stisnuti na add an exclusion i odabrati downloads folder kako ne bi reagiralo kada instaliramo mimikatz. Kada imamo instaliran mimikatz u downloads folder otvorimo powershell i promijenimo direktorij sa putanjom do mimikatz, kada dođemo do putanje možemo pokrenuti mimikatz i to bi trebalo izgledati kao na sljedećoj slici:



```
mimikatz 2.2.0 x64 (oe.eo)
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> cd C:\Users\sally\Downloads\mimikatz_trunk\x64
PS C:\Users\sally\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 ***/

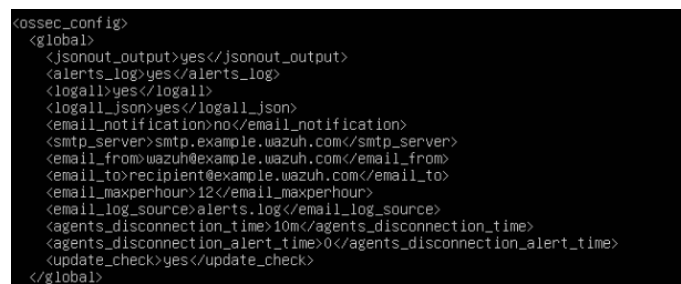
mimikatz #
```

Slika 24: Instalacija mimikatz

Na žalost wazuh ne prepoznaje nikakvu prijetnju nakon što je pokrenut mimikatz, jedan od razloga bi mogao biti to što sysmon nije triggerao nikakav event ili rule od wazuh jer prema defaultu wazuh ne prati sve logove, samo one koji su triggerani, to ćemo promijeniti konfiguracijom ossec.conf datoteke, na taj način wazuh će moći pratiti sve logove

3.7 Konfiguracija logova za Wazuh

Upisom naredbe nano /var/ossec/etc/ossec.conf uci ćemo u ossec.conf datoteku u kojoj ćemo promijeniti stavku <logall></logall> iz no u yes kao što je na slici:



```
<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>yes</logall>
    <logall_json>yes</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>
```

Slika 25: Konfiguracija logova

Nakon toga restartamo wazuh menager sa naredbom `systemctl restart wazuh-manager.service`, to će natjerati wazuh da arhivira sve logove i stavi ih u datoteku naziva `archives`

```
root@wazuh:/var/ossec/logs/archives# ls
2025 archives.json archives.log
```

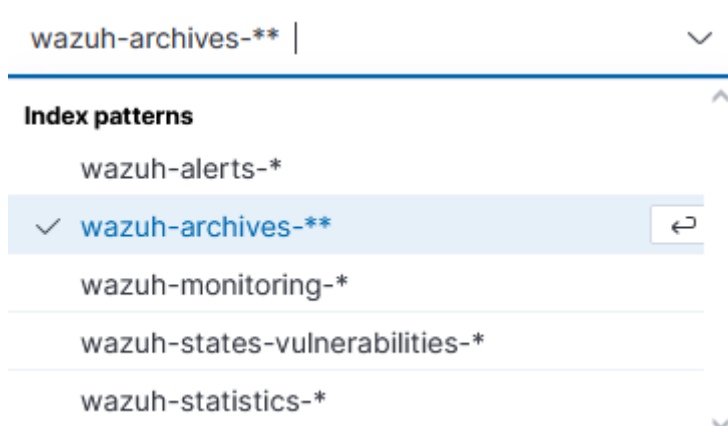
Slika 26: Konfiguracija logova

Kako bi wazuh počeo unositi te logove moramo promijeniti konfiguraciju filebeat datoteke. Unutar filebeat datoteke treba promijeniti `archives enabled` u `true`

```
filebeat.modules:
- module: wazuh
  alerts:
    enabled: true
  archives:
    enabled: true
logging.level: info
logging.to_files: true
logging.files:
  path: /var/log/filebeat
  name: filebeat
  keepfiles: 7
  permissions: 0644
```

Slika 27: Konfiguracija logova

Kada smo to promijenili možemo otići u wazuh dashboard i napraviti novi „archives“ indeks u kojem ćemo vidjeti sve generirane logove bez obzira da li je alert triggeran ili ne.



Slika 28: Archive index

Sada možemo vidjeti mimikatz alertove:

```
> Jan 11, 2025 @ 17:16:35.118
agent.ip: 192.168.11.128 agent.name: agent007 agent.id: 004 manager.name: wazuh data.win.eventdata.originalFilename: mimikatz.exe data.win.eventdata.image: C:\Users\Vi1e\Downloads\mimikatz_trunk\vs4\mimikatz.exe data.win.eventdata.product: mimikatz data.win.eventdata.imageLoaded: C:\Users\Vi1e\Downloads\mimikatz_trunk\vs4\mimikatz.exe data.win.eventdata.description: mimikatz for Windows data.win.eventdata.signed: false data.win.eventdata.signaturesStatus: Unavailable
data.win.eventdata.processId: (5bc29b7f-9787-6782-a688-000000002500) data.win.eventdata.processId: 8212 data.win.eventdata.startTime: 2025-01-11 16:08:39.511
data.win.eventdata.hash: SHA1-E386EABC4F83CEC8F23A5CF48B3AAE8D09 MD5-29FD6400C7FE1E2B0C287AD73A18A5 SHA256-61C8B18A23580CF492A8BA47765456618831E7A413496ICD0A8926

> Jan 11, 2025 @ 17:16:34.933
agent.ip: 192.168.11.128 agent.name: agent007 agent.id: 004 manager.name: wazuh data.win.eventdata.originalFilename: mimikatz.exe data.win.eventdata.image: C:\Users\Vi1e\Downloads\mimikatz_trunk\vs4\mimikatz.exe data.win.eventdata.product: mimikatz data.win.eventdata.parentProcessId: (5bc29b7f-9787-6782-a688-000000002500)
data.win.eventdata.description: mimikatz for Windows data.win.eventdata.parentProcessId: (5bc29b7f-9787-6782-a688-000000002500) data.win.eventdata.parentCommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" data.win.eventdata.processId: (5bc29b7f-9787-6782-a688-000000002500) data.win.eventdata.parentProcessId: 8x4289f
data.win.eventdata.parentProcessId: 18448 data.win.eventdata.processId: 8212 data.win.eventdata.currentDirectory: C:\Users\Vi1e\Downloads\mimikatz_trunk\vs4\

> Jan 11, 2025 @ 17:16:25.574
agent.ip: 192.168.11.128 agent.name: agent007 agent.id: 004 manager.name: wazuh data.win.eventdata.originalFilename: mimikatz.exe data.win.eventdata.image: C:\Users\Vi1e\Downloads\mimikatz_trunk\vs4\mimikatz.exe data.win.eventdata.product: mimikatz data.win.eventdata.imageLoaded: C:\Users\Vi1e\Downloads\mimikatz_trunk\vs4\mimikatz.exe data.win.eventdata.description: mimikatz for Windows data.win.eventdata.signed: false data.win.eventdata.signaturesStatus: Unavailable
data.win.eventdata.processId: (5bc29b7f-9787-6782-a688-000000002500) data.win.eventdata.processId: 12088 data.win.eventdata.startTime: 2025-01-11 16:05:29.938
data.win.eventdata.hash: SHA1-E386EABC4F83CEC8F23A5CF48B3AAE8D09 MD5-29FD6400C7FE1E2B0C287AD73A18A5 SHA256-61C8B18A23580CF492A8BA47765456618831E7A413496ICD0A8926

> Jan 11, 2025 @ 17:16:23.785
agent.ip: 192.168.11.128 agent.name: agent007 agent.id: 004 manager.name: wazuh data.win.eventdata.originalFilename: mimikatz.exe data.win.eventdata.image: C:\Users\Vi1e\Downloads\mimikatz_trunk\vs4\mimikatz.exe data.win.eventdata.product: mimikatz data.win.eventdata.parentProcessId: (5bc29b7f-9787-6782-a688-000000002500)
data.win.eventdata.description: mimikatz for Windows data.win.eventdata.parentProcessId: (5bc29b7f-9787-6782-a688-000000002500) data.win.eventdata.parentCommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" data.win.eventdata.processId: (5bc29b7f-9787-6782-a688-000000002500) data.win.eventdata.parentProcessId: 8x4289f
data.win.eventdata.parentProcessId: 18448 data.win.eventdata.processId: 12088 data.win.eventdata.currentDirectory: C:\Users\Vi1e\Downloads\mimikatz_trunk\vs4\

> Jan 11, 2025 @ 16:59:38.923
agent.ip: 192.168.11.128 agent.name: agent007 agent.id: 004 manager.name: wazuh data.win.eventdata.originalFilename: mimikatz.exe data.win.eventdata.image: C:\Users\Vi1e\Downloads\mimikatz_trunk\vs4\mimikatz.exe data.win.eventdata.product: mimikatz data.win.eventdata.imageLoaded: C:\Users\Vi1e\Downloads\mimikatz_trunk\vs4\mimikatz.exe data.win.eventdata.description: mimikatz for Windows data.win.eventdata.signed: false data.win.eventdata.signaturesStatus: Unavailable
data.win.eventdata.processId: (5bc29b7f-9787-6782-a688-000000002500) data.win.eventdata.processId: 7744 data.win.eventdata.startTime: 2025-01-10 20:28:33.311
```

Slika 29: mimikatz alertovi

3.8 Kreiranje pravila

Napadač bi mogao samo preimenovati mimikatz file i alert bi se izbjegao, zato ćemo napraviti pravilo gdje se detektira originalni naziv datoteke. Wazuh ima opciju kreiranja pravila, tako ćemo mi kreirati svoje prilagođeno pravilo za mimikatz, možemo uzeti neko već napravljeno pravilo od wazuha, specifično sysmon pravilo kao što je na slici:

Rules (4483)						
From here you can manage your rules.						
Search						
WQL Custom rules						
ID	Description	Group	Regulatory compliance	Level	File	Path
1	Generic template for all syslog rules.	syslog		0	0010-rules_config.xml	ruleset/rules
2	Generic template for all firewall rules.	firewall		0	0010-rules_config.xml	ruleset/rules
3	Generic template for all ids rules.	ids		0	0010-rules_config.xml	ruleset/rules
4	Generic template for all web rules.	web-log		0	0010-rules_config.xml	ruleset/rules
5	Generic template for all web proxy rules.	squid		0	0010-rules_config.xml	ruleset/rules
6	Generic template for all windows rules.	windows		0	0010-rules_config.xml	ruleset/rules
7	Generic template for all wazuh rules.	ossec		0	0010-rules_config.xml	ruleset/rules
200	Grouping of wazuh rules.	wazuh		0	0016-wazuh_rules.xml	ruleset/rules
201	Agent event queue rule	agent_flooding_wazuh		0	0016-wazuh_rules.xml	ruleset/rules
202	Agent event queue is level full.	agent_flooding_wazuh	PCI DSS GDPR	7	0016-wazuh_rules.xml	ruleset/rules
Rows per page: 10						
1 2 3 4 5 ... 449						

Slika 30: Kreiranje pravila

```

<!--
Copyright (C) 2015, Wazuh Inc.
-->

<!--
Sysmon Event ID 1 rules: 92000 - 92100
-->

<group name="sysmon,sysmon_eid1,detections,windows,">

  <rule id="92000" level="4">
    <if_group>sysmon_event1</if_group>
    <field name="win.eventdata.parentImage" type="pcr2">(?!)(c|w)script\.exe</field>
    <options>no_full_log</options>
    <description>Scripting interpreter spawned a new process</description>
    <mitre>
      <id>T1059.005</id>
    </mitre>
  </rule>

  <rule id="92001" level="6">
    <field name="win.eventdata.commandLine" type="pcr2">(?!)(c|w)script\.exe\.\\(bat|cmd|lnk|pif|vbe|js|vbs|ps1)</field>
    <options>no_full_log</options>
    <description>Scripting interpreter spawned new scripting interpreter</description>
    <mitre>
      <id>T1059</id>
    </mitre>
  </rule>

  <rule id="92002" level="6">
    <field name="win.eventdata.commandLine" type="pcr2">\\cmd\\.exe</field>
    <options>no_full_log</options>
    <description>Scripting interpreter spawned Windows command shell instance</description>
    <mitre>
      <id>T1059.003</id>
    </mitre>
  </rule>

```

Slika 31: Kreiranje pravila

Sada možemo kopirati `<rule></rule>` dio i zaljepiti u svoje custom pravilo kako bi nam bilo lakše pri izradi svojeg pravila.

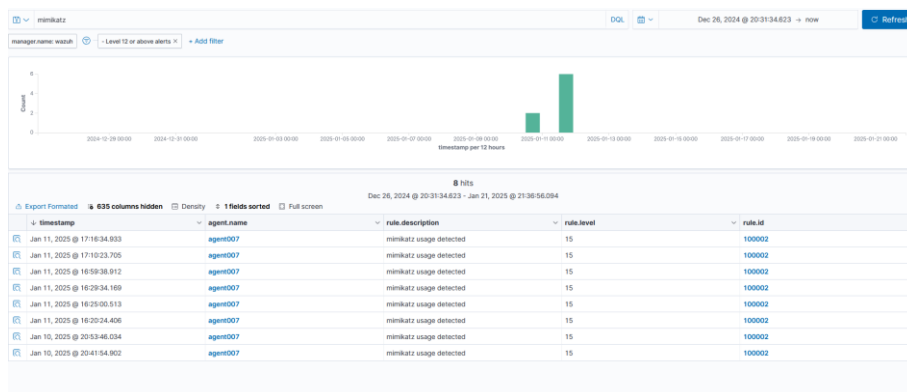
```

1 <!-- Local rules -->
2
3 <!-- Modify it at your will. -->
4 <!-- Copyright (C) 2015, Wazuh Inc. -->
5
6 <!-- Example -->
7 <group name="local,syslog,sshd,">
8
9   <!--
10   Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
11   -->
12   <rule id="100001" level="5">
13     <if_sid>5716</if_sid>
14     <srcip>1.1.1.1</srcip>
15     <description>sshd: authentication failed from IP 1.1.1.1.</description>
16     <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
17   </rule>
18
19   <rule id="100002" level="15">
20     <if_group>sysmon_event1</if_group>
21     <field name="win.eventdata.originalFileName" type="pcr2">(?!)(mimikatz|\\exe</field>
22     <description>mimikatz detektiran</description>
23     <mitre>
24       <id>T1003</id>
25     </mitre>
26   </rule>
27
28 </group>
29

```

Slika 32: Kreiranje pravila

Kada se restarta wazuh i pokrene mimikatz, alert bi se trebao vidjeti kao na slici:



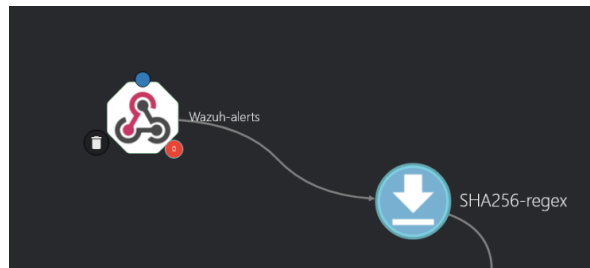
Slika 33: Mimikatz alert

3.9 Shuffle

U Shuffleu se kreira novi tijek rada klikom na "Change Me" ikonu, te se dodaju aplikacije i okidači (triggers).

3.9.1 Dodavanje webhook okidača

Odabire se opcija "Triggers" i povlači (drag and drop) Webhook okidač nazvan "Wazuh-Alerts".



Slika 34: Webhook okidač

Generira se URI za webhook koji se kopira i dodaje u datoteku ossec.conf unutar wazuha za povezivanje sa shuffleom.

```
ossec.conf:ip:
<global>
  <jsonout_output>yes</jsonout_output>
  <alerts_log>yes</alerts_log>
  <logall>yes</logall>
  <logall_json>yes</logall_json>
  <email_notification>no</email_notification>
  <smtp_server>smtp.example.wazuh.com</smtp_server>
  <email_from>wazuh@wazuh.com</email_from>
  <email_to>recipient@example.wazuh.com</email_to>
  <email_maxperhour>12</email_maxperhour>
  <email_log_source>alerts.log</email_log_source>
  <agents_disconnection_time>10</agents_disconnection_time>
  <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
  <update_check>yes</update_check>
</global>

<integration>
  <name>shuffle</name>
  <hook_url>https://shuffle.io/api/v1/hooks/webhook_12b12345-8000-4081-8522-3388a079a18</hook_url>
  <rule_id>100002</rule_id>
  <alert_format>json</alert_format>
</integration>

<alerts>
  <log_alert_level>3</log_alert_level>
  <email_alert_level>12</email_alert_level>
</alerts>
```

Slika 35: Dodavanje URI

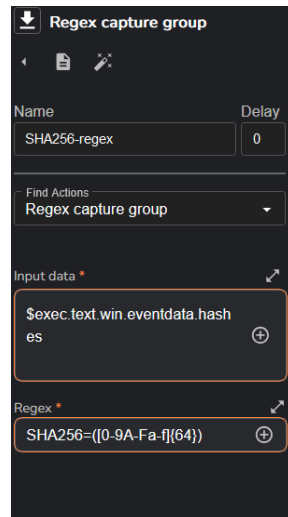
Kada wazuh detektira sigurnosni događaj, alarm se automatski proslijeđuje kroz webhook na shuffle.

3.9.2 Provjera reputation score sa virustotal

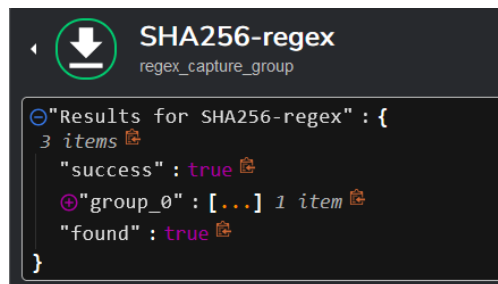
Kako bi koristili virustotal potrebno je parsirati hash, u shuffleu imamo regex opciju koja će nam to omogućiti

```
„2HV1=E38EY8C46L483ICECEf532V2CL48B38V4VE8DE8"WD2=5AEfDE4DD3C3EETE5B053B1VD13V18V2"2HV32E=EIC88J8V53  
„p92p62_“ :
```

Slika 36: Regex

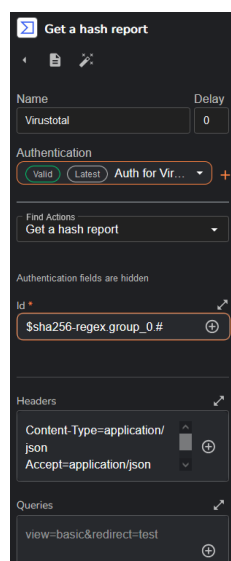


Slika 37: Regex



Slika 38: Regex

Kada spajamo virustotal moramo ga prvo povezati sa shuffleom tako da koristimo API ključ koji dobijemo od virustotala



Slika 39: API autentifikacija

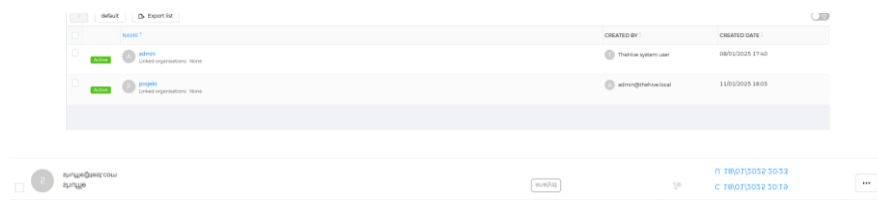
Kada smo ga spojili dobivamo rezultate na slici ispod, „malicious“: 63 nam govori da je 63 skenera detektiralo tu datoteku kao malicioznu

```
⊖  
"last_analysis_stats": {  
  8 items  
    "malicious": 63  
    "suspicious": 0  
    "undetected": 9  
    "harmless": 0  
    "timeout": 0  
    "confirmed-timeout": 0  
    "failure": 0  
    "type-unsupported": 4  
}
```

Slika 40: Podaci od Virustotala

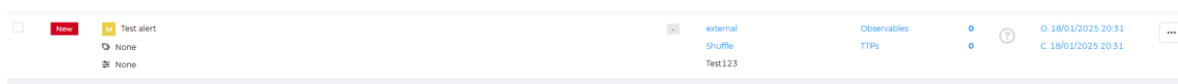
3.9.3 Slanje alerta na TheHive

Za integraciju TheHive platforme za upravljanje incidentima, prvo smo kreirali novu organizaciju i korisnika unutar TheHive-a te generirali API ključ za korisnički račun servisa.



Slika 41: Kreiranje organizacije

Zatim smo konfigurirali shuffle kako bi uz pomoć API ključa kreirali alarme u TheHive platformi i postavili smo ključna polja. Nakon ponovnog pokretanja workflowa, alarm se pojavio u platformi za upravljanje incidentima.



Slika 42: Alert u TheHiveu

Svi analitičari koji pregledavaju stranicu s alarmima sada mogu vidjeti alarm i dodijeliti ga određenom članu tima. Na kraju, implementirali smo slanje e-mail obavijesti. Kako bih spriječili potencijalno pretrpavanje.

Send email shuffle

Name: email_1 Delay: 0

Find Actions: Send email shuffle

Recipients *: test@example.com,support@sh

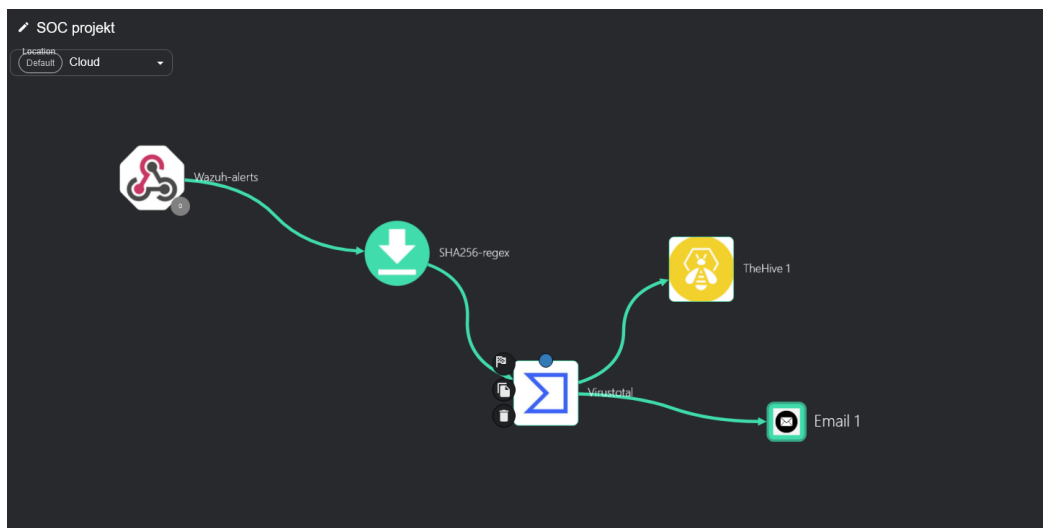
Subject *: Mimikatz detektiran

Simple Advanced

Slika 43: Postavljanje e-maila

Shuffle Email App 3 Sent by email app: Mimikatz detektiran - vrijeme:\$exec.all_fields.full_log.win.eventdata.utcTime Naslov: \$exec.title

Slika 44: E-mail obavjest



Slika 45: Završni workflow

4. Zaključak

Implementacija ovog sustava pokazala je kako moderni alati poput wazuha, TheHive-a i shufflea mogu pružiti sveobuhvatno rješenje za upravljanje sigurnosnim incidentima. Integracija ovih alata unutar cloud okruženja ne samo da olakšava detekciju prijetnji, već i omogućuje bržu reakciju i obradu incidenata putem automatiziranih tijekova rada. Ovaj projekt jasno demonstrira potencijal korištenja open-source tehnologija u povećanju efikasnosti i točnosti unutar security operations centra (SOC).

Jedna od ključnih prednosti sustava je fleksibilnost koju omogućuje shuffle, koji povezuje različite alate i omogućuje korisnicima prilagodbu tijekova rada prema specifičnim potrebama organizacije. Wazuh se istaknuo kao snažan alat za detekciju prijetnji, dok je TheHive pružio platformu za upravljanje incidentima. Automatizacija putem e-mail obavijesti dodatno je unaprijedila komunikaciju i ubrzala vrijeme reakcije analitičara.

Osim što sustav nudi skalabilnost i pouzdanost, projekt je također otvorio mogućnosti za daljnja unaprjeđenja. Primjerice, integracija dodatnih alata za obogaćivanje podataka o incidentima ili implementacija naprednijih pravila za detekciju mogla bi značajno povećati učinkovitost sustava.

5. Popis slika

Slika 1: Vizualizacija sustava	3
Slika 2: Sysmon instalacija	4
Slika 3: Status sysmona u prozoru „Services“	5
Slika 4: Firewall	5
Slika 5: Naredba za instalaciju Wazuh	6
Slika 6: Naredba za uzimanje korisničkih podataka	6
Slika 7: Wazuh interface	6
Slika 8: Instalacija Wazuh agenta	7
Slika 9: Instalacija Jave	7
Slika 10: Instalacija Cassandra	8
Slika 11: Instalacija ElasticSearcha	8
Slika 12: Instalacija TheHivea	8
Slika 13: .yaml file (Cassandra)	9
Slika 14: .yaml file (Cassandra)	9
Slika 15: .yaml file (Cassandra)	9
Slika 16: .yaml file (ElasticSearch)	10
Slika 17: .yaml file (ElasticSearch)	10
Slika 18: Putanja	10
Slika 19: Promjena vlasnika	11
Slika 20: Konfiguracija TheHivea	11
Slika 21: Konfiguracija TheHivea	11
Slika 22: Potrošnja memorije	12
Slika 23: Konfiguracija ossec	12
Slika 24: Instalacija mimikatz	13
Slika 25: Konfiguracija logova	13
Slika 26: Konfiguracija logova	14
Slika 27: Konfiguracija logova	14
Slika 28: Archive index	14
Slika 29: mimikatz alertovi	15
Slika 30: Kreiranje pravila	15
Slika 31: Kreiranje pravila	16
Slika 32: Kreiranje pravila	16
Slika 33: Mimikatz alert	16
Slika 34: Webhook okidač	17
Slika 35: Dodavanje URI	17
Slika 36: Regex	17
Slika 37: Regex	18
Slika 38: Regex	18
Slika 39: API autentifikacija	18
Slika 40: Podaci od Virustotala	19
Slika 41: Kreiranje organizacije	19
Slika 42: Alert u TheHiveu	19
Slika 43: Postavljanje e-maila	20
Slika 44: E-mail obavjest	20
Slika 45: Završni workflow	20