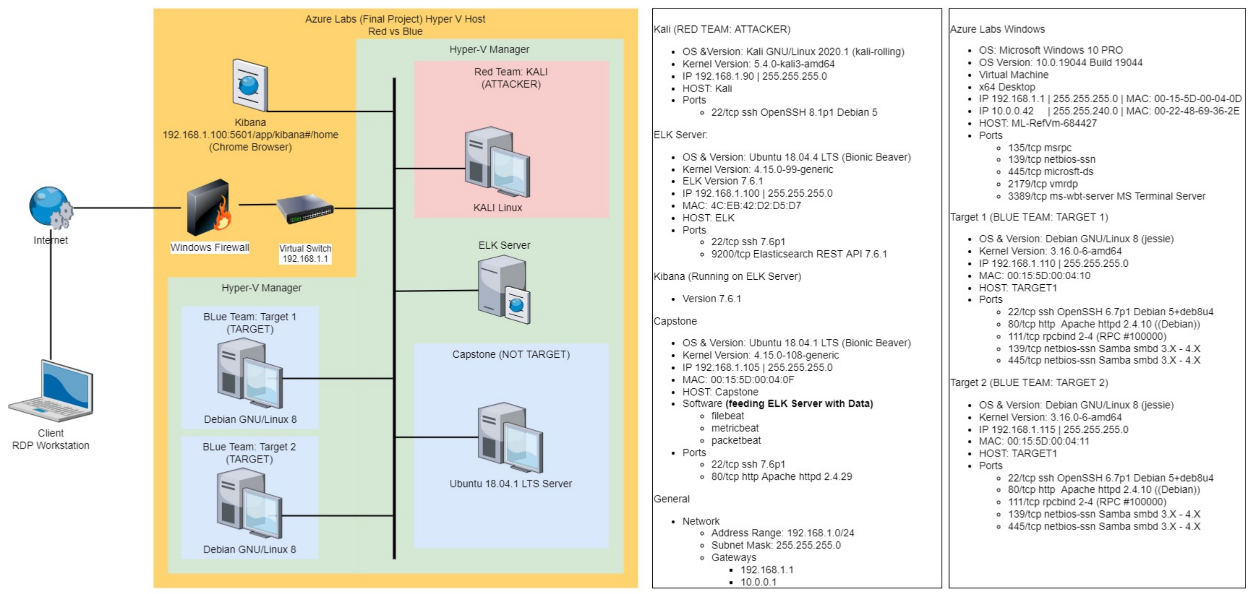
Blue Team: Summary of Operations

Table of Contents

Network Topology

Azure Lab (Final Project): Red vs Blue Network Architecture


Azure Labs (Final Project) Hyper V Host
Red vs Blue
Hyper-V Manager
Red Team: KALI (ATTACKER)
KALI Linux
Kibana
192.168.1.100:5601/app/kibana#/home
(Chrome Browser)
Internet
Windows Firewall
Virtual Switch 192.168.1.1
ELK Server
Hyper-V Manager
BLue Team: Target 1 (TARGET)
Debian GNU/Linux 8
BLue Team: Target 2 (TARGET)
Debian GNU/Linux 8
Capstone (NOT TARGET)
Ubuntu 18.04.1 LTS Server
Client
RDP Workstation

Kali (RED TEAM: ATTACKER)

- OS &Version: Kali GNU/Linux 2020.1 (kali-rolling)
- Kernel Version: 5.4.0-kali3-amd64
- IP 192.168.1.90 | 255.255.255.0
- HOST: Kali
- Ports
  - 22/tcp ssh OpenSSH 8.1p1 Debian 5

ELK Server:

- OS & Version: Ubuntu 18.04.4 LTS (Bionic Beaver)
- Kernel Version: 4.15.0-99-generic
- ELK Version 7.6.1
- IP 192.168.1.100 | 255.255.255.0
- MAC: 4C:EB:42:D2:D5:D7
- HOST: ELK
- Ports
  - 22/tcp ssh 7.6p1
  - 9200/tcp Elasticsearch REST API 7.6.1

Kibana (Running on ELK Server)

- Version 7.6.1

Capstone

- OS & Version: Ubuntu 18.04.1 LTS (Bionic Beaver)
- Kernel Version: 4.15.0-108-generic
- IP 192.168.1.105 | 255.255.255.0
- MAC: 00:15:5D:00:04:0F
- HOST: Capstone
- Software **(feeding ELK Server with Data)**
  - filebeat
  - metricbeat
  - packetbeat
- Ports
  - 22/tcp ssh 7.6p1
  - 80/tcp http Apache httpd 2.4.29

General

- Network
  - Address Range: 192.168.1.0/24
  - Subnet Mask: 255.255.255.0
  - Gateways
    - 192.168.1.1
    - 10.0.0.1

Azure Labs Windows

- OS: Microsoft Windows 10 PRO
- OS Version: 10.0.19044 Build 19044
- Virtual Machine
- x64 Desktop
- IP 192.168.1.1 | 255.255.255.0 | MAC: 00-15-5D-00-04-0D
- IP 10.0.0.42 | 255.255.240.0 | MAC: 00-22-48-69-36-2E
- HOST: ML-RefVm-684427
- Ports
  - 135/tcp msrpc
  - 139/tcp netbios-ssn
  - 445/tcp microsft-ds
  - 2179/tcp vmrdp
  - 3389/tcp ms-wbt-server MS Terminal Server

Target 1 (BLUE TEAM: TARGET 1)

- OS & Version: Debian GNU/Linux 8 (jessie)
- Kernel Version: 3.16.0-6-amd64
- IP 192.168.1.110 | 255.255.255.0
- MAC: 00:15:5D:00:04:10
- HOST: TARGET1
- Ports
  - 22/tcp ssh OpenSSH 6.7p1 Debian 5+deb8u4
  - 80/tcp http Apache httpd 2.4.10 ((Debian))
  - 111/tcp rpcbind 2-4 (RPC #100000)
  - 139/tcp netbios-ssn Samba smbd 3.X - 4.X
  - 445/tcp netbios-ssn Samba smbd 3.X - 4.X

Target 2 (BLUE TEAM: TARGET 2)

- OS & Version: Debian GNU/Linux 8 (jessie)
- Kernel Version: 3.16.0-6-amd64
- IP 192.168.1.115 | 255.255.255.0
- MAC: 00:15:5D:00:04:11
- HOST: TARGET1
- Ports
  - 22/tcp ssh OpenSSH 6.7p1 Debian 5+deb8u4
  - 80/tcp http Apache httpd 2.4.10 ((Debian))
  - 111/tcp rpcbind 2-4 (RPC #100000)
  - 139/tcp netbios-ssn Samba smbd 3.X - 4.X
  - 445/tcp netbios-ssn Samba smbd 3.X - 4.X

Notes:

- For the details of, and methodology to retrieve the information in this topology, refer to the Readme.md
- The topology is also provided in .jpg and .pdf files for easy reference while reading the analyses documents.

The following table is a summary of the machines that were identified on the network (for more detail i.e. **OS and OS Versions**, see the Topology above):

VM	IP Address	Description
Kali	`192.168.1.90`	A standard Kali install that will be used to attack other machines.
Capstone	`192.168.1.105`	The vulnerable target VM that students can use to test alerts. Filebeat and Metricbeat will forward logs to the ELK machine.
ELK	`192.168.1.100`	The same ELK setup that you created in Project 1. It holds the Kibana dashboards that you will use in Day 2.
Target 1	`192.168.1.110`	Exposes a vulnerable WordPress server. Sends logs to ELK.
Target 2	`192.168.1.115`	A more difficult WordPress target. Should be ignored unless all other portions of the project are completed. Sends logs to ELK.

Description of Targets

The target of this attack was: `Target 1` (`192.168.1.110`) and `Target 2 (192.168.1.115)` .

Target 1 & 2 are Apache web servers and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers. As such, the following alerts have been implemented:

Monitoring the Targets

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

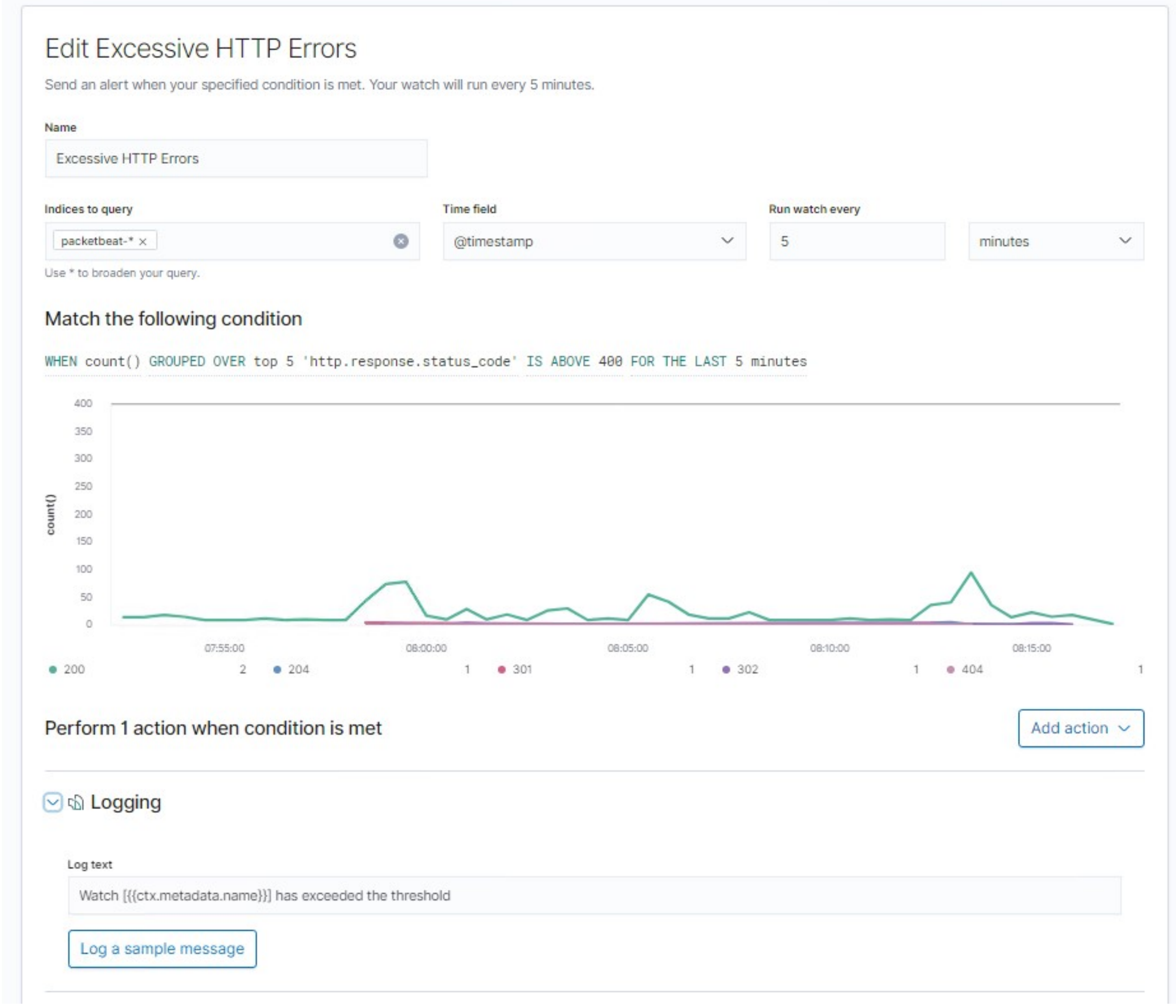
Method to Create Alerts

1. Access Kibana at `192.168.1.100:5601`
2. Click on **Management** > **License Management** and enable the Kibana Premium Free Trial.
3. Click **Management** > **Watcher** > **Create Alert** > **Create Threshold Alert**
4. Implement three the alerts.

Alert 1: Excessive HTTP Errors (Responce Codes)

Note: the value 400 below does not relate to the HTTP Response Code `400` , but it is a threshold of 400! It applies to the count of traffic at a rate of > 400 responses every 5 minutes, which will trigger this Alert. For example, 500 GETs of / at the server will fire the alert even though the only response code is `200`

- **Metric**: `Packetbeat:` http.response.status_code > 400
- **Threshold**: `grouped http response status codes COUNT above 400 every 5 minutes`
  - **`When count() GROUPED OVER top5 ‘http.response.status_code’ is above 400 for the last 5 minutes`**
- **Vulnerability Mitigated**:
  - Denial of Service
  - Brute Force Attacks
  - Suspicious activity
- **Reliability**: This alert will not generate an excessive amount of false positives identifying brute force attacks. **`Medium`**



Edit Excessive HTTP Errors

Send an alert when your specified condition is met. Your watch will run every 5 minutes.

Name: Excessive HTTP Errors

Indices to query: packetbeat-*

Use * to broaden your query.

Time field: @timestamp

Run watch every: 5 minutes

Match the following condition

```
WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
```

Perform 1 action when condition is met

Add action

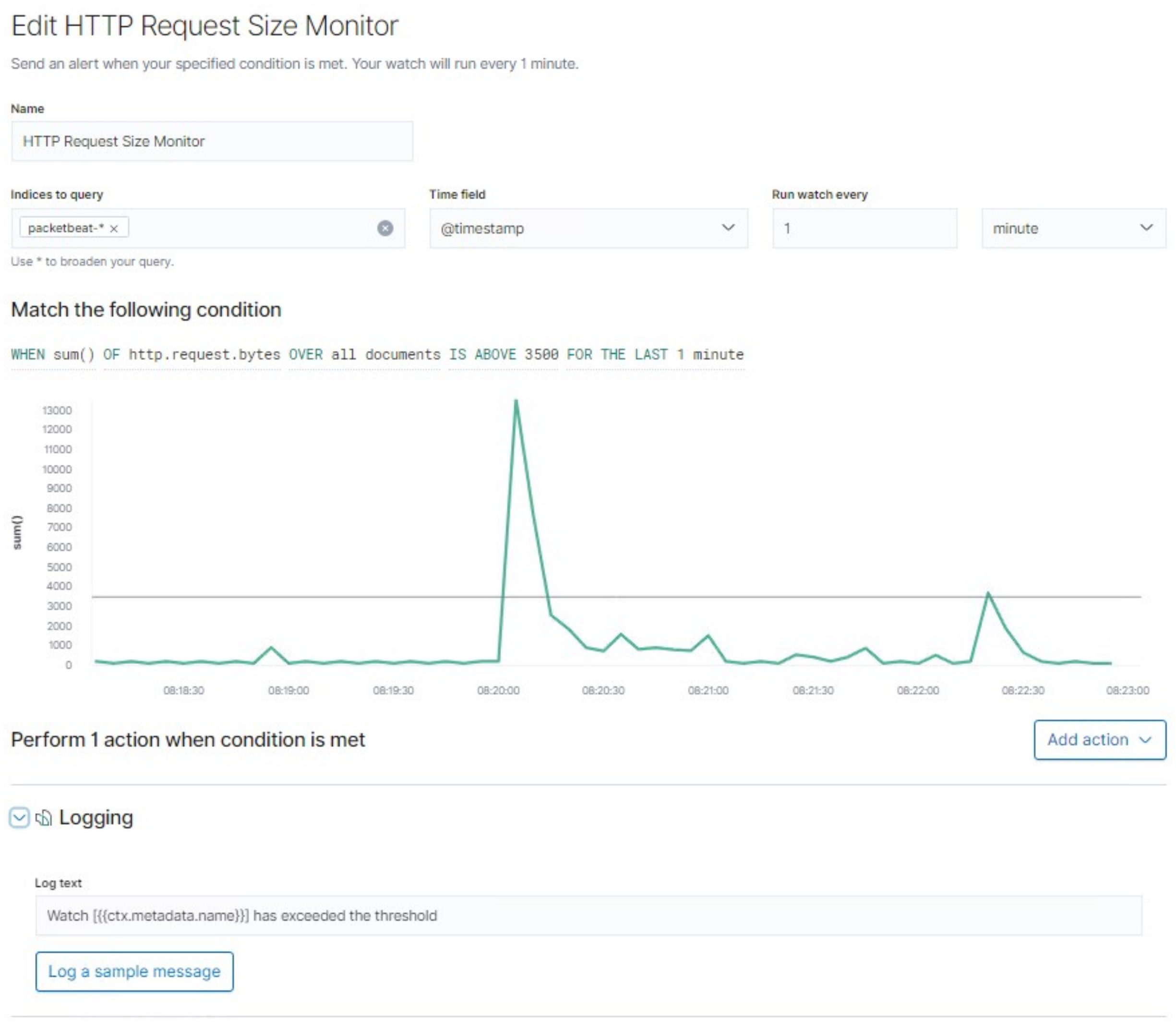
Logging

Log text: Watch [{{ctx.metadata.name}}] has exceeded the threshold

Log a sample message

Alert 2: HTTP Request Size Monitor

- **Metric**: `Packetbeat:` http.request.bytes
- **Threshold**: The sum of the requested bytes is over 3500 in 1 minute
  - **`When sum() of http.request.bytes OVER all documents is ABOVE 3500 for the LAST 1 minute`**
- **Vulnerability Mitigated**: By controlling the number of http request sizes through a filter, protection is enabled to detect or prevent DDOS attacks for IPS/IDS.
- **Reliability**: No, this alert doesn't generate an excessive amount of false positives because DDOS attacks submit requests within seconds, not within minutes. **`Medium`**


Edit HTTP Request Size Monitor

Send an alert when your specified condition is met. Your watch will run every 1 minute.

Name: HTTP Request Size Monitor

Indices to query: packetbeat-*

Use * to broaden your query.

Time field: @timestamp

Run watch every: 1 minute

Match the following condition

```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```

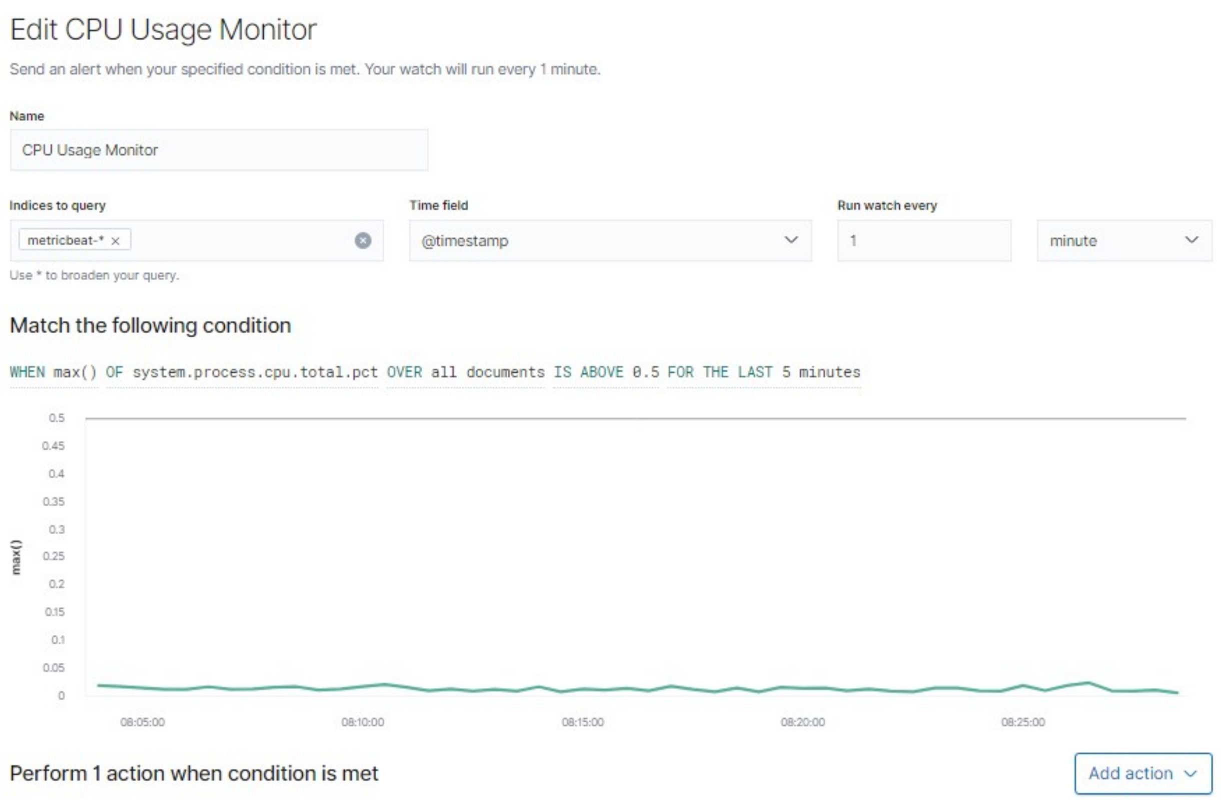
Perform 1 action when condition is met

Logging

Log text: Watch [{{ctx.metadata.name}}] has exceeded the threshold

Alert 3: CPU Usage Monitor

- **Metric**: `Metricbeat:` system.process.cpu.total.pct
- **Threshold**: The maximum cpu total percentage is over .5 in 5 minutes
  - **`WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes`**
- **Vulnerability Mitigated**: Controlling the CPU usage percentage at 50%, it will trigger a memory alert only if the CPU remains at or above 50% consistently for 5 minutes. Virus or Malware
- **Reliability**: Yes, this alert can generate a lot of false positives due to CPU spikes occurring when specific integrations are initiated at the start of processing. **`High`**


Edit CPU Usage Monitor

Send an alert when your specified condition is met. Your watch will run every 1 minute.

Name: CPU Usage Monitor

Indices to query: metricbeat-*

Use * to broaden your query.

Time field: @timestamp

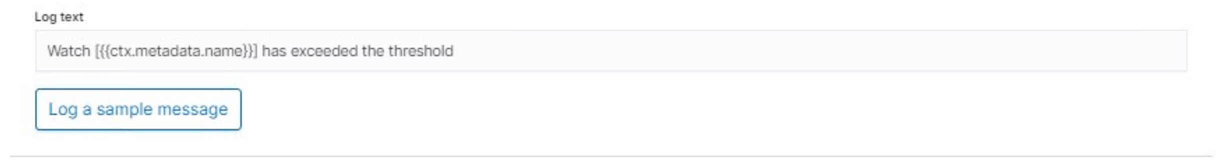
Run watch every: 1 minute

Match the following condition

```
WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
```

Perform 1 action when condition is met

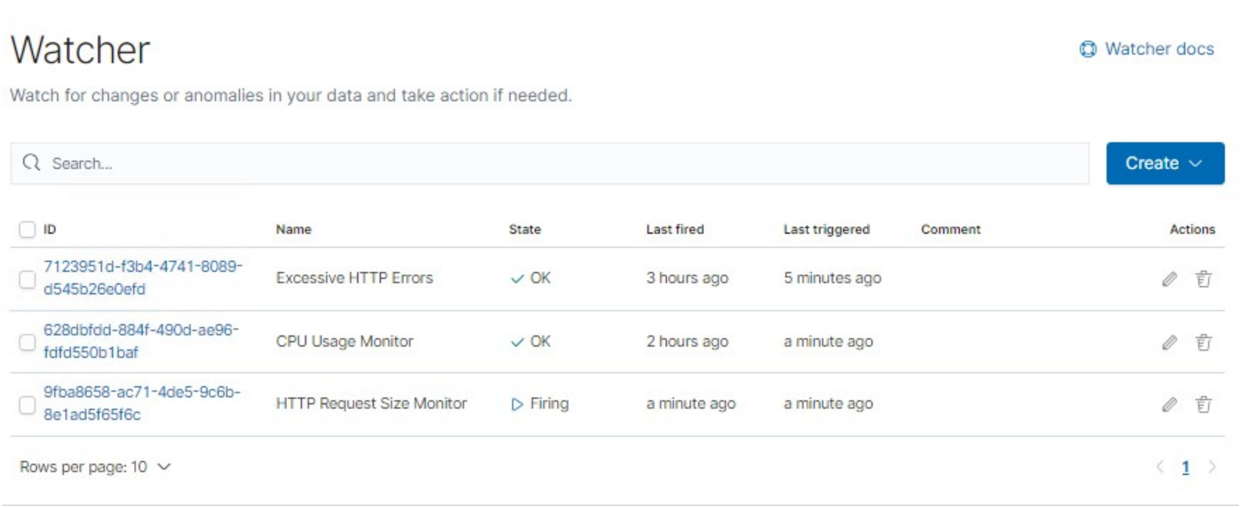
132 lines (88 sloc) | 6.69 KB


Log text: Watch [{{ctx.metadata.name}}] has exceeded the threshold

Viewing Logs Messages

There are a few way to to view these log messages and their associated data options.

- First, you can see when alerts are firing directly from the Watcher screen. All 3 Alerts are implemented, and functioning as expected as can be seen from the Watcher Statistics


Watcher

Watch for changes or anomalies in your data and take action if needed.

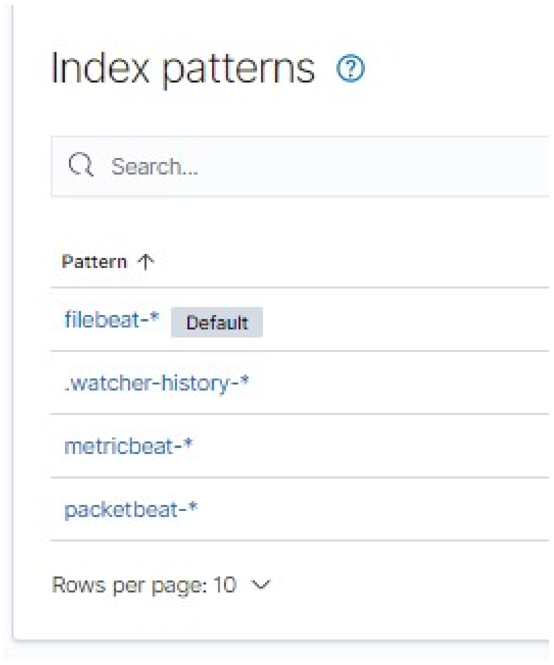
ID	Name	State	Last fired	Last triggered	Comment	Actions
7123951d-f3b4-4741-8089-d545b26e0efd	Excessive HTTP Errors	✓ OK	3 hours ago	5 minutes ago		
628dbfdd-884f-490d-ae96-fdfd550b1baf	CPU Usage Monitor	✓ OK	2 hours ago	a minute ago		
9fba8658-ac71-4de5-9c6b-8e1ad5f65f6c	HTTP Request Size Monitor	▷ Firing	a minute ago	a minute ago		

Rows per page: 10

- To view network traffic associated with these messages, we need to create a new `Index Pattern`:
- Click on **Management** > **Index Patterns** and click on the button for `Create Index Pattern`.

- Make sure to turn on the toggle button labeled `Include System Indices` on the top right corner.

- Create the pattern `.watcher-history-*`.



Pattern
filebeat-* Default
.watcher-history-*
metricbeat-*
packetbeat-*

- After you have this new index pattern, you can search through it using the `Discovery` page.
- Enter `result.condition.met` in as search filter to see all the traffic from your alerts.

Suggestions for Going Further (Optional)

- Each alert above pertains to a specific vulnerability/exploit. Recall that alerts only detect malicious behavior, but do not stop it. For each vulnerability/exploit identified by the alerts above, the following patches should mitigate against the exploits.
- Vulnerability 1: Weak Passwords
  - **Patch**: Enforce password policy (complexity, MFA, expiry, etc.)
  - **Why It Works**: Brute force attacks in this exercise would be next to impossible.
- Vulnerability 2: Python Privelege Escalation
  - **Patch**: Kibana can be configured to monitor for this attack and Steven's privileges can be revoked.
  - **Why It Works**: Attacks can be detected and Steven must enter a password before obtaining sudo privileges.
- Vulnerability 3: SQL Data Breaches
  - **Patch**: Restrict Access to certain key tables
  - **Why It Works**: Key tables will be inaccessible when users with dodgy security practices are compromised.
- Vulnerability 4: Alert 1: DoS Denial of Service/Brute Force Attacks
  - **Patch**: Block IP addresses, and install firewall
  - **Why It Works**: IP addresses are blocked so they cannot bruteforce or DoS you and a firewall will further harden the network
- Vulnerability 5: Alert 2: DoS Denial of Service/Brute Force Attacks
  - **Patch**: Block IP addresses, and install firewall
  - **Why It Works**: IP addresses are blocked so they cannot bruteforce or DoS you and a firewall will further harden the network
- Vulnerability 6: Alert 3: Malware on System/Suspicious activity i.e. extraction of large amount of data or deleting / encrypting system
  - **Patch**: Patch software and antivirus
  - **Why It Works**: Patched systems and software limits attack vectors. Antivirus stops an attack by quarantining malware/stopping it from operating