Defensive: Summary of Operations

Table of Contents

- Network Topology
- Description of Targets
- Monitoring the Targets

Network Topology

- Kali
  - Operating System: linux
  - Purpose: Attacker / pen test machine
  - IP Address: 192.168.1.90
- ELK
  - Operating System: linux
  - Purpose: Analyze data
  - IP Address: 192.168.1.100
- Target 1
  - Operating System: linux
  - Purpose: WordPress server
  - IP Address: 192.168.1.110
- Capstone
  - Operating System: linux
  - Purpose: The Vulnerable Web Server
  - IP Address: 192.168.1.105

The following machines where identified on the network:

- Name of VM 1:Target 1
  - **Operating System**:Linux
  - **Purpose**:Word press server
  - **IP Address**:192.168.1.110
- Name of VM 2
  - **Operating System**:
  - **Purpose**:
  - **IP Address**:

Description of Targets

The target of this attack was: Target 1 ip. 192.168.1.110

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

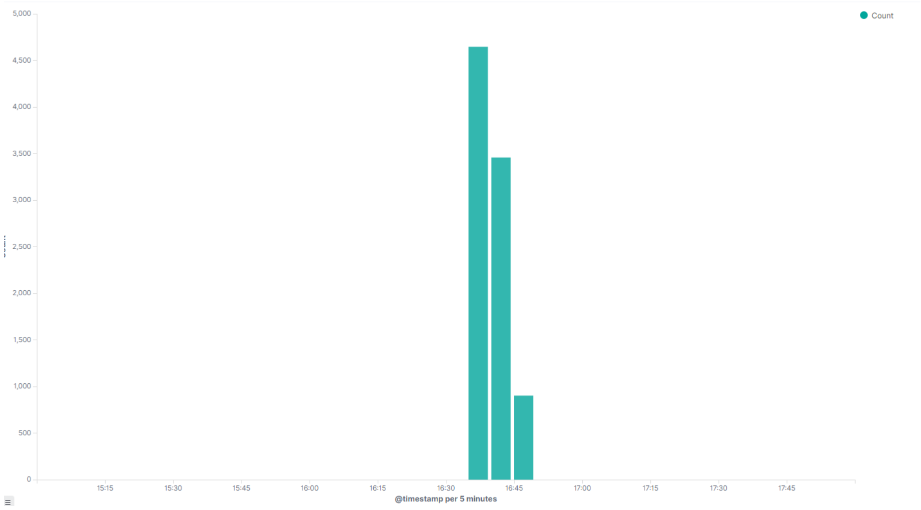
Monitoring the Targets

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

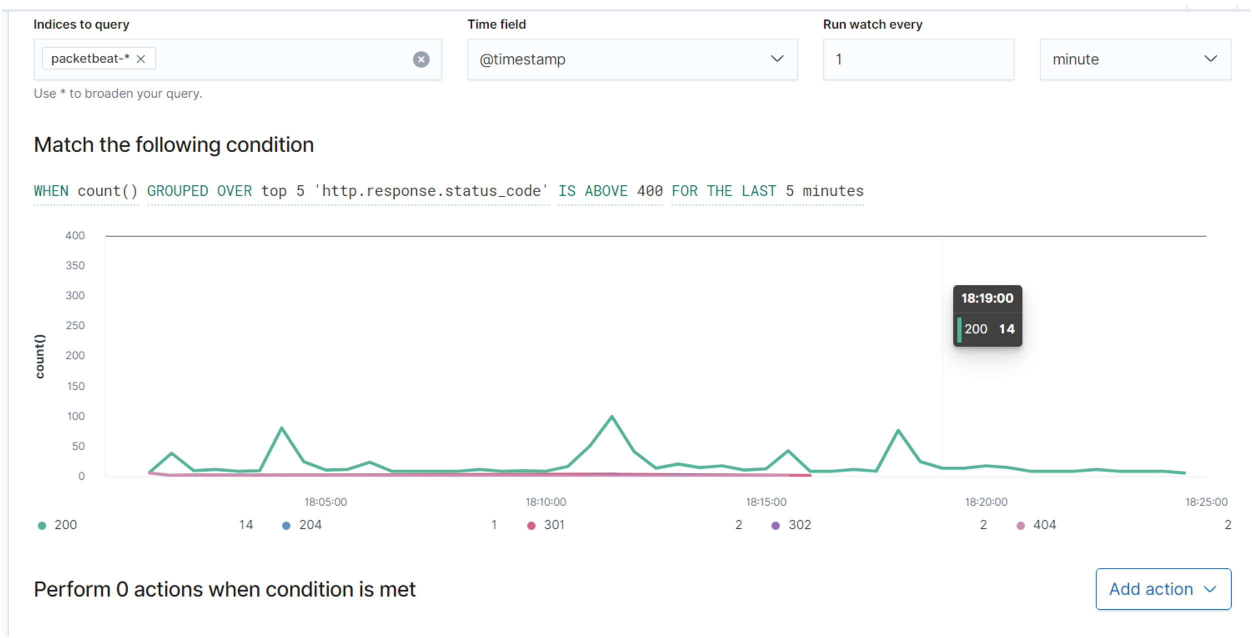
Name of Alert 1

Excessive HTTP Errors is implemented as follows:

- **Metric**: http.respnose.status_code
- **Threshold**:http://response.status_code> 400
- **Vulnerability Mitigated**: Brute Force Attack
- **Reliability**: This alert does not generate false positives. I feel that it is a highly reliable alert for monitoring a brute force attack.
-



-

- 

  Indices to query

  packetbeat-* ×

  Use * to broaden your query.

  Time field

  @timestamp

  Run watch every

  1 minute

  Match the following condition

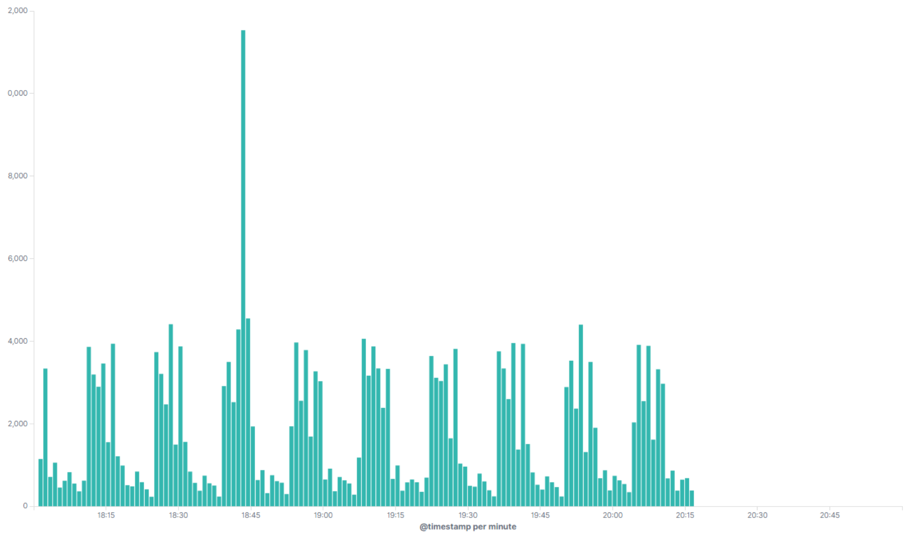
  ```
  WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
  ```

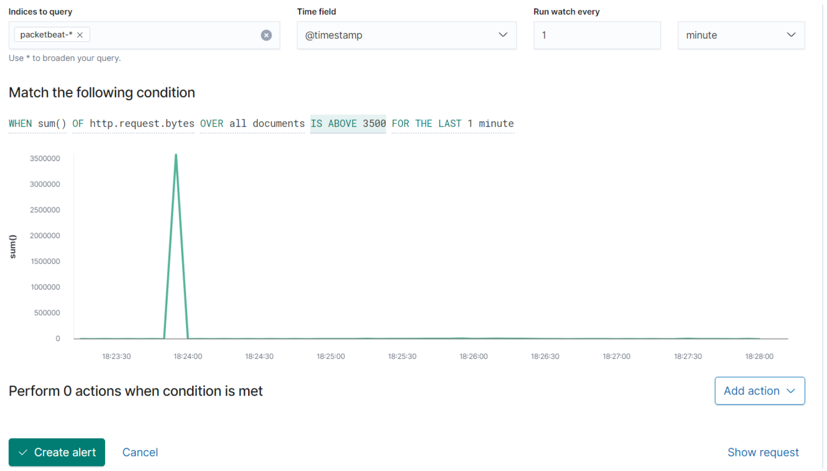
  ● 200 14 ● 204 1 ● 301 2 ● 302 2 ● 404 2

  Perform 0 actions when condition is met

Name of Alert 2

HTTP Request Size Monitor is implemented as follows:

- **Metric**:http.request.bytes
- **Threshold**: 3500 hits in 1 min
- **Vulnerability Mitigated**: DDOS
- **Reliability**: No false positives. I feel the reliability is medium due to the number of hits over 3500 but not much higher than 5000.
- 

- 

Indices to query

packetbeat-*

Use * to broaden your query.

Time field

@timestamp

Run watch every

1

minute

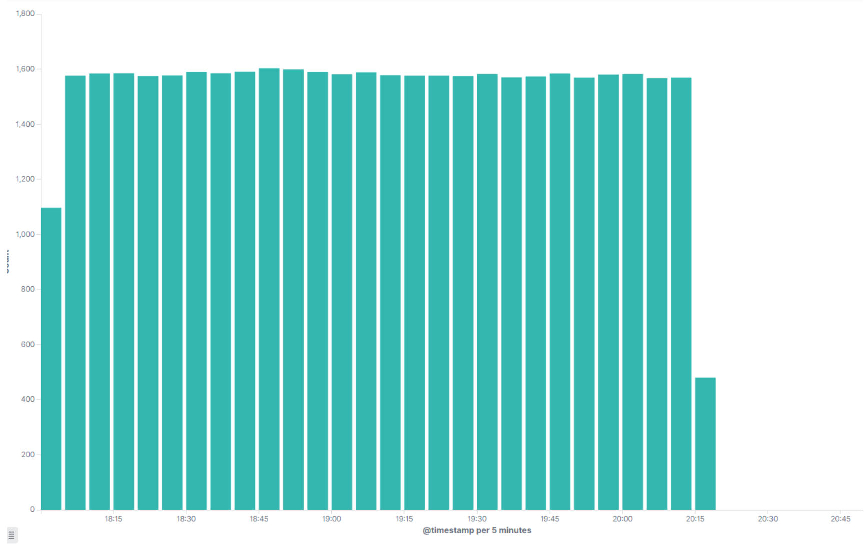
Match the following condition

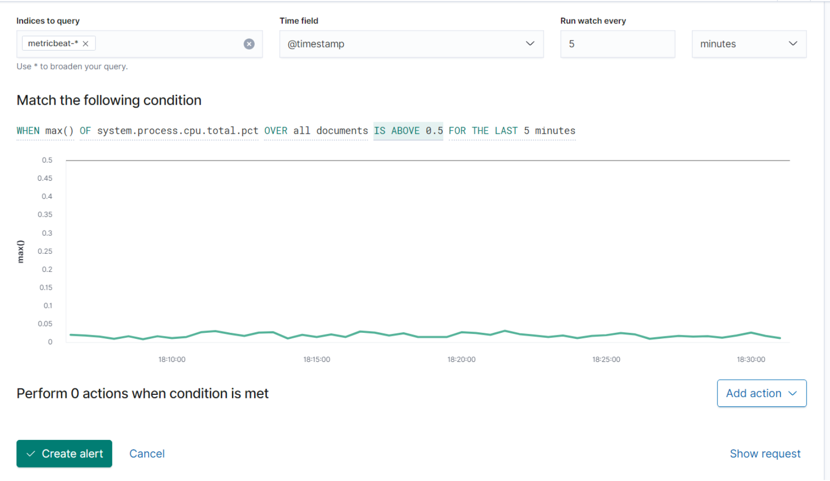
```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```

Perform 0 actions when condition is met

Name of Alert 3:

CPU Usage Monitor is implemented as follows:

- **Metric**: system.process.cpu.total.pct
- **Threshold**: 0.5 usage every 5 minutes
- **Vulnerability Mitigated**: Malware
- **Reliability**: TODO: This alert will generate a lot of false positives. I would rate this alert low.
- 



Indices to query

metricbeat-* ×

Use * to broaden your query.

Time field

@timestamp

Run watch every

5

minutes

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 0 actions when condition is met