Blue Team: Summary of Operations

Table of Contents

Network Topology

The following machines were identified on the network using ipconfig/all and cat /etc/os-release for OS information:

- Host Machine Containing HyperV Network ML-RefVm-684427
  - **Windows 10**
  - **Host machine containing virtual network through HyperV**
  - **IP Address: 192.168.1.1**
- Kali
  - **Linux Kali GNU/Linux 2020.1**
  - **Kali installation that is used for attacking**
  - **IP Address: 192.168.1.90**
- Capstone
  - **Linux Ubuntu 18.04.1**
  - **Vulnerable target VM used to test alerts, forwards Filebeat and Metricbeat to Elk Machine**
  - **IP Address**: **192.168.1.105**
- ELK
  - **Linux Ubuntu 18.04.4**
  - **Holds Kibana dashboards that are used to monitor potential attacks to the system**
  - **IP Address**: **192.168.1.100**
- Target 1
  - **Linux Debian GNU/Linux 8**
  - **Target machine with vulnerable Wordpress server, logs sent to ELK**
  - **IP Address**: **192.168.1.110**
- Target 2
  - **Linux Debian GNU/Linux 8**
  - **More Difficult Target machine with vulnerable Wordpress server, logs sent to ELK**

    - **IP Address**: **192.168.1.115**

Description of Targets

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

There was a 2nd Target, Target 2 - 192.168.1.115, but it was not attacked in this project.

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:

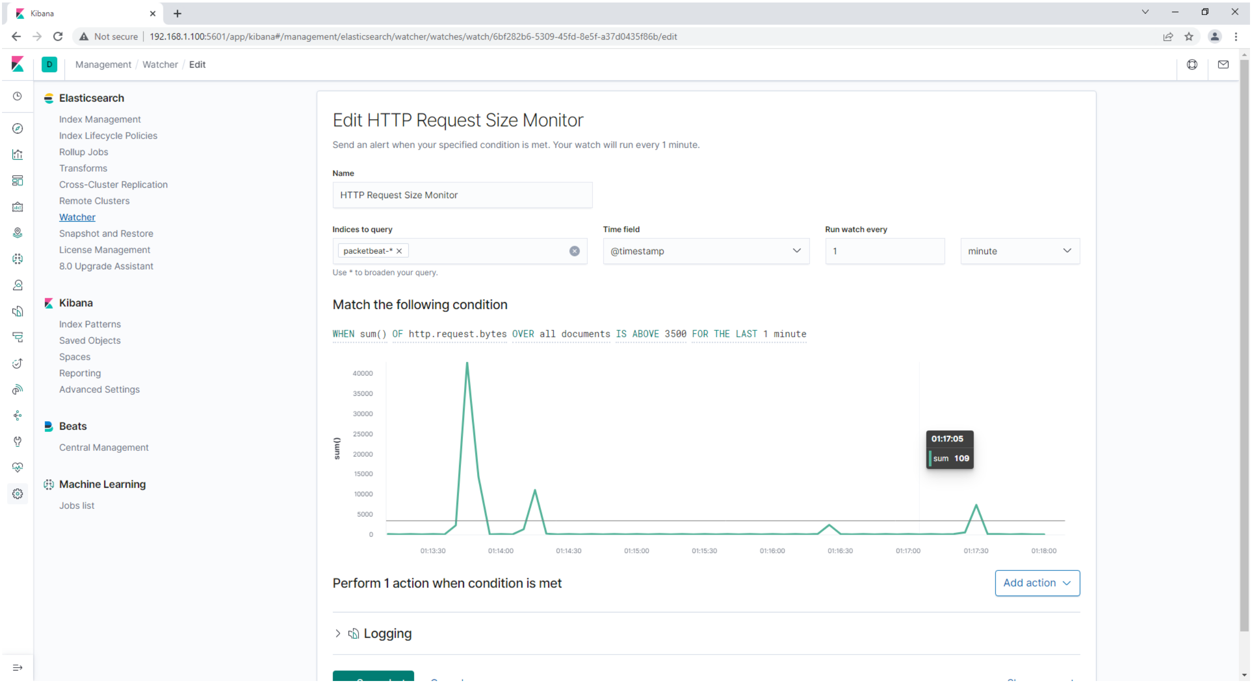
Monitoring the Targets

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

HTTP Request Size Monitor

Alert 1 is implemented as follows:

- **Metric**: Sum of http.request.bytes
- **Threshold**: Above 3500 for 1 minute
- **Vulnerability Mitigated**: Attempts to map the network using NMAP or using Wordpress scans to pull data from that application, could also possibly indicate Denial of Service attacks.
- **Reliability**: Because the threshold is for a short period of time, reliability should be high due to the fact denial of service and network mapping/scanning is generally run automatically in short periods of time



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

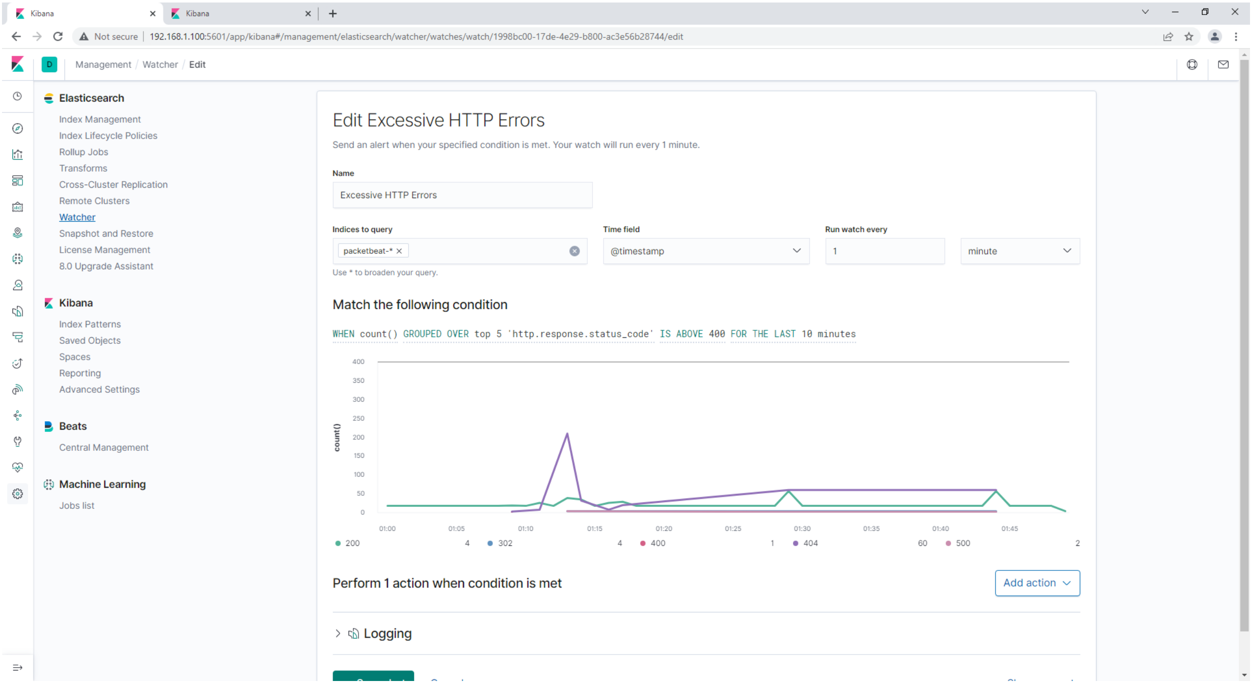
```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```

Perform 1 action when condition is met

Excessive HTTP Errors

Alert 2 is implemented as follows:

- **Metric**: Count of grouped over top 5 http.response.status_code
- **Threshold**: Above 400 for the last 5 minutes
- **Vulnerability Mitigated**: Bruteforce attacks, or other attempts to access resulting in a high number of failed access attempts
- **Reliability**: This alert is set with a high enough threshold that it will filter out general errors, and only alert when suspicious activity is occurring, allowing for high reliability



Edit Excessive HTTP Errors

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

Name: Excessive HTTP Errors

Indices to query: packetbeat-*

Use * to broaden your query.

Time field: @timestamp

Run watch every: 1 minute

Match the following condition

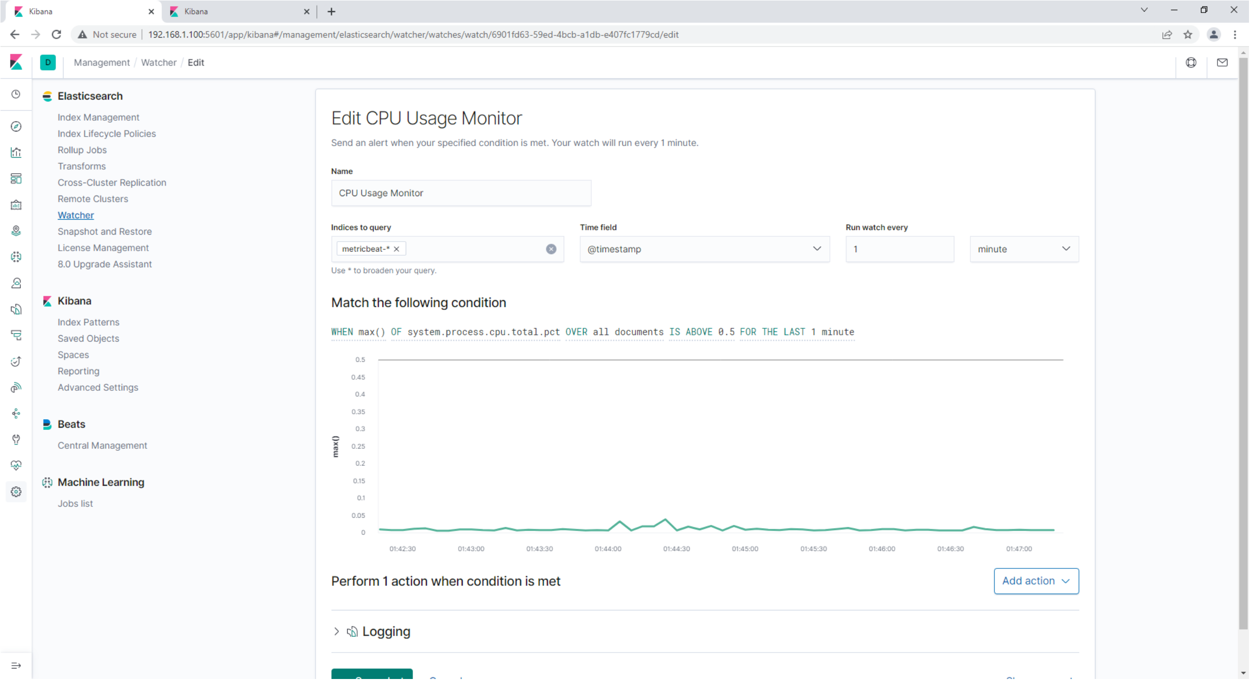
```
WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 10 minutes
```

Perform 1 action when condition is met

CPU Usage Monitor

Alert 3 is implemented as follows:

- **Metric**: Max number of system.process.cpu.total.pct over all documents
- **Threshold**: Above .5 for the last 1 minute
- **Vulnerability Mitigated**: Possible malware or other activity causing a spike in CPU usage, viruses or other attacking using processing power
- **Reliability**: Depending on usage, this could generate false positives for general use if authorized users are using programs requiring a lot of computing power, thresholds and timeframes should be adjusted to make sure resources are only being used when they should, current reliability could catch certain attacks, but is probably still low due to configurations.



192.168.1.100:5601/app/kibana#/management/elasticsearch/watcher/watches/watch/6901fd63-59ed-4bcb-a1db-e407fc1779cd/edit

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 1 minute
```

Perform 1 action when condition is met

Suggestions for Going Further

- 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, suggest a patch. E.g., implementing a blocklist is an effective tactic against brute-force attacks. It is not necessary to explain *how* to implement each patch.

The logs and alerts generated during the assessment suggest that this network is susceptible to several active threats, identified by the alerts above. In addition to watching for occurrences of such threats, the network should be hardened against them. The Blue Team suggests that IT implement the fixes below to protect the network:

- Malware or other malicious programs
  - **Patch**: Make sure all security programs are up to date and reliable software is installed
  - **Why It Works**: Security programs are kept up to date as long as they are updated regularly, this can be done automatically and will watch for any malware or malicious programing that is running on a system
- NMAP Scanning or Wordpress Scanning
  - **Patch**: Block network mapping and scanning and make sure the latest patched versions of wordpress are installed
  - **Why It Works**: This will block attempts to map the network and to enumerate wordpress information to get recon for possible exploits, preventing these simple attempts to gain enough reconnaissance to attempt to exploit a system
- Bruteforce Password Attacks

- **Patch**: Limit attempts for failed login attempts to 10 in a 15 minute period (or similar limit depending on company needs) before implementing a lockout/reset requirement
- **Why It Works**: This will prevent bruteforce password attacks by locking the username from further attempts after the limit is reached, so a continuous bruteforce attack cannot be run on a single username without locking the account and requiring a reset