

Blue Team: Summary of Operations

Table of Contents

- Network Topology
- Description of Targets
- Monitoring the Targets
- Patterns of Traffic & Behavior
- Suggestions for Going Further

Network Topology

The following machines were identified on the network:

- **Kali**
 - **Operating System:** Debian Kali 5.4.0
 - **Purpose:** The Penetration Tester (Used to attack other machines)
 - **IP Address:**192.168.1.90
- **ELK**
 - **Operating System:**Ubuntu 18.04
 - **Purpose:**The ELK (Elasticsearch & Kibana) Stack
 - **IP Address:**192.168.1.100
- **Target 1**
 - **Operating System:** Debian GNU/Linux 8
 - **Purpose:** Exposes vulnerable WordPress server. Sends logs to ELK
 - **IP Address:**192.168.1.110/24
- **Target 2**
 - **Operating System:** Debian GNU/Linux 8
 - **Purpose:** Web Server (WordPress Host)
 - **IP Address:**192.168.1.115/24
- **Capstone**
 - **Operating System:** Ubuntu 18.04
 - **Purpose:**The Vulnerable Web Server
 - **IP Address:**192.168.1.105

Description of Targets

Two VMs on the network were vulnerable to attack: Target 1 (192.168.1.110) and Target 2 (192.168.1.115). Only Target 1 is covered and was attacked.

Each VM functions as an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers.

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: CPU usage monitoring, Excessive http errors, Http request size monitoring.

Monitoring the Targets

This scan identifies the services below as potential points of entry: `nmap -A 192.168.1.90/24`

Target 1

- Port 22/TCP Open SSH OpenSSH 6.7p1 Debian 5+deb8u4
- Port 80/TCP Open HTTP Apache httpd 2.4.10 (Debian)

```
Nmap scan report for 192.168.1.110
Host is up (0.00080s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
|_ ssh-hostkey:
|   1024 26:81:c1:f3:5e:01:ef:93:49:3d:91:1e:ae:8b:3c:fc (DSA)
|   2048 31:58:01:19:4d:a2:80:a6:b9:0d:40:98:1c:97:aa:53 (RSA)
|   256 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0 (ECDSA)
|_ 256 0e:85:71:a8:a2:c3:08:69:9c:91:c0:3f:84:18:df:ae (ED25519)
80/tcp    open  http         Apache httpd 2.4.10 ((Debian))
|_ _http-server-header: Apache/2.4.10 (Debian)
|_ _http-title: Raven Security
111/tcp   open  rpcbind      2-4 (RPC #100000)
|_ rpcinfo:
|   program version    port/proto  service
|   100000   2,3,4      111/tcp     rpcbind
|   100000   2,3,4      111/udp     rpcbind
|   100000   3,4        111/tcp6    rpcbind
|   100000   3,4        111/udp6    rpcbind
|   100024   1          37667/tcp   status
|   100024   1          42075/udp   status
|   100024   1          49788/udp6  status
|_  100024   1          59693/tcp6  status
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 4.2.14-Debian (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ _clock-skew: mean: -3h19m59s, deviation: 5h46m24s, median: 0s
|_ _nbstat: NetBIOS name: TARGET1, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
```

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

Excessive HTTP Errors

Excessive HTTP Errors is implemented as follows:

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

Metric:

WHEN count() GROUPED OVER top 5 'http.response.status_code'

Threshold:

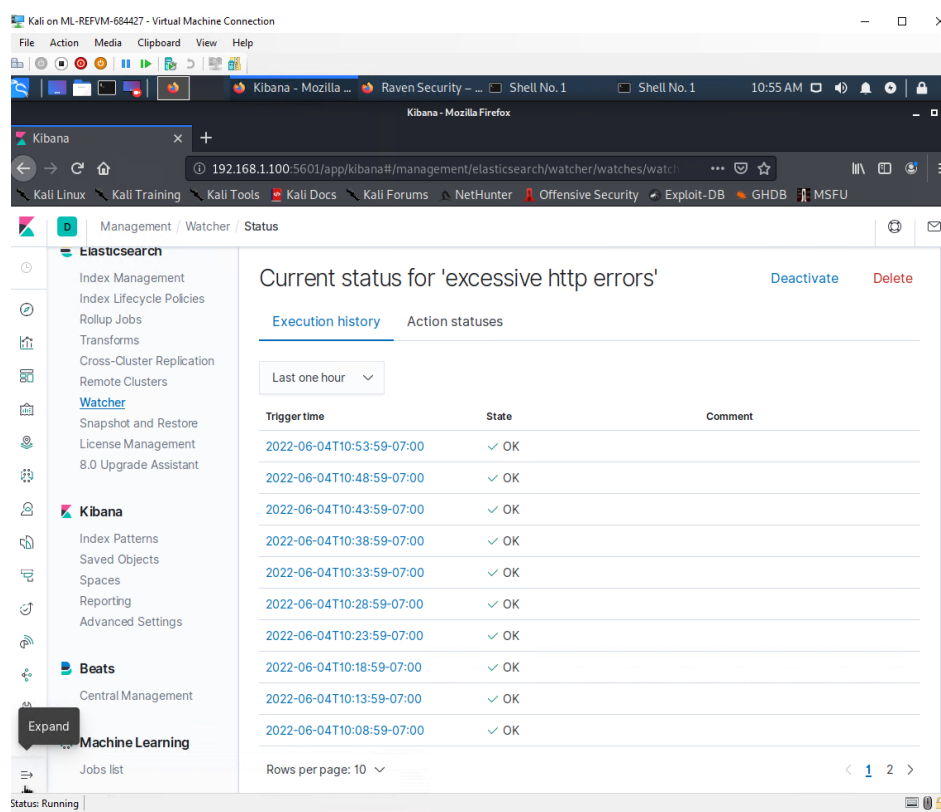
IS ABOVE 400

Vulnerability Mitigated:

Enumeration/Brute Force

Reliability:

The alert is highly reliable. Measuring by error codes 400 and above will filter out any normal or successful responses. 400+ codes are client and server errors which are of more concern. Especially when taking into account these error codes going off at a high rate.



HTTP Request Size Monitor

HTTP Request Size Monitor is implemented as follows:

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

Metric:

WHEN sum() of http.request.bytes OVER all documents

Threshold:

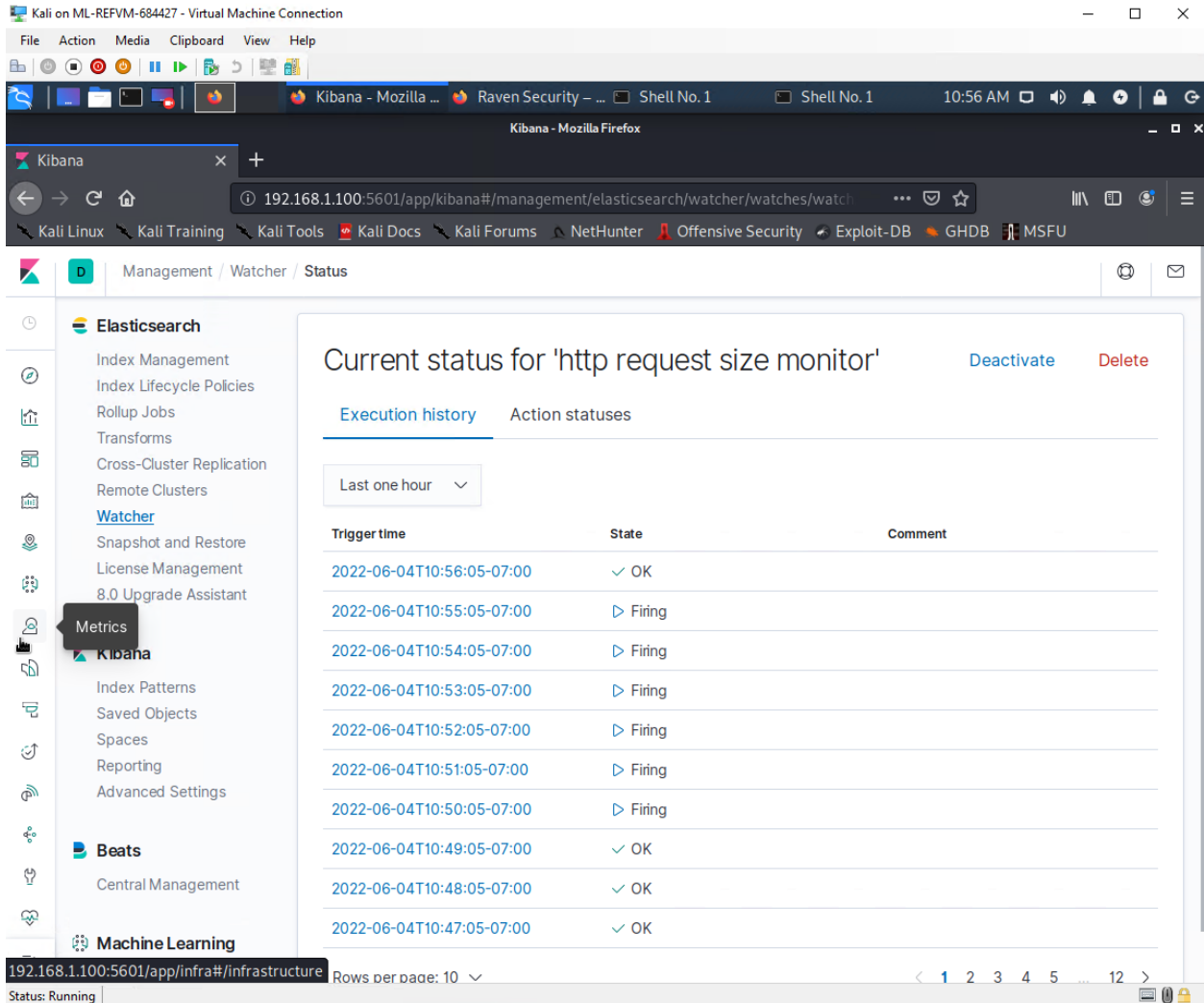
IS ABOVE 3500

Vulnerability Mitigated:

Code injection in HTTP requests (XSS and CRLF) or DDOS

Reliability:

Alert could create false positives. It comes in at medium reliability. There is a possibility for a large non malicious HTTP request or legitimate HTTP traffic.



The screenshot shows the Kibana interface in a browser window. The left sidebar contains the navigation menu with categories like Elasticsearch, Metrics, Beats, and Machine Learning. The main content area displays the 'Current status for 'http request size monitor'' page. It includes a 'Deactivate' button and a 'Delete' button. Below this, there is a table showing the execution history of the watch. The table has columns for 'Trigger time', 'State', and 'Comment'. The status is 'Running' at the bottom left.

Trigger time	State	Comment
2022-06-04T10:56:05-07:00	✓ OK	
2022-06-04T10:55:05-07:00	▷ Firing	
2022-06-04T10:54:05-07:00	▷ Firing	
2022-06-04T10:53:05-07:00	▷ Firing	
2022-06-04T10:52:05-07:00	▷ Firing	
2022-06-04T10:51:05-07:00	▷ Firing	
2022-06-04T10:50:05-07:00	▷ Firing	
2022-06-04T10:49:05-07:00	✓ OK	
2022-06-04T10:48:05-07:00	✓ OK	
2022-06-04T10:47:05-07:00	✓ OK	

CPU Usage Monitor

CPU Usage Monitor is implemented as follows:

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

Metric:

WHEN max() OF system.process.cpu.total.pct OVER all documents

Threshold:

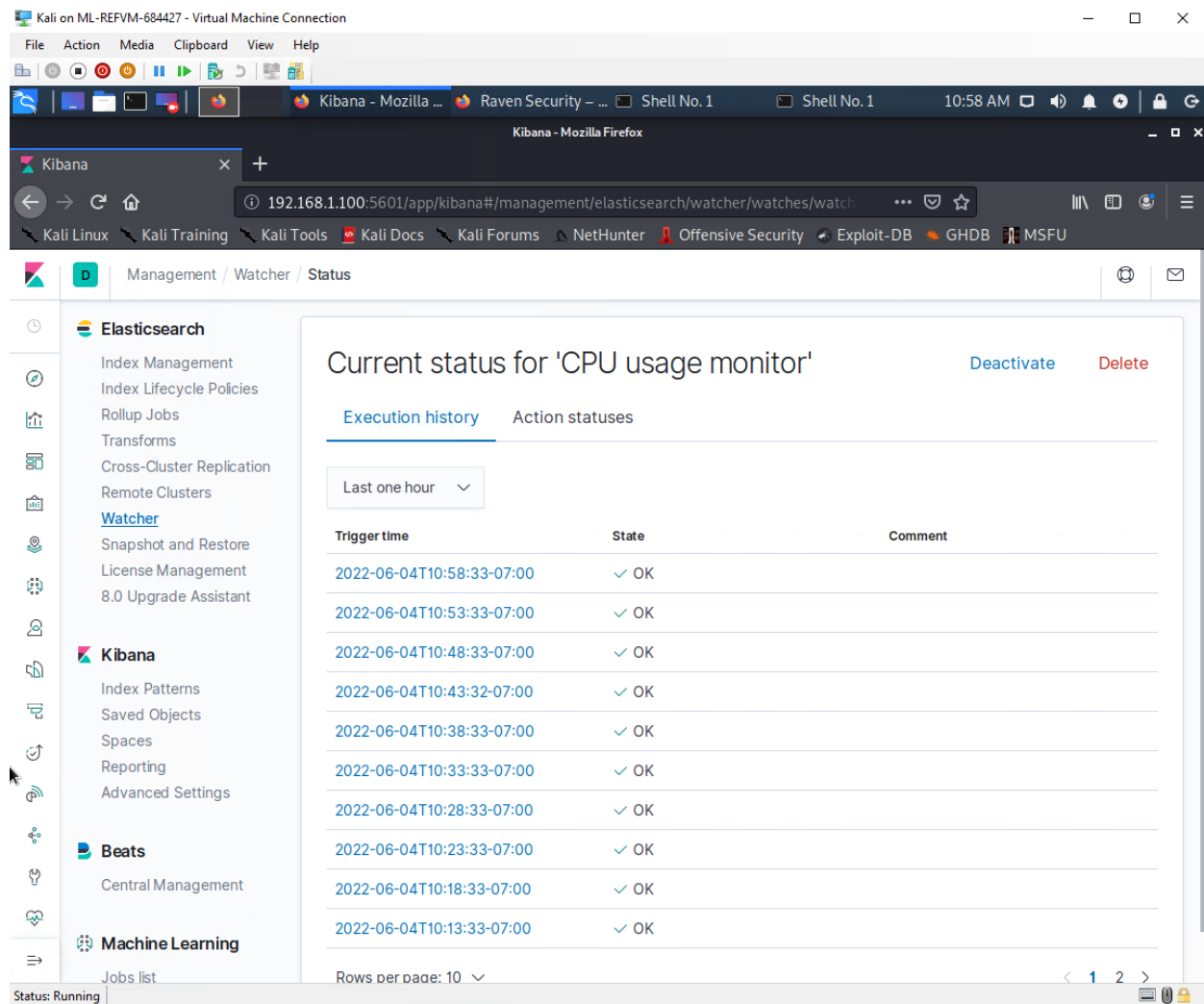
IS ABOVE 0.5

Vulnerability Mitigated:

Malicious software, programs (malware or viruses) running taking up resources

Reliability:

The alert is highly reliable. Even if there isn't a malicious program running this can still help determine where to improve on CPU usage.



The screenshot shows a web browser window with the Kibana interface. The browser's address bar displays the URL `192.168.1.100:5601/app/kibana#/management/elasticsearch/watcher/watches/watcher`. The Kibana interface has a sidebar on the left with a navigation menu. The main content area is titled "Current status for 'CPU usage monitor'" and includes a "Deactivate" button and a "Delete" button. Below the title, there are two tabs: "Execution history" (selected) and "Action statuses". The "Execution history" tab shows a table of execution events for the last one hour. The table has three columns: "Trigger time", "State", and "Comment". All events show a state of "OK".

Trigger time	State	Comment
2022-06-04T10:58:33-07:00	✓ OK	
2022-06-04T10:53:33-07:00	✓ OK	
2022-06-04T10:48:33-07:00	✓ OK	
2022-06-04T10:43:32-07:00	✓ OK	
2022-06-04T10:38:33-07:00	✓ OK	
2022-06-04T10:33:33-07:00	✓ OK	
2022-06-04T10:28:33-07:00	✓ OK	
2022-06-04T10:23:33-07:00	✓ OK	
2022-06-04T10:18:33-07:00	✓ OK	
2022-06-04T10:13:33-07:00	✓ OK	

Rows per page: 10

