## **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
  + **IP Address**:
    - 192.168.1.90
* ELK
  + **Operating System**:
    - Linux Ubuntu
  + **Purpose**:
    - The ELK (Elasticsearch and Kibana) Stack
  + **IP Address**:
    - 192.168.1.100
* Capstone
  + **Operating System:**
    - Ubuntu 18.04
  + **Purpose**:
    - HTTP Server
  + **IP Address:**
    - 192.168.1.105
* Target 1
  + **Operating System:**
    - Linux Debian
  + **Purpose:**
    - HTTP Server / WordPress
  + **IP Address:**
    - 192.168.1.110
* Target 2
  + **Operating System:**
    - Linux Debian
  + **Purpose:**
    - HTTP Server
  + **IP Address:**
    - 192.168.1.115

### **Description of Targets**

The target of this attack was: Target 1 (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**

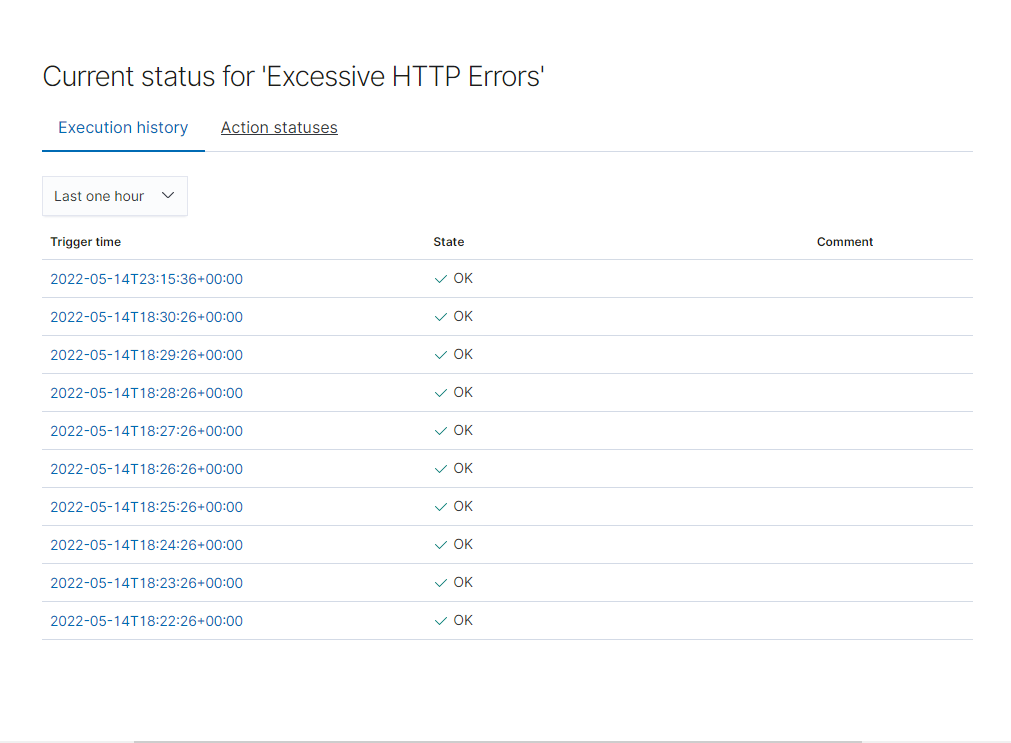
Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

**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.**

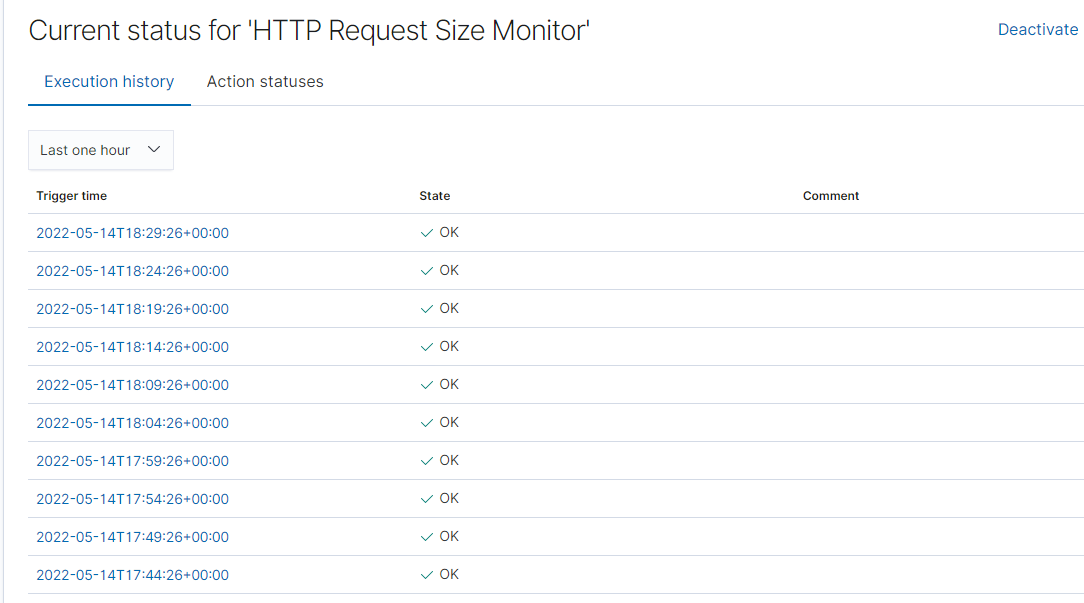
****

#### **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 a medium reliability. There is a possibility for a large non malicious HTTP request or legitimate HTTP traffic.**



**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.**

