# **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
  + **IP Address**:
    - 192.168.1.90
* Capstone
  + **Operating System**:
    - Ubuntu 18.04
  + **Purpose**:
    - The Vulnerable Web Server
  + **IP Address**:
    - 192.168.1.105
* ELK
  + **Operating System**:
    - Ubuntu 18.04
  + **Purpose**:
    - The ELK (Elasticsearch and Kibana) Stack
  + **IP Address**:
    - 192.168.1.100
* Target 1
  + **Operating System**:
    - Debian GNU/Linux 8
  + **Purpose**:
    - The WordPress Host
  + **IP Address**:
    - 192.168.1.110

### **Description of Targets**

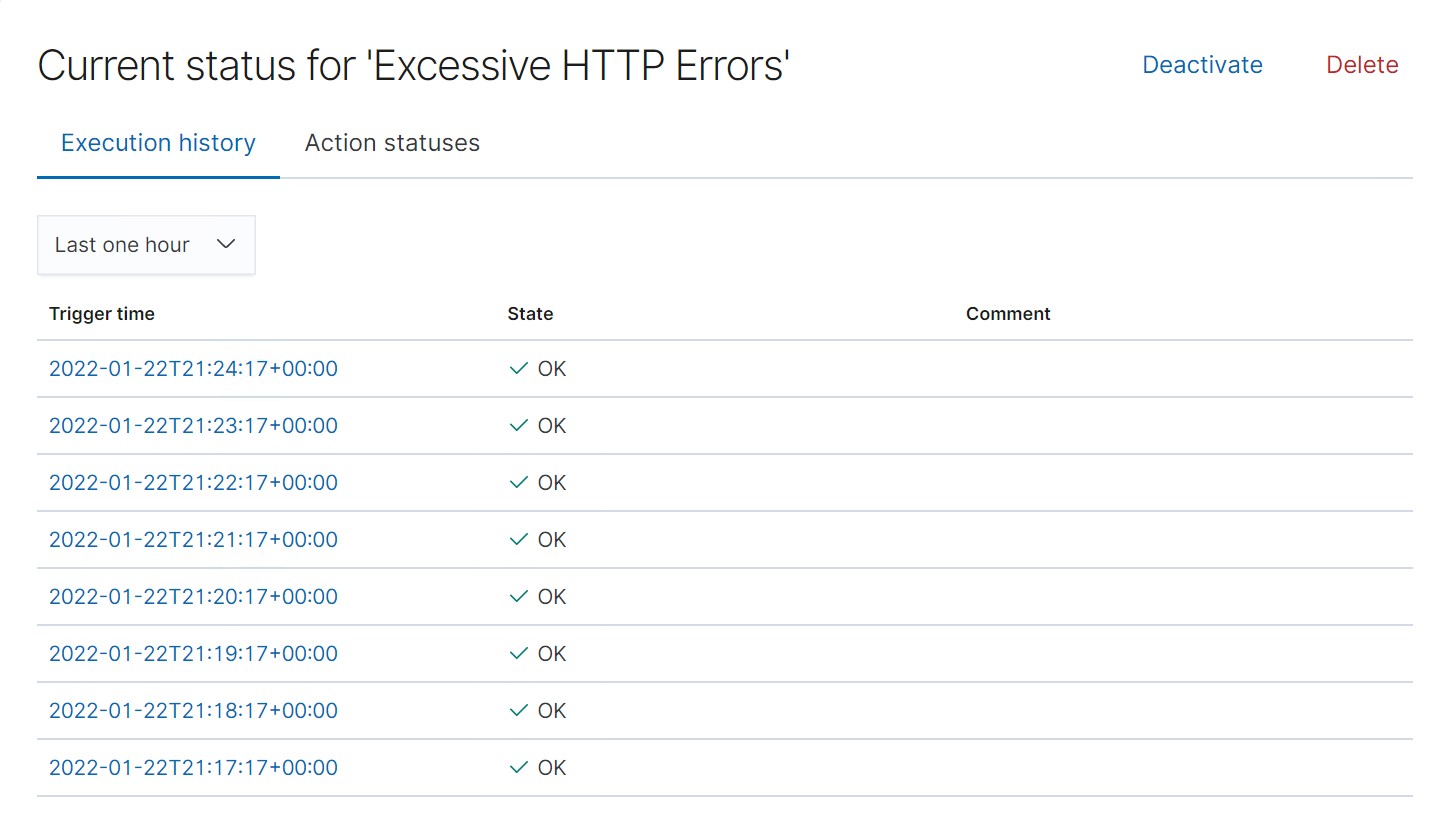
The target of this attack was: Target 1 (192.168.1.110).

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:

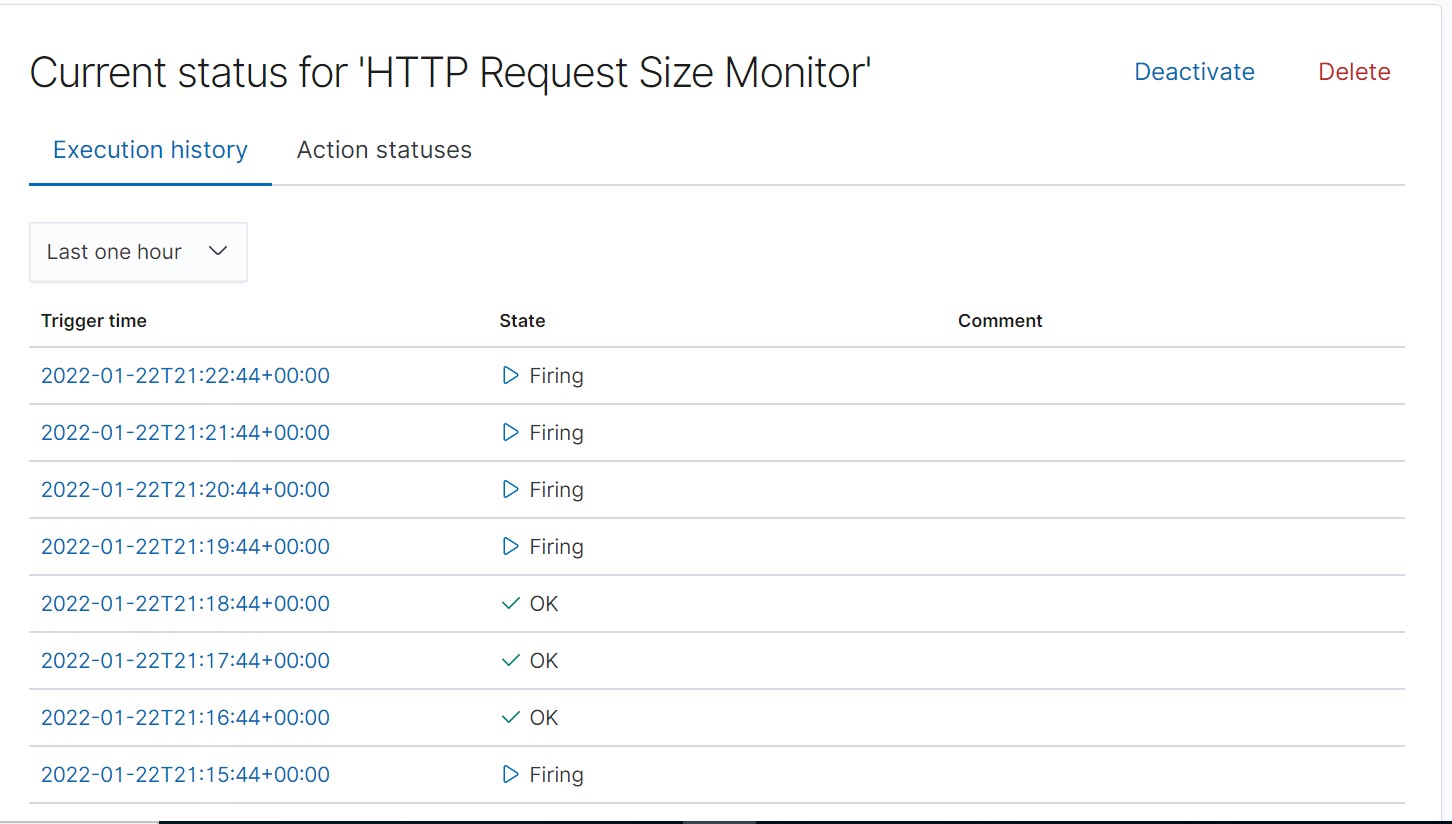
#### **Excessive HTTP Errors**



Excessive HTTP Errors is implemented as follows:

* **Metric**: WHEN count() GROUPED OVER top 5 ‘http.response.status\_code
* **Threshold**: IS ABOVE 400
* **Vulnerability Mitigated**: Enumeration/Brute Force
* **Reliability**: This alert is highly reliable. Measuring by error codes 400 and above will filter out any other codes, which are generally for normal or successful responses. 400+ codes are also client and server errors, which are of more concern for us. This is especially important in the scenario of these error codes going off at a high rate

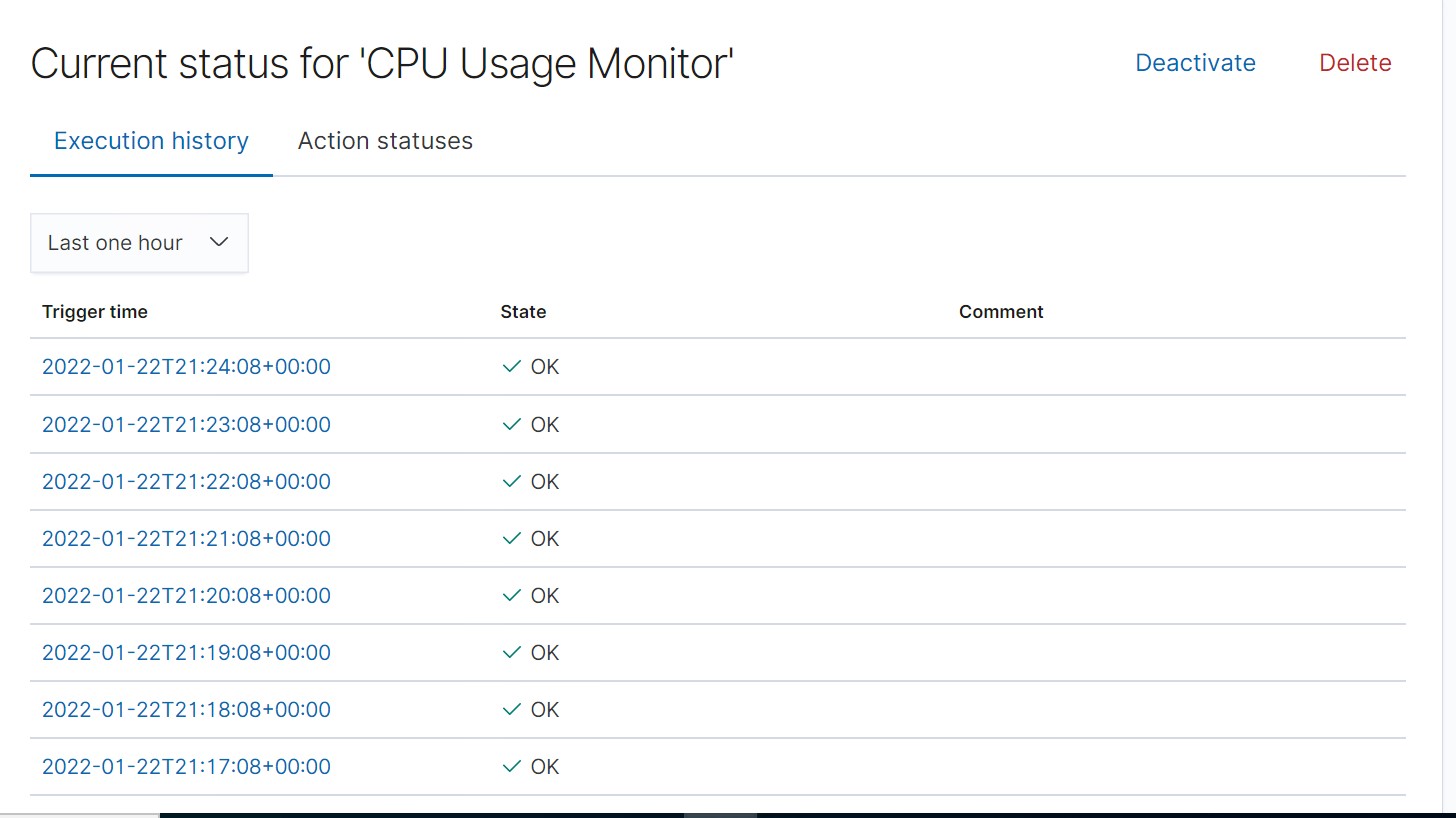
#### **HTTP Request Size Monitor**



HTTP Request Size Monitor is implemented as follows:

* **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**: Medium reliability. This alert could create false positives; a lot of non malicious HTTP requests could just be legitimate HTTP traffic.

#### **CPU Usage Monitor**



CPU Usage Monitor is implemented as follows:

* **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 and taking up resources
* **Reliability**: High. Even if there isn’t a malicious program running and taking up resources, this alert can still help us determine where to improve on CPU usage.