# Blue Team: Summary of Operations

## Table of Contents

* Network Topology
* Description of Targets
* Monitoring the Targets
* Patterns of Traffic and Behavior
* Suggestions for Going Further

## Network Topology

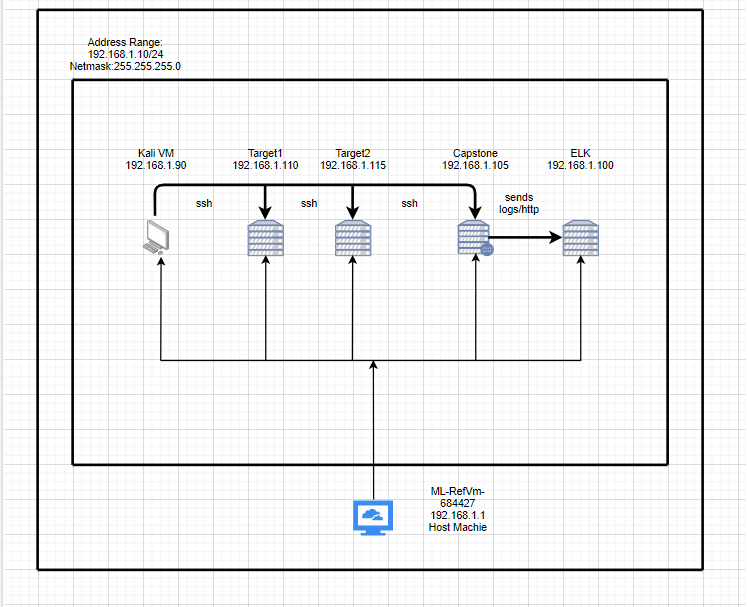
The following machines were identified on the network:

**[Name of VM 1]**

* Operating System: Debian/Linux 8 (jessie)
* Purpose: Apache Server
* IP Address: 192.168.1.110

**[Name of VM 2]**

* Operating System: Debian/Linux 8 (jessie)
* Purpose: Apache Server
* IP Address: 192.168.1.115



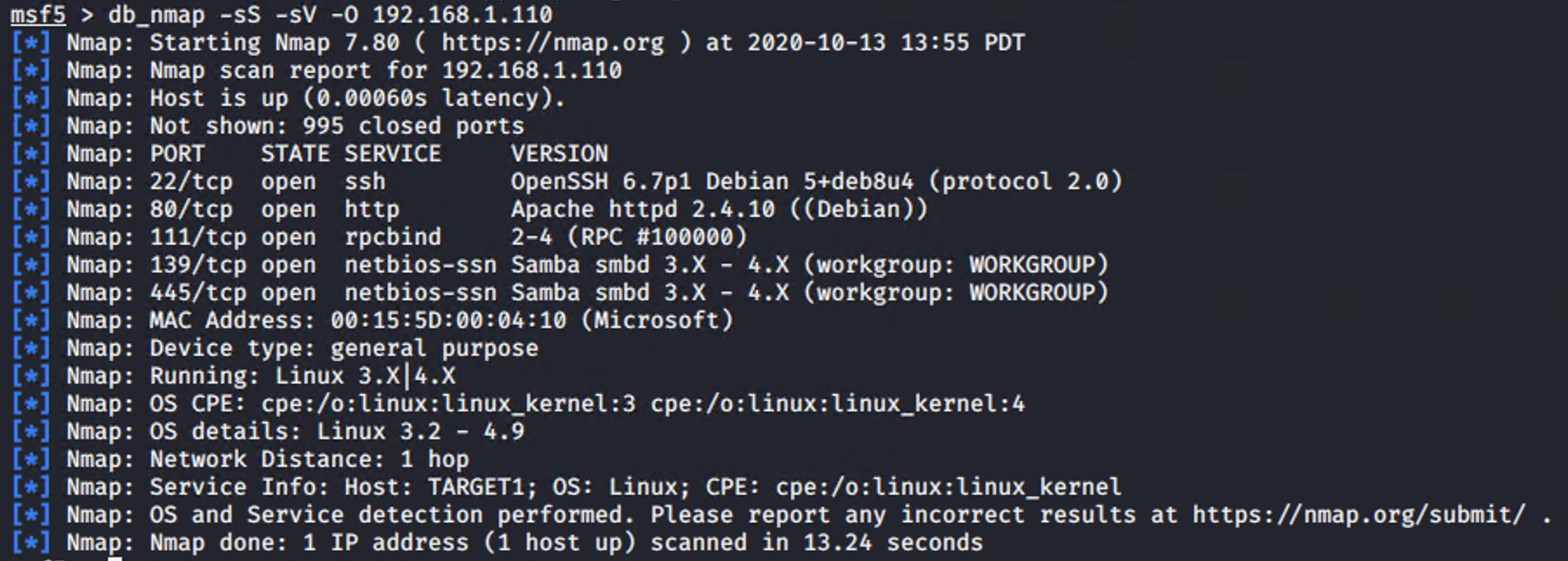
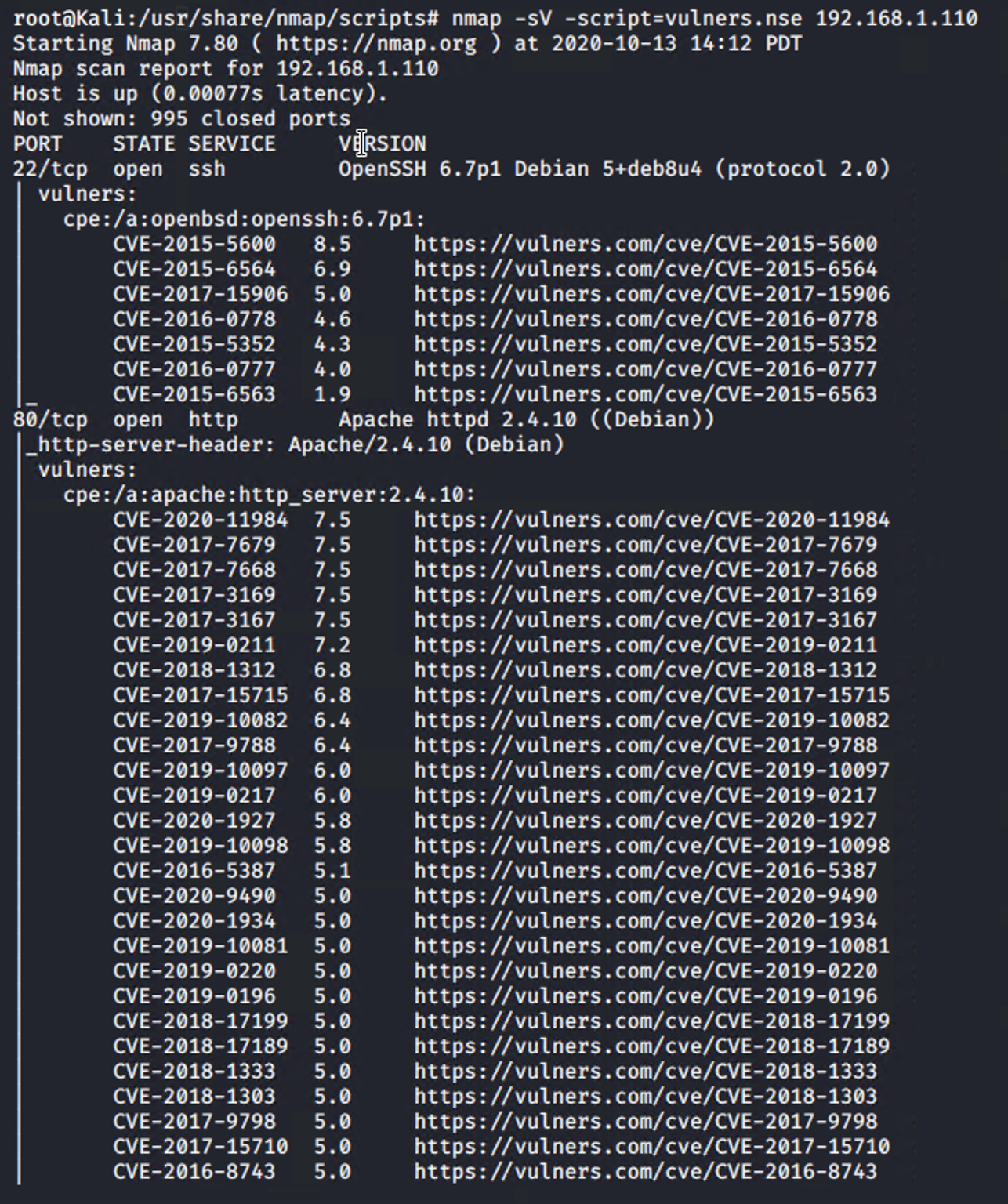
## Description of Targets

Fill in the following:

* Two VMs on the network were vulnerable to attack: Target 1 IP: 192.168.1.110
* Each VM functions as an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers.

## Monitoring the Targets

This scan identifies the services below as potential points of entry:

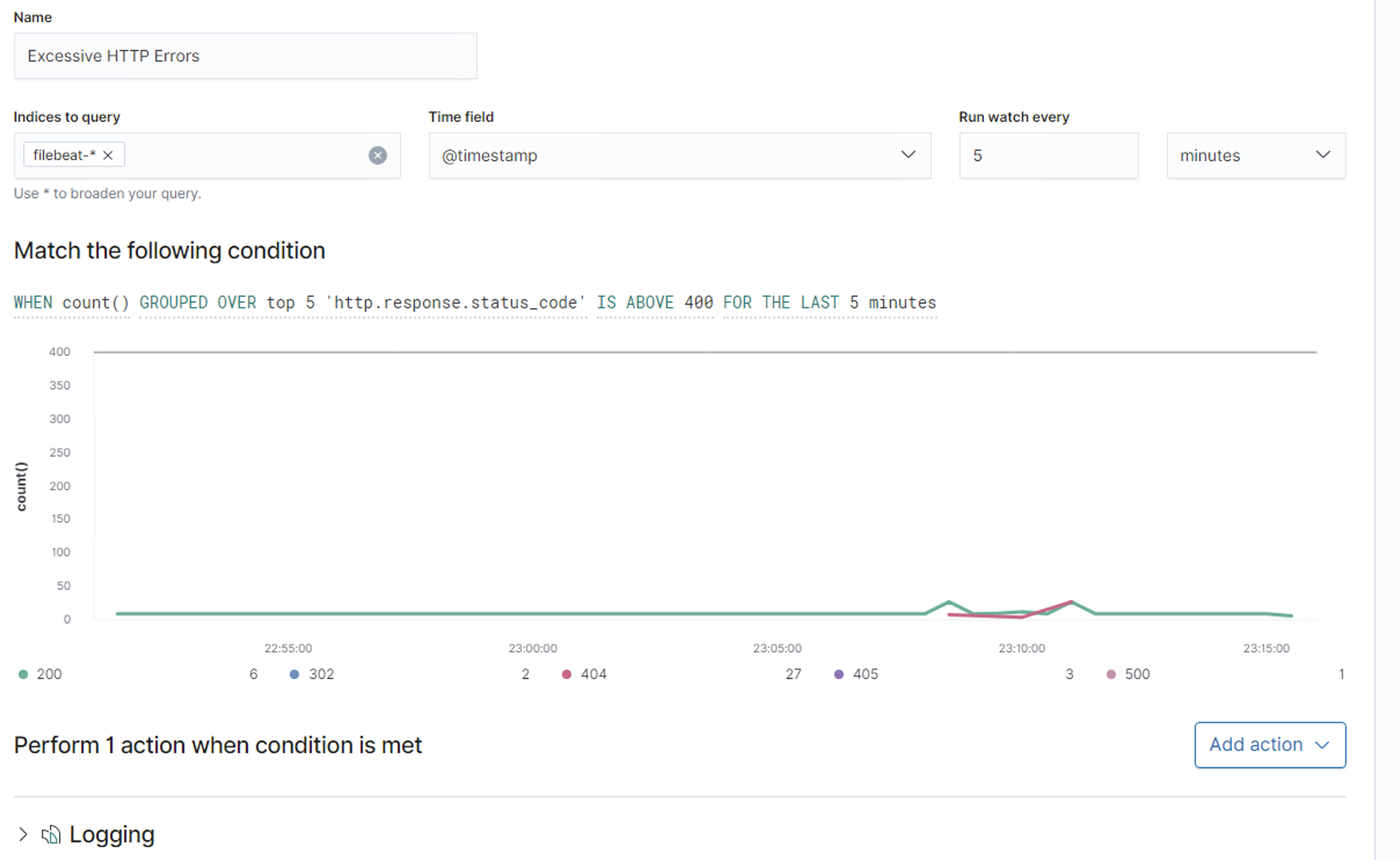
* **Target 1**
  + 
  + 
* **Target 2**
  + List of
  + Potentially vulnerable
  + Services

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below: (Note: Add at least three alerts. You can add more if time allows.)

**Name of Alert 1**

Excessive HTTP Errors is implemented as follows:

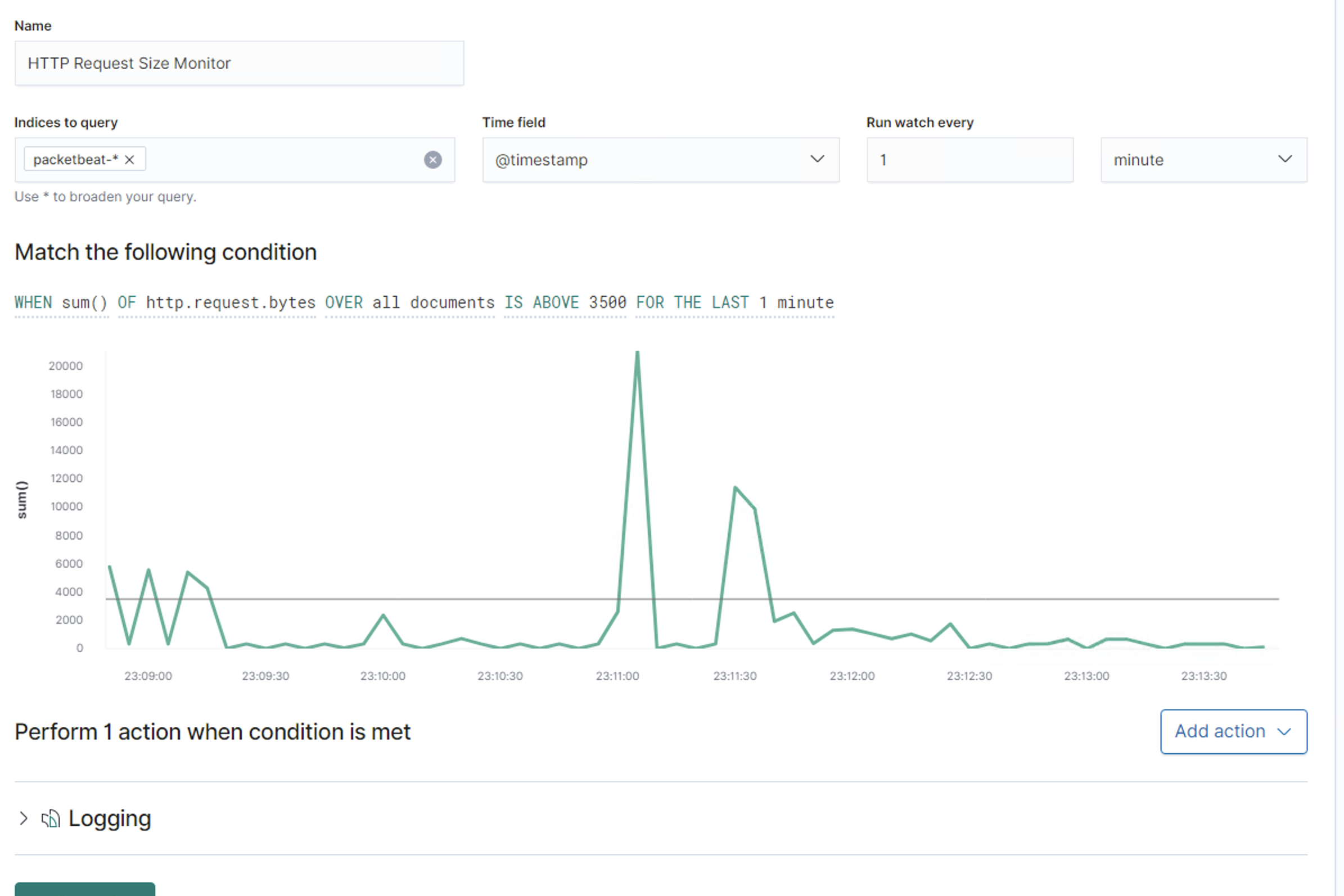
* Metric: WHEN count() GROUPED OVER top 5 ‘http.response.status\_code’
* Threshold: IS ABOVE 400 FOR THE LAST 5 minutes
* Vulnerability Mitigated: No
* Reliability: Medium

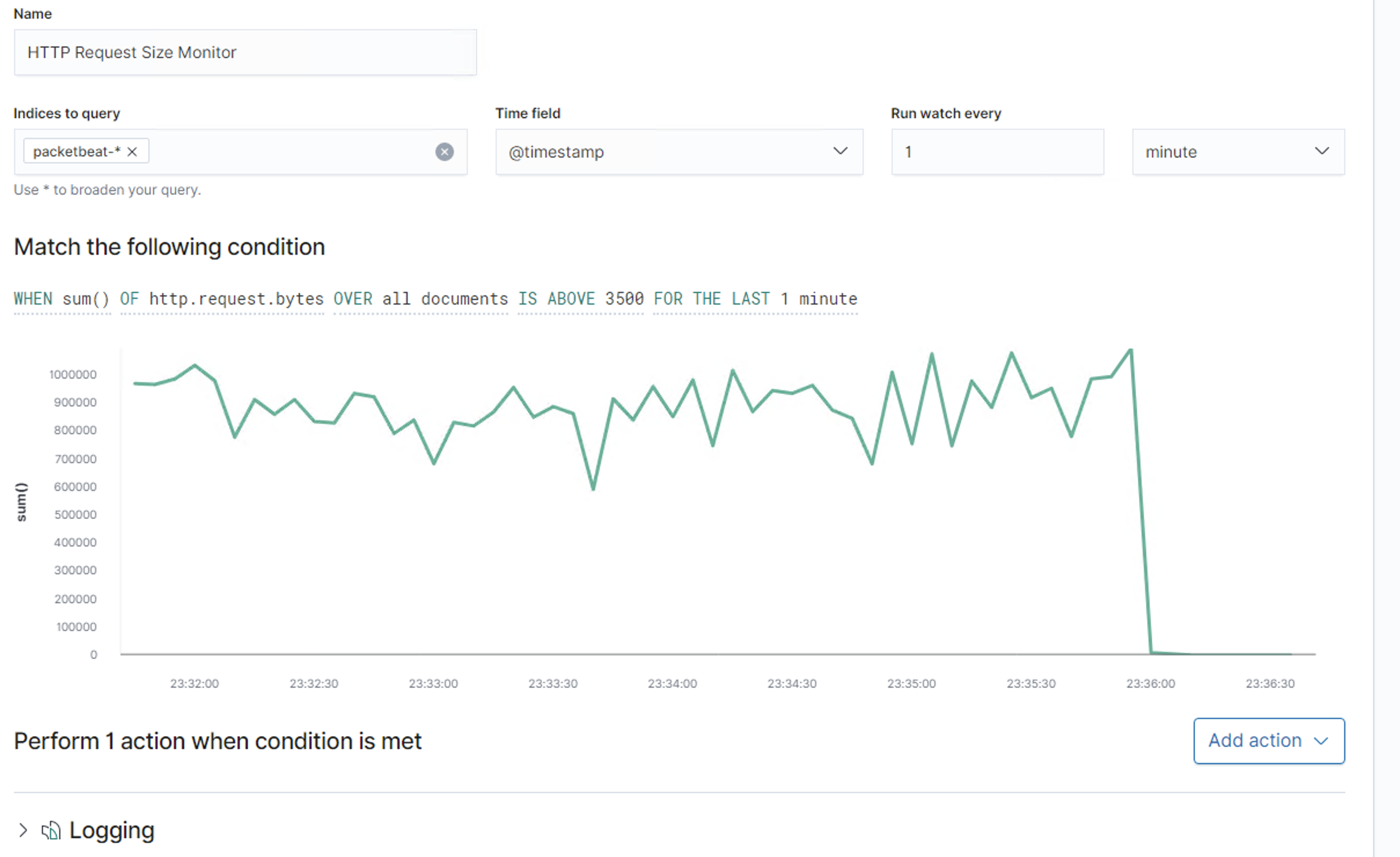


**Name of Alert 2**

HTTP Request Size Monitor is implemented as follows:

* Metric: WHEN sum() OF http.request.bytes OVER all documents
* Threshold: IS ABOVE 3500 FOR THE LAST 1 minute
* Vulnerability Mitigated: No
* Reliability: Medium





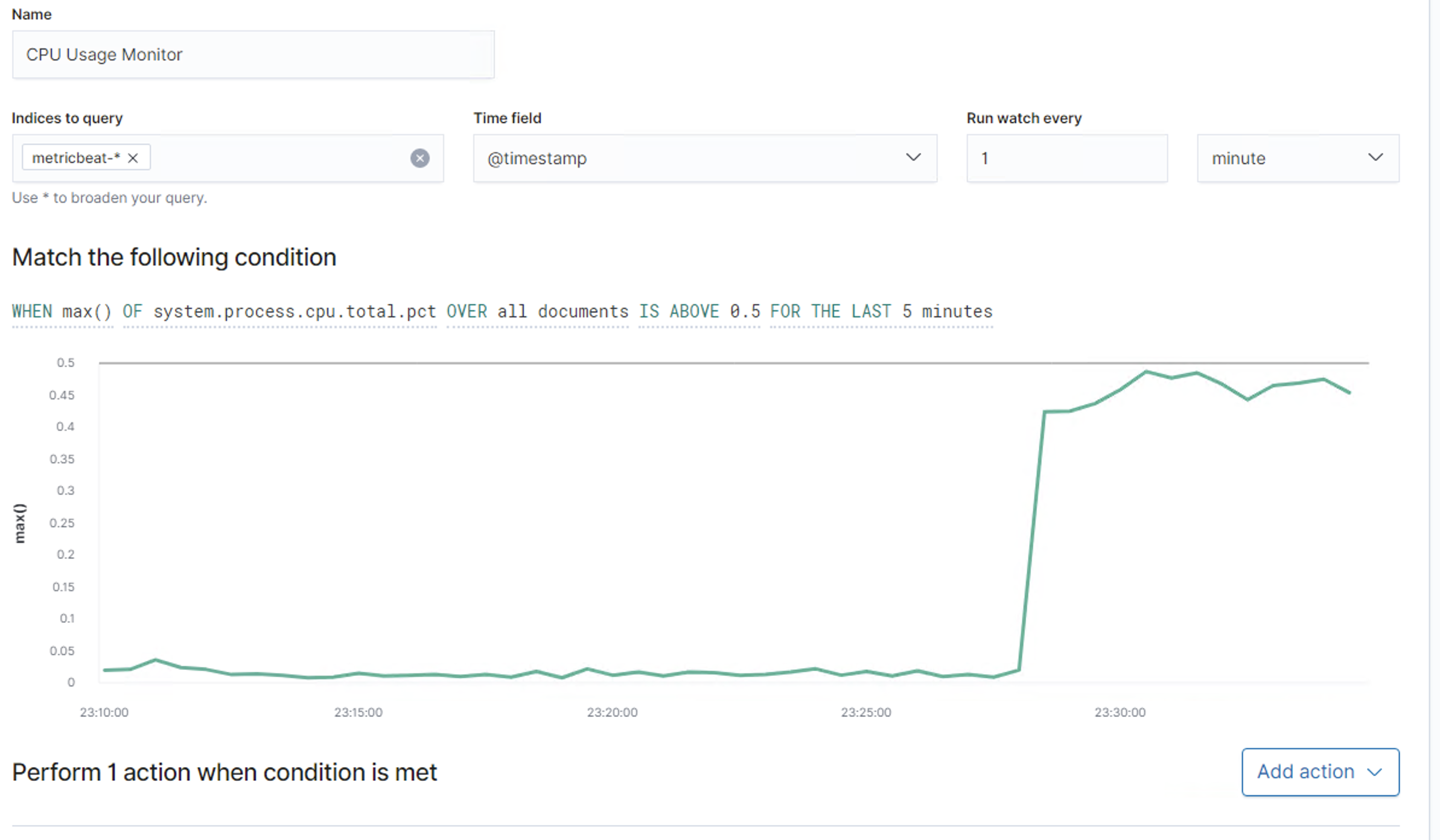


**Name of Alert 3**

CPU Usage Monitor is implemented as follows:

* Metric: WHEN max() OF system.process.cpu.total.pct OVER all documents
* Threshold: IS ABOVE 0.5 FOR THE LAST 5 minutes
* Vulnerability Mitigated: No
* Reliability: Medium





## Suggestions for Going Further

**Suggest a patch for each vulnerability identified by the alerts above.** Remember: alerts only detect malicious behavior. They do not prevent it.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. 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:

**Vulnerability 1:** CVE-2015-5600

* Patch: Install recent security update 2015-006 on platform

(Vendor: openbsd, Product: openssh, Version: 6.9)

* Why It Works: This update will prevent Permissions, Privileges, and Access Controls from allowing remote attackers to conduct brute-force attacks or cause a denial of service (CPU consumption) using a long and duplicative list in the -oKbInteractiveDevices command option.

Source: https://kc.mcafee.com/corporate/index?page=content&id=SB10157

**Vulnerability 2:** CVE-2020-11984

* Patch: Security Update for apache2 ( -openSUSE Leap 15.1 (i586 x86\_64):

(Vendor: apache, Product: http\_server, Version: 2.4.44)

Source: https://lists.opensuse.org/opensuse-security-announce/2020-08/msg00071.html

* Why It Works: Fixed an information disclosure bug in mod\_proxy\_uwsgi and prevents modification of data or Denial Of Service

Source: https://security.netapp.com/advisory/ntap-20200814-0005/

**Vulnerability 3:** CVE-2017-7679

* Patch: httpd24-httpd security update, package updates
* Why It Works:
  + Description: In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, mod\_mime can read one byte past the end of a buffer when sending a malicious Content-Type response header.
  + Fixes A buffer over-read flaw was found in the httpd's mod\_mime module. A user permitted to modify httpd's MIME configuration could use this flaw to cause httpd child process to crash.

Source: <https://www.securitytracker.com/id/1038711>

https://access.redhat.com/errata/RHSA-2017:2483