Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

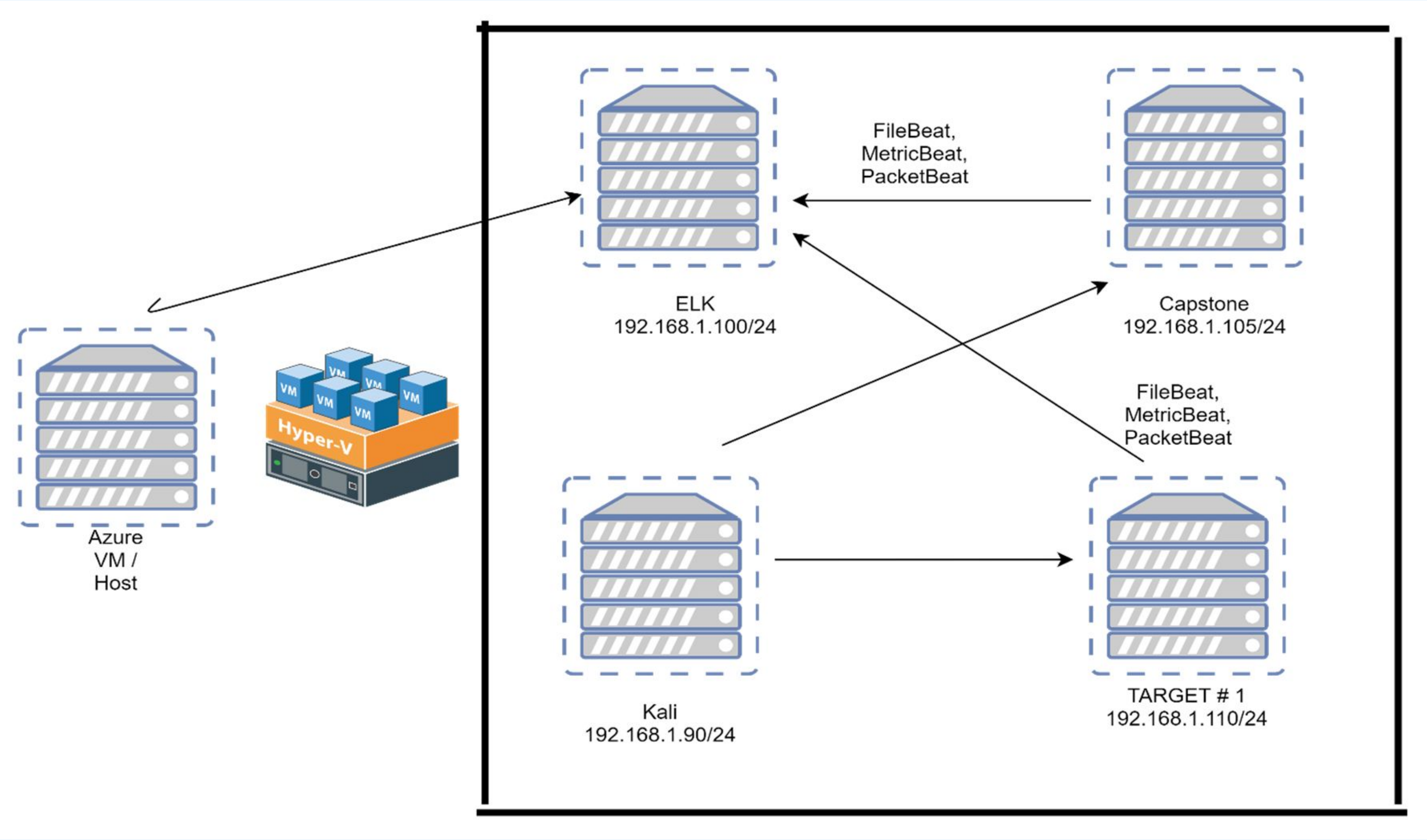
Alerts Implemented

Hardening

Implementing Patches

Network Topology
& Critical Vulnerabilities

Network Topology



Network Topology

Azure VM / Host

ELK
(192.168.1.100/24)

Capstone
(192.168.1.105/24)

Kali
(192.168.1.90/24)

Target # I
(192.168.1.110/24)

Critical Vulnerabilities: Target 1

Vulnerability	Description	Impact
Oversimplified Usernames	First names as usernames can be easily found through reconnaissance or social engineering.	‘Michael’ and ‘Steven’ are predictable usernames. In conjunction with weak passwords, port 22 becomes a vulnerability.
Weak Passwords	Commonly used passwords or simple words without any complexity.	We were able to find Michael’s password using Hydra. We cracked Steven’s password hash using John the Ripper.
Root Accessibility	Authorization to execute commands to escalate privileges.	We were able to escalate to root using a python script from GTFOBins. `python -c 'import os; os.system("/bin/sh")'`
Successful WPScan	Provides a way to discover usernames of accounts on WordPress.	We were able to discover ‘Michael’ and ‘Steven’ as usernames on the vulnerable WordPress site.

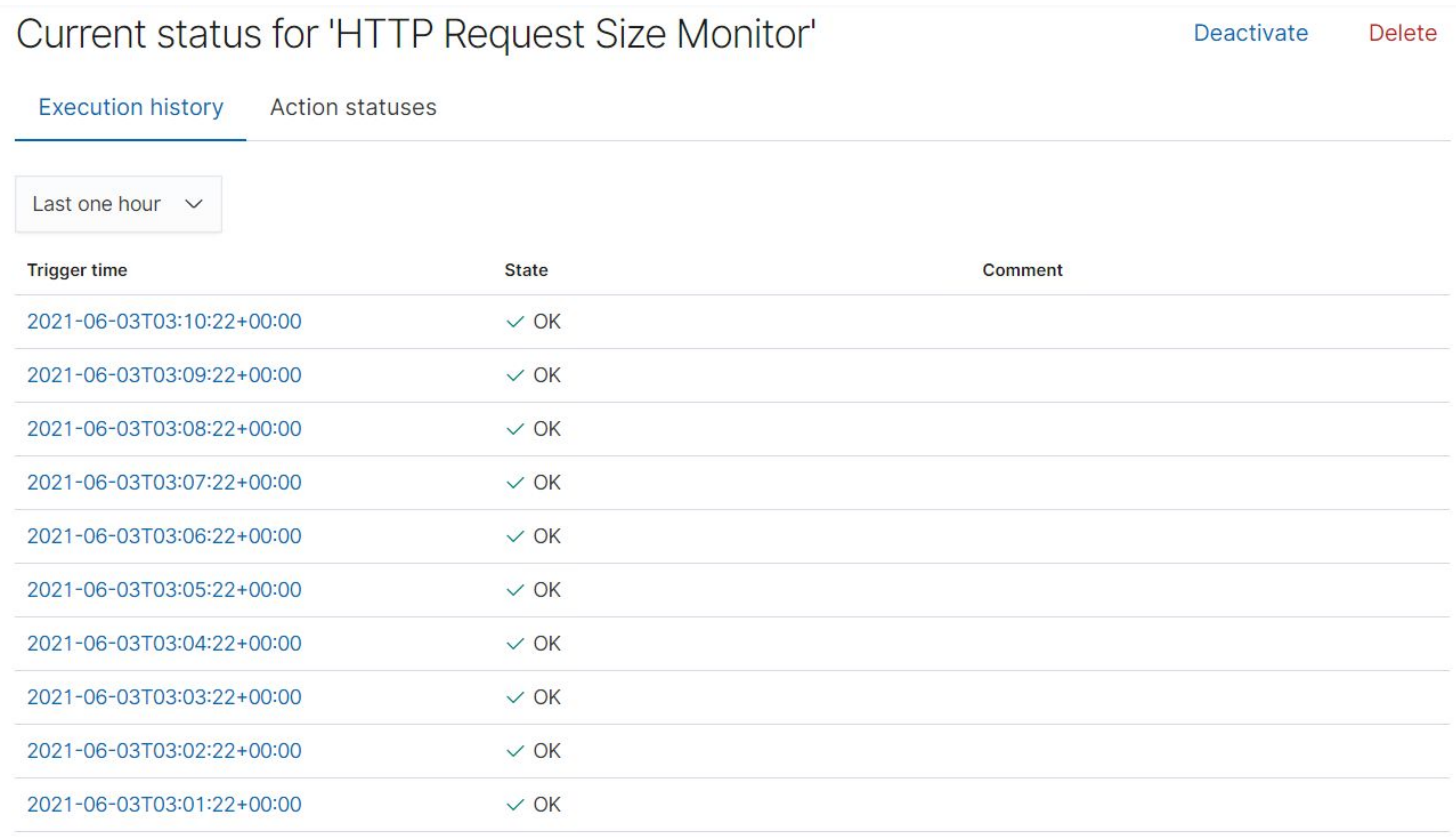
Alerts Implemented

HTTP Request Size Monitor

- **Metric:** Metricbeat
- **Threshold:** When packet size exceeds 3500 bytes in the last 1 minute.
- **Vulnerability Mitigated:** Possible exfiltration and infiltration of data/files either malicious or not given packet size.
- **Reliability:** Low - because there could be files downloaded onto or off the platform. A typical image file would be around 11.8 Kilobytes (11,800 bytes).

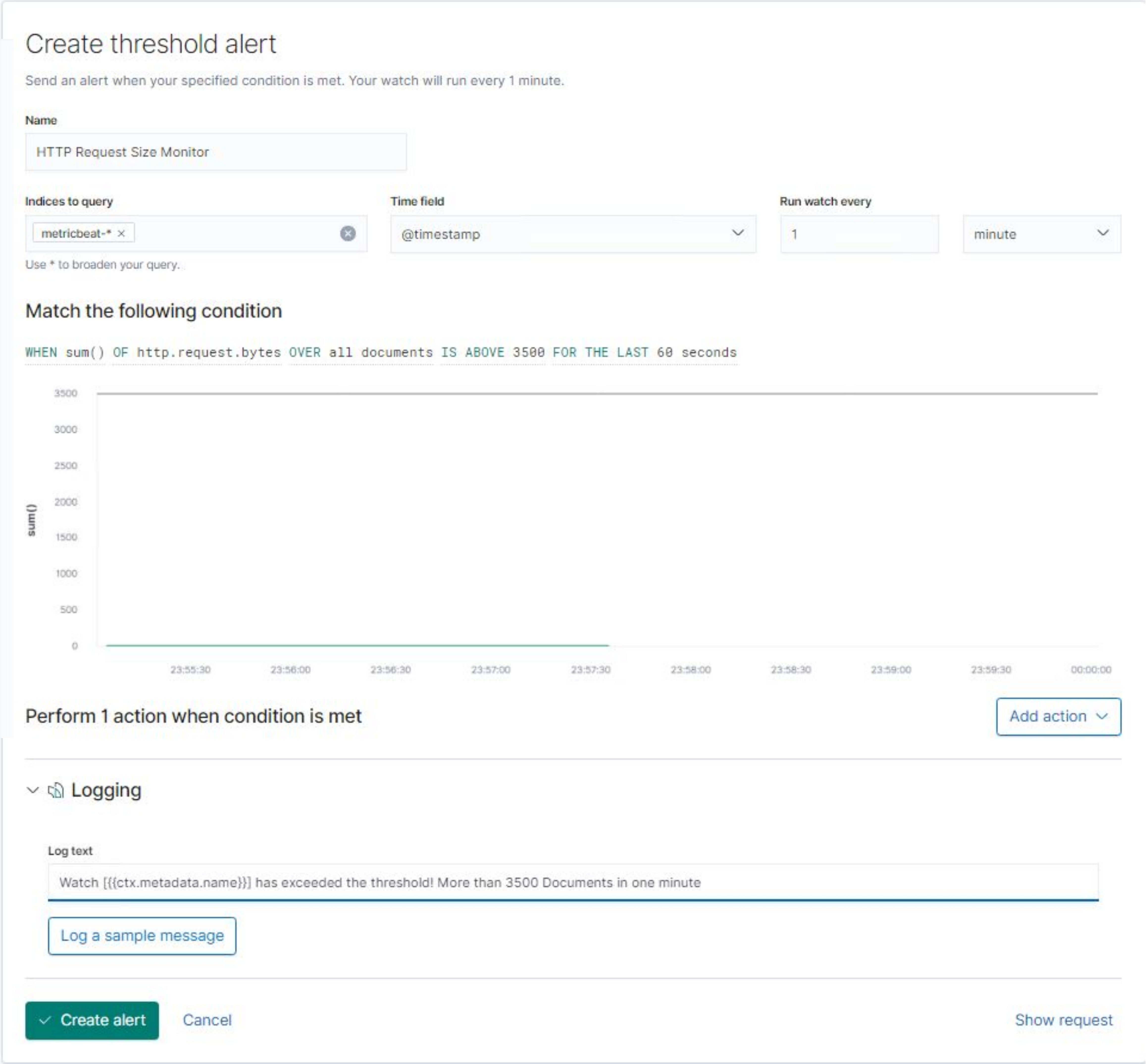
Alert: WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute

HTTP Request Size Monitor



Current status for 'HTTP Request Size Monitor'

Trigger time	State	Comment
2021-06-03T03:10:22+00:00	✓ OK	
2021-06-03T03:09:22+00:00	✓ OK	
2021-06-03T03:08:22+00:00	✓ OK	
2021-06-03T03:07:22+00:00	✓ OK	
2021-06-03T03:06:22+00:00	✓ OK	
2021-06-03T03:05:22+00:00	✓ OK	
2021-06-03T03:04:22+00:00	✓ OK	
2021-06-03T03:03:22+00:00	✓ OK	
2021-06-03T03:02:22+00:00	✓ OK	
2021-06-03T03:01:22+00:00	✓ OK	



Create threshold alert

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

Name: HTTP Request Size Monitor

Indices to query: metricbeat-*

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 60 seconds
```

Perform 1 action when condition is met

Logging

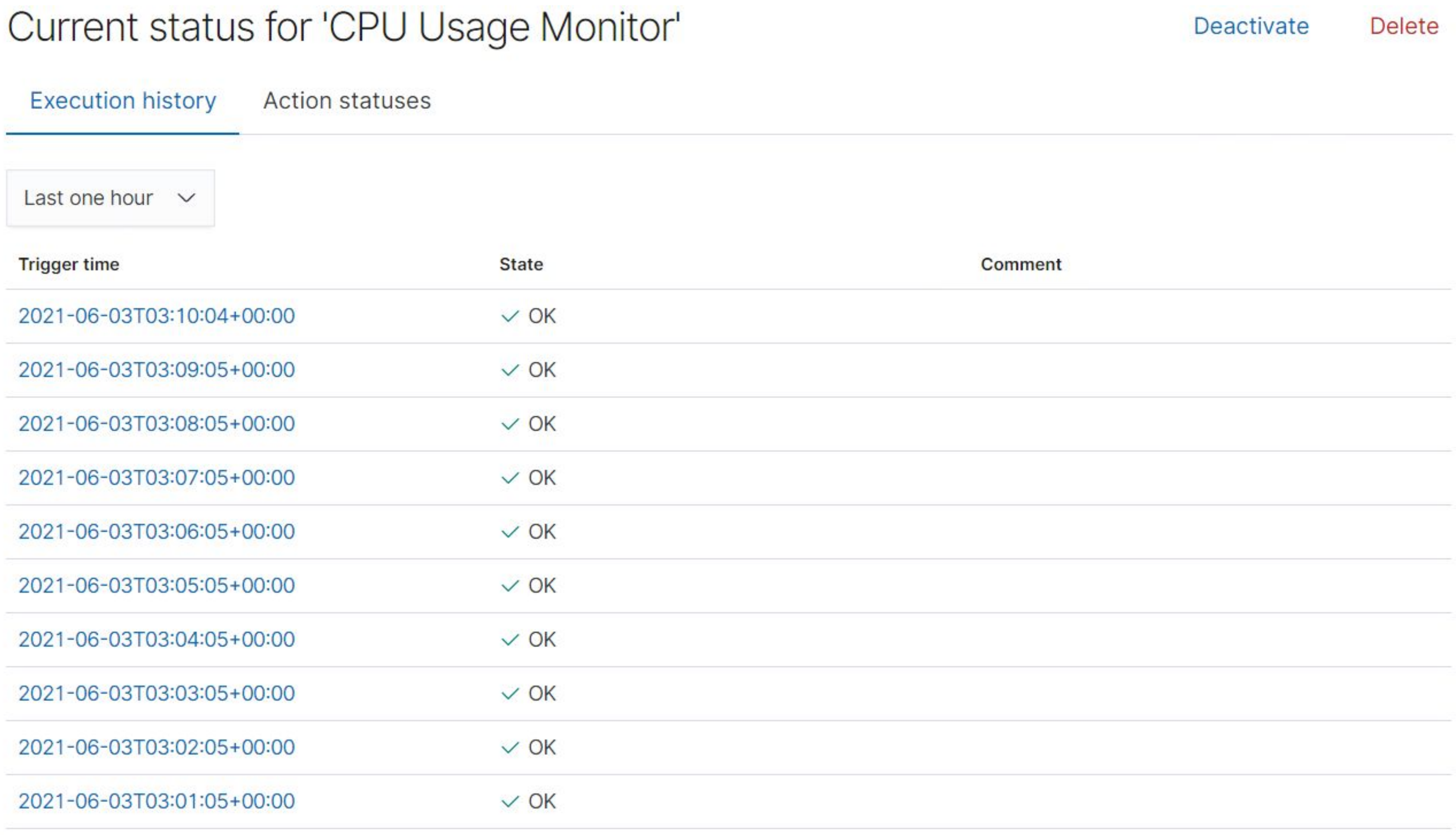
Log text: Watch [{{ctx.metadata.name}}] has exceeded the threshold! More than 3500 Documents in one minute

CPU Usage Monitor

- **Metric:** Metricbeat
- **Threshold:** CPU usage over 50% in the last 5 minutes.
- **Vulnerability Mitigated:** Brute force attacks.
- **Reliability:** Medium Reliability.

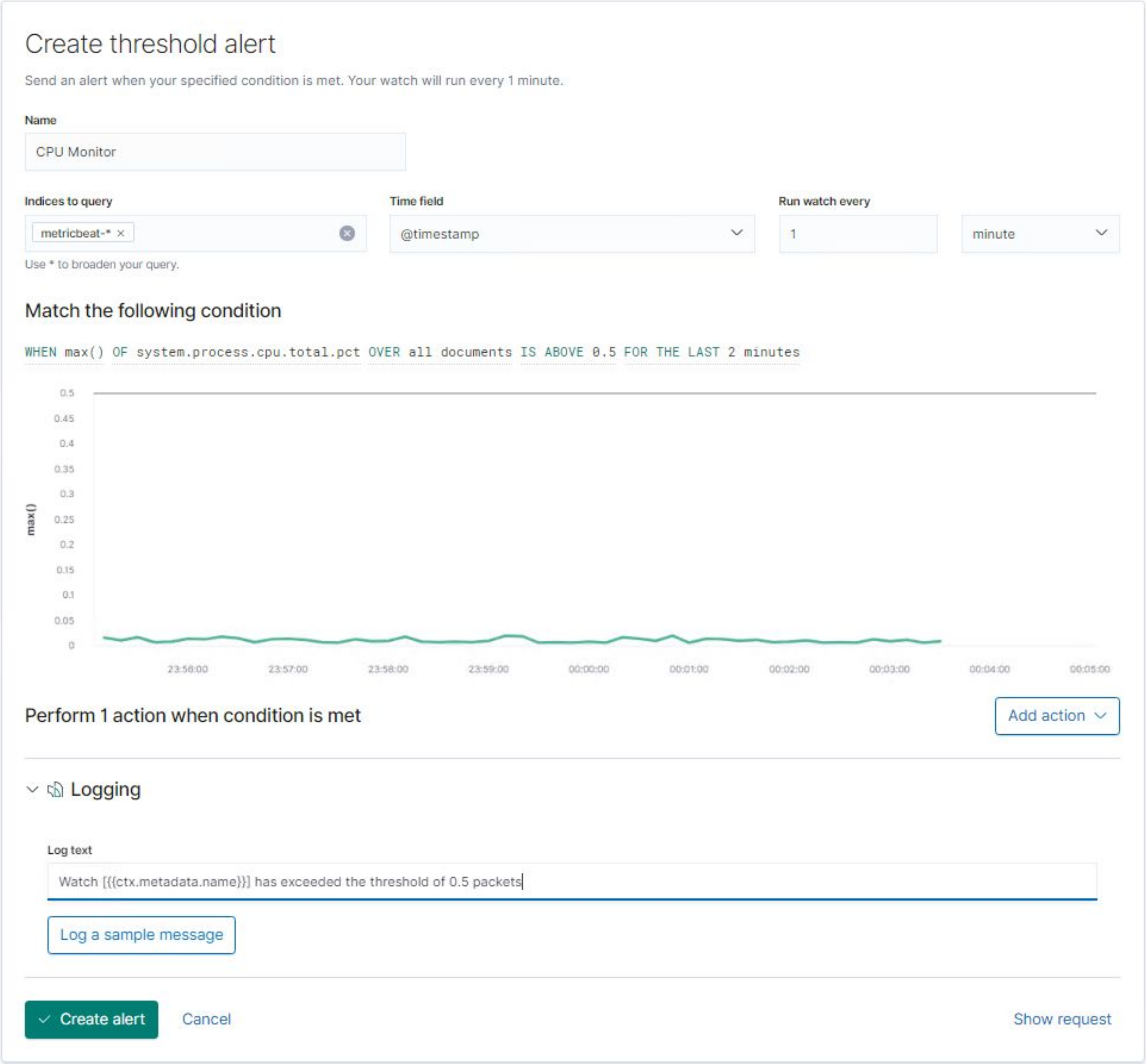
Alert: WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes

CPU Usage Monitor



Current status for 'CPU Usage Monitor'

Trigger time	State	Comment
2021-06-03T03:10:04+00:00	✓ OK	
2021-06-03T03:09:05+00:00	✓ OK	
2021-06-03T03:08:05+00:00	✓ OK	
2021-06-03T03:07:05+00:00	✓ OK	
2021-06-03T03:06:05+00:00	✓ OK	
2021-06-03T03:05:05+00:00	✓ OK	
2021-06-03T03:04:05+00:00	✓ OK	
2021-06-03T03:03:05+00:00	✓ OK	
2021-06-03T03:02:05+00:00	✓ OK	
2021-06-03T03:01:05+00:00	✓ OK	



Create threshold alert

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

Name: CPU 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 2 minutes
```

Perform 1 action when condition is met

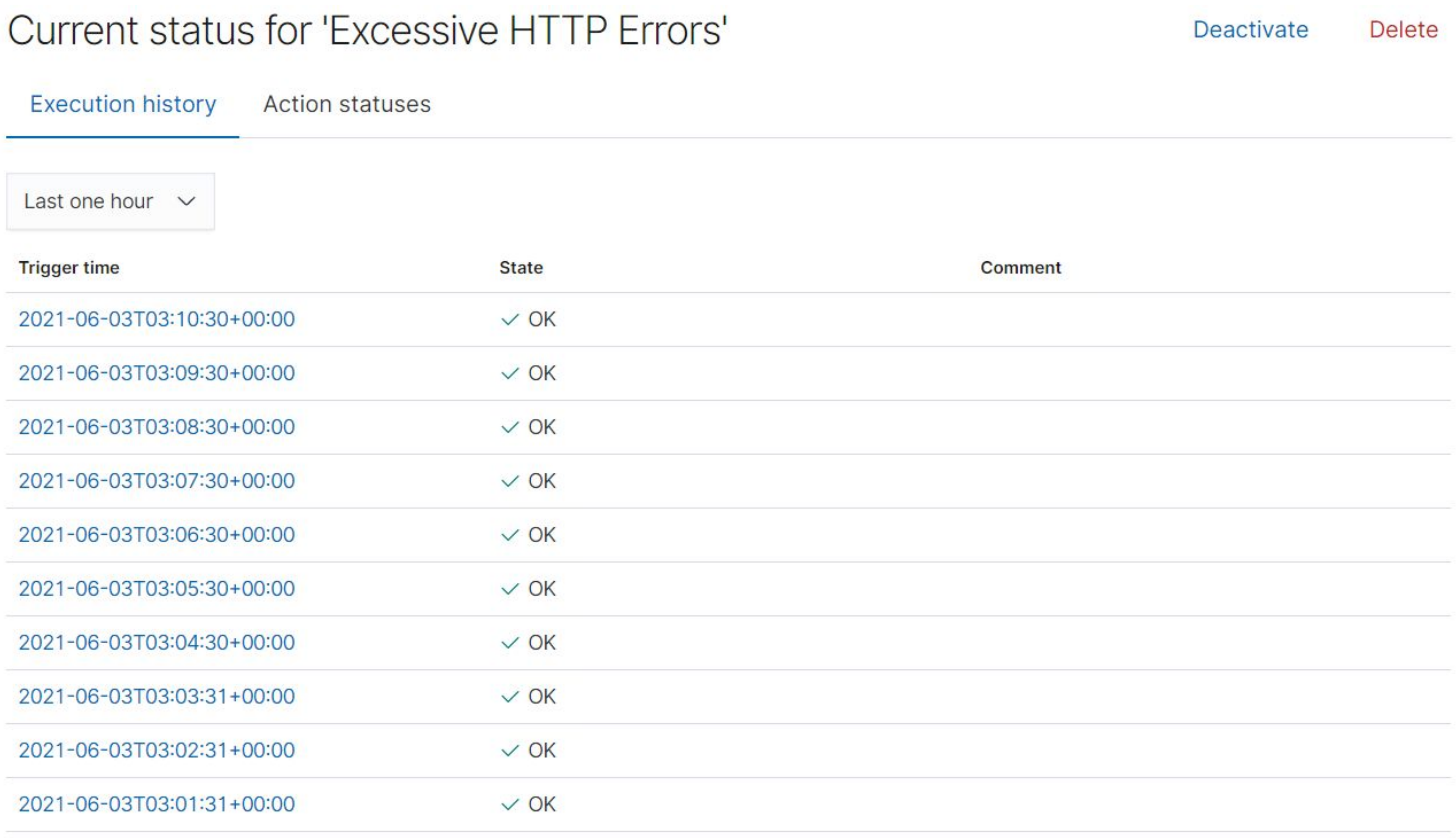
Log text: Watch [{{ctx.metadata.name}}] has exceeded the threshold of 0.5 packets

Excessive HTTP Errors

- **Metric:** Packetbeat
- **Threshold:** HTTP Responses are over 400 in the last 5 minutes (if there are more files being downloaded from your website or onto your website, it means that there are going to be more HTTP requests being displayed.)
- **Vulnerability Mitigated:** Bad user experience, DDOS Attacks, WP Scans
- **Reliability:** Medium Reliability

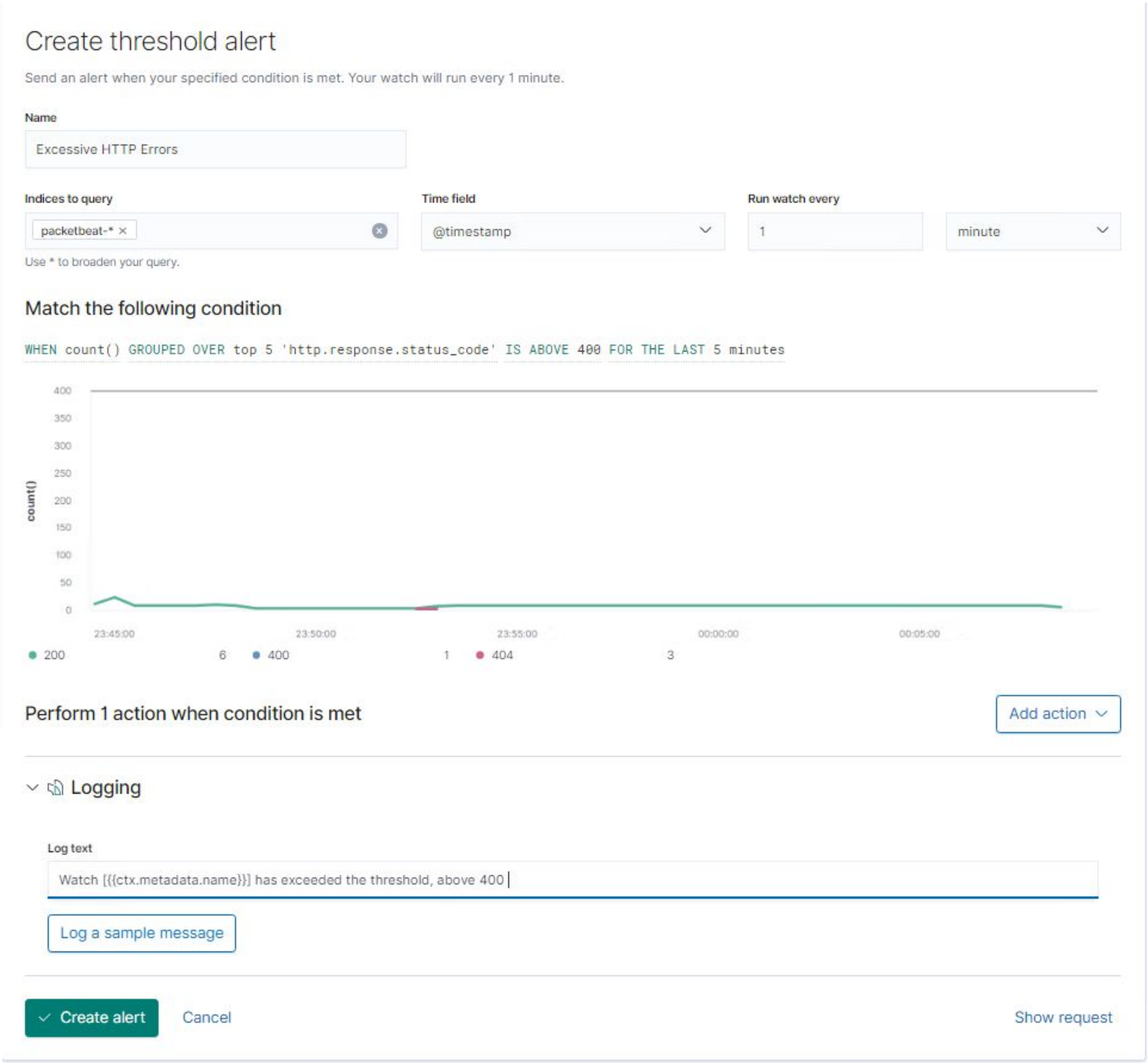
Alert: WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes

Excessive HTTP Errors



Current status for 'Excessive HTTP Errors'

Trigger time	State	Comment
2021-06-03T03:10:30+00:00	✓ OK	
2021-06-03T03:09:30+00:00	✓ OK	
2021-06-03T03:08:30+00:00	✓ OK	
2021-06-03T03:07:30+00:00	✓ OK	
2021-06-03T03:06:30+00:00	✓ OK	
2021-06-03T03:05:30+00:00	✓ OK	
2021-06-03T03:04:30+00:00	✓ OK	
2021-06-03T03:03:31+00:00	✓ OK	
2021-06-03T03:02:31+00:00	✓ OK	
2021-06-03T03:01:31+00:00	✓ OK	



Create threshold alert

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 5 minutes
```

Perform 1 action when condition is met

Logging

Log text: Watch [{{ctx.metadata.name}}] has exceeded the threshold, above 400

Hardening

Hardening Against Oversimplified Usernames on Target 1

- Implement User Account Controls and Group Policy rules that requires more complex usernames.
- Set a corporate policy that generates random usernames.

Hardening Against Weak Passwords on Target 1

- Implement UAC and Group Policy Rules that require more complex passwords.
- Salt stored password hashes.
- Implement the use of a password manager.

Hardening Against CPU Usage on Target 1

Our alert for Target 1 is: when the CPU usage is 50% or greater in the last 5 minutes an alert will trigger.

Our patch will be:

1. Conditionally black list any IP address that has more than 15 failed attempts over a 2 hour period for 24 hours

Hardening Against Excessive HTTP Errors on Target 1

Our alert for Target 1 is: when there are 400+ HTTP Requests in the last 5 minutes an alert will trigger.

Our patch will be:

1. Block Users that are triggering these errors.
2. Blacklist the User(s) for 15 minutes.

Hardening Against WordPress User Enumeration

- Ensure WordPress is up to date with scheduled update checks and installations.
- Change data bases to not use wordpress or wp in the name.
- Use plugin on WordPress to block User Enumeration programs