Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

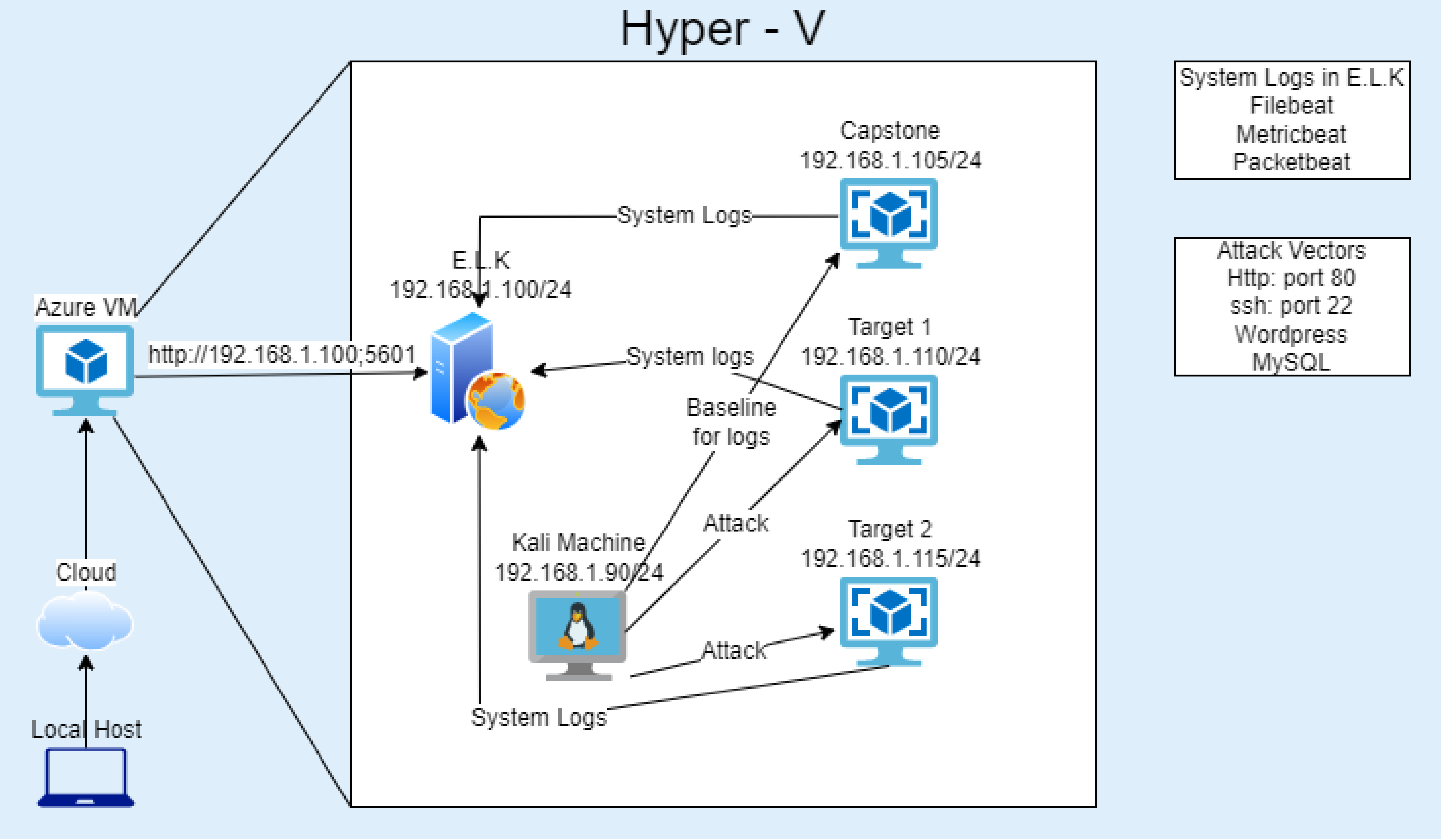
Alerts Implemented

Hardening

Implementing Patches

Network Topology
& Critical Vulnerabilities

Network Topology



Hyper - V

Azure VM

http://192.168.1.100;5601

Cloud

Local Host

E.L.K
192.168.1.100/24

Capstone
192.168.1.105/24

Target 1
192.168.1.110/24

Target 2
192.168.1.115/24

Kali Machine
192.168.1.90/24

System Logs

System logs

Baseline
for logs

Attack

Attack

System Logs

System Logs in E.L.K
Filebeat
Metricbeat
Packetbeat

Attack Vectors
Http: port 80
ssh: port 22
Wordpress
MySQL

Network
Address
Range:192.168.1.1
Netmask:/24
Gateway:

Machines
IPv4: 192.168.1.90
OS Llinux
Hostname: Kali VM

IPv4: 192.168.1.100
OS:Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

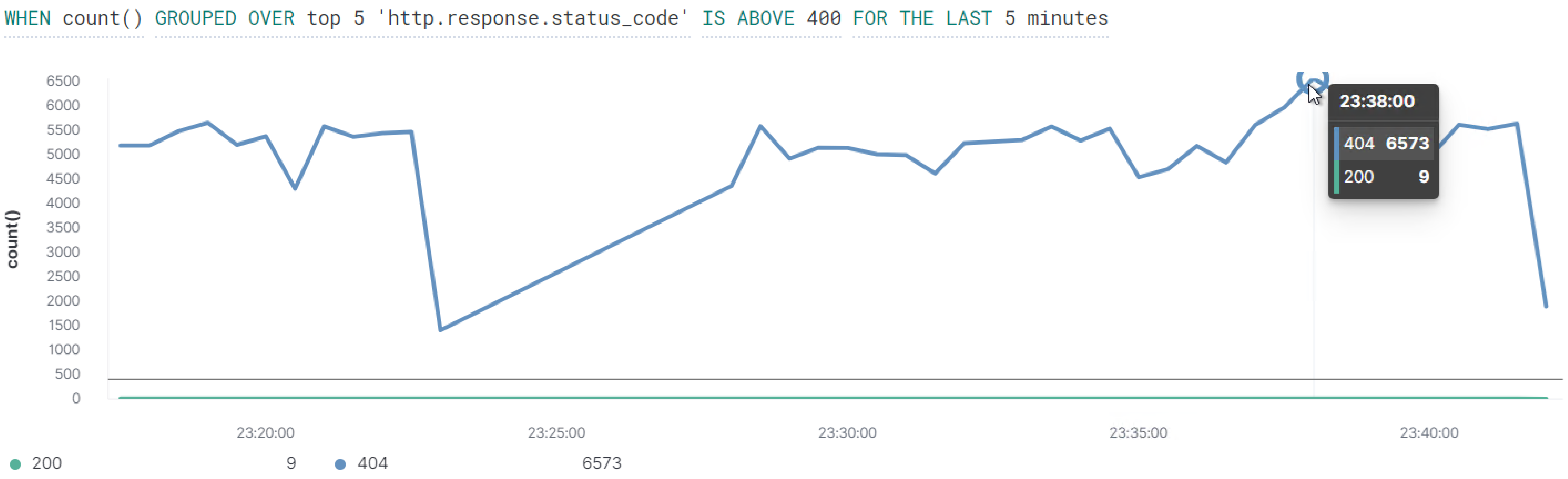
Vulnerability	Description	Impact
OpenSSH 6.7p1	Out of date version of OpenSSH	Unpatched version increases risk of known exploits
Apache httpd 2.4.10	Out of date version of Apache	Unpatched version increases risk of known exploits
User Passwords	Users using basic uncomplicated passwords which have little to no entropy	Easy to crack passwords lead to threat actors gaining access to a network with minimal effort
User Privileges	User’s not in the sudoers file have special permissions to run Python	Using Python when logged in as certain users allows for instant root access without asking for root password

Alerts Implemented

Excessive HTTP Errors

Summarize the following:

- Which **metric** does this alert monitor? HTTP response codes
- What is the **threshold** it fires at? 400 for the last 5 minutes
- Below shows our best attempt at spamming HTTP errors using an automated script:

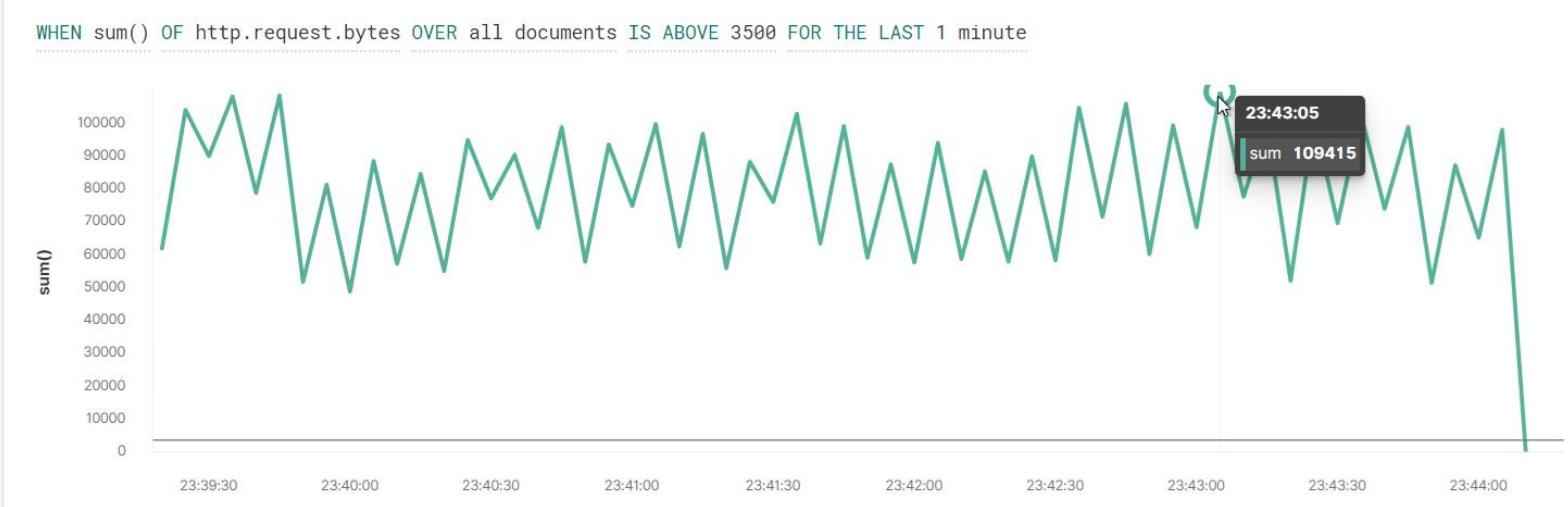


```
WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
```

HTTP Request Size Monitor

Summarize the following:

- Which **metric** does this alert monitor? HTTP Request Bytes
- What is the **threshold** it fires at? 3500 in 1 minute
- Below shows our best attempt at spamming excessive HTTP requests using the same script:

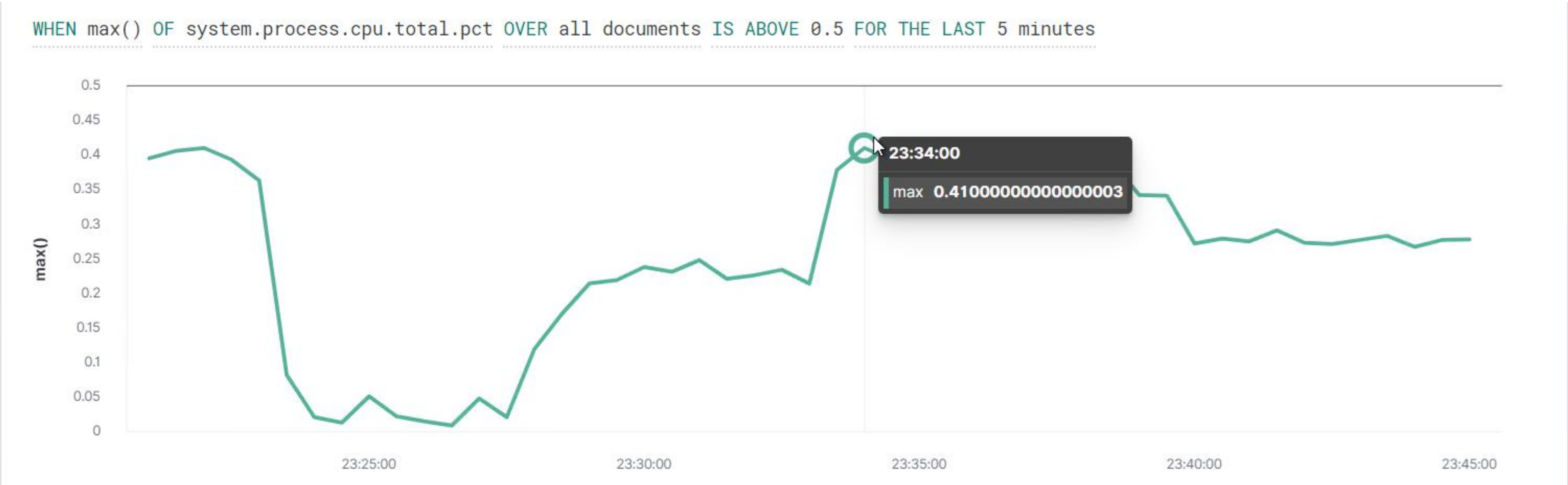


```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```

CPU Usage Monitor

Summarize the following:

- Which **metric** does this alert monitor? CPU Usage
- What is the **threshold** it fires at? Percentage of CPU usage over all documents exceeds 0.5 for last 5 minutes.
- Try as we might we couldn’t get the CPU usage over 0.5, so this would most likely need tuned a little more.



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

Hardening

Hardening Against OpenSSH 6.7p1 on Target 1

- Why the patch works.

  It includes the most recent security patch of OpenSSH)

- How to install it (include commands)
  - sudo apt-get update
  - sudo apt-get install openssh-server
  - sudo service ssh restart

Hardening Against Apache httpd 2.4.10 on Target 1

- Why the patch works.

  It includes the most recent security patch of Apache (Apache 2.4.46)

- How to install it (include commands)
  - sudo add-apt-repository ppa:ondrej/apache2
  - sudo apt update
  - sudo apt install apache2 -y
  - sudo systemctl restart apache2

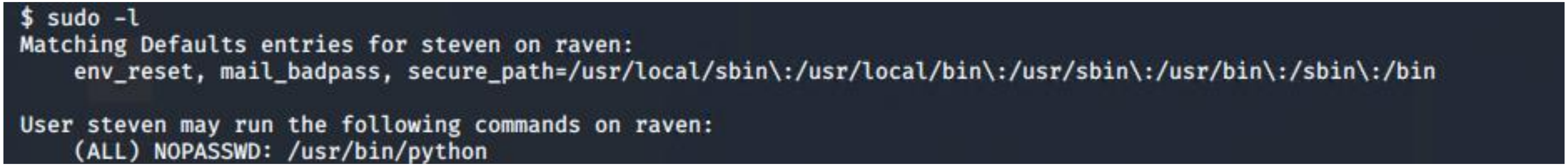
Hardening Against User Passwords on Target 1

To guard against this it would be highly advisable to require users to regularly change their passwords and change them requiring the use of special characters and character length to increase entropy and decrease the chance of them being cracked. Cracking Stevens and Michaals passwords required minimal effort at best.

Implementing some sort of MFA can provide a even more secure means for the user to safely and securely log in.

Hardening Against User Privileges on Target 1

To mitigate against this the administrator needs to ensure only users who have need to run Python has access, in following up with the previous vulnerability, it should also require any account with such privileges to have a much more secure password.



```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
```