Final Engagement

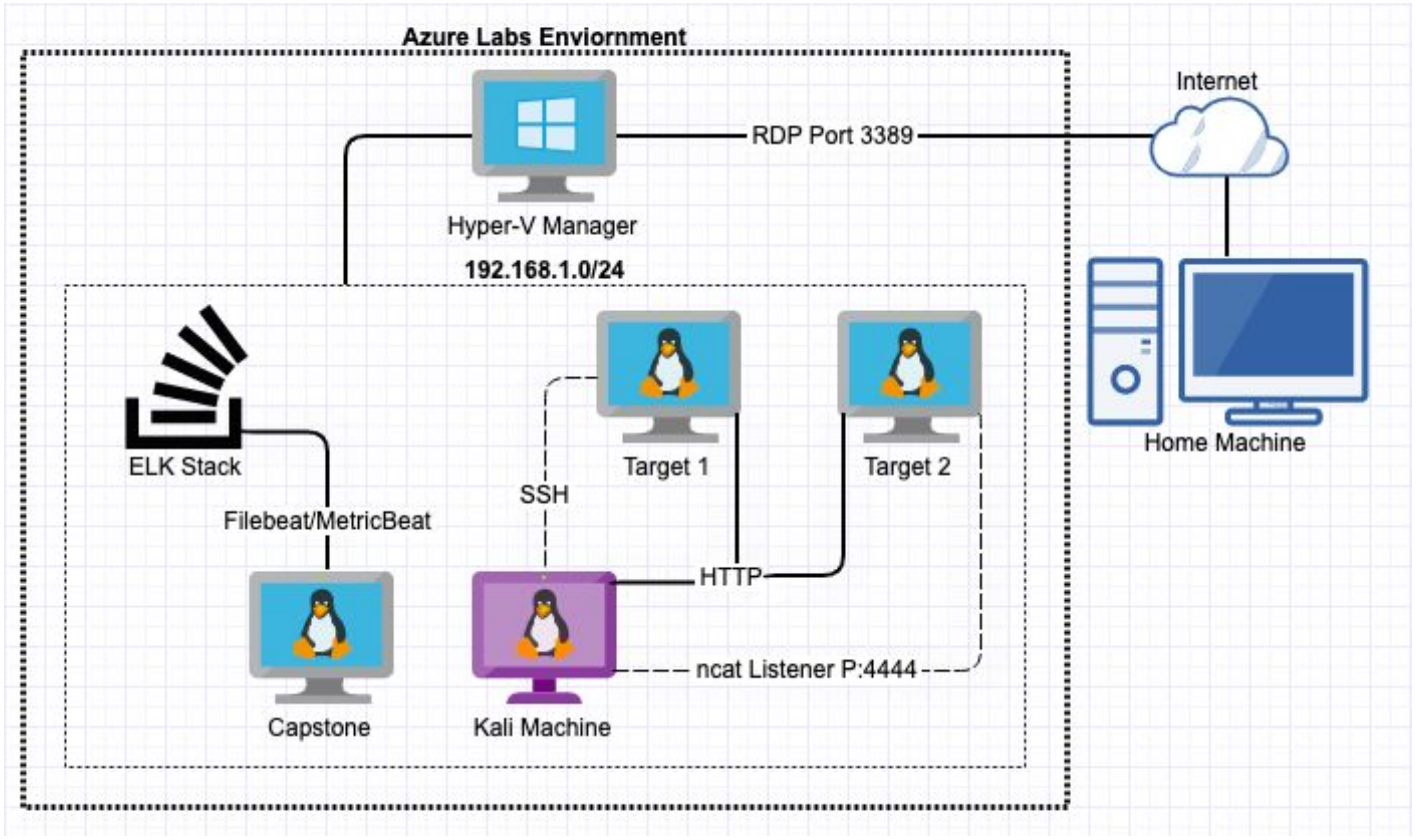
Defensive Report of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0/24
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Debian Kali 5.4.0
Hostname: Kali

IPv4: 192.168.1.110
OS: Debian GNU/Linux 8
Hostname: Target 1

IPv4: 192.168.1.115
OS: Debian GNU/Linux 8
Hostname: Target 2

IPv4: 192.168.1.105
OS: Ubuntu 18.04
Hostname: Capstone

IPv4: 192.168.1.100
OS: Ubuntu 18.04
Hostname: ELK

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Weak Passwords	Was able to find passwords using dictionary brute force against web form	Allowed attacker to gain access to protected web directories
Wordpress User Enumeration	Utilized wpscan enumeration to gather user information for the web server	Allows attacker to gather usernames to gain access to the web server
Unprotected and Unsalted Hash	Used Rainbow table to compare an unprotected hash to a corresponding password	Allowed attacker to gain access to WebDav to alter contents of web server
Privilege Escalation	Used Stevens sudo Python access to escalate from ‘Steven to root’	Allowed privilege escalation to root

Critical Vulnerabilities: Target 2

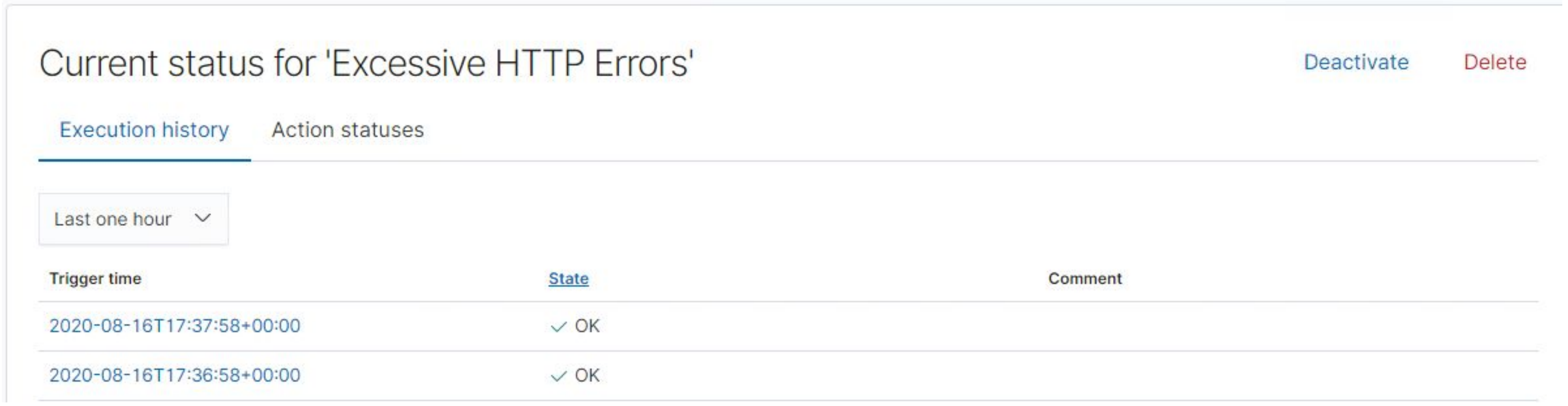
Our assessment uncovered the following critical vulnerabilities in **Target 2**.

Vulnerability	Description	Impact
Directory Listing using wpscan and gobuster	Web directory listings available online	Allows unauthorized access to directories & files
PHPMailer RCE: CVE 2016-10033	Allows extra parameters in mailSend function	Allows execution of code including a user shell
Privilege Escalation	Gained through default username/password for root	Allows root access

Alerts Implemented

Excessive HTTP Errors

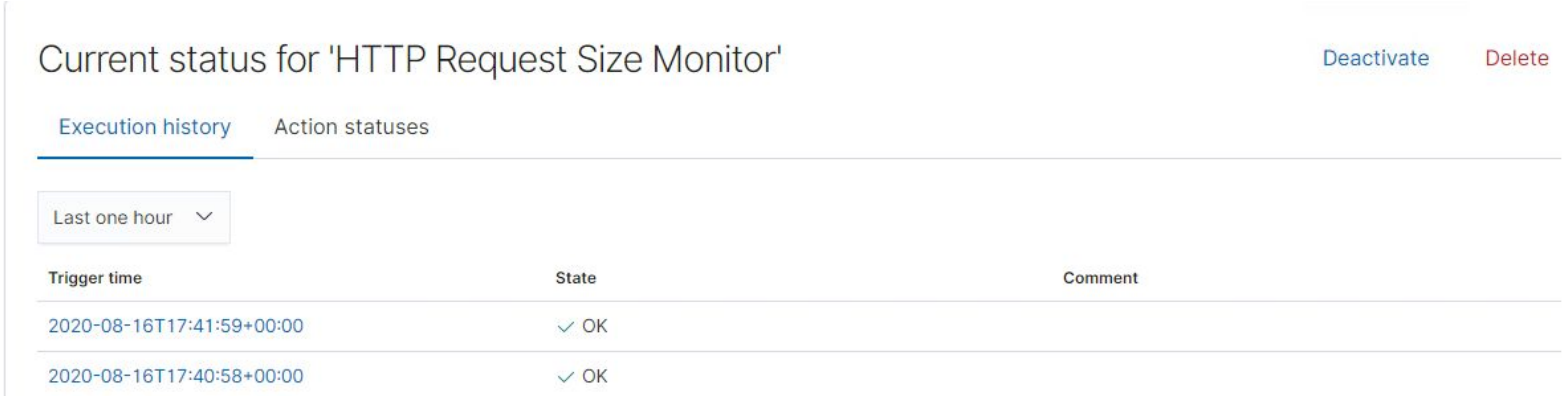
- Which **metric** does this alert monitor?
  - Count grouped over top 5 ‘http.response.status_code’
- What is the **threshold** it fires at?
  - Above 400 in 5-minute period
- Screenshot of the alert in action:



Trigger time	State	Comment
2020-08-16T17:37:58+00:00	✓ OK	
2020-08-16T17:36:58+00:00	✓ OK	

HTTP Request Size Monitor

- Which **metric** does this alert monitor?
  - Sum of http.request.bytes over all documents
- What is the **threshold** it fires at?
  - Above 3500 in 1-minute period
- Screenshot of the alert in action:

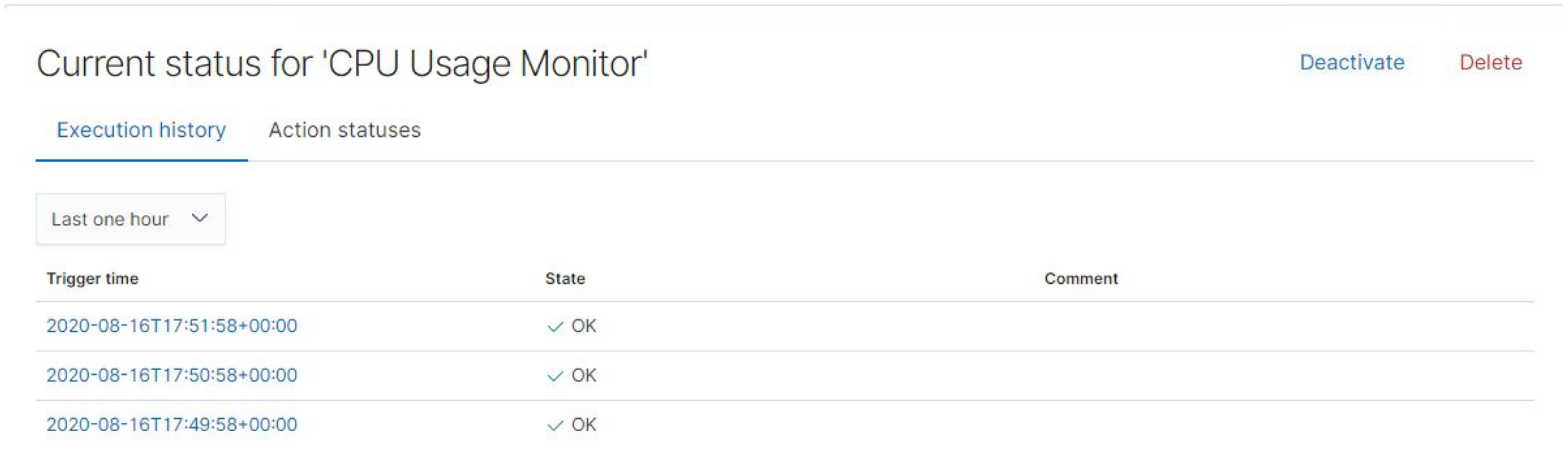


Current status for 'HTTP Request Size Monitor'

Trigger time	State	Comment
2020-08-16T17:41:59+00:00	✓ OK	
2020-08-16T17:40:58+00:00	✓ OK	

CPU Usage Monitor

- Which **metric** does this alert monitor?
  - Max of system.process.cpu.total.pct over all documents
- What is the **threshold** it fires at?
  - Above 0.5 in 5-minute period
- Screenshot of the alert in action:



Current status for 'CPU Usage Monitor'

Trigger time	State	Comment
2020-08-16T17:51:58+00:00	✓ OK	
2020-08-16T17:50:58+00:00	✓ OK	
2020-08-16T17:49:58+00:00	✓ OK	

Hardening

Hardening Against Wordpress Enumeration on Target 1

How to patch Target 1 against Wordpress Enumeration:

- Disable the WordPress REST API and XML-RPC if it’s not needed.
- You can also configure the web server to block requests to /?author=<number>.
- Prohibit exposure of /wp-admin and /wp-login.php.

Why it works:

- WPScan uses REST API to enumerate users.
- XML-RPC uses HTTP as it’s transport mechanism for data.
- WordPress permalinks can be set to include an author (user) and not exposing WordPress logins adds to brute force attack defense.

How to install it:

- Configure WordPress settings and server to achieve these objectives.

Hardening Against Unprotected and Unsalted Hash on Target 1

How to patch Target 1 against Unprotected and Unsalted Hashes:

- Thoroughly secure all passwords in the system with protected and salted hashes via password management tools.

Why the patch works:

- Salting passwords hides the real hash value by adding an additional bit of data and altering it. This slight alteration makes any brute force attack more challenging to crack.

How to install it:

- Implement a good algorithm to implement a strong number generator on your hashes.
- OWASP suggests SecureRandom as a cryptographically-strong random data.

Hardening Against Privilege Escalation on Target 1

How to patch Target 1 against Privilege Escalation:

- Administrator permissions should be limited to essential personal with privilege given to individuals for specific assignments.
    - Be aware of hidden administrators.
        - The local administrator account on workstations and servers.
        - Service accounts with weak or unchanging passwords.

Why this works:

- Limiting access to permissions allows for accountability, thus should there be any compromise, the attacker will not be able to escalate.

How to install it:

- In order to prevent user error a good tool to rely on is auditd to aid in finding any compromised accounts.
- Ensure proper configuration of the sudoers files.

Hardening Against Directory Exploration on Target 2

How to patch Target 2 against Directory Exploration:

- A number of tools have been designed to act based on the log activity. For example one that is used is Fail2Ban, this tool can be configured to temporarily ban a remote IP address with firewall rules.

Why the patch works:

- This configuration temporarily bans IP addresses with firewall rules if it generates too many 404s within a time period.

How to install it:

- apt-get update && apt-get upgrade -y (ensure system is up to date)
- apt-get install fail2ban (Install Fail2ban)

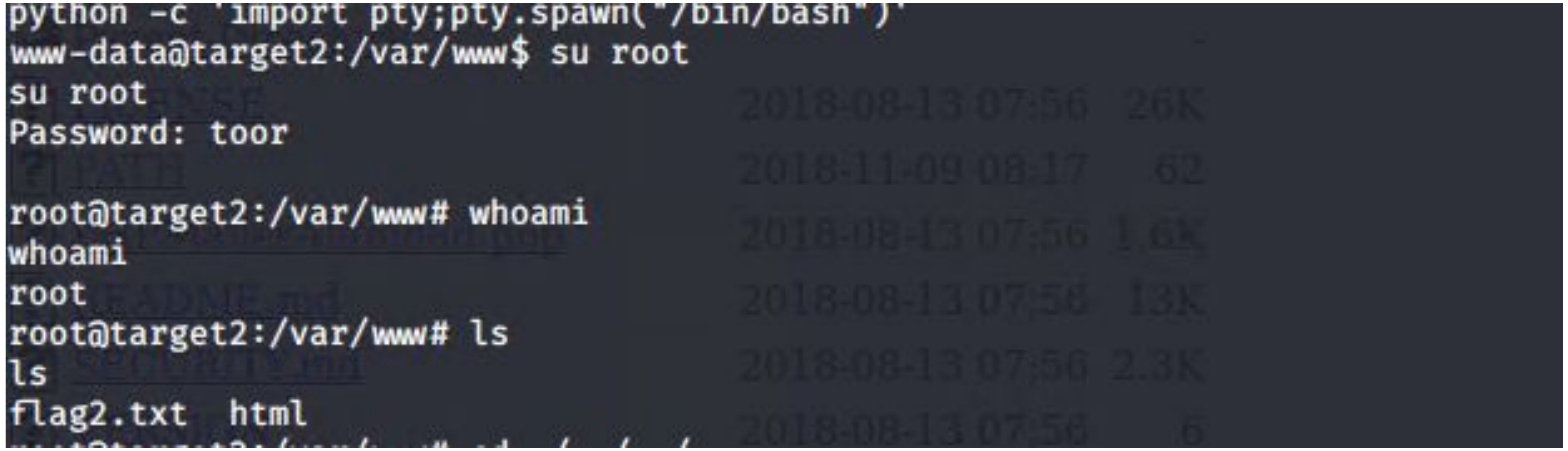
Hardening Against [PHPMailer RCE: CVE 2016-10033] on Target 2

- An effective web application firewall (WAF) can dynamically learn your applications’ “normal” behavior, correlate that with crowd-sourced threat intelligence, and then update in real time to deliver 0day protection against vulnerabilities.

Hardening Against [Privilege Escalation] on Target 2

Summarize the following:

- How did you exploit the vulnerability?
    - Used su root in metaspolit shell to gain information needed to perform escalation
    - Used sudo Python access to escalate to root
        - sudo python -c 'import pty; pty.spawn("bin/bash")'
- What did the exploit achieve?
    - Achieved root access on the machine



```
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@target2:/var/www$ su root
su root
Password: toor

root@target2:/var/www# whoami
whoami
root
root@target2:/var/www# ls
ls
flag2.txt  html
```

Implementing Patches

Implementing Patches with Ansible

Playbook Overview

- One could utilize ansible and a cron job to automate system wide updates as well as keep necessary tools up to date. Ansible can also be used to verify system health (ie. ensuring web servers are up and running)

```
---
- name: Update apt-get repo and cache
  hosts: webservers
  apt: update_cache=yes force_apt_get=yes cache_valid_time=3600
  - name: Check if reboot is required
  - register: reboot_required_file
  - stat: path=/var/run/reboot-required get_md5=no
```

Patch for Unsalted Passwords

- The best and only patch for an unsalted password is to add salt
  - Salt should be added using Cryptographically Secure Pseudo-Random Number Generator (CSPRNG)
    - These provide higher levels of randomness and unpredictability
    - Salt also shouldn’t be reused so these tools help make sure that every unique user and password gets its own unique salt.
- Good rules of thumb for salt lengths should be the same as the output of the hash function
  - For example, SHA256 is 32 bytes, so the salt should be at least 32 bytes.
- Slow hash functions
  - This helps prevent brute force attacks or dictionary attacks by making the hash function slower so attackers can’t overpower can’t run through possible hashes as quickly.
- Require single use tokens, such as emails directly to user to reset passwords with links that expire and is tied to the user account
  - Make sure the new password has a new salt attached to it as well

Patches for Privilege Escalation

- Robust password requirements for all users
  - Require longer lengths, special characters and numbers, words not found in dictionary or related to user name, as well as requirements on how often they should be updated.
    - Another layer is to prevent passwords from being reused.
    - Enterprise password management tools are also available if budgets permit
      - These tools can scan your hostnames and IP addresses to identify weak credentials

- User and Entity Behavior Analytics (UEBA) tools
  - These tools provide assistance in discovering compromised identities
  - Establishes baseline behavior of users by analyzing what and how frequently a user accesses such as resources, data files and services as well as their locations.
  - Tools allow alerts to be set if users deviate from baseline activity.

Patches for Privilege Escalation

- Implement multi-step authentication for data input fields and databases
- Encrypt all data in transport and at rest
- Require escalation for users to receive write access
- Train all users and establish culture of security
    - Users are more likely to be compliant if they understand vulnerabilities and reasons for things like password requirements and multi-step authentication