Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

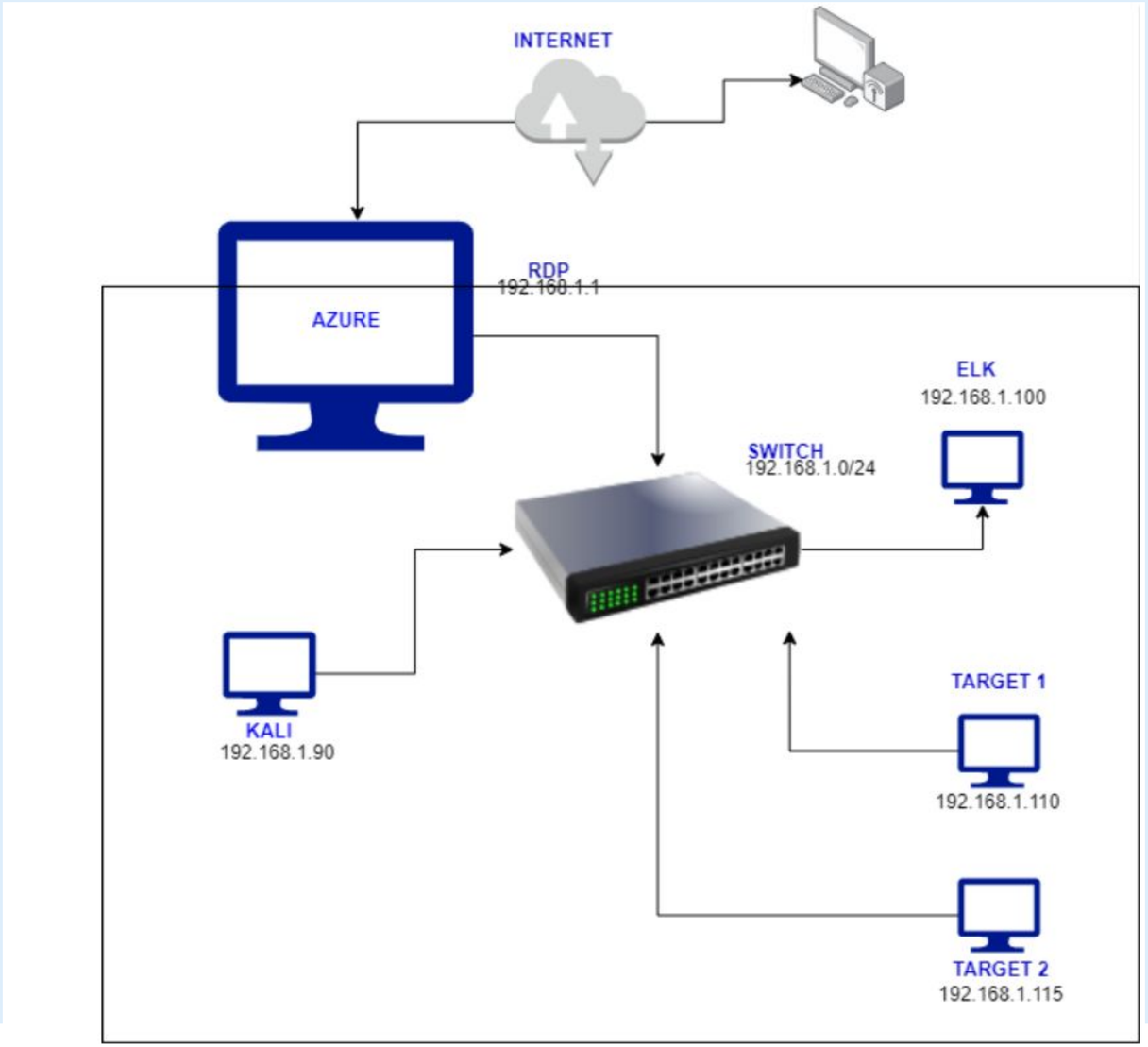
Alerts Implemented

Hardening

Implementing Patches

Network Topology & Critical Vulnerabilities

Network Topology



Network
Address Range: 192.168.1.0 - 192.168.1.255
Netmask:255.255.255.0
Gateway: 192.168.1.0

Machines
IPv4:192.168.1.100
OS: WIndows
Hostname:Target2

IPv4: 192.168.1.110
OS: linux
Hostname:Target1

IPv4: 192.168.1.115
OS: linux
Hostname:Target2

IPv4: 192.168.1.90
OS: Linux
Hostname:Target1

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
1. Port 22 open (SSH)	Port allows secure shell login and secure file transfer.	Allows users with credentials to access server through an external IP
2. Port 80 open (HTTP, httpd apache)	Port 80 is open to allow web traffic. The apache server runs on this port.	Allows users with credentials to upload/download files through an external IP, allows hackers to exploit apache server.
3. Weak passwords	There is no clear password policy that enforces password complexity	Easy for hackers to crack passwords and access sensitive accounts/files.

Alerts Implemented

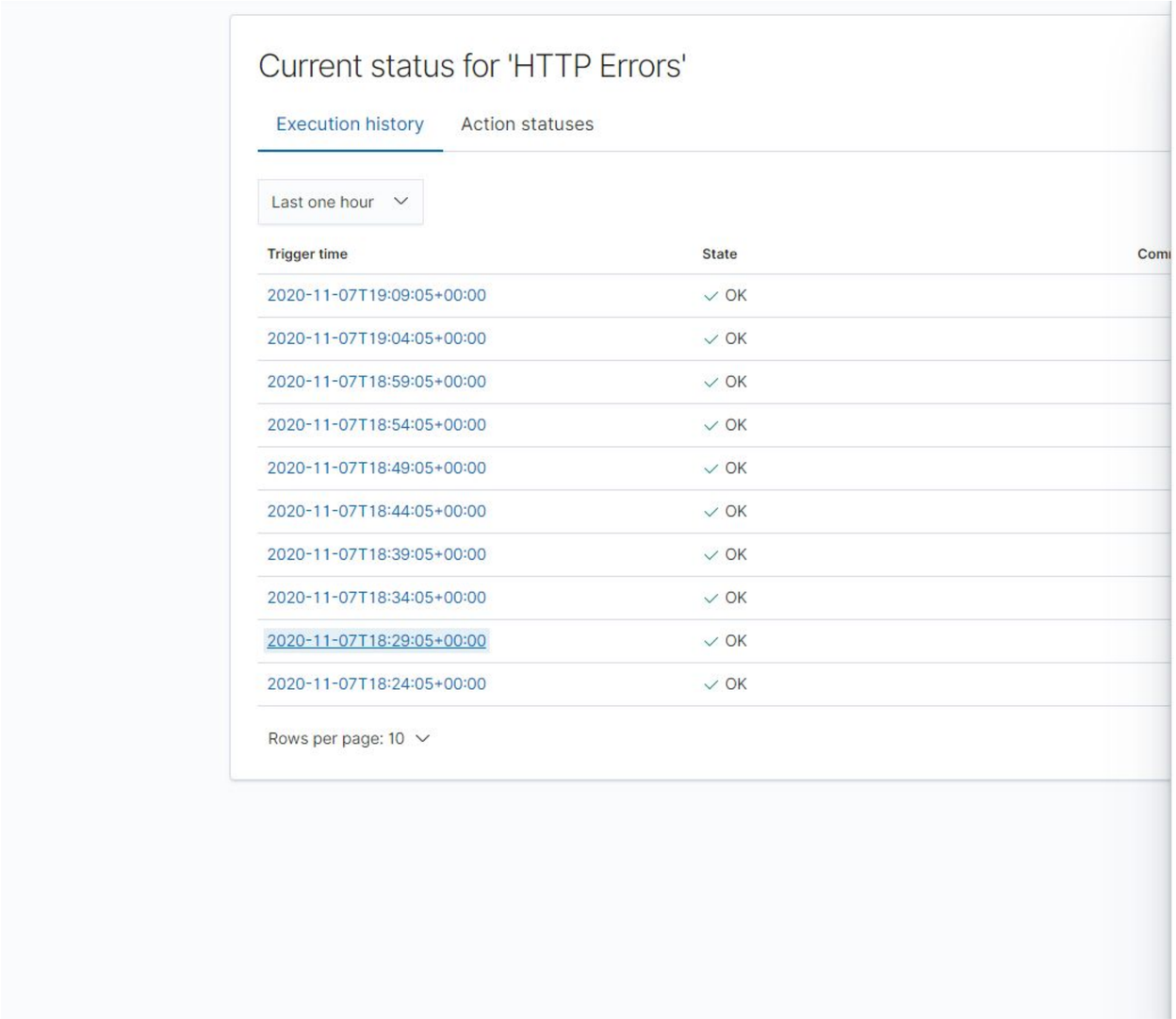
Alert 1: Excessive HTTP Errors

- Which **metric** does this alert monitor?

  The count of HTTP requests

- What is the **threshold** it fires at?

  Above 400 requests in the last 5 minutes



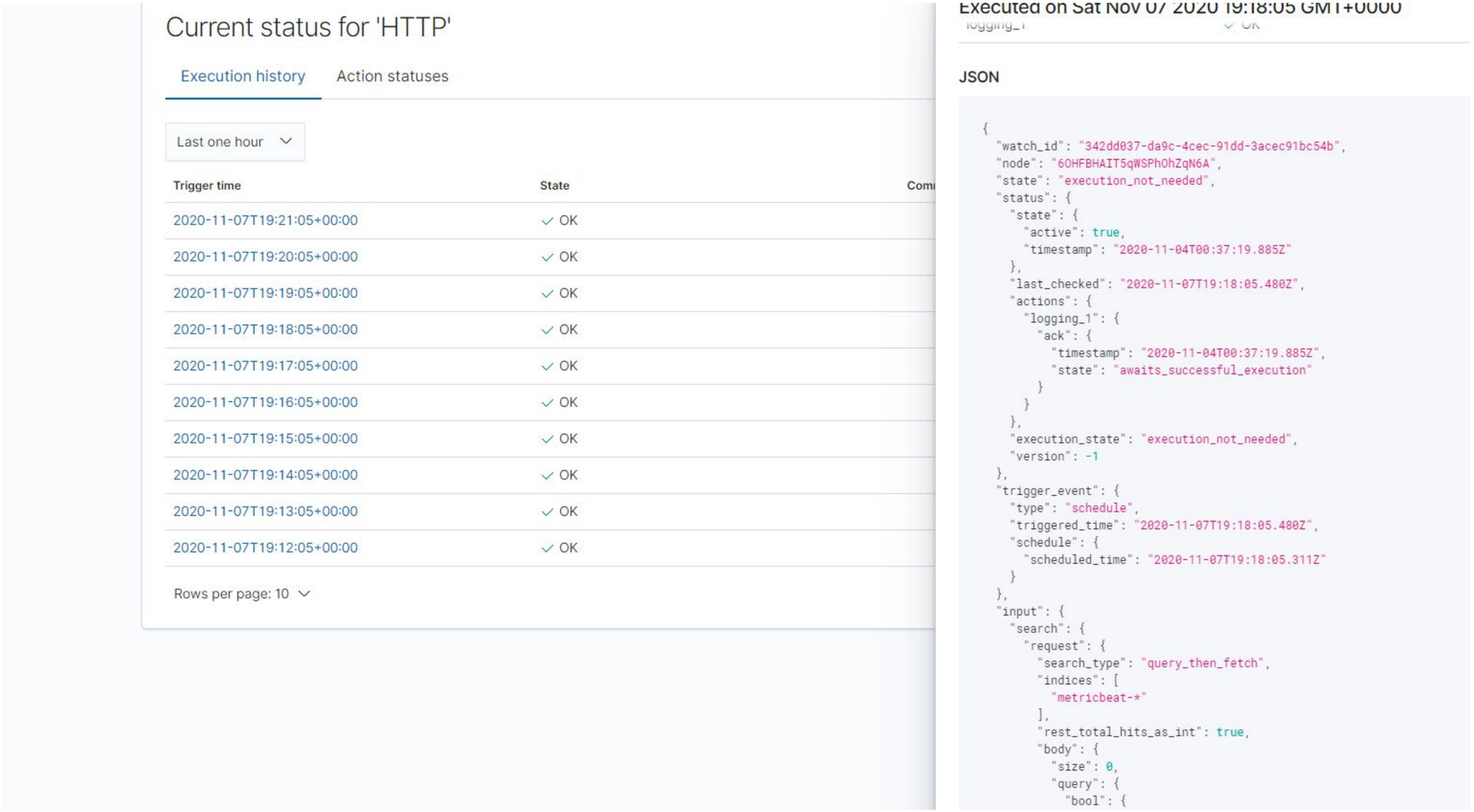
Alert 2: HTTP Request Size Monitor

- Which **metric** does this alert monitor?

  The size of HTTP requests

- What is the **threshold** it fires at?

  Above 3500 for the last 1 minute



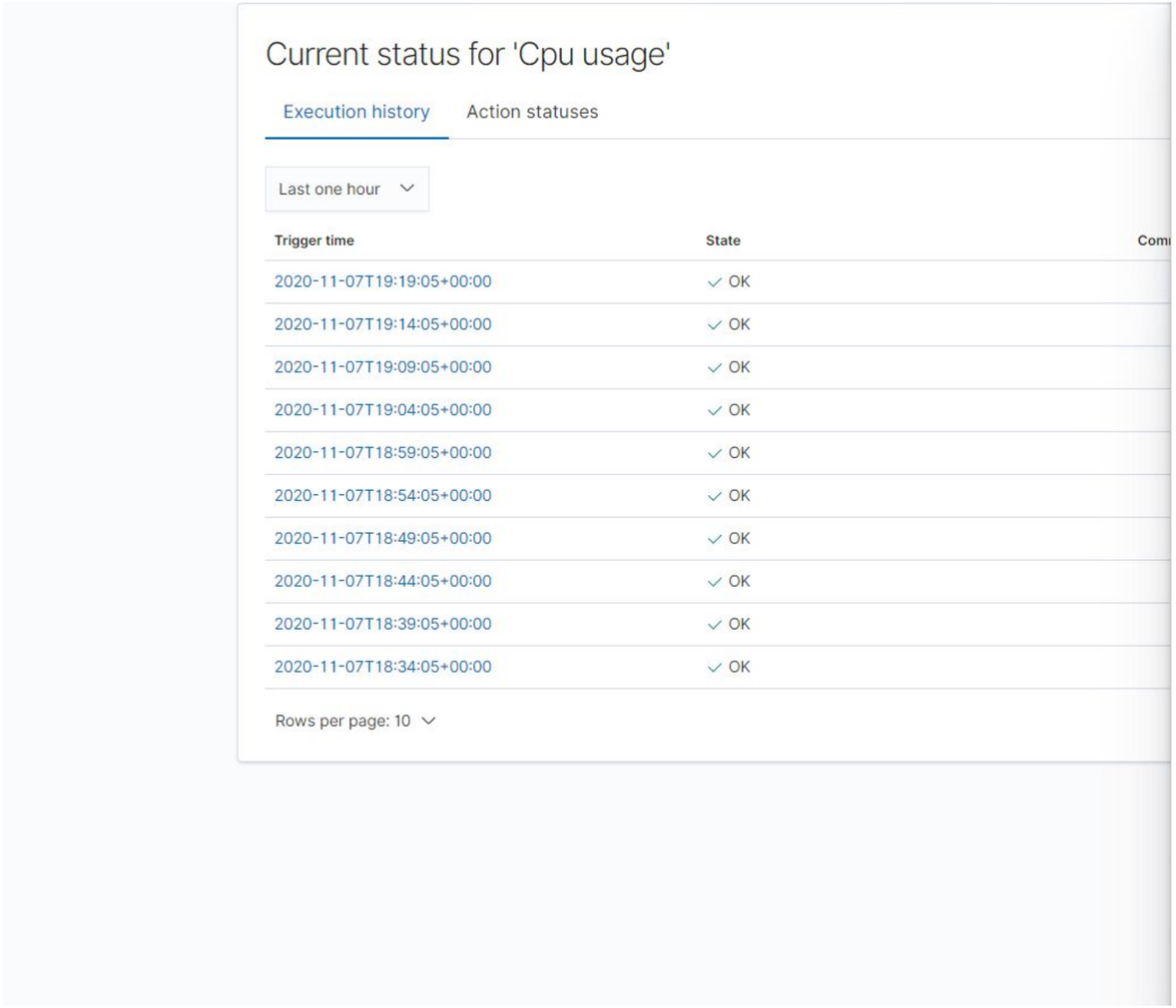
Alert 3: CPU Usage Monitor

- Which **metric** does this alert monitor?

  CPU usage for all documents

- What is the **threshold** it fires at?

  When CPU is above 0.5 for the last 5 minutes



Hardening

Hardening Against Open SSH Port on Target 1

Why the patch works:

- The below patches prevent users from gaining unauthorized access to the application
  - Create an IP filter for the SSH port on your firewall
  - Disable root login and empty password field
  - private and public key requirement with passphrase
  - select custom SSH port

How to install it:

- Command for IP filter on on SSH port
  - `iptables -A INPUT -p tcp -s <IP address> -m tcp --dport 899 -j ACCEPT`
- Command to disable root login and empty password field and select custom SSH port
  - `nano /etc/ssh/sshd_config`
    - `PermitRootLogin  no ; PermitEmptyPasswords  no ; Port <Select port i.e. 899>`

Hardening Against Open HTTP Port on Target 1

Why the patch works:

- The below patches are used to prevent users from accessing port 80
  - Ensure that the apache server is up to date. if not, then install the latest version
  - Disable HTTP trace request

How to install it:

- disable HTTP trace
  - `nano /WebServer/Conf/httpd.conf ; TraceEnable off`
- White List IP’s on the filewall
  - `iptables -A INPUT -p tcp -s <IP address> -m tcp --dport 80 -j ACCEPT`

Hardening Against Weak Passwords on Target 1

1. **Configure password complexity in DEB based systems:**
   a. **Load the CrackLib Recipe module of PAM, pam_cracklib to test and enforce password strength requirements**
2. **Enable Multi-Factor Authentication (i.e. RSA / Physical Device, Captcha)**

```
sudo apt-get install -y libpam-cracklib
```

Edit file: `/etc/pam.d/common-password`

```
$ sudo cp /etc/pam.d/common-password /root/
$ sudo vi /etc/pam.d/common-password
```

Then Update:

password requisite pam_cracklib.so retry=3 minlen=16 difok=3 ucredit=-1 lcredit=-2 dcredit=-2 ocredit=-2

retry=3 : Prompt user at most 3 times before returning with error. The default is 1.

minlen=16 : The minimum acceptable size for the new password.

difok=3 : The number of character changes in the new password that differentiate it from the old password.

ucredit=-1 : The new password must contain at least 1 uppercase characters.

lcredit=-2 : The new password must contain at least 2 lowercase characters.

dcredit=-2 : The new password must contain at least 2 digits.

ocredit=-2 : The new password must contain at least 2 symbols

Hardening Against Weak Passwords on Target 1 (Cont'd)

Set Up Multi-Factor Authentication for users

1. Install the Google PAM Module

2. Configuring 2FA for a User

3. Activating 2FA in Ubuntu

ssh lillian@your_server_ip

sudo apt-get update

sudo apt-get install libpam-google-authenticator

google-authenticator

**Complete configurations (answer Yes)*

sudo nano /etc/pam.d/common-auth

 # and here are more per-package modules (the "Additional" block)

session required pam_unix.so

session optional pam_systemd.so

 # end of pam-auth-update config

auth required pam_google_authenticator.so nullok

Implementing Patches

Implementing Patches with Ansible

We have implemented a UFW and PAM solution independent of application security solutions to add an extra level of security:

1. **UFW solution hardening against port 80 and 22:**

- Since our servers are exposed to HTTP and SSH we can use UFW to simplify lower-level packet filtering technologies, such as iptables. See examples below of how you may allow connections.

```
sudo nano /etc/default/ufw -> IPV6= yes                                    (1)
sudo ufw default deny incoming                                             (2)
sudo ufw default allow outgoing                                            (3)
sudo ufw allow from <IP address/subnet> to any port 22                     (4)
sudo ufw allow in on <public network interface name> to any port 80        (5)
sudo ufw enable                                                            (6)
```

2. **PAM solution hardening against weak passwords:**

- Using the pam_stack.so module, our playbook calls for three password methods and one session method.

```
password required /lib/security/$ISA/pam_cracklib.so retry=3                          (1)
password sufficient /lib/security/$ISA/pam_unix.so nullok use_authtok md5 +           (2)
password required /lib/security/$ISA/pam_deny.so                                      (3)
session required /lib/security/$ISA/pam_limits.so                                     (4)
```

[Start of Network Analysis]