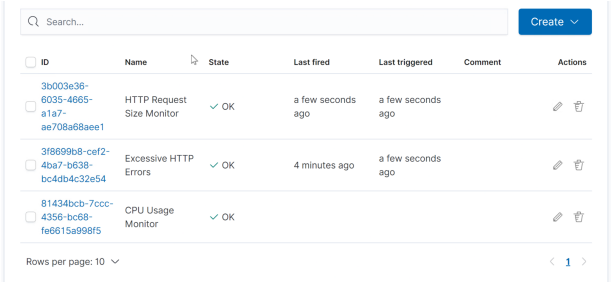
# Blue Team: Summary of Operations

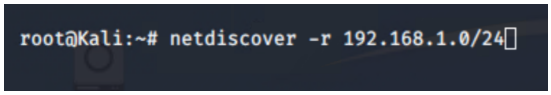
## Table of Contents



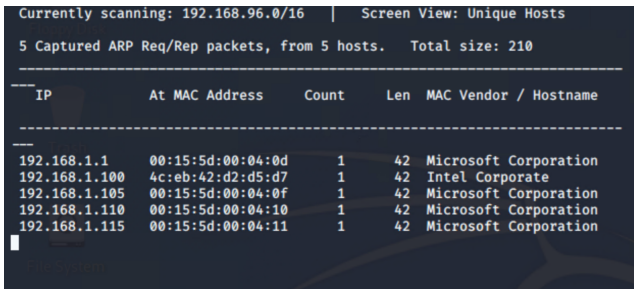
Creating Kibana Alerts (HTTP Request Size Monitor, Excessive HTTP Errors, CPU Usage Monitor)




Discovering IP information




IP Results




ARP-SCAN

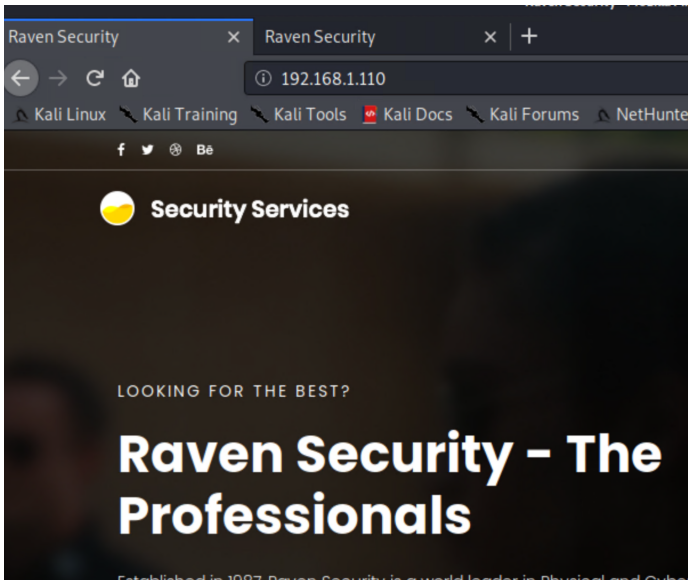
```
root@Kali:~# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:15:5d:00:04:12, IPv4: 192.168.1.90
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-sca
n)
192.168.1.1     00:15:5d:00:04:0d       Microsoft Corporation
192.168.1.100   4c:eb:42:d2:d5:d7       Intel Corporate
192.168.1.105   00:15:5d:00:04:0f       Microsoft Corporation
192.168.1.110   00:15:5d:00:04:10       Microsoft Corporation
192.168.1.115   00:15:5d:00:04:11       Microsoft Corporation

6 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.498 seconds (102.48 hosts/sec
). 5 responded
root@Kali:~#
```

NMAP

```
root@Kali:~# nmap -sS -A -T4 194.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-06 14:35 PDT
```

Target machine websites




Operating system information VM1

```
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Operating system information VM2

```
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET2; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

NMAP

```
|   Computer name: raven
|   NetBIOS computer name: TARGET1\x00
|   Domain name: local
|   FQDN: raven.local
|_  System time: 2021-05-07T07:59:49+10:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-05-06T21:59:49
|_  start_date: N/A

TRACEROUTE
HOP RTT      ADDRESS
1   10.49 ms 192.168.1.110

OS and Service detection performed. Please report any incorrec
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.30 seconds
root@Kali:~# nmap -p- -sS -AT4 192.168.1.110
```

```
root@Kali:~# nmap -p- -sS -AT4 192.168.1.110 >& target1.txt
```

NIKTO

```
root@Kali:~# nikto -host 192.168.1.110
- Nikto v2.1.6
```

```
root@Kali:~# nikto -host 192.168.1.115
- Nikto v2.1.6
```

This might be interesting:

```
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.110
+ Target Hostname:    192.168.1.110
+ Target Port:        80
+ Start Time:         2021-05-06 15:15:01 (GMT-7)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the u
ser agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user a
gent to render the content of the site in a different fashion to the MIME t
ype
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server may leak inodes via ETags, header found with file /, inode: 41b3,
size: 5734482bdcb00, mtime: gzip
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37).
 Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting...
```

DIRB

```
root@Kali:~# dirb http://192.168.1.110
```

```
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET2; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

## Network Topology

The following machines were identified on the network:

- Target 1
  - Operating System: Linux
  - Purpose: Apache Web Server
  - IP Address: 192.168.1.110

- Attacking Machine
  - Operating System: Kali Linux
  - Purpose: Threat Actor
  - IP Address: 192.168.1.90

- Elk Stack
  - Operating System: Ubuntu Linux
  - Purpose: Logging, Intrusion Detection System

    - IP Address: 192.168.1.100

- Capstone
    - Operating System: Ubuntu
    - Purpose: Sending logs to ELK & Apache Web Server
    - IP Address:192.168.1.105

- Azure
    - Operating System: Microsoft Windows RPC
    - Purpose: Jump Box/Azure Cloud Environment
    - IP Address: 192.168.1.1

Exposed ports:

- 22/tcp open SSH
- 80/tcp opeh http
- 111/tcp open
- 139/tcp open
- 445/tcp open

## Description of Targets

The target of this attack was: Target 1 192.168.1.110

## Monitoring the Targets

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

Alert 1 - Excessive HTTP Errors

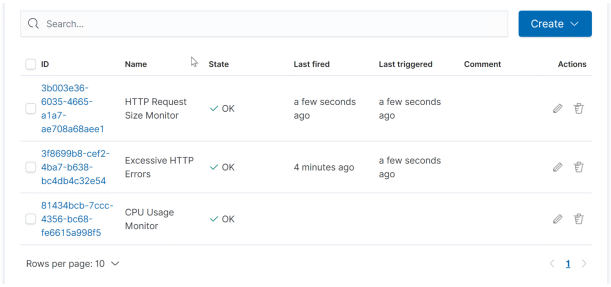
- **Metric**: when count grouped over 5 'http.response.status_code' IS ABOVE 400  FOR THE LAST 5 minutes
- **Threshold**: 400 (when the code is over 400 5 times in 5 minutes)
- **Vulnerability Mitigated**: Identifies failed login attempts to identify a possible brute force attack
- **Reliability**: The alert is highly reliable. Measuring by error codes 400 and above will filter out any normal or successful responses. 400+ codes are client and server errors which are of more concern. Especially when taking into account these error codes going off at a high rate.

Alert 2 - HTTP Request Size Monitor

- **Metric**: http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
- **Threshold**: 3500
- **Vulnerability Mitigated**: Code injection in HTTP requests (XSS and CRLF) or DDOS
- **Reliability**: TODO: Alert could create false positives. It comes in at a medium reliability. There is a possibility for a large non malicious HTTP request or legitimate HTTP traffic.

Alert 3 - CPU Usage Monitor

- **Metric**: system.process.cpu.total.pct OVER all documents
- **Threshold**: Above 0.5
- **Vulnerability Mitigated**: Malicious software, programs (malware or viruses) running taking up resources
- **Reliability**: The alert is highly reliable. Even if there isn't a malicious program running this can still help determine where to improve on CPU usage.




## Suggestions for Going Further (Optional)

The logs and alerts generated during the assessment suggest that this network is susceptible to several active threats, identified by the alerts above. In addition to watching for occurrences of such threats, the network should be hardened against them. The Blue Team suggests that IT implement the fixes below to protect the network:

### Excessive HTTP Errors

- Patch: WordPress Hardening
  - Implement regular updates to WordPress
    - WordPress Core
    - PHP version
    - Plugins
  - Install security plugin(s)

        - Ex. Wordfence (adds security functionality)
    - Disable unused WordPress features and settings like:
        - WordPress XML-RPC (on by default)
        - WordPress REST API (on by default)
    - Block requests to /?author= by configuring web server settings
    - Remove WordPress logins from being publicly accessible specifically:
        - /wp-admin
        - /wp-login.php
- Why It Works:
    - Regular updates to WordPress, the PHP version and plugins is an easy way to implement patches or fixes to exploits/vulnerabilities.
    - Depending on the WordPress security plugin it can provide things like:
        - Malware scans
        - Firewall
        - IP options (to monitor/block suspicious traffic)
    - REST API is used by WPScan to enumerate users
        - Disabling it will help mitigate WPScan or enumeration in general
    - XML-RPC uses HTTP as it's method of data transport
    - WordPress links (permalinks) can include authors (users)
        - Blocking request to view the all authors (users) helps mitigate against user enumeration attacks
    - Removal of public access to WordPress login helps reduce the attack surface

## HTTP Request Size Monitor

- Patch: Code Injection/DDOS Hardening
    - Implementation of HTTP Request Limit on the web server
        - Limits can include a number of things:
            - Maximum URL Length
            - Maximum length of a query string
            - Maximum size of a request
    - Implementation of input validation on forms
- Why It Works:
    - If an HTTP request URL length, query string and over size limit of the request a 404 range of errors will occur.
        - This will help reject these requests that are too large.
    - Input validation can help protect against malicious data anyone attempts to send to the server via the website or application in/across a HTTP request.