Blue Team: Summary of Operations

Table of Contents

- Network Topology
- Description of Targets
- Monitoring the Targets
- Patterns of Traffic and Behavior
- Suggestions for Going Further

Network Topology

The following machines were identified on the network:

- **Network**
- Address Range: 192.168.1.0/24
- Netmask: 255.255.255.0
- Gateway: 192.168.1.1

- **Machines**
- IPv4: 192.168.1.90
- OS: Linux
- Hostname: Kali Linux

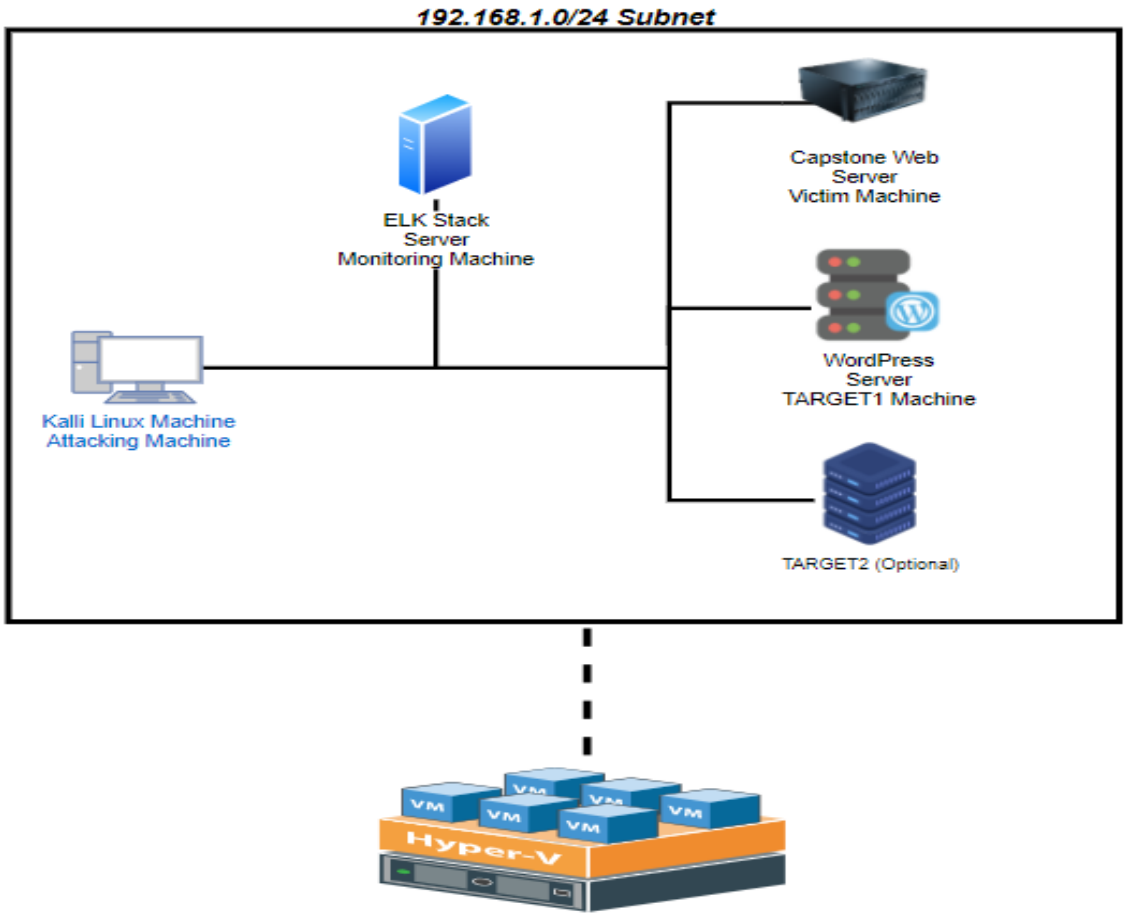
- IPv4: 192.168.1.110
- OS: Linux
- Hostname: TARGET1

- IPv4: 192.168.1.115
- OS: Linux
- Hostname: TARGET2

- IPv4: 192.168.1.100
- OS: Linux
- Hostname: Elasticsearch

- IPv4: 192.168.1.105
- OS: Linux
- Hostname: Capstone

- IPv4: 192.168.1.1
- OS: Windows
- Hostname: Hyper-V



Description of Targets

Fill in the following:

- Two VMs on the network were vulnerable to attack: Target 1 192.168.1.110 and Target 2 192.168.1.115.

- Each VM functions as an Apache web server and has SSH enabled, so ports `80` and `22` are possible ports of entry for attackers.

Monitoring the Targets

This scan identifies the services below as potential points of entry:

Target 1

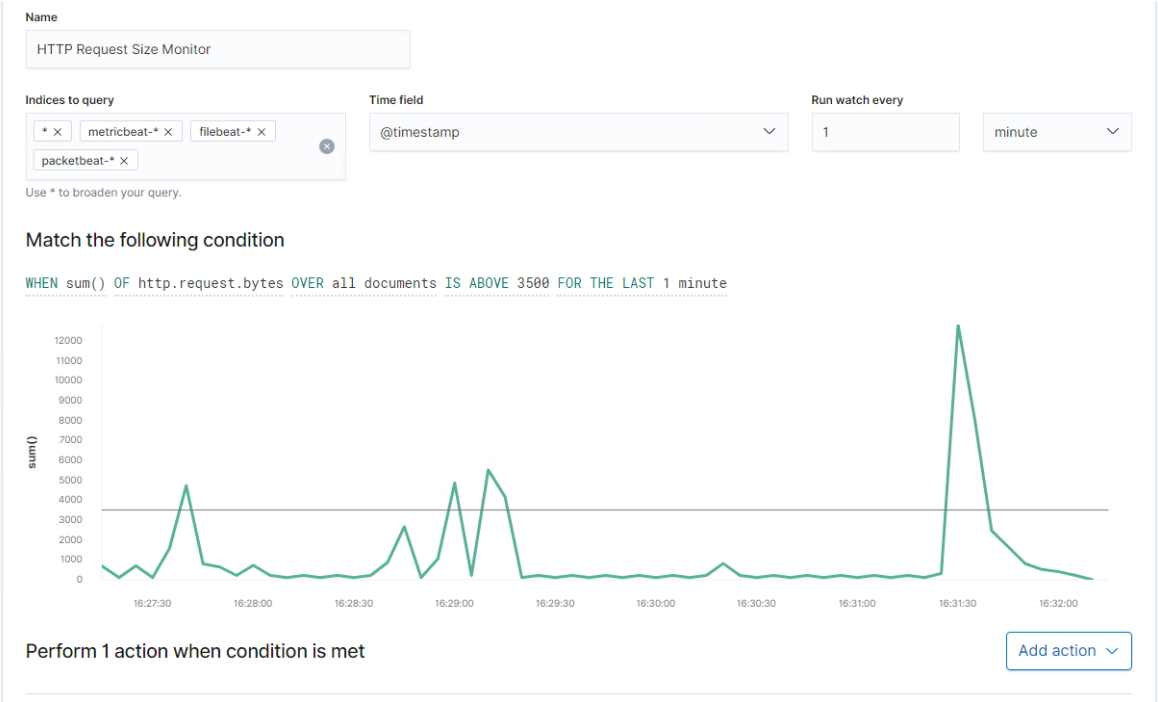
1. Port 22 OpenSSH
2. Port 80 open http Apache server
3. Port 111 rcpbind over open tcp
4. Port 139 netbios-ssn over open tcp
5. Port 445 netbios-ssn over open tcp

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below: (Note: Add at least three alerts. You can add more if time allows.)

HTTP Request Size Monitor

HTTP Request Size Monitor is implemented as follows:

- Metric: Sum of http.request.bytes
- Threshold: 3500 in 1 minute
- Vulnerability Mitigated: Port scans, brute force attack, DDos attack
- Reliability: Low, creates a lot of false positives. Should increase the threshold.



Excessive HTTP Errors

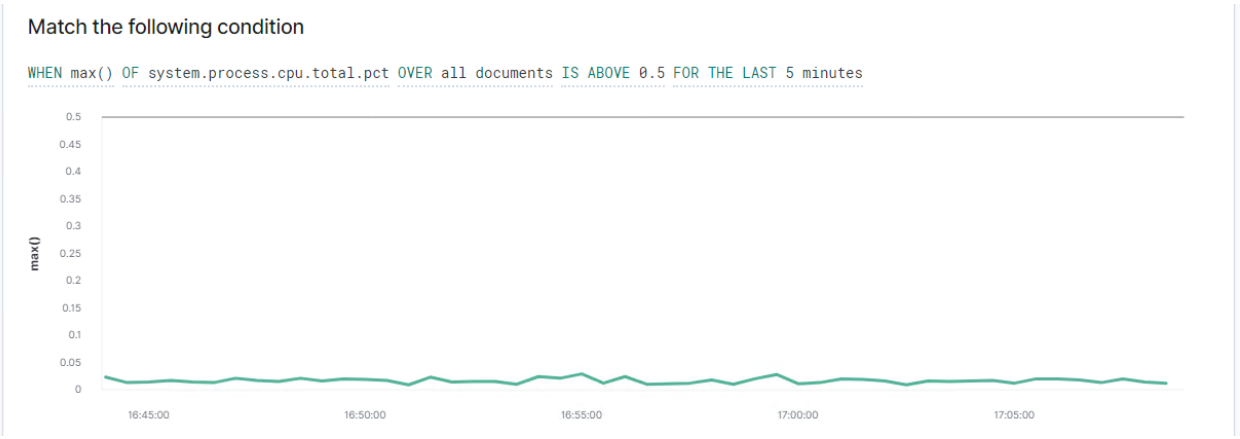
Excessive HTTP Errors is implemented as follows:

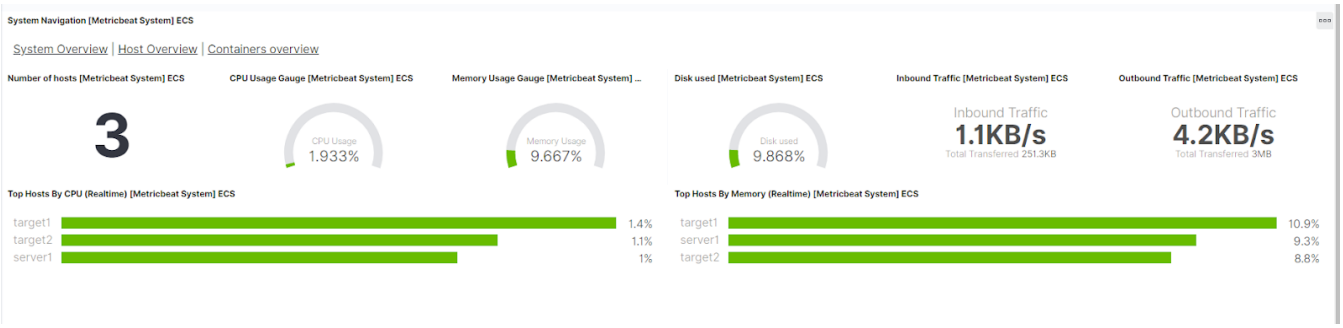
- Metric: Count of http.response.status_code
- Threshold: 400 in last 5 minutes
- Vulnerability Mitigated: Brute force attack
- Reliability: Medium, some legitimate activity may trigger this alert

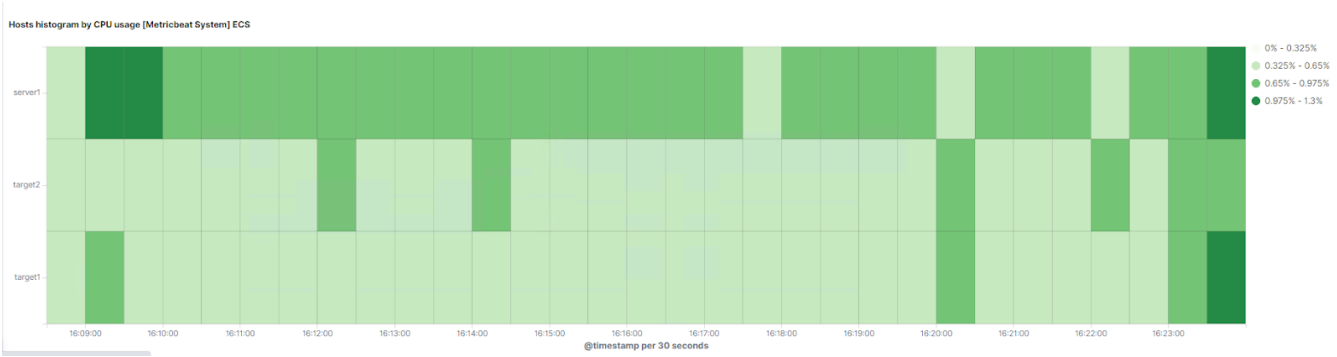
CPU Usage

CPU Usage is implemented as follows:

- Metric: Max of system.process.cpu.total.pct
- Threshold: 0.5 over last 5 minutes
- Vulnerability Mitigated: Brute force attack, port scan
- Reliability: High, generates little false positive.







Suggestions for Going Further

Suggest a patch for each vulnerability identified by the alerts above. Remember: alerts only detect malicious behavior. They do not prevent it. It is not necessary to explain how to implement each patch.

The logs and alerts generated during the assessment suggest that this network is susceptible to several active threats. 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:

Vulnerability 1: OpenSSH

- Patch: Whitelist approved sources to establish an ssh connection in the firewall. Use ssh keys and not simple passwords
- Why It Works: Blocks traffic from all unwanted sources and further encrypts user passwords

Vulnerability 2: Wordpress through http port 80

- Patch: Update Wordpress version
- Why It Works: Installs security patches reducing vulnerabilities

Vulnerability 3: Brute force ssh password

- Patch: Update lockout policies
- Why It Works: Minimizes the number of login attempts greatly reducing susceptibility to brute force attack

Vulnerability 4: netbios-ssn over open tcp ports 139 and 445

- Patch: Update Samba to make sure latest patches are installed
- Why It Works: Reduces vulnerabilities associated with the netbios-ssn tcp connection

Vulnerability 5: rcpbind over open tcp port 111

- Patch: Whitelist allowed hosts by updating hosts.allow and hosts.deny files
- Why It Works: Limits access to portmapper services to pre-approved hosts