# 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:

**[Name of VM 1]**

* Operating System: Linux
* Purpose: Wordpress server
* IP Address: 192.168.1.110

**[Name of VM 2]**

* Operating System: Linux
* Purpose: Wordpress server
* IP Address: 192.168.1.115

A screenshot of a cell phone

Description automatically generated

## Description of Targets

* 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:

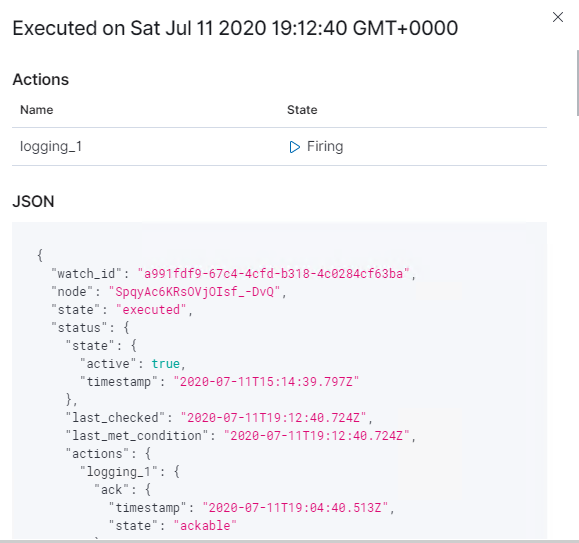
* **Target 1**
  + List of ports 22 , 80 and 111
  + Potentially vulnerable SSH , HTTP and rpcbind
  + Services wordpress service
* **Target 2**
  + List of ports 22 , 80 and 111
  + Potentially vulnerable SSH , HTTP and rpcbind
  + Services wordpress server

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

**Name of Alert 1**

# HTTP Requests size monitor is implemented as follows:

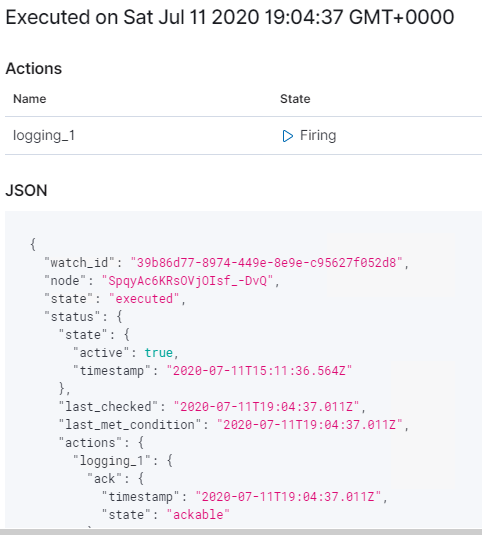
* Metric: HTTP request size monitor
* Threshold: 3500/1 minute
* Vulnerability Mitigated: Monitoring any malicious file transfer
* Reliability: This alert fired with many false positive and negative which reflect low reliability



**Name of Alert 2**

HTTP errors is implemented as follows:

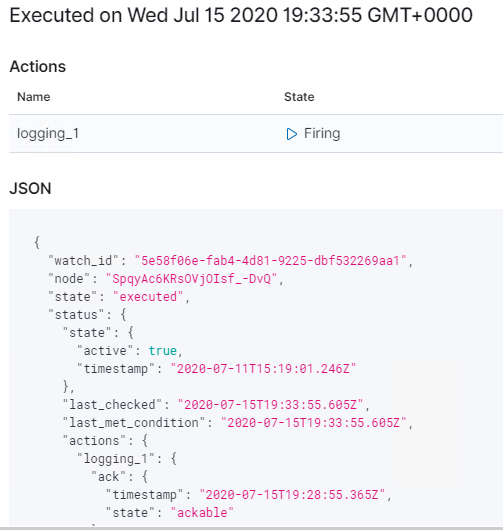
* Metric: HTTP errors
* Threshold: 400/5 minutes
* Vulnerability Mitigated: Identifies brute force attacks and directory enumerations
* Reliability: No false positives and it’s high reliable alert



**Name of Alert 3**

CPU usage is implemented as follows:

* Metric: CPU usage
* Threshold: 0.5/5 minutes
* Vulnerability Mitigated: Identify the machine’s uptime and when it has been compromised by malicious DB injection or hijacking it’s web sessions
* Reliability: The alert reflects some false positive which indicates medium reliability



## 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**

* Patch: The best way to mitigate the ssh vulnerability is to keep your server packages updated, and that also includes OpenSSH to do that run

apt-get update openssh-server

**Vulnerability 2**

* Patch: apt-get update

apt-get upgrade

apt-get install apache2-utils

* Why It Works: In Apache HTTP Server, you can achieve this by creating a .htpasswd file and adding a few configuration directives described below.

The .htpasswd file stores combinations of usernames and password hashes which the web server will use to authenticate users. You can create a .htpasswd file using the htpasswd command line or using an online password file generator.

Several Linux distributions install the htpasswd utility together with Apache itself, however, most Debian and Ubuntu users will need to install the apache2-utils package as follows.

Once htpasswd is installed, run the following command to create a new .htpasswd file with a single user. The following command will create a new .htpasswd file located at /srv/auth/.htpasswd with a username of myuser. htpasswd will then prompt you to enter and then confirm the password of your choice.

htpasswd -c /srv/auth/.htpasswd myuser

To enable basic HTTP authentication on the WordPress administration area, you need to activate the directive described below on the wp-admin directory and reference the .htpasswd file created earlier. Insert the following lines into the appropriate <Directory> section of your server’s Apache configuration file or in an .htaccess file within the wp-admin directory.

AuthType Basic

AuthUserFile /srv/auth/.htpasswd

AuthName "WordPress Authenticated area."

Require valid-user

The AuthType directive is specifying that the authentication type. In this case, Basic authentication is being configured.

The AuthUserFile directive specifies the full path to the .htpasswd file. This file is the file that shall be used to store password hashes which the server shall later use to authenticate users with.

The AuthName directive contains an arbitrary message which the browser will present to the user upon authentication. The Require valid-user setting simply instructs Apache to allow any valid user to authenticate.