**Brett Ott**

**Project 1 Submission**

**nanAutomated ELK Stack Deployment**

The files in this repository were used to configure the network depicted below.

Brett Ott Network Diagram

Diagram

Description automatically generated

<https://drive.google.com/file/d/1-2y04BLn1GEKQ6QhEQdmjEBpubpGTpk1/view?usp=sharing>

These files have been tested and used to generate a live ELK deployment on Azure. They can be used to either recreate the entire deployment pictured above. Alternatively, select portions of the YAML file may be used to install only certain pieces of it, such as Filebeat.

Text

Description automatically generated

<https://drive.google.com/file/d/1-3OBt3xyxq6W1nyGPUQbCeQ7xB3nukBO/view?usp=sharing>

This document contains the following details:

- Description of the Topology

- Access Policies

- ELK Configuration

  - Beats in Use

  - Machines Being Monitored

- How to Use the Ansible Build

**Description of the Topology**

The main purpose of this network is to expose a load-balanced and monitored instance of DVWA, the D\*mn Vulnerable Web Application.

Load balancing ensures that the application will be highly stable, in addition to restricting access to the network. Utilizing a load balancer helps to ensure that the network remains available to be used as intended, it also helps to protect the confidentiality of the network by restricting access to only those who are authorized to access it. Utilizing a jump box offers the advantage of acting as a first line defense against network intrusion as well as offering a consolidated starting point to administer the rest of the network.

Integrating an ELK server allows users to easily monitor the vulnerable VMs for changes to the system logs and system resources.

The configuration details and purpose of each machine in the network may be found below.

Chart, table, treemap chart

Description automatically generated

**Access Policies**

The machines on the internal network are not exposed to the public Internet. However, the Web1, Web2 and ELK machines can accept connections via Port 80 from my personal machine. The only whitelisted IP address is:

71.164.113.222

Machines within the network can also be accessed by SSH via Port 22 from the Jump Box machine at IP 10.1.0.4

A summary of the access policies in place can be found in the table below.

Table

Description automatically generated

**Elk Configuration**

Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because it doesn’t require any special coding knowledge to execute. Using Ansible playbooks also makes it very easy to create multiple copies of the same environment quickly.

The playbook implements the following tasks:

* Install docker module
* Install python3
* Increase the amount of memory available to the environment
* Download and install the ELK container
* Enable the docker container to start automatically on boot

The following screenshot displays the result of running `docker ps` after successfully configuring the ELK instance.



<https://drive.google.com/file/d/1-Dt9Dhe645T0jgSSN6qA277RanEq3uLc/view?usp=sharing>

**Target Machines & Beats**

This ELK server is configured to monitor the following machines:

Web1- 10.1.0.5

Web2- 10.1.0.6

We have installed the following Beats on these machines:

Filebeat

Metricbeat

These Beats allow us to collect the following information from each machine:

Filebeat monitors system logs for Web1 and Web2 and records changes such as access attempts. Metricbeat keeps an eye on the system resources for Web1 and Web2 and can graphically display the load on CPU Usage, Memory Usage etc. Knowing the baseline, or what normal looks like for each machine, helps to identify if something is off that may be an indicator of an attack, such as a denial of service attack.

-

**Using the Playbook**

In order to use the playbook, you will need to have an Ansible control node already configured. Assuming you have such a control node provisioned:

SSH into the control node and follow the steps below:

- Copy the install\_elk.yml file to the /etc/ansible directory.

- Update the hosts file to include the ip address for the ELK machine and include: ansible\_python\_interpreter=/usr/bin/python3 after the ip address

- Run the playbook, and navigate to 20.9.39.118:5601 to check that the installation worked as expected.