## Automated ELK Stack Deployment

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

Images/2\_Red\_Team\_Network\_With\_ELK

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, or alternatively, select portions of the playbook file may be used to install only certain pieces of it, such as Filebeat.

/etc/ansible/install-elk.yml

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 Damn Vulnerable Web Application.

Load balancing ensures that the application will be highly secure, in addition to restricting access to the network. Load balancers protect the network from DDos attacks. The advantage of a jump box is that it is a single access point that can be more secured.

Integrating an ELK server allows users to easily monitor the vulnerable VMs for changes to the files and system metrics. Filebeat watches for changes to the file system, while Metricbeat collects statistics and sends them to a specified output.

The configuration details of each machine may be found below.

| Name | Function | IP Address | Operating System |

|-----------------------------|-------------|---------------|-----------------------|

| Jump-Box-Provisioner | Gateway | 10.1.0.4 | Linux |

| Web-1 | Server | 10.1.0.5 | Linux |

| Web-2 | Server | 10.1.0.6 | Linux |

| Elk-VM | Server | 10.2.0.4 | Linux |

### Access Policies

The machines on the internal network are not exposed to the public Internet.

Only the jump box machine can accept connections from the Internet. Access to this machine is only allowed from the following IP addresses:

172.13.10.252

Machines within the network can only be accessed by the jump box. Only the Jump-Box-Provisioner (10.1.0.4) machine is allowed to access the Elk-VM.

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

| Name | Publicly Accessible | Allowed IP Addresses |

|-----------------------------|-------------------------|-----------------------------|

| Jump-Box-Provisioner | Yes | 172.13.10.252 |

| Web-1 | No | 10.1.0.4 |

| Web-2 | No | 10.1.0.4 |

| Elk-VM | No | 10.1.0.4 |

### Elk Configuration

Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because commands can be applied to multiple servers with one playbook.

The playbook implements the following tasks:

Install docker.io

Install python-pip

Increase max map count to 262144

Launch docker container- Elk

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

Images/Docker\_PS.png

### Target Machines & Beats

This ELK server is configured to monitor the following machines: 10.1.0.5/10.1.0.6

The following Beats have been installed on these machines: Filebeat and Metricbeat

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

Filebeat collects changes done to files.

Images/filebeat.png

Metricbeat collects statistics.

Images/metricbeat.png

### 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 playbook file to file directory: etc/ansible/file/filebeat-config.yml

- Update the playbook host file ( /etc/ansible/host) to include Elk-Vm IP Addresses

- Run the playbook, and navigate to kibana ( http://{Elk\_Public\_IP}/app/kibana) to check that the installation worked as expected.