**Automated ELK Stack Deployment**

The files in this repository were used to configure the network depicted in the Azure.ppt file which is also stored on the repository:

<https://github.com/JHaynes8183/Azure-Virtual-Network>

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

* Filebeat-playbook.yml is stored in the repo listed above

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 fault tolerant because of redundancy, in addition to restricting access to the network.

* The load balancer addresses availability. It provides the external IP address that the rest of the internet can access and it receives the traffic coming into the website, then distributes it across multiple servers.
* A jumpbox is used as a provisioner that automatically configures multiple machines identically at once instead of having to manually log in to each machine and configure them one by one.

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

* Filebeat watches for changes to specific files such as files generated by Apache, Azure, MySQL databases, etc.
* Metricbeat records specific information about the machines including performance metrics such as CPU usage, and uptime.

The configuration details of each machine may be found below.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Function | IP Address | Operating System |
| Jump Box | Gateway | 10.0.0.4 | Ubuntu (GNU/Linux) |
| DVWA-VM1 | Web Server | 10.0.0.17 | Ubuntu (GNU/Linux) |
| DVWA-VM2 | Web Server | 10.0.0.18 | Ubuntu (GNU/Linux) |
| DVWA-VM3 | Web Server | 10.0.0.19 | Ubuntu (GNU/Linux) |
| DVWA-VM4 | Web Server | 10.0.0.20 | Ubuntu (GNU/Linux) |
| ELK-Server | ELK Stack | 10.0.0.7 | Ubuntu (GNU/Linux) |

**Access Policies**

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

Only the Jump-Box-Provisioner machine can accept connections from the Internet. Access to this machine is only allowed from the following IP addresses:

* 74.194.137.8 (my home IP)

Machines within the network can only be accessed by SSH.

* The Jump-Box-Provisioner is allowed to access the ELK-Server from it’s private IP address of 10.0.0.4 using SSH

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

|  |  |  |
| --- | --- | --- |
| Name | Publicly Accessible | Allowed IP Addresses |
| Jump-Box-Provisioner | Gateway | 74.194.137.8 (home IP) |
| DVWA-VM1 | Web Server | 10.0.0.4 (jump box) |
| DVWA-VM2 | Web Server | 10.0.0.4 (jump box) |
| DVWA-VM3 | Web Server | 10.0.0.4 (jump box) |
| DVWA-VM4 | Web Server | 10.0.0.4 (jump box) |
| ELK-Server | ELK Stack | 10.0.0.4 (jump box) and 74.194.137.8 (home IP) |

**Elk Configuration**

Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because:

* Automating configuration with Ansible allows you to configure multiple machines at once time with accuracy.

The install-elk.yml playbook implements the following tasks:

* Install docker.io
* Install pip
* Install the docker python module using pip
* Increase the virtual memory of the ELK Server
* Download and launch the docker elk container and identify which published ports the container can be started with

The following screenshot displays the result of running docker ps after successfully configuring the ELK instance. *<When I moved all my VMs behind the load balancer, I could not start my ELK-Server container any longer, not sure what I did to break it. Tried running the install-elk.yml playbook again a couple of times and it always ran successfully, but I still could not find the container on the ELK-Server any longer>*

**Target Machines & Beats**

This ELK server is configured to monitor the following machines:

* DVWA-VM1 at 52.233.92.69 external / 10.0.0.17 internal
* DVWA-VM2 at 52.233.92.70 external / 10.0.0.18 internal
* DVWA-VM3 at 52.148.153.245 external / 10.0.0.19 internal
* DVWA-VM4 at 52.156.107.98 external / 10.0.0.20 internal

While ELK supports 8 Beats, we have installed only the following 2 Beats on these machines:

* Filebeat was installed on the ELK-Server VM. Filebeat helps generate and organize log files on specified files and then organizes and sends the log files to Logstash and Elasticsearch
* Metricbeat was also installed on the ELK-Server VM. Metricbeat collects metrics about specific machines.

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

* Filebeat will collect logs to monitor the Apache server and the MySQL database logs generated by the DVWA virtual machines.
* Metricbeat will collect machine uptime and provide alerts when CPU usage gets too high.

**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 filebeat-configuration.yml file to etc/ansible/files and the filebeat-playbook.yml file to /etc/ansible/roles.
* Update the filebeat-configuration.yml file to include the internal IP address of the ELK-Server.
* Run the playbook, and navigate to the ELK-Server to check that the installation worked as expected.

Answer the following questions to fill in the blanks:

* The filebeat-playbook.yml file is the actual playbook.
* The filebeat-configuration.yml file is updated to run the playbook on a specific machine by adding the IP of the ELK-Server to the output.elasticsearch section and the setup.kibana section of the filebeat-configuration.yml file.
* Navigate to the public IP address and port 5601 to confirm that the ELK-Server is running – http://52.229.9.203:5601

**Bonus** - specific commands the user will need to run to download the playbook, update the files, etc.

Create the filebeat-configuration.yml file

* *nano filebeat-configuration.yml*
* Paste the file from [GitHub](https://github.com/the-Coding-Boot-Camp-at-UT/UT-MCC-CYBER-PT-01-2020-U-C/blob/master/Week%2014%20-%20Cloud%20Sec%20%26%20Project/Elk-Stack-Project/Activities/Stu_Day_2/Solved/config_files/filebeat-configuration.yml) into the filebeat-configuration.yml file
* Modify the file in nano by replacing the IP address with my ELK-Server internal IP address for elasticsearch and kibana then save to /etc/ansible/files
* Ctrl-x to save file

Create the filebeat-playbook.yml file

* *Nano filebeat-playbook.yml*
* Download the .deb file
* *dpkg -I filebeat-7.4.0-amd64.deb*
* Copy file from ansible container to ELK-Server
* *./filebeat modules enable system*
* *./filebeat setup*
* *.filebeat -e*

Install filebeat using the playbook

* *ansible-playbook filebeat-playbook.yml*