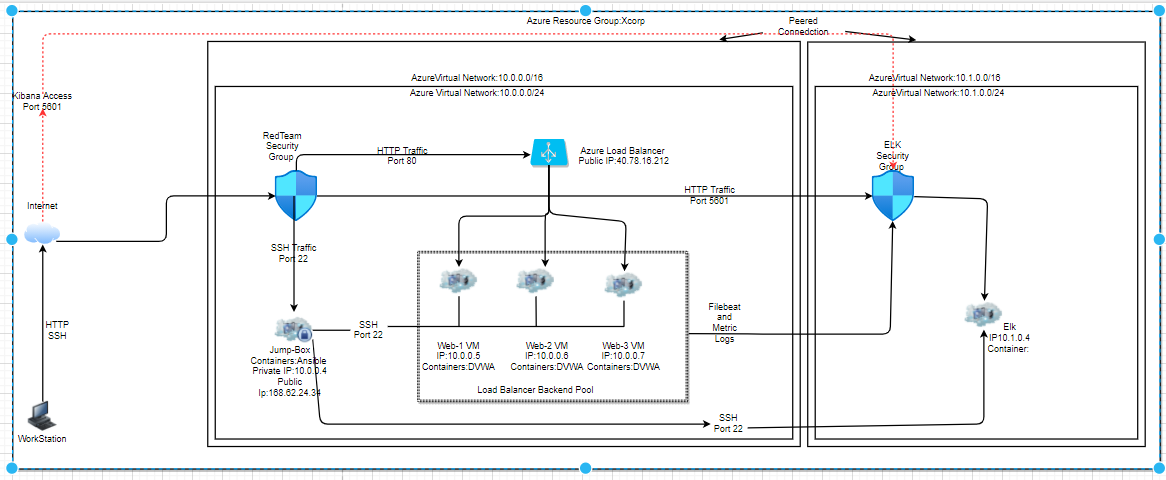
**CyberSecurity Project 1**

**Automated ELK Stack Deployment**

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



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 ansible playbook file may be used to install only certain pieces of it, such as Filebeat.

Listed here are the Playbooks utilized in the Deployment depicted above:

* [ELK-Install file](file:///C:\Users\mifta\Iftee-Cybersecurity-Project\Ansible\install-elk.yml)
* [Filebeat-playbook file](file:///C:\Users\mifta\Iftee-Cybersecurity-Project\Ansible\install-filebeat.yml)
* [Metricbeat-playbook file](file:///C:\Users\mifta\Iftee-Cybersecurity-Project\Ansible\install-metricbeat.yml)

***An example of the Filebeat-playbook can be seen here:***

# ---

- name: Install Filebeat

hosts: webservers

remote\_user: azdmin

become: true

tasks:

# Download.deb

- name: Download.filebeat.deb

command: curl https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.4.0-amd64.deb -O

# Install filebeat

- name: Download filebeat.deb

command: dpkg -i filebeat-7.4.0-amd64.deb

# Copy filebeat config

- name: Copy config files

copy:

src: /etc/ansible/files/filebeat-configuration.yml

dest: /etc/filebeat/filebeat.yml

# Setup filebeat

- name: setup filebeat

command: filebeat modules enable system

- command: filebeat setup

- command: sudo service filebeat start

**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 available (provides redundancy for our DVWA server), in addition to restricting http/web traffic to the network.

* **What aspect of security do load balancers protect?**
  + Load Balancers are intelligent network security devices that are responsible for effectively distributing the incoming network traffic to various servers in the network. Load balancers conduct continuous health checks on servers to ensure they can handle requests. Thus, in terms of the CIA triad, Load balancers protect the availability of resources. Load Balancing plays important role as it adds additional layers of security to network.
    - For Example: Protection against DDoS attack: The load balancer can detect and drop distributed denial-of-service (DDoS) traffic before it gets to your website.
* **What is the advantage of a jump box?**
  + Having a Jump box on a network, such as the Jump\_Box\_Provisoner on the XCorp's Red Team Network Diagram, is advantageous because it acts as a gateway router between VMs. In turn, forcing traffic through a single node.
    - Securing and monitoring a single node is called fanning in. This process makes it simpler rather than having to secure and monitor each VM behind the gateway. The jump box is exposed to the public internet, and in the case of XCorp's Red Team Network Diagram, is connected to the jump box’s SSH port (22). It controls access to the other machines by allowing connections from specific IP addresses and forwarding to those machines.

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

* **What does Filebeat watch for?**
  + Filebeat module collects/ watches for and analyzes logs created by the system logging service of common Unix/Linux based distributions. Specifically, it logs information about the file system, including when and which files have changed. Filebeat helps keep things simple by offering a lightweight way (low memory footprint) to forward and centralize logs and files, making the use of SSH unnecessary when you have a number of servers, virtual machines, and containers that generate logs.
* **What does Metricbeat record?**
  + Metricbeat collects/records metrics and statistics from the operating system and from services running on the server.

***The configuration details of each machine may be found below.***

| **Name** | **Function** | **IP Address** | **Operating System** |
| --- | --- | --- | --- |
| Jump-Box-Provisioner | Gateway | 10.0.0.4  Public IP:168.62.24.34 | Linux (ubuntu 18.04) |
| Red\_Web\_1 | DVWA (Server) | 10.0.0.5 | Linux (ubuntu 18.04) |
| Red\_Web\_2 | DVWA (Server) | 10.0.0.6 | Linux (ubuntu 18.04) |
| ELK\_VM | ELK Stack (Log-Server) | 10.1.0.4 | Linux (ubuntu 18.04) |

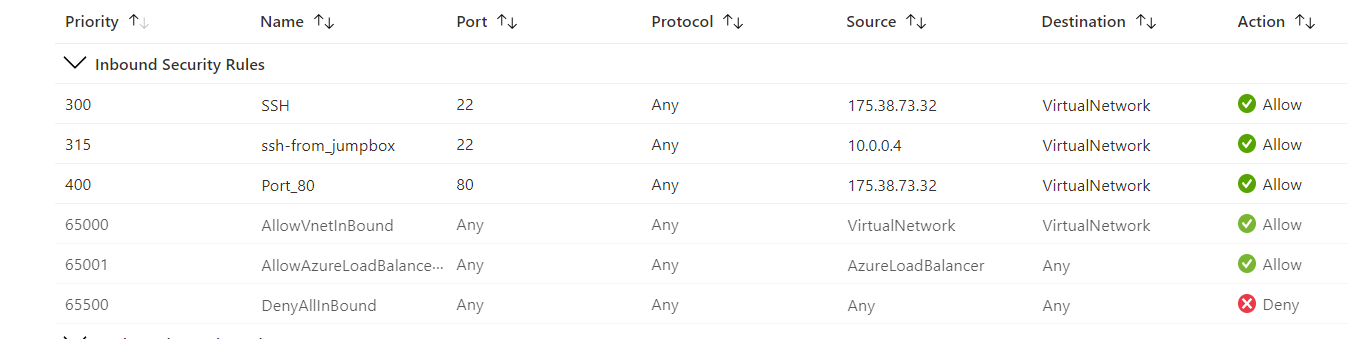
**Access Policies**

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

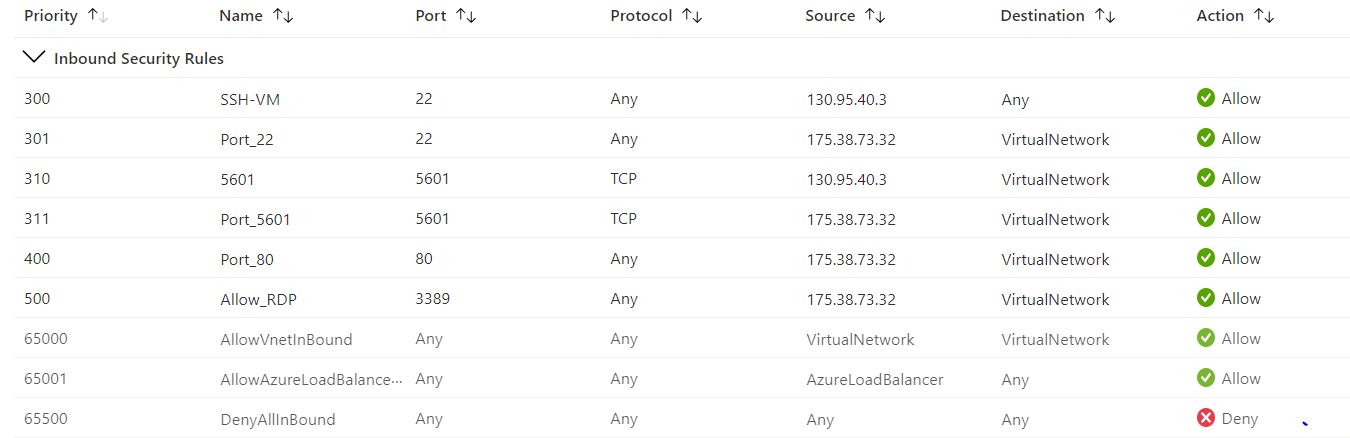
* Only the Load Balancer can accept connections from the Internet. Access to this machine is only allowed from the following IP addresses:
  + 175.38.73.32 (My personal IP)
  + **40.78.16.212** (Load Balancer IP)

***An example of how the Red-Team-NSG (Network-Security-Group) is setup can be seen here:***

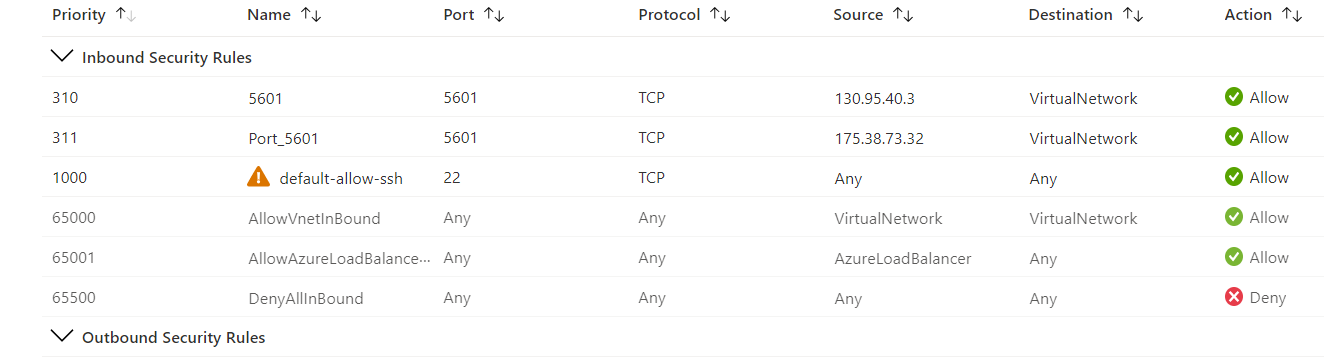
## Jump-Box-Provisioner-nsg



## UWA\_RedTeamXCorp\_Iftee\_NSG1



## Elk-VM-1-nsg



* Machines within the network can only be accessed by Jump-Box-Provisioner.
* **Which machine did you allow to access your ELK VM?**
  + When configuring the ELK VM, I needed to ensure that my Local machine had access to the ELK stack web server running on port 5601. In my ELK-NSG, I created an incoming rule that allowed for TCP traffic over this port from my public IP address (175.38.73.32).
    - **note**: The Red-Team-NSG is set up to allow SSH from 10.0.0.4 (Jump-Box-Provisioner’s Private IP Address) to the Red-Team Virtual Network. This pertains to the ELK-Network because a Peer Connection is set up between the vNets. This will allow traffic to pass between vNets and regions. The connection that took place in this deployment was configured to allow traffic to pass in both directions. The ELK-VM must use the same SSH keys as the WebVM's. This is where the ssh keys were created on the Ansible container that's running on the Jump-Box-Provisioner.

***SSH from the Ansible container shell (on the Jumpbox VM) to ELK-VM:***

#Example:

root@a33e3954dcda# cd /root/.ssh

root@a33e3954dcda:/root/.ssh# ssh azadmin@10.1.0.4

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 | My Public IP Address: 175.38.73.32 |
| Load Balancer | Yes | 40.78.16.212 |
| Web-1 | No | 10.0.0.4 |
| Web-2 | No | 10.0.0.4 |
| Web-3 | No | 10.0.0.4 |
| ELK-VM | No | 10.0.0.4 /10.0.0.5/ 10.0.0.6/ 10.0.0.7/ My Public IP Address |

**Elk Configuration**

Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because we would have had to separately configure an Elasticsearch database, a Logstash server, and a Kibana server, wire them together, and then integrate them into the existing network. Which would have required 3 more Virtual Machines. Instead, we are utilizing Ansible container to streamline/ simplify configuration and installation all at once.

***The playbook implements the following tasks:***

1. Install docker.io; Installing Docker engine to be used for running containers.

# Use apt module

- name: Install docker.io

apt:

update\_cache: yes

name: docker.io

state: present

1. Install python3-pip; Package used to install Python software.

# Use apt module

- name: Install python3-pip

apt:

force\_apt\_get: yes

name: python3-pip

state: present

# Use pip module (It will default to pip3)

1. Install Docker module; Installs the Pip packages. This configuration is required by Anisble to control the state of Docker containers.

# Use pip module (It will default to pip3)

- name: Install Docker module

pip:

name: docker

state: present

1. Increase the virtual memory; The ELK container will not run without this setting. Thus, we configure the target VM (the machine being configured) to use more memory.

# Use command module

name: Increase virtual memory

command: sysctl -w vm.max\_map\_count=262144

1. Use Sysctl module; We need to use/ configure this so that the setting is automatically ran if the VM has been restarted.

# Use sysctl module

- name: Use more memory

sysctl:

name: vm.max\_map\_count

value: 262144

state: present

reload: yes

1. Downloads the docker container called 'sebp/elk:761'; The configurations in this container are set to follow port mappings.

# Use docker\_container module

- name: download and launch a docker elk container

docker\_container:

name: elk

image: sebp/elk:761

state: started

restart\_policy: always

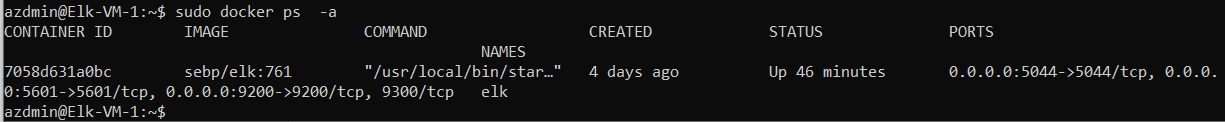
published\_ports:

- 5601:5601

- 9200:9200

- 5044:5044

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



**Target Machines & Beats**

This ELK server is configured to monitor the following machines:

* 10.0.0.5 ,10.0.0.6 and 10.0.07

We have installed the following Beats on these machines:

* Filebeats
* Metricbeats

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

* **Filebeats:**
  + We can use Filebeat to collect, parse, and visualize ELK logs in a single command. Specifically, it logs information about the file system, including when and which files have changed. In the case of this Deployment, we will be using Filebeats to monitor the Apache server and MySQL database logs generated by DVWA.
* **Metricbeats:**
  + Metricbeat collects/records metrics and statistics from the operating system and from services running on the server. Metricbeat provides insight into how much work the machine is doing. Excessively high CPU usage is typically a cause for concern, as overworked computers are at greater risk for failure.

**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: -**note:** The Filebeat installation instructions require you to create a Filebeat configuration file. This file is edited so that it has the correct settings to work with my current ELK server.

**SSH into the control node and follow the steps below:**

* Copy the install-elk.yml file to the /etc/ansible/ directory on your ansible node machine (where ansible container is running).
* Update the install-elk.yml file to reflect the hosts you would like to be affected by the ansible playbook (in this example, our host was called [elk] within the ansible hosts file)
* Run the playbook (ansible-playbook install-elk.yml), and navigate to [your\_elkserver\_ip]:5601 to check that the installation worked as expected.

**Which file is the playbook? Where do you copy it?**

* The Filebeat-configuration is the playbook and you copy the /etc/ansible/file/filebeat-configuration.yml to the destination of the webserver's /etc/filebeat/filebeat.yml

# Copy filebeat config

- name: Copy config files

copy:

src: /etc/ansible/files/filebeat-configuration.yml

dest: /etc/filebeat/filebeat.yml

**Which file do you update to make Ansible run the playbook on a specific machine? How do I specify which machine to install the ELK server on versus which to install Filebeat on?**

* Edit the /etc/ansible/host file to add webserver/elkserver ip addresses.
  + [webservers] and [elk] are groups. When you run the playbooks with Ansible, you speficy which group to run them on. By this, you can run certain playbooks on some machines and not others.

**Which URL do you navigate to in order to check that the ELK server is running?**

# Use the public IP address of your new VM.

http://[your.ELK-VM.External.IP]:5601/app/kibana

**Verify that you can access your server:**

* Navigate to the Filebeat installation page on the ELK server GUI.
* On the same page, scroll to Step 5: Module Status and click Check Data.
* Scroll to the bottom of the page and click Verify Incoming Data.