AUTOMATED ELK STACK DEVELOPMENT

TANVEER KHAN

TABLE OF CONTENTS

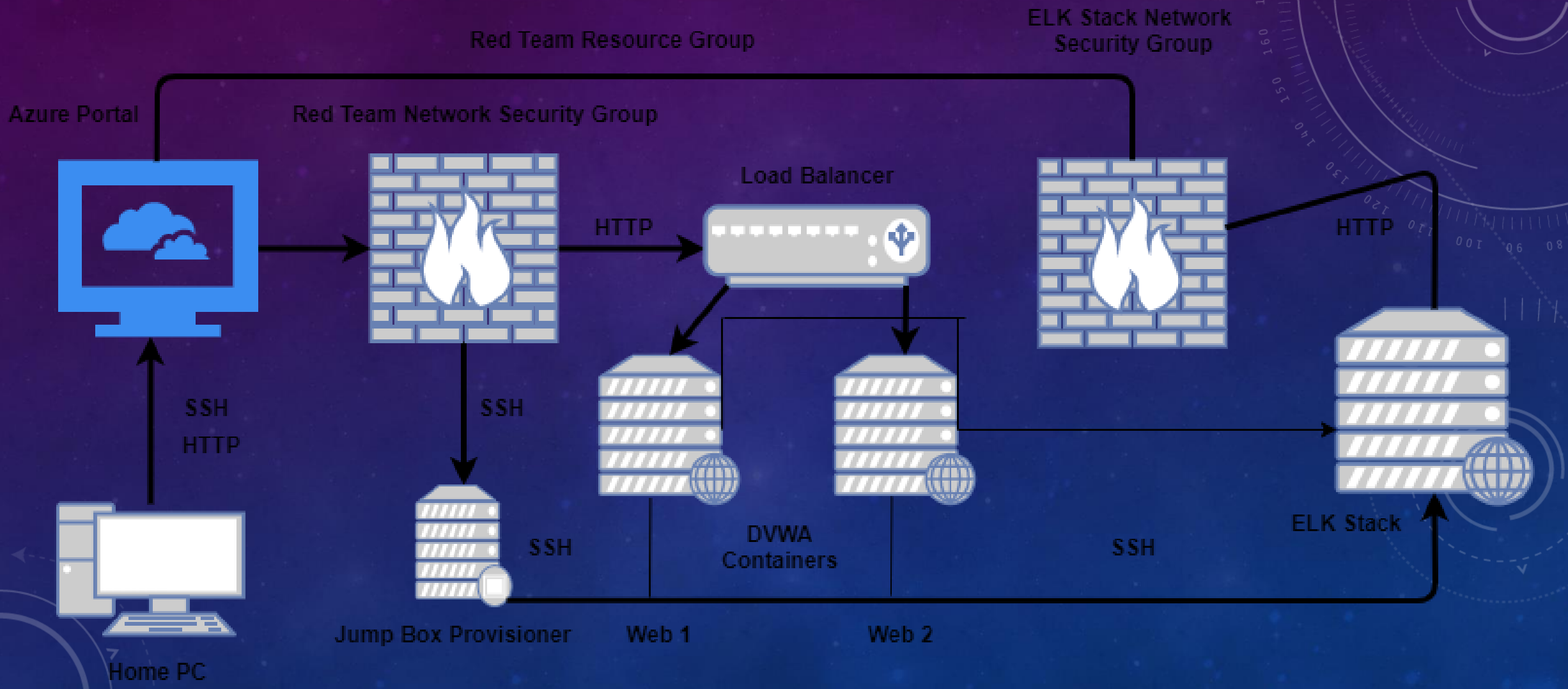
This document contains the following details:

- Description of the Topology
- Access Policies
- ELK Stack Configuration
  - Beats in Use
  - Machines Being Monitored
- How to Use the Ansible Build

DESCRIPTION OF THE TOPOLOGY

- The primary objective of this network is to offer a load-balanced and monitored instance of the D*mn Vulnerable Web Application, or DVWA.
- Load balancing ensures that the application is highly available for clients while also limiting network threats. Services will continue even if a server goes down or needs to be updated.
- The benefit of deploying a jump box is that only the jump box can use SSH to connect to the virtual network.
- Users can quickly monitor the susceptible VMs for changes to the logs and system traffic by integrating an ELK server.
    - Filebeat keeps track of the specified log files or locations, gathers log events, and sends them to Elasticsearch or Logstash for indexing.
    - Metricbeat is a metric shipper that is exceptionally simple to use, efficient, and effective for monitoring your system and the operations that operate on it.

Network Topology



THE NETWORK

Below are the configuration details for each machine:

Name	Function	IP Address	Operating System
Jump Box Provisioner	Gateway	Public: 40.86.73.173 Private: 10.0.0.8	Linux
Web-1	Server	Private: 10.0.0.9	Linux
Web-2	Server	Private: 10.0.0.10	Linux
ELK Stack Server	Monitor	Public: 52.137.81.98 Private: 10.2.0.4	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:
  - 5061 Kibana port
- Machines within the network can only be accessed by the Jump Box Provisioner.

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

Name	Publicly Accessible	IP Address
Jump Box	Yes	73.73.60.19
Web-1	No	10.0.0.5
Web-2	No	10.0.0.6
ELK Stack	No	10.0.0.4

ELK STACK CONFIGURATION

- The ELK virtual machine's configuration was automated using Ansible. There was no manual configuration, which was beneficial because it was simple and prevented any easily refuted weaknesses.
- The playbook implements the following tasks:
  - Install docker.io
  - Install python3-pip
  - Install docker via pip
  - Increase virtual memory
  - Download and launch a docker ELK Stack container – This initiates docker and establishes the ports being used.
- The following screenshot displays the result of running docker ps after successfully configuring the ELK instance:

```
sysadmin@ELK-Stack:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                                                                                  NAMES
592624d80386        sebp/elk:761        "/usr/local/bin/star…"   9 days ago          Up 25 minutes       0.0.0.0:5044->5044/tcp, 0.0.0.0:5601->5601/tcp, 0.0.0.0:9200->9200/tcp, 9300/tcp   elk
```

TARGET MACHINES & BEATS

This ELK server is configured to monitor the following machines:

Name	IP Address
Web-1	10.0.0.5
Web-2	10.0.0.6

I have installed the following Beats on these machines:

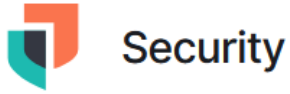
Name	IP Address
Web-1	10.0.0.5
Web-2	10.0.0.6
ELK Stack Server	

We may obtain the following data from each machine using these Beats:

- Filebeat gathers log information and displays it in monitoring clusters.
- Metricbeat gathers metrics and statistics and displays them in a given output, such as Elasticsearch or Logstash.

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 (.yml) file to Ansible directory. /etc/ansible
  - Update the host file to include webserver and ELK.
  - Run the playbook and navigate to Kibana to check that the installation worked as expected.
    - (http://[your.VM.IP]:5601/app/kibana)
  - If successful, you should see the following webpage:



APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

Add APM

Logs

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

Add log data

Metrics

Collect metrics from the operating system and services running on your servers.

Add metric data

SIEM

Centralize security events for interactive investigation in ready-to-go visualizations.

Add events

Add sample data
Load a data set and a Kibana dashboard

Upload data from log file
Import a CSV, NDJSON, or log file

Use Elasticsearch data
Connect to your Elasticsearch index

Congratulations on your first step toward network automation!