

**ANALYSIS OF CLOUD NETWORK INTRUSION ISSUE
AND A FEASIBLE DETECTION METHODOLOGY**

(A CAPSTONE PROJECT REPORT)

Submitted By

RANGITH RAVINDRAN NARAYANAGURU

A report submitted to the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In partial fulfilment of the requirement for the degree

Of

Master of Science

in

Internetworking

University of Alberta

(2011 - 2013)

Acknowledgement

Foremost, I would like to express my gratitude to my Project Supervisor Mr. Raghavendra Jayaram, for continuous guidance through all the stages of my Project and helping me out at difficult times.

My sincere thanks also goes to Dr. Mike MacGregor, Professor and Chair for Department of Computer Science, University of Alberta, for granting permission to implement my ideas into this project and also for encouraging me with his positive views about the project.

I would also like to thank Mr. Shahnawaz Mir, Program Coordinator, MINT, University of Alberta, for helping me by providing necessary tools at the right time to complete my project work.

TABLE OF CONTENTS

S.No	TITLE	Page no.
1	Introduction	4
2	Cloud Service Models	5
3	Cloud Deployment Models	6
4	Problem Statement	7
5	Security issues in Cloud Environment	7
6	Method of Examination	11
7	Reproducing a Network Attack on Cloud	13
8	Problem Analysis	19
9	Mitigation Plan	23
10	Architecture for Proposed NIDS model	26
11	Conclusion	31

- I. Appendix A - Hardware and Software configuration details
- II. Appendix B - Instructions for setting up Cloud network
- III. Appendix C - Blackbuntu test result screenshots
- IV. References

INTRODUCTION

Cloud computing has become one of the most exciting and rapidly growing technologies in the IT field. Cloud computing is typically the use of computing resources like networks, servers, storage, applications and services shared among number of end users over a network. Cloud service providers deliver the service of resource sharing over the internet and manage the infrastructure and platforms on which the applications are run with ease. Cloud is an on demand self service which is provisioned rapidly and managed effectively with minimal effort.

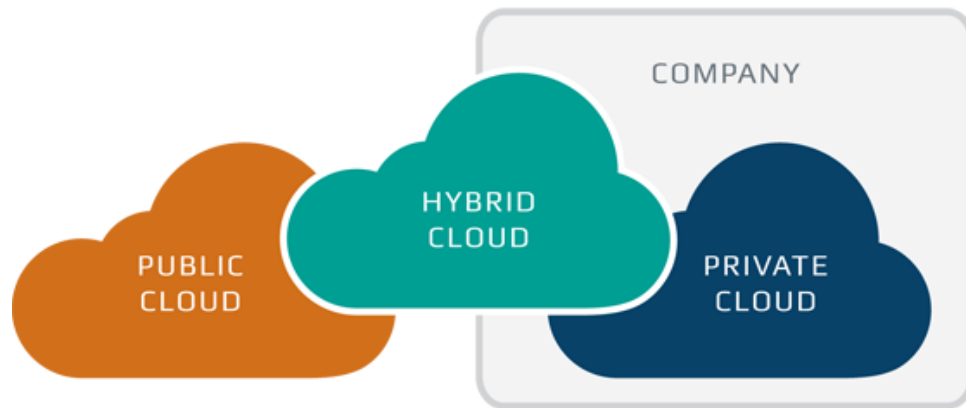
CLOUD SERVICE MODELS

Platform as Service - The PaaS- model provides a platform wherein the developers are provided with a programming and runtime environment. Such an environment can be used by the developers to develop, execute and deploy their own applications by utilizing the service provided by the cloud platform. Some examples are Windows Azure, Google App Engine etc.

Infrastructure as Service - IaaS-model provides the cloud compute resources in a virtualized environment i.e., it provides access to virtualized computer hardware resources which include machines, network and storage. All the compute resources that are provided are typically virtual machines, which includes memory, network resources, memory and storage as well as Operating system. The user is provided with the access the provision the resources as per need and use them as much as possible but cannot control the underlying hardware and network.

Software as service - SaaS-model provides the user with access to software application that are provided by a service provider. Some examples are Salesforce, Dropbox, Google, Zoho etc. The users can only access and use the application but do not have any control on the management of the application. The SaaS provider maintains the application as well as the user data and provides the storage and/or backup. SaaS is a simple and quick way to implement applications.

CLLOUD DEPLOYMENT MODELS



Public Clouds: An organization which owns the cloud services will rent out their resources to general public based on their need and customers have the option of self-provisioning the available resources through a web interface. Public cloud reduces the costs that the customers have to incur like the investments on data center infrastructure. In public clouds all such costs are shared between customers.

Private Cloud: In this case, the computing resources are hosted within an organization's infrastructure and the resources are dedicated just for that organization's sole use. The computing resources are totally under the organization's control and all the sensitive data are secured behind the organization's firewall.

Hybrid Cloud: As the name suggests it combines one or more public and private clouds and all the data and application are migrated between them with high use of technology. It provides the advantages of both public and private cloud i.e., organizations can utilize the cost benefits of a public cloud as well as the confidential data stays safe as in a private cloud.

PROBLEM STATEMENT

This project intends to do a study on the network intrusion issues to identify the key issues that has to be addressed and also to identify requirements that has to be met by a suitable intrusion detection system. Then with the above said results, I would like to analyse the network using any of one of the different intrusion detection methodologies like SNORT, tcpdump, Network Flight recorder or tamandua that are already in use. This analysis would help me find the network vulnerabilities that the private cloud network is facing and I would like to sum up by suggesting on what could be done to prevent the network from such vulnerabilities.

SECURITY ISSUES IN CLOUD ENVIRONMENT

Security issues in any network is bottleneck to the system and needs to be prevented, rather than addressing the issue after the attack occurred. In cloud networks, preserving the network from hackers is even more important because of the number of users affected and the scale of damage to network, systems and data that it would cost. At network layer cloud suffers from some of the traditional attacks such as IP spoofing, Address Resolution Protocol (ARP) spoofing, Routing Information Protocol (RIP) attack, DNS poisoning, man-in-the-middle attack, port scanning, Insider attack, Denial of Service (DoS), Distributed Denial of Service (DDoS) etc.

1. **Denial of Service attack**, one of the most common attacks performed by hackers around the world. This attack aims at making a cloud network unavailable for its users. The hacker floods the cloud network host with too many invalid requests, that the host uses all its resources to respond to all the invalid requests. In such cases the legitimate requests are denied a response. DDoS is similar to DoS, but the specialty of DDoS is that the attacks do not come from a single network or host but from a number of different hosts or networks which have been previously compromised. In cloud computing where infrastructure is shared by many users the impact of DDoS is extremely devastating.

2. **Port scanning**, it is one of the great tools for a network administrator to identify vulnerabilities but, it is also one of the most popular techniques used by the hackers to identify the network and its services that they can break into. Lets consider a cloud server which offers so many services to its users and listens to different ports that are well-known. In such cases the intruder can just run one of the many port scanning tests and identify the ports that can be exploited. One of the most preferred port scanning techniques by the hackers is Stealth scan, (SYN & FIN) which prevents logging of the scan. Since most of the port scans are logged by the services that are listening to the port, sees an incoming connection but no data, so it logs an error. SYN scan is also called as half open scan that opens the TCP connection partially and stops it halfway through. The port scanner used in this case will generate a SYN packet and send it to the target port. If the target port is open it will send back an acknowledgement packet immediately. Once the port scanner received the SYN-ACK it will send back a RST packet and closes the connection even before the handshake is complete. This stops the service from being notified about the incoming connection.

3. In **Spoofing**, the attacker creates an IP packet that is forged i.e., packet that uses a different IP address than the actual IP of the attacker. This is in general called IP spoofing. Since a forged IP address is used, the attacker might not see a response back from the victim server, in our case the cloud server probably due to misdirection of IP packets. This means that IP spoofing does not allow the attacker to have a well formed connection with the victim server. **Source Routing** is a technique whereby the sender of a packet can specify the route that a packet should take through the cloud network. Remember that as a packet travels through the network, each router will examine the "destination IP address" and choose the next hop to forward the packet to. In source routing, the sender makes some or all of these decisions. In **strict** source routing, the sender specifies the exact route the packet must take, which is virtually never used.

4. The **SYN flood attack** sends TCP connections requests faster than a cloud server can process them and is a type of DDoS. The attacker creates a random source address for each packet, SYN flag is set in each packet is a request to open a new connection to the server from the spoofed IP address. Victim responds to spoofed IP address, then waits for confirmation that never arrives, waits for few more minutes, then victim's connection table fills up waiting for the replies. Once the table fills up, all new connections are ignored, legitimate users are ignored as well, and cannot access the server.

5. **DNS spoofing** works by forcing a DNS client to generate requests to a DNS server and spoof the response that is coming back from the server. One of the ways that this can be achieved is by recursive querying to the DNS server. Lets say that a client sends a request to the server to resolve a name to address. When the DNS server receives the request, it sends out proper queries to all necessary servers to find out the original address associated with it. Now if an intruder who can predict the response from the server and can spoof the response to send it back to the client. Such a response should be able to satisfy the client's request and also reach the client machine before the actual response from the server reaches it. Then the intruder succeeds in making the client believe that they are actually browsing on the original website but in reality they are being redirected to the intruder's site.

6. **Ping of Death**, is a serious type of attack on a computer's operating system, where a malicious or a malformed ping is sent to the target system. Any usual ping is only 56 bytes and can be 86 bytes if the Internet protocol header is also taken into consideration. The largest packet size that a computer could handle is 65,535 bytes which corresponds to a IPv4 packets. Most of the operating systems cannot handle ping packets of size larger than 65,535 bytes. This is one of the easiest methods to exploit a system. Sending a 65,536 byte ping packet is a violation according to RFC 791. Such packets could still be sent if it is fragmented, when the target system reassembles the packet, a buffer overflow occurs causing a system crash.

METHOD OF EXAMINATION

The different types of security issues briefed above has given us an idea of the possible attacks that can happen over a network or a system. The question in mind is that how do we find if our private cloud network is safe from these kind of attacks. The answer doesn't come easily, there are different steps to be followed religiously in order to detect, prevent and protect our private cloud from such attacks. The initial procedures that has to be done on any network or system is to find out the vulnerabilities in them. This is almost similar to reproducing the attack in our own network.

Techniques used in finding and recruiting vulnerable machines to reproduce an attack on the hosts are,

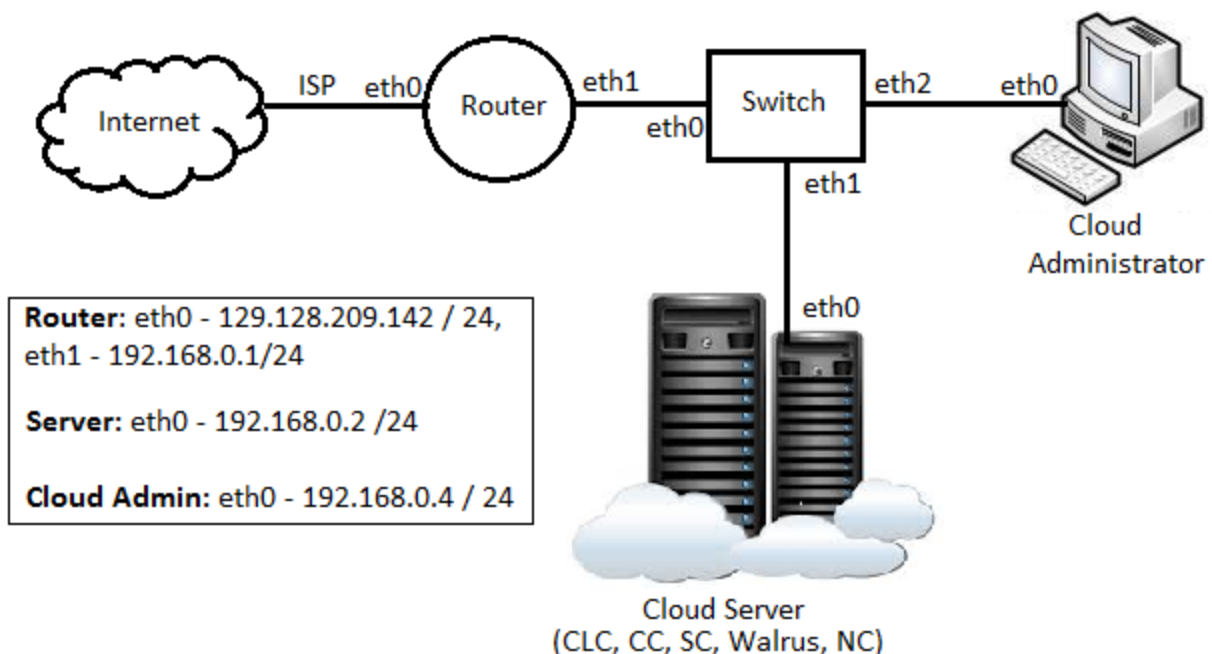
- **Random scanning:** In this type of technique, the malicious code is not only spread from the master machine, but also from machines that are already being affected by the malicious code sent from the master. The infected machine probes IP addresses randomly from the IP address space and checks their vulnerabilities. Whenever another vulnerable host or machine is found, it breaks into it and tries to infect the target with the same malicious code that is installed on itself.

- **Hit-list scanning:** in this type of scanning a list of potential vulnerable machines is collected first and only then the attackers start scanning the network. Once the list is in hand the attacker starts scanning the machines available in the list to find a vulnerable machine. When a vulnerable machine is identified, the malicious code is installed on it and then the list of potential vulnerable machines is split between the attacker and the first vulnerable machine in order to start scanning separately. In this way each vulnerable machine found will share the list and installs the malicious code faster.
- **Topological scanning:** This type of scanning finds new targets with the help of information available on the victim machine. Here the compromised hosts scan for URLs in the machine's disk and renders them in order to check their vulnerability and infect them. This technique is as fast as Hit-list scanning and is more accurate.
- **Local subnet scanning:** This type of scanning works behind a firewall and the malicious code is installed on one of the vulnerable system in the local network. In this way it scans only the local addresses that behind the firewall and can easily break into any vulnerable systems that are available in that local network, which would otherwise be protected by the firewall. This if combined with other scanning techniques can help identify the vulnerable hosts in an extremely high speed by using one of the compromised machines to start using other scanning techniques like random scanning to probe outside the local network and scan other machines available outside the list of local addresses.

REPRODUCING A NETWORK ATTACK ON CLOUD

Let us consider reproducing a **DoS or DDoS** attack on a cloud network,

Experimental setup used for all testing is as shown the diagram below and verify Appendix for additional details about other softwares and hardwares used.



- The very first step is **Examining the network:**
 - To examine a network and a victim server and corresponding services provided by it in the cloud, a random scanning technique called **nmap** can be used. Nmap by itself is an attack to the network if used for port scanning which helps in identifying the open ports and services provided by them.

- I have setup Blackbuntu on Virtualbox installed on a machine available in the private cloud network. Blackbuntu comes with all testing tools pre-installed on it making it easier to generate an attack of any kind.
- Since this system is present inside the private cloud network, it does not go through the firewall, which makes even more easier to perform an attack and all these attacks will go unnoticed by the firewall.
- The command used is `nmap <ip address>`, in my case it is the ip address of the cloud front end server which is - 192.168.0.2 / 24
- Preprocessor port scan which is built in snort is used to detect port scan attacks. It uses generator ID 122 to generate the alert for TCP Portscan.
- Snort immediately detects portscan and starts generating alerts mentioning the ip address of the attacker machine which trying to perform an nmap scan.
- **Alert Generated on Snort: 06/09-09:31:21.758499 [**] [122:1:0] (portscan) TCP Portscan [**] [Priority:3] {PROTO:255} 192.168.0.10 -> 192.168.0.2**
- Once the port scanning is completed, all the ports that are open will be displayed along with the services that it is currently providing. (Refer: Picture 1, Appendix C).

- Once vulnerable hosts are identified the tools used for performing the attack are installed on the particular hosts. This preparation stage is no longer tough, since the attack tools are already prepared programs that automatically finds the vulnerable systems, then break into such systems and then install the necessary programs for the attack. Once the code has the control over the particular system, it starts scanning nearby hosts or vulnerable computers and do the same on them. In this manner a large attack network is built which identifies the vulnerable hosts that are available in the network for compromise, making it very easy and faster to identify the victim systems. This kind of automated process creates a DoS and DDoS attack network that consists of master and slave machines.

- Next step is **Performing the DoS attack:** I am going to perform a SYN Flood attack using hping3, which is a DoS attack.
 - I am choosing port 8080/tcp, which is open and provides http-proxy service.
 - Perform the attack from the Blackbuntu virtual machine by using the command: `hping3 -S -p 8080 --flood --rand-source 192.168.0.2`
 - Above command means that, I am trying to send SYN packets to a specified port (8080). The use of `--flood` specifies that the packets are sent as fast as possible and does not wait for a reply or display the replies. `--rand-source` specifies to send those packets with some random source address. (Refer picture 2, Appendix C)
 - To detect such attacks, snort rule could be configured with count in time feature which will count all the TCP packets which contain the

SYN flag set which have been received from a particular IP within a specified amount of time.

- When snort finds out that too many SYN packets are received within a stipulated time interval, it is classified as a ping flood attack.
- **Alert generated by snort on the cloud front end machine will be, 06/10-17:03:19.728174 [**] [1:12121:0] DoS SYN flood attack detected! [**] [Priority:0]{TCP}192.168.0.10:60701 ->192.168.0.2:8080**

Problems Caused by DoS attack on the cloud server:

- **DoS** and **DDoS** have disastrous results, due to the characteristics of such attacks: both attacks are distributed in the sense that the attack is performed with the help of various bots and machines that are compromised to the malicious code.
- The DoS attacks performed denied access to the users who are trying to access cloud administrator's web page that is running on the victim machine (cloud front end server).
- The most important aspect of any network that an attacker would be interested in would be Available bandwidth. Since the network was flooded with useless packets, even a legitimate ICMP packet was rejected service.
- The next important aspect that an attacker is interested in consuming would be the CPU power. The attacker would aim to consume the entire memory by running several useless processes and thereby occupy the entire process tables. This eventually lead to a breakdown on the victim (i.e., the cloud front

end server), since the victim will not be able to execute any other process in turn affecting the cloud users who are trying to access the other services provided by the victim.

- Then finally it occupied all the services offered by the victim so that nobody else can access any of it. This was achieved by leaving TCP connections half open and by occupying the victim's data structures thereby not allowing other users to establish a TCP connection with the victim.

Reproduce another attack on the cloud network called **BED (Bruteforce Exploit Detection)** attack, which is a Web fuzzing attack.

- BED is a plain text protocol fuzzer that usually checks very regular bugs in a software like buffer overflows, format string bugs, integer overflows and similar.
- BED is one of the simple and easiest tools available ready to do a penetration testing.
- Blackbuntu has BED pre-installed on it, So I had to just login to Blackbuntu Virtual machine and go to /pentest/fuzzer/bed/ to start the testing.
- Command used for testing BED is,

```
./bed.pl -s http -t 192.168.0.2 -p 8080
```


(Refer Picture 3, Appendix C)
- The above command will use (-s) http plugin to test (-p) port number 8080 of the (-t) target ip 192.168.0.2 (Cloud server).
- Now BED will begin to perform different overflow tests on the target server like format string, buffer overflow, unicode, misc, random number, etc.
- It uses several strings, which are sent to the target daemon and waits for an unnatural reaction.

- I configured Snort to detect Web fuzzing attacks and alert generated by

Snort for a BED attack will look like:

06/09-21:47:45.456569 [] [1:1104:6] WEB-MISC whisker space splice attack [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.0.10:65234 -> 192.168.0.2:8080**

06/09-21:47:47.482994 [] [1:1171:7] WEB-MISC whisker HEAD with large datagram [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.0.10:65236 -> 192.168.0.2:8080**

Problems caused by Brute force Exploit Detection (BED) attack are:

- Several tests that are performed on the target server will wait for unnatural reaction from the target.
- When the server is weak, series of such tests can crash the daemon.
- The BED test will notify which string caused the daemon to crash. This information can be used to create an exploit to further explore the target server and take advantage of its vulnerabilities.

PROBLEM ANALYSIS

- Handling network intrusions coming from outside the local network is quite easier in private cloud networks, since the server will not be expecting requests coming from IPs outside the local network range.
- The issue becomes highly complicated when the attack is originating from an insider of the private network.

Using Wireshark, I analysed the TCP SYN Flood (DoS) attack that was performed on the private cloud,

- Consider an insider who sends a TCP connection request, which looks like a legitimate call in normal eyes. But, the requests are forged TCP SYN Flood (DoS) attack in a larger perspective.
- The traffic was captured by wireshark and stored in logs for future analysis.
- Then the packet transfer traffic over a period of one second from the logs was analysed, where I noticed that in one millisecond over several hundred TCP SYN packets were flooding the network eating up the bandwidth of the network.
- Also, TCP SYN request packets were sent from a single IP address, with the same sequence number '0' and same Acknowledgement number '0'.

- A netstat on the server during the attack showed that there were clearly more connections that were in the SYN_RECEIVED state and when looked at various packets, it was noted that all these packets did not respond back with an ACK to any of the SYN-ACK replies sent from the server.
- Another important feature that was noticed using Wireshark was the Round trip time taken by each packet to reach its destination and return back, drastically increased up to approximately 10 microseconds and stayed constant until end of the attack, as compared to RTT close to zero during normal traffic.
- As the RTT increased the server was forced to retransmit same packets again congesting the network even more and making the situation worse.
- On the application side, when a legitimate user tried to login to the cloud it had its connection timed out, since the server was still waiting for ACK from all the other TCP SYN requests and rejected any new requests in the queue.
- Then I wanted to test what will happen if the same SYN Flood packets are sent from different IPs belonging to the local network. So TCP connection requests were generated from two different IPs simultaneously.
- This time around the server was compromised even in a less shorter time, because the snort could not understand the traffic flow.
- When packets were analysed after the attack, I noticed that in a second several TCP SYN packets were sent from both the IP addresses at an unimaginable speed.

Issues and other points to be considered:

- Initially the Snort was designed to catch DoS attacks that came from an outsider, which let the insider to easily perform an attack and compromise the server.
- Then when snort was configured to catch internal attacks it did catch certain attacks, but considered initial TCP SYN packets to be legitimate and allowed the traffic to reach the server. Only when the count of SYN packets went high did the snort identified it as potential attack and sent an alert, but by then the server was already overwhelmed with malicious traffic.
- Above is just a sample where one user tries to conduct an attack. Consider the situation where more than one user conducts the attack. It would consume lots of CPU memory and throughput to identify a live attack.
- A basic sniffer or network traffic analyser can analyse and detect intrusions only when they are prescribed to detect such intrusions. They are not intelligent enough to study patterns and analyse the pattern to identify it as an intrusion or an attack.
- This becomes an overhead to the network administrator who is taking care of such a big network. The Network admin has to do all the pattern analysis and then identify it as an intrusion. But, by this time the server is under vulnerability or has already been compromised to the attacker.

- Another valuable point to consider will be the size of the attack itself. In this case the attacker is an insider, who would probably have an idea of the entire network, which makes it easy for him to compromise the other users in the local network.
- When the number of user systems compromised to become botnets to the host attacker increases, the chances of the server being compromised increases. Since the server might think that the packets are coming from the internal network sent by the users of the cloud and will tend to consider them as legitimate requests.
- The internet traffic is so random that detecting an attack is so difficult just by looking at the traffic patterns. Instead there should be a system which is intelligent enough to study the network traffic, analyze the current conditions and compare them with previous records. Also correlate with other NIDS modules sitting on the several servers in the cloud to collect data from them. With this compiled results on hand, it has to identify a plan to respond to such conditions without affecting rest of the network.
- Unfortunately , Snort can only be used either as a packet logger or as event handler, but not both at the same time.

MITIGATION PLAN

What could be used to detect attack then?

- Analyzing the traffic and the nature of the packets is very important in detecting an attack. When a small number of packets were studied, the important bottleneck noted was the similarities between the attack packets and legitimate packets. They looked almost similar giving us clear indications that it is not easy to identify that it is an attack unless deeply analyzed.
- One difference that could be noted at packet level was that all the forged tcp ip packets were with the same byte size. This could be a differentiating factor. Another factor that could be taken into consideration is the pattern of the packets that comes from a source.
- If a set of packets exhibit a certain mischievous behavior it can be classified as an intrusion and compared with historical data to do further analysis.
- If one packet exhibits an interesting combination of protocol type, service requested and / or flags set - then such a packet can be classified as an intrusion and can be considered for further investigation.
- All these are possible only by human intervention and analysis or by an intelligent software that can apply some logic to identify the variation in pattern, classify them accordingly and do further investigation to make appropriate decision.
- Most of the solutions available in the market, do not have such intelligent software. Even if available only serves conventional networks but not a cloud network. In a cloud network it is not only necessary to identify attack at one server, it is also necessary to communicate to other servers before it falls prey to the attack.

Measures that could be taken:

- Though cloud network attacks are hard to detect and rectify there could be some precautions taken in advance to prevent such attacks. These **Preventive measures** might not prevent all the attacks, but to some extent, cloud network could be saved from several vulnerabilities.
- The preventive mechanisms try to eliminate the possibility of attacks altogether or to enable potential victims to endure the attack without denying services to legitimate clients.
- With regard to attack prevention, countermeasures can be taken on victims or on zombies. This means modification of the system configuration to eliminate the possibility of accepting an attack or participating unwillingly in an attack.
- Hosts should guard against illegitimate traffic from or toward the machine. By keeping protocols and software up-to-date, we can reduce the weaknesses of a computer.
- A regular scanning of the machine is also necessary in order to detect any "anomalous" behavior. Examples of system security mechanisms include monitoring access to the computer and applications, and installing security patches, firewall systems, virus scanners, and intrusion detection systems automatically.
- The **reactive mechanisms** try to detect the attack and respond to it immediately. Hence, they restrict the impact of the attack on the victim. Again, there is the danger of characterizing a legitimate connection as an attack. For that reason it is necessary for researchers to be very careful.

- **Signature-based** methods search for patterns (signatures) in observed network traffic that match known attack signatures from a database. The advantage of these methods is that they can easily and reliably detect known attacks, but they cannot recognize new attacks. Moreover, the signature database must always be kept up-to date in order to retain the reliability of the system.
- **Anomaly-based** methods compare the parameters of the observed network traffic with normal traffic. Hence it is possible for new attacks to be detected. However, in order to prevent a false alarm, the model of "normal traffic" must always be kept updated and the threshold of categorizing an anomaly must be properly adjusted.
- Finally, **hybrid systems** combine both these methods. These systems update their signature database with attacks detected by anomaly detection. Again the danger is great because an attacker can fool the system by characterizing normal traffic as an attack. In that case an Intrusion Detection System (IDS) becomes an attack tool. Thus IDS designers must be very careful because their research can boomerang.
- After detecting the attack, the reactive mechanisms respond to it. The relief of the impact of the attack is the primary concern.
- Some mechanisms react by limiting the accepted traffic rate. This means that legitimate traffic is also blocked. In this case the solution comes from traceback techniques that try to identify the attacker. If attackers are identified, despite their efforts to spoof their address, then it is easy to filter their traffic. Filtering is efficient only if attackers' detection is correct. In any other case filtering can become an attacker's tool.

ARCHITECTURE FOR PROPOSED NIDS MODEL

As already discussed, the best solution for detecting, preventing and reacting to an attack is deploying an Intrusion detection system for the network. Such a system should combine traditional network detection techniques with logic based detection techniques which have the capability of updating the knowledge database of NIDS continuously. Use of such state-of-the-art techniques is very important to prevent an attack and also to safeguard the network and its components during an attack.

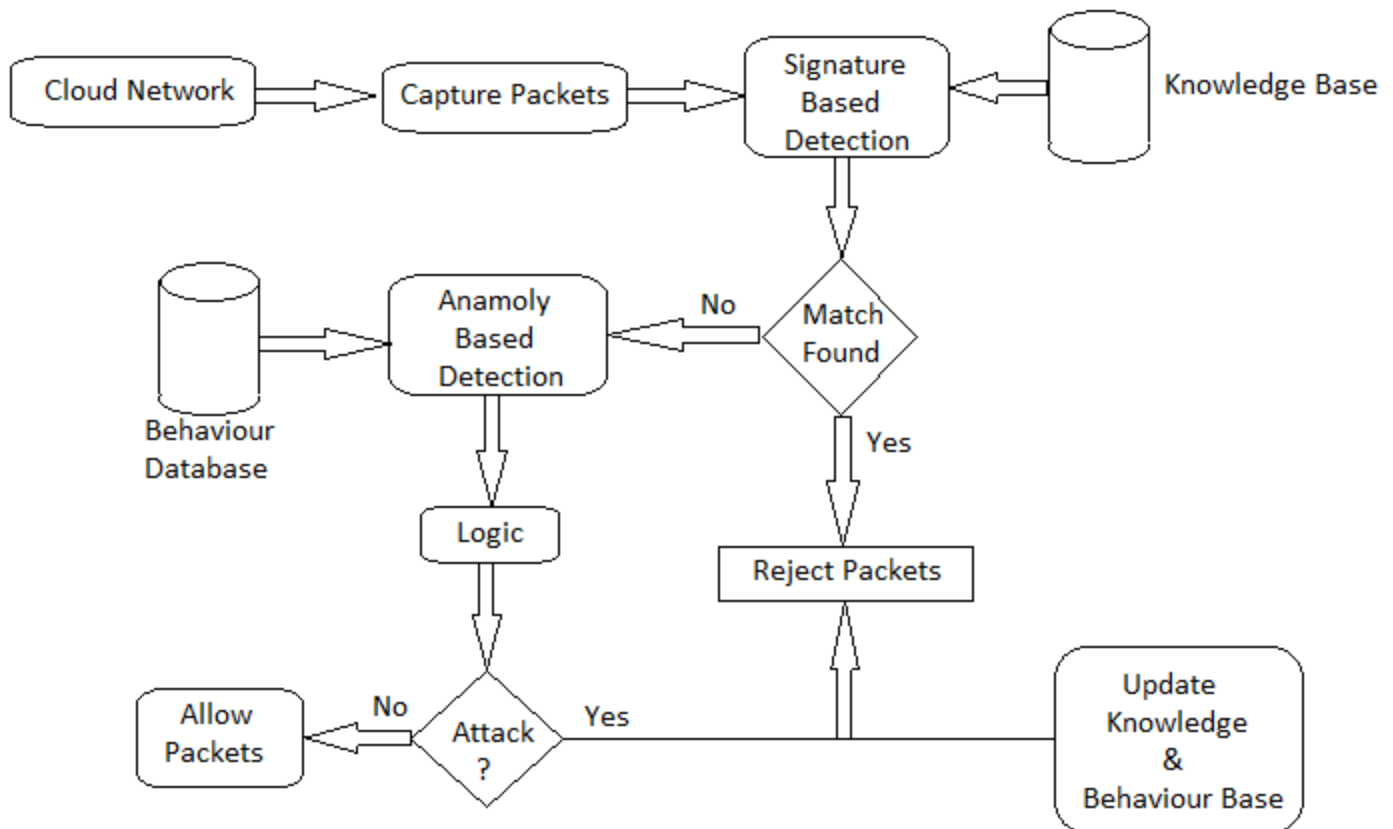
In this section I would like to propose an architecture which will be capable of detecting, preventing and reacting to an attack. This system will be capable of reducing false positives and false negatives which will save time, money and manpower. This NIDS should bring more accuracy and should detect attack faster to avoid a compromise and is also scalable and compatible to any kind of cloud network.

My proposal is purely based on Probability, mathematical abstraction of events that are non-deterministic in nature. The non-deterministic events may occur in a random manner or could also be single occurrences. Such occurrences when studied over a period of time will exhibit a pattern that describes the characteristics of those random events. With the help of the displayed pattern and its characteristics, the nature of occurrence of such events can be predicted.

The probability of occurrence of certain packet composition and the probability of such packets to be an intrusion are studied and the behaviour database is updated with such patterns along with its probability value. For example, lets say our cloud

server allows only tcp packets coming in for http service but is not configured to provide a ftp service. In such a case, when packet ‘A’ follows a pattern where it is a ‘TCP’ packet with ‘HTTP’ as service option and a certain flag, such packets have a high probability of being legitimate packets. If packet ‘B’ is a ‘TCP’ packet but with ‘FTP’ service and certain flag, the probability of such packets being an intrusion is high. When this probability logic is combined with filtering techniques based on selectivity and specificity the rate of detecting an attack can be improved by reducing errors.

Logical representation of working of the proposed NIDS



Each NIDS module will have the following components in them,

a straight Packet Sniffer like Snort to capture packets, knowledge database to store rules related to known network attacks, Behaviour Database which stores the network events capturing patterns of packets traversing the network which includes both normal and intrusion packets. The Behaviour database is initially trained to study the behaviour of packets and they are classified as a normal packet or an intrusion based on their pattern. Certain packet compositions do not resemble a normal legitimate packet, such packet compositions and its pattern of occurrence are noted down by the Behaviour database.

Procedure:

- Snort installed on the cloud server or any other admin systems will keep sniffing packets and send it to the Signature based detection logic.
- Signature based detection logic communicates with the Knowledge database and finds out if it has any packet pattern matching the current packet in examination and finds if it is an intrusion.
- If a match is found for a packet in the knowledge base then such packets are rejected.
- In case of a match not found the packets are sent to the next level, where the packets go through Anomaly based detection.
- The Anomaly detection logic studies the packet composition and sends only the specific format by removing the redundant information that the Behaviour base will not be interested in. When such a pattern matches any dataset in the base then it can be classified as an intrusion.

- If such a packet which is classified as intrusion follows the same pattern of occurrence as noted in the database that pattern will have more probability that the packet is an intrusion and the packets will be rejected.
- Once a packet and its pattern are identified as intrusion both the databases will be updated in order to avoid such packet in the future.
- The packets will be allowed into the network if they are found to be legitimate packets and their patterns will be noted down in the behaviour base.

Need of NIDS cluster:

- Placing these NIDS modules in the network plays an important part in protecting the cloud network from vulnerabilities.
- The NIDS module can be placed on the front end server, which helps in communications between cloud network and the outside world. This helps in protecting the system from external network intrusions but cannot identify the intrusions that happen inside the cloud network.
- If the NIDS module is placed on the back end server, which is responsible for internal network interactions and takes control of internal network intrusions. However, it will not detect external network intrusions.
- NIDS installed on the separate virtual machines will help the VM user to have a protection shield from intrusions. In that case each VM machine should get NIDS installed on it. Such a configuration is not plausible in real time where users can be added or removed from the cloud network anytime.
- The better solution would be to have a NIDS on the front end server and one for each backend server.

- Then combine all the NIDS on the back end servers to form a cluster which will communicate with each other. In such a case, if one of the back end servers encounter a new attack it will update the knowledge and behaviour databases in order to alert the other back end servers that there is a new intrusion pattern detected.
- This kind of clustering will also help in identifying distributed denial of service attacks, where the intruder tries to intrude the cloud network by attacking all the cloud back end servers in a random fashion considering that such attempts will not be combinedly monitored.

Advantages of the proposed NIDS architecture are:

- The accuracy of identifying an attack and the precision of it being a real attack will increase.
- Such high detection rates will result in lesser false negatives and false positives.
- Cost of training the NIDS module and cost of installation will not be anything higher than cost that could probably be spent on repairing the damage that is done to the cloud network after a serious attack.
- New rules can be added into the database dynamically without modifying any existing configurations and / or rules. This ensures that the system is scalable and is easily available.
- Existing NIDS systems can only identify either the known attacks or the unknown attacks, while this one will detect both more effectively.
- The NIDS is more compatible with all platforms and network configurations and has no dependencies carried along with it.

CONCLUSION

Any network intrusion like DoS, DDoS and other network level malicious attacks cannot be just identified by the traditional network intrusion detection techniques. In this project I studied the network issues and its characteristics and succeeded in inducing different network attacks and spot those attacks using Snort. Also, I have come with a proposal for detecting all kinds of known and unknown attacks using a new technique that combines both Signature based detection and Anomaly based detection techniques. The future of this project could be implementing this technique in real time to identify the positives and negatives and identify ways to improve the attack detection mechanism in cloud environment.

APPENDIX A

I decided to install all the Eucalyptus cloud components on a single machine as the test environment. For which I chose Eucalyptus Faststart 3.2.2 as the cloud installation software.

Hardware

- A machine with Intel core i7 processor, with 400 Gb Hard disk space and 4 Gb RAM as the cloud server
- A laptop with 4Gb RAM and 500 Gb Hard disk space as the client machine

Networking Modes

- There are four networking modes that can be chosen in Eucalyptus System, Managed mode, Managed No-Vlan mode and Static mode. Each one of them supports different specifications.
- I have used Static mode where I can configure Eucalyptus with a set of ip addresses that can be assigned to VM instances. Every time a VM instance is instantiated, Eucalyptus acts as a DHCP server and assigns the new instance with the net available ip address.
 - The router is configured to perform NAT overload, to allow the internal users to access the internet. The single ISP provided public ip address is mapped to number of internal private ip addresses allowing access to the internet.
 - The Cloud server and the administrator are provided with the next available ip addresses in the private ip range 192.168.0.0/24.

APPENDIX B

Configuring Eucalyptus:

To install Eucalyptus Cloud-in-a-Box:

1. Insert the installation CD and boot the target system from the CD. Once the boot screen loads, choose “Install Cloud-in-a-Box” and the installation starts.
2. Skip the media verification option and move on to the next step.
3. Choose the language, keyboard options as English and go to next
4. The next step will be to choose the networking options. As I mentioned earlier we go in for Static networking mode.
5. In the next step choose the timezone and click next
6. Provide a password for the root user and re-enter to confirm the password.
7. Next step will be to provide cloud configuration options, where the range of available public ip addresses should be provided. In this case, I provided 100 IP addresses from the range 192.168.0.0/24 (192.168.0.100-192.168.0.200). The new virtual instances created by Eucalyptus will receive IP addresses from within this specific range.
8. The next step would be to set the disk install options. Eucalyptus should be the primary application on the system by default, so I choose to configure Eucalyptus using the entire space available on the Hard disk.
9. Now the Eucalyptus installation starts and everything else is taken care by Eucalyptus itself i.e., Eucalyptus software will be first installed and a default Eucalyptus machine image will be built. Once all these steps are completed

the system will prompt to reboot.

10. Once the system reboots, a series of questions will be asked just for the first boot process and we will have to accept the licence, create a non-root login and turn on NTP.
11. Once the system is ready check to find out if the cloud is up and running by clicking on the Eucalyptus web admin console available on the desktop. Check if all the components are enabled, sometimes there are chances that some of the components are not registered properly or not enabled properly.
12. So to register and enable such components, follow these steps.

Starting Eucalyptus:

- Make sure each Eucalyptus component resolves to an IP address and this can be verified in the following file `/etc/hosts` on the server.
- The component in the Eucalyptus cloud should be started in the following order, first start the Cloud Controller (CLC) by using the commands,
`/usr/sbin/euca_conf --initialize`
`service eucalyptus-cloud start`
- Then start Walrus by using these commands,
`service eucalyptus-cloud start`
- Start the CC by,
`service eucalyptus-cc start`
- Start the SC,
`service eucalyptus-cloud start`
- Start the NC,
`service eucalyptus-nc start`
- The following steps can be performed to verify the startup,

- The CLC is listening on ports 8443 and 8773
- Walrus is listening on port 8773
- The SC is listening on port 8773
- If you are using the subscription only VMware Broker, it is listening on port 8773
- The CC is listening on port 8774
- The NCs are listening on port 8775
- Log files are being written to /var/log/eucalyptus/

Registering Eucalyptus Components:

- In case the components were not registered properly follow these steps to register them
- Register CLC by

```
/usr/sbin/euca_conf --register-cloud --partition eucalyptus --host [Secondary_CLC_IP]--component [CLC_Name]
```
- Register Walrus by, implementing this command on the CLC server,

```
/usr/sbin/euca_conf --register-walrus --partition walrus --host [walrus_IP_address] --component [walrus_name]
```
- Register CC by,

```
/usr/sbin/euca_conf --register-cluster --partition [partition_name] --host [CC_IP_address] --component [cc_name]
```
- Register SC by,

```
/usr/sbin/euca_conf --register-sc --partition [partition_name] --host [SC_IP_address] --component [SC_name]
```
- Register NC by,

```
/usr/sbin/euca_conf --register-nodes "[node0_IP_address] ...  
[nodeN_IP_address]"
```

Administration guide:

- Administrator can do a number of tasks related to system management, identity management, and resource management.
- All these tasks are described in the following document <http://www.eucalyptus.com/docs/3.2/ag/>
- To access the Eucalyptus cloud administrator's management web interface, go to <https://192.168.0.2:8443>

Reboot an Instance | Eucalyptus | Eucalyptus

https://192.168.0.2:8443/#start: | Google

EUCALYPTUS admin@eucalyptus

QUICK LINKS

- System Management
 - Start
 - Service Components
- Identity Management
 - Accounts
 - Groups
 - Users
 - Policies
 - Keys
 - Certificates
- Resource Management
 - Images
 - VmTypes
 - Usage Report

START GUIDE

Welcome to your Eucalyptus-Powered Cloud

Following are some quick guides to the things you can do with your cloud's Web dashboard.

MANAGE IDENTITIES AND YOUR PROFILE

- View or change your personal profile details
- Show all accounts under your management
- Show your account's groups
- Show your account's users
- Show the keys you have

MANAGE CLOUD SERVICES AND RESOURCES

- View and configure cloud service components
- Download and view images
- View and configure virtual machine types
- Generate cloud resource usage report

REGISTER YOUR CLOUD

Use the following information to register your cloud with third-party cloud management platform:

Cloud URL: <https://192.168.0.2:8443/register>

Cloud ID: dc8b62cb-78f6-4577-ab34-2f83ebe499a3

LOG | Eucalyptus 3.2.2

- The admin web interface is very useful where all the commands that can be

executed on a terminal window can be done using this visual web interface.

- The identity management lets you create new cloud users, add them to groups, assign keys and policies and view certificates available or add new certificates, create new accounts etc.
- You can also view the service components which are registered and manage them.

The screenshot shows the Eucalyptus web interface. The browser address bar displays `https://192.168.0.2:8443/#config`. The page title is "UCALYPTUS" and the user is logged in as "admin@eucalyptus". The left sidebar contains "QUICK LINKS" with categories: System Management (Start, Service Components, Identity Management), Resource Management (Images, VmTypes, Usage Report), and Identity Management (Accounts, Groups, Users, Policies, Keys, Certificates). The main content area displays a table titled "SERVICE COMPONENTS" with the following data:

Name	Partition	Type	Host	Port	Status
192.168.0.2	eucalyptus	cloud controller	192.168.0.2	8773	ENABLED
cc_01	CLUSTER01	cluster controller	192.168.0.2	8774	ENABLED
sc_01	CLUSTER01	storage controller	172.31.252.1	8773	ENABLED
walrus	walrus	walrus	192.168.0.2	8773	ENABLED

At the bottom of the page, there is a "LOG" button and the version "Eucalyptus 3.2.2".

- The resource management section of cloud admin's looks different from a user admin's console. Now click on VmTypes, it displays all the Virtual images that are available in the cloud, lets you add and configure new virtual images as shown below

The screenshot shows the Eucalyptus web interface. The browser address bar indicates the URL is <https://192.168.0.2:8443/#vmtype>. The page title is "EUCALYPTUS" and the user is logged in as "admin@eucalyptus". The main content area is titled "VIRTUAL MACHINE TYPES" and contains a table with the following data:

Name	CPUs	Memory (MB)	Disk (GB)
m1.small	1	512	5
c1.medium	2	512	10
m1.large	2	1024	15
m1.xlarge	2	2048	20
c1.xlarge	4	4096	20

The left sidebar shows a "QUICK LINKS" menu with categories: System Management (Start, Service Components), Identity Management (Accounts, Groups, Users, Policies, Keys, Certificates), and Resource Management (Images, VmTypes, Usage Report). The "VmTypes" link is highlighted. At the bottom right, there is a "LOG" button and the version "Eucalyptus 3.2.2".

- Usage report - It provides a detailed report on the images created, used and terminated, memory used, and report on all the resources that have been used for a certain period of time.

User guide:

- Let us configure the Admin system as one of the users and launch an instance of the default CentOS image from there to test the cloud environment.
- On the Admin system connected to eth2 of the switch, open a browser
- First step is to sign up for an user account in the Eucalyptus cloud, so point your browser to <http://192.168.0.2:8443/> (the cloud server's IP address)

- Click on the Apply link to access the application for creating an User account, fill out the form and submit it. An approval email will be sent to the inbox, click on the confirmation link and this creates an User account with the private cloud.
- Login into the account and download the credentials to a secured place on the system, the download will contain the zip-file with your public/private key pair, a bash script, and several other required files.
- The user's management web interface will be missing the Vmtypes, usage reports links and will not be allowed to manage cloud resources.

The screenshot shows a web browser window with the URL `https://192.168.0.2:8443/#`. The page header includes the Eucalyptus logo and the user `admin@helpgod`. A search bar is visible on the right. The main content area is titled "Welcome to your Eucalyptus-Powered Cloud" and includes a "START GUIDE" link. Below this, there is a section for "MANAGE IDENTITIES AND YOUR PROFILE" with links for "View or change your personal profile details", "Show your account's groups", "Show the keys you have", "Show all accounts under your management", and "Show your account's users". Further down, there are sections for "MANAGE CLOUD SERVICES AND RESOURCES", "REGISTER YOUR CLOUD", and "DOWNLOAD TOOLS FROM EUCALYPTUS". The "DOWNLOAD TOOLS FROM EUCALYPTUS" section lists "Eucalyptus-compatible Tools" and a link to `euca2ools`. The footer of the page contains a "LOG" button and the version number "Eucalyptus 3.2.2".

Reboot an Instance Eucalyptus Eucalyptus

https://192.168.0.2:8443/#account:

EUCALYPTUS admin@helpgod

QUICK LINKS

- System Management
 - Start
- Identity Management
 - Accounts**
 - Groups
 - Users
 - Policies
 - Keys
 - Certificates
- Resource Management
 - Images

ACCOUNTS

New users New groups Add policy Approve Reject

ID	Name	Registration status
169657970072	helpgod	CONFIRMED

Use 'Ctrl' or 'Shift' for multiple selections. 1-1 of 1

LOG Eucalyptus 3.2.2

Reboot an Instance Eucalyptus Eucalyptus

https://192.168.0.2:8443/#user:accountid=169657970072

EUCALYPTUS admin@helpgod

QUICK LINKS

- System Management
 - Start
- Identity Management
 - Accounts
 - Groups
 - Users**
 - Policies
 - Keys
 - Certificates
- Resource Management
 - Images

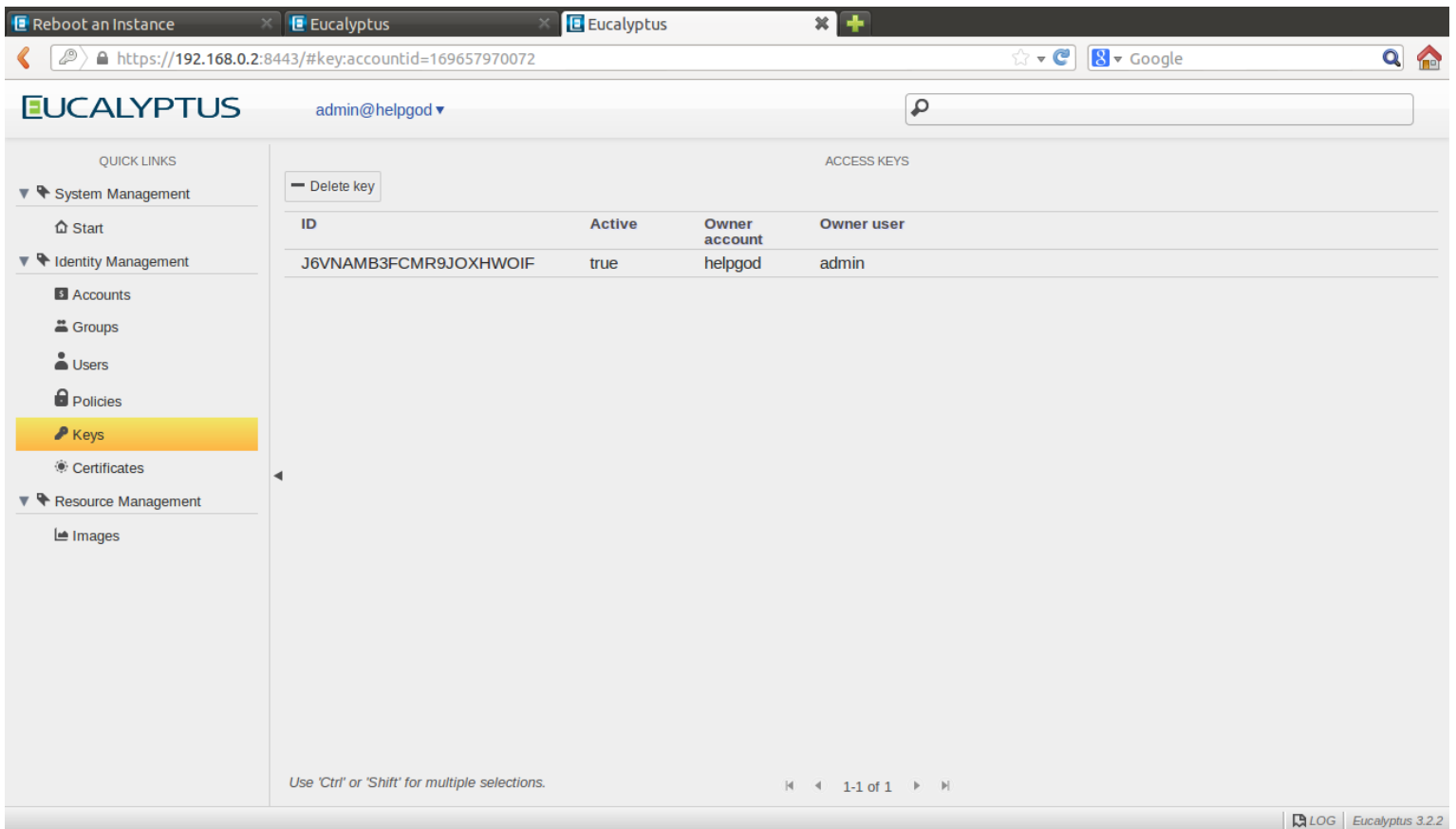
USERS

Delete users Add to groups Remove from groups Add policy Add key Add certificate Approve Reject

ID	Name	Path	Owner account	Enabled	Registration status
T4NVFHDWOOHWJK9KAFKSM	admin	/	helpgod	true	CONFIRMED

Use 'Ctrl' or 'Shift' for multiple selections. 1-1 of 1

LOG Eucalyptus 3.2.2



- Now Unzip your credentials zip file to a directory of your choice. In the following example we download the credentials zip file to ~/.euca, then change access permissions, as shown:

```
mkdir ~/.euca
```

```
cd ~/.euca
```

```
unzip <filepath>/euca2-<user>-x509.zip
```

```
chmod 0700 ~/.euca
```

```
chmod 0600 *
```

- Euca2ools can be installed by the command, yum install euca2ools or apt-get install euca2ools and once it is installed, we have to make sure to source the eucarc by using the command “**source eucarc**” which is very important

because it sets the environment variables necessary to allow the interaction between the euca2ools and our Eucalyptus cloud.

- Eucalyptus Faststart has CentOS image by default, now lets run an instance of this image to start with.
- To find the default image that's there on the server already, type in the command:

euca-describe-images

the result would be something like this:

```
IMAGE eki-D313397A admin/vmlinuz-2.6.28-11-generic.manifest.xml  
508678674223 available public i386 kernel instance-store
```

IMAGE emi-72613A2E

```
admin/euca-centos-5.8-2012.05.14-x86_64.manifest.xml 508678674223  
available public i386 machine eki-D313397A eri-F9A83F12 instance-store  
IMAGE eri-F9A83F12 admin/initrd.img-2.6.28-11-generic.manifest.xml  
508678674223 available public i386 ramdisk instance-store
```

- **IMAGE emi-72613A2E** is the image ID. Now Create a key pair using the **euca-add-keypair** command. This command stores the public half of the key pair and has it ready for your Eucalyptus cloud instances, and will output the private half of the key pair. Save the output to the local machine in order to use it while trying to access an instance. It can be saved by using following commands,

euca-add-keypair euca-demo > euca-demo.private

- Then the permissions of the private key pair has to be changed in order to allow access only to you,

chmod 0600 euca-demo.private

- Now authorize security groups to allow network access to SSH and VNC by using the following commands,

```
euca-authorize -P tcp -p 22 -s 0.0.0.0/0 default
```

```
euca-authorize -P tcp -p 5900-5910 -s 0.0.0.0/0 default
```

- Now run the instance using the command,

```
euca-run-instances -k euca-demo emi-72613A2E
```

the output would be something like this,

```
RESERVATION  r-CCE33FC0  449455269925  default INSTANCE
i-68A24092  emi-72613A2E  0.0.0.0  0.0.0.0  pending  euca-demo
0
m1.small  2012-05-17T10:36:46.232Z  PARTI00  eki-D313397A
eri-F9A83F12
monitoring-disabled  0.0.0.0  0.0.0.0  instance-store
```

- After sometime if you type in the command,

```
euca-describe-instances i-68A24092
```

the result will be,

```
RESERVATION  r-CCE33FC0  449455269925  default INSTANCE
i-68A24092  emi-72613A2E  192.168.0.110  10.93.7.76  running
euca-demo  0
m1.small  2012-05-17T10:36:46.232Z  PARTI00  eki-D313397A
eri-F9A83F12
monitoring-disabled  192.168.0.110  10.93.7.76  instance-store
```

- You could note that the instance is now in the running state and it is given both public and private IP addresses which we can use to access the instance.

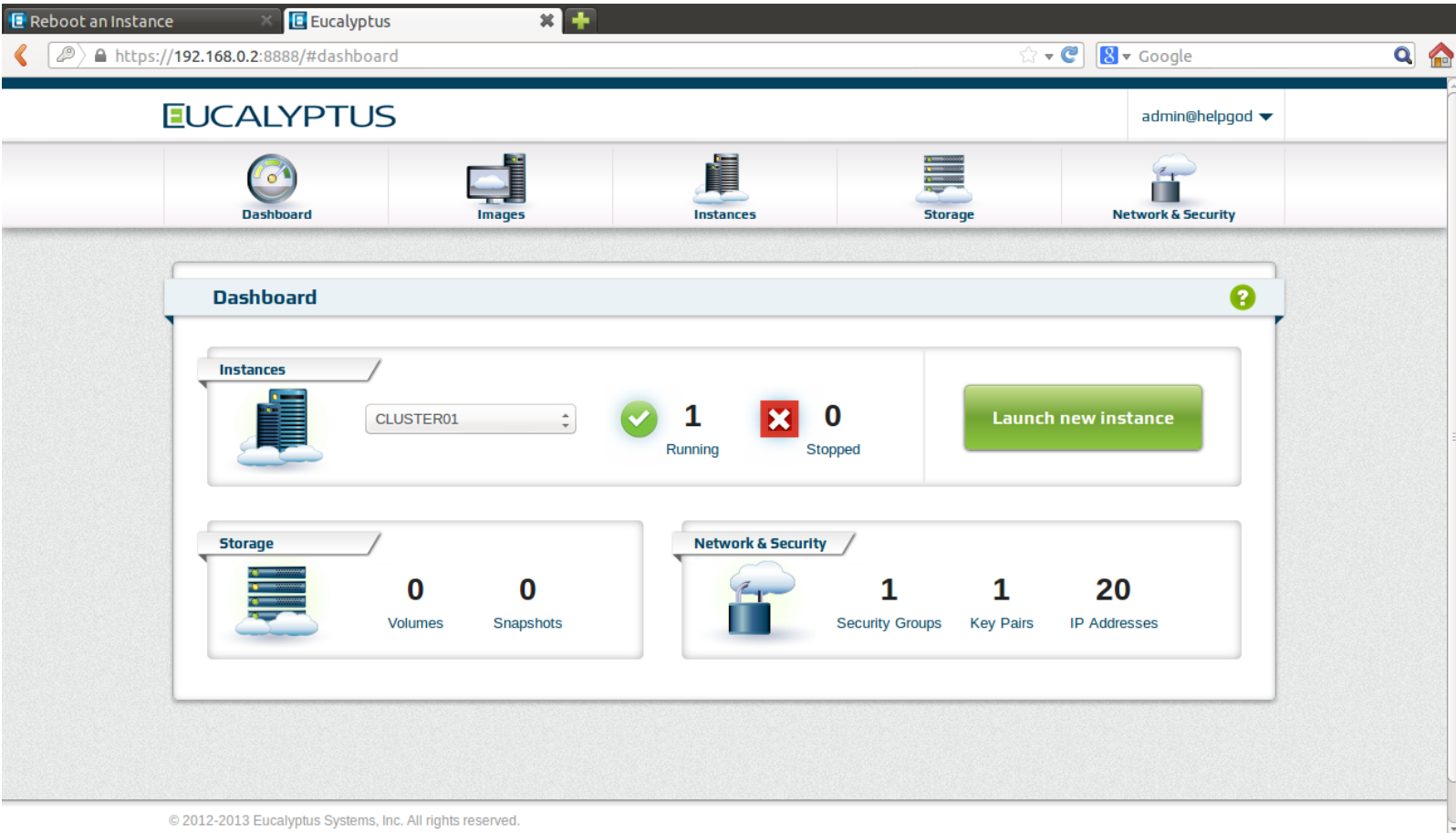
- To login to an instance open a terminal and type the command,
ssh -i euca-demo.private 192.168.0.110
returns the following result,
Warning: Permanently added '192.168.0.110' (RSA) to the list of known hosts.
Last login: Thu Mar 20 05:29:32 2013 from eucahost-9-91.eucalyptus
-bash-3.2#

Now we are logged into the first virtual instance in the private cloud.

Additional documentation for Eucalyptus Users and Administrators are available online in the following link : <http://www.eucalyptus.com/docs>

All the above said instructions can also be done on a web interface, as shown below:

- Log onto <https://192.168.0.2:8888/>, user web console using the user credential



- The above image shows the number instances running, security groups available and number of ip addresses allocated to that particular user group

Reboot an Instance | Eucalyptus | <https://192.168.0.2:8888/#sgroup> | admin@helpgod

Dashboard | Images | Instances | Storage | Network & Security

Manage security groups

Search security groups Refresh

Create new security group | More actions

1 security groups found. Showing: 10 | 25 | 50 | 100

<input type="checkbox"/>	NAME	DESCRIPTION
<input type="checkbox"/>	default	default group

Navigation: <<< < 1 > >>>

Reboot an Instance | Eucalyptus | <https://192.168.0.2:8888/#keypair> | admin@helpgod

Dashboard | Images | Instances | Storage | Network & Security

Manage key pairs

Search key pairs Refresh

Create new key pair | Import key pair | More actions

1 keys found. Showing: 10 | 25 | 50 | 100

<input type="checkbox"/>	NAME	KEY FINGERPRINT
<input type="checkbox"/>	helpgod	e9:e7:9f:b5:fa:12:35:05:c4:2d:31:67:b0:0a:fc:c0:fe:f6:d8:77

Navigation: <<< < 1 > >>>

Reboot an Instance | Eucalyptus

https://192.168.0.2:8888/#eip

Dashboard | Images | Instances | Storage | Network & Security

Manage IP addresses

Filter by: All addresses | Search IP addresses | Refresh

Allocate IP address | More actions

20 IP addresses found. Showing: 10 | 25 | 50 | 100

<input type="checkbox"/>	PUBLIC IP ADDRESS	ASSIGNED TO INSTANCE
<input type="checkbox"/>	192.168.0.110	
<input type="checkbox"/>	192.168.0.111	
<input type="checkbox"/>	192.168.0.112	
<input type="checkbox"/>	192.168.0.113	
<input type="checkbox"/>	192.168.0.114	
<input type="checkbox"/>	192.168.0.115	
<input type="checkbox"/>	192.168.0.116	
<input type="checkbox"/>	192.168.0.117	
<input type="checkbox"/>	192.168.0.118	
<input type="checkbox"/>	192.168.0.119	

© 2012-2013 Eucalyptus Systems, Inc. All rights reserved.

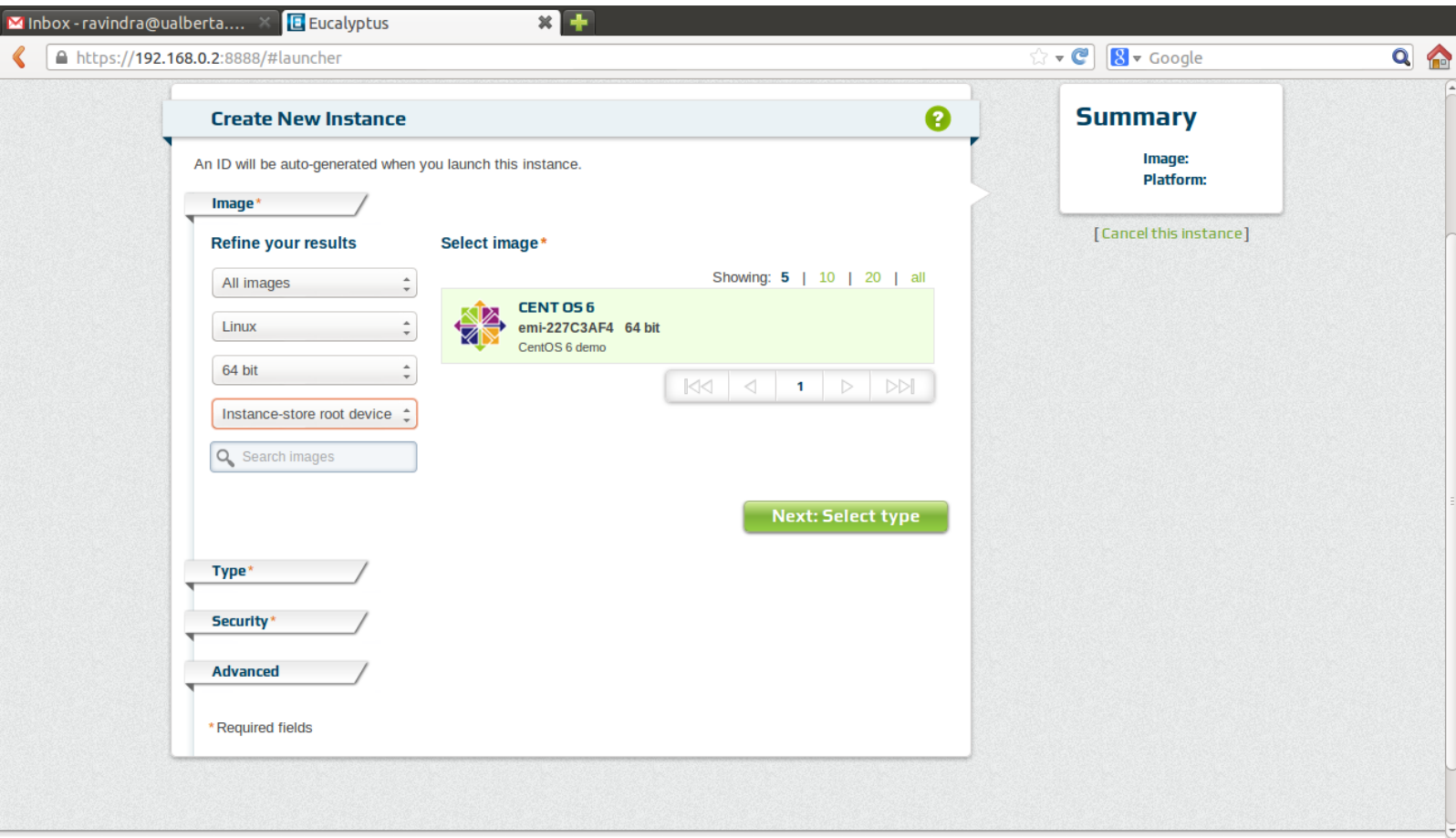
- The Images section will list the images that are available in the cloud that can be utilized, in our cloud we have only one CentOS image available as shown below,

The screenshot shows the Eucalyptus web console interface. At the top, there is a navigation bar with the Eucalyptus logo and a user profile 'admin@helpgod'. Below this is a menu with icons for Dashboard, Images, Instances, Storage, and Network & Security. The main content area is titled 'Manage Images' and includes a filter section with dropdowns for 'All images', 'All platforms', '32 and 64 bit', and 'All root devices', along with a search box and a 'Refresh' button. Below the filters, it indicates '1 images found. Showing: 10 | 25 | 50 | 100'. A table lists the image details:

NAME	IMAGE ID	ARCHITECTURE	DESCRIPTION	ROOT DEVICE	
ks-centos6-201304040153	emi-227C3AF4	x86_64	CentOS 6 demo	instance-store	Launch Instance

At the bottom of the table, there are pagination controls showing '1' of 1 items.

- Now click on Launch instance in the above page or on the main page and create new instance screen appears



- Choose the image that you want to create an instance with and click on Next:Select type,

Inbox - ravindra@ualberta... x Eucalyptus x +

https://192.168.0.2:8888/#launcher


Create New Instance

An ID will be auto-generated when you launch this instance.

Image*

Type*

Select instance size:



m1.small c1.medium m1.large m1.xlarge c1.xlarge

m1.small defaults: 1 CPUs, 512 memory (MB), 5 disk (GB,root device)

Select launch options:

Number of Instances

Availability zone

Next: Select security

Security*

Advanced

Summary

Image:
Platform:

Type:
m1.small

Instances:
1

Zone:
No Preference

[Cancel this instance]

- Choose any virtual machine type based on the memory and storage requirements, here am choosing m1.large and click on Next

The screenshot displays the Eucalyptus web interface for creating a new instance. The main form, titled "Create New Instance", includes the following elements:

- A note: "An ID will be auto-generated when you launch this instance."
- Form fields: "Image*", "Type*", and "Security*" (all marked as required).
- "Select security options:" section with:
 - "Key name" dropdown set to "helpgod" with a link "Or: Create new key pair".
 - "Security group" dropdown set to "default" with a link "Or: Create new security group".
- "Security group default incoming traffic rules" section:
 - Rule: tcp 22 0.0.0.0/0
 - Rule: tcp 5900-5910 0.0.0.0/0
- "Launch instance(s)" button with a link "Or: Select advanced options".
- "Advanced" expandable section.
- A footer note: "* Required fields".

The "Summary" panel on the right provides a preview of the instance configuration:

- Image: [blank]
- Platform: [blank]
- Message: "Image: Required fields are missing!"
- Type: m1.large
- Instances: 1
- Zone: CLUSTER01
- Key pair: helpgod
- Security group: default
- Link: "[Cancel this Instance]"

- In Select Security group page, you can either create a new key name and security group or use the already existing ones. I am using the existing keyname and security group and click on Launch instance.
- Now the instance will be in pending state and will take a couple of minutes to go the running state

Inbox - ravindra@ualberta... Eucalyptus

https://192.168.0.2:8888/#instance

Google

Dashboard Images Instances Storage Network & Security

Manage instances

Filter by All instances All root devices Search instances Refresh

Launch new instance More actions 1 instances found. Showing: 10 | 25 | 50 | 100

INSTANCE ID	STATUS	IMAGE ID	AVAILABILITY ZONE	PUBLIC ADDRESS	PRIVATE ADDRESS	KEY NAME	SECURITY GROUP	LAUNCH TIME
<input type="checkbox"/> i-575D440D		emi-227C3AF4	CLUSTER01	0.0.0.0	0.0.0.0	helpgod	default	01:05:10 AM May 2nd 2013

Running Pending Stopping Stopped Shutting-down Terminated

© 2012-2013 Eucalyptus Systems, Inc. All rights reserved.

Inbox - ravindra@ualberta... Eucalyptus

https://192.168.0.2:8888/#instance

Google

Dashboard Images Instances Storage Network & Security

Manage instances

Filter by All instances All root devices Search instances Refresh

Launch new instance More actions 1 instances found. Showing: 10 | 25 | 50 | 100

INSTANCE ID	STATUS	IMAGE ID	AVAILABILITY ZONE	PUBLIC ADDRESS	PRIVATE ADDRESS	KEY NAME	SECURITY GROUP	LAUNCH TIME
<input checked="" type="checkbox"/> i-575D440D		emi-227C3AF4	CLUSTER01	192.168.0.120	172.31.255.140	helpgod	default	01:05:10 AM May 2nd 2013

Running Pending Stopping Stopped Shutting-down Terminated

© 2012-2013 Eucalyptus Systems, Inc. All rights reserved.

- Now if you go to the dashboard, you can see an instance running

The screenshot shows the Eucalyptus dashboard in a web browser. The browser's address bar displays the URL `https://192.168.0.2:8888/#dashboard`. The dashboard header includes the Eucalyptus logo and a user profile for `admin@helpgod`. A navigation bar contains icons for Dashboard, Images, Instances, Storage, and Network & Security. The main content area features a 'Dashboard' section with a 'Instances' widget. This widget shows a dropdown menu set to 'CLUSTER01', a green checkmark icon with the number '1' and the label 'Running', a red 'X' icon with the number '0' and the label 'Stopped', and a green 'Launch new instance' button. Below this, there are two more widgets: 'Storage' showing '0 Volumes' and '0 Snapshots', and 'Network & Security' showing '1 Security Groups', '1 Key Pairs', and '20 IP Addresses'. At the bottom of the page, a copyright notice reads: '© 2012-2013 Eucalyptus Systems, Inc. All rights reserved.'

- Now you could open a terminal and do `ssh` to access the instance using the credentials that were downloaded.
- To terminate an instance go to instances page and click on more actions

Reboot an Instance | Eucalyptus

https://192.168.0.2:8888/#instance

UCALYPTUS | admin@helpgod

Dashboard | Images | Instances | Storage | Network & Security

Manage instances

Filter by: Running instances | All root devices | Search instances | Refresh

Launch new instance | More actions

1 instances found. Showing: 10 | 25 | 50 | 100

INSTANCE ID	STATUS	PUBLIC ADDRESS	PRIVATE ADDRESS	KEY NAME	SECURITY GROUP	LAUNCH TIME
<input checked="" type="checkbox"/> i-28C6407E	Running	192.168.0.130	172.31.254.122	helpgod	default	12:04:16 AM May 3rd 2013

Instance details:

INSTANCE TYPE	OS	KERNEL ID	RESERVATION ID	ACCOUNT ID	IMAGE MANIFEST
m1.large	linux	eki-0FDB3	r-AA324179	169657970072	centos6/ks-centos6-201304040153.manifest.xml

Running |
 Pending |
 Stopping |
 Terminated

More actions menu:

- Connect
- Stop
- Start
- Reboot
- Launch more like this
- Terminate
- Get console output
- Attach volume
- Detach volume
- Associate IP address
- Disassociate IP address

- Click on Terminate and a pop up will appear click on yes

Reboot an Instance | Eucalyptus | https://192.168.0.2:8888/#instance

Dashboard | Images | Instances | Storage | Network & Security

Manage instances

Filter by: Running instances | All root devices | Search instances | Refresh

Launch new instance | More actions

1 instances found. Showing: 10 | 25 | 50 | 100

INSTANCE ID	STATUS	PUBLIC ADDRESS	PRIVATE ADDRESS	KEY NAME	SECURITY GROUP	LAUNCH TIME
<input checked="" type="checkbox"/> i-28C6407E	Running	192.168.0.130	172.31.254.122	helpgod	default	12:04:16 AM May 3rd 2013

Instance: m1.large | OS: linux | KERNEL ID: eki-0FDB3

More actions menu: Connect, Stop, Start, Reboot, Launch more like this, Terminate, Get console output, Attach volume, Detach volume, Associate IP address, Disassociate IP address

Running | Pending | Stopping | Terminated

© 2012-2013 Eucalyptus Systems, Inc. All rights reserved.

Reboot an Instance | Eucalyptus | https://192.168.0.2:8888/#instance

Dashboard | Images | Instances | Storage | Network & Security

Manage instances

Filter by: Running instances | All root devices | Search instances | Refresh

Launch new instance | More actions

1 instances found. Showing: 10 | 25 | 50 | 100

INSTANCE ID	STATUS	PUBLIC ADDRESS	PRIVATE ADDRESS	KEY NAME	SECURITY GROUP	LAUNCH TIME
<input checked="" type="checkbox"/> i-28C6407E	Running	192.168.0.130	172.31.254.122	helpgod	default	12:04:16 AM May 3rd 2013

Instance: m1.large | OS: linux | KERNEL ID: eki-0FDB3

More actions menu: Connect, Stop, Start, Reboot, Launch more like this, Terminate, Get console output, Attach volume, Detach volume, Associate IP address, Disassociate IP address

Running | Pending | Stopping | Stopped | Shutting-down | Terminated

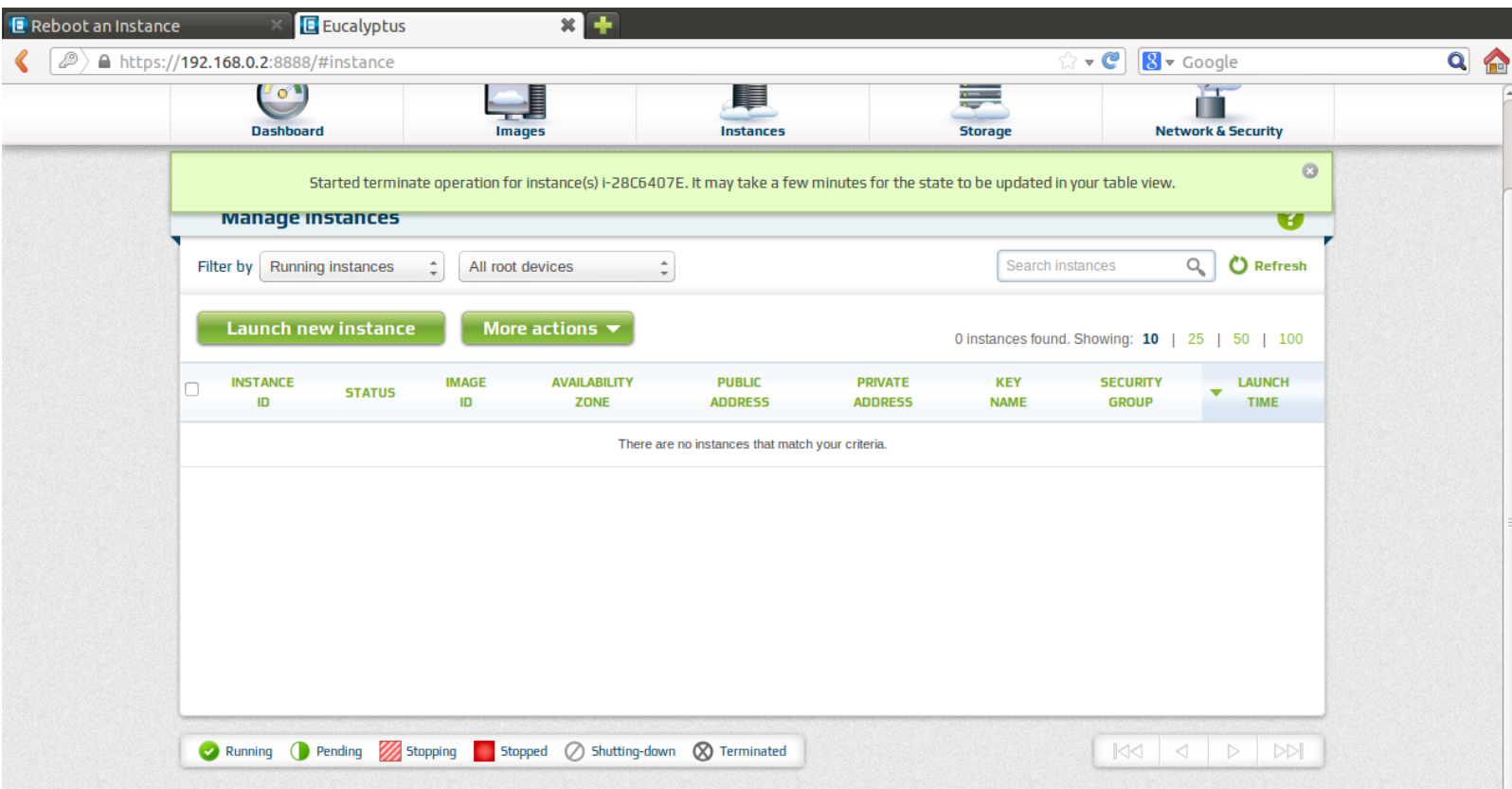
Terminate instance

Are you sure you want to terminate the following instances?

INSTANCE
i-28C6407E

Yes, terminate | Cancel

© 2012-2013 Eucalyptus Systems, Inc. All rights reserved.



© 2012-2013 Eucalyptus Systems, Inc. All rights reserved.

Snort Installation

- Install the following dependencies on the Eucalyptus frontend server with CentOS on it,
 - **gcc version (4.4.6 including libraries)**
 - **flex (2.5.35)**
 - **bison (2.4.1)**
 - **zlib (1.2.3 including zlib-devel)**

- **libpcap (1.0.0 including libpcap-devel)**
- **pcre (7.84 including pcre-devel)**
- **libdnet(1.11 or 1.12 including libdnet-devel)**
- **tcpdump (4.1.0)**
- Obtain SNORT (version 2.9.4.x), DAQ (version 2.0.0), and snort rules from www.snort.org and download them to your CentOS 6.3/6.4 box.
- To compile, execute and install snort related packages, root user privileges are required.

```
cd /usr/local/src
```

```
tar -zxvf <path to>daq-2.0.0.tar.gz
```

```
tar -zxvf <path to>snort-2.9.4.x.tar.gz
```

- Now configure, compile and install DAQ by using the following commands,

```
cd /usr/local/src/daq-2.0.0
```

```
./configure
```

```
make
```

```
make install
```

```
cd /usr/local/lib
```

```
ldconfig -v /usr/local/lib
```

- Now compile snort using the following commands,

```
cd /usr/local/src/snort-2.9.4.x
```

```
./configure --enable-sourcefire
```

```
make
```

```
make install
```

```
cd /usr/local/lib
```

ldconfig -v /usr/local/lib

- Register as a user in www.snort.org and download the rules
- Then issue these commands to place the snort configuration files and the files related to snort rules in the /etc/snort directory,

cd /etc

mkdir -p snort

cd snort

tar -zvxf <path to>snortrules-snapshot-<nnnn>.tar.gz

cp ./etc/* .

touch /etc/snort/rules/white_list.rules /etc/snort/rules/black_list.rules

- Place the snort.conf and threshold.conf files in /etc/snort directory and rules files in the /etc/snort/rules.
- Now add a user and group called snort in the system

groupadd -g 40000 snort

useradd snort -u 40000 -d /var/log/snort -s /sbin/nologin -c

SNORT_IDS -g snort

cd /etc/snort

chown -R snort:snort *

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

- Locate and modify the following lines in the snort.conf file,

var RULE_PATH /etc/snort/rules

ipvar HOME_NET 192.168.1.0/24

ipvar EXTERNAL_NET !\$HOME_NET

var SO_RULE_PATH /etc/snort/so_rules

var PREPROC_RULE_PATH /etc/snort/preproc_rules

```
var WHITE_LIST_PATH /etc/snort/rules
```

```
var BLACK_LIST_PATH /etc/snort/rules
```

- Now follow these commands to take ownership of directories and change file permissions that are related to snort and DAQ,

```
cd /usr/local/src
```

```
chown -R snort:snort daq-2.0.0
```

```
chmod -R 700 daq-2.0.0
```

```
chown -R snort:snort snort-2.9.4.x
```

```
chmod -R 700 snort-2.9.4.x
```

```
chown -R snort:snort snort_dynamicsrc
```

```
chmod -R 700 snort_dynamicsrc
```

- Download the initialization script from www.snort.org/docs website and place it in the /etc/init.d directory.
- Create a symbolic link (symlink) for snort, using command

```
cd /usr/sbin
```

```
ln -s /usr/local/bin/snort snort
```

- Create another file in the directory /etc/sysconfig called 'snort' and add these lines to the file and provide the user and group snort with permissions '700':

```
#### General Configuration
```

```
INTERFACE=eth0
```

```
CONF=/etc/snort/snort.conf
```

```
USER=snort
```

```
GROUP=snort
```

```
PASS_FIRST=0
```

```
#### Logging & Alerting
```

LOGDIR=/var/log/snort

ALERTMODE=fast

DUMP_APP=1

BINARY_LOG=1

NO_PACKET_LOG=0

PRINT_INTERFACE=0

- Issue the following commands to create a directory at /var/log/snort, set permissions to '700' and assign them to user and group 'snort'

cd /var/log

mkdir snort

chmod 700 snort

chown -R snort:snort snort

cd /usr/local/lib

chown -R snort:snort snort*

chown -R snort:snort snort_dynamic*

chown -R snort:snort pkgconfig

chmod -R 700 snort*

chmod -R 700 pkgconfig

cd /usr/local/bin

chown -R snort:snort daq-modules-config

chown -R snort:snort u2*

chmod -R 700 daq-modules-config

chmod 700 u2*

cd /etc

chown -R snort:snort snort

chmod -R 700 snort

- Now snort is installed and ready to use. To initially test snort installation, use this command,

cd /usr/local/bin

./snort -T -i eth0 -u snort -g snort -c /etc/snort/snort.conf

- If snort was installed properly it gives this output,

Snort successfully validated the configuration!

Snort exiting

APPENDIX C

Picture 1. Nmap test result on Blackbuntu attacker machine:



```
Applications Places System testing... BlackB... USA 93°F Mon Jun 10, 11:16
BlackBuntu
--append-output: Append to rather than clobber specified output files
--resume <filename>: Resume an aborted scan
--stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Nmap.Org for more portable XML
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
-6: Enable IPv6 scanning
-A: Enable OS detection, version detection, script scanning, and traceroute
--datadir <dirname>: Specify custom Nmap data file location
--send-eth/--send-ip: Send using raw ethernet frames or IP packets
--privileged: Assume that the user is fully privileged
--unprivileged: Assume the user lacks raw socket privileges
-V: Print version number
-h: Print this help summary page.
EXAMPLES:
nmap -v -A scanme.nmap.org
nmap -v -sn 192.168.0.0/16 10.0.0.0/8
nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (http://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
root@bb03~# nmap 192.168.0.2

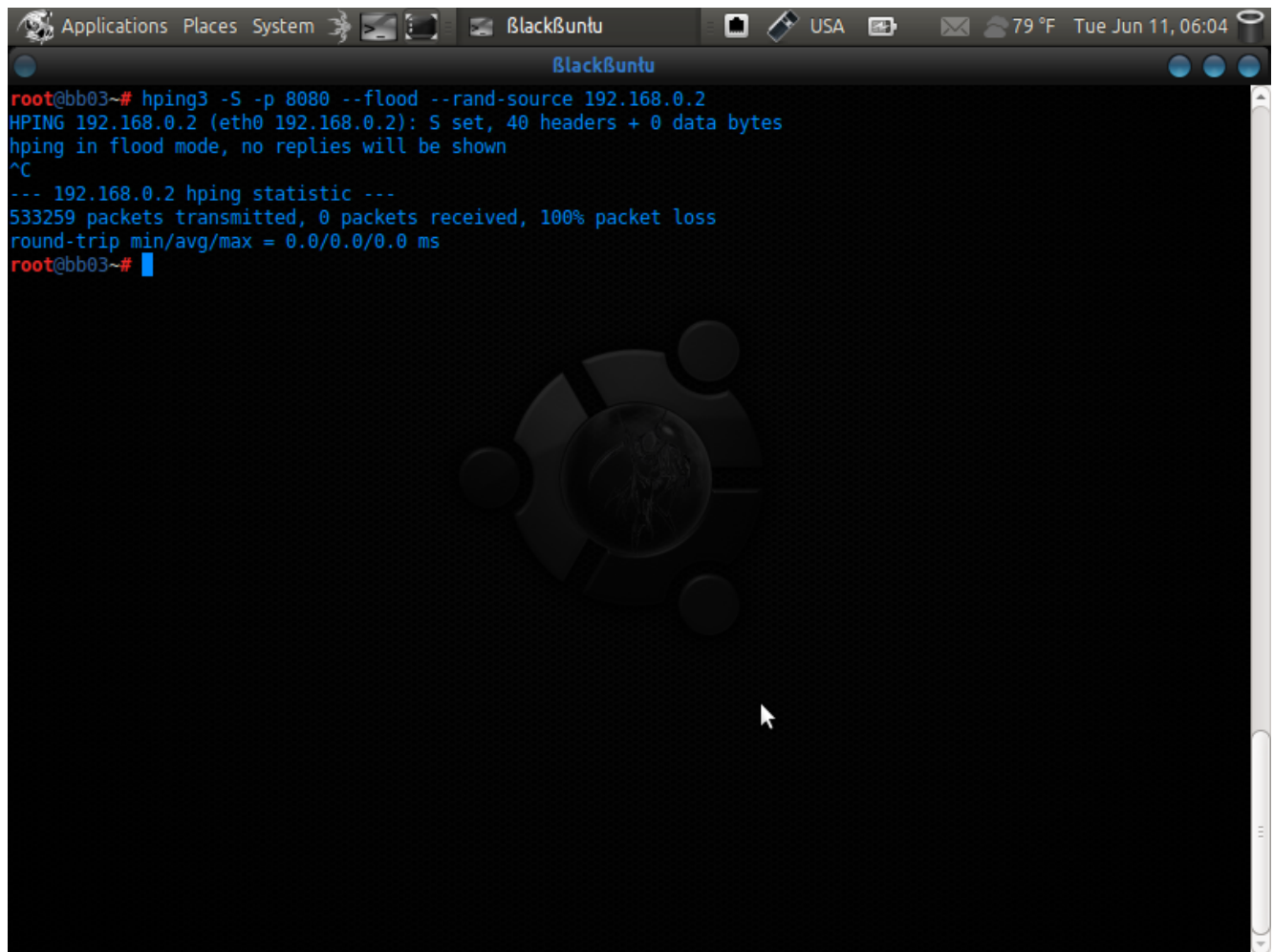
Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-10 11:13 ICT
Nmap scan report for 192.168.0.2
Host is up (0.026s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
111/tcp   open  rpcbind
5000/tcp  open  upnp
8080/tcp  open  http-proxy
8443/tcp  open  https-alt
8888/tcp  open  sun-answerbook

Nmap done: 1 IP address (1 host up) scanned in 90.65 seconds
root@bb03~#
```

Snort test results for nmap:

```
06/09-09:31:21.758499 [**] [122:1:0] (portscan) TCP Portscan [**] [Priority:3]
{PROTO:255} 192.168.0.10 -> 192.168.0.2
06/09-09:31:21.758543 [**] [122:1:0] (portscan) TCP Portscan [**] [Priority:3]
{PROTO:255} 192.168.0.10 -> 192.168.0.2
06/09-09:31:22.760457 [**] [122:1:0] (portscan) TCP Portscan [**] [Priority:3]
{PROTO:255} 192.168.0.10 -> 192.168.0.2
```

Picture 2. Screenshot for hping3

A screenshot of a Linux terminal window titled "BlackBuntu". The terminal shows the execution of the command `hping3 -S -p 8080 --flood --rand-source 192.168.0.2`. The output indicates that the flood mode is active and that 533259 packets were transmitted with 0 received, resulting in a 100% packet loss. The round-trip time statistics are shown as 0.0/0.0/0.0 ms. A faint Ubuntu logo is visible in the background of the terminal.

```
root@bb03~# hping3 -S -p 8080 --flood --rand-source 192.168.0.2
HPING 192.168.0.2 (eth0 192.168.0.2): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.0.2 hping statistic ---
533259 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@bb03~#
```

Snort test results for hping3:

```
06/10-17:03:19.728174 [**] [1:12121:0] DoS SYN flood attack detected! [**]
[Priority:0]{TCP}192.168.0.10:60701 ->192.168.0.2:8080
06/10-17:03:29.670712 [**] [1:12121:0] DoS SYN flood attack detected! [**]
[Priority:0]{TCP}192.168.0.10:60896 ->192.168.0.2:8080
06/10-17:03:40.561631 [**] [1:12121:0] DoS SYN flood attack detected! [**]
[Priority:0]{TCP}192.168.0.10:61069 ->192.168.0.2:8080
```

Picture 3. Screenshot for BED (BruteForce Exploit Detection)

```
root@bb03/pentest/fuzzer/bed# ./bed.pl -s http -t 192.168.0.2 -p 8080
BED 0.5 by mjm ( www.codito.de ) & eric ( www.snake-basket.de )

+ Buffer overflow testing:
  testing: 1   HEAD XAXAX HTTP/1.0 .....
  testing: 2   HEAD / XAXAX .....
  testing: 3   GET XAXAX HTTP/1.0 .....
  testing: 4   GET / XAXAX .....
  testing: 5   POST XAXAX HTTP/1.0 .....
  testing: 6   POST / XAXAX .....
  testing: 7   GET /XAXAX .....
  testing: 8   POST /XAXAX .....
+ Formatstring testing:
  testing: 1   HEAD XAXAX HTTP/1.0 .....
  testing: 2   HEAD / XAXAX .....
  testing: 3   GET XAXAX HTTP/1.0 .....
  testing: 4   GET / XAXAX .....
  testing: 5   POST XAXAX HTTP/1.0 .....
  testing: 6   POST / XAXAX .....
  testing: 7   GET /XAXAX .....
  testing: 8   POST /XAXAX .....
* Normal tests
+ Buffer overflow testing:
  testing: 1   User-Agent: XAXAX .....
  testing: 2   Host: XAXAX .....
  testing: 3   Accept: XAXAX .....
  testing: 4   Accept-Encoding: XAXAX .....
  testing: 5   Accept-Language: XAXAX .....
  testing: 6   Accept-Charset: XAXAX .....
  testing: 7   Connection: XAXAX .....
  testing: 8   Referer: XAXAX .....
  testing: 9   Authorization: XAXAX .....
  testing: 10  From: XAXAX .....
  testing: 11  Charge-To: XAXAX .....
  testing: 12  Authorization: XAXAX .....
  testing: 13  Authorization: XAXAX : foo .....
```

Snort test results for BED:

06/09-21:47:45.456569 [**] [1:1104:6] WEB-MISC whisker space splice attack
[**] [Classification: Attempted Information Leak] [Priority: 2] {TCP}
192.168.0.10:65234 -> 192.168.0.2:8080

06/09-21:47:47.482994 [**] [1:1171:7] WEB-MISC whisker HEAD with large
datagram [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP}
192.168.0.10:65236 -> 192.168.0.2:8080

References:

1. Chi-Chun Lo, Chun-Chieh Huang, Joy Ku, “A Cooperative Intrusion Detection System Framework for Cloud Computing Networks”, Parallel Processing Workshops (ICPPW), 2010, 39th International Conference on 13-16 Sept. 2010.
2. W. Yassin, N.I. Udzir, Z. Muda, A. Abdullah and M.T. Abdullah, “A Cloud-based Intrusion Detection Service framework”, Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on 26-28 June 2012
3. Saketh Bharadwaja, Weiqing Sun, Mohammed Niamat, Fangyang Shen, “Collabra: A Xen Hypervisor based Collaborative Intrusion Detection System”, Information Technology: New Generations (ITNG), 2011 Eighth International Conference on 11-13 April 2011
4. Chirag N. Modi, Dhiren R. Patel, Avi Patel, Rajarajan Muttukrishnan, “Bayesian Classifier and Snort based Network Intrusion Detection System in Cloud Computing”, Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on 26-28 July 2012.
5. Sebastian Roschke, Feng Cheng, Christoph Meinel, “Intrusion Detection in the Cloud”, Dependable, Autonomic and Secure Computing, 2009. DASC '09. Eighth IEEE International Conference on 12-14 Dec. 2009
6. Chih-Hung Lin , Chin-Wei Tien, Hsing-Kuo Pao, “Efficient and Effective NIDS for Cloud Virtualization Environment”, Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on 3-6 Dec. 2012
7. Gustavo Nascimento, Miguel Correia, “Anomaly-based intrusion detection in software as a service”, Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on 27-30 June 2011.

8. S N Dhage, B B Meshram, R Rawat, S Padawe, M Paingaokar, A Misra, "Intrusion Detection System in Cloud Computing Environment", International Conference and Workshop on Emerging Trends in Technology (ICWET 2011) – TCET, Mumbai, India.
9. Ms. Parag K. Shelke, Ms. Sneha Sontakke, Dr. A. D. Gawande, "Intrusion Detection System for Cloud Computing", International Journal of Scientific & Technology Research Volume 1, Issue 4, May 2012.
10. Sanchika Gupta, Anjali Sardana, Padam Kumar, Ajith Abraham, "A secure and lightweight approach for critical data security in cloud", Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on 21-23 Nov. 2012
11. Ashalatha R, "A SURVEY ON SECURITY AS A CHALLENGE IN CLOUD COMPUTING", International Journal of Advanced Technology & Engineering Research (IJATER) , National Conference on Emerging Trends in Technology (NCET-Tech)
12.
<http://www.jameslovecomputers.com/network-fuzzer-fuzz-various-protocols-witeh-bed-pl/>
13. http://home.iitk.ac.in/~subhali/reports/report_ee673.pdf
14. <http://www.iplocation.net/tools/denial-of-service.php>
15. <http://linuxgazette.net/126/cherian.html>
16. <http://insecure.org/sploits/ping-o-death.html>
17.
[http://www.iss.net/security_center/advice/Underground/Hacking/Methods/Technical / Spoofing/default.htm](http://www.iss.net/security_center/advice/Underground/Hacking/Methods/Technical/Spoofing/default.htm)
18. http://en.wikipedia.org/wiki/Man-in-the-middle_attack
19. http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-4/dos_atta

cks.html

20. <http://packetstormsecurity.com/files/view/54973/port-scanning-techniques.txt>

21. http://en.wikipedia.org/wiki/Smurf_attack

22.

http://www.iss.net/security_center/advice/Underground/Hacking/Methods/Technical/default.htm

23. <http://www.dummies.com/how-to/content/denial-of-service-attacks-and-how-to-guard-against.html>

24. http://www.firewalls.com/blog/ssh_brute_force_attack/

25. http://en.wikipedia.org/wiki/Brute-force_attack

26. <http://www.rackaid.com/resources/how-to-block-ssh-brute-force-attacks/>

27. <http://shishirceh.blogspot.ca/2011/05/hping3-examples.html>

28. <http://en.wikipedia.org/wiki/Nmap>

29. http://en.wikipedia.org/wiki/IP_address_spoofing

30. <http://www.securitytube-tools.net/index.php@title=Bed.html>

31. <https://www.owasp.org/index.php/Fuzzing>

32. <http://www.codenomicon.com/products/buzz-on-fuzzing.shtml>

33. <http://web2.uwindsor.ca/courses/cs/aggarwal/cs60564/Assignment1/Won.pdf>

34. http://www.sans.org/reading_room/whitepapers/detection/denial-service-attacks-mitigation-techniques-real-time-implementation-detailed-analysis_33764