Distributed Port-Scan Attack in Cloud Environment

Prachi Deshpande[1], Aditi Aggarwal[1], S.C.Sharma[1], P.Sateesh Kumar[1]
[1]Indian Institute of Technology Roorkee, Roorkee-India-247 667
{deprachi3, aditi127, scs60fpt, prof.sateesh} gmail.com

Ajith Abraham [2,3]
[2] Machine Intelligence Research Labs (MIR Labs), WA, USA.
[3] IT4Innovations - Center of Excellence, VSB - Technical University of Ostrava, Czech Republic.
ajith.abraham@ieee.org

Abstract— **Cloud Computing is becoming a promising technology for processing a huge chunk of data. Hence, its security aspect has drawn the attentions of researchers and academician. The security of the cloud environment must be reliable as well as scalable.**

The cloud environment is vulnerable to many security attacks. Attacks can be launched individually or in tandem. In this article, the overview of port-scan attack and the response of IDS are studied. The experimentation is carried out using virtual-box and SNORT, the open-source IDS.

Keywords- ***Cloud computing; Firewall; Distributed attacks; Intrusion Detection System; Port-scan; Security.***

I. INTRODUCTION

According to National Institute of Standards and Technology (NIST), Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1].

Cloud computing refers to a collection of computing and communication resources that are shared by many different users. It is considered as internet based computing service provided by various infrastructure providers on an on-demand basis. It provides high performance computing for many data intensive and scientific applications with easy scalability. Deshpande et al [2] illustrated a collection of various errors and the possible solution to set up a private cloud.

Security in cloud computing is key aspect which is most desired by a cloud user. Data privacy and security concerns are discussed in [3] with provision of trusted third party as a solution for providing security solutions. Intrusion detection system (IDS) based approaches was proposed for cloud security in [4]. Most general security attacks in the cloud environment includes flooding, Denial of service, root-kits, port-scan, malwares [5-6]. An evolutionary design is proposed in [7] for intrusion detection. Further, in this regard, an IDS using hybrid intelligence is proposed in [8], which is helpful under variety of conditions. Further the approach in [8] is extended for the mobile computing environment in [9] by Alvaro et.al. To improve the performance of public cloud monitoring, a lightweight monitoring framework was proposed in [10].The article discussed various performance related issues in cloud computing and its security. Different type of intrusion detection systems in cloud with their limitations is nicely categorized in [11].

The Criminal psychology starts with the finding the loopholes in the system. First step toward launching the attack is to get the information about the system by port-scanning. With the aid of port-scanning, attacker can get information like open ports, supported network services, and protocols used by the host.

The attacks can be launched in various stages, of which the first stage is to get maximum information about the target. Scanning with stealth scanner is preferred by intelligent attacker to retrieve information of the target. On the basis of collected information, attacker tries to gain access of the target. After successful access of the target, the attacker tries to get the enhanced privileges to achieve its goal. Malicious code is inserted by attacker after gaining the required privileges.

II. PROPOSED METHOD

In this article, port-scan attack is evaluated. Basic information about the user can be easily extracted by using it. The port-scan attack is classified as Horizontal port-scan, Vertical port-scan and block scan [12].

The Brute force port-scan and stealth scan attacks are also used for port-scanning. 'Brute force' scanners scan the port in a sequential manner one after another for the range specified by the user. These scanners can easily get detected. Stealth scanner technique is more sophisticated one in which attackers send a single packet with a specified flag. When attacker gets reply for this packet, ports are determined [12].

Providing security and privacy to cloud user is essential as the attackers' can be outsider as well as an insider i.e. a virtual machine exploiting vulnerabilities of the system to launch the attack. To provide security, use of intrusion detection system is suggested along with the analysis of placement of intrusion detection system inside cloud [4].

Nowadays various techniques are used to evade the intrusion detection system. A careful scan at a rate lower than the threshold can easily go undetected [12]. 'Decoys'

978-1-4799-1409-8/13/$31.00 2013 IEEE

are the hosts which are up and idle. These systems can be used to launch port-scanning attack along with the actual attacker. They help in hiding the IP of the attacker. IPs of the attacker and the IPs of all the 'Decoys' are mixed. Victim may not be able to decide the IP of the attacker. Combining 'Decoys' and stealth port-scanning techniques, port-scanning attack can be launched on target machines by the attacking virtual machines (VM). One of the major challenges for the attacker is to find out the 'Decoys' available on the network. Fig.1 shows the proposed model with different VM acting as an attacker.

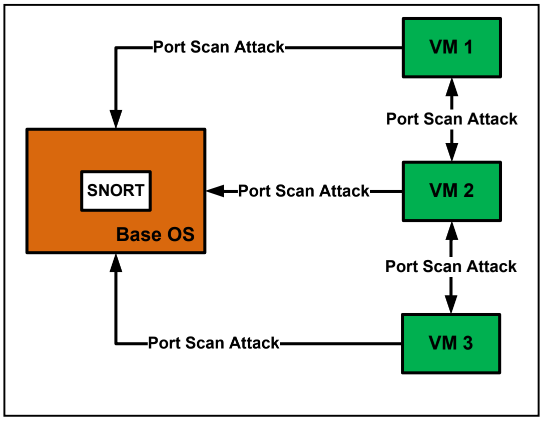


Figure 1. Port-Scanning attack by VM.

Once some of the 'Decoys' are found, those can be used persistently to launch the attack on the victim. In 'decoy' scanning many 'Decoys' launch the attack on the victim. Similarly, in distributed port-scanning, multiple virtual machines launch the attack on a host in the cloud infrastructure. In 'decoy' scanning idle hosts are used by a single machine for launching the attack while in distributed scanning multiple machines launch attack with their own resolve. They are not being directed by any single virtual machine.

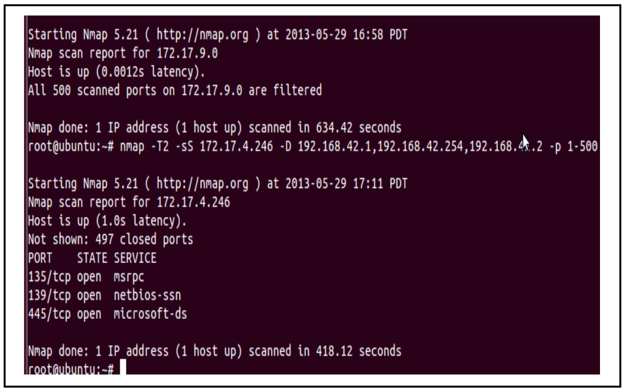
In this analysis, the base operating system is Windows 2007. Virtual-box software is installed on windows operating system for creation of VM. Two Ubuntu VM are created with the help of virtual-box. SNORT is used as intrusion detection system. A 'SNORT-2.9.4.6' has been deployed on the base operating system to identify the effects of intrusion on base operating system and VM. Various open source tools such as 'Nmap', 'Metasploit', and 'Scapy' are used for scanning the target machine information

The primary goal of this analysis is to launch port-scanning attack such that it should not be detected by an intrusion detection system. 'Nmap-6.25' (Network Mapper) is used to launch port-scanning attack from one virtual machine to another. Various options are available with Nmap for this purpose.

Apart from 'Nmap', 'Metasploit' and Scapy are also used to verify the performance. A graphical user interface called 'Zenmap' is provided by Nmap suite so as to provide user friendliness to all the users. 'Zenmap' provides all the features of Nmap. In stealthy scanning, a very slow rate system scan is achieved by transmitting packets at slow pace. After sending the first packet, 'Nmap' waits for some time and then sends the second packet. Between the successive transmissions of every two packets 'Nmap' waits for specified time delay. Most of the IDS work on the principal of 'X' number of probes in 'Y' time units. By launching stealthy scan, criterion of IDS is not satisfied and it may not be able to detect the attack.

III. Results And Discussions

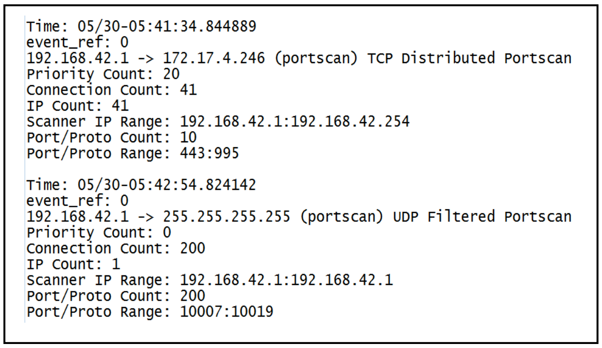
During analysis, firstly, using 'Nmap' tool two VM are scanning the target IPs i.e. 172.17.4.246. The IPs scanning the target using 'Nmap' utility are 192.168.42.1., 192.168.42.2 and 192. 168.42.254. Ubuntu 12.10 VM and Ubuntu 12.04 VM are used for scanning the target. Fig.2 shows the scanning results of Ubuntu 12.04 VM.



```
Starting Nmap 5.21 ( http://nmap.org ) at 2013-05-29 16:58 PDT
Nmap scan report for 172.17.9.0
Host is up (0.0012s latency).
All 500 scanned ports on 172.17.9.0 are filtered

Nmap done: 1 IP address (1 host up) scanned in 634.42 seconds
root@ubuntu:~# nmap -T2 -sS 172.17.4.246 -D 192.168.42.1,192.168.42.254,192.168.42.2 -p 1-500

Starting Nmap 5.21 ( http://nmap.org ) at 2013-05-29 17:11 PDT
Nmap scan report for 172.17.4.246
Host is up (1.0s latency).
Not shown: 497 closed ports
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 418.12 seconds
root@ubuntu:~#
```

Figure 2. Scanning environment target using Ubuntu 12.04 VM

The 'Nmap' tool uses Ubuntu as the launch pad for port-scanning purpose. The 'Nmap' scans the target IP and provides the information about the available services and ports. This is helpful for the attacker to gain the privileged access of the target. Fig.3 shows the log entries of IDS response for the port-scan attack using 'Nmap'

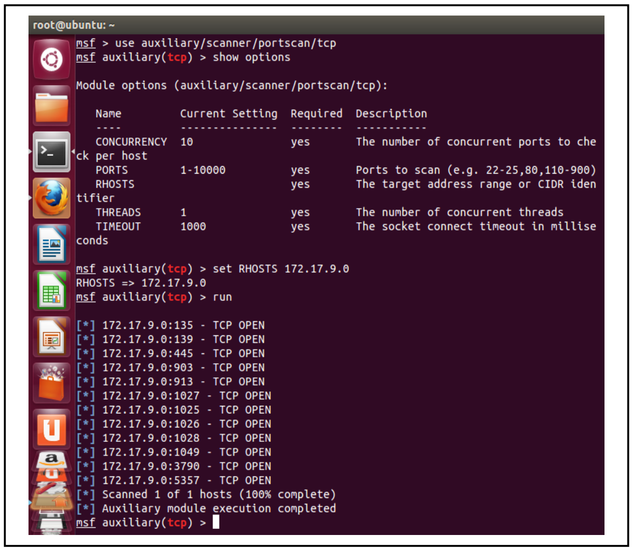


```
Time: 05/30-05:41:34.844889
event_ref: 0
192.168.42.1 -> 172.17.4.246 (portscan) TCP Distributed Portscan
Priority Count: 20
Connection Count: 41
IP Count: 41
Scanner IP Range: 192.168.42.1:192.168.42.254
Port/Proto Count: 10
Port/Proto Range: 443:995

Time: 05/30-05:42:54.824142
event_ref: 0
192.168.42.1 -> 255.255.255.255 (portscan) UDP Filtered Portscan
Priority Count: 0
Connection Count: 200
IP Count: 1
Scanner IP Range: 192.168.42.1:192.168.42.1
Port/Proto Count: 200
Port/Proto Range: 10007:10019
```

Figure 3. Log entries of IDS response

SNORT has different priority levels for detection of attacks. These priorities indicate the bad responses against the possible attacks. Higher priority count indicates more number of bad responses. Further, the attacks are launched by using ‘Metasploit’ tool. It is used to launch TCP scan on the target virtual machine using the ‘Msfconsole’ interface provided by Metasploit. ‘Msfconsole’ provides different option to the user for launching exploits. Fig.4 shows the ‘Metasploit’ environment to launch TCP scan on the target machine.

TCP port-scanning is launched by writing the command for it in the first line. The command ‘show options’ is to check available module options. It provides the name of the option along with its description and the current settings. The values of the options can be modified according to the need. In this analysis, ‘RHOSTS’ option is used to specify the IP address of the target. The TCP ports which are open are shown in the Fig.4. It shows that the TCP ports 135, 139, 445, 903, 913, 1026, 1025, 1029, 1027, 1028, 3790 and 5357 are open on the target host. Port number is appended after the IP address of the target followed by a colon. A large variety of port-scanning attacks can be launched using ‘Msfconsole’ such as ‘ack firewall scan’, ‘ftp bounce port-scan’, ‘syn port-scan’, ‘tcp port-scan’ and ‘xmas port-scan’. ‘Ping scan’ and ‘NAT-PMP’ external port-scanning are also available.



```
root@ubuntu: ~
msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to che
ck per host
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                        yes       The target address range or CIDR iden
tifier
   THREADS      1                yes       The number of concurrent threads
   TIMEOUT      1000             yes       The socket connect timeout in millise
conds

msf auxiliary(tcp) > set RHOSTS 172.17.9.0
RHOSTS => 172.17.9.0
msf auxiliary(tcp) > run

[*] 172.17.9.0:135 - TCP OPEN
[*] 172.17.9.0:139 - TCP OPEN
[*] 172.17.9.0:445 - TCP OPEN
[*] 172.17.9.0:903 - TCP OPEN
[*] 172.17.9.0:913 - TCP OPEN
[*] 172.17.9.0:1027 - TCP OPEN
[*] 172.17.9.0:1025 - TCP OPEN
[*] 172.17.9.0:1026 - TCP OPEN
[*] 172.17.9.0:1028 - TCP OPEN
[*] 172.17.9.0:1049 - TCP OPEN
[*] 172.17.9.0:3790 - TCP OPEN
[*] 172.17.9.0:5357 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(tcp) > █
```

Figure 4. ‘Metasploit’ environment for TCP scan

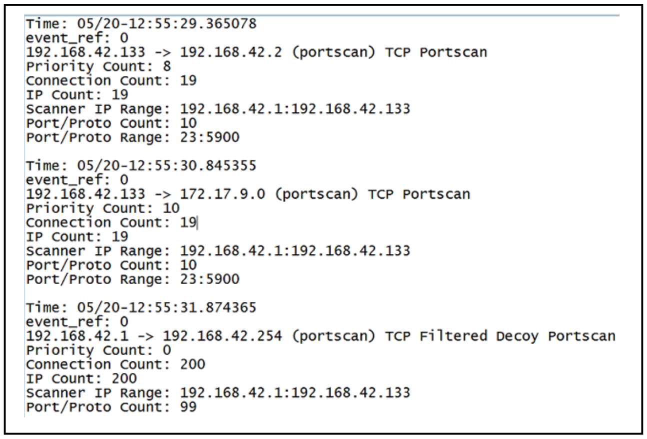
Fig.5. shows scanning results using ‘Metasploit’ tool. At port 445, the information about the operating system on the target machine is revealed. It also exposes the various services available at each port of the target machine. This information is very much important for the attacker to gain the privileged access of the target.

The log entry of SNORT when ‘Metasploit’ is used is shown in Fig.6. It also shows that with ‘0’ priority level, ‘‘Decoys’’ are detected. Fig.7 shows the usage of ‘Scapy’ to launch SYN scan. Scapy is launched using Scapy command from the terminal. The destination of packets is specified using the ‘dst’ command. Ports which are to be scanned on target are listed using ‘dport’.

SYN flag is set by making flags equal to ‘S’. Any other flags can also be specified. In this analysis, six packets are sent to the target as six ports are specified. In turn, 134 packets are received from the target. In total, five answers are received. It indicates that the status of five ports is known to the attacking virtual machine.

Port	protocol	Name	Information
135	tcp	dcerpc	Endpoint Mapper (151) services
138	tcp	smb	
445	tcp	smb	Windows 7 Home Basic (Build 7601) (language: Unknown) (name: WINDOWS) (domain: workgroup)
1025	tcp	dcerpc	D95afe70-a6d5-4259-822 e-2c84da 1ddb0d v1.0
1026	tcp	dcerpc	b25a52bf-e5dd-4f4a-aea 6 8ca7272a0e86 v1.0keylso

Figure 5. Scanning results using ‘Metasploit’ tool

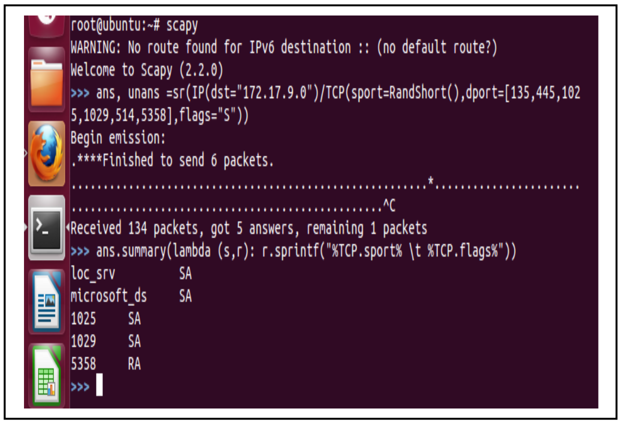


```
Time: 05/20-12:55:29.365078
event_ref: 0
192.168.42.133 -> 192.168.42.2 (portscan) TCP Portscan
Priority Count: 8
Connection Count: 19
IP Count: 19
Scanner IP Range: 192.168.42.1:192.168.42.133
Port/Proto Count: 10
Port/Proto Range: 23:5900

Time: 05/20-12:55:30.845355
event_ref: 0
192.168.42.133 -> 172.17.9.0 (portscan) TCP Portscan
Priority Count: 10
Connection Count: 19
IP Count: 19
Scanner IP Range: 192.168.42.1:192.168.42.133
Port/Proto Count: 10
Port/Proto Range: 23:5900

Time: 05/20-12:55:31.874365
event_ref: 0
192.168.42.1 -> 192.168.42.254 (portscan) TCP Filtered Decoy Portscan
Priority Count: 0
Connection Count: 200
IP Count: 200
Scanner IP Range: 192.168.42.1:192.168.42.133
Port/Proto Count: 99
```

Figure 6. Log entry of Snort with ‘Metasploit’

After TCP SYN scan is being launched by the attacker, the status of the ports within the square brackets is analyzed. ‘SA’ and ‘RA’ indicates the ‘port is open’ and ‘port is closed’, respectively. Fig. 7 shows that ports 1025 and 1029 are open while port 5358 is closed.

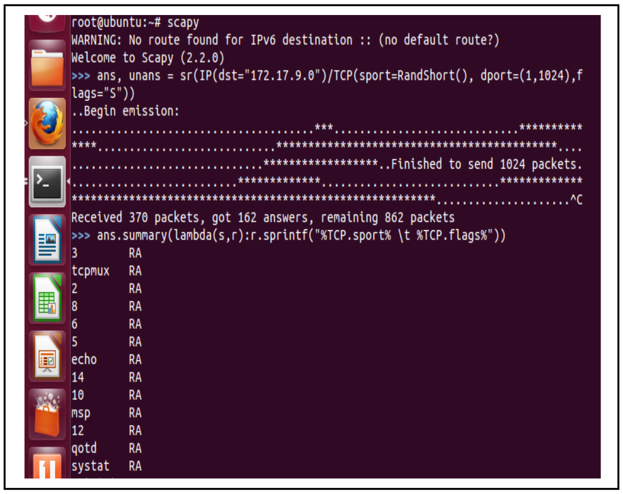
Instead of specifying individual ports, a range of ports can also be specified. To specify a range of ports parenthesis is used instead of square brackets. The range of ports to be scanned is specified in the Parenthesis. Fig. 8 shows launching of ‘SYN scan’ using ‘Scapy’ to a range of ports on the target virtual machine. Since 1024 ports are being scanned, 1024 packets are being sent to the target virtual machine by the attacker. 370 packets are received in response from the victim target machine and 162 answers are obtained.



```
root@ubuntu:~# scapy
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> ans, unans =sr(IP(dst="172.17.9.0")/TCP(sport=RandShort(),dport=[135,445,102
5,1029,514,5358],flags="S"))
Begin emission:
.****Finished to send 6 packets.
.........................................................*......................
...............................................^C
Received 134 packets, got 5 answers, remaining 1 packets
>>> ans.summary(lambda (s,r): r.sprintf("%TCP.sport% \t %TCP.flags%"))
loc_srv         SA
microsoft_ds    SA
1025    SA
1029    SA
5358    RA
>>> 
```

Figure 7. Use of 'Scapy' to launch SYN scan

The summary of the result is obtained by using the 'summary ()' command. The ports shown in Fig.8 are closed because TCP flag is RA.SNORT is run in IDS mode on the victim to see whether it can detect the scan launched by 'Scapy' or not.



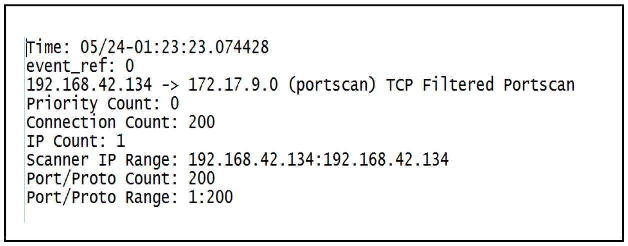
```
root@ubuntu:~# scapy
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> ans, unans = sr(IP(dst="172.17.9.0")/TCP(sport=RandShort(), dport=(1,1024),f
lags="S"))
..Begin emission:
.......................................***.................................**********
****...........................*******************************************....
...............................*****************..Finished to send 1024 packets.
...........................*************...........................************
***********************************************************.....................^C
Received 370 packets, got 162 answers, remaining 862 packets
>>> ans.summary(lambda(s,r):r.sprintf("%TCP.sport% \t %TCP.flags%"))
3        RA
tcpmux   RA
2        RA
8        RA
6        RA
5        RA
echo     RA
14       RA
10       RA
msp      RA
12       RA
qotd     RA
systat   RA
```

Figure 8. SYN scan using 'Scapy' on a range of ports

The Fig.9 shows the report maintained by the SNORT. The log entry shows the IP address of the attacker and victim. The type of attack being launched is also specified. Priority count is kept zero. Higher the priority count more dangerous is the scan. For attacker, lower priority count is desired so that the victim assumes that the log entry to be a false positive.

By analyzing the different tools for port-scanning, it can be concluded that with high priority, the attacks can be detected by SNORT. This priority level information can be used to enhance the prevention mechanism. Further, 'Metasploit' and 'Scapy' is the best option for launching the attacks.

With port-scan attack, various information of the target is identified by the attacker. By using it, further attacks can be launched by the attacker to get the privileged access of the target machine/system.



```
Time: 05/24-01:23:23.074428
event_ref: 0
192.168.42.134 -> 172.17.9.0 (portscan) TCP Filtered Portscan
Priority Count: 0
Connection Count: 200
IP Count: 1
Scanner IP Range: 192.168.42.134:192.168.42.134
Port/Proto Count: 200
Port/Proto Range: 1:200
```

Figure 9. Log entries for 'Scapy' scan.

In this analysis, more than one VM is launching the attack. The situation become worst when the all the attacking VM shares the scan information with each other. To cope with such condition, script of IDS has to be modified.

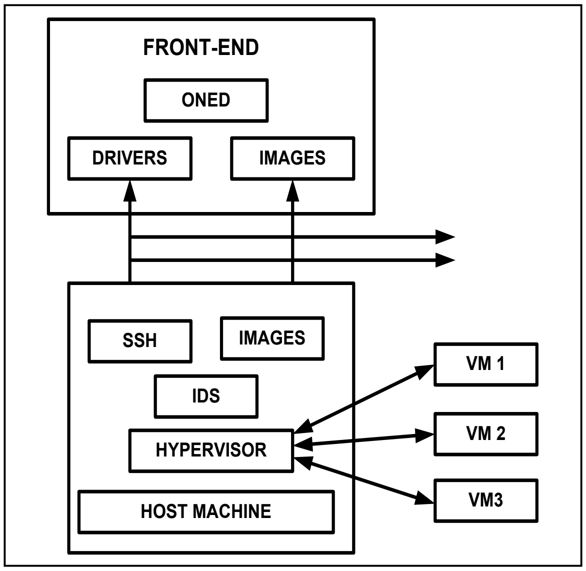


Figure 10. Private cloud using Ubuntu [2]

Fig.10 shows a private cloud implemented by [2]. Table I gives the analysis of the capabilities of different tools used for port-scan attack.

TABLE I. ANALYSIS OF PORT SCANNING TOOLS

Tools Used	**Parameters**		
	Scanning Method	***Packet Crafting by User Specification***	***Priority Count With Varied Sense Level***
Nmap	Not Specific	Not allowed	0 or 20
Metasploit	TCP	Possible	0,8 or 10
Scapy	UDP	Easy to perform	0

Table I shows that with varying sense level, SNORT can detect any type of port- scan attack.

IV. Conclusions

The port-scan attack is verified using SNORT IDS and VMs. This attack can be used by the intruders to gain the privileged access of the target system as it provides information like open ports, operating system, protocols used and network services of target machine.

In this analysis, 'Nmap', 'Metasploit' and 'Scapy' tools are used for launching the attacks. It has been found that 'Metasploit' and 'Scapy' are providing more detailed information about the target machine and its environment.

Services information of target OS is given by 'Metasploit'. Using this information the exploits can be built such as privilege escalations. With the help of 'Scapy' different packets can be crafted utilizing the information gathered by port scanning.

Detection and prevention of port-scan attack can naturalize the possibility of future attacks on the cloud environment. In future, port-scan attack is to be further verified on the private cloud environment. Also the condition of attackers' interaction with each other has to be verified on the cloud set up in [2] and a defense mechanism against these conditions is to be proposed.

References

[1] E. Brown, "NIST issues cloud computing guidelines for managing security and privacy," National Institute of Standards and Technology Special Publication 800-144, Jan. 2012.

[2] P. Deshpande, S. Sharrna, P. Kumar, "Deployment of private cloud: Go through the errors first," Proc. of Conference on Advances in Communication and Control Systems 2013 (CAC2S 2013), Deharadun-India, Apr.2013, pp.638-641.

[3] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," Future Generation Computer System, vol.28, Mar. 2012, pp.583-592.doi: http://dx.doi.org/10.1016/j.future.2010.12.006.

[4] S. Roschke, F. Cheng, and C. Meinel, "Intrusion Detection in the Cloud" in 8th IEEE International Conference on Dependable, Autonomic and Secure Computing, Dec.2009,pp.729-734. doi: 10.1109/DASC.2009.94

[5] R. K. C. Chang, "Defending against Flooding-Based Distributed denial-of-service attacks: A tutorial," IEEE Communications Magazine, Oct. 2002, pp.42-51.doi: 10.1109/MCOM.2002.1039856.

[6] N. Gruschka and M. Jensen, "Attack Surfaces: A taxonomy for attacks on cloud services," 3rd IEEE International conference on cloud computing, Miami, FL, Jul. 2010, pp.276-279. doi: 10.1109/CLOUD.2010.23.

[7] A. Abraham, C. Grosan and C. M.-Vide, "Evolutionary design of intrusion detection programs," International Journal of Network Security, vol.4, no.3, Mar. 2007, pp. 328-339.

[8] S. Peddabachigari, A. Abraham, C. Grosan and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," Journal of Network and Computer Applications, Elsevier Science, vol. 30, no.1,Jan.2007, pp.114-132.

[9] A. Herrero, E. Corchado, M. Pellicer and A. Abraham, "MOVIH-IDS: A Mobile-Visualization hybrid intrusion detection system," Neurocomputing Journal, vol. 72, no. 13-15, Aug. 2009, pp. 2775-2784.

[10] K. Ma, R. Sun and A. Abraham, "Towards lightweight framework for monitoring public cloud," Fourth International Conference on Computational Aspects of Social Networks (CASoN-2012), pp. 361-365, 2012.

[11] C. Modi, D.Patel, B. Borisaniya, H. Patel, and A. Patel, "A survey of intrusion detection techniques in cloud," Journal of Network and Computer Applications, vol. 36, no. 1, Jan. 2013, pp.42–57. dio: http://dx.doi.org/10.1016/j.jnca.2012.05.003.

[12] M. Bhuyan, D. Bhattacharyya, and J. Kalita, "Surveying port-scans and their detection methodologies", The Computer Journal, vol.54, no.10, Oct. 2011, pp. 1565-1581.

[13] Y. Chen, A. Abraham and B. Yang, Hybrid Flexible Neural Tree Based Intrusion Detection Systems, International Journal of Intelligent Systems, John Wiley and Sons, USA, Volume 22, pp. 1-16, 2007.

[14] A. Abraham, R. Jain, J. Thomas and S.Y. Han, D-SCIDS: Distributed Soft Computing Intrusion Detection Systems, Journal of Network and Computer Applications, Elsevier Science, Volume 30, Issue 1, pp. 81-98, 2007.

[15] S. Chebrolu, A. Abraham and J. Thomas, Feature Deduction and Ensemble Design of Intrusion Detection Systems, Computers and Security, Elsevier Science, Volume 24/4, pp. 295-307, 2005.

[16] S. Mukkamala, A. Sung and A. Abraham, Intrusion Detection Using Ensemble of Soft Computing and Hard Computing Paradigms, Journal of Network and Computer Applications, Elsevier Science, Vol. 28, Issue 2, pp. 167-182, 2005.

[17] S. Mukkamala, A. Sung, A. Abraham and Vitorino Ramos, Intrusion Detection Systems Using Adaptive Regression Splines, Enterprise Information Systems VI , Seruca, I.; Cordeiro, J.; Hammoudi, S.; Filipe, J. (Eds.) Springer-Verlag, ISBN: 1-4020-3674-4, pp. 211-218, 2006.