Introduction to Information Security

Programming Project - 2

Part I

Setup and Configuration

Virtualization Environment: VMware Workstation
Virtual Machines: Ubuntu 13.10 x86

Details of Virtual Machines

Three virtual machines were created using Ubuntu 13.10 x86 iso. The details are as follows:

1) **Virtual Machine 1**
   Name: IIS_Node_1
   Username: manish
   IP Address: 192.168.139.133
   Note: This is the VM on which firewall was setup
   SSH Server: openSSH Server has been installed

2) **Virtual Machine 2**
   Name: IIS_Node_2
   Username: manish2
   IP Address: 192.168.139.131
   Note: This is the VM from which the incoming ping requests and SSH connections for the IIS_Node_1 would be served

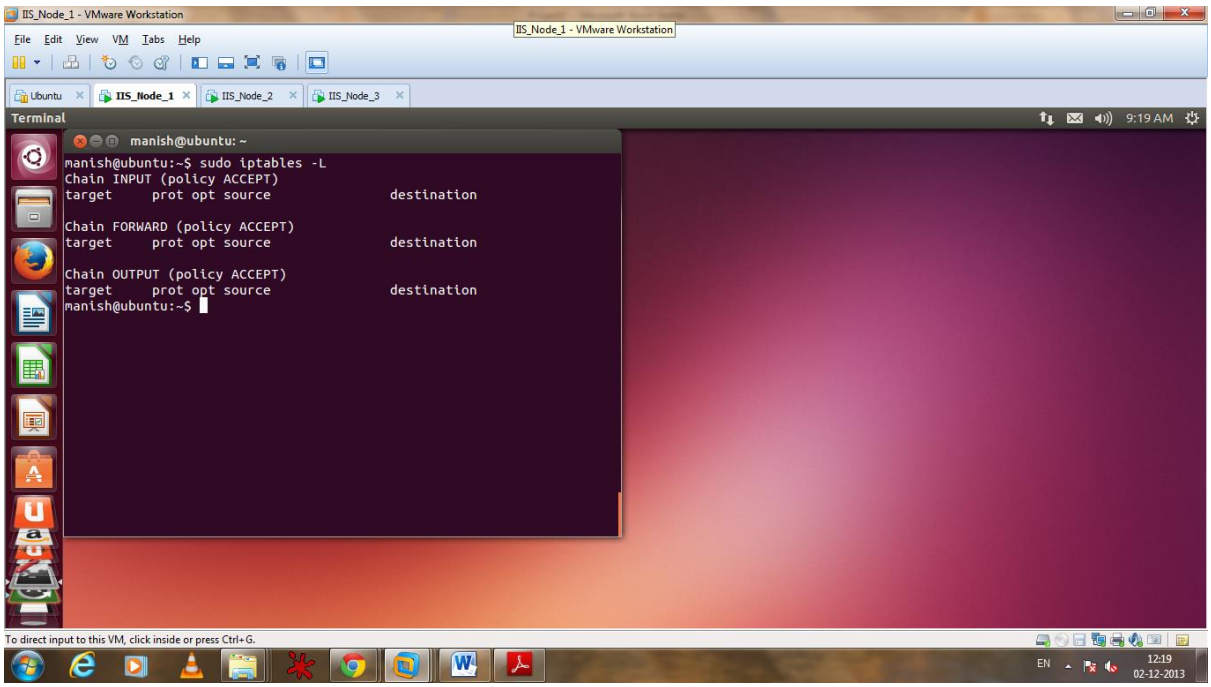
3) **Virtual Machine 3**
   Name: IIS_Node_3
   Username: manish3
   IP Address: 192.168.139.132
   Note: This is the VM from which all the requests for IIS_Node_1 would be declined

All the three virtual machines were connected via an internal network.

Initially, the check for the firewall table resulted in an empty table which meant that all requests were allowed by default.

Command to check firewall table: sudo iptables -L

Screenshot 1: Default – Allowed for all the chains



Screenshot 1

The ping requests and SSH connects from 'IIS_Node_2' and 'IIS_Node_3' to 'IIS_Node_1" were checked. All requests and connections were allowed.

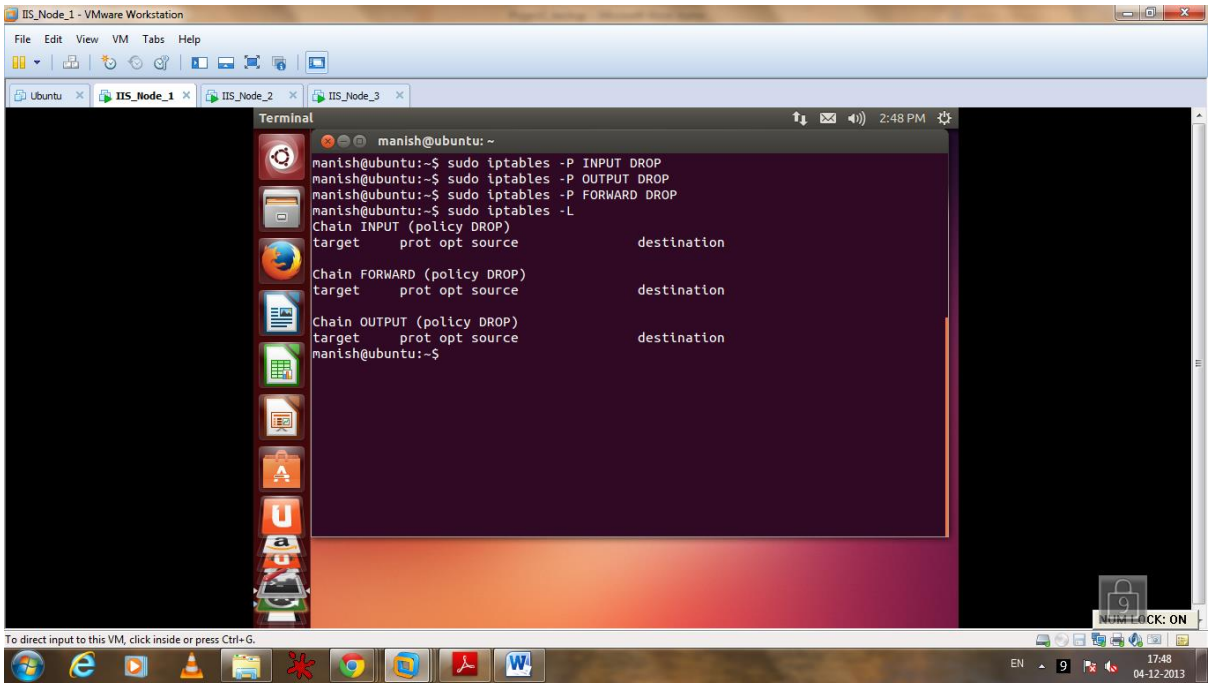
1. Change the default policy to DROP for all chains.

After initial check, the default policy on 'IIS_Node_1' was configured to drop for all the chains.

Commands Used: 1. sudo iptables -P INPUT DROP
2. sudo iptables -P OUTPUT DROP
3. sudo iptables -P FORWARD DROP

Here, -P specifies the policy for one of the three built-in chains i.e. INPUT, OUTPUT, and FORWARD. So, we set default policy to DROP for all the 3 built-in chains.

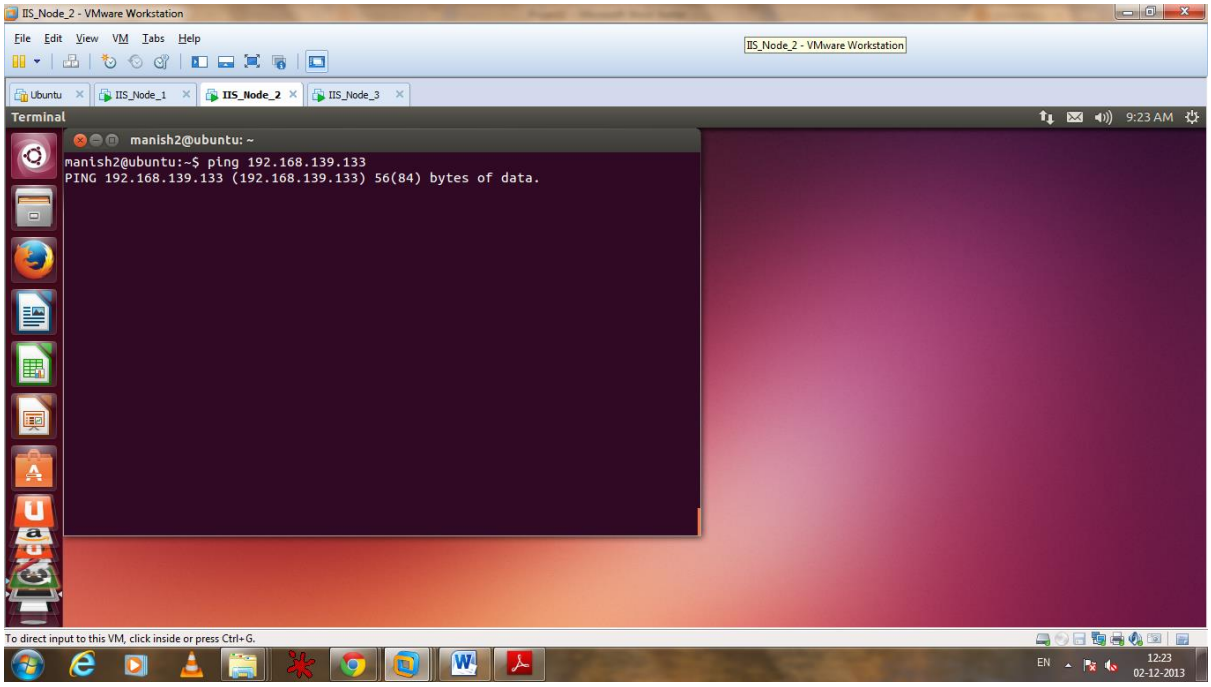
Screenshot 2: The screenshot of firewall table after setting default policy to drop for all the chains.



Screenshot - 2

After this, ping request from 'IIS_Node_2' to 'IIS_Node_1' (192.168.139.133) was tested. The ping request didn't work.

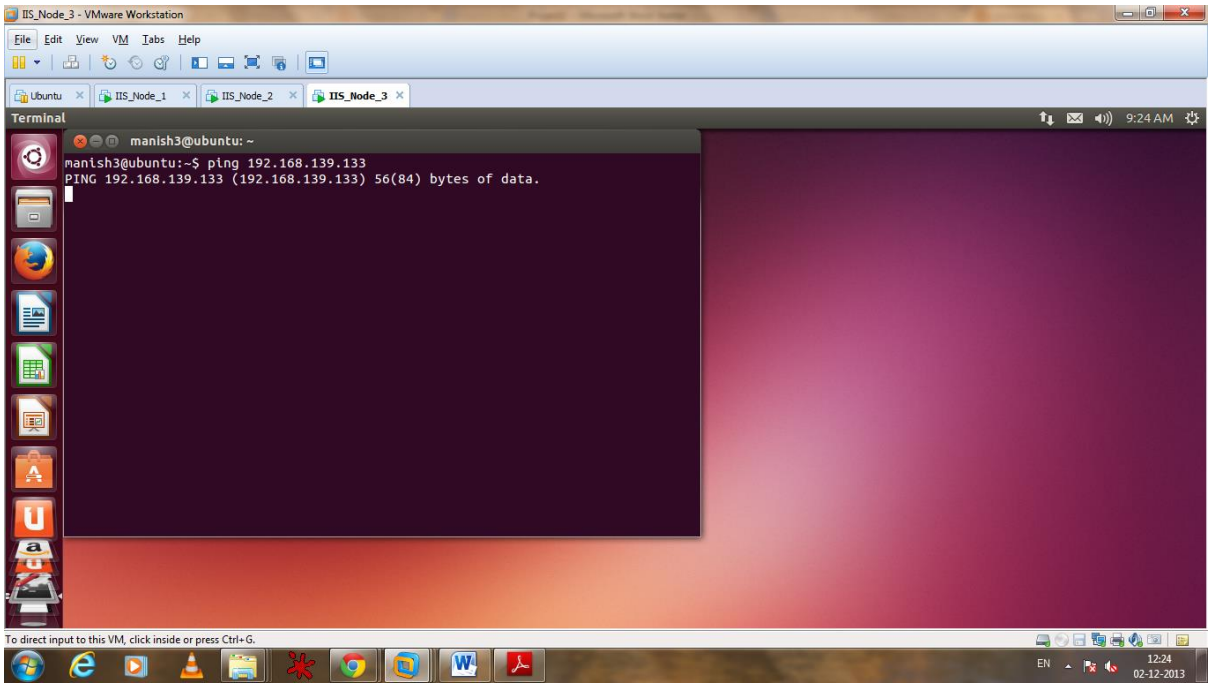
Screenshot 3: Ping from IIS_Node_2 to IIS_Node_1



Screenshot 3

Ping request from 'IIS_Node_3' to 'IIS_Node_1' (192.168.139.133) was also tested. The ping request didn't work.

Screenshot 4: Ping from IIS_Node_3 to IIS_Node_1



Screenshot 4

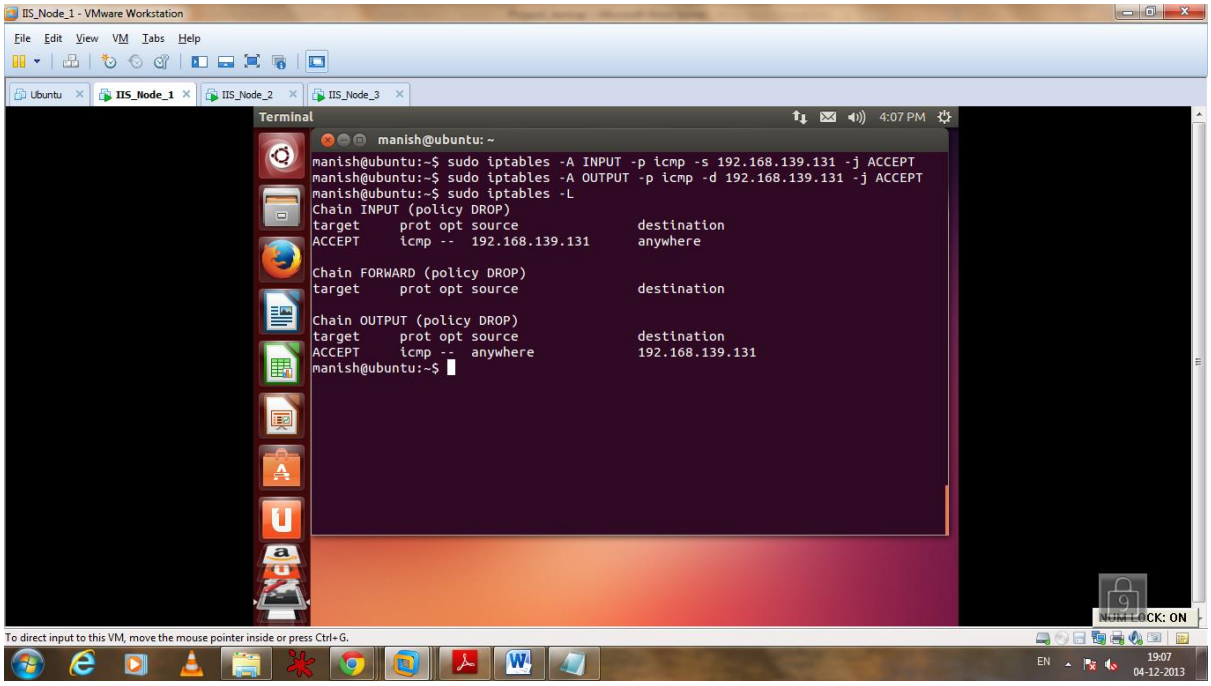
2. Serve incoming PING requests from Node 2 alone.

Now, the ping request from IIS_Node_2 is to be served alone. So, command was used to accept ICMP protocol requests coming from IP 192.168.139.131.

Commands Used: 1. sudo iptables –A INPUT –p icmp –s 192.168.139.131 –j ACCEPT
2. sudo iptables –A OUTPUT –p icmp –d 192.168.139.131 –j ACCEPT

Here, -A appends to the rules in a chain.
–p specifies the protocol.
-s specifies the source
-j specifies jump to target ACCEPT
-d specifies the destination

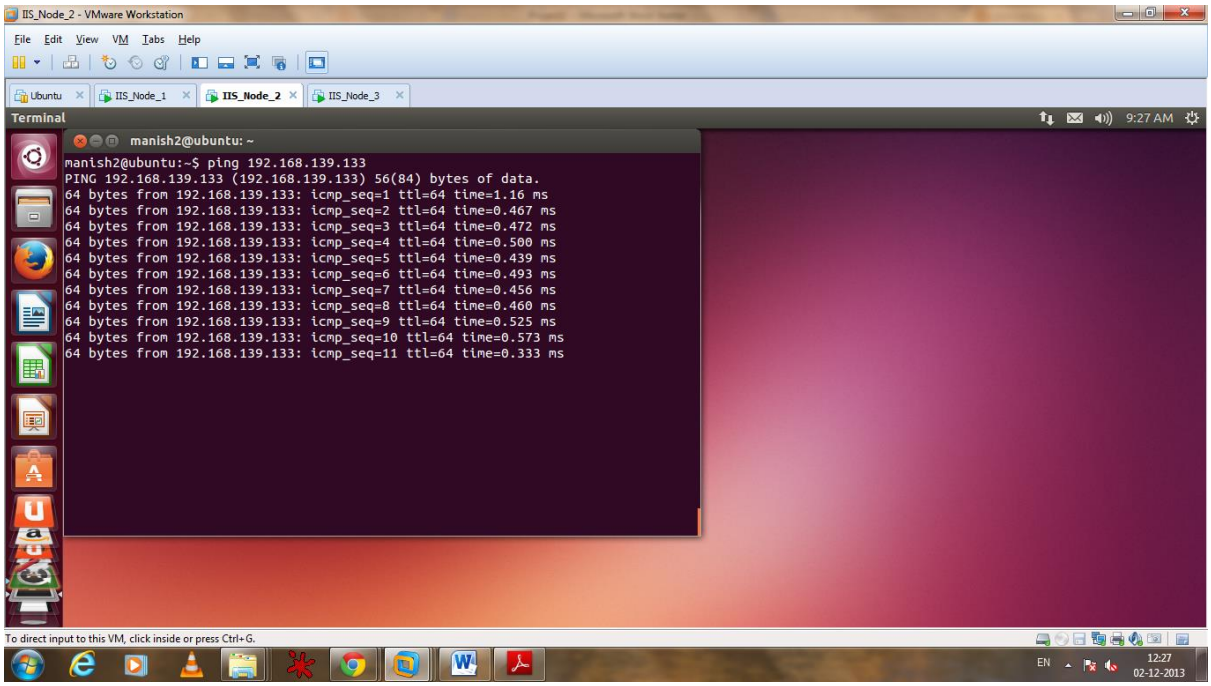
Screenshot 5: Rules to allow PING requests from IIS_Node_2 were added in the INPUT chain (to allow ping) and OUTPUT chain (to allow reply for the PING requests).



Screenshot 5

After this, ping request from 'IIS_Node_2' to 'IIS_Node_1' was tested which was successful.

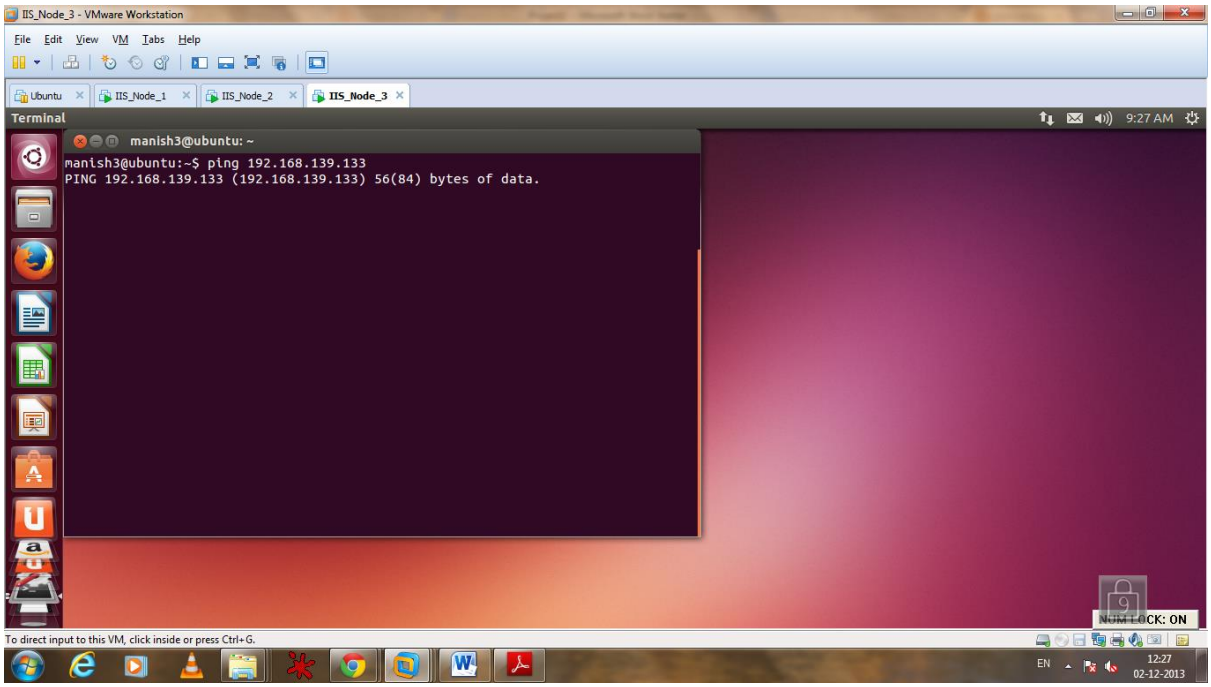
Screenshot 6: The ping request from 'IIS_Node_2' to 'IIS_Node_1'



Screenshot 6

Ping request from 'IIS_Node_3' to 'IIS_Node_1' was also checked. It was not successful.

Screenshot 7: The ping request from 'IIS_Node_3' to 'IIS_Node_1'



Screenshot 7

So, final result was that the ping requests from "IIS_Node_2" were served alone.

3. Serve incoming SSH connections from Node 2 alone.

Next goal was to serve incoming SSH connections from 'IIS_Node_2' only.

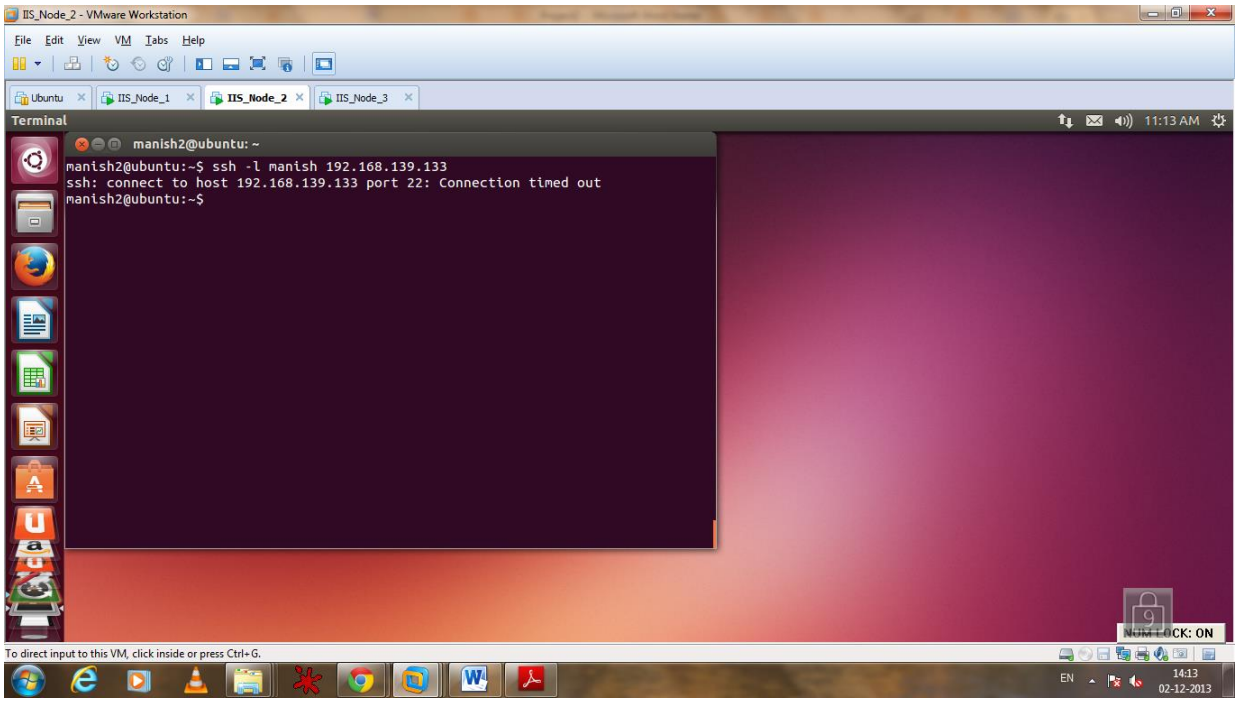
Before that,

OpenSSH server was installed on 'IIS_Node_1' using following command.

Command for OpenSSH Server: sudo apt-get install openssh-server

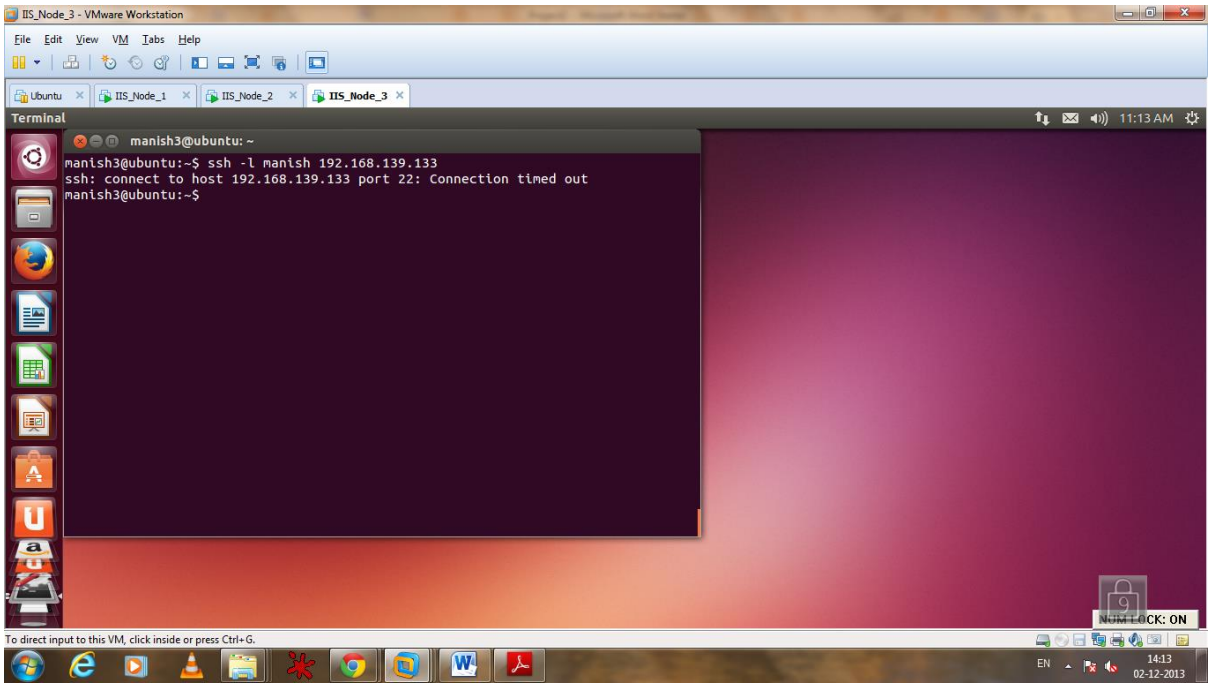
Then, SSH connections from 'IIS_Node_2' and 'IIS_Node_3' to 'IIS_Node_1' were tested which resulted in "Connection Timed Out".

Screenshot 8: SSH Connection from 'IIS_Node_2' to 'IIS_Node_1'



Screenshot 8

Screenshot 9: SSH Connection from 'IIS_Node_3' to 'IIS_Node_1'



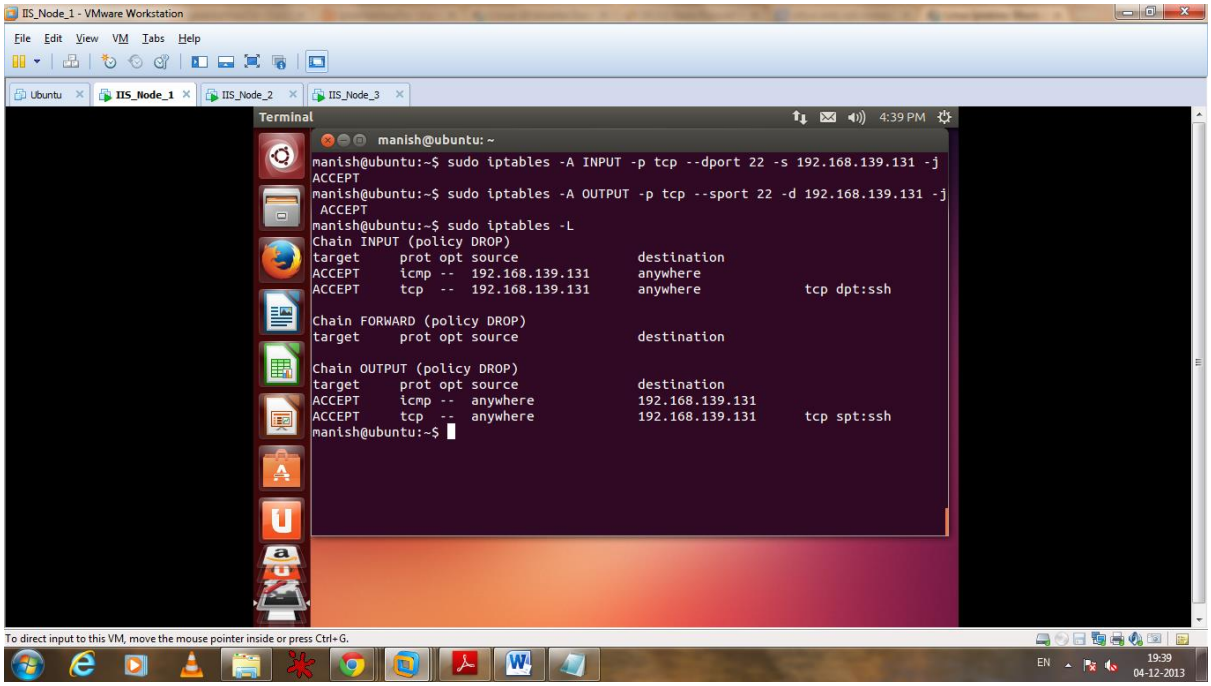
Screenshot 9

After that, rules were added in the INPUT and OUTPUT Chains on 'IIS_Node_1' to allow SSH connection from 'IIS_Node_2' (192.168.139.131) for protocol TCP and port 22(SSH). This rule should come before the drops rule otherwise this would be ignored.

Commands Used: 1. sudo iptables –A INPUT –p tcp --dport 22 –s 192.168.139.131 –j ACCEPT
2. sudo iptables –A OUTPUT –p tcp --sport 22 –d 192.168.139.131 –j ACCEPT

Here, -A specifies append to the rules in a particular chain (INPUT/OUTPUT)
-p specifies the protocol (tcp)
-dport specifies the destination port (22)
-sport specifies the source port (22)
-s specifies the source address
-d specifies the destination address
-j specifies jump to target ACCEPT

Screenshot 10: Rules added to allow SSH Connections from 'IIS_Node_2'



Screenshot 10

This is also the final firewall table on 'IIS_Node_1' which contains two rules under INPUT chain (Default – Policy DROP):

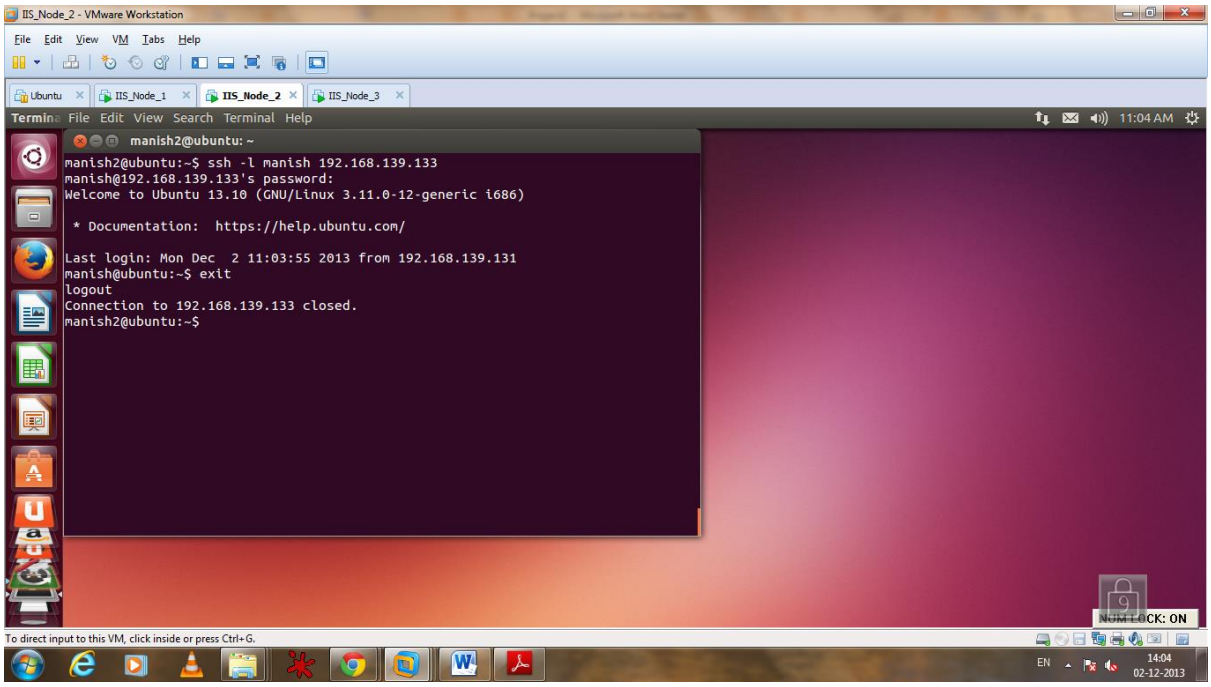
target	prot	opt	source	destination	
ACCEPT	icmp	--	192.168.139.131	anywhere	
ACCEPT	tcp	- -	192.168.139.131	anywhere	tcp dpt:ssh

And two rules under OUTPUT chain (Default – Policy DROP):

target	prot	opt	source	destination	
ACCEPT	icmp	--	anywhere	192.168.139.131	
ACCEPT	tcp	- -	anywhere	192.168.139.131	tcp spt:ssh

After this, SSH connection from 'IIS_Node_2' to 'IIS_Node_1' was tested which was successful.

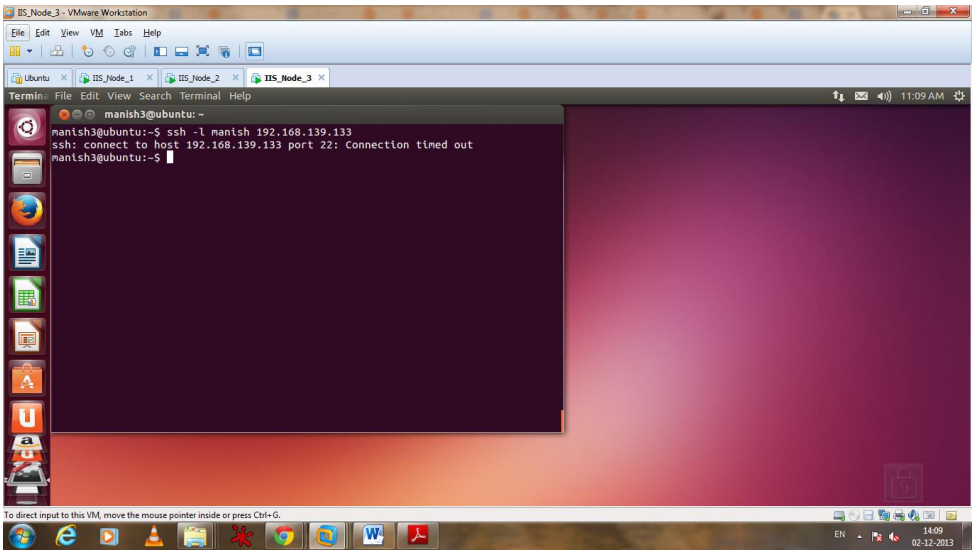
Screenshot 11: Successful SSH connection from 'IIS_Node_2' to 'IIS_Node_1'



Screenshot 11

And finally, SSH connection from 'IIS_Node_3' to 'IIS_Node_1' was tested which resulted in 'connection timed out'.

Screenshot 12: SSH Connection from 'IIS_Node_3' to 'IIS_Node_1' – Timed Out



Screenshot 12

So, finally, SSH connection was served only for 'IIS_Node_2'.

PART II

1) **What is the domain name requested by the client in the first DNS query?**
   The domain name requested by the client in the first DNS query is **syrianmalware.co**

2) **Were there any social networking sites contacted by this client? If so, list them.**
   There were 2 social networking sites to which this client contacted.
   A) **www.facebook.com**
      31.13.73.97
   B) **twitter.com**
      199.59.148.82, 199.59.150.7, 199.59.150.39

3) **What is the last HTTP object file that was downloaded by this client onto his local machine? Is the file malicious? Provide the link to VirusTotal that shows the analysis of this file.**

   The last HTTP request by this client was:
   http://syrianmalware.com/samples/185c8d11c0611cae7c81f4458bf1adea.zip

   The details of the get request are:
   Request datetime *2013-11-10 08:05:20.372092*
   Request user-agent *Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.57 Safari/537.36*
   Request referrer *http://syrianmalware.com/*
   Contacted host *208.113.163.195:80*
   Server response code *200*
   Response content
   sha256 *7d6e4fcdce01b32a0532d66348958e916af2ec664b41dd2301eb44994b6e1a0c*
   Response content file type *Zip archive data*

   This file was a zipped one with name '185c8d11c0611cae7c81f4458bf1adea.zip'. This zipped file contained an executable file 'ActiveX.exe' which was extracted with password 'infected'.

   The actual maliciousness of the file can be confirmed only after reverse engineering the binary as the results given by the anti-malware engines may be false positives.

   But, as so many anti-malware engines flagged the binary as infected, we can say that the executable is malicious.

Check with VirusTotal:

Analysis of the executable on VirusTotal resulted in 44 anti-malware engines (out of 48) flagging the executable as malicious. The link of the result is given below:
https://www.virustotal.com/en/file/cfdd3a78a895b3f49a39402eb28b0d2134cc3086849a41a6fdfe7d829a0d4dcd/analysis/1386014527/

Check with Jotti:

Analysis of the executable on Jotti resulted in 18 anti-malware engines (out of 22) flagging the executable as malicious. The link of the result is given below:
http://virusscan.jotti.org/en/scanresult/54ce76c4ce49e77216484a6f411fb36a8b924cb8/8ce1d57f120e99bd49843c55096912b942f915bc

4) Give the packet number of the GET request for this file.

The packet number for this get request is 1021.