UNIVERSITY OF TRENTO - Italy
Information Engineering
and Computer Science Department

STATEFUL FIREWALL

CISCO techniques against DoS/DDoS Attacks

Authors

JOSHUA TETTEH OCANSEY

SAMUEL NANA ESHUN

ACHILE SOH

DUC MANH HOANG

Supervisor

DR. LUCA ALLODI

TABLE OF CONTENT

STATEFUL FIREWALL

1.1 Introduction

One of the most basic firewall types used in modern networks is the stateful inspection firewall. This type of firewall has long been a standard method used by firewalls to offer a more in-depth inspection method over the previous packet inspection firewall methods. This paper will look at what a stateful firewall is and how it is used to secure a network while also offering better network usability and easier network firewall configuration with Cisco technology.

1.2 What is Stateful Firewall

Stateful firewall monitors the state of active connections and uses this information to determine which packets to be allowed through the firewall. Stateful Firewall does 2 basic functions namely:

- Dynamic Packet Filtering: it examines the session between the Internal and external networks and if a packet matches the firewall rules, it allows the packet to pass through the firewall but create or update its state table and also creating temporary ACL to allow the return traffic through.
- Packet Inspection: It offers more advanced inspection capacity by targeting vital packets for layer 7 examination such as packet connection initialization. It also inspects packet information from Layer 2 to layer 4.
- Some examples of stateful firewall in Cisco are Context Based Access Control CBAC, Network Based Access Recognition NBAR, Committing Access Rate CAR and Reflective Access Control List R-ACL. In this paper, we will focus more the Cisco technologies such as CBAC and CAR.
- Other types of Stateful firewall are Check point firewall and iptables.

1.3 How Stateful works

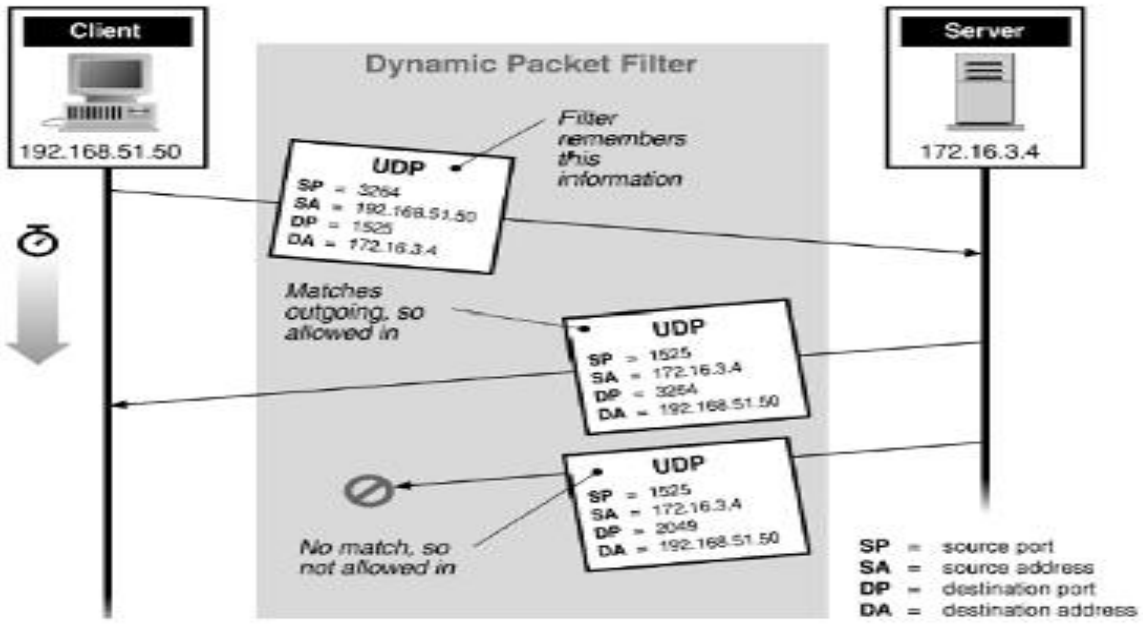


Fig 1: Demonstration of Stateful Firewall with UDP packets

- When a client from the internal network initializes a session with a server from the External network, the Firewall holds in memory the state of the connection: the IP addresses and ports numbers involved in the connection and the Sequence numbers of the packets traversing the connection.
- The firewall updates state-table with the above information and it also create temporary Access Control List in order to allow the return packet of the session to pass through the firewall.
- Once the session has ended, the entry in the state-table is discarded

1.4 Stateless verses Stateful

Stateless firewalls watch network traffic, and restrict or block packets based on source and destination addresses or other static values. They are not 'aware' of traffic patterns or data flows. Stateless firewalls are typically faster and perform better under heavier traffic loads.

Stateful firewalls can watch traffic streams from end to end. They are aware of communication paths and can implement various IP Security (IPsec) functions such as tunnels and encryption.

Stateful firewalls can tell what stage a TCP connection is in. Stateful firewalls are better at identifying unauthorized and forged communications.

CISCO FIREWALL

2.1 Introduction to Cisco Stateful Firewall

All traffic that goes through the firewall is inspected and is either allowed through or dropped. A simple packet filter can check for the correct source address, destination address, and ports, but it does not check that the packet sequence or flags are correct. A filter also checks every packet against the filter, which can be a slow process.

If it is a new connection, the firewall has to check the packet against access lists and perform other tasks to determine if the packet is allowed or denied. To perform this check, the first packet of the session goes through the "session management path," and depending on the type of traffic, it might also pass through. The session management path is responsible for the following tasks:

- Performing the access list checks
- Performing route lookups
- Allocating NAT translations.
- Establishing sessions in the "fast path”

If the connection is already established, the firewall does not need to recheck packets; most matching packets can go through the fast path in both directions. The fast path is responsible for the following tasks:

- IP checksum verification
- Session lookup
- TCP sequence number check
- NAT translations based on existing sessions
- Layer 3 and Layer 4 header adjustments

2.2 Reflexive Access Control List Firewall R-ACL

Reflexive ACLs enable IP packets to be filtered based on upper-layer session information. They are generally used in one of two ways:

- To allow outbound traffic out of an interface facing away from the internal network and filtering inbound traffic based on existing sessions originating inside the internal network

- To allow all inbound traffic to an interface facing toward the internal network and filtering outbound traffic based on the existing session originating inside the internal network

The former of these two is more typical with a network that does not utilize a demilitarized zone (DMZ). The latter is used to allow traffic into a DMZ but to not allow that traffic into the internal network without a previous connection initiated inside the internal network.

Reflexive ACLs can be defined only with extended named IP ACLs. They cannot be defined with numbered, standard named IP ACLs or with other protocol ACLs. Reflexive ACLs can be used in conjunction with other standard and static extended IP ACLs.

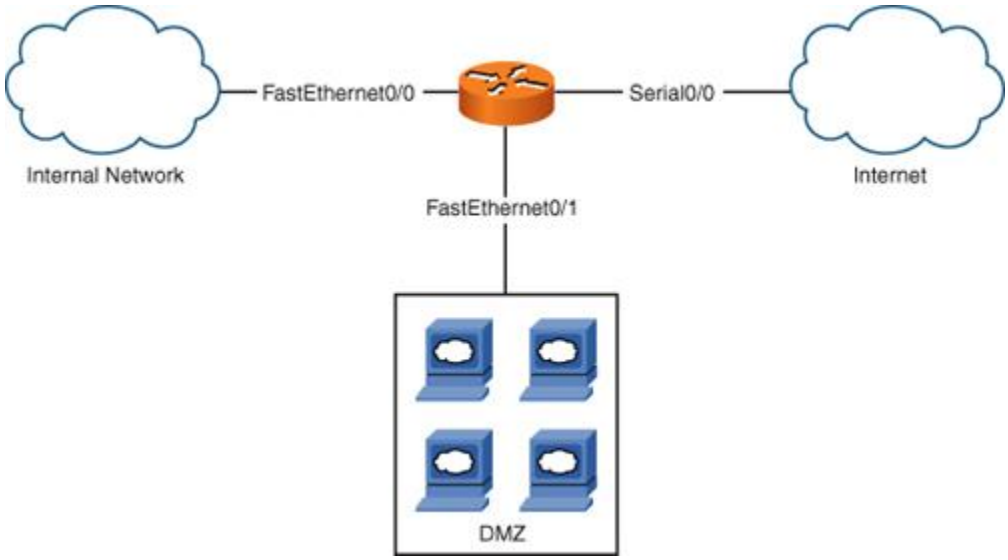


Fig 2: Topology to demonstrate Reflexive ACL

- router(config)# ip access-list extended incoming
- router(config-ext-nacl)# permit tcp any any reflect tcp-traffic
- router(config)# ip access-list extended outgoing
- router(config-ext-nacl)# evaluate tcp-traffic
- router(config)# interface FastEthernet0/0
- router(config-if)# ip address 172.16.1.1 255.255.255.0
- router(config-if)# ip access-group incoming in
- router(config-if)# ip accesss-group outgoing out

The commands demonstrate, by using **Fig 2** the process of permitting all TCP traffic inbound and outbound TCP traffic that was initiated from inside the network.

2.3 Context Based Access Control, CBAC

CBAC is a context-based firewall that performs the following:

- Inspects traffic in one direction for network, transport, and application layer information
- Extracts relevant port information
- Dynamically creates access list entries for return traffic
- Closes ports at the end of a connection

2.3.1 Traffic Filtering

CBAC intelligently filters TCP and UDP packets based on application-layer protocol session information. You can configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall, and CBAC can be used for intranet, extranet, and Internet perimeters of your network.

Without CBAC, traffic filtering is limited to access list implementations that examine packets at the network layer, or at most, the transport layer. However, CBAC examines not only network layer and transport layer information but also examines the application-layer protocol information (such as FTP connection information) to learn about the state of the session. This allows support of protocols that involve multiple channels created as a result of negotiations in the control channel. Most of the multimedia protocols, as well as some other protocols (such as FTP, RPC, and SQL*Net), involve multiple channels.

Using CBAC, Java blocking can be configured to filter HTTP traffic based on the server address or to completely deny access to Java applets that are not embedded in an archived or compressed file. With Java, you must protect against the risk of users inadvertently downloading destructive applets into your network. To protect against this risk, you could require all users to disable Java in their browser. If this is not an acceptable solution, you can create a CBAC inspection rule to filter Java applets at the firewall, which allows users to download only applets residing within the firewall and trusted applets from outside the firewall. For extensive content filtering of Java,

Active-X, or virus scanning, you might want to consider purchasing a dedicated content filtering product.

2.3.2 Traffic Inspection

CBAC inspects traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions. Inspecting packets at the application layer, and maintaining TCP and UDP session information, provides CBAC with the ability to detect and prevent certain types of network attacks such as SYN-flooding. CBAC helps to protect against DoS attacks in other ways. CBAC inspects packet sequence numbers in TCP connections to see if they are within expected ranges—CBAC drops any suspicious packets. You can also configure CBAC to drop half-open connections, which require firewall processing and memory resources to maintain. Additionally, CBAC can detect unusually high rates of new connections and issue alert messages.

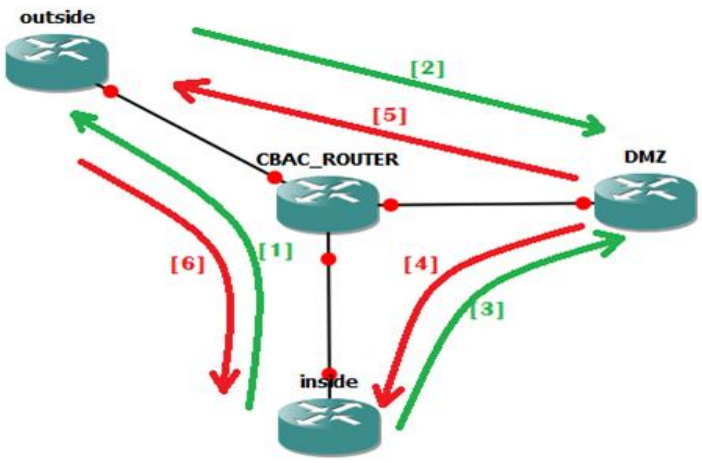


Fig 3: below demonstrate how dynamic packet filtering of CBAC works

- **[1]**Allow hosts on the inside (LAN) to have access to any host in the outside area (Internet).
- **[2]** Allow anyone on the Internet to connect to specific hosts and ports on the DMZ.

- **[3]** Allow inside users to connect to specific hosts and ports on the DMZ.
- **[4]** Deny any connection from hosts in the DMZ to the LAN.
- **[5]** Deny any connection from hosts in the DMZ to the Internet.
- **[6]** Deny any connection from the Internet to the LAN.

2.4 Committed Access Rate, CAR

Committed access rate (CAR) is a feature from Cisco that is used in network optimization and security. It limits the input or output traffic rate on an interface or sub-interface based on criteria such as IP precedence, IP access list or incoming interface. When the traffic reaches the set limit, CAR specifies certain actions to be carried out. These actions can be configured using CAR commands that use the values of traffic rate limit, burst rate allowed and the action to be performed when the traffic reaches or exceeds the set limit

CAR enables you to limit traffic entering or leaving an interface, and it can match on any of the following criteria: all IP traffic, IP precedence value, MAC address, or information that matches a permit statement in either a standard or an extended ACL. CAR typically is implemented at the perimeter of your network for egress traffic, and it enables you to have different rate-limiting policies for different types of traffic. For example, you might have different rate limits for ICMP traffic compared to HTTP traffic to a web server.

The configuration of CAR is done on a router's interface with the rate-limit command. This command specifies the rate policy to be used for the matching traffic. The input and output parameters specify the direction in which CAR should be performed. Three rate functions are defined in your CAR configuration:

- The average rate, specified in bits per second (bps), for the matching traffic. This is measured by a long-term average of the transmitted rate of traffic on the interface. Traffic under this rate is considered to be conforming.
- The normal burst size, specified in bits per second (bps). This determines how long traffic can burst above the average rate before it is considered nonconforming.
- The excessive burst rate, specified in bits per second (bps). Traffic that exceeds the excessive burst rate is considered nonconforming.

Below is an example of using CAR to mitigate ICMP flood and Smurf attacks on an ISP interface

- ISP(config)# access-list 100 permit icmp any any echo
- ISP(config)# access-list 100 permit icmp any any echo-reply
- ISP(config)# interface serial0
- ISP(config-if)# rate-limit output access-group 100 64000 4000 4000 conform-action transmit exceed-action drop

DOS/DDOS ATTACKS

In a denial-of-service (DoS) attack, an attacker attempts to prevent legitimate users from accessing information or services. By targeting your computer and its network connection, or the computers and the network of the sites you are trying to use, an attacker may be able to prevent you from accessing email, websites, online accounts (banking, etc.), or other services that rely on the affected computer.

Examples of DOS attacks are ICMP flood attack and SYN flood attack

3.1 ICMP Flood Attack

An ICMP request requires the server to process the request and respond, so it takes CPU resources. Attacks on the ICMP protocol, including smurf attacks, ICMP floods, and ping floods take advantage of this by inundating the server with ICMP requests without waiting for the response. This attack seeks to overwhelm the server's ability to respond, thereby blocking valid requests.

Smurf attack is one specific form of a flooding DoS attack that occurs on the public Internet. It solely depends on incorrect configuration network equipment that permit packets that are supposed to be sent to all hosts of a computer on a specific network, not via any machine but only via network’s broadcast address. Then the network actually works or serves as a smurf amplifier. In an attack like this, the killers or the perpetrators will send IP packets in huge number displaying the fake source address as to show that it is a victim’s address. The bandwidth of the network is consumed quite quickly, and it also stops legal packets from reaching their destination.

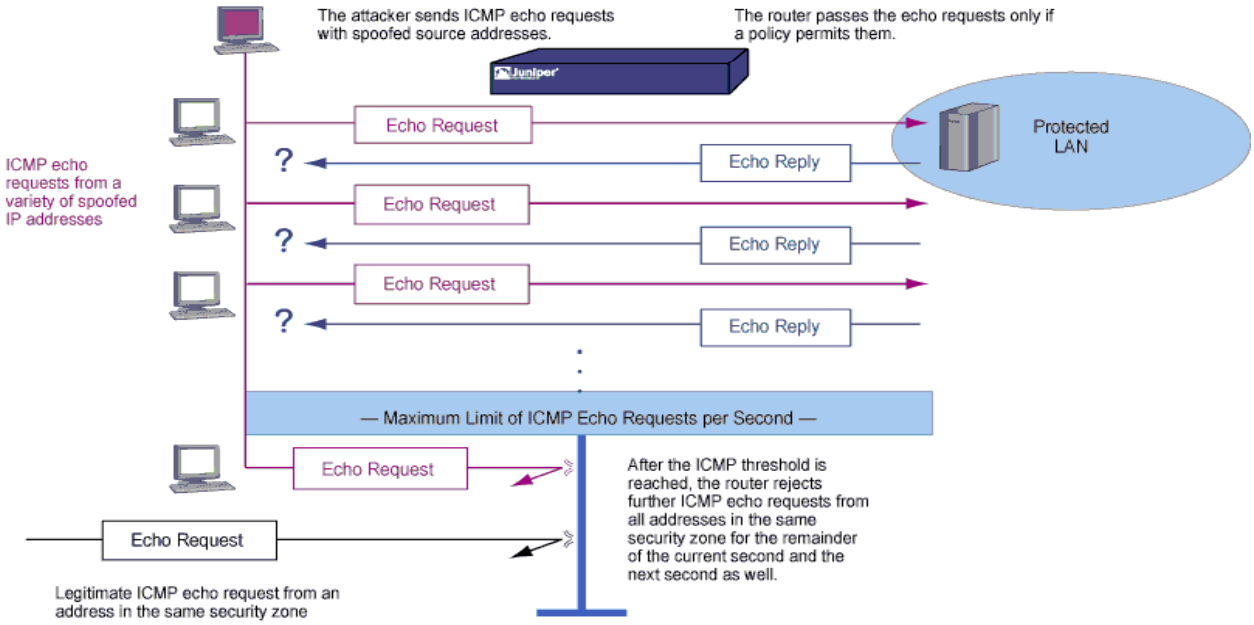


Fig 4: demonstrate ICMP flood attack

3.2 SYN Flood Attack

SYN flood is a result of TCP/SYN packets flooding sent by hosts, mostly with a fake address of the sender. The handling of these packets is done in the same manner as connection request, which makes the server produce a semi-open connection, as it sends TCP/SYN-ACK packet back (Approve/Acknowledge), and waits for a packet to be received as a response from the address of a sender (ACK Packet's response). Actually, the sender never responds as his address is not real. The saturation of available connections takes place by the semi-open connections that the server can actually make so that it cannot respond to legal requests even after the attack is over.

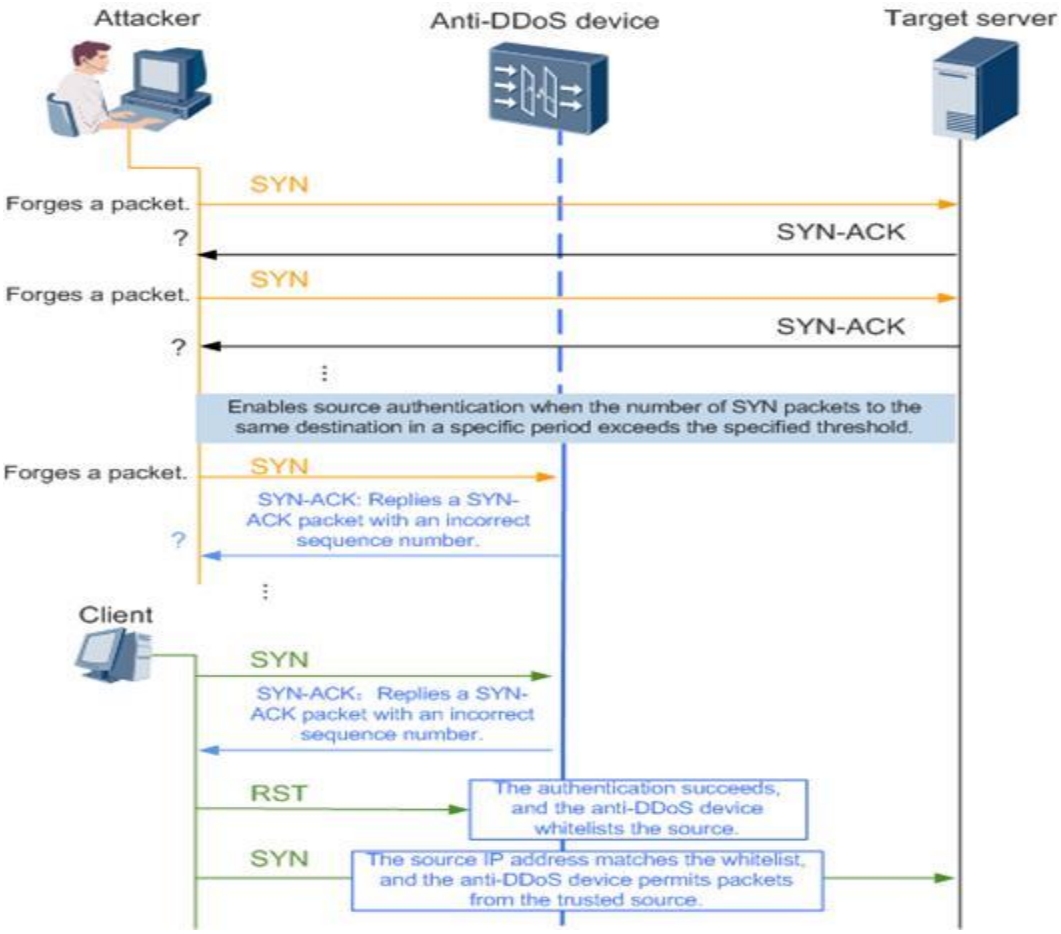


Fig 5: below demonstrate SYN Flood Attack

LABORATORY SETUP AND ACTIVITIES

4.1 Laboratory Objectives

The main objective of the lab session is to demonstrate how an attacker can launch a DOS attack and how a Network Security Engineer with a help with GNS3 simulator can exploit Cisco technologies to mitigate ICMP flood and SYN flood attacks. The activities consist of the following:

- Demonstration of ICMP flood Attack
- Demonstration of SYN Flood Attack
- Reconnaissance or Vulnerability Scanning
- ICMP mitigation techniques with CBAC
- ICMP mitigation technique with CAR
- SYN mitigation technique with CBAC
- SYN mitigation technique with CAR

Some of the tools exploited for the lab activities are GNS3 Emulator, Virtualbox, XOIC, Wireshark, ZenMap, tcpdump, Cacti bandwidth measurement and Capsa Network Analysis.

4.2 Graphical Network Simulator GNS3 Emulator

GNS3 is a Graphical Network Simulator that allows emulation of complex networks. GNS3 allows the emulation of Cisco Internetwork Operating Systems. It allows you to run a Cisco IOS in a virtual environment on your windows or Linux computers. GNS3 takes this a step further by providing a graphical environment for a user to create network topology

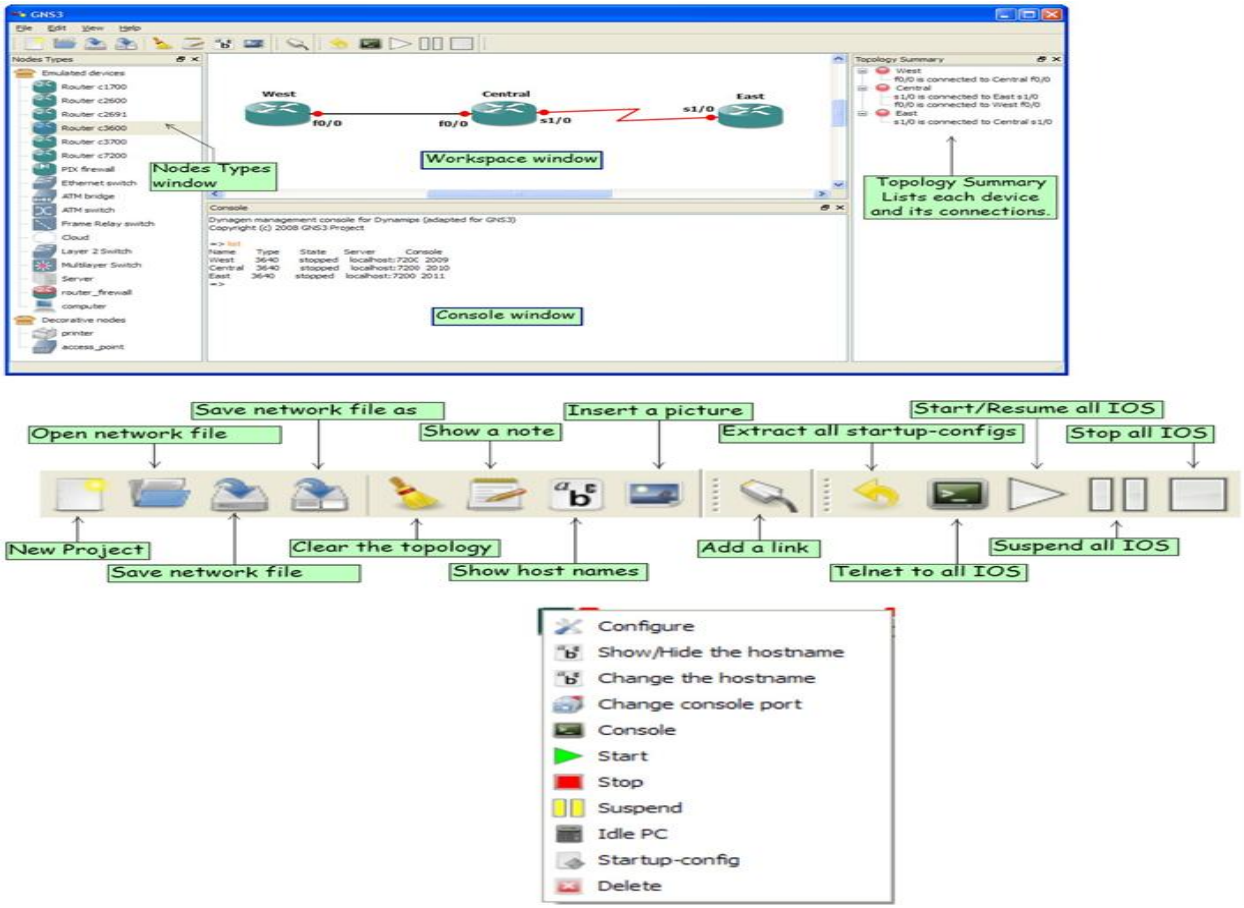


Fig6: below shows dashboard of GNS3 software and its features.

4.3 Laboratory Topology

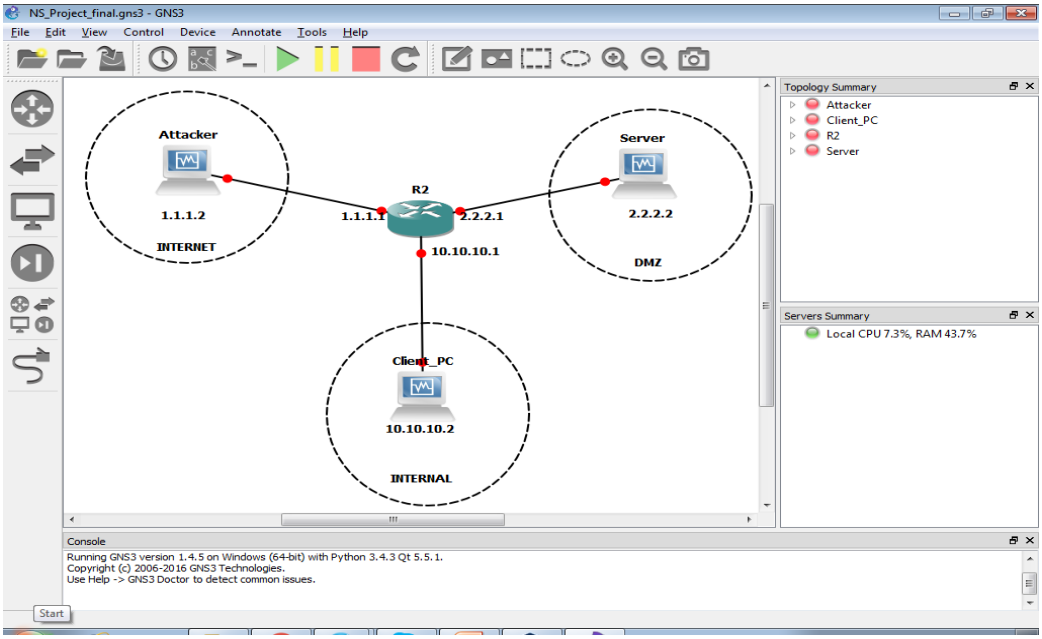


Fig 7: Lab topology designed with GNS3.

The Cisco router is configured to be connected to three VM computers such as Attacker (Windows), Client (Windows) and Server(Linux). The Router is configured with the following commands in Cisco.

Configure IP address for INTERNET interface

- R1#configure terminal
- R1(config)#interface fastEthernet 0/0
- R1(config-if)#ip address 1.1.1.1 255.255.255.0
- R1(config-if)#no shutdown
- R1(config-if)#exit

Configure IP address for DMZ interface

- R1(config)#interface fastEthernet 0/1
- R1(config-if)#ip address 2.2.2.1 255.255.255.0
- R1(config-if)#no shutdown
- R1(config-if)#exit

Configure IP address for INTERNAL interface

- R1(config)#interface fastEthernet 2/0
- R1(config-if)#ip address 10.10.10.1
- R1(config-if)#ip address 10.10.10.1 255.255.255.0
- R1(config-if)#no shutdown
- R1(config-if)#exit

Configure Router as DNS server and setup DNS host

- R1(config)#ip dns server
- R1(config)#ip host ubuntu.com 2.2.2.2
- R1(config)# end

4.3 Simple ICMP Flood Attack with XOIC

From the topology in Fig 7 above, we login into the 'Attacker' VM and then from its Desktop we opened XOIC software to launch an ICMP flood attack to the 'Server'. XOIC is DOS tool used to send multiple and incessant traffic to its victim.

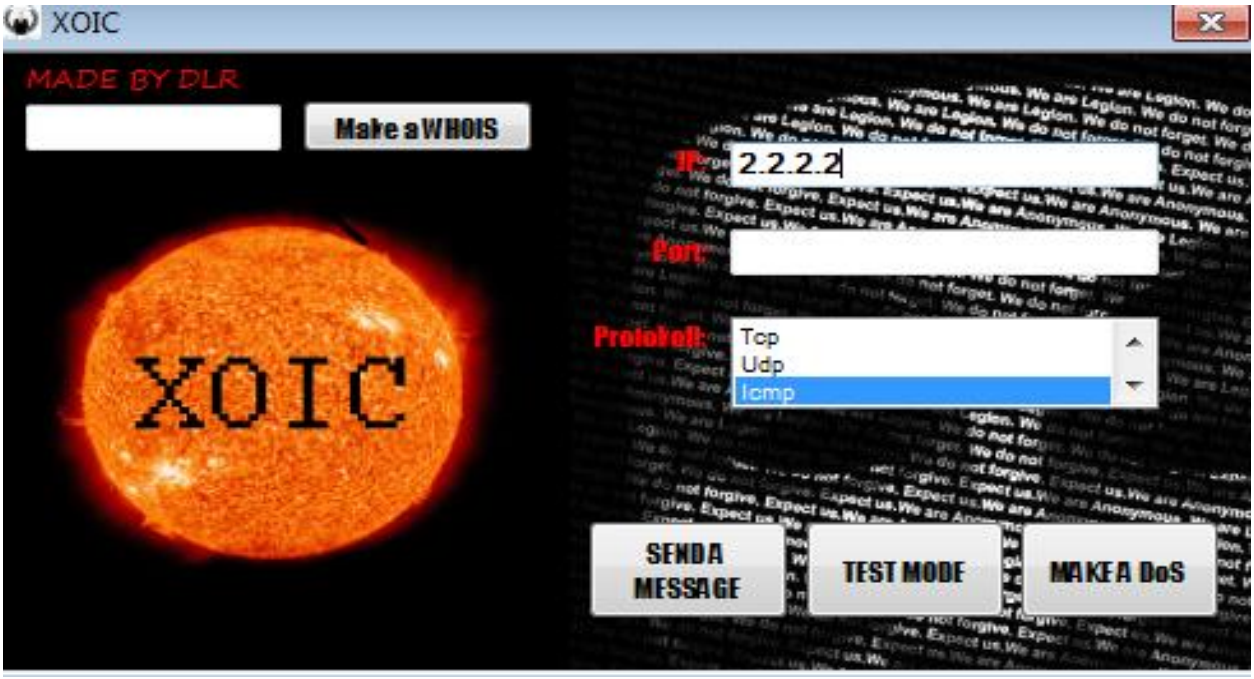


Fig 8: ICMP Flood Attack with XOIC software.

4.3.2 Verify ICMP Flood Attack

- Right-click the server node on the GNS3 topology and click on 'start capture' to monitor and observe the traffic on the wireshark
- Login into the Server computer, open terminal (crt + Alt + T). Type sudo tcpdump –i eth0 to observe the attack to the server

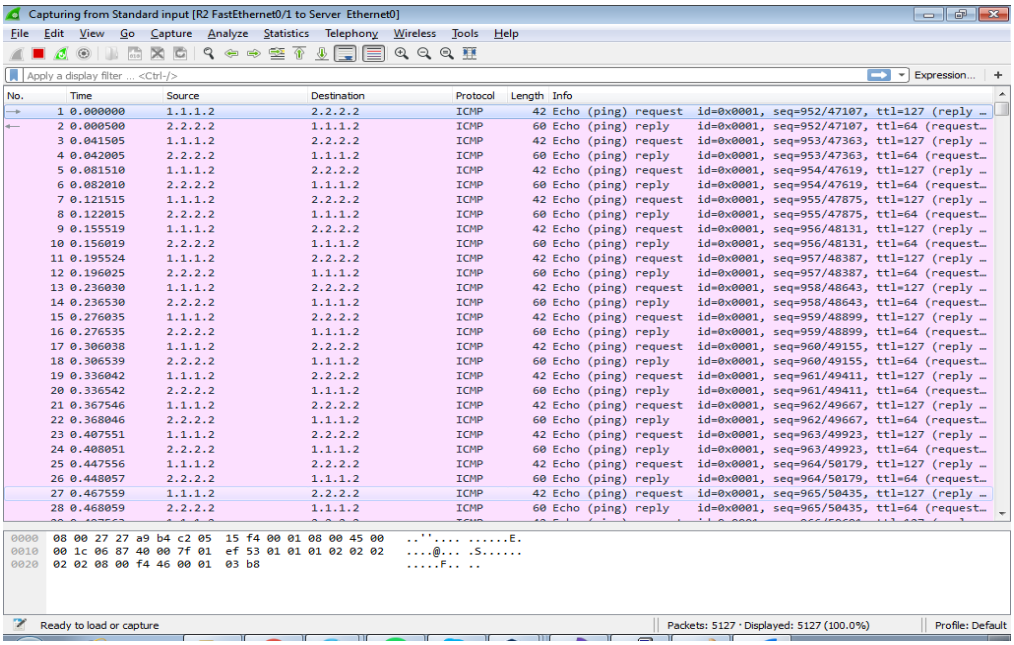


Fig 9: Wireshark Verification of ICMP flood Attack.

4.4 SYN Flood Attack

4.4.1 Reconnaissance with nmap-zenmap

We assume the ‘Attacker’ is unaware of the information concerning the victim’s Network except its hostname of the ‘Server’.

The ‘Attacker’ use nmap to scan the victim’s network to understand the topology of network and services

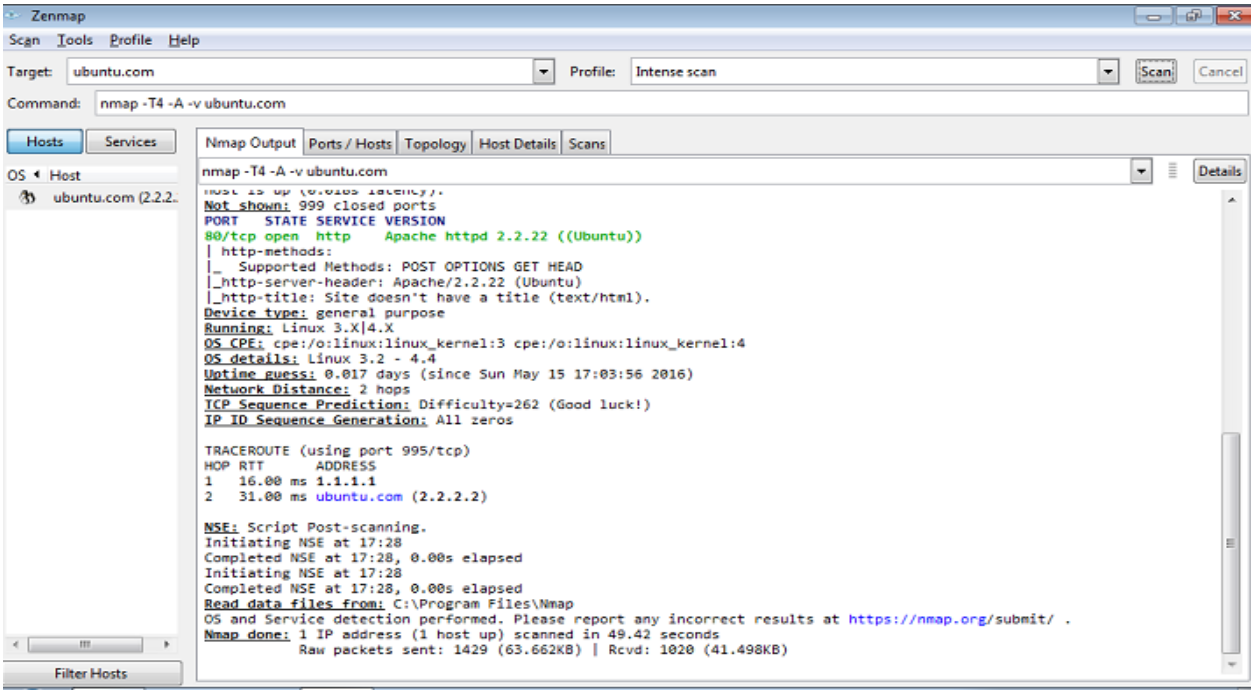


Fig 10: Zenmap result after scanning the network.

4.4.2 SYN Flood Attack to http server

From Fig 10, we can observe that http port 80 is opened on address 2.2.2.2. The 'Attacker' uses this information to send open-end syn packets to the 'Server'.

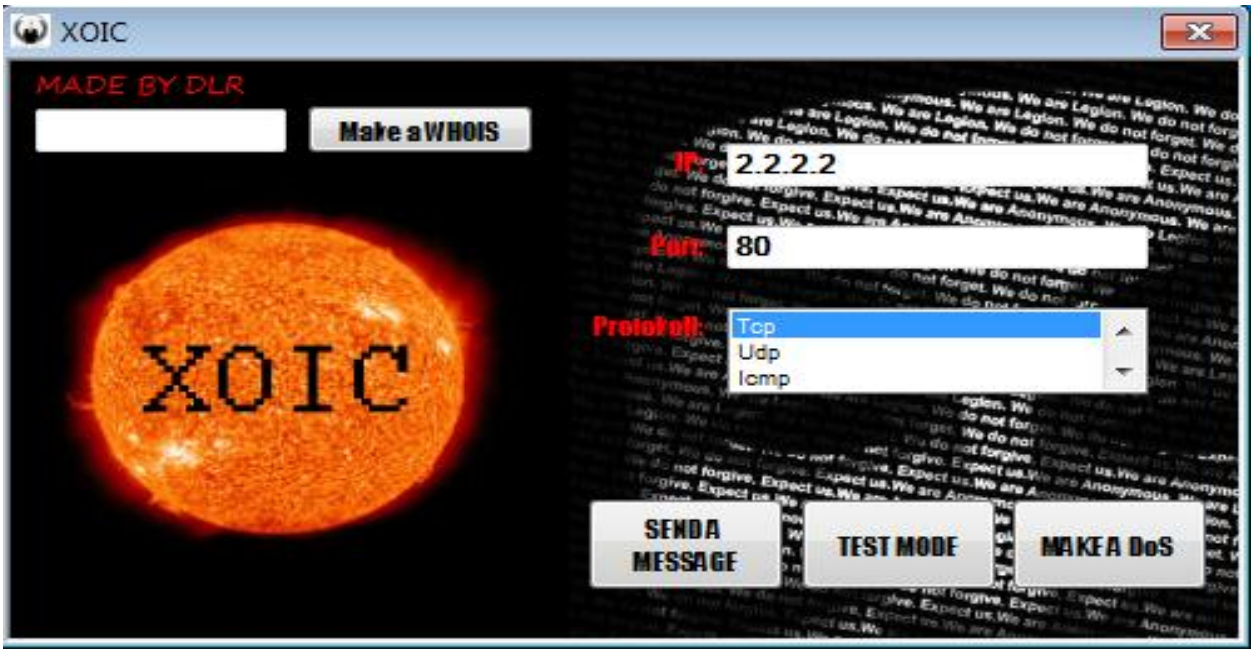


Fig 11: Syn flood Attack to http server.

4.4.3 Verify SYN Flood Attack

- Right-click the server node on the GNS3 topology and click on ‘start capture’ to monitor and observe the traffic on the wireshark
- Login into the Server computer, open terminal (crt + Alt + T). Type sudo tcpdump –i eth0 to observe the attack to the server

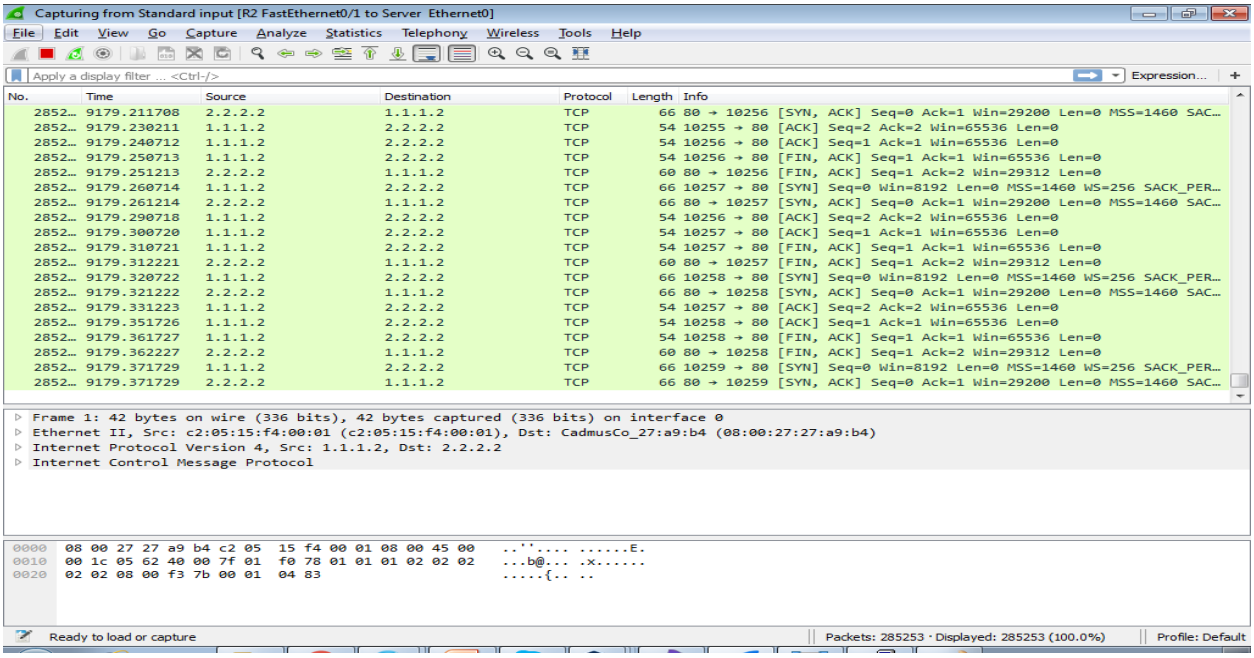


Fig 12: SYN Flood Verification with wireshark.

4.5 Context Based Access Control CBAC Firewall

Context-Based Access Control CBAC --- CBAC enables dynamic modification of access lists to allow certain incoming flows by first inspecting and recording flows initiated from the protected internal network. Apart from L2 to L4 CBAC is able to inspect all the way to the application layer, taking into consideration characteristics of a flow on a per-protocol basis (or context)

4.5.1 CBAC technique Against ICMP Flood Attack

On the Router console we execute these commands:

- Router(config)#access-list 100 deny icmp any any
- Router(config)#access-list 100 permit ip any any
- Router(config)#ip inspect name GROUP11 icmp

- Router(config)#interface fastEthernet 0/0
- Router(config-if)#ip access-group 100 in
- Router(config-if)#ip inspect GROUP11 out

The above commands deny only ICMP traffic from entering the Firewall while allowing dynamically allowing the ICMP packets initiated from the internal network access through from the external interface fast Ethernet 0/0.

4.5.2 Verify CBAC against ICMP flood Attack

From the Router console, we type the following commands

- Router# show ip access-lists 100
  - ❖ Extended IP access list 100
  - ❖ 10 deny icmp any any (4816 matches)
  - ❖ 20 permit ip any any (231 matches)

- Router# show ip inspect all
  - ❖ Session audit trail is disabled
  - ❖ Session alert is enabled
  - ❖ Inspection Rule Configuration
  - ❖ Inspection name GROUP11
  - ❖ icmp alert is on audit-trail is on timeout 10
  - ❖ Established Sessions
  - ❖ Session 63F056B8 (2.2.2.2:8)=>(1.1.1.2:0) icmp SIS_OPEN

The first command ***show ip access-list 100*** shows the number of ICMP packets drop by the firewall and other protocol allowed with a period.
The second command ***show ip inspect all*** shows CBAC configuration and traffic inspection on ICMP that though ICMP traffic is a block from the external interface any return ICMP traffic initiated by the internal host (trusted network) has been allowed access. For example, there has been established ICMP connection between host 2.2.2.2 to external address 1.1.1.1.

4.5.3 CBAC technique Against SYN Flood Attack

Again on the router console we execute these commands

- Router # Configure terminal
- Router(config)# access-list 199 permit tcp any host 2.2.2.2 eq www established
- Router(config)# ip inspect name FIREWALL http
- Router(config)# interface fastEthernet 0/0
- Router(config-if)# ip inspect FIREWALL out
- Router(config-if)#ip access-group 199 in
- Router(config-if)# end

The above commands allows only established http traffic and drop incomplete http packets and also dynamically allow http traffic through to firewall when the traffic is initiated from the within the internal network.

4.5.4 Verify CBAC against SYN Flood Attack

- Router# ***show ip access-lists 199***

    - ❖ Extended IP access list 199
    - ❖ 10 permit tcp any host 2.2.2.2 eq www established (4 matches)

- Check from message from XOIC (DOS software) on the ‘Attacker’ VM

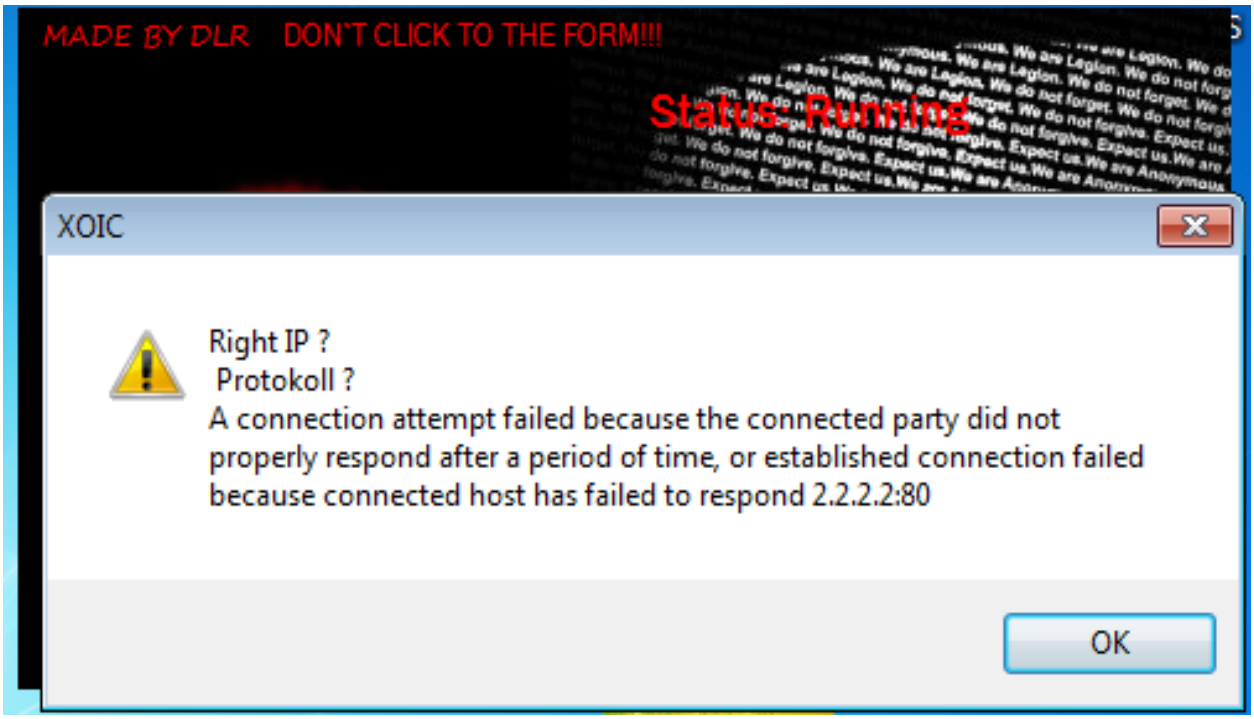


Fig 13: the inscription on the XOIC shows that CBAC has been successful in blocking incomplete SYN packets from the attacker.

4.6 Committed Access Rate, CAR

Committed Access Rate CAR --- CAR limits traffic rate entering and leaving an interface with either of the following matches: ip address, ip precedence value, MAC address or any information that matches a permit or deny statement in Access Control List, ACL

4.6.1 Network Traffic Analysis

Before implementing CAR, make sure that you have a baseline of traffic flowing through the router under normal circumstances, categorized by traffic type and destination addresses. do not choose arbitrary rate-limiting values for the rate-limit command for the TCP SYN setup of the web connections. Before you define any limits, you should understand your traffic patterns: Putting in a value that is too small or too large might create additional problems.

In order to understand your traffic, you can use Network Traffic analysis tools such as Capsa Free Network Analyzer and Cacti graphing Solution. While cacti provide you with the measure of outbound and inbound traffic within a period Capsa Network Analyzer gives you detail on the type of traffic on both inbound and outbound directions.

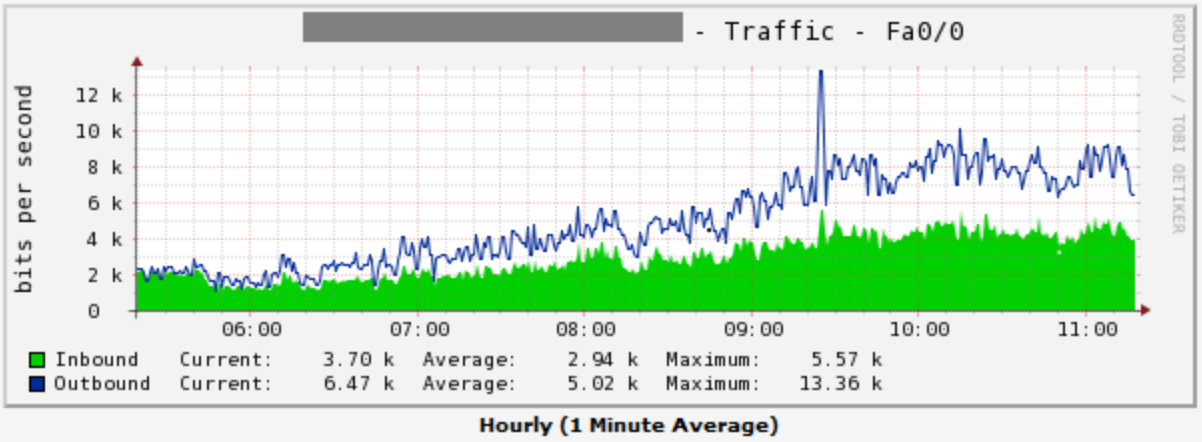


Fig 14: Traffic measure with Cacti graphing

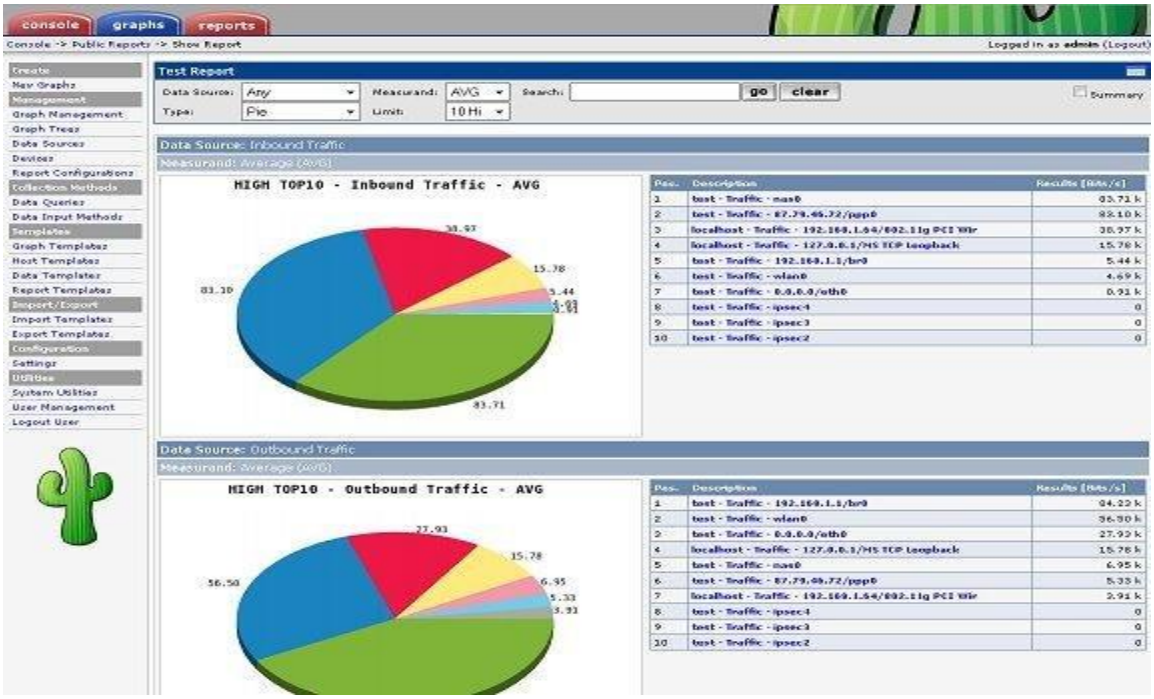


Fig 15: Shows traffic in terms of pie chart on Cacti graphing solution.

Fig 16: Show different types of traffic measure in Bar chart using Capsa Network Analyzer

4.6.2 CAR technique Against ICMP Flood Attack

On the router console, we execute these commands

Router # configure terminal

- Router(config)# access-list 150 permit icmp any any echo
- Router(config)# access-list 150 permit icmp any any echo-reply
- Router(config)# interface fastEthernet 0/0
- Router(config-if)# rate-limit input access-group 150 8000 1500 2000 conform-action transmit exceed-action drop
- Router(config-if)# end

The above commands set ICMP echo and echo replies limits to 8kbps of bandwidth, with a burst size of 2 kbps of bandwidth. When it is with 1.5kbps to 2kbps it will transmit but when it exceeds the burst size it drops the traffic.

4.6.3 Verify CAR against ICMP Flood Attack

On the privilege mode of the router console we execute the command below

Router# show interfaces fastEthernet 0/0 rate-limit

- FastEthernet0/0 Output
    - matches: access-group 150
    - params: 8000 bps, 1500 limit, 2000 extended limit
    - conformed 15272 packets, 678776 bytes; action: transmit
    - exceeded 715 packets, 30590 bytes; action: drop
    - last packet: 20ms ago, current burst: 1438 bytes
    - last cleared 00:11 ago, conformed 8000 bps, exceeded 0 bps

4.7 CAR technique against SYN Flood Attack

On the router, Console execute these commands

Router # Configure terminal

- Router(config)#access-list 120 deny tcp any host 2.2.2.2 eq www established
- Router(config)#access-list 120 permit tcp any host 2.2.2.2 eq www
- Router(config)#interface fastEthernet 0/0
- Router(config-if)#rate-limit input access-group 120 8000 1500 2000 conform-action transmit exceed-action drop
- Router(config-if)#end

In this Lab example, CAR uses the ***established*** keyword to block out all http TCP packets except TCP SYN, which is used to establish connections, from being rate limited. Also, it restricts the TCP SYNs from any attacking host to the 2.2.2.2 server; another traffic sent to other internal servers is not rate limited by this configuration.

4.8 Verify CAR against SYN Flood Attack

. Router# ***show interfaces fastEthernet 0/0 rate-limit***

- FastEthernet0/0 Output
- matches: access-group 120
- params: 8000 bps, 1500 limit, 2000 extended limit
- conformed 293 packets, 18166 bytes; action: transmit
- exceeded 28 packets, 1668 bytes; action: drop
- last packet: 179948ms ago, current burst: 1520 bytes

- last cleared 00:22:50 ago, conformed 0 bps, exceeded 0 bps

The above result shows that at least 28 packets were drop when they did not conform to the CAR rules.

APPENDIX A: CISCO Basic Commands

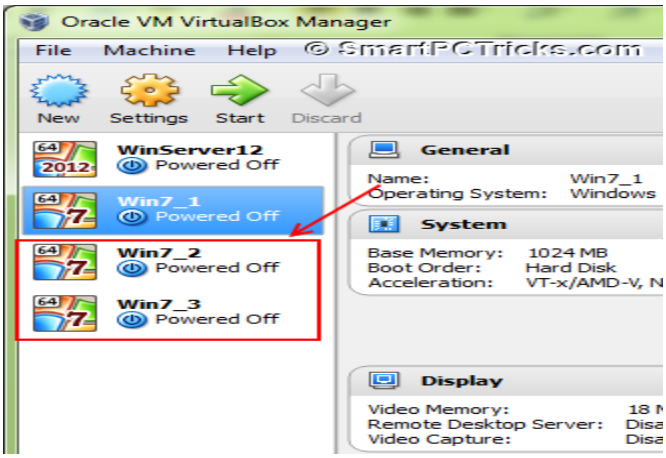
Requirement	Cisco Command
Set a console password to cisco	Router(config)#**line con 0** Router(config-line)#**login** Router(config-line)#**password cisco**
Set a telnet password	Router(config)#**line vty 0 4** Router(config-line)#**login** Router(config-line)#**password cisco**
Stop console timing out	Router(config)#**line con 0** Router(config-line)#**exec-timeout 0 0**
Set the enable password to cisco	Router(config)#**enable password cisco**
Set the enable secret password to peter. This password overrides the enable password and is encrypted within the config file	Router(config)#**enable secret peter**
Enable an interface	Router(config-if)#**no shutdown**
To disable an interface	Router(config-if)#**shutdown**
Set the clock rate for a router with a DCE cable to 64K	Router(config-if)**clock rate 64000**
Set a logical bandwidth assignment of 64K to the serial interface	Router(config-if)**bandwidth 64** **Note that the zeroes are not missing**
To add an IP address to a interface	Router(config-if)#**ip addr 10.1.1.1**

	255.255.255.0
To enable RIP on all 172.16.x.y interfaces	Router(config)#**router rip** Router(config-router)#**network 172.16.0.0**
Disable RIP	Router(config)#**no router rip**
To enable IRGP with a AS of 200, to all interfaces	Router(config)#**router igrp 200** Router(config-router)#**network 172.16.0.0**
Disable IGRP	Router(config)#**no router igrp 200**
Static route the remote network is 172.16.1.0, with a mask of 255.255.255.0, the next hop is 172.16.2.1, at a cost of 5 hops	Router(config)#**ip route 172.16.1.0 255.255.255.0 172.16.2.1 5**
Disable CDP for the whole router	Router(config)#**no cdp run**
Enable CDP for he whole router	Router(config)#**cdp run**
Disable CDP on an interface	Router(config-if)**#no cdp enable**

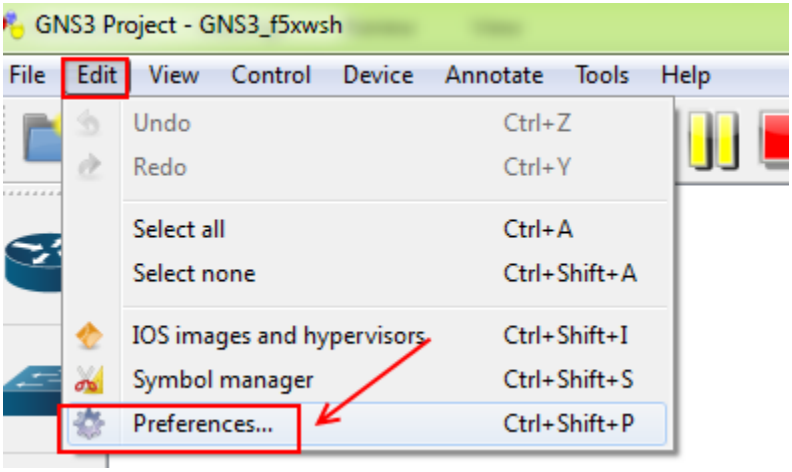
APPENDIX B: How to Connect VIrtualBox to GNS3

Step 1: Setting up of VMs

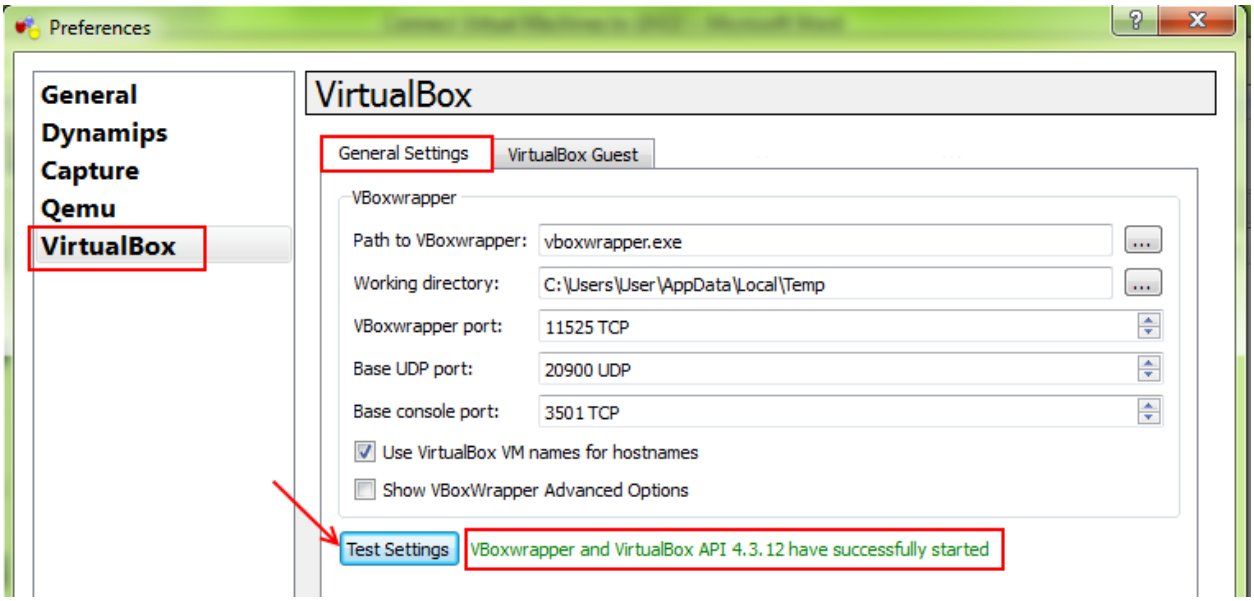
Make sure you installed VirtualBox VMs, for us, 4 VMs were installed. Among these, we will connect Win7_1 and Win7_2 to GNS3.



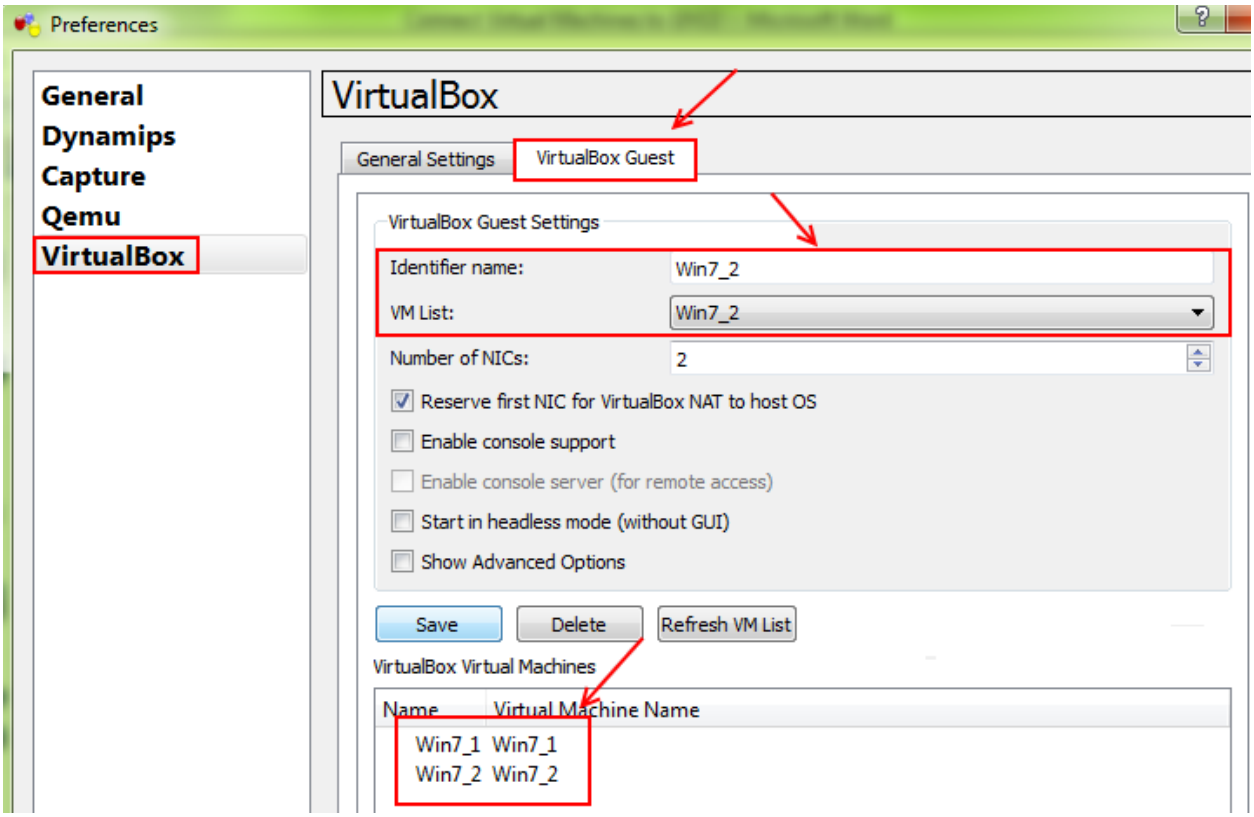
Step 2: Run GNS3, Edit → Preferences → VirtualBox



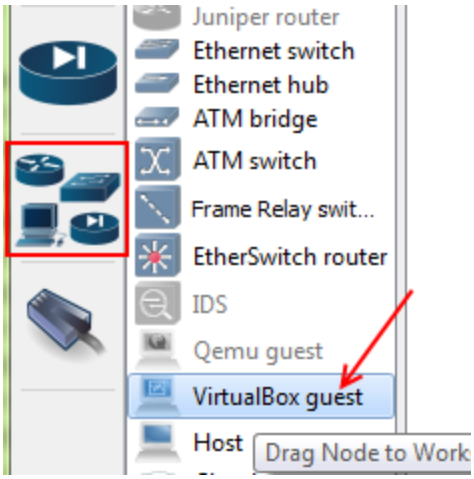
In the General Settings tab click ‘Test Settings’ button, you will see a message ‘VBoxwrapper and VirtualBox API have successfully started’.



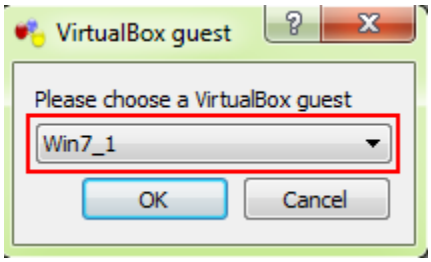
Step 3:Select 'VirtualBox Guest' tab, you may find all VMs in the VM List. Select suitable one after setting an Identifier Name. Set required VMs and the click Save button.



Step 4: Browse VirtualBox guest to the work space.

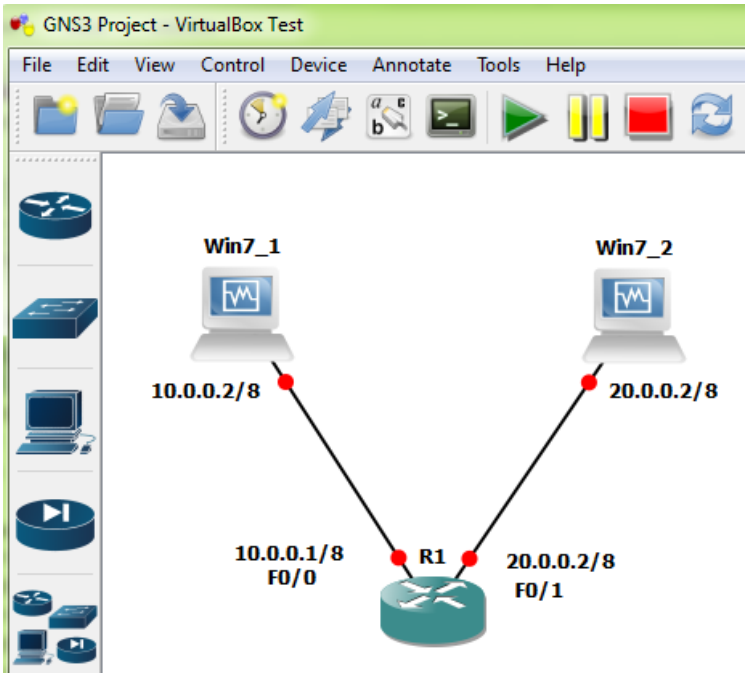


If you have more than one VM, you will be asked to select one among those.



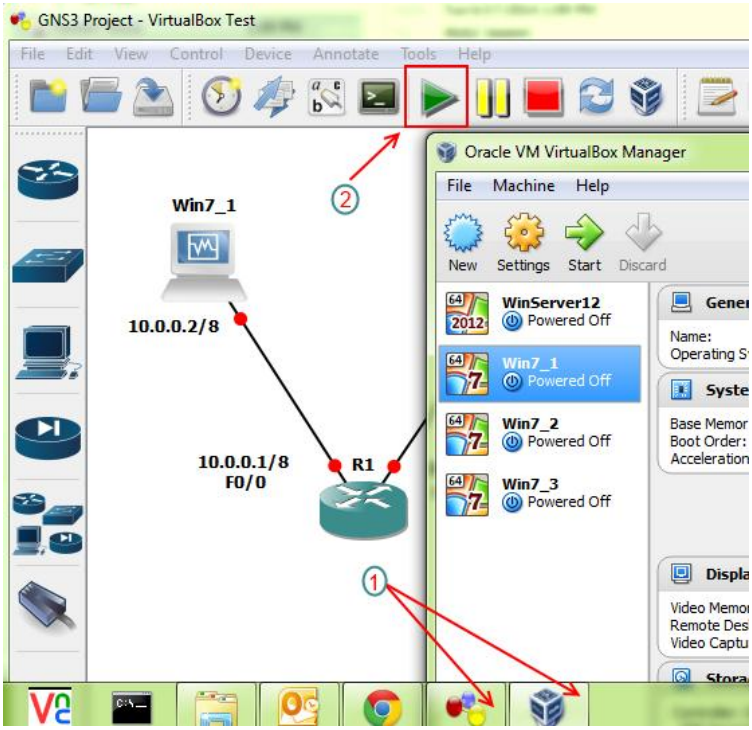
Step 5: GNS3 Network scenario with VirtualBox

Build a simple scenario with VirtualBox guest. For us, we did with two VMs and one router.



Step 6: Run Simulation

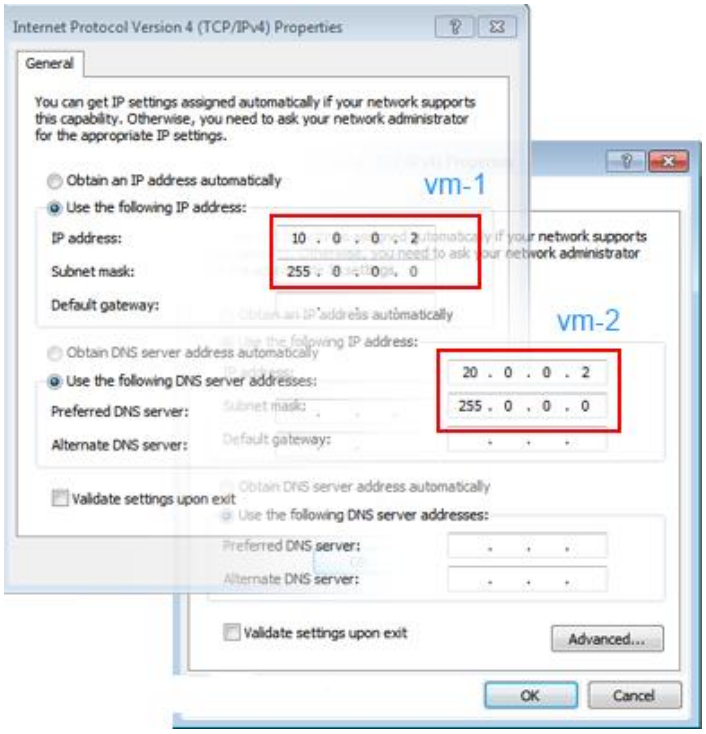
Now let's run our Network. Before running the simulation open VirtualBox side by side with GNS3. [Only open VirtualBox, do not run any VMs]. Then click Start button to begin the simulation.



After starting the simulation we can see all of the VMs that you added in the example scenario will run automatically.

Step 7: Set Static IP for VMs

Once the VMs are started change the IP of virtual machines. In our example, we assigned 10.0.0.2/8 and 20.0.0.2/8 to the VMs



Step 8: Cisco Router Configuration for VirtualBox Interface.

Now configure the router and assign IP address to each port. Configuration commands are followed.

R1#***configure terminal***

- R1(config)#interface fastEthernet 0/0
- R1(config-if)#ip address 10.0.0.1 255.0.0.0
- R1(config-if)#no shutdown
- R1(config-if)#exit
- R1(config)#interface fastEthernet 0/1
- R1(config-if)#ip address 20.0.0.1 255.0.0.0
- R1(config-if)#no shutdown

Step 9: Ping test and verification

Try to ping from the router to the VMs and vice versa.

```
R1#
R1#ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
R1#ping 20.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
R1#
```

BIBLIOGRPAHY

http://www.tomax7.com/mcse/cisco_routerconfig.htm

http://www.ciscopress.com/articles/article.asp?p=345618&seqNum=5

http://www.ciscopress.com/articles/article.asp?p=345618&seqNum=4

http://resources.intenseschool.com/cisco-ios-firewall-stateful-gns3-lab-context-based-access-control-cbac-pt-iv/

http://www.ciscopress.com/articles/article.asp?p=1716288&seqNum=2

https://www.techopedia.com/definition/31001/committed-access-rate-car

https://www.sans.org/reading-room/whitepapers/firewalls/cbac-cisco-ios-firewall-feature-set-foundations-806

https://micronicstraining.com/cbac-ios-firewall/

http://www.techrepublic.com/blog/data-center/control-unwanted-traffic-on-your-cisco-router-with-car-96521/

http://www.smartpctricks.com/2014/06/connect-gns3-to-virtualbox.html

https://engineering.purdue.edu/dcsl/reading/2005/Stateful_Firewalls.pdf

http://docstore.mik.ua/orelly/networking_2ndEd/fire/ch08_01.htm

http://blog.pluralsight.com/gns3

https://www.youtube.com/watch?v=hRshwowRW7w

https://www.youtube.com/watch?v=l6BjEIin2Jw&list=PL86F49AB06DAF09C6&index=3

https://www.youtube.com/watch?v=pnkU3lx08G4