

Network Security Laboratory
23rd May 2016.



UNIVERSITY OF TRENTO - Italy
Information Engineering
and Computer Science Department



STATEFUL FIREWALL LAB

Group 11

- * **JOSHUA TETTEH OCANSEY**
- * **SAMUEL ESHUN**
- * **ACHILE SOH**
- * **DUC MANH HOANG**

CONTENTS



* INTRODUCTION

- I. What is Stateful Firewall
- II. Difference between Stateful and Stateless
- III. Example of Stateful firewall
- IV. How CBAC Firewall works
- V. How CAR Firewall works
- VI. Introduction to GNS3 Emulator
- VII. Introduction to Cisco Technology

CONTENT



* LAB ACTIVITIES

- I. Exercise 1 ---- ICMP Flood Attack
 - Lab 1 ---- CBAC
 - Lab 2 ----- CAR
- II. Exercise 2 ----- SYN Flood Attack
 - Lab 3 ----- CBAC
 - Lab 4 ----- CAR
- III. Conclusion



INTRODUCTION

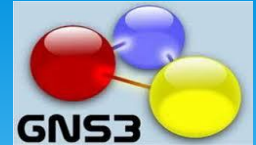
INTRODUCTION



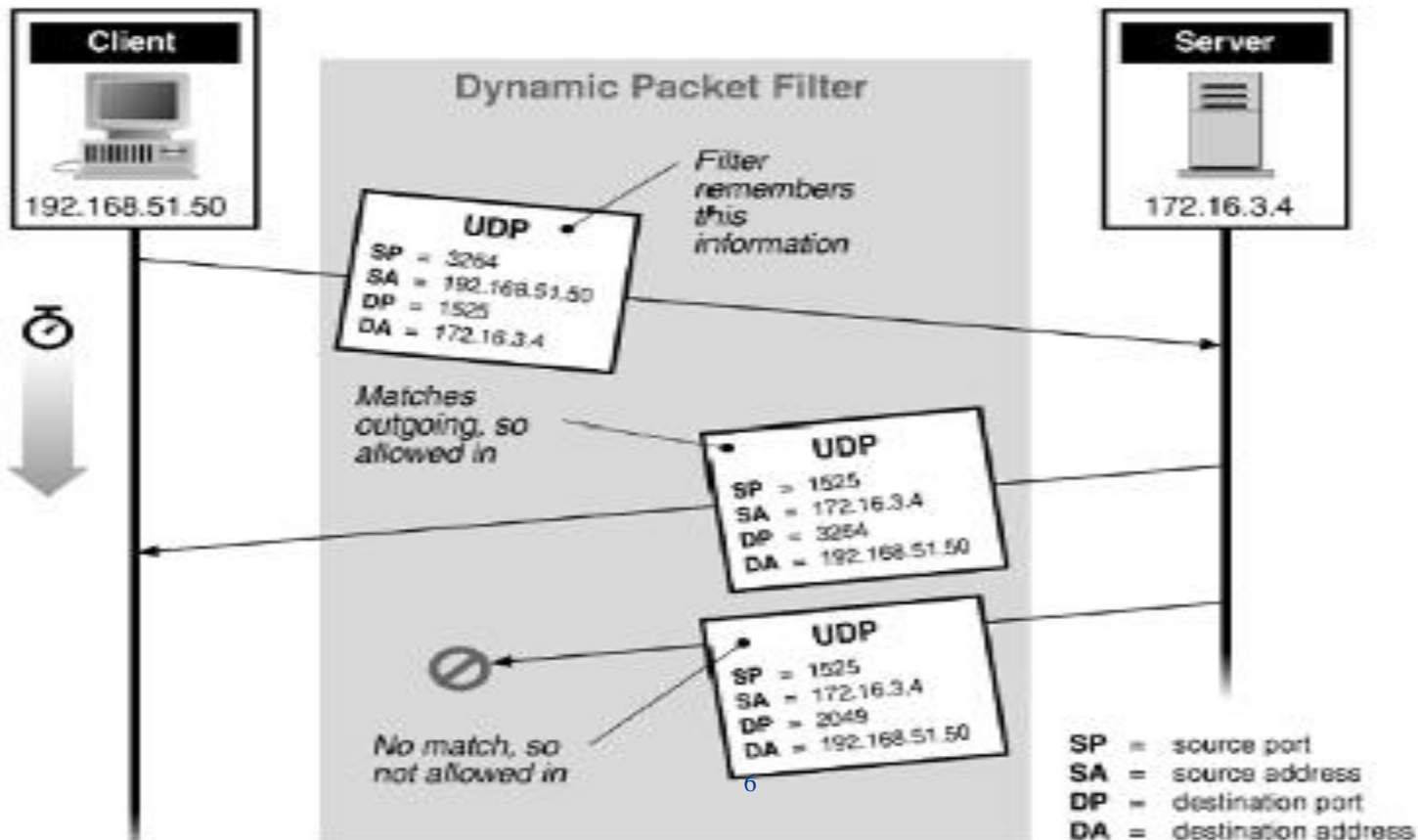
* WHAT IS STATEFUL FIREWALL

- I. Dynamic Packet Filtering
- II. Session Examination
- III. Content Evaluation

INTRODUCTION



* STATEFUL PACKET FILTERING



INTRODUCTION



* STATEFUL VRS STATELESS

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.

INTRODUCTION



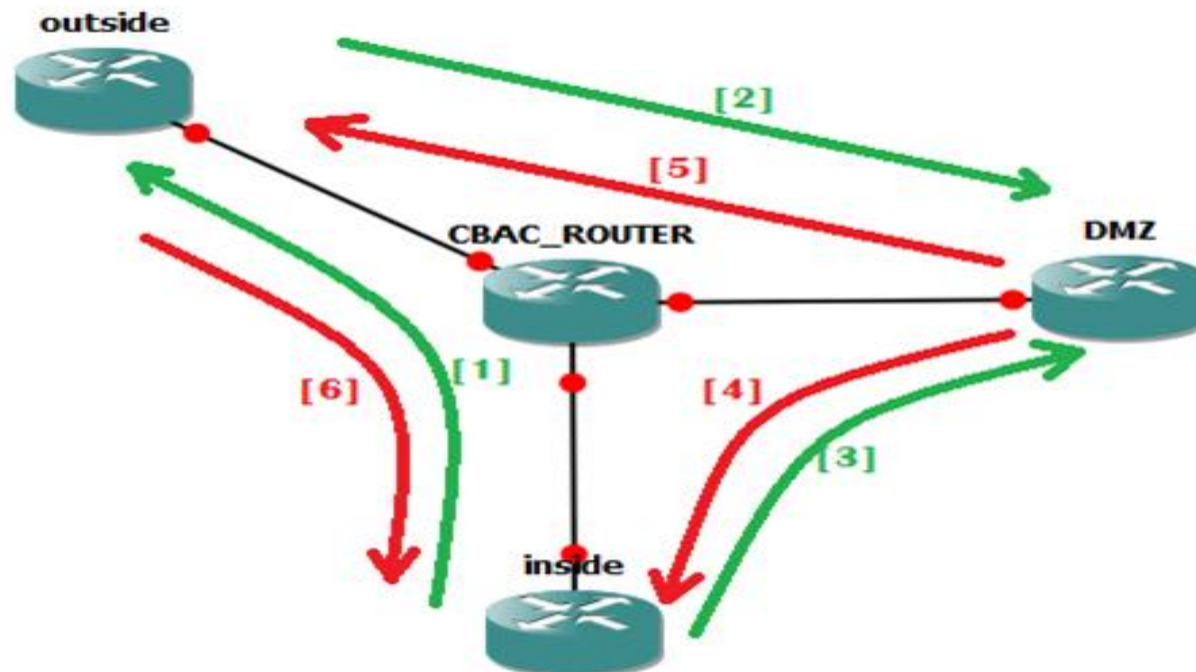
* EXAMPLES OF STATEFUL FIREWALL

- I. Context Base Access Control CBAC
- II. Committing Access Rate CAR
- III. Reflective Access Control List R-ACL

INTRODUCTION



* HOW CBAC WORKS:



INTRODUCTION



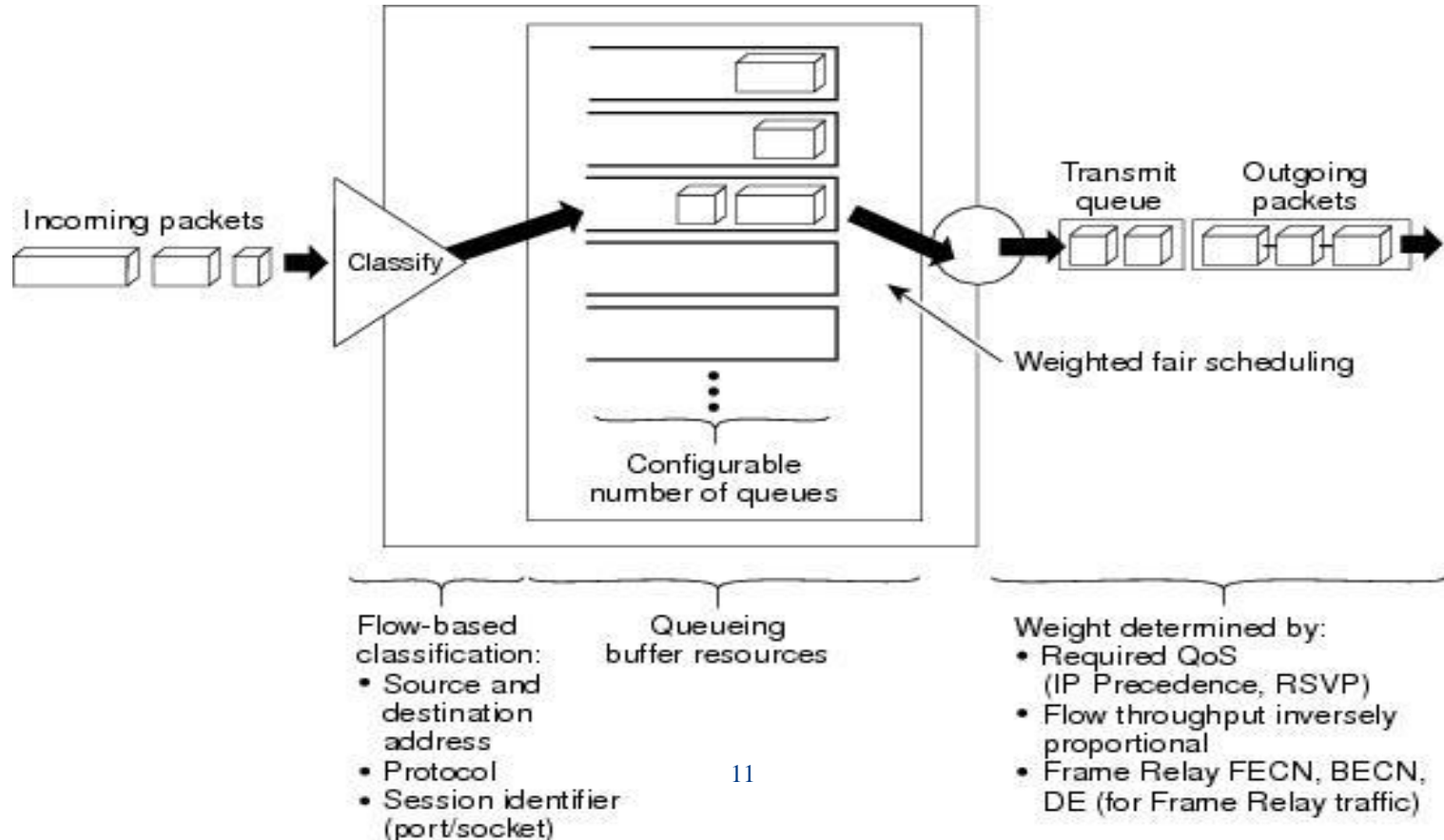
* HOW CBAC WORKS

- I. Dynamically manages extended ACL
- II. Manages session between Internal and External networks
- III. Scan for protocol violations and suspicious activities
- IV. Alarming and Logging

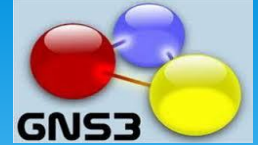
INTRODUCTION



* HOW CAR WORKS



INTRODUCTION



* HOW CAR WORKS

- I. Rate limit-- average rate, normal burst size and excess burst size
- II. Conform or exceed action--- packet Transmit or packet Drop
- III. Matching criteria--- Restrict flow based on ACL list, IP precedence and MAC address
- IV. Rate policies ---- Specifies matching criteria condition on which to perform the rating limit.

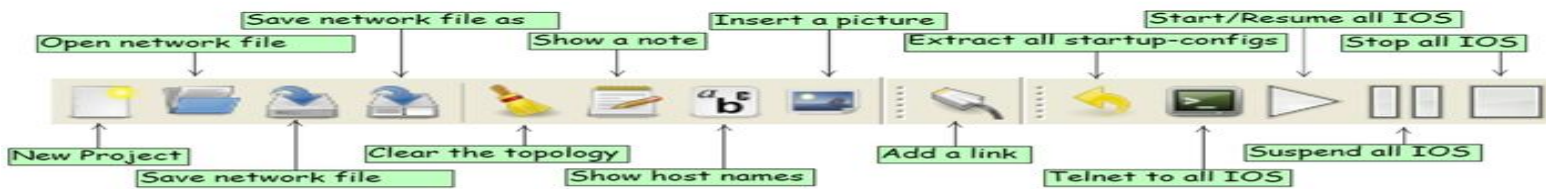
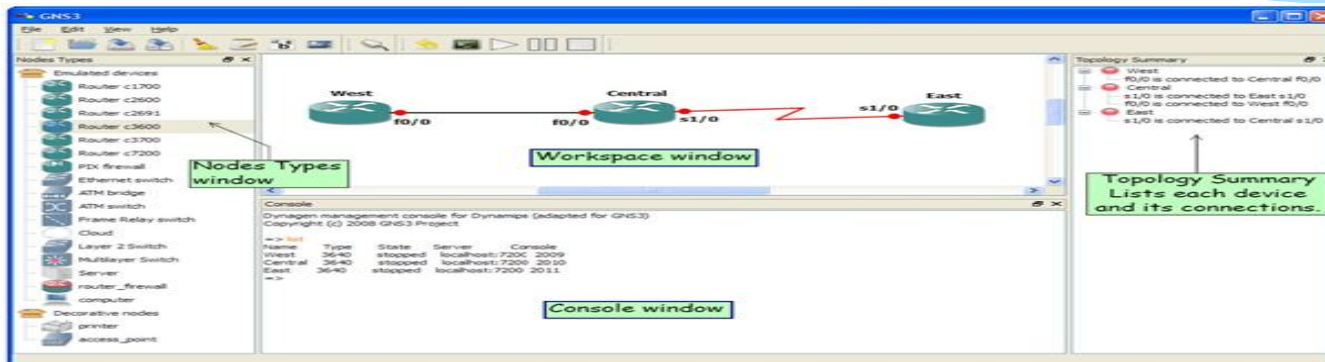
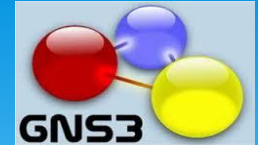
It is effective in protecting networks from DOS attacks like ICMP flood and SYN flood attacks.

INTRODUCTION



- * GNS3 INTRODUCTION
- * **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.

INTRODUCTION



- Configure
- Show/Hide the hostname
- Change the hostname
- Change console port
- Console
- Start
- Stop
- Suspend
- Idle PC
- Startup-config
- Delete



INTRODUCTION

* CISCO BASIC COMMANDS

Commands or Actions	Purpose
Enable Router> <i>enable</i>	Enables privilege EXEC mode
Configure terminal Router# <i>configure terminal</i>	Enters the global Configuration mode
End	Returns to privilege mode
Exit	Exit privilege EXEC mode
Show running config Router# <i>show running config</i>	Displays a running configuration file
Write memory Router# <i>write memory</i>	Save running configuration file to memory
Interface {fast gigabitethernet} Router(config)# <i>interface fa 1/0</i>	Specify and Enters the interface mode

FURTHER READING



- * GNS3

- * <http://blog.pluralsight.com/gns3>

- * <https://www.youtube.com/watch?v=hRshwowRW7w>

- * <https://www.youtube.com/watch?v=l6BjElin2Jw&list=P L86F49AB06DAF09C6&index=3>

- * CISCO

- * <https://www.youtube.com/watch?v=pnkU3lx08G4>

- * <http://www.cisco.com/c/en/us/support/docs/security/ios-firewall/13814-32.html>

- * http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfcarr.html



LAB ACTIVITIES

LAB



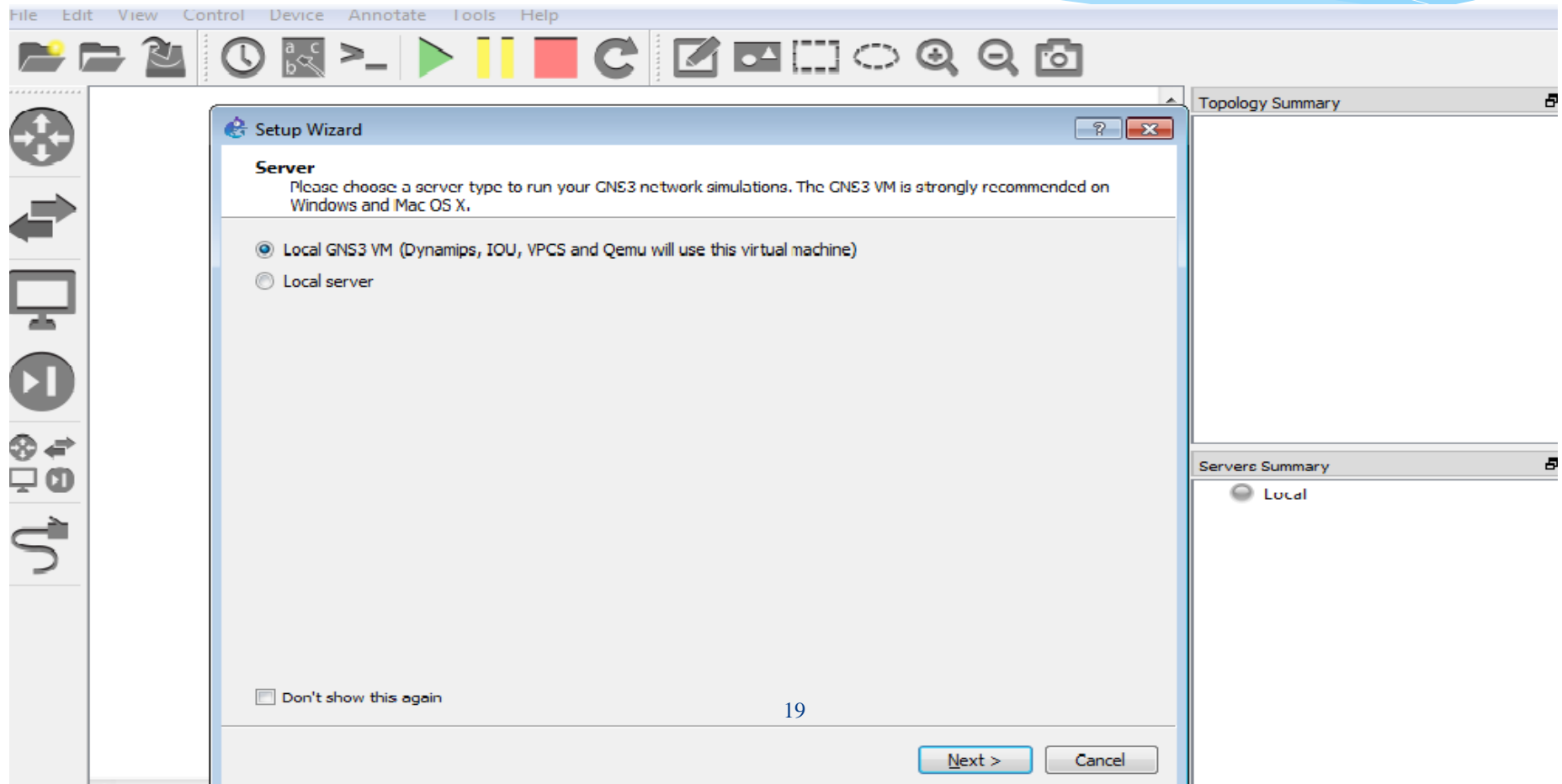
* Double Click Virtualbox icon to Start VirtualBox from your Desktop

Note: **Please DO NOT Start any Virtual Machine**

GNS 3



* Open GNS3 from your Desktop

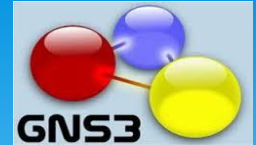


GNS3



- * Click on “Cancel” to cancel the setup wizard
- * Click on “recent project” to select ‘
Group11_NS_project.gns3

GNS3



* LABORATORY TOPOLOGY

The screenshot displays the GNS3 interface for a project named "NS_Project_final.gns3". The main workspace shows a network topology with a central router labeled "R2" with IP addresses 1.1.1.1 and 2.2.2.1. R2 is connected to three other devices:

- Attacker**: IP 1.1.1.2, located in the "Internet" zone.
- Server**: IP 2.2.2.2, located in the "DMZ" zone.
- Client_PC**: IP 10.10.10.2, located in the "INTERNAL" zone.

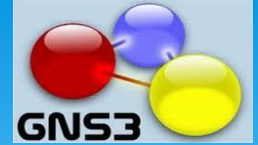
The Client_PC also has a secondary IP address of 10.10.10.2. The interface includes a menu bar (File, Edit, View, Control, Device, Annotate, Tools, Help), a toolbar with various icons, and a sidebar with navigation tools. On the right side, there are two summary panels:


- Topology Summary**: Lists the devices in the topology: Attacker, Client_PC, R2, and Server.
- Servers Summary**: Shows system resource usage: Local CPU 33.5%, RAM 56.6%.

At the bottom, a console window displays the following text:

```
GNS3 management console.  
Running GNS3 version 1.4.5 on Windows (64-bit) with Python 3.4.3 Qt 5.5.1.  
Copyright (c) 2006-2016 GNS3 Technologies.  
Use Help -> GNS3 Doctor to detect common issues.
```

GNS3



- * Click on  to start GNS3
- * The nodes in the topology will be start :
The Attacker, the client PC and the Server will boot

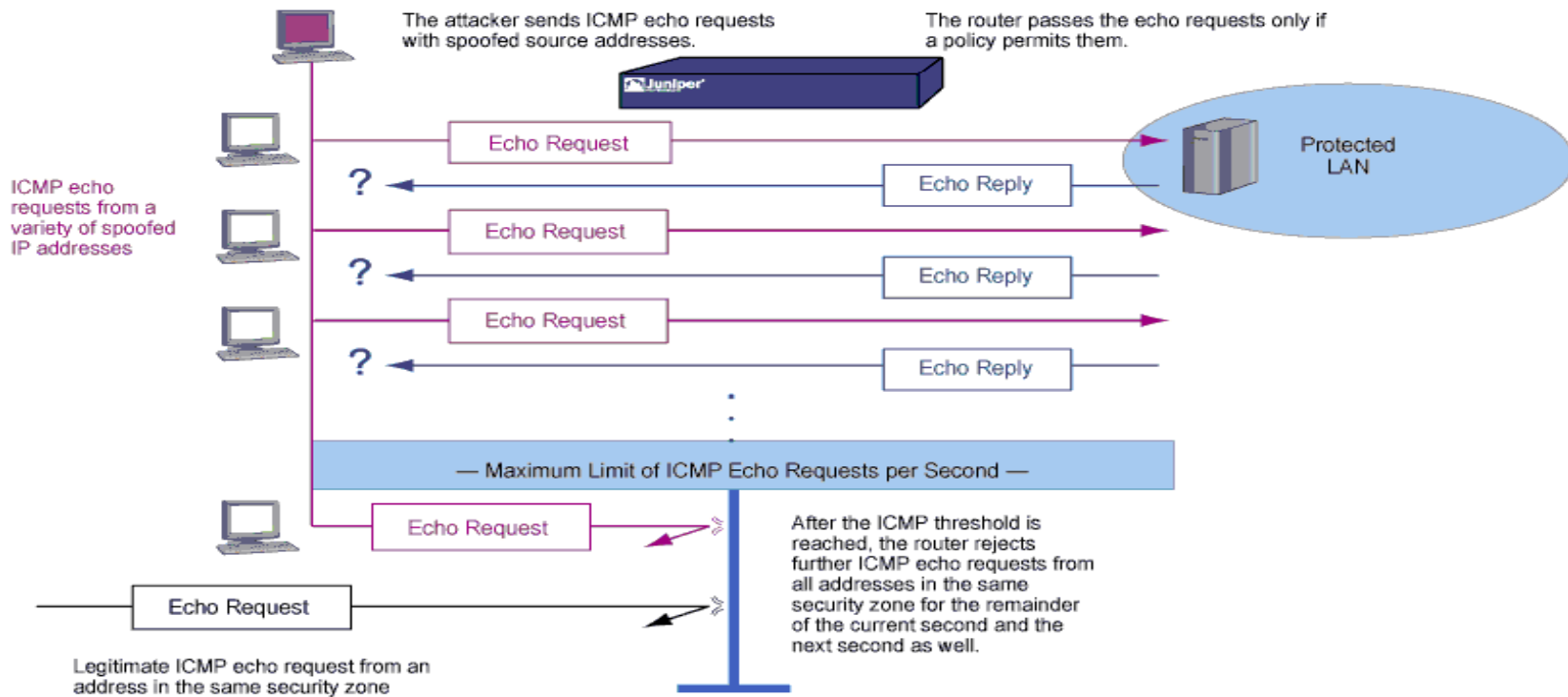
Note:

Make sure all the Virtual Computers have booted fully

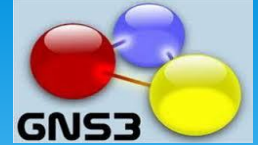
EXERCISE 1



* ICMP FLOOD ATTACK



EXERCISE 1



ICMP FLOOD ATTACK

- * Login into the Attacker computer (virtual computer)
- * Open command prompt from the start menu
- * At the prompt ***ping ubuntu.com*** (assuming the attacker does not know the IP address of the victim but has its host name”).
- * Click on XOIC_1.2 on attacker’s desktop (this tool is used for DOS attack)

EXERCISE 1



- * ICMP FLOOD
- * Click on “MAKE A DoS” to launch attack to the server



EXERCISE 1



* VERIFY ICMP FLOOD ATTACK

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

EXERCISE 1



Capturing from Standard input [R2 FastEthernet0/1 to Server Ethernet0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=952/47107, ttl=127 (reply ...)
2	0.000500	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=952/47107, ttl=64 (request ...)
3	0.041505	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=953/47363, ttl=127 (reply ...)
4	0.042005	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=953/47363, ttl=64 (request ...)
5	0.081510	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=954/47619, ttl=127 (reply ...)
6	0.082010	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=954/47619, ttl=64 (request ...)
7	0.121515	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=955/47875, ttl=127 (reply ...)
8	0.122015	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=955/47875, ttl=64 (request ...)
9	0.155519	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=956/48131, ttl=127 (reply ...)
10	0.156019	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=956/48131, ttl=64 (request ...)
11	0.195524	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=957/48387, ttl=127 (reply ...)
12	0.196025	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=957/48387, ttl=64 (request ...)
13	0.236030	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=958/48643, ttl=127 (reply ...)
14	0.236530	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=958/48643, ttl=64 (request ...)
15	0.276035	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=959/48899, ttl=127 (reply ...)
16	0.276535	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=959/48899, ttl=64 (request ...)
17	0.306038	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=960/49155, ttl=127 (reply ...)
18	0.306539	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=960/49155, ttl=64 (request ...)
19	0.336042	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=961/49411, ttl=127 (reply ...)
20	0.336542	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=961/49411, ttl=64 (request ...)
21	0.367546	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=962/49667, ttl=127 (reply ...)
22	0.368046	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=962/49667, ttl=64 (request ...)
23	0.407551	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=963/49923, ttl=127 (reply ...)
24	0.408051	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=963/49923, ttl=64 (request ...)
25	0.447556	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=964/50179, ttl=127 (reply ...)
26	0.448057	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=964/50179, ttl=64 (request ...)
27	0.467559	1.1.1.2	2.2.2.2	ICMP	42	Echo (ping) request id=0x0001, seq=965/50435, ttl=127 (reply ...)
28	0.468059	2.2.2.2	1.1.1.2	ICMP	60	Echo (ping) reply id=0x0001, seq=965/50435, ttl=64 (request ...)

```
0000 08 00 27 27 a9 b4 c2 05 15 f4 00 01 08 00 45 00  ....E.
0010 00 1c 06 87 40 00 7f 01 ef 53 01 01 01 02 02 02  ....@...S.....
0020 02 02 08 00 f4 46 00 01 03 b8                ....F. ....
```

Ready to load or capture | Packets: 5127 · Displayed: 5127 (100.0%) | Profile: Default

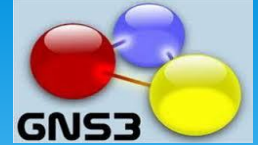
EXERCISE 1



* MITIGATION

1. **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*)
2. **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.

EXERCISE 1: LAB 1



* CBAC

Double-click on the router in GNS3 platform

1. Router > *enable*
2. Router # *configure terminal*
3. Router(config)#*access-list 100 deny icmp any any*
4. Router(config)#*access-list 100 permit ip any any*
5. Router(config)#*ip inspect name GROUP11 icmp*
6. Router(config)#*interface fastEthernet 0/0*
7. Router(config-if)#*ip access-group 100 in*
8. Router(config-if)#*ip inspect GROUP11 out*
9. Router(Config-if)# *end*

EXERCISE 1: LAB 1



Commands or Actions	Purpose
Router(config)# <i>access-list 100 deny icmp any any</i>	ACL rule to drop any icmp packet from any source to any destination
Router(config)# <i>access-list 100 permit ip any any</i>	ACL rule to allow any IP packet from any source to any destination
Router(config)# <i>ip inspect name GROUP11 icmp</i>	Enables CBAC inspection on ICMP traffic
Router(config-if)# <i>ip access-group 100 in</i>	Apply ACL rules on interface
Router(config-if)# <i>ip inspect GROUP11 out</i>	Apply CBAC inspection rule on interface

EXERCISE 1: LAB 1



* VERIFY CBAC DEFENSE FOR ICMP FLOOD ATTACK

1. Check the behavior of traffic from Wireshark

2. 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*)

EXERCISE 1: LAB 1



* VERIFY CBAC DEFENSE FOR ICMP FLOOD ATTACK

3. 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

EXERCISE 1: LAB 2



- * For us to see the effect of the CAR method, let us disable the CBAC from the router

```
Router(config)#interface fast
```

```
Router(config)# interface fastEthernet 0/0
```

```
Router(config-if)# no ip access-group 100 in
```

```
Router(config-if)# no ip insepct GROUP11 out
```

```
Router(config-if)# exit
```

(We observed from the wireshark logs that the attack has started again after the disabling the firewall.)

EXERCISE 1: LAB 1



Commands or Actions	Purpose
<code>interface fastEthernet 0/0</code>	Enter the interface mode
<code>no ip access-group 100 in</code>	Disable ACL rule on interface
<code>no ip insepect GROUP11 out</code>	Disable CBAC inspection on interface
<code>exit</code>	Return to Global configuration mode.

EXERCISE 1: LAB 2



* CAR

Router # *configure terminal*

Router(config)# **access-list 150 permit icmp any any
echo**

Router(config)# *interface fastEthernet 0/0*

Router(config-if)# *rate-limit input access-group 150
8000 1500 2000 conform-action transmit exceed-action
drop*

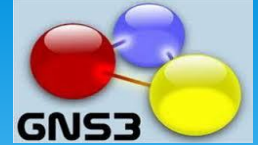
Router(config-if)# *end*

EXERCISE 1: LAB 2



Commands or Actions	Purpose
access-list 150 permit icmp any any echo	ACL rule to drop any icmp packet from any source to any destination
<i>rate-limit input access-group 150 8000 1500 2000 conform-action transmit exceed-action drop</i>	Specify the rate-limit policy for ACL rule to conform or exceed packet size

EXERCISE 1: LAB 2



* VERIFY CAR FIREWALL: ICMP FLOOD ATTACK

1. Let us check the behavior of the traffic on the wireshark monitor

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

FastEthernet0/0

Output

matches: access-group 150

params: 8000 bps, 1500 limit, 2000 extended limit

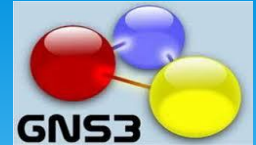
conformed 15272 packets, 678776 bytes; action: transmit

exceeded 715 packets, 30590 bytes; action: drop

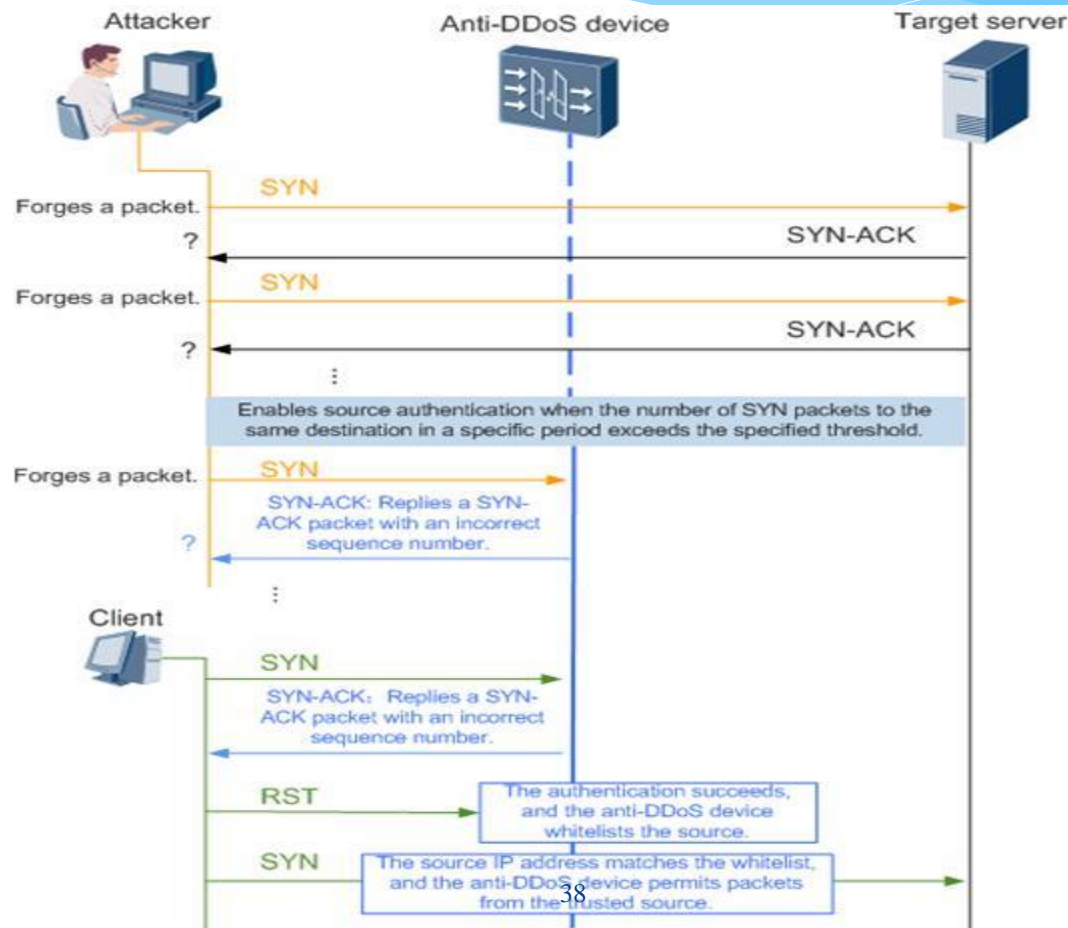
last packet: 20ms ago, current burst: 1438 bytes

last cleared 00:11 ago, conformed 8000 bps, exceeded 0 bps

EXERCISE 2



* SYN FLOOD ATTACK



EXERCISE 2



- * SYN FLOOD ATTACK

- * Login unto the Attacker computer
- * Open nmap-Zenmap GUI from Attacker's desktop
- * Type *ubuntu.com* at the target and click on scan
(Again, we assume the only information the attacker knows is the host name of the victim)

The attacker use nmap to scan the victim's network to understand the topology of network and services

EXERCISE 2



* SYN FLOOD ATTACK

The screenshot shows the Zenmap application window. The target is set to 'ubuntu.com' and the profile is 'Intense scan'. The command entered is 'nmap -T4 -A -v ubuntu.com'. The main display area shows the output of the scan, including port 80/tcp being open with Apache httpd 2.2.22. The output also includes details about the device type (Linux 3.X|4.X), OS CPE, uptime guess, network distance, and traceroute results. The scan is completed in 44.042 seconds.

Zenmap

Scan Tools Profile Help

Target: Profile:

Command:

Hosts Services

Nmap Output Ports / Hosts Topology Host Details Scans

OS Host

ubuntu.com (2.2.2)

```
nmap -T4 -A -v ubuntu.com
Host is up (0.010s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.2.22 ((Ubuntu))
|_ http-methods:
|_ Supported Methods: POST OPTIONS GET HEAD
|_ http-server-header: Apache/2.2.22 (Ubuntu)
|_ http-title: Site doesn't have a title (text/html).
Device type: general purpose
Running: Linux 3.X|4.X
OS_CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS_details: Linux 3.2 - 4.4
Uptime_guess: 0.017 days (since Sun May 15 17:03:56 2016)
Network_Distance: 2 hops
TCP_Sequence_Prediction: Difficulty=262 (Good luck!)
IP_ID_Sequence_Generation: All zeros

TRACEROUTE (using port 995/tcp)
HOP RTT ADDRESS
1 16.00 ms 1.1.1.1
2 31.00 ms ubuntu.com (2.2.2.2)

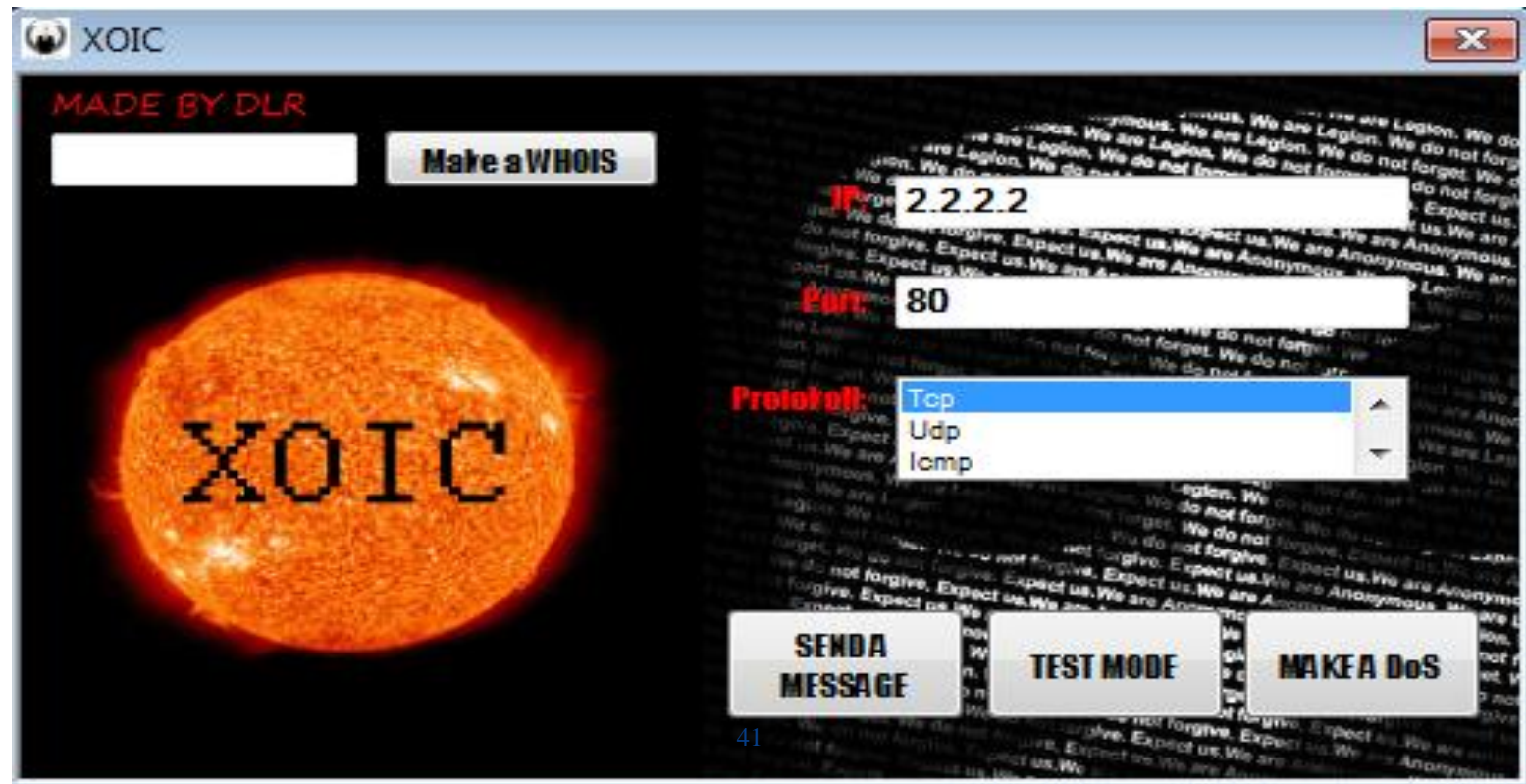
NSE: Script Post-scanning.
Initiating NSE at 17:28
Completed NSE at 17:28, 0.00s elapsed
Initiating NSE at 17:28
Completed NSE at 17:28, 0.00s elapsed
Read data files from: C:\Program Files\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.042 seconds
Raw packets sent: 1429 (63.662KB) | Rcvd: 1020 (41.498KB)
```

Filter Hosts

EXERCISE 2



- * Now !!! Let us attack the http server with port 80
- * Double-click on the XOIC_1.2 to open attack tool



EXERCISE 2



* VERIFY SYN FLOOD ATTACK

1. Let us check the behavior of the traffic on the wireshark monitor
2. Login into the Server computer, open terminal (crt + Alt + T). Type *sudo tcpdump -i eth0* to observe the attack to the server.

EXERCISE 2



Capturing from Standard input [R2 FastEthernet0/1 to Server Ethernet0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

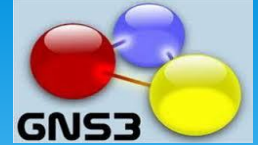
Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
2852...	9179.211708	2.2.2.2	1.1.1.2	TCP	66	80 → 10256 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SAC...
2852...	9179.230211	1.1.1.2	2.2.2.2	TCP	54	10255 → 80 [ACK] Seq=2 Ack=2 Win=65536 Len=0
2852...	9179.240712	1.1.1.2	2.2.2.2	TCP	54	10256 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
2852...	9179.250713	1.1.1.2	2.2.2.2	TCP	54	10256 → 80 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0
2852...	9179.251213	2.2.2.2	1.1.1.2	TCP	60	80 → 10256 [FIN, ACK] Seq=1 Ack=2 Win=29312 Len=0
2852...	9179.260714	1.1.1.2	2.2.2.2	TCP	66	10257 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PER...
2852...	9179.261214	2.2.2.2	1.1.1.2	TCP	66	80 → 10257 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SAC...
2852...	9179.290718	1.1.1.2	2.2.2.2	TCP	54	10256 → 80 [ACK] Seq=2 Ack=2 Win=65536 Len=0
2852...	9179.300720	1.1.1.2	2.2.2.2	TCP	54	10257 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
2852...	9179.310721	1.1.1.2	2.2.2.2	TCP	54	10257 → 80 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0
2852...	9179.312221	2.2.2.2	1.1.1.2	TCP	60	80 → 10257 [FIN, ACK] Seq=1 Ack=2 Win=29312 Len=0
2852...	9179.320722	1.1.1.2	2.2.2.2	TCP	66	10258 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PER...
2852...	9179.321222	2.2.2.2	1.1.1.2	TCP	66	80 → 10258 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SAC...
2852...	9179.331223	1.1.1.2	2.2.2.2	TCP	54	10257 → 80 [ACK] Seq=2 Ack=2 Win=65536 Len=0
2852...	9179.351726	1.1.1.2	2.2.2.2	TCP	54	10258 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
2852...	9179.361727	1.1.1.2	2.2.2.2	TCP	54	10258 → 80 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0
2852...	9179.362227	2.2.2.2	1.1.1.2	TCP	60	80 → 10258 [FIN, ACK] Seq=1 Ack=2 Win=29312 Len=0
2852...	9179.371729	1.1.1.2	2.2.2.2	TCP	66	10259 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PER...
2852...	9179.371729	2.2.2.2	1.1.1.2	TCP	66	80 → 10259 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SAC...

- ▶ Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
- ▶ Ethernet II, Src: c2:05:15:f4:00:01 (c2:05:15:f4:00:01), Dst: CadmusCo_27:a9:b4 (08:00:27:27:a9:b4)
- ▶ Internet Protocol Version 4, Src: 1.1.1.2, Dst: 2.2.2.2
- ▶ Internet Control Message Protocol

```
0000 08 00 27 27 a9 b4 c2 05 15 f4 00 01 08 00 45 00  ..''.....E.
0010 00 1c 05 62 40 00 7f 01 f0 78 01 01 01 02 02 02  ...b@...x.....
0020 02 02 08 00 f3 7b 00 01 04 83                   .....{... .. 43
```

EXERCISE 2: LAB 3



CBAC FIREWALL

* 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*

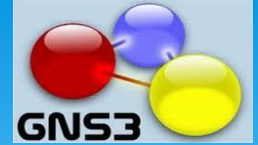
Router(config-if)# *end*

EXERCISE 2: LAB 3



Commands or Actions	Purpose
<i>access-list 199 permit tcp any host 2.2.2.2 eq www established</i>	ACL rule to permit http packet from any source to host and ensure TCP3-handshake
<i>ip inspect name FIREWALL http</i>	Enable CBAC inspection for http traffic
<i>ip inspect FIREWALL out</i>	Apply CBAC inspection rule on interface
<i>ip access-group 199 in</i>	Apply ACL rule on interface

EXERCISE 2: LAB 3



* VERIFY CBAC FIREWALL: SYN FLOOD ATTACK

1. Open Wireshark and observe logs

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

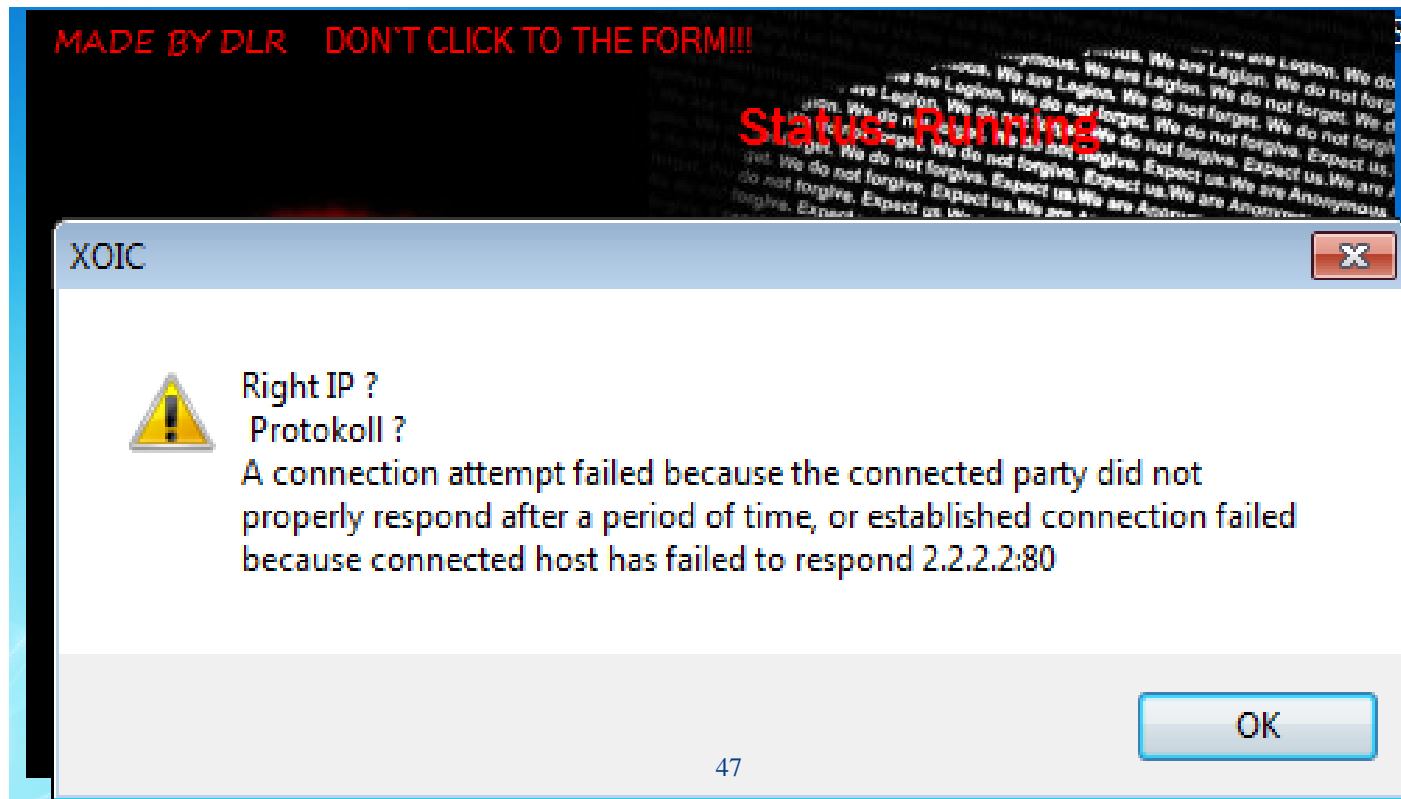
Extended IP access list 199

10 permit tcp any host 2.2.2.2 eq www established (4 matches)

3. Login into the Attacker machine and observe the attacking tool used

EXERCISE 2: LAB 3

* VERIFY CBAC FIREWALL: SYN FLOOD ATTACK



EXERCISE 2: LAB 4



Let disable the previous firewall method so we can implement the CAR method.

```
Router# configure terminal
```

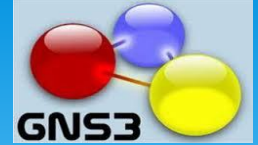
```
Router(config)# interface fastEthernet 0/0
```

```
Router(config-if)# no ip inspect FIREWALL out
```

```
Router(config-if)# no ip access-group 199 in
```

```
Router(config-if)# end
```


EXERCISE 2: LAB 4



* CAR FIREWALL

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*

EXERCISE 2: LAB 4



Commands or Actions	Purpose
<i>access-list 120 deny tcp any host 2.2.2.2 eq www established</i>	ACL rule to drop any http packet from any source to a specific host that does not ensure tcp 3-handshake
<i>access-list 120 permit tcp any host 2.2.2.2 eq www</i>	ACL rule to allow any http packet from any host to specific host (server)
<i>interface fastEthernet 0/0</i>	Enter the interface mode
<i>rate-limit output access-group 120 8000 1500 2000 conform-action transmit exceed-action drop</i>	Apply Rate-limit policy on ACL rule to transmit or reject traffic based on the policy

EXERCISE 2: LAB 4



* VERIFY CAR FIREWALL: SYN FLOOD ATTACK

1. 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

EXERCISE 2: LAB 4



2. We can observe the logs in Wireshark

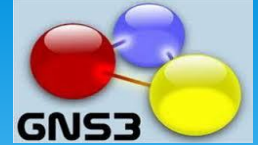
The screenshot shows a Wireshark capture of network traffic. The main pane displays a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The traffic consists of multiple TCP connections between source IP 1.1.1.2 and destination IP 2.2.2.2. The connections are established using the SYN sequence, and some are terminated with FIN. The 'Info' column provides details for each packet, such as sequence numbers, acknowledgment numbers, window sizes, and lengths. For example, packet 4599 shows a SYN packet from 1.1.1.2 to 2.2.2.2 on port 80. Packet 4598 shows the corresponding SYN-ACK from 2.2.2.2 to 1.1.1.2. Packet 4597 shows a FIN packet from 1.1.1.2 to 2.2.2.2, and packet 4596 shows the FIN-ACK from 2.2.2.2 to 1.1.1.2. The bottom pane shows the details of the selected packet (No. 42), identifying it as an Ethernet II frame, an Internet Protocol Version 4 packet, and a Telnet Control Message Protocol message.

No.	Time	Source	Destination	Protocol	Length	Info
4599..	15810.974079	1.1.1.2	2.2.2.2	TCP	54	13968 → 80 [ACK] Seq=2 Ack=2 Win=55536 Len=0
4598..	15810.964881	1.1.1.2	2.2.2.2	TCP	54	13969 → 80 [ACK] Seq=1 Ack=1 Win=55536 Len=0
4597..	15810.954882	1.1.1.2	2.2.2.2	TCP	54	13069 → 80 [FIN, ACK] Seq=1 Ack=1 Win=55536 Len=0
4596..	15810.945382	2.2.2.2	1.1.1.2	TCP	60	80 → 13969 [FIN, ACK] Seq=1 Ack=2 Win=29312 Len=0
4595..	15811.004883	1.1.1.2	2.2.2.2	TCP	66	[TCP Port numbers reused] 13910 → 80 [SYN] Seq=0 Win=8192 Len=0
4594..	15811.005383	2.2.2.2	1.1.1.2	TCP	66	80 → 13910 [SYN, ACK] Seq=0 Ack=1 Win=29280 Len=0 MSS=1460 SAC...
4593..	15811.004887	1.1.1.2	2.2.2.2	TCP	54	13909 → 80 [ACK] Seq=2 Ack=2 Win=55536 Len=0
4592..	15811.004888	1.1.1.2	2.2.2.2	TCP	54	13910 → 80 [ACK] Seq=1 Ack=1 Win=55536 Len=0
4591..	15811.004889	1.1.1.2	2.2.2.2	TCP	54	13910 → 80 [FIN, ACK] Seq=1 Ack=1 Win=55536 Len=0
4590..	15811.005389	2.2.2.2	1.1.1.2	TCP	60	80 → 13910 [FIN, ACK] Seq=1 Ack=2 Win=29312 Len=0
4589..	15811.004891	1.1.1.2	2.2.2.2	TCP	66	[TCP Port numbers reused] 13911 → 80 [SYN] Seq=0 Win=8192 Len=0
4588..	15811.004891	2.2.2.2	1.1.1.2	TCP	66	80 → 13911 [SYN, ACK] Seq=0 Ack=1 Win=29280 Len=0 MSS=1460 SAC...
4587..	15811.005805	1.1.1.2	2.2.2.2	TCP	54	13910 → 80 [ACK] Seq=2 Ack=2 Win=55536 Len=0
4586..	15811.005806	1.1.1.2	2.2.2.2	TCP	54	13911 → 80 [ACK] Seq=1 Ack=1 Win=55536 Len=0
4585..	15811.116897	1.1.1.2	2.2.2.2	TCP	54	13912 → 80 [FIN, ACK] Seq=1 Ack=1 Win=55536 Len=0
4584..	15811.117307	2.2.2.2	1.1.1.2	TCP	60	80 → 13912 [FIN, ACK] Seq=1 Ack=2 Win=29312 Len=0
4583..	15811.126899	1.1.1.2	2.2.2.2	TCP	66	[TCP Port numbers reused] 13912 → 80 [SYN] Seq=0 Win=8192 Len=0
4582..	15811.127399	2.2.2.2	1.1.1.2	TCP	66	80 → 13912 [SYN, ACK] Seq=0 Ack=1 Win=29280 Len=0 MSS=1460 SAC...
4581..	15811.126900	1.1.1.2	2.2.2.2	TCP	54	13911 → 80 [ACK] Seq=2 Ack=2 Win=55536 Len=0

▶ Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▶ Ethernet II, Src: c2:05:15:f4:00:01 (c2:05:15:f4:00:01), Dst: CadmusCo_27:a9:b4 (08:00:27:27:a9:b4)
▶ Internet Protocol Version 4, Src: 1.1.1.2, Dst: 2.2.2.2
▶ Telnet Control Message Protocol

```
0000 08 00 27 27 a9 b4 c2 05 15 f4 00 01 08 00 15 00  ...*... ..E...
0010 00 1c 05 62 40 06 f7 b1 fb 78 b2 01 01 02 02 b2  ...b@... ..X....
0020 02 02 08 0d 75 7b 00 01 00 00 00 00 00 00 00 00  .....f... ..
```

CONCLUSION



- * Stateful firewall and importance
- * Examples of Stateful Firewall
- * Introduction to GNS3 and Cisco
- * Exercise One was on ICMP flood Attack and how it can be mitigated by CBAC and CAR
- * Exercise two was the SYN flood Attack and how CBAC and CAR can be used to block such attack.

GRAZIE !!!



THANK YOU



QUESTIONS

