REMOTE PENETRATION TESTING TOOLKIT

By
NC Muhammad Talha Younus
PC Aoon Ben Mustafa
PC Talha Ahmad
NC Faizan Ali

Submitted to the Faculty of Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology,
Islamabad
in partial fulfillment for the requirements of BE Degree in
Electrical (Telecommunication) Engineering

JUNE, 2017

CERTIFICATE OF CORRECTNESS AND APPROVAL

It is certified that the work contained in this thesis “Remote Penetration Testing Toolkit”, was carried out by Muhammad Talha Younus, Aoon Ben Mustafa, Faizan Ali and Talha Ahmad under the supervision of Lecturer Waleed Bin Shahid, for the partial fulfilment of Degree of Electrical (Telecommunication) Engineering, is correct and approved.

Approved by

____

(Lecturer Waleed Bin Shahid)
Department of Information Security
Project Directing Staff (DS)
Military College of Signals (MCS, NUST)

Dated: ____ June, 2017

ABSTRACT

Penetration Testing is performed on networks and systems now-a-days but for each test, the pen-tester has to be present on the network; performing pen-test. This can be problem especially in intelligence based applications. Process of penetration testing and devices used for this are very expensive. Our project aims at making a Remote Penetration Testing Toolkit using Raspberry Pi. Raspberry Pi will be running Rasbian (Jessie ~release 11-01-2017~) and a pen-testing suite containing embedded attacks. This device will be controlled by a Command and Control Server over a wireless connection (Wi-Fi/4G Dongle). The device will be installed inline or standalone in a network. Once initiated, it will perform attacks which have been defined earlier and at the end, it will generate a brief report of the test. User will be able to add or remove attacks for a specific test.

DECLARATION

No portion of this work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

“Allah Almighty for His countless blessings

Instructors and friends for their help

And our beloved parents for their prayers and support”

ACKNOWLEDGEMENT

Praise be to Almighty Allah Who guided and enabled us to undertake this project.

We would like to thank our project supervisor Lecturer Waleed Bin Shahid for his assistance, supervision and generous support throughout our Final Year Project. We would like to extend our gratitude towards Lecturer Narmeen Shafqat of Department of Information Security, Military College of Signals (MCS, NUST) for her help and guidance throughout this time period.

We would like to thank lab staff, our seniors, friends and all those who have helped us, either directly or indirectly, in the completion of this project.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS

OS	Operating System
OSI	Open Systems Interconnection
Wi-Fi	Wireless Fidelity
IP	Internet Protocol
GHz	Giga Hertz
LAN	Local Area Network
GB	Giga Bytes
RAM	Random Access Memory
USB	Universal Serial Bus
GPIO	General Purpose Input Output
HDMI	High Definition Multimedia Interface
mm	millimeter
SD	Storage Device
FHS	Filesystem Hierarchy Standard
ARM	Advanced RISC Machines
RISC	Reduced Instruction Set Computer
C&C	Command and Control
GUI	Graphical User Interface
WPA	Wireless Protected Access
PSK	Pre-Shared Key
WEP	Wired Equivalent Privacy
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
BSD	Berkeley Software Distribution

ICMP	Internet Control Message Protocol
DNS	Domain Name Server
TOS	Type of Service

1. INTRODUCTION

1.1 Overview

Information Security has always been an important concern of modern day communication procedures. One way to ensure system and communication safety is to do penetration testing. **Penetration testing** (also called **pen-testing**) is the practice of testing a computer system, network or Web application to find vulnerabilities that an attacker could exploit. This is done by simulating a hacker's attack on the system/network/application.

1.2 Problem Statement

Penetration Tests are performed on networks and systems now-a-days but for each test, the pen-tester has to be present on the network; performing pen-test. This can be a problem especially in intelligence based applications. Another major problem of penetration testing is the cost. Devices used for this purpose are very costly. Also the process itself is very costly. Increase in distance also increases the cost. If a pen-tester is sitting in some city or country and he has to perform pen-test on a network in some other city or country, he has to move him and his equipment to the network. This causes an addition to the cost. At the end, results of the attacks cannot be managed.

1.3 Approach

We are using Raspberry Pi (with Rasbian Jessie ~release 11-1-2017~ as its OS) as our pen-testing device. There are attacks embedded in the device. User is able to add or remove attacks according to the test. This device is controlled over wireless connection by a Command and Control Server which is simply a laptop with Linux-Backtrack installed. Now user is able to perform pen-test on a network from anywhere in the world. He has to just send the device and get it installed in the network. At the end, it automatically generates a brief report of the attacks. This solves the problems of cost, man-engagement and management of reports.

1.4 Objectives

This project is based on the concepts of Computer & Communication Networks and Network Security. One main objective of this project is to make a penetration testing device which should go undetected in a network. This is because detection of pen-testing device can cause a huge problem during intelligence or spying applications. A penetration testing suite will be designed for Raspberry Pi with attacks embedded in it. The device should also be customizable in terms of embedded attacks. For a specific network, user should be able to add or remove attacks. Our main focus is to provide penetration testing for military, intelligence, banking and other general applications at low cost.

2. BACKGROUND

2.1 Existing Literature

- With the advancement of technology, everything now-a-days is online. We have access to our data everywhere. Organizations are working in networks. There is inter and intra organizational data communication. With this advancement, the concern of data security has also increased. To check data security, one has to perform attack on his own network. There are organizations which provide these services. Now there is a competition among those organizations. Main concern is cost. Which company provides services at low cost, it is hired.
- OSI Model is a 7 layered network model.
    1. **Physical Layer:** The layer responsible for communication of raw data (both transmission and reception) over some physical medium like Ethernet or Wi-Fi.
    2. **Data Link Layer:** The layer responsible for transmission of data segments between two hosts of a network.
    3. **Network Layer:** The layer responsible for internetwork communication using IP addressing.
    4. **Transport Layer:** The layer responsible for reliable transfer of data using the methods of segmentation and error, flow and congestion control.
    5. **Session Layer:** The layer responsible for establishing and managing a successful session for communication between two hosts in a network.
    6. **Presentation Layer:** The layer responsible for data formatting, compression, encoding and decoding so that it becomes easier for application to understand and process it.
    7. **Application Layer:** The layer responsible for providing a high level user interface.

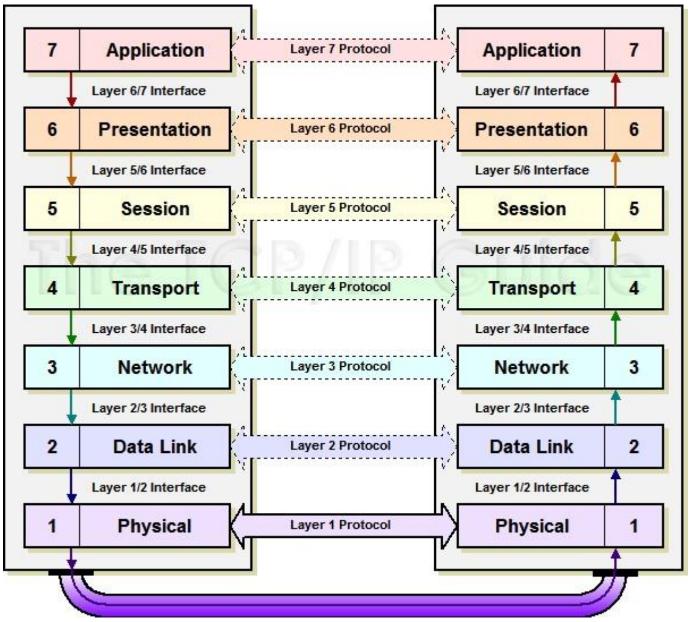


Figure 1: OSI Model

- Attacks can be embedded in a device for auto operation. Once installed and initiated, pen-testing device can perform attacks automatically.
- Raspberry Pi can be controlled remotely over a wireless connection (Wi-Fi/4G Dongle) by any laptop.

2.2 Use of Existing Literature

- If we could control our device remotely, it could decrease our cost. And if this device is a Raspberry Pi, this cost will reduce to just $50.
- If we see the OSI model, Network Layer assigns IP address to the device. Working on data link layer will not assign IP address to the address. This property can be used to make a device go undetected in a network. This is very important in intelligence based applications and spying.

3. REQUIREMENTS AND SPECIFICATIONS

3.1 Raspberry Pi

Raspberry Pi 3 is the device which we will be using in this project. Following are the specifications of Raspberry Pi:

- CPU: 1.2 GHz quad-core
- 802.11n Wireless LAN
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port

- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface
- Display interface
- Micro SD card slot
- Price: $35

Availability of Ethernet Port and Wi-Fi card make us able to pen-test both wired and wireless networks. For in-line connection, USB port will be converted to Ethernet port using standard USB to Ethernet converter.

Figure 2: Raspberry Pi 3

Figure 3: USB to Ethernet Converter

3.2 Linux-BackTrack (Kali Linux)

Kali Linux is an open source, Debian-based Linux distribution designed specifically for Ethical Hacking, Pen-Testing and Security Assessment. It contains hundreds of tools used at different steps in Ethical Hacking, Penetration Testing or Reverse Engineering in case of a security breach in a network.

On 13th March, 2013, a complete rebuild of Linux BackTrack was released in name of Kali Linux which adheres completely to development standards of Debian.

Kali Linux has following key specifications:

Nearly 700 Pen-Testing tools: Kali Linux contains almost every tool needed to perform any type of activity on a network. If you want to perform just reconnaissance or you are planning a full fledge attack on network, Kali Linux has got your back. It will provide you with a variety of tools for your need.

Free of cost: Best point of Kali Linux is that you do not have to pay for anything to use. Even it contains many (free) tools from other platforms which are charged for their using them.

Open source: Kali Linux is Open Source! Which means it has a development tree available. If you want to tweak or rebuild different tools for your specific need, you are good to go. You can find all the code which goes into Kali Linux. Isn't it awesome?

FHS adhering: Kali Linux is designed to adhere completely to the FHS (Filesystem Hierarchy Standard) which allows you to locate libraries, files, logs, directories, etc.

Wireless device support: Among other great specifications, there is another! It has been designed to support a huge variety of wireless devices. It comes compatible with a good number of hardware and supports a variety of USB and wireless devices. This property completely adheres to Linux standards.

Custom kernel: Kali Linux's kernel has patches of injections so that the development team can easily do wireless assessments.

Securely developed by trusted individuals: The team responsible for development of Kali Linux is a trusted group to use packages and repositories. All of this is done in a proper secure environment using multiple secure rule set.

Everything is signed: Each and every tool plus package is signed by the individual or team who is responsible for its development. In addition to this, all repositories are also signed by developers. This means you can easily use every package and tools without any fear of being harmed.

Support for different languages: As a pen-tester can be from anywhere from the world and can be speaking any language, therefore, Kali Linux provides support for different languages so that the individual using it may be able to operate it in his/her native language.

Support for ARM: Kali Linux is not limited to your laptops and computers only. It also has a wide range of support for ARM based boards like Raspberry Pi and BeagleBone Black which are more inexpensive and perform just like any other computers. ARM installations of Kali Linux have all tools included in them.

4. METHODOLOGY

4.1 Basic Penetration Testing

Following is the basic methodology which is employed while doing penetration testing:

4.1.1 Establishing Goal

This is the first and basic step of penetration testing. In this step, you decide what you want to do. A proper baseline of penetration test is defined in this step. All of the do's and don'ts of pen-testing procedure are agreed between parties. What type of test would be performed i.e. Back Box, White Box or Grey Box. What will be provided to pen-tester and what will pen-tester give back to organization. What methodology will be employed? How it will be performed? Which port, processes and applications will be scanned? In short, a proper documentation of goals to be achieved in pen-testing is prepared on the base of which, further process is performed. Identifying gaps in security.

4.1.2 Reconnaissance

In this step, a proper scanning of the network, application or any other target is done keeping in view all the limitations defined in first step. All necessary data is gathered about the target. Reconnaissance can be active or passive. One can directly scan the target to gather data or use other sources to get all required information. This is most important step among all. This is because, chances of successful pen-testing increase with increase in information about the target. When you know more, you can exploit more. This step gives vulnerabilities of the network as a result.

4.1.3 Exploitation

Getting information about all vulnerabilities in a network, there come the stage of actual exploitation of those vulnerabilities. In this step, target is actually attacked and penetrated using different tools and frameworks. If this step is successful, you are into the target and now you can move forward.

4.1.4 Taking Control

After successful exploitation of vulnerabilities in the target installation, you are into the target but what to do now? How to remain there? How to move forwards. For this, different activities like making backdoors, escalating privileges and gaining administrative rights are performed on target. This gives full control over the target.

4.1.4 Pivoting

Now you have full control of one process, node or host in target. Next step is to move forward. This is somewhat easy as compared to steps earlier as you are a part of target now. Now other hosts are tried for pen-testing and a full control is tried over them.

4.1.5 Reporting

After the completion of penetration testing, a proper report of it needs to be generated. The organization is told about all vulnerabilities in the target. It covers detailed information about target, its vulnerabilities, risks and methods to mitigate those risks. This step is as vital as any other step in the process.

4.2 Remote Penetration Testing

Following will be the methodology which will be employed in our Remote Penetration Testing Toolkit:

- The Penetration testing Toolkit will be installed inline or standalone in a network to be pen-tested. In this way, it will capture all of the information exchange and will penetrate into the network. It will be working on Layer 2 of communication model which is Data Link Layer.
- Remote Penetration Testing Toolkit will be implemented on Raspberry Pi and controlled through secure shell by a Command and Control Server.
- Linux will be the platform used for Pen-testing.
- Once initiated, it will perform security assessment of network using pre-defined exploits.
- At the end, a brief report will be generated.

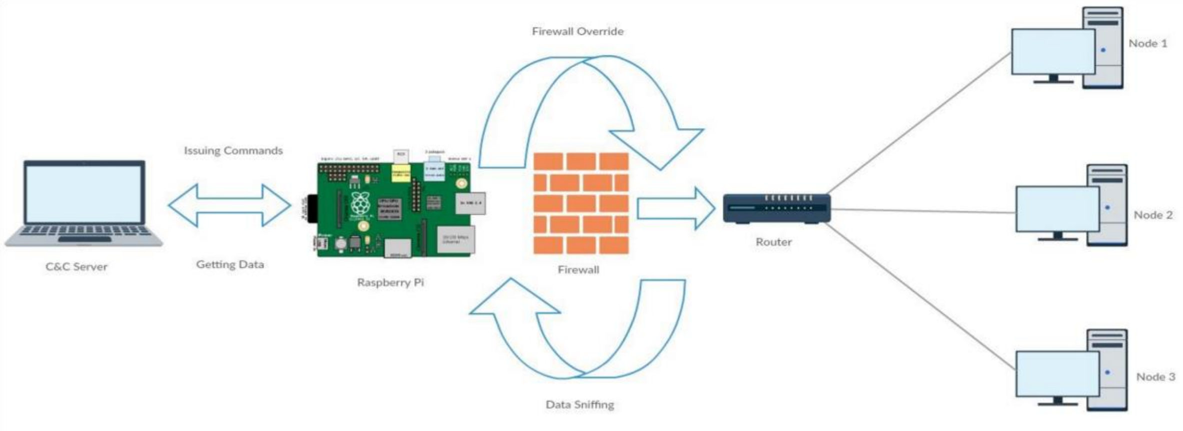


Figure 4: Network Diagram

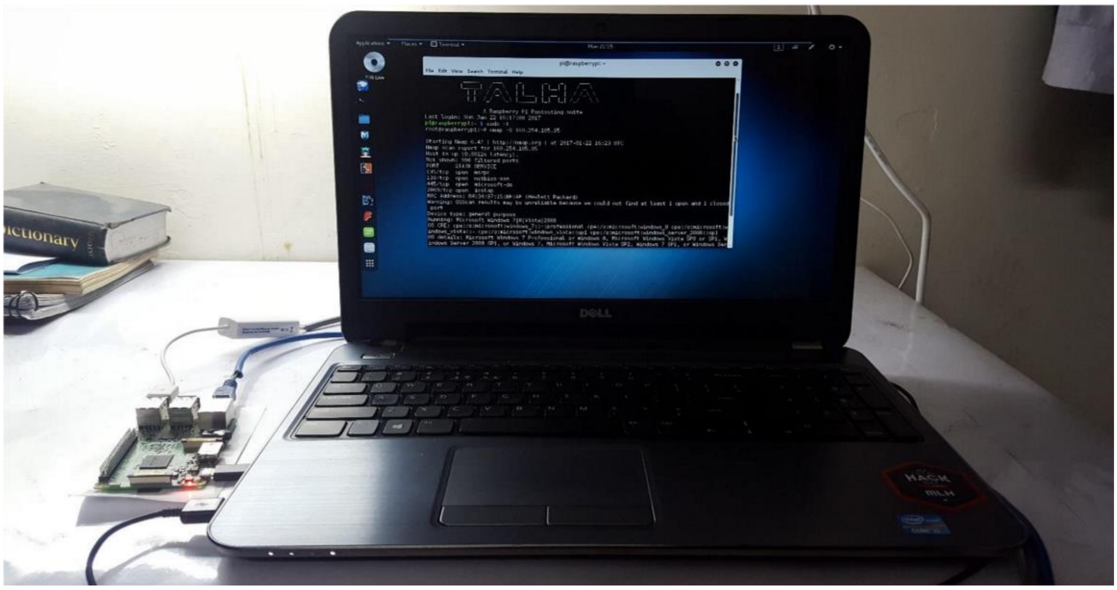
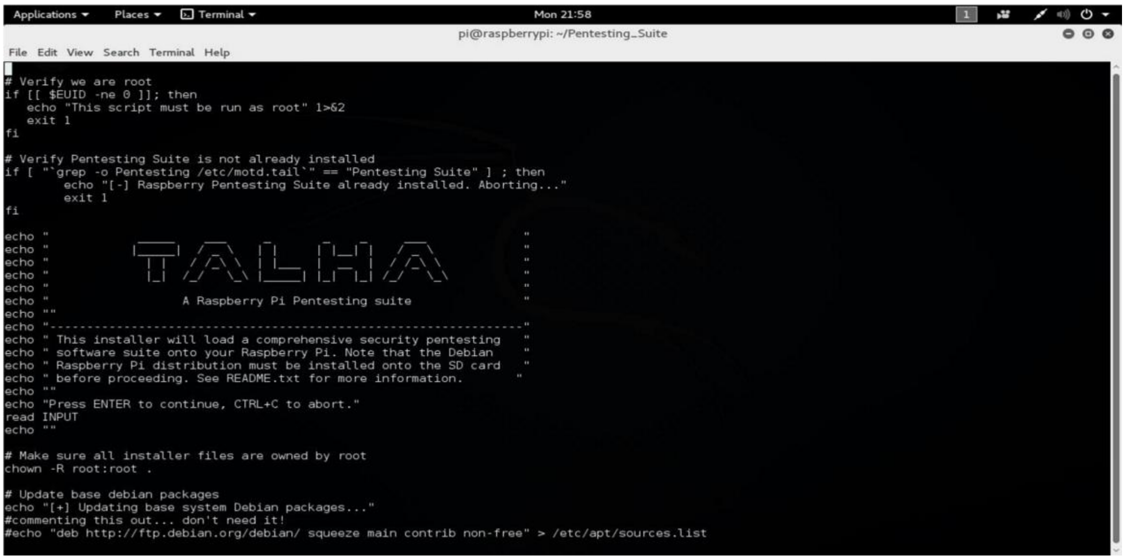


Figure 5: Raspberry Pi and C&C Server



Applications
Places
Terminal
Mon 21:58
pi@raspberrypi: ~/Pentesting_Suite
File Edit View Search Terminal Help

```
# Verify we are root
if [[ $EUID -ne 0 ]]; then
   echo "This script must be run as root" 1>&2
   exit 1
fi

# Verify Pentesting Suite is not already installed
if [ "`grep -o Pentesting /etc/motd.tail`" == "Pentesting Suite" ] ; then
        echo "[-] Raspberry Pentesting Suite already installed. Aborting..."
        exit 1
fi

echo "          _____    ___    _       _   _    ___          "
echo "         |_____|  / _ \  | |     | |_| |  / _ \         "
echo "           | |   / /_\ \ | |     |  _  | / /_\ \        "
echo "           |_|  /_/   \_\|_____| |_| |_|/_/   \_\       "
echo "                                                        "
echo "                A Raspberry Pi Pentesting suite         "
echo ""
echo "--------------------------------------------------------"
echo " This installer will load a comprehensive security pentesting   "
echo " software suite onto your Raspberry Pi. Note that the Debian    "
echo " Raspberry Pi distribution must be installed onto the SD card   "
echo " before proceeding. See README.txt for more information.      "
echo ""
echo "Press ENTER to continue, CTRL+C to abort."
read INPUT
echo ""

# Make sure all installer files are owned by root
chown -R root:root .

# Update base debian packages
echo "[+] Updating base system Debian packages..."
#commenting this out... don't need it!
#echo "deb http://ftp.debian.org/debian/ squeeze main contrib non-free" > /etc/apt/sources.list
```

Figure 6: Installation Script for Pen-Testing Suite

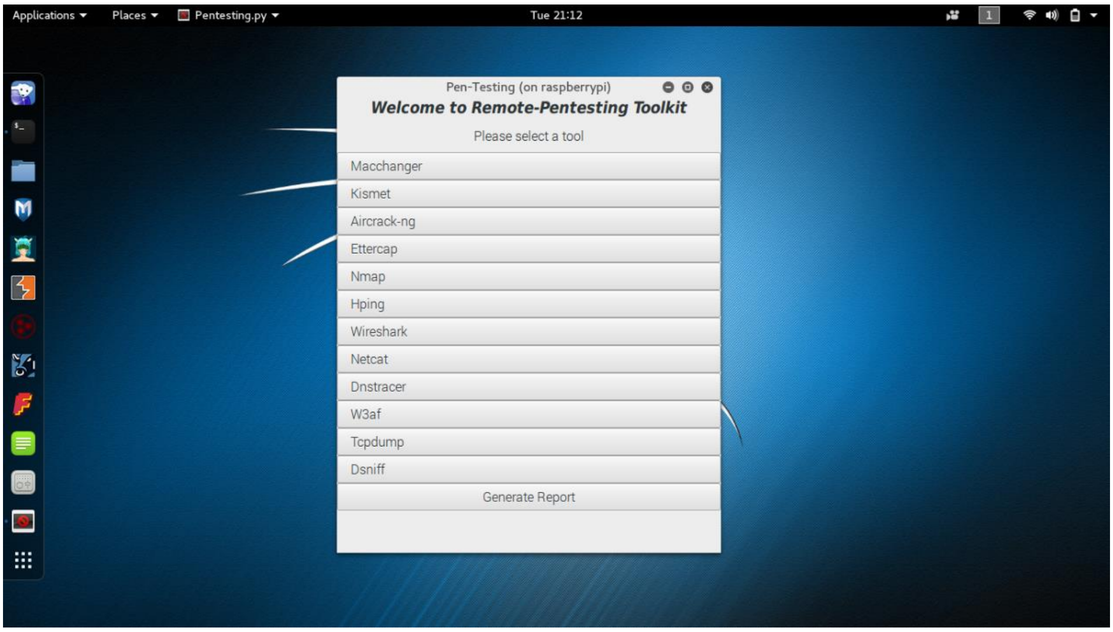


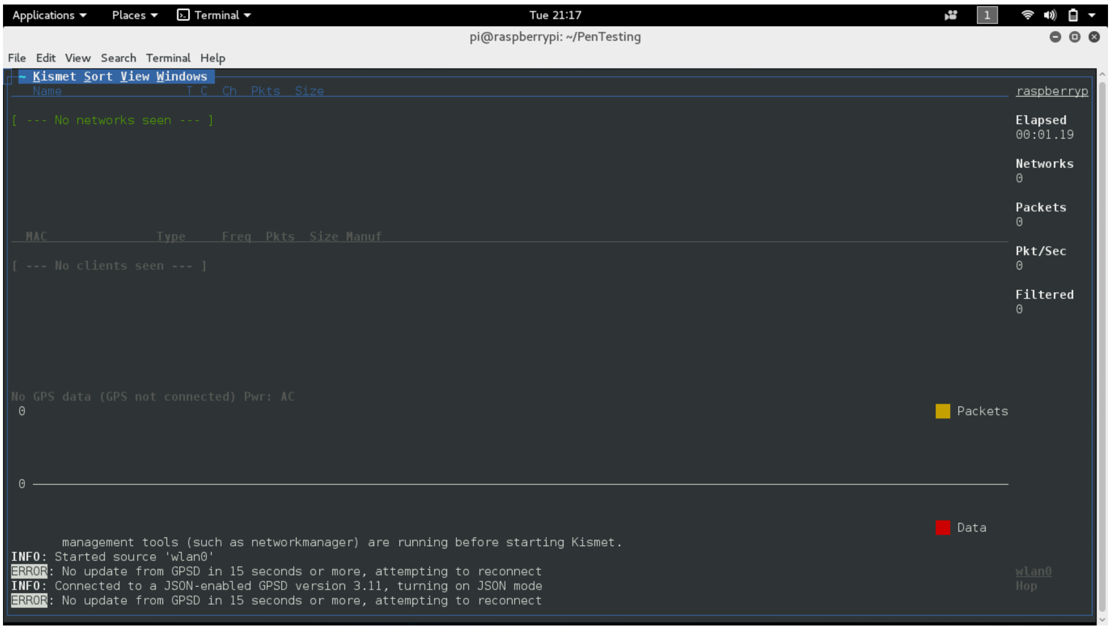
Figure 7: GUI for Pen-Testing Suite

5. ATTACKING TOOLS

Following attacking tools have been embedded in the penetration testing suite for Raspberry Pi. These attacks are customizable for a given test.

5.1 Kismet

Kismet is an open source wireless network analyzer running under the Linux systems. It is able to detect any 802.11 a/b/g wireless networks around it. 802.11 a/b/g protocols are WLAN (Wireless Local Area Network) standards. Kismet detects networks by passively sniffing providing it the advantages to discover the "hidden" wireless networks and being itselfl undetectable. The kismet program is composed by a server called "kismet_server" and a client "kismet_client" which can connect to many servers.



Applications
Places
Terminal
Tue 21:17
1
pi@raspberrypi: ~/PenTesting
File Edit View Search Terminal Help
~ Kismet Sort View Windows

```
  Name                T C  Ch  Pkts  Size                                          raspberryp

[ --- No networks seen --- ]                                                       Elapsed
                                                                                   00:01.19

                                                                                   Networks
                                                                                   0

                                                                                   Packets
                                                                                   0
  MAC                 Type     Freq  Pkts  Size Manuf
                                                                                   Pkt/Sec
[ --- No clients seen --- ]                                                        0

                                                                                   Filtered
                                                                                   0



No GPS data (GPS not connected) Pwr: AC
 0                                                                  Packets




 0 ___________________________________________________________________________



                                                                    Data
       management tools (such as networkmanager) are running before starting Kismet.
INFO: Started source 'wlan0'
ERROR: No update from GPSD in 15 seconds or more, attempting to reconnect          wlan0
INFO: Connected to a JSON-enabled GPSD version 3.11, turning on JSON mode          Hop
ERROR: No update from GPSD in 15 seconds or more, attempting to reconnect
```

Figure 8: Kismet

5.2 Aircrack-ng

Aircrack-ng is a complete suite of tools toassess Wi-Fi network security . All tools are command line which allows for heavy scripting. It works primarily Linux but also Windows, OS X etc.

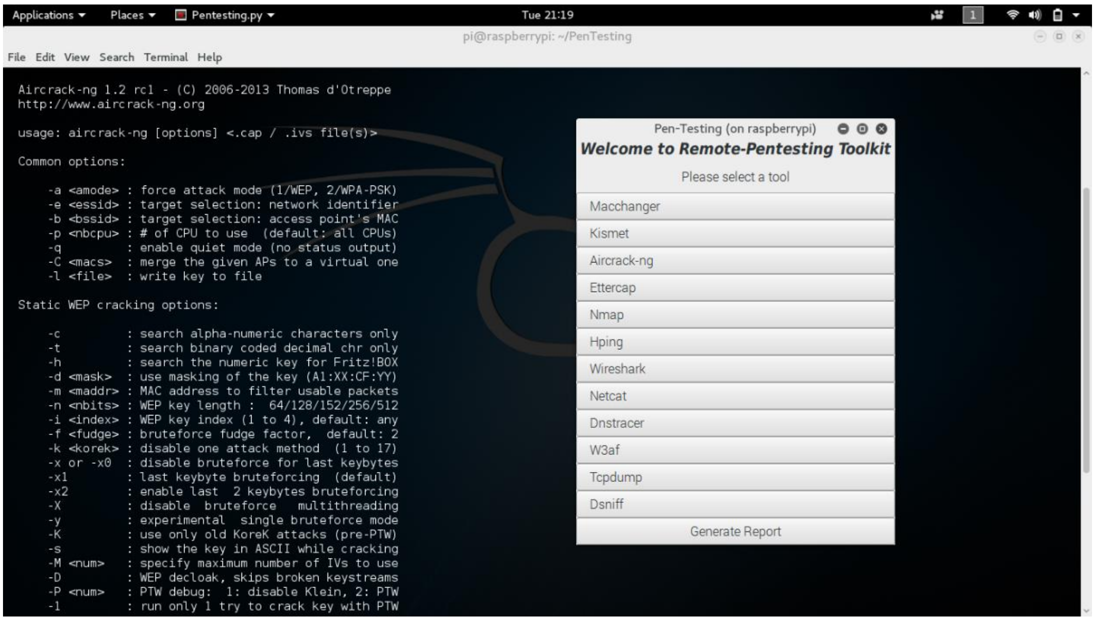
It focuses on different areas of Wi-Fi security:

Monitoring: Packet capture and export of data to text files for further process ing by third party tools.

Attacking: Replay attacks, authentication, fake access points and others via packet injection.

Testing: Checking Wi-Fi cards and driver capabilities (capture and injection).

Cracking: WEP and WPA PSK (WPA 1 and 2).

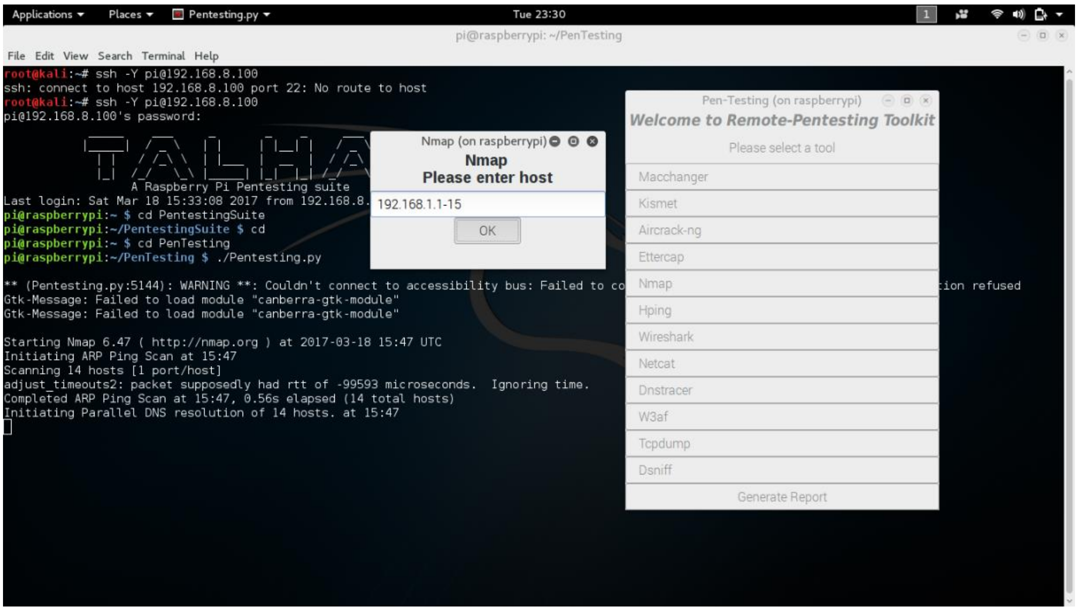


```
Aircrack-ng 1.2 rc1 - (C) 2006-2013 Thomas d'Otreppe
http://www.aircrack-ng.org

usage: aircrack-ng [options] <.cap / .ivs file(s)>

Common options:

    -a <amode> : force attack mode (1/WEP, 2/WPA-PSK)
    -e <essid> : target selection: network identifier
    -b <bssid> : target selection: access point's MAC
    -p <nbcpu> : # of CPU to use  (default: all CPUs)
    -q         : enable quiet mode (no status output)
    -C <macs>  : merge the given APs to a virtual one
    -l <file>  : write key to file

Static WEP cracking options:

    -c         : search alpha-numeric characters only
    -t         : search binary coded decimal chr only
    -h         : search the numeric key for Fritz!BOX
    -d <mask>  : use masking of the key (A1:XX:CF:YY)
    -m <maddr> : MAC address to filter usable packets
    -n <nbits> : WEP key length :  64/128/152/256/512
    -i <index> : WEP key index (1 to 4), default: any
    -f <fudge> : bruteforce fudge factor,  default: 2
    -k <korek> : disable one attack method  (1 to 17)
    -x or -x0  : disable bruteforce for last keybytes
    -x1        : last keybyte bruteforcing  (default)
    -x2        : enable last  2 keybytes bruteforcing
    -X         : disable  bruteforce   multithreading
    -y         : experimental  single bruteforce mode
    -K         : use only old KoreK attacks (pre-PTW)
    -s         : show the key in ASCII while cracking
    -M <num>   : specify maximum number of IVs to use
    -D         : WEP decloak, skips broken keystreams
    -P <num>   : PTW debug:  1: disable Klein, 2: PTW
    -1         : run only 1 try to crack key with PTW
```

Figure 9: Aircrack-ng

5.3 Nmap

Nmap ("Network Mapper") is a free and open source utility for network exploration and security auditing. Nmap uses raw IP packets in novel ways to determine what hosts are available on thel network, what ervices (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, and what type of packet filters/firewalls are in use.It was designed to rapidly scan large networks,but work fineagainst single hosts. Nmap runs on all major computer operating systems, and both console and graphical versions are available.

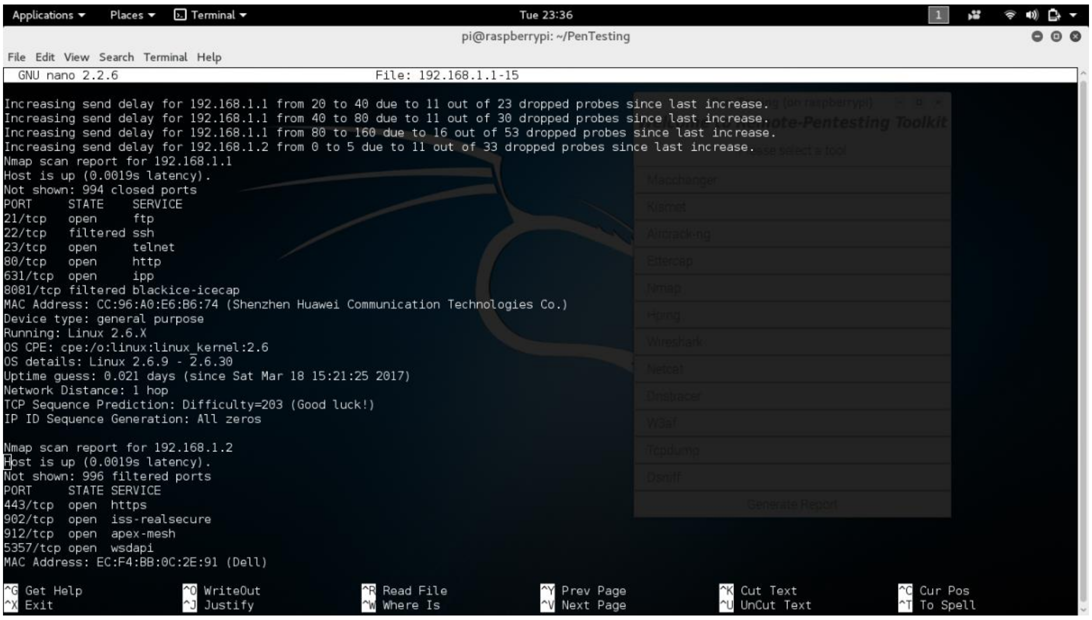


Nmap
Please enter host

192.168.1.1-15

```
root@kali:~# ssh -Y pi@192.168.8.100
ssh: connect to host 192.168.8.100 port 22: No route to host
root@kali:~# ssh -Y pi@192.168.8.100
pi@192.168.8.100's password:

             A Raspberry Pi Pentesting suite
Last login: Sat Mar 18 15:33:08 2017 from 192.168.8.
pi@raspberrypi:~ $ cd PentestingSuite
pi@raspberrypi:~/PentestingSuite $ cd
pi@raspberrypi:~ $ cd PenTesting
pi@raspberrypi:~/PenTesting $ ./Pentesting.py

** (Pentesting.py:5144): WARNING **: Couldn't connect to accessibility bus: Failed to co                    ion refused
Gtk-Message: Failed to load module "canberra-gtk-module"
Gtk-Message: Failed to load module "canberra-gtk-module"

Starting Nmap 6.47 ( http://nmap.org ) at 2017-03-18 15:47 UTC
Initiating ARP Ping Scan at 15:47
Scanning 14 hosts [1 port/host]
adjust_timeouts2: packet supposedly had rtt of -99593 microseconds.  Ignoring time.
Completed ARP Ping Scan at 15:47, 0.56s elapsed (14 total hosts)
Initiating Parallel DNS resolution of 14 hosts. at 15:47
```

Figure 10: Nmap



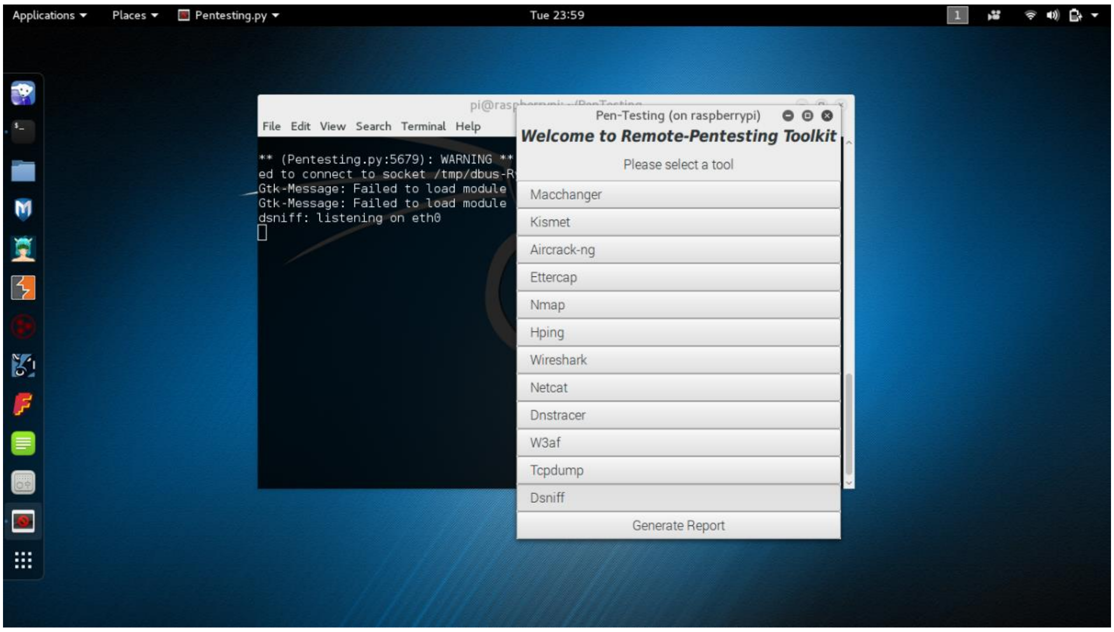
Applications
Places
Terminal
Tue 23:36
pi@raspberrypi: ~/PenTesting
File Edit View Search Terminal Help

```
  GNU nano 2.2.6                          File: 192.168.1.1-15

Increasing send delay for 192.168.1.1 from 20 to 40 due to 11 out of 23 dropped probes since last increase.
Increasing send delay for 192.168.1.1 from 40 to 80 due to 11 out of 30 dropped probes since last increase.
Increasing send delay for 192.168.1.1 from 80 to 160 due to 16 out of 53 dropped probes since last increase.
Increasing send delay for 192.168.1.2 from 0 to 5 due to 11 out of 33 dropped probes since last increase.
Nmap scan report for 192.168.1.1
Host is up (0.0019s latency).
Not shown: 994 closed ports
PORT     STATE    SERVICE
21/tcp   open     ftp
22/tcp   filtered ssh
23/tcp   open     telnet
80/tcp   open     http
631/tcp  open     ipp
8081/tcp filtered blackice-icecap
MAC Address: CC:96:A0:E6:B6:74 (Shenzhen Huawei Communication Technologies Co.)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.30
Uptime guess: 0.021 days (since Sat Mar 18 15:21:25 2017)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=203 (Good luck!)
IP ID Sequence Generation: All zeros

Nmap scan report for 192.168.1.2
Host is up (0.0019s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE
443/tcp  open  https
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
5357/tcp open  wsdapi
MAC Address: EC:F4:BB:0C:2E:91 (Dell)

^G Get Help     ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit         ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

Figure 11: Output file for Nmap

5.4 Dsniff

The ability to access the raw packets on a network interface, has long been an important tool for system and network administrators. For debugging purposes it is often helpful to look at the network traffic down to the wire level to see what is exactly being transmitted. Dsniff, as the name implies,is a network sniffer but designed for testing of different s ort.

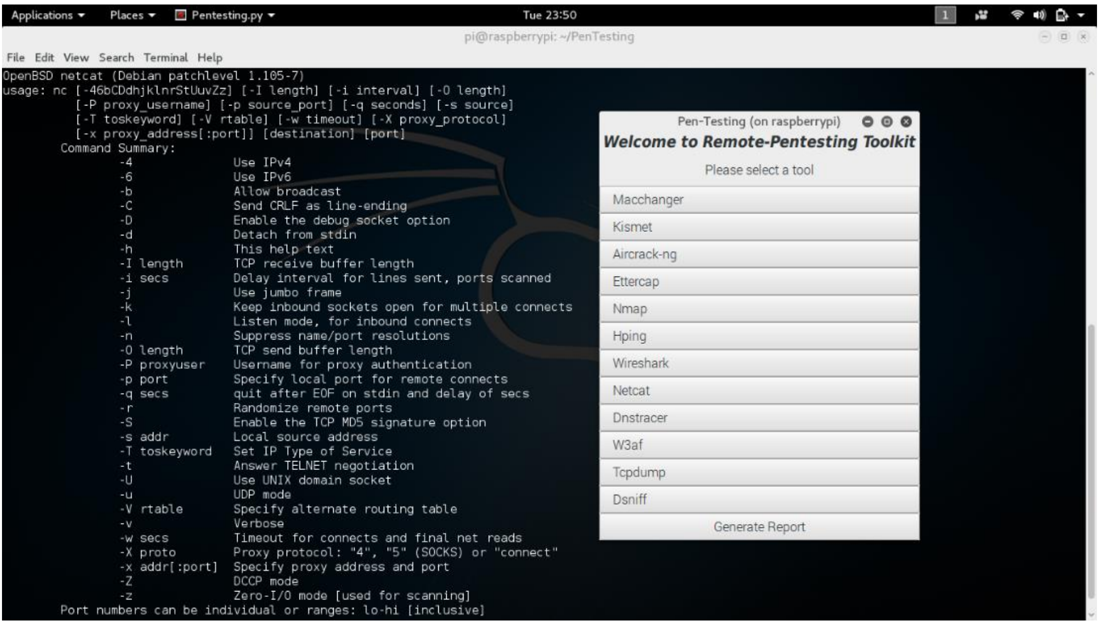


```
** (Pentesting.py:5679): WARNING **
ed to connect to socket /tmp/dbus-R
Gtk-Message: Failed to load module
Gtk-Message: Failed to load module
dsniff: listening on eth0
```

Figure 12: Dsniff

5.5 Netcat

Often referred to as a Swiss army knife of networking tools, this versatile command can assist you in monitoring, testing, and sending data across network connections. Netcat is not restricted to sending TCP and UDP packets. It also can listen on a port for connections and packets. This gives us the opportunity to connect two instances of netcat in a client-server relationship.

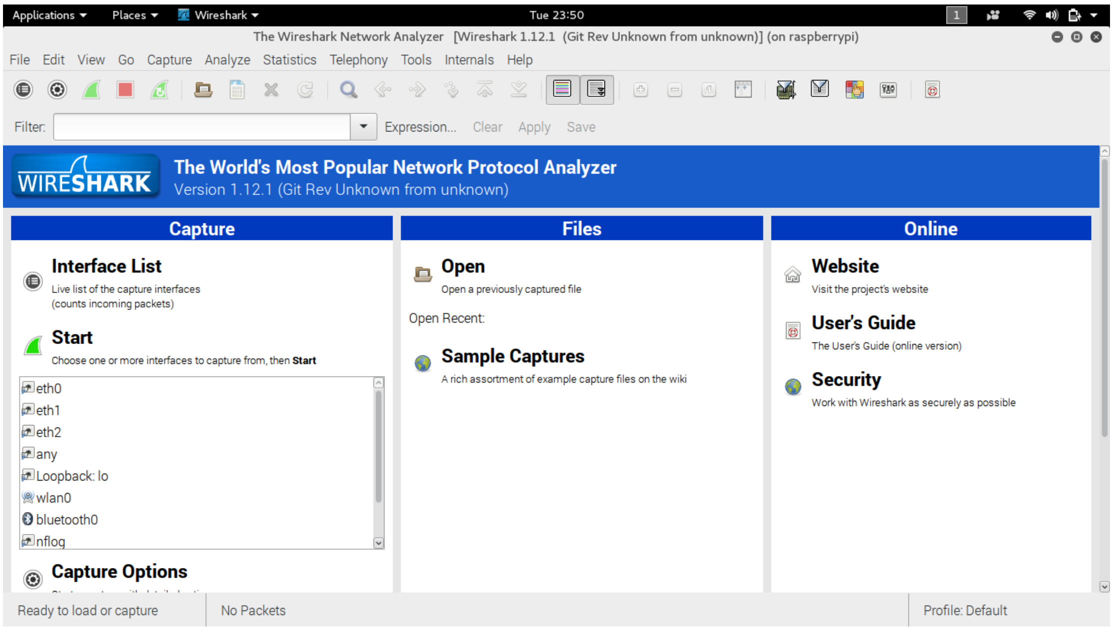


```
OpenBSD netcat (Debian patchlevel 1.105-7)
usage: nc [-46bCDdhjklnrStUuvZz] [-I length] [-i interval] [-O length]
          [-P proxy_username] [-p source_port] [-q seconds] [-s source]
          [-T toskeyword] [-V rtable] [-w timeout] [-X proxy_protocol]
          [-x proxy_address[:port]] [destination] [port]
        Command Summary:
                -4              Use IPv4
                -6              Use IPv6
                -b              Allow broadcast
                -C              Send CRLF as line-ending
                -D              Enable the debug socket option
                -d              Detach from stdin
                -h              This help text
                -I length       TCP receive buffer length
                -i secs         Delay interval for lines sent, ports scanned
                -j              Use jumbo frame
                -k              Keep inbound sockets open for multiple connects
                -l              Listen mode, for inbound connects
                -n              Suppress name/port resolutions
                -O length       TCP send buffer length
                -P proxyuser    Username for proxy authentication
                -p port         Specify local port for remote connects
                -q secs         quit after EOF on stdin and delay of secs
                -r              Randomize remote ports
                -S              Enable the TCP MD5 signature option
                -s addr         Local source address
                -T toskeyword   Set IP Type of Service
                -t              Answer TELNET negotiation
                -U              Use UNIX domain socket
                -u              UDP mode
                -V rtable       Specify alternate routing table
                -v              Verbose
                -w secs         Timeout for connects and final net reads
                -X proto        Proxy protocol: "4", "5" (SOCKS) or "connect"
                -x addr[:port]  Specify proxy address and port
                -Z              DCCP mode
                -z              Zero-I/O mode [used for scanning]
        Port numbers can be individual or ranges: lo-hi [inclusive]
```

Figure 13: Netcat

5.6 Wireshark

Wireshark is famous traffic analyzer which functions in Deep inspection of hundreds of protocols, with more being added all the time, for Live capture and offline analysis, for Standard three-pane packet browser, for Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility, Rich VoIP analysis and Capture files compressed with gzip.

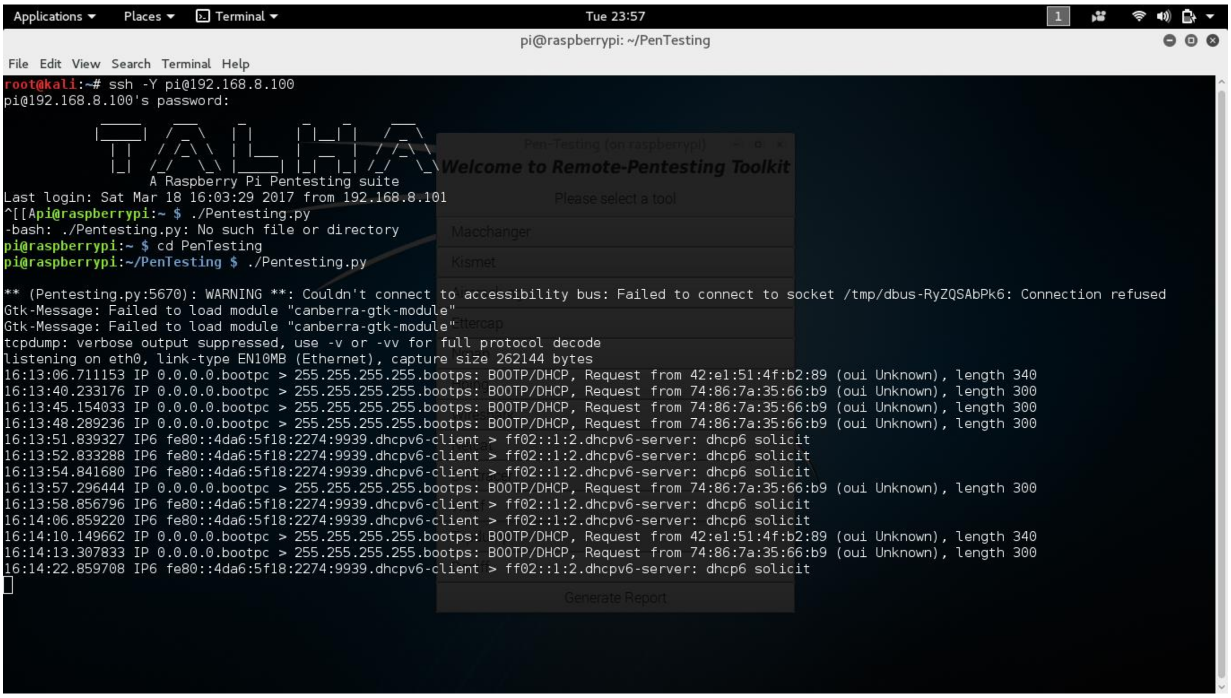


Applications
Places
Wireshark
Tue 23:50
The Wireshark Network Analyzer [Wireshark 1.12.1 (Git Rev Unknown from unknown)] (on raspberrypi)
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
WIRESHARK
The World's Most Popular Network Protocol Analyzer
Version 1.12.1 (Git Rev Unknown from unknown)
Capture
Interface List
Live list of the capture interfaces (counts incoming packets)
Start
Choose one or more interfaces to capture from, then Start
eth0
eth1
eth2
any
Loopback: lo
wlan0
bluetooth0
nflog
Capture Options
Files
Open
Open a previously captured file
Open Recent:
Sample Captures
A rich assortment of example capture files on the wiki
Online
Website
Visit the project's website
User's Guide
The User's Guide (online version)
Security
Work with Wireshark as securely as possible
Ready to load or capture
No Packets
Profile: Default

Figure 14: Wireshark

5.7 Tcpdump

Tcpdump is a common packet analyzer that runs under the command line It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached. Distributed under the BSD license. TCPDump allows you to filter network traffic for useful information.



Applications
Places
Terminal
Tue 23:57
pi@raspberrypi: ~/PenTesting
File Edit View Search Terminal Help
Pen-Testing (on raspberrypi)
Welcome to Remote-Pentesting Toolkit
Please select a tool
Macchanger
Kismet
Ettercap
Generate Report

```
root@kali:~# ssh -Y pi@192.168.8.100
pi@192.168.8.100's password:

         _____   ___    _       _   _    ___
        |_   _| / _ \  | |     | | | |  / _ \
          | |  / /_\ \ | |     | |_| | / /_\ \
          |_| /_/   \_\|_____| |_| |_|/_/   \_\
                A Raspberry Pi Pentesting suite
Last login: Sat Mar 18 16:03:29 2017 from 192.168.8.101
^[[Api@raspberrypi:~ $ ./Pentesting.py
-bash: ./Pentesting.py: No such file or directory
pi@raspberrypi:~ $ cd PenTesting
pi@raspberrypi:~/PenTesting $ ./Pentesting.py

** (Pentesting.py:5670): WARNING **: Couldn't connect to accessibility bus: Failed to connect to socket /tmp/dbus-RyZQSAbPk6: Connection refused
Gtk-Message: Failed to load module "canberra-gtk-module"
Gtk-Message: Failed to load module "canberra-gtk-module"
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
16:13:06.711153 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 42:e1:51:4f:b2:89 (oui Unknown), length 340
16:13:40.233176 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 74:86:7a:35:66:b9 (oui Unknown), length 300
16:13:45.154033 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 74:86:7a:35:66:b9 (oui Unknown), length 300
16:13:48.289236 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 74:86:7a:35:66:b9 (oui Unknown), length 300
16:13:51.839327 IP6 fe80::4da6:5f18:2274:9939.dhcpv6-client > ff02::1:2.dhcpv6-server: dhcp6 solicit
16:13:52.833288 IP6 fe80::4da6:5f18:2274:9939.dhcpv6-client > ff02::1:2.dhcpv6-server: dhcp6 solicit
16:13:54.841680 IP6 fe80::4da6:5f18:2274:9939.dhcpv6-client > ff02::1:2.dhcpv6-server: dhcp6 solicit
16:13:57.296444 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 74:86:7a:35:66:b9 (oui Unknown), length 300
16:13:58.856796 IP6 fe80::4da6:5f18:2274:9939.dhcpv6-client > ff02::1:2.dhcpv6-server: dhcp6 solicit
16:14:06.859220 IP6 fe80::4da6:5f18:2274:9939.dhcpv6-client > ff02::1:2.dhcpv6-server: dhcp6 solicit
16:14:10.149662 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 42:e1:51:4f:b2:89 (oui Unknown), length 340
16:14:13.307833 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 74:86:7a:35:66:b9 (oui Unknown), length 300
16:14:22.859708 IP6 fe80::4da6:5f18:2274:9939.dhcpv6-client > ff02::1:2.dhcpv6-server: dhcp6 solicit
```

Figure 15: Tcpdump

5.8 Netmask

Netmask is another simple tool which makes an ICMP netmask request. By determining the netmasks of various computers on a network, you can better map your subnet structure and infer trust relationships.it is a generation and conversion program. It accepts and produces a variety of common network address and netmask formats. Not only can it convert address and netmask notations, but it will optimize the masks to generate the smallest list of rules. This is very handy if you've ever configured a firewall or router and some nasty network administrator.

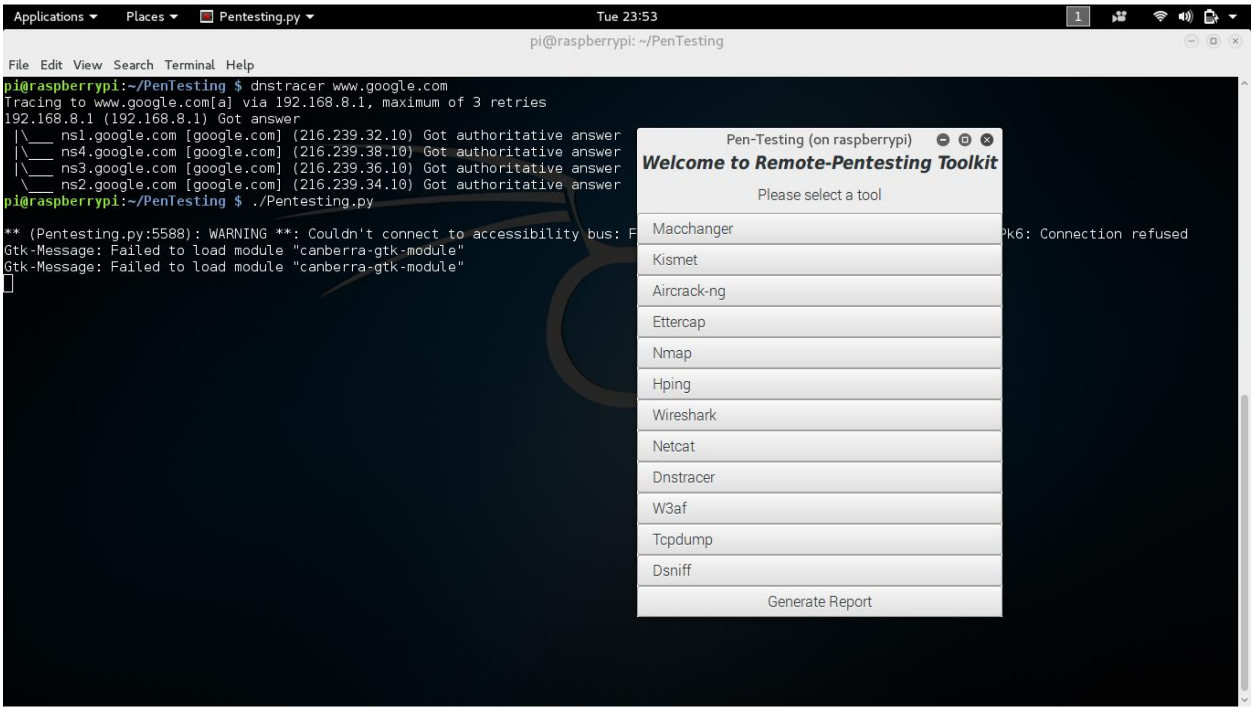
5.9 Ettercap

Ettercap is a comprehensive suite for man in the middle attacks. It features sniffing of live connections content filtering on the fly and many other interesting tricks. It supports active and passive dissection of many protocols and includes many features for network and host analysis.

Figure 16: Ettercap

5.10 Dnstracer

Dnstracer determines where a given Domain Name Server (DNS) gets its information from, and follows the chain of DNS servers back to the servers knows the data. DNSTRACER is a DNS Information gathering tool which extract unique DNS information which none other DNS tool does. It actually shows how the DNS request is processed.



```
pi@raspberrypi:~/PenTesting $ dnstracer www.google.com
Tracing to www.google.com[a] via 192.168.8.1, maximum of 3 retries
192.168.8.1 (192.168.8.1) Got answer
 |\___ ns1.google.com [google.com] (216.239.32.10) Got authoritative answer
 |\___ ns4.google.com [google.com] (216.239.38.10) Got authoritative answer
 |\___ ns3.google.com [google.com] (216.239.36.10) Got authoritative answer
  \___ ns2.google.com [google.com] (216.239.34.10) Got authoritative answer
pi@raspberrypi:~/PenTesting $ ./Pentesting.py

** (Pentesting.py:5588): WARNING **: Couldn't connect to accessibility bus: F                                        k6: Connection refused
Gtk-Message: Failed to load module "canberra-gtk-module"
Gtk-Message: Failed to load module "canberra-gtk-module"
```

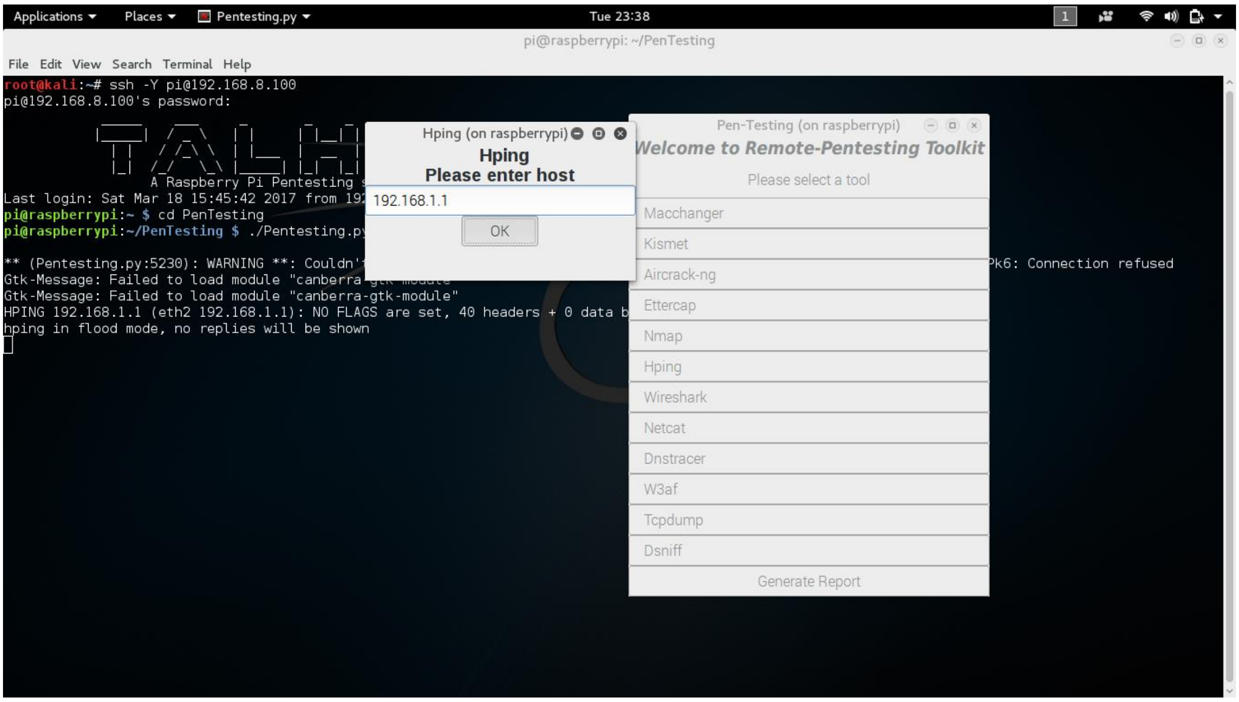
Figure 17: Dnstracer

5.11 Fping

Fping is a ping like program which uses the Internet Control Message Protocol (ICMP) echo request to determine if a host is up. Fping is different from ping in that you can specify any number of hosts on the command line, or specify a file containing the lists of hosts to ping. Instead of trying one host until it timeouts or replies, fping will send out a ping packet and move on to the next host in a round-robin fashion. If a host replies, it is noted and removed from the list of hosts to check. If a host does not respond within a certain time limit and or retry limit it will be considered unreachable.

5.12 Hping

Hping is a command-line oriented TCP/IP packet assembler/analyzer.The interface is inspired to the ping UNIX command, but hping isn’t only able to send ICMP echo requests. It supports TCP ,UDP, ICMP and RAW-IP protocols, has a traceroute mode, the ability to send files between a covered channel, and many other features. Hping was mainly used as a security tool in the past. Now hping is used for Firewall testing, advanced port scanning, Network testing, using different protocols, TOS, fragmentation and Advanced traceroute, under all the supported protocols.

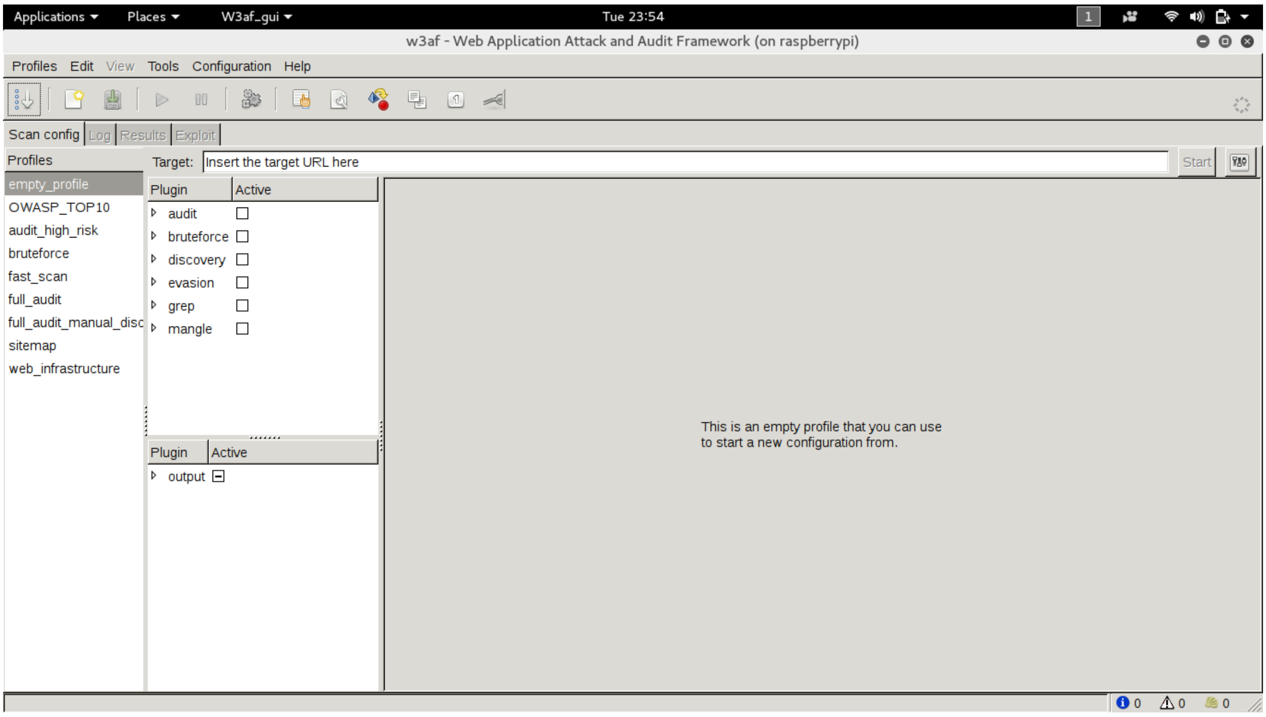


```
root@kali:~# ssh -Y pi@192.168.8.100
pi@192.168.8.100's password:

        A Raspberry Pi Pentesting s
Last login: Sat Mar 18 15:45:42 2017 from 192
pi@raspberrypi:~ $ cd PenTesting
pi@raspberrypi:~/PenTesting $ ./Pentesting.py

** (Pentesting.py:5230): WARNING **: Couldn'                                        k6: Connection refused
Gtk-Message: Failed to load module "canberra-gtk-module"
Gtk-Message: Failed to load module "canberra-gtk-module"
HPING 192.168.1.1 (eth2 192.168.1.1): NO FLAGS are set, 40 headers + 0 data b
hping in flood mode, no replies will be shown
```

Figure 18: Hping

5.13 W3af

w3af (web application attack and audit framework) is an open-source web application security scanner. The project provides a vulnerability scanner and exploitation tool for Web applications. It provides information about security vulnerabilities for use in penetration testing engagements. The scanner offers a graphical user interface and a command-line interface.



This is an empty profile that you can use
to start a new configuration from.

Figure 19: W3af

6. SCOPE

- **Testing Security Controls:** Penetration test tells us whether our system or network is exploitable or not.
- **Ensure System Security:** If we see any flaw in our system security, we can overcome that flaw in order to ensure system security.
- **Prevention from Data Breach:** Overcoming security flaws will prevent breach into our data by the hacker.
- **Banking and other networks:** Banking networks are most threatened by hackers. Pen-testing of banking networks will ensure their security.
- **Spying:** One of main scopes is spying of enemy's networks. Raspberry Pi is a very smart device. It can be hidden anywhere in the wires of network. Pen-tester will be at home monitoring enemy's network.

7. FUTURE WORK

This project covers the problem of cost and movement of manpower. Once installed in network, penetration tester has to launch attacks manually. This can be automated by writing a script by which device would be able to automatically scan, enumerate, exploit and report the network vulnerabilities.

8. REFERENCES

Lakhani, A., & Muniz, J. (2015). *Penetration Testing With Raspberry Pi.* Birmingham: Packt Publishing Ltd.

APPENDICES

Appendix A

9. Script for Install.sh

```
 #!/bin/bash
#A Raspberry Pentesting Suite

echo ""

# Verify we are root
if [[ $EUID -ne 0 ]]; then
  echo "This script must be run as root" 1>&2
  exit 1
fi

# Verify Pentesting Suite is not already installed
if [ "`grep -o Pentesting /etc/motd.tail`" == "Pentesting Suite" ] ; then
        echo "[-] Raspberry Pentesting Suite already installed. Aborting..."
        exit 1
fi

echo "         _____   ___    _      _   _    ___            "
echo "        |_____| / _ \  | |    | |__| |  / _ \          "
echo "          | |  / /_\ \ | |___ |  __  | / /_\ \         "
echo "          |_| /_/   \_\|_____||_|  |_|/_/   \_\        "
echo "                                                 "
echo "               A Raspberry Pi Pentesting suite            "
echo ""
echo "-------------------------------------------------------------"
echo " This installer will load a comprehensive security pentesting  "
echo " software suite onto your Raspberry Pi. Note that the Debian   "
echo " Raspberry Pi distribution must be installed onto the SD card  "
echo " before proceeding. See README.txt for more information.      "
echo ""
echo "Press ENTER to continue, CTRL+C to abort."
read INPUT
echo ""

# Make sure all installer files are owned by root
chown -R root:root .

# Update base debian packages
echo "[+] Updating base system Debian packages..."
#commenting this out... don't need it!
#echo "deb http://ftp.debian.org/debian/ squeeze main contrib non-free" >
/etc/apt/sources.list
aptitude -y update
aptitude -y upgrade
echo "[+] Base system Debian packages updated."

# Install baseline pentesting tools via aptitude
echo "[+] Installing baseline pentesting tools/dependencies..."
aptitude -y install telnet btscanner libnet-dns-perl hostapd nmap dsniff netcat nikto xprobe python-scapy wireshark tcpdump ettercap-graphical hping3 medusa macchanger nbtscan john ptunnel p0f ngrep tcpflow openvpn iodine httptunnel cryptcat sipsak yersinia smbclient sslsniff tcptraceroute pbnj netdiscover netmask udptunnel dnstracer sslscan medusa ipcalc dnswalk socat onesixtyone tinyproxy dmitry fcrackzip ssldump fping ike-scan gpsd darkstat swaks arping tcpreplay sipcrack proxychains proxytunnel siege wapiti skipfish w3af libssl-dev libpcap-dev libpcre3 libpcre3-dev libnl-dev libncurses-dev subversion python-twisted-web python-pymssql iw mc zip links w3m lynx arj dbview odt2txt gv catdvi djvulibre-bin python-boto python-tz pkg-config

echo "[+] Baseline pentesting tools installed."

# Remove unneeded statup items
echo "[+] Remove unneeded startup items..."
update-rc.d -f gpsd remove
update-rc.d -f tinyproxy remove
update-rc.d -f ntp remove
#apt-get -y purge portmap
#apt-get -y autoremove gdm
apt-get -y autoremove
echo "[+] Unneeded startup items removed."

# Install wireless pentesting tools
echo "[+] Installing wireless pentesting tools..."
aptitude -y install kismet
cd src/aircrack-ng-1.2-rc1
chmod +x evalrev
make install
cd ../..
airodump-ng-oui-update
echo "[+] Wireless pentesting tools installed."

# Install Metasploit -- Note this will require changing the default RAM allocation
echo "[+] Installing latest Metasploit Framework..."
aptitude -y install ruby irb ri rubygems libruby ruby-dev libpcap-dev
mkdir /opt/metasploit
wget http://downloads.metasploit.com/data/releases/framework-latest.tar.bz2
tar jxvf framework-latest.tar.bz2 -C /opt/metasploit
ln -sf /opt/metasploit/msf3/msf* /usr/local/bin/
echo "[+] Latest Metasploit Framework installed."

# Install Perl/Python tools to /pentest
```

```
echo "[+] Installing Perl/Python tools to /pentest..."
cp -a src/pentest/ /
chown -R root:root /pentest/
chmod +x /pentest/cisco-auditing-tool/CAT
chmod +x /pentest/easy-creds/easy-creds.sh
chmod +x /pentest/goohost/goohost.sh
chmod +x /pentest/lbd/lbd.sh
chmod +x /pentest/sslstrip/sslstrip.py
echo "[+] Perl/Python tools installed in /pentest."

# Install SET
echo "[+] Installing latest SET framework to /pentest..."
git clone https://github.com/trustedsec/social-engineer-toolkit/ /pentest/set/
cd src/pexpect-2.3/
python setup.py install
cd ../..
echo "[+] SET framework installed in /pentest."

# Update motd to show Raspberry Pwn release
cp src/motd.tail.raspberry /etc/motd.tail
#Update motd for pi user to show Raspberry Pentesting Suite release
cp src/motd.tail.raspberry /etc/motd

# Install Exploit-DB
echo "[+] Installing Exploit-DB to /pentest..."
mkdir -p /pentest/exploitdb
cd /pentest/exploitdb/
wget  http://www.exploit-db.com/archive.tar.bz2
tar -xjvf archive.tar.bz2
echo "[+] Exploit-DB installed in /pentest."

echo "[+] Setting default RAM allocation (disabled!)"
echo "[!] If your RPi board only has 256MB ram please set split to"
echo "    224/32 using raspi-config."
#cp /boot/arm224_start.elf /boot/start.elf

echo ""
echo "------------------------------------------------------------"
echo "Raspberry Pentesting Suite installed successfully!"
echo "------------------------------------------------------------"
echo ""


echo "[+] In order for the new RAM allocation to take effect, we must"
echo "[+] now reboot the pi. Press [Ctrl-C] to exit without rebooting."
echo ""
read
reboot
```

Appendix B

10. Script for Uninstall.sh

```
#!/bin/bash
# A Raspberry Pentesting Suite

echo ""

# Verify we are root
if [[ $EUID -ne 0 ]]; then
  echo "This script must be run as root" 1>&2
  exit 1
fi

echo "          _____   ___    _     _   _    ___           "
echo "         |_____|  / _ \   | |     | |__| |   / _ \         "
echo "           | |   / /_\ \  | |___  |  __  |  / /_\ \        "
echo "           |_|  /_/   \_\ |_____| |_|   |_| /_/   \_\        "
echo "                                            "
echo "        === Raspberry Pentesting Suite UNINSTALLER ===       "
echo ""
echo "------------------------------------------------------------"
echo " This UNINSTALLER will remove the Raspberry Pwn pentesting      "
echo " software suite from your Raspberry Pi.                    "
echo ""
echo "Press ENTER to continue, CTRL+C to abort."
read INPUT
echo ""

echo "[+] Removing baseline pentesting tools/dependencies..."
aptitude -y remove nmap dsniff netcat nikto xprobe python-scapy wireshark tcpdump ettercap hping3 medusa macchanger nbtscan john ptunnel p0f ngrep tcpflow openvpn iodine httptunnel cryptcat sipsak yersinia smbclient sslsniff tcptraceroute pbnj netdiscover netmask udptunnel dnstracer sslscan medusa ipcalc dnswalk socat onesixtyone tinyproxy dmitry fcrackzip ssldump fping ike-scan gpsd darkstat swaks arping tcpreplay sipcrack proxychains proxytunnel siege wapiti skipfish w3af libssl-dev libpcap-dev libpcre3 libpcre3-dev libnl-dev libncurses-dev subversion python-twisted-web python-pymssql git mc zip links w3m lynx arj dbview odt2txt gv catdvi djvulibre-bin python-boto python-tz

echo "[+] Removing wireless pentesting tools..."
aptitude -y remove kismet
cd src/aircrack-ng-1.2-rc1
make uninstall
```

```
cd ../..

# Remove /pentest
echo "[+] Removing /pentest..."
rm -rf /pentest

# Restore original motd
cp src/motd.tail.original /etc/motd.tail
# Restore original pi user motd
cp src/motd.tail.original /etc/motd

echo ""
echo "--------------------------------------------------"
echo "Raspberry Pentesting Suite UNINSTALLED successfully!"
echo "--------------------------------------------------"
echo ""
exit 1
```