Zachary R. Heaton **01- 03-June-2020**
Project WriteUp - Aircrack-Wifi & Intelligence Extraction

This project was put together in order to discover how a wifi network with poor security could be exploited and then at risk of data extraction, ransomware, or other malicious activity. It also serves to prove a point that network security is not a subject to take lightly in business or home planning because the tools available make exploitation only that much more accessible to threat actors.

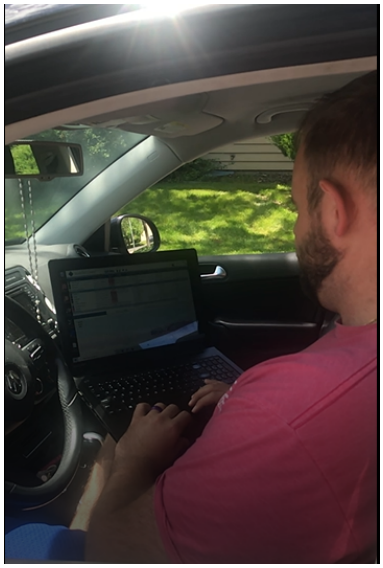
Scenario: A threat actor can drive to different residential neighborhoods or commercial sites and perform a simple WIFI scan. In the correct context a target could be selected for a number of reasons. In the example below, Michael is a government contractor that the threat actor had tailed to his home. Once Michael's home address was known the threat actor was able to scan and discover Michael's home wifi network and begin the process of gaining access and stealing critical intelligence.

*********ENTIRE SCENARIO IS FICTION*******ALL EVIDENCE IS SIMPLE GOOGLE IMAGE SEARCH*********

Step 1 - Identifying the target.

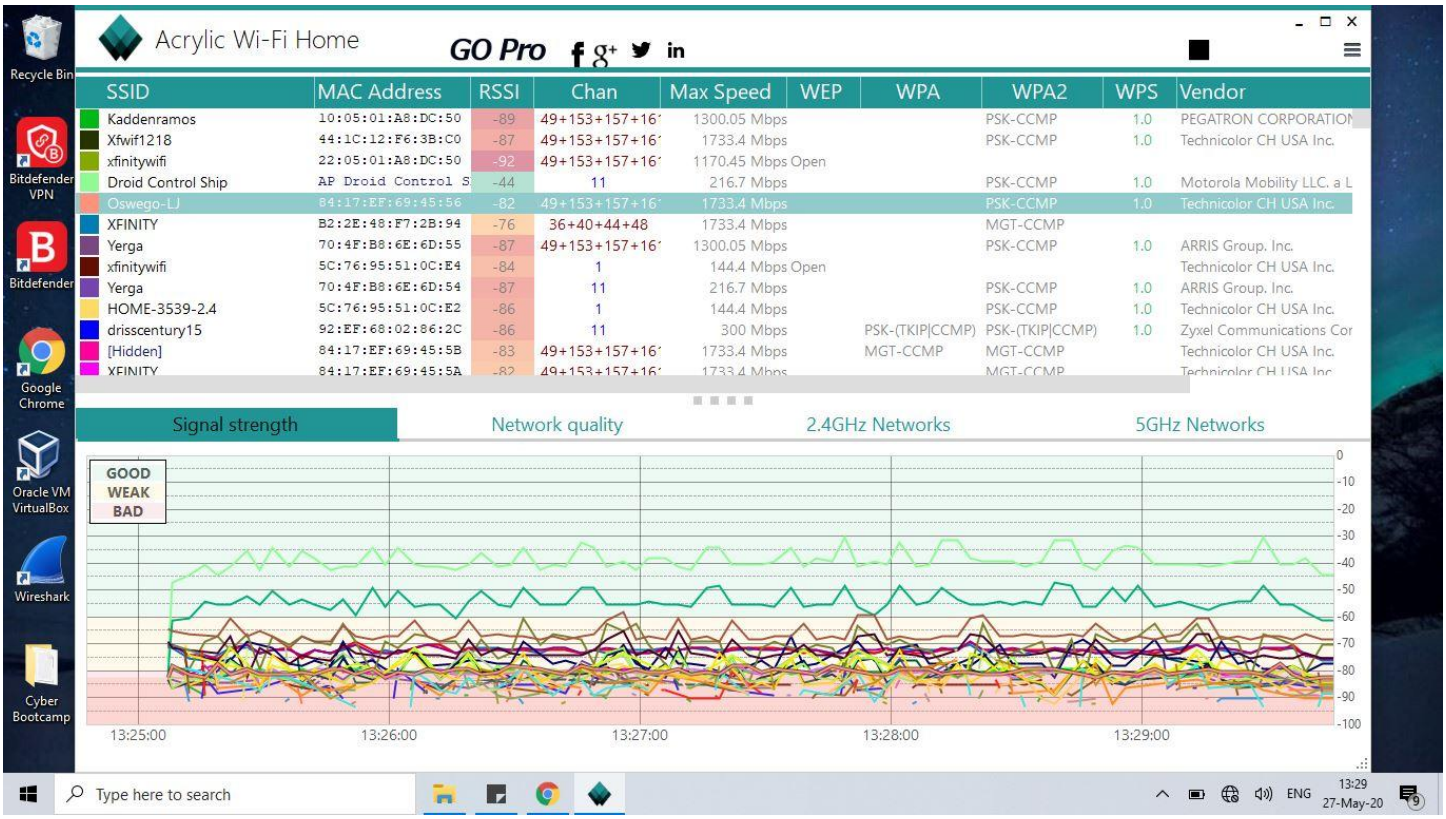
The stage is the traditional process of espionage. The target can be followed from work or otherwise monitored. Once the address is discovered the technical surveillance can be done in almost complete secret. The below pictures demonstrate and model that a person can be monitoring your home wifi network from the comfort of a parked car.





There are a few excellent tools for wifi scanning such as Wireshark and NetGrab. This project utilized one called Acrylic because of how fast it loads. Given the scenario it is preferable upon curb pull up to be able to scan as quickly as is possible. Clicking the name Acrylic will link to the website for more information on the service. Acrylic works by using the Network Adapter on the laptop to begin an active scan of any and all networks within range. Wifi networks run on a single wave-length of 2.4Gz or 5Gz normally and it is for this reason that networks can be detected. In a simpleton explanation it is similar to Newton's glass pyramid and how light reflects into ROYGBIV. Similar concepts work with radio signals in their own light (pun-intended).

The surveillance of Michael before this indicated that he is a huge fan of Star Wars and for that reason it was not difficult to identify a network that likely belonged to his house. A screenshot of Acrylic is shown below of the scan where Michael's network "Droid Control Ship" was detected.



Step 2 - Obtain Network Credentials

If the identity of Michael's wifi network was already known the above steps could be skipped. The rationale for including them is because Acrylic loads so fast on most devices that it makes for an excellent scanning tool. The next steps are about gaining access to Michael's network.

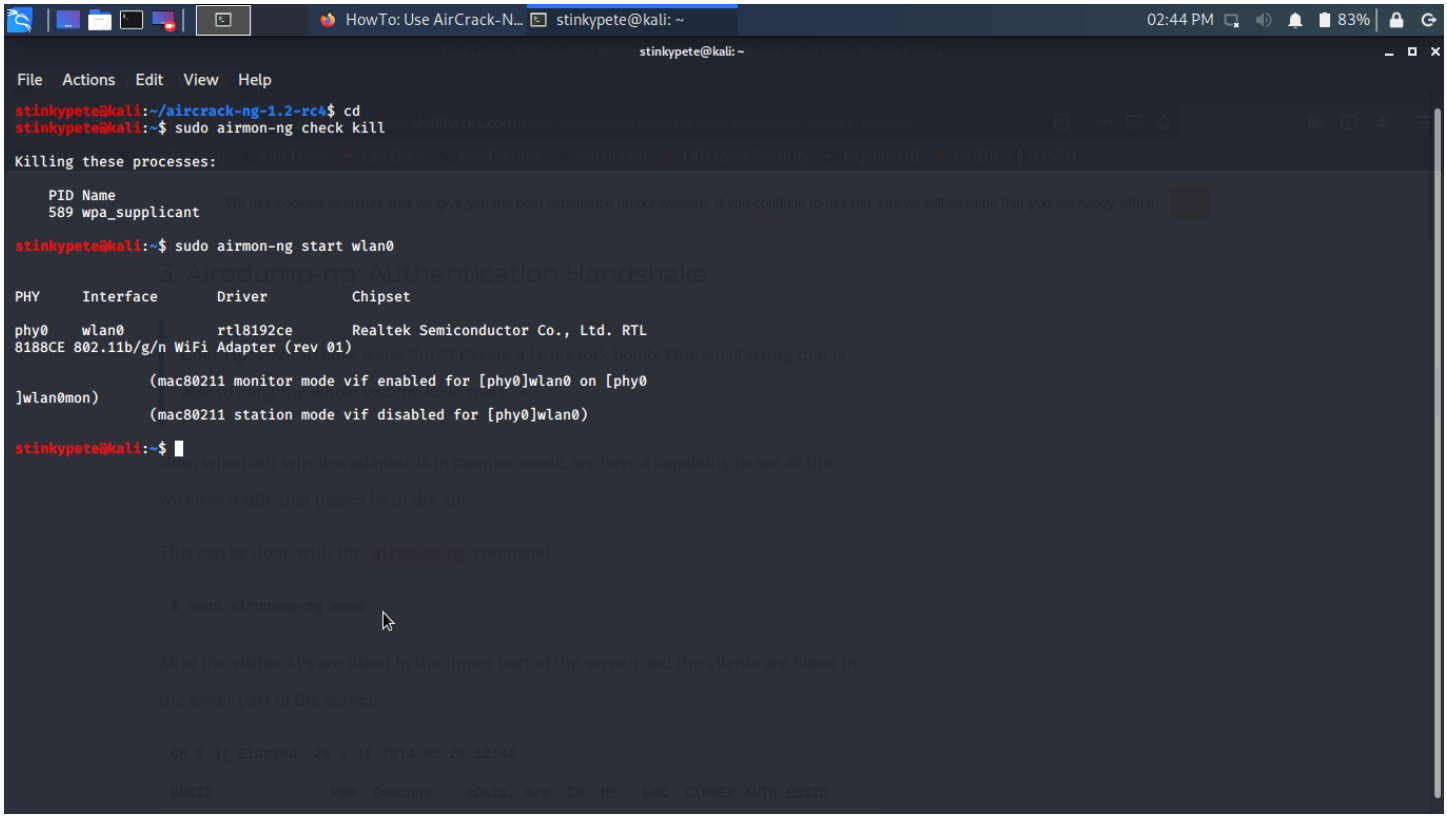
This will start with a suite of tools called Aircrack-ng. NG is shorthand for "new generation" as the newer version is far more advanced than the older version. The general idea of how Aircrack-ng works is by capturing authentication packets. Packets and internet transfer protocols are out of scope of this document but the general idea is that there are specific packets that carry login credentials. This is how the wifi router knows whether or not to permit a particular device access to its services. Those packets are transferred when someone logs onto the wifi or a device is connected and reconnected. Aircracking works by capturing that data and then using tools that will be explained later to crack the password into plain text.

Any computer could be theoretically used for Aircracking but one of the better operating systems is Kali Linux simply because of the extra tools already installed. Aircrack-ng may have to be downloaded using the following commands:

```
$sudo apt-get install build-essential libssl-dev libnl-3-dev pkg-config libnl-genl-3-dev
$wget http://download.aircrack-ng.org/aircrack-ng-1.2-rc4.tar.gz  -O - | tar -xz
$ cd aircrack-ng-1.2-rc4
$ sudo make
$ sudo make install
```

Once Aircrack-ng is installed it is a good idea to ensure that the tools are all working properly. One of the tools is “Airmon” which is short for “air monitoring”. This is the same thing that Acrylic did in stage one and if performed here it will confirm that Aircrack-ng is connected to the network hardware and functioning properly. *It should be noted that aircrack-ng will not work properly from a virtual machine unless there is a usb internet dongle or similar setup.* The reason for the service not working with a virtual machine is because of the way that a type 2 hypervisor interacts with the hardware. Again, a great start for a google search but out of scope for this document.

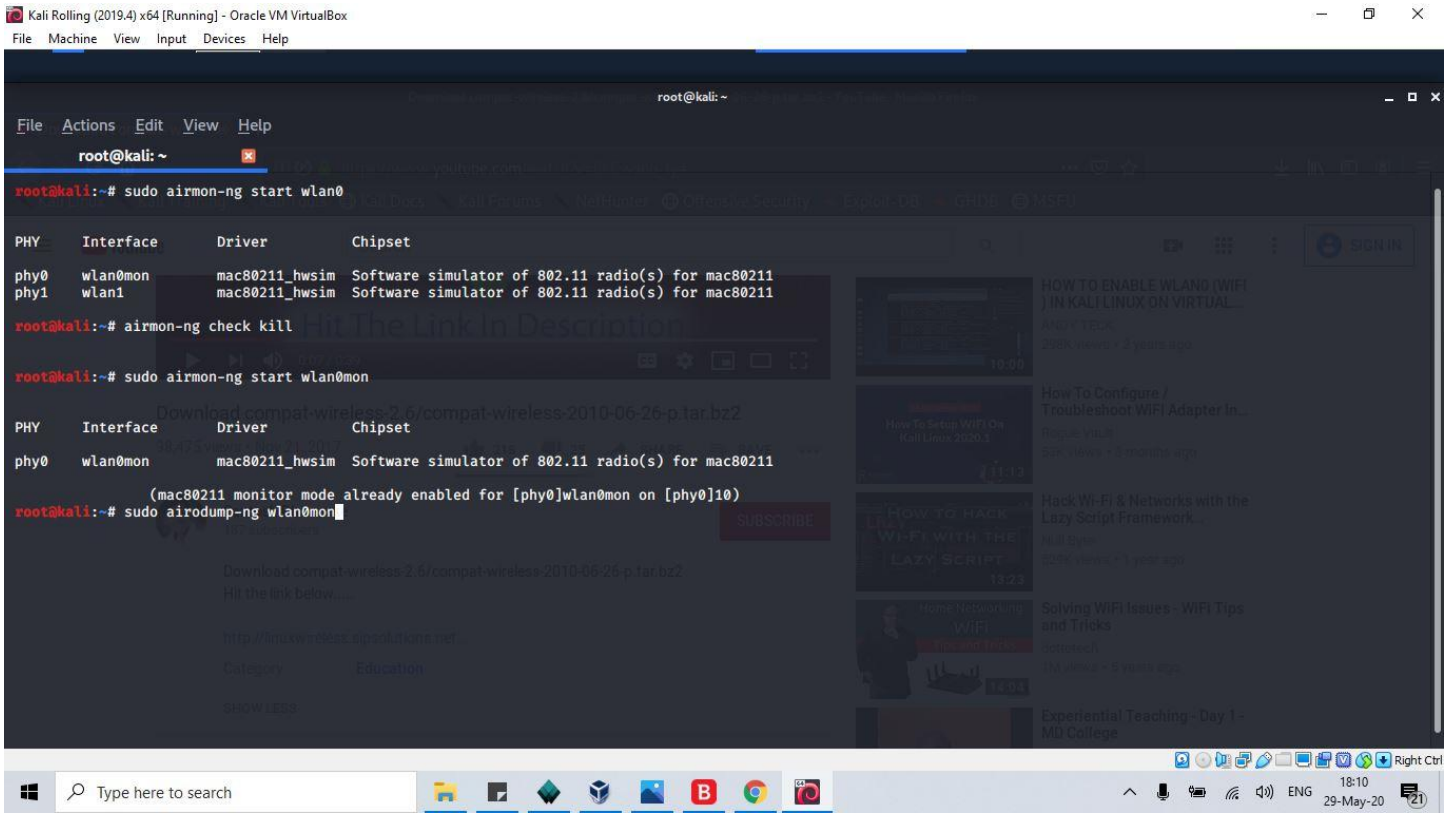
The screenshots below articulate the process of discovering the BSSID. This is sometimes the same as the MAC address but it can differ in advanced setups. The general idea is that “Basic Service Set Identifiers” (BSSID) identifies the connection whereas the MAC address identifies a piece of hardware. Often the same but it is not best practice to become accustomed to that definition. It can differ under correct circumstances.



Screenshot 1

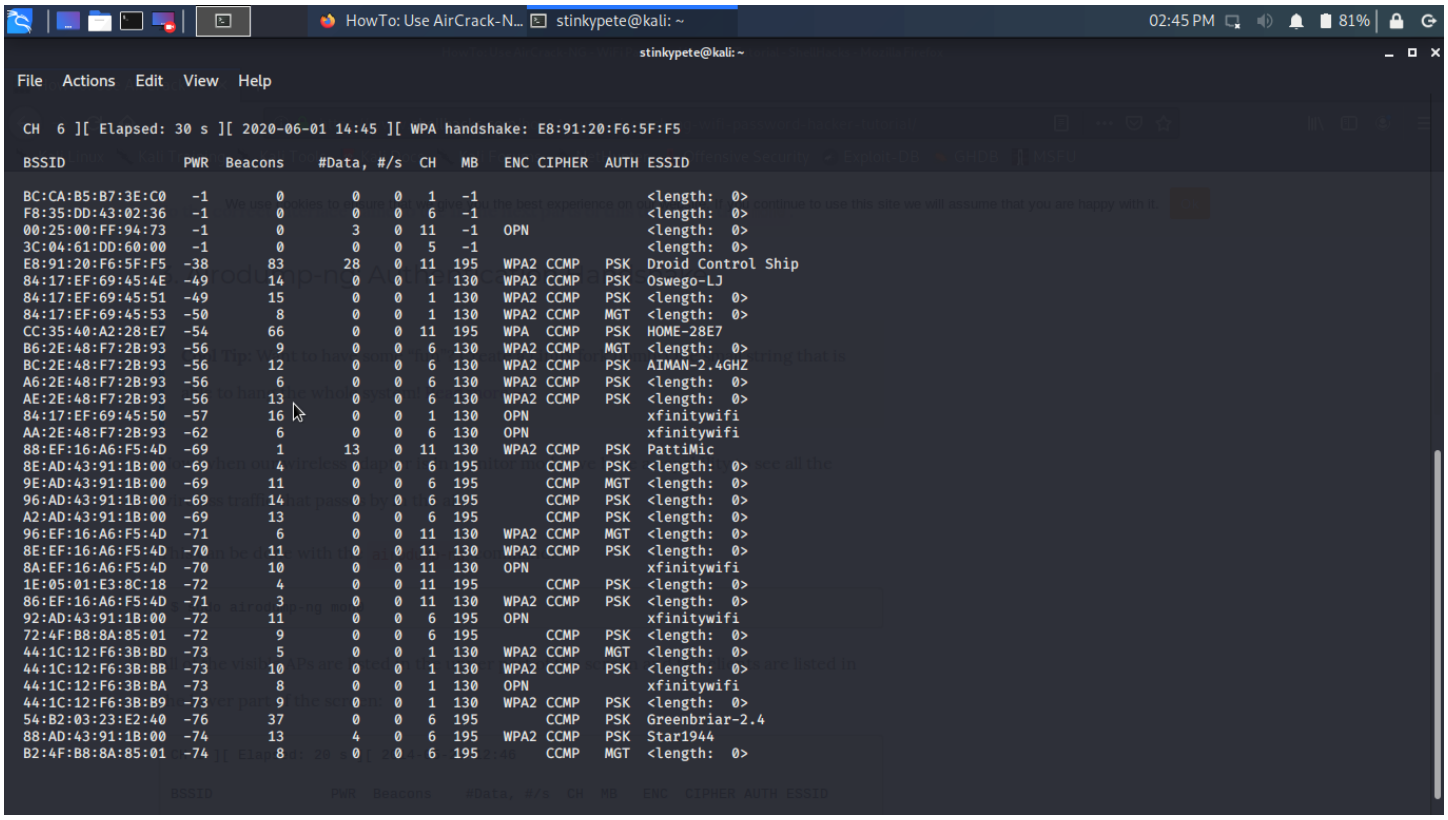
Screenshot 1 shows the command **[sudo airmon-ng check kill].** This ensures that there are no other uses of the network adapter other than the aircrack-ng suite. It should be known that there is most likely a way to reverse this but during the process of this project if internet connectivity is desired it is usually a system restart that is required. The next command should be **[sudo airmon-ng start wlan0].** This command will provide information about where the network adapter is connecting. It is visible in screenshot one that this particular machine was connecting on wlan0mon. This will be important information for the next command.

Screenshot 2 below shows the command **[sudo airmon-ng start wlan0mon].** Once executed the terminal will chance and display the BSSID of “Droid Control Ship”, Michael’s network. It will also display the other networks in range but keep in mind your local, state, and federal laws regarding unlawful access. This project was performed within a private network setting with all legal considerations attended.



Screenshot 2

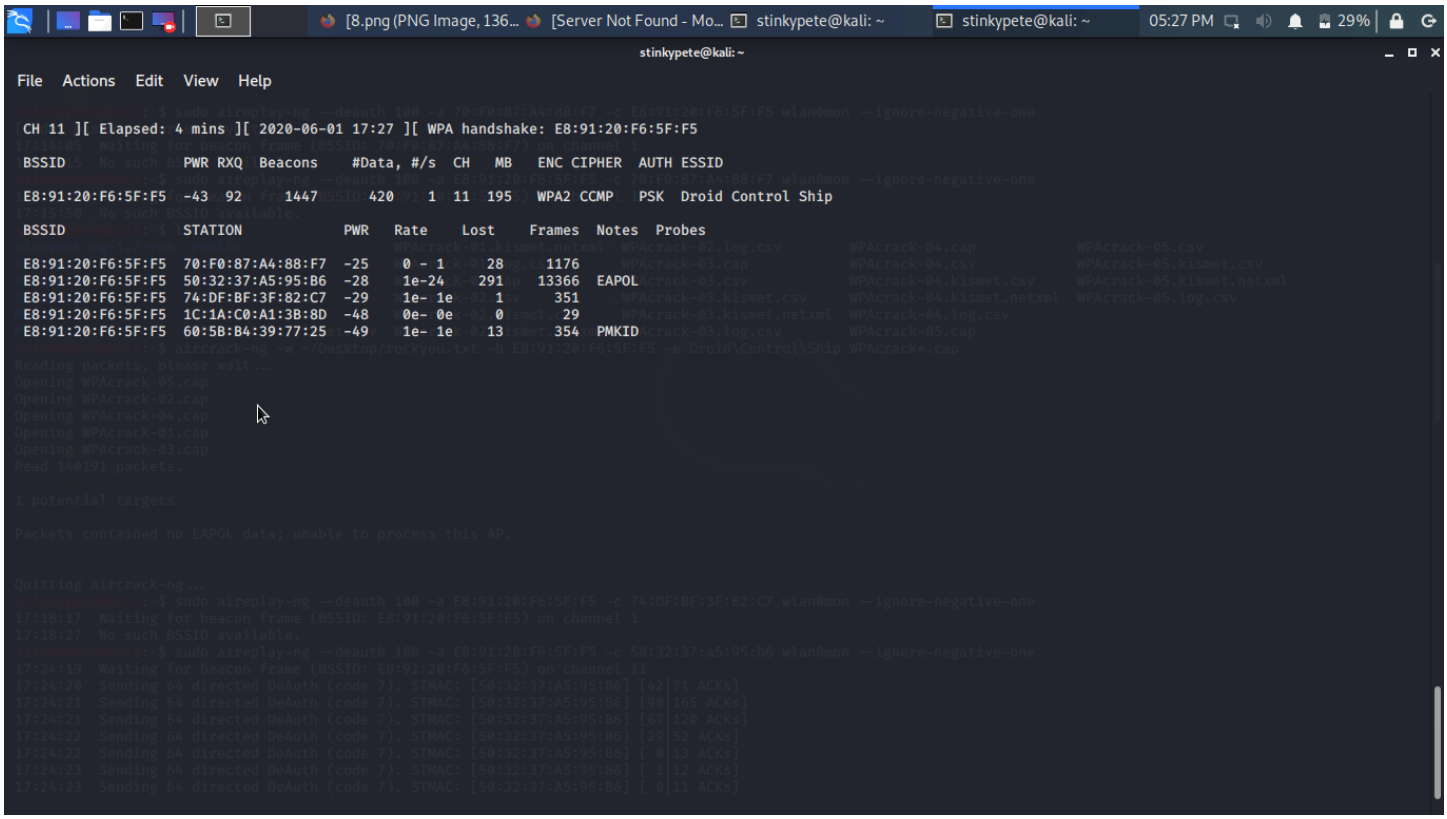
Screenshot 3 shows that the BSSID of “Droid Control Ship” is E8:91:20:F6:5F:F5. It also reveals that it is operating on channel 11. This information will be important in the following steps.



Screenshot 3

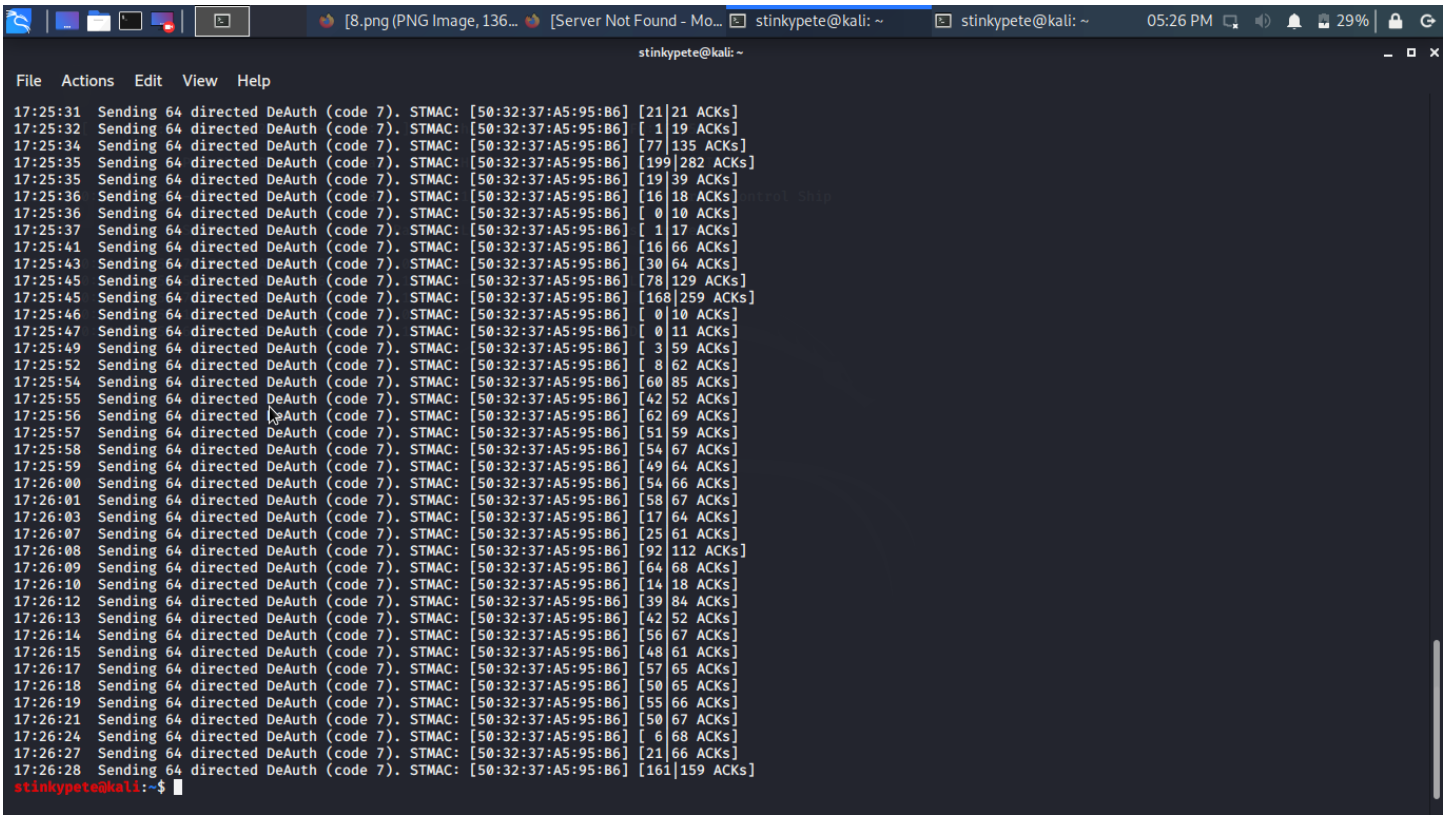
The next steps could be done quietly or aggressively depending on the footprint the attacker wishes to leave behind. At this stage airmon will now be turned over to the airodump. Airodump is a tool that actively dumps packets into a .cap file which will include authentication packets. As described above, those authentication packets will have the data needed to crack Michael's wifi password. These can be obtained in a few different ways. The most organic and secretive manner is to simply wait a long enough time where someone has logged onto the wifi network. This leaves virtually no trace because the airodump is simply listening and grabbing. This method runs the risk of capturing a massive .cap file that may or may not have any authentication packets in it.

If patience is not an option there are ways to speed up the process. Just as there are authentication packets there is such a thing as a deauthentication command. This requires a specific client (a device on the network) which the attacking computer will pummel with x amount of deauthentication commands. Twenty five deauthentication packets is usually sufficient. These packets are transmitted and force whichever client is elected to sign off of the network. When the packets stop the device will attempt to reconnect and therefore airodump can pick up one or multiple authentication packets. This all starts with the command **[$ sudo airodump-ng -c 1 --bssid E8:91:20:F6:5F:F5 -w WPAcrack wlan0mon --ignore-negative-one].** It should be noted that the bssid will change to the results depending on the target. This particular network was through that router signal and that particular interface (wlan0mon). Screenshot 4 below shows the results of the airodump command. It is obvious that there are 5 machines that are visibly connected to the network. There could possible be more but off of a basic scan there the five to choose from. If the organic path has been chosen a good sign that an authentication packet has been captured is when either one of these disappears and reappears to if an entirely station appears.



Screenshot 4

In this scenario, Michael is under suspicion for selling government secrets. There is no time to wait around for a device to connect. Deauthorization packets, as described above, will be sent in order to quickly capture the authentication packets when the device reconnects. This is done *IN A SECONDARY TERMINAL* with the command **[sudo aireplay-ng --deauth 25 -a E8:91:20:F6:5F:F5 -c 50:32:37:A5:95:86 mon0 --ignore-negative-one].** It should be remembered that some of that input is machine specific. The -a is the particular BSSID, -c is the client which will be forced to deauthorize, and the wlan0mon may or may not be different under different parameters. The output should look something like what is screenshot 5. Remember that 25 is a good quick number that provides a good chance at collecting a packet. A higher number is an option but during that time there is no internet connectivity to the device and the user may notice. Screenshot 5 below shows what this output should look like.

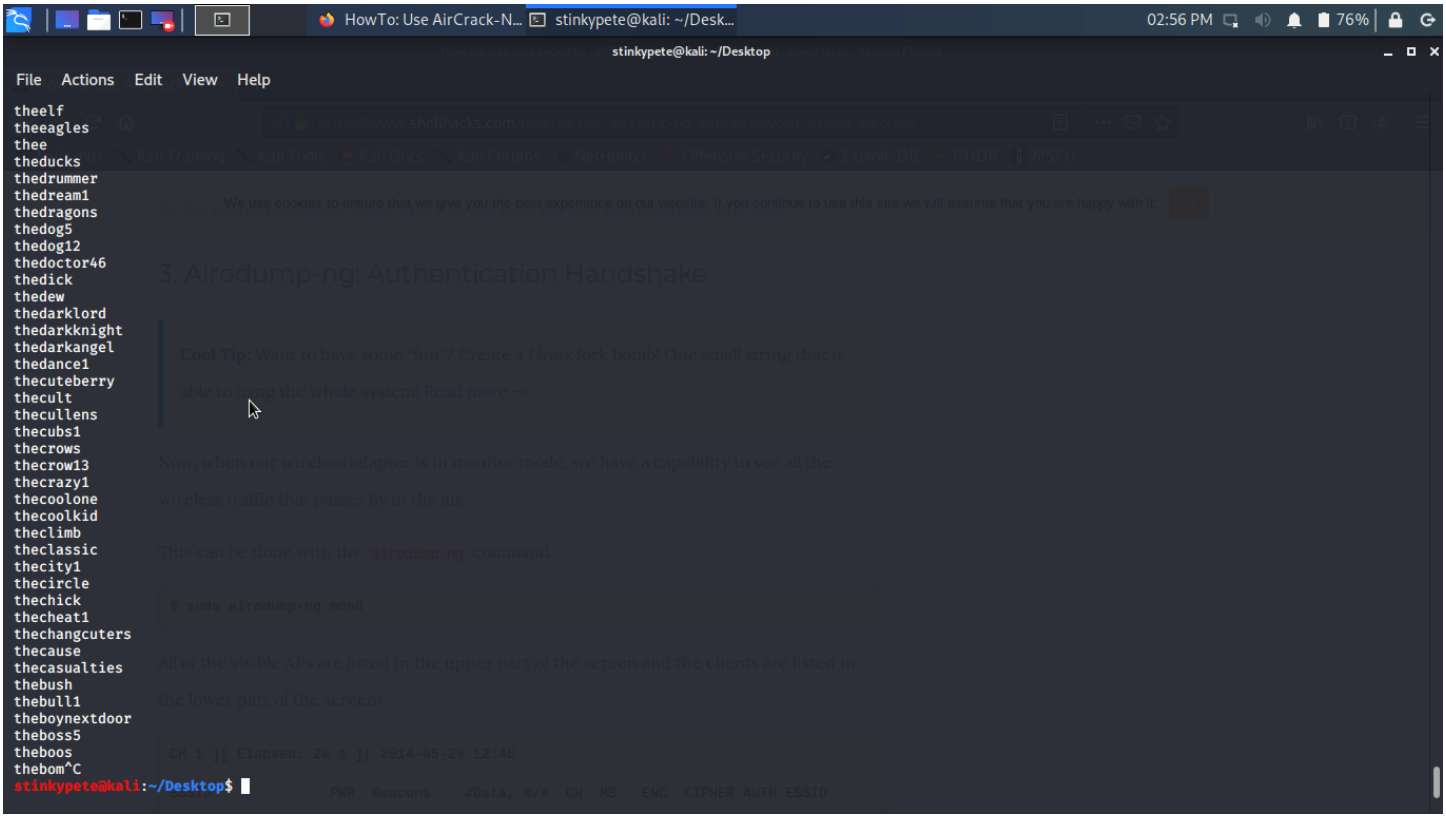


Screenshot 5

Once those deauthentication packets have been sent the next step will be to return to the terminal where the airodump is capturing traffic to close out the airodump. In order to do this use the command **[ctrl + c].** The next command should be a simple **[ls]** where it will be obvious that a .cap file has been captured. This file is where all the internet traffic is stored and could be analyzed further if desired. In this scenario the goal is the login credentials. Aircrack-ng is on deck with only one further step before possibly cracking Michael's password. That is to explain what is and the need to have a wordlist such as "rockyou.txt".

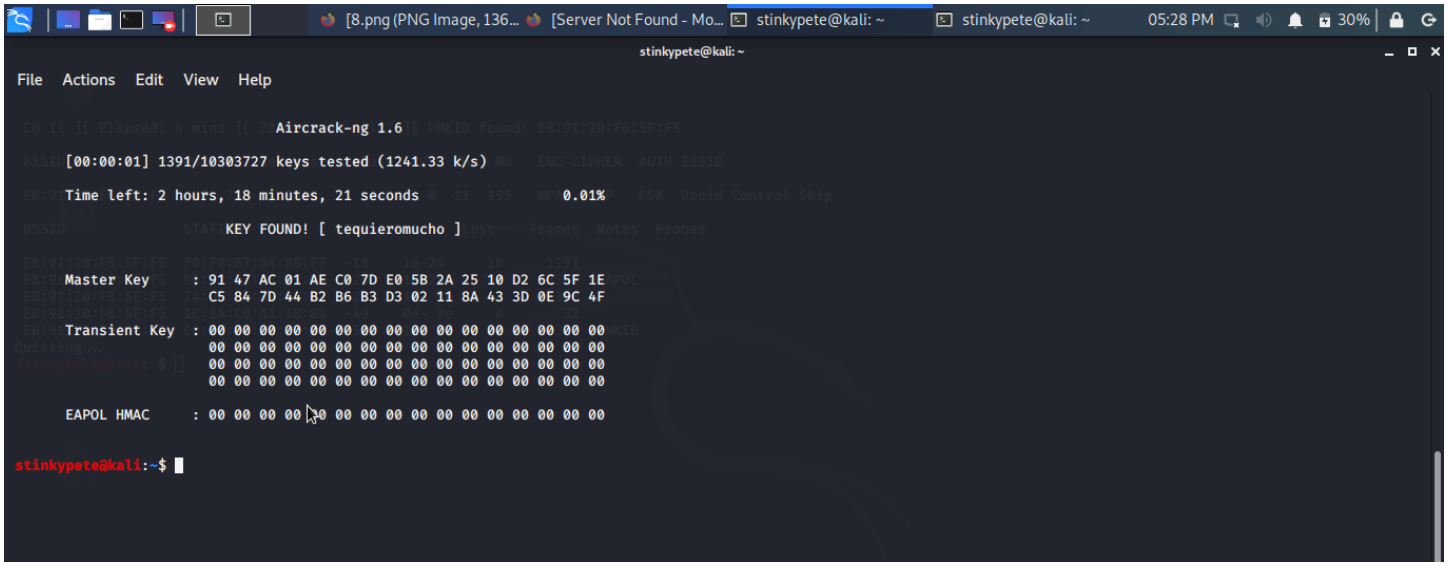
The method by which the most popular password cracking tools operate (john the ripper, hydra, aircrack-ng) is to use a word list. These are text files that can be found online and typically contain thousands or millions of the most common passwords beyond the top five passwords that could be cracked using social engineering techniques.

One of the better wordlists comes pre-installed on Kali-Linux and is titled "rockyou.txt". A simple **[locate]** command can find it and possibly a **[gunzip]** command to unzip. Screenshot 6 below shows this file.



Screenshot 6

Screenshot 6 only shows some of the “rockyou.txt” file. In reality it contains 4.3 million common passwords that can be sent through tools like John the Ripper, Hydra, or this case:Aircrack-ng. With this file the only other step is to use Aircrack to essentially combine the .cap, wordlist, and bssid in order to provide the output which should be Michael’s password. The command to do this is **[$ aircrack-ng -w ~/Desktop/rockyou.txt -b E8:91:20:F6:5F:F5 WPAcrack-01.cap].** It is important again to remember that the path for rockyou.txt, bssid, and file name of the .cap file will be user specific each time this task is performed. The output of aircrack-ng is below in screenshot 7. The password for Michael’s wifi is “tequieromucho”. The password for his network was weak enough that it only took Aircrack-ng 21 21 seconds to crack it.



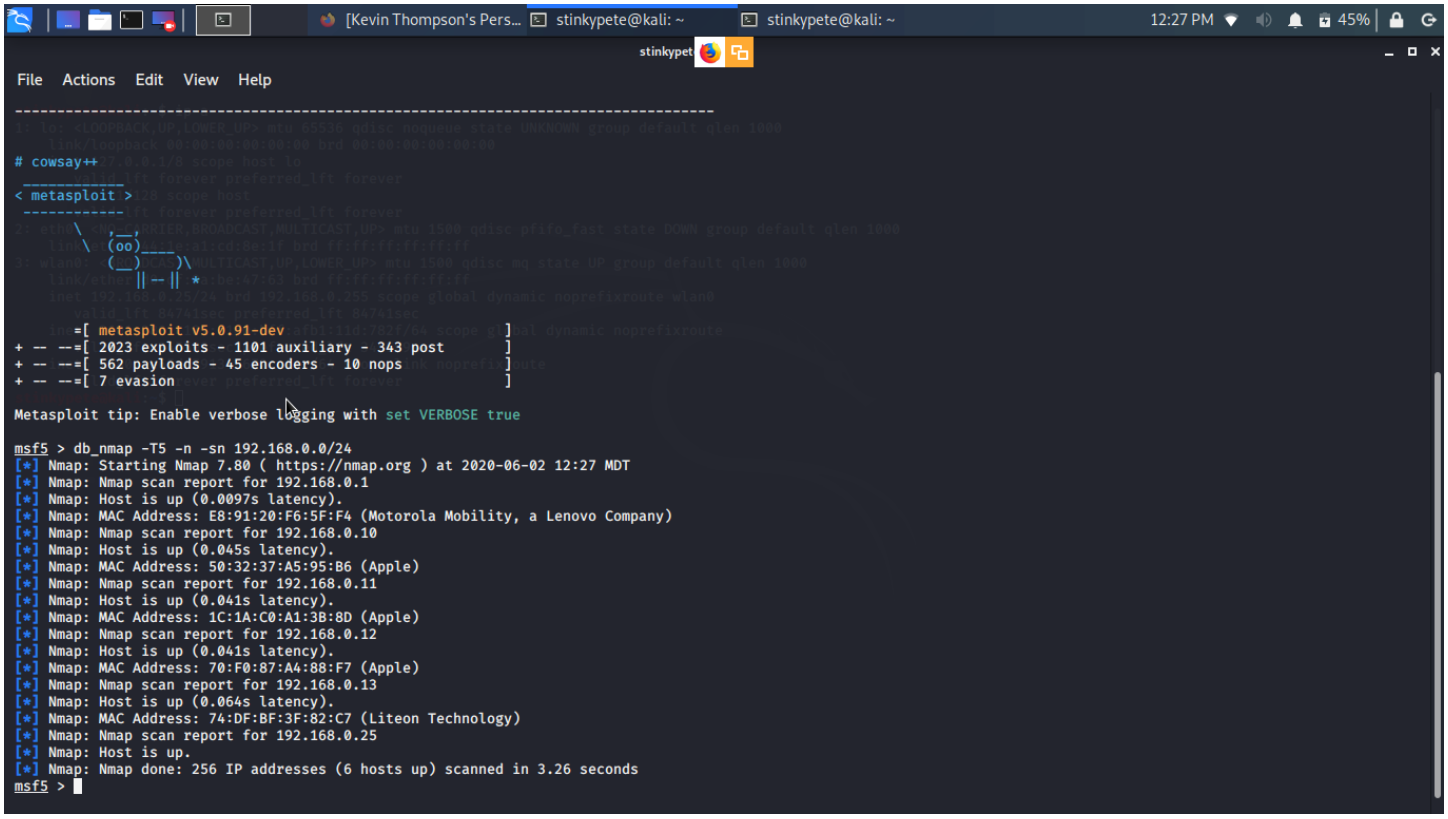
Screenshot 7

Step 3 - Exploit An Individual Machine

Now that Micahel's wifi password is known it is possible to log onto the network with his credentials. This is done using the GUI and logging on. Once on it is wise to determine the attacking IP address. This project was run from Kali-Linux where the command is **[ifconfig]**. The IP address being used by the attacking computer here is *192.168.0.25.*

With the IP identified it is possible to move on to using Metasploit. Metasploit is an incredible powerful suite of tools. Essentially it is a database of exploitation techniques that are designed by the hacking and cyber defense communities alike to hack and pen-test. It is possible to curate a unique attack but with thousands of options it is typically not necessary to do this. This project started Metasploit using the commands **[sudo msfdb]** which loads the database and **[sudo msfconsole]** which will transform the terminal into a Metasploit terminal. The reason for being inside a Metasploit terminal is because Metasploit takes some tools that can be run in the normal terminal and adds further functionality to offer more information. An excellent example of this is the NMAP tool.

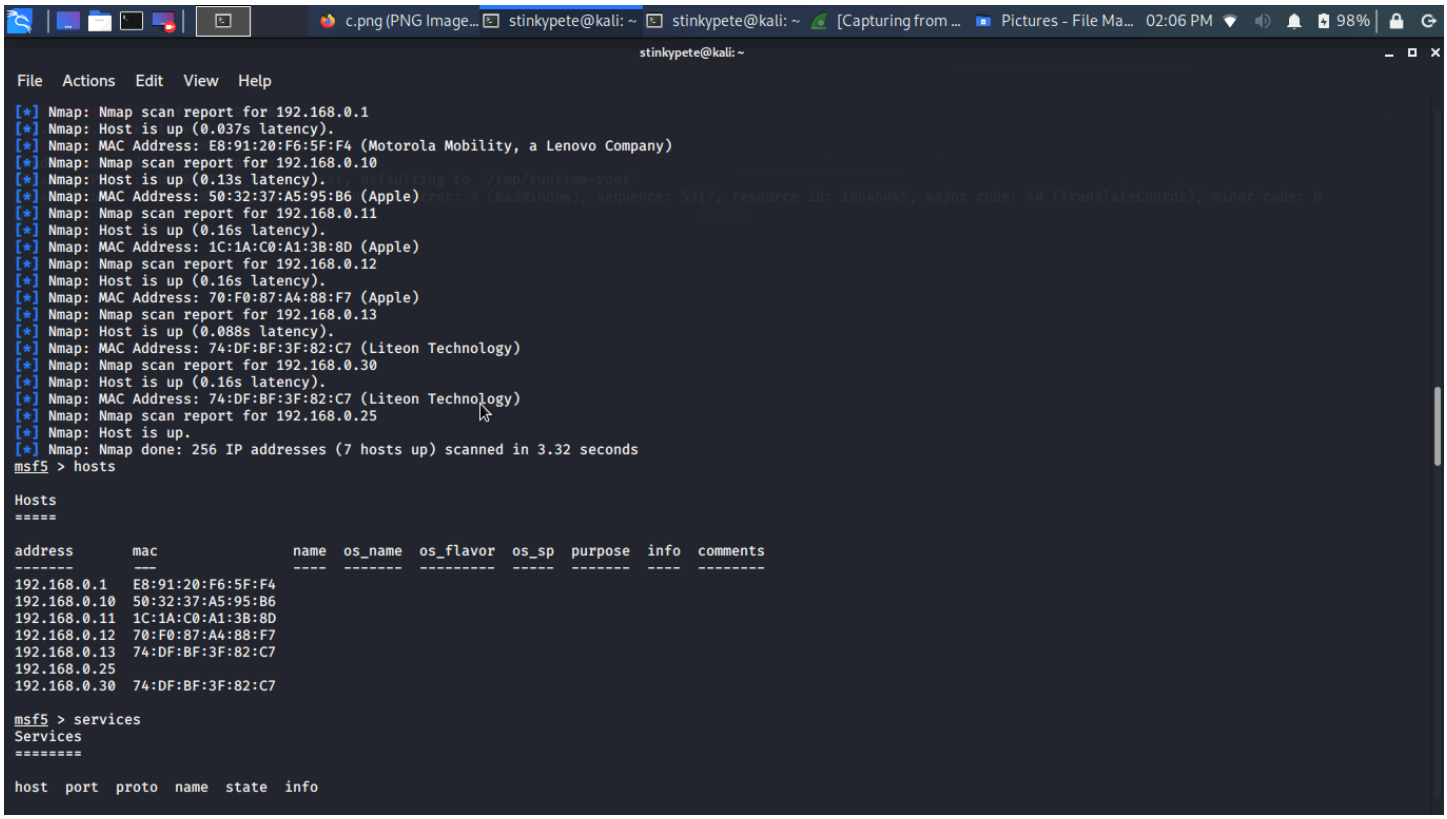
Within Metasploit the NMAP command can have options added to the line which will collect additional information for Metasploit to digest and offer back. Screenshot 8 below shows the command being initiated.



Screenshot 8

The command in screenshot 9 is **[db_nmap -T5 -n -sn 192.168.0.0/24].** The command has a few options which will be explained. The [-T5] is an option that in simple words is the "aggressive level". If the goal is to stay hidden; it is better to use the [-T1] although that scan could turn into hours. A [-T5] scan only should take sixty seconds at most granted the network does not have a lot connected to it. Another great way of explaining this is that [-T5] is akin to a guy walking down the street and banging on doors to see who is home. A [-T1] would be like someone camping out, tapping phone lines, and simply observing. The [-T1] enters the network as normal traffic from a random website and is harder to trace.

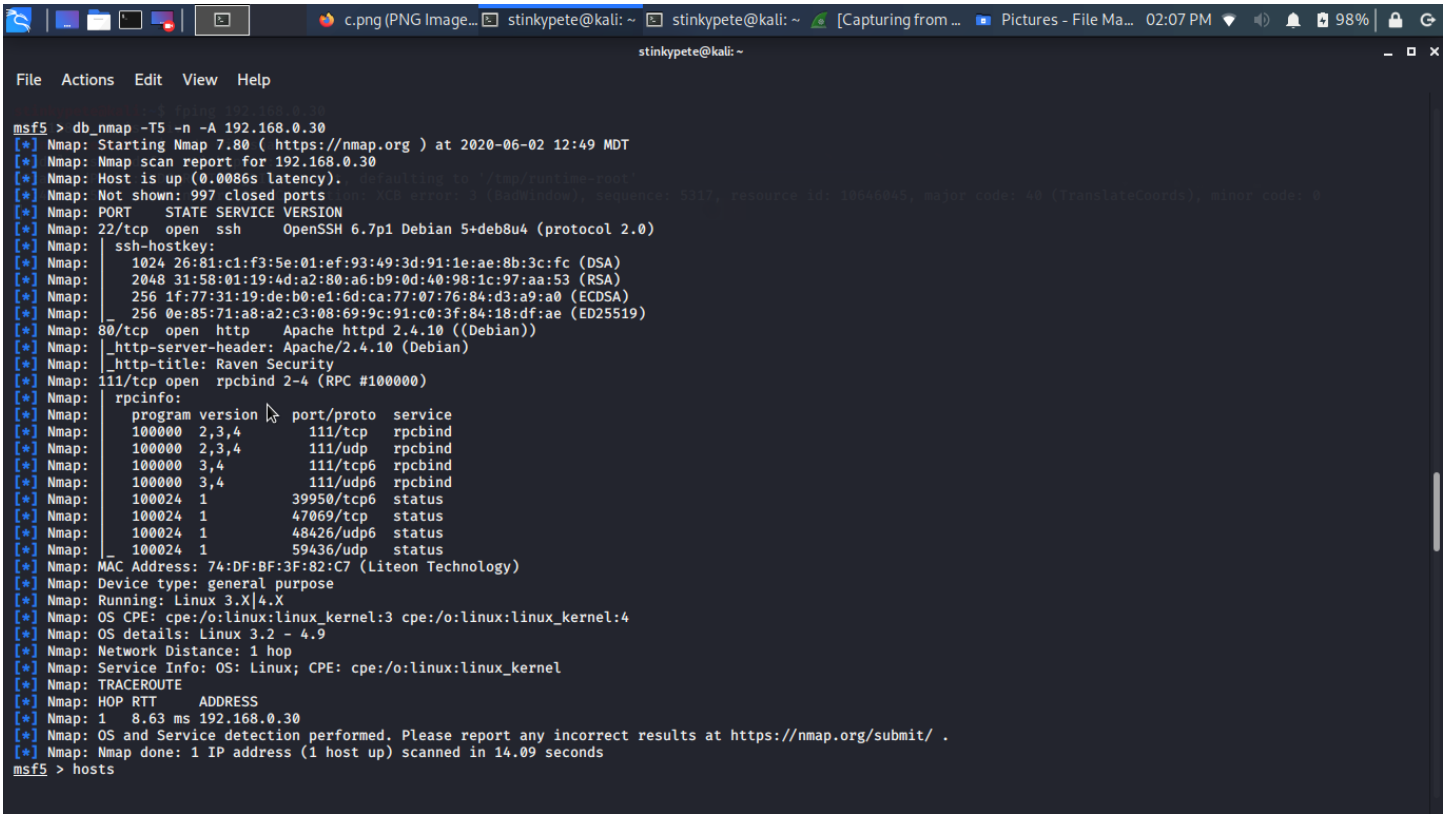
This is followed by the [-n] and [-sn] options. The [-n] option tells the scanner to return what it finds in the form of IP addresses without DNS resolution. The reason for this is because it makes the scan move faster. The [-sn] option is scanning via ping. Ping is an ICMP method that is simply sending signals looking for devices present on the network. It will return what it finds. The last option is naturally the IP address with the [0/24] cidr notation meaning scanning any and everything on this network. When the scan was complete it produced the intelligence in screenshot 10.



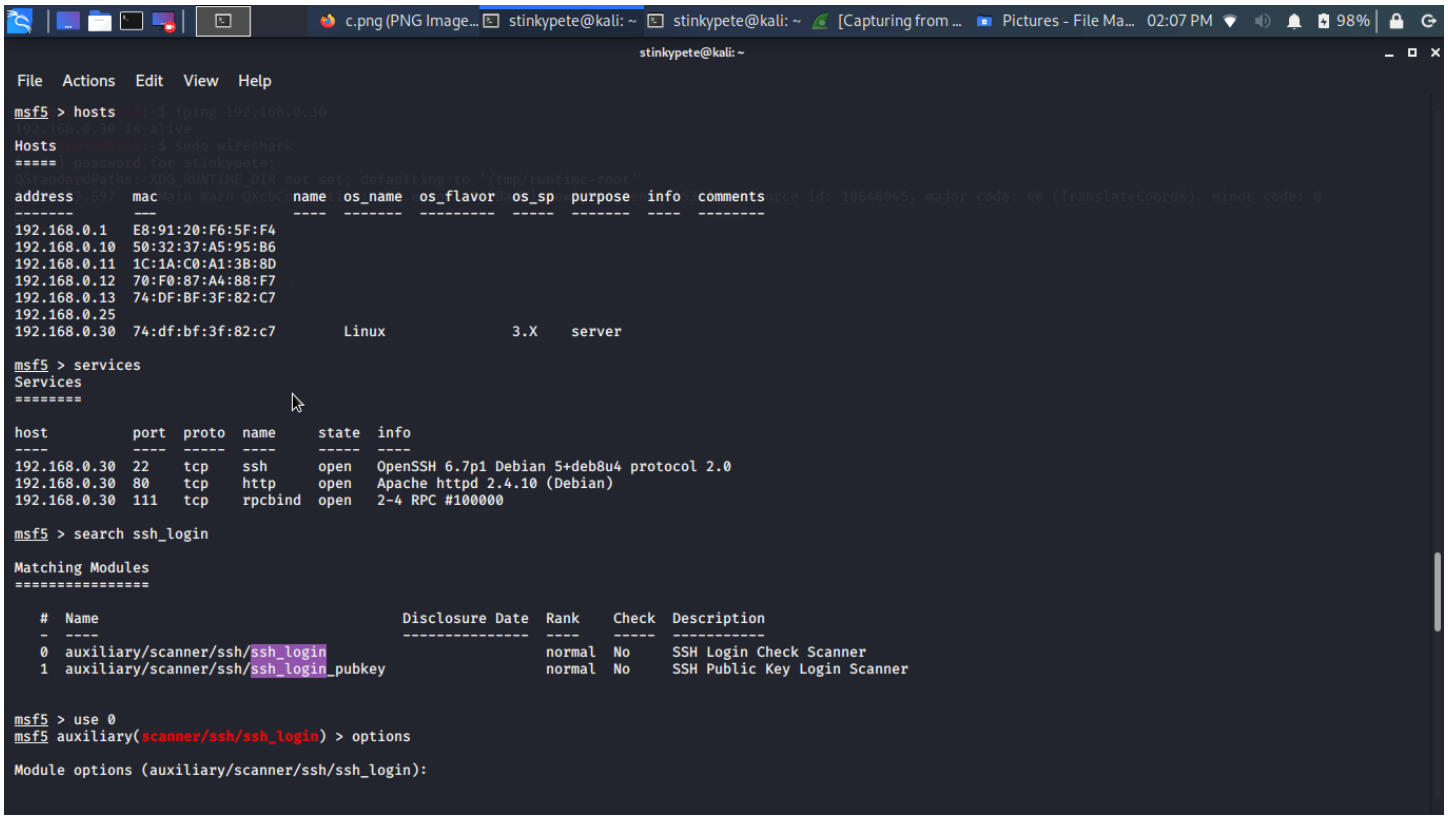
Screenshot 9

Screenshot 10 does not reveal a terrible amount of information because the programmed scan only asked “what devices are on this network”. That is what was asked for and that is what was given. It does fall into the category of “all intelligence is valuable intelligence” but the better option was to go back with a more specific scan. The [-T5] and [-n] options will remain the same but this time the option [-A] will be added. [-A] is an awesome creation because it is shorthand for “advanced” and runs at least four scan options under one title. It runs the [-O] tag to identify the operating system, [-sC] to scan for any scripts running, [-sV] to identify version information, and [traceroute] which sometimes use ping information to identify physical location. Again, this is all done by adding the option [-A]. The full command should read **[db_nmap -T5 -n -A 192.168.0.0/24]**.

Once the scan with [-A] is complete it is important to remember a few additional commands. That command above told Metasploit to process the information by processing and displaying are two different things. The world of computer hacking is very much the scene in Aladin where Robin Williams/Will Smith (pick your version) plays games with the specificity of the requests. You get ***exactly*** what you ask for. The way to manage this is after the scan is complete run the commands **[hosts]** which will reveal who/what is on the network and **[services]** which will reveal information about how they are running. Screenshots 11 and 12 display the output of the scan which has even more information than what is discussed in this report.

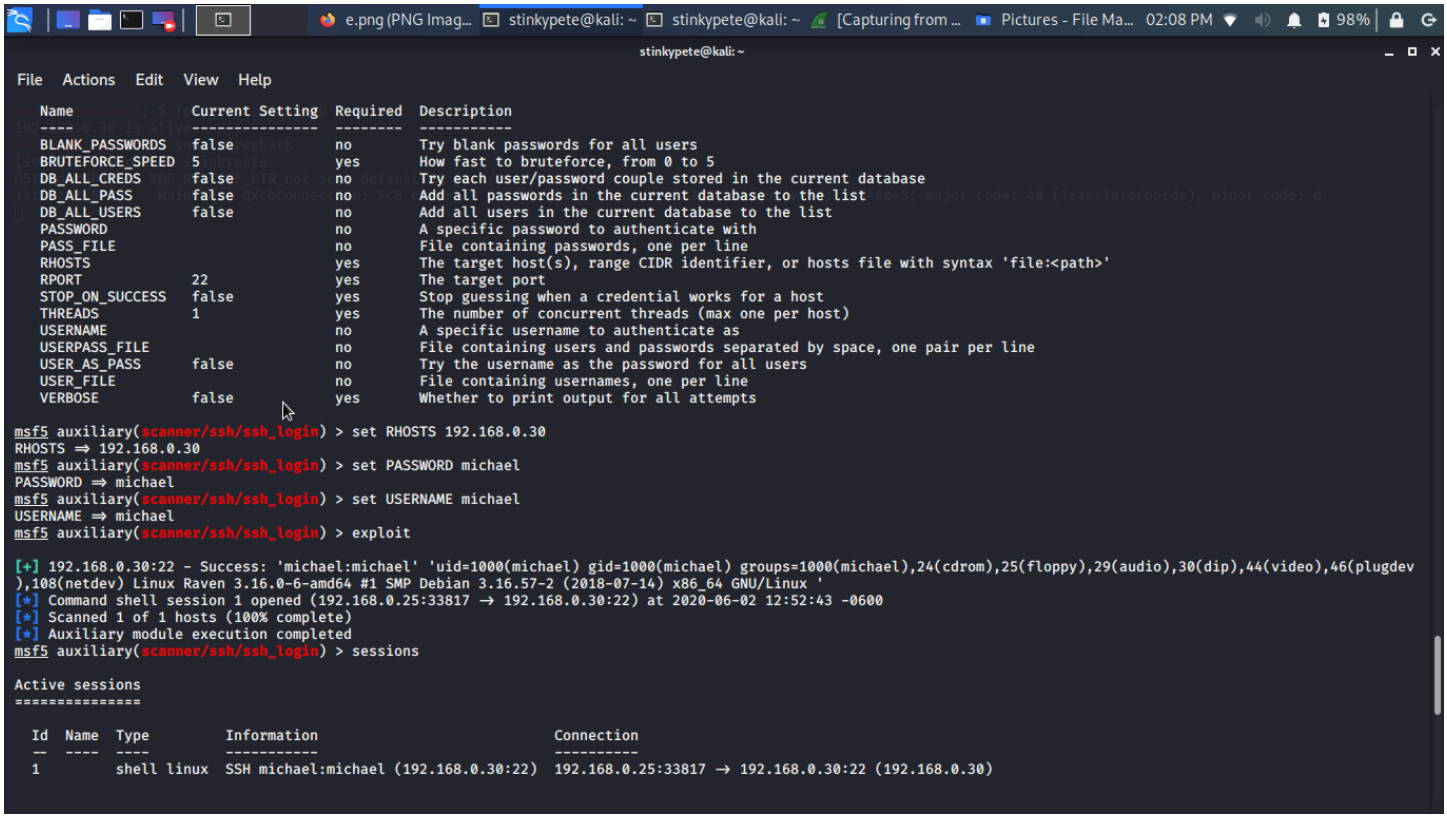


Screenshot 10



Screenshot 11

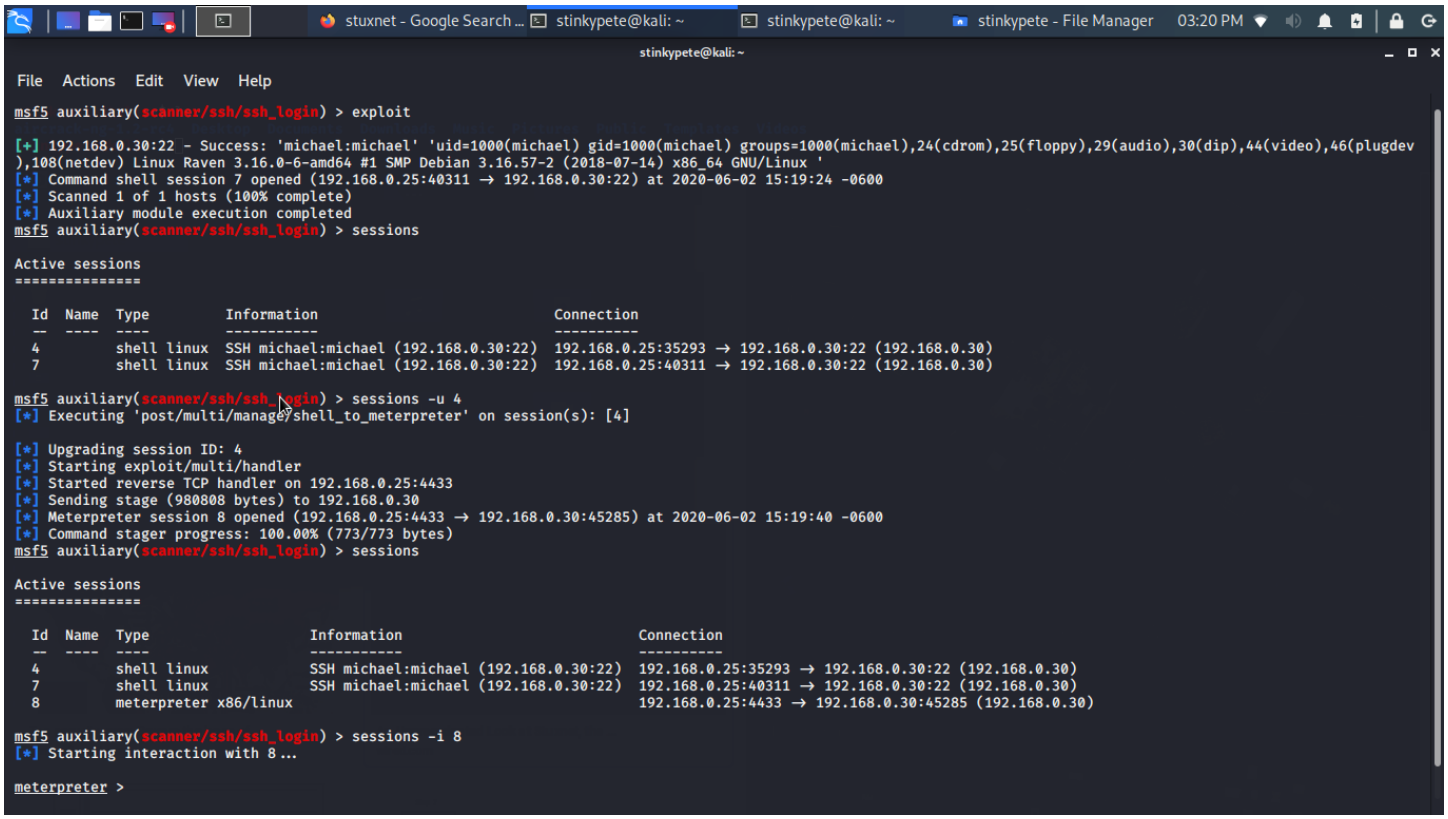
The advanced scan offered plenty of intelligence to move forward. It is obvious that there is far more activity on the IP address 192.168.0.30. There are most likely other devices on the network but considering this IP should come with the nickname “hack me”. If this is indeed Michael’s computer; he has a lot to learn about security. It has open http access which would most likely provide a gateway with possible avenues and it is possible remote controllable. This intelligence helps because it can be used in Metasploit. SSH protocol typically requires a login. These logins can often be exploited. Within Metasploit the [search] command will help locate already designed exploits for specific tasks. As stated earlier it is possible to design a custom exploit but that is out of scope for this report. It helps to know in Metasploit that the underscore [_] is an ally. The search criteria was simply **[search ssh_login]** and screenshot 12 above shows that two possibilities emerged and the first was chosen because it was a more simplistic choice than playing with ssh keys. SSH keys are parts of the cryptography on ssh and can be cracked but for this report the login was an easier choice. In order to use that particular exploit in Metasploit the command is the word “use” followed by the exploit, example: **[use auxiliary/scanner/ssh/ssh_login].** Screenshot 12 below shows this exploit along with the **[options]** command which will also be explained below.



Screenshot 12

Once the exploit is selected there are often options that must be set in order to run the exploit. This is done with the command **[options].** This produces a list of options that are required and not required. The biggest options required by this particular exploit are a “receiving host” or “RHOSTS” and some sort of way to convey login credentials. The exploit offers two different ways to do this. The attacker can create a “Pass_File” and a “User_File” into which can be listed predictable usernames and passwords. The exploit using that method will try all and then come back to tell you which, if any, were successful. The other option for this is to use the “password” and “user” option which will allow one username and one password to be set. This approach is often excellent if security is weak because social engineering allows for amazing guess work at passwords. Considering Michael’s password for the network was “tequieromucho” and was cracked in twenty-one seconds it is logical to think that he hasn’t established good passwords in this location. A few

guesses that are not shown in the screenshots and it was determined that the login for his linux server was username: michael and password:michael. The next few screenshots will show that connection being established and dive into a topic of the **[sessions]** command which will help to elevate privileges.

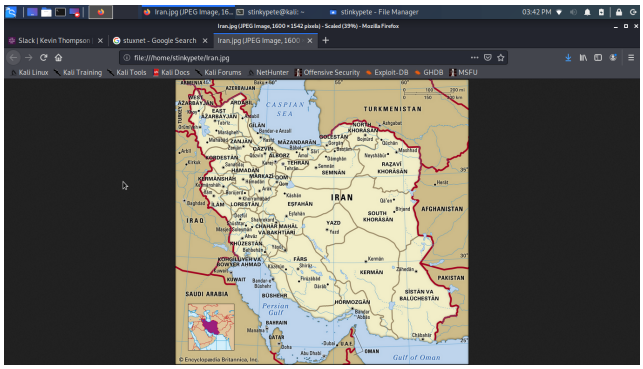


Screenshot 13

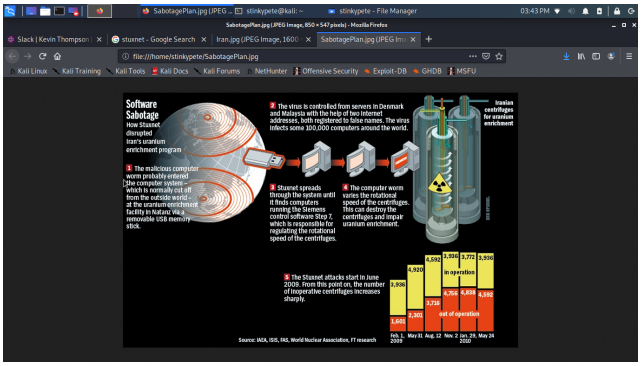
The **[sessions]** command in this exploit forces Metasploit to indicate what connections have been made available. Performing it the first time it is clear that a shell has been opened. The next step is the command **[sessions -u 4].** This command will take the session that is open on ID 4 and *upgrade* it to a meterpreter shell. Meterpreter in short explanation is a remote access payload that gives control over the machine. It is similar to the scene in 2Fast2Furious where the police shoot the fictional EMP harpoon that turns off Paul Walker's car. Think of that scene but rather than turning off the car a meterpreter shell would give remote access. The next command is **[sessions -i 8]** where it will switch the identity of the session into meterpreter. There is an entire set of commands with meterpreter shells. At the bottom of screenshot 13 it is clear that the attacker is on Michael's Linux server. The next stage is to explore and extract any information that could exonerate or condemn Michael.

Step 4 - Intelligence Extraction and Warning Upload

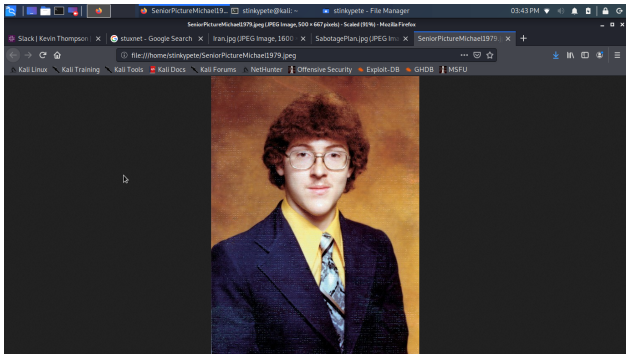
Once on Michael's Linux server it is abundantly clear that Michael is up to no good and needs to be reported to the authorities. There was no attempt at encryption or even hiding any evidence. The tool within Meterpreter is the [download] command. This will take anything on Michael's Linux server and use Meterpreter to download it to the attacking host, Kali-Linux. Screenshots 14-17 show the images pulled off of Michael's Linux server. Screenshot 18 shows the contents of a profile also on his machine that indicates he was initially blackmailed into working for the government and most likely his actions are revenge of some sort. There was also a file on the machine called "LaunchCodes.txt" but the file was empty. This indicates that Michael may have been planning something very bad. Screenshot 19 shows the action of deleting the contents of Michael's Linux server and replacing them with a note that tells him he has been caught. That was not a necessary step but if the guy was planning on hurting people or committing treason it made no sense to make it easy for him.



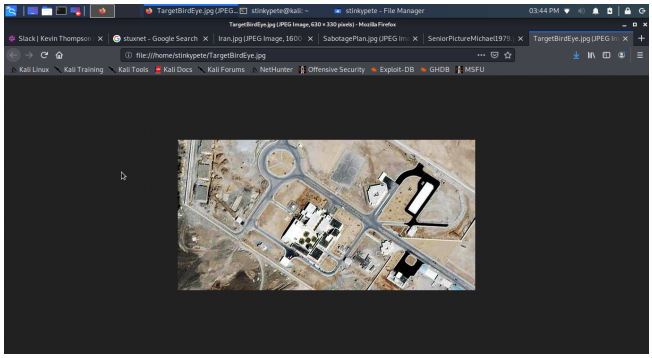
Screenshot 14
File Name: *Iran.jpg*



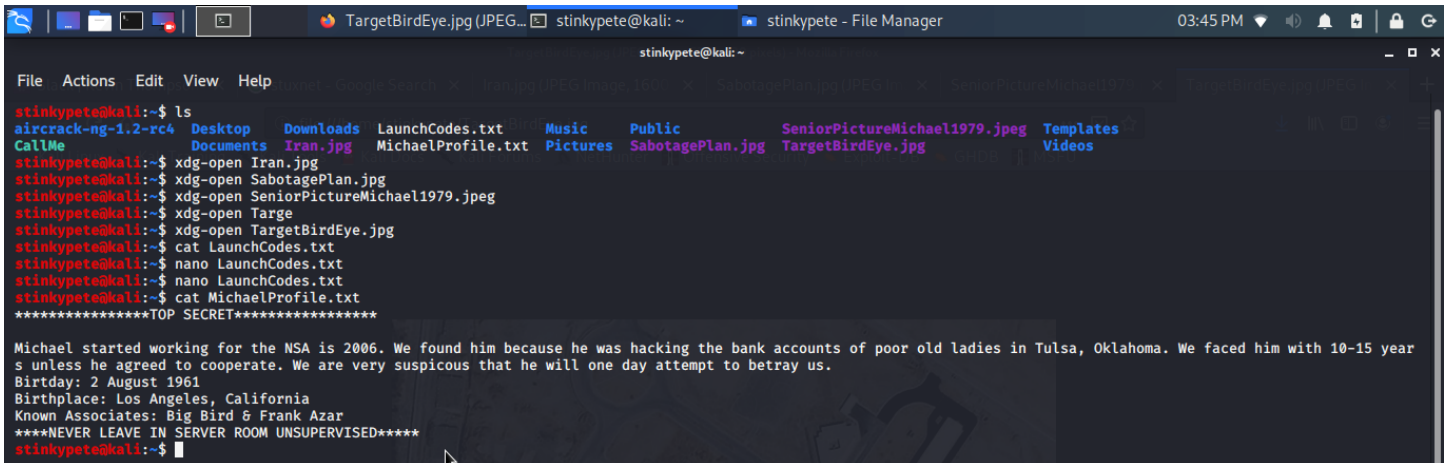
Screenshot 15
File Name: *SabotagePlan.jpg*



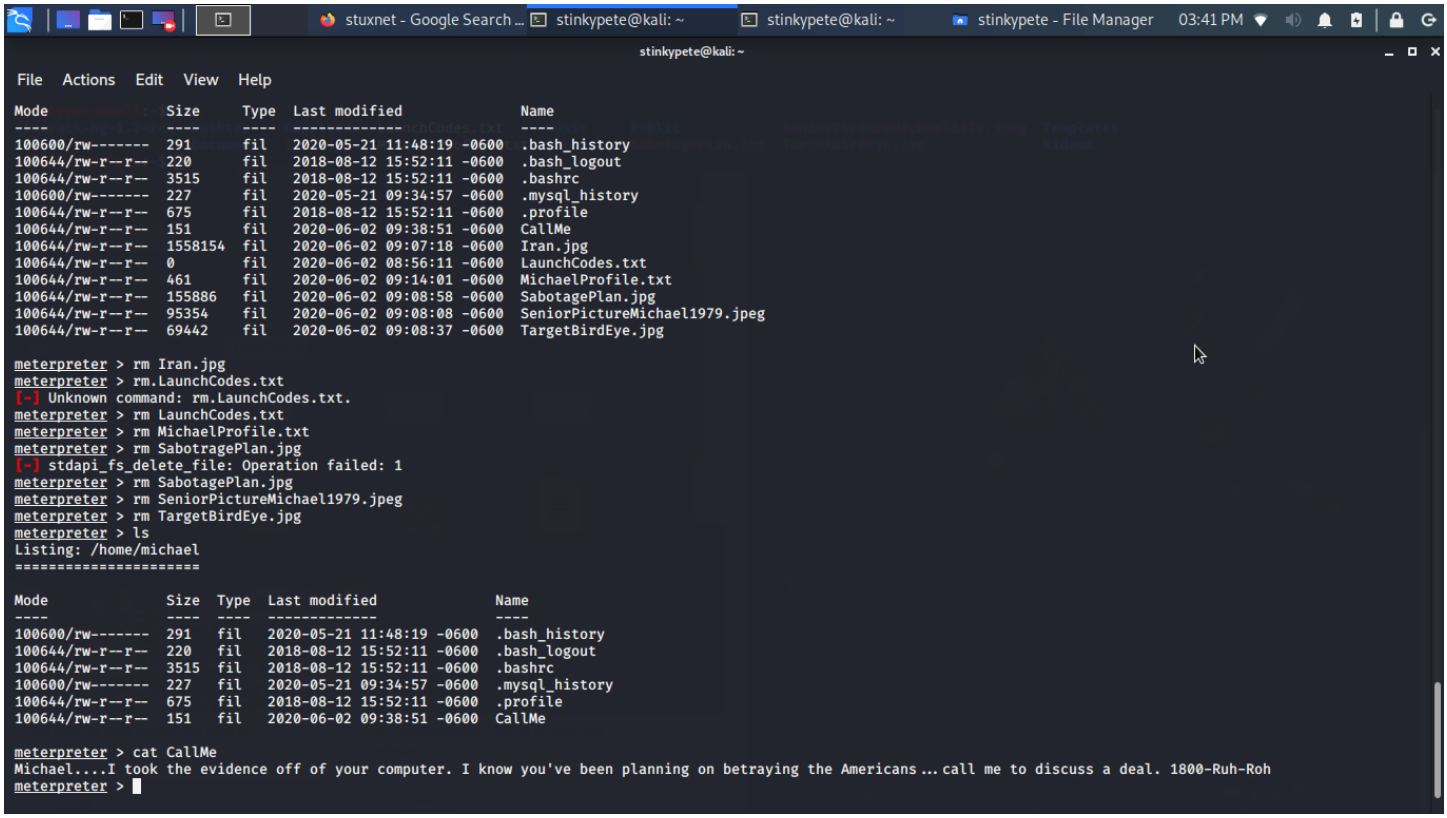
Screenshot 16
File Name: *SeniorPictureMichael1979.jpeg*



Screenshot 17
File Name: *TargetBirdEye.jpg*



Screenshot 18



Screenshot 19

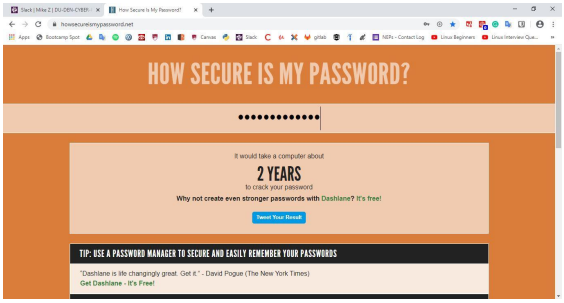
Step 5 - Helping Michael Improve His Security

While Michael is a criminal and help should not be given there is still an excellent discussion about what modifications he could have made that would have made the attack slightly more difficult. The true end story with any cybersecurity or IT setup is that if the information is sensitive don't put it on a machine. The word "unhackable" is fictional in reality and a hard better phrasing is "unhackable today at this moment". Everything is unhackable until it is hacked. The art of defense versus offense is a constantly evolving game with no end.

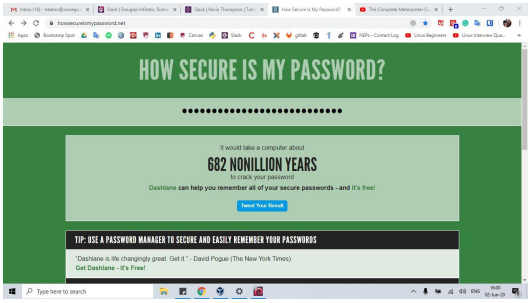
What could have been done to better secure Michael's network? A few recommendations are listed below with brief explanations as to why they could improve the security.

Building a Better Password

Michael's passwords were incredibly weak. An excellent tool to prevent this from happening is to evaluate the password being used. There are websites such as "howsecureismypassword" that can rate the password and offer how long it would take a standard computer to brute-force hack. Taking his password of "tequieromucho" would apparently take 2 years to hack. In reality it is a common password on rockyou.txt and took 21 seconds. If the logic of 2 years = 21 seconds is applied then it is logical to desire a complex password that is alphanumeric, containing symbols, and at least 20 characters. Socially the recommended number of characters is 8 but to stay ahead of the game the number should be 20. Screenshots 20 and 21 show Michael's original password in 20 as compared to the password for the attacking Kali box in 21. The password for the Kali box was generated using a tool called BitWarden. Bitwarden is an excellent password generator and vault storage service provider. As seen in screenshot 21 it can help create and store passwords that take 682 nonillion (10^30) years to be cracked. Using the division principle above it means that they are far less likely to be cracked. It also makes it reasonable to have a different password for every account. Had Michael been using these types of passwords it is unlikely that this type of operation would have been feasible.



Screenshot 20



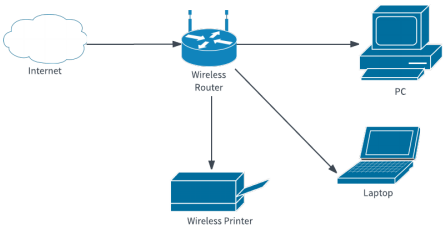
Screenshot 21

Disabling SSH

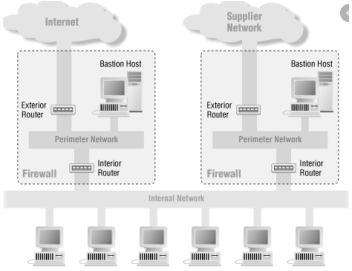
SSH or secure-shell is a fantastic tool. It can help system admins login to remote computers and even help users navigate complicated tasks that they are not trained to do. While it is an excellent tool and protocol it does not mean it should be left open and accessible all of the time. Riding lawn mowers are excellent tools but it isn't wise to park them in the driveway with the key ready to be stolen. SSH is something that system administrators can allow when needed and disable when not needed. This prevents unwelcome third parties from guessing the password in similar events to what happened to Michael. There is an excellent discussion that dives into this link here on stackexchange.

Improving Network Architecture

Another critical part of security that would have made this operation more difficult would have been better network architecture. Screenshots 22 and 23 below show two different architectures.



Screenshot 22



Screenshot 23

Screenshot 22 is an example of the setup that Michael was running alongside the majority of homes across the world. It is a simple box that connects machines to the internet. In an “advanced” setup at this scale there may be a firewall but that was not even present in Michael’s case. This is contrasted to screenshot 23. In screenshot 23 you can see that the connection to the internet is in the box on the left within its own firewall. Traffic must pass through that firewall to get onto the internal network and there can even be subnetworks with other firewalls to pass through or air gaps to be jumped. If Michael was planning on betraying his country he should have at least had some sort of better architecture.

Encrypt Evidence

Another surprising part about Michael’s computer is the lack of encryption. Encryption is the process by which letters, numbers, characters, and shapes are ciphered to produce an equivalent matching version that is only readable to the intended recipient. They are based on mathematical formulas and algorithms such as MD5. There are several different methodologies and the trick to them is that if the sender and recipient both have the key they are easily opened. If they are intercepted by an unwelcome party, as with Michael, they are unreadable. Michael stored information like the above in plain text which is readable to any one.

Social Engineering

The final recommendation that would have made this more difficult would have been the elements of social engineering. That is to say; people are predictable. Micahel could have had motion detectors or cameras in or around his house and when a suspicious Volkswagen Tiguan is parked outside his house for hours at a time it would raise suspicion. His password and login was his name and his password to the wifi was discovered in 21 seconds. The scenario really shouts that Michael wanted to get caught.