Wi-Fi Redux: Never Trust Untrusted Networks

Young B. Choi
Department of Science, Technology, and Mathematics
Regent University
Virginia Beach, VA 23464-9800
USA

Kenneth P. LaCroix
Department of Science, Technology, and Mathematics
Regent University
Virginia Beach, VA 23464-9800
USA

***Abstract*—This study analyzes the dangers posed to computer user information and their equipment as they connect to untrusted networks, such as those found in coffee shops. Included in this study is a virtualized lab consisting of the target and attacker nodes and router to facilitate communication. Also included are a binary for reverse connection and a modified binary that was created to connect back to the attacker node and bypasses most Anti-virus software.**

***Keywords*—Wi-Fi; Untrusted Network; Pineapple; MITM; DNS Spoofing; Least Privilege**

I. Wireless Fidelity (Wi-Fi)

A. Introduction

Wi-Fi is convenient and in general terms, often fast. Its use is also ubiquitous in today's connected world. At times a user may be unaware that a device is connected to a Wi-Fi access point instead of cellular technology such as LTE; the transition is often seamless. Convenience functions include Wi-Fi Protected Setup (WPS), device auto association and the choice to use encryption technology. This research explores how the convenience of Wi-Fi and its convenience related functions may be used in attacks which could lead to data theft, invasion of privacy or the compromising of devices.

B. Wi-Fi Use

The world is very data hungry, with the amount of data consumed and generated growing at an exceedingly increased rate. In fact, some estimates put the total monthly traffic for mobile users around 3.6 EB [1]. And, to stay connected on the go, many users flock to coffee shops or similar businesses that offer free Wi-Fi. Companies often leave the Access Point (AP) unencrypted or display the network password in clear sight or available after purchasing good or services.

Although there is a growing trend in the adoption of the WPA2 encryption method, one website, whose data is collected from volunteers who War Drive, reports roughly 6% of scanned networks still do not use any encryption method at all [2]. Furthermore, when the user utilizes these "free" untrusted networks, they may not think about the potential for misuse and how a malicious actor may be able to obtain sensitive information or possibly break into their system via malicious software (malware).

C. Wi-Fi Pineapple

One attack vector that poses a threat to consumers on untrusted public networks such as coffee shops is the well-known Man-in-The-Middle (MITM) attack. An MITM is accomplished by an attacker placing a node in between the router and the victim so that all or selective network traffic passes through the attacking node, which then puts the attacking node in complete control of said traffic. There are several ways to implement an MITM on wired and wireless networks.

For wireless networks, the "Wi-Fi Pineapple" (hereafter Pineapple) manufactured by Hak5 LLC, is a consumer device that easily facilitates and automates much of the process of deploying a Wi-Fi honeypot, thus placing the clients connected to the instrument in an MITM situation. The Pineapple works in part by exploiting 802.11 broadcasted beacon frames that are sent out at specific intervals from clients seeking to connect to remembered networks. Included, in section 4 of a beacon frame is the Service Set Identifier [3] or network name, to which the Pineapple responds affirmatively either selectively or all the time, depending on the Pineapple's configuration.

The beacon frame response from the Pineapple to the client facilitates the opportunity for association (to connect) to the spoofed AP. The association can be seamless for unencrypted remembered networks and often requiring user intervention for encrypted networks, depending on client side factors such as the operating system or user settings. Deauthentication frames, which "invalidate the authentication relationship" [4] can be sent to clients from the Pineapple in the hopes that the clients start associating with the rogue AP, not the legitimate AP (see Figure1).

In the context of most coffee shop networks, deauthentication attacks do not require user intervention due to there being no authentication, nor in most cases are the users alerted that their device has associated with a rogue AP.

II. Lab Setup

A lab intended to mimic a real-world situation might consist of three virtual machines: a victim node, attacker nodes, and a router. The router is the modern and open source pfSense firewall (see: https://pfsense.org) with the default configuration of 192.168.1/24. The victim node runs Microsoft Windows 7 with a wireless card. And, the attacker node is running the popular open source penetration testing distribution, Kali Linux (see: https://kali.org) with the Pineapple directly connected via USB, providing the Wi-Fi and MITM, on the 172.16.42/24 network. See Figure 2.

III. Attack Vectors

A. Introduction

Once an attacker decides to attack a device, system or network, he/she will need to determine, from the investigation, what type attack to execute. Not all attacks have to be on systems and networks. In fact, many attackers may find that people hacking or social engineering is a very effective avenue to gain entrance into systems and networks. The Pineapple and other attack types such as DNS Spoofing, in part, relies on uneducated users who may not notice what is occurring behind the scenes.

B. The Pineapple and Remembered Networks

As mentioned earlier about device association, the Pineapple's power relies on the fact that target devices have at least one recognized Wi-Fi network. When seeking to associate, the client will send out probes looking for any remembered networks [5]. The Pineapple uses this to exploit a client's default behavior of trying to automatically associate when an AP sends a crafted beacon response that matches the initial probe from the client.

The Pineapple includes a module as part of PineAP called Dogma that allows the attacker to specify the rate at which the Pineapple will send out beacon responses, 200-400 times normal for a standard AP [6]. Dogma's aim is to reinforce the legitimacy of the AP and aid in device association. The odds for most phones and computers to have at least one remembered and unsecured network are high. In fact, most users are likely not to know that such a list of remembered networks exist and are likely to include the Wi-Fi networks of Starbucks, Walmart, Best Buy, Airports, etc.

C. DNS Spoofing

Once device association has occurred with the Pineapple (or any rogue AP), the traffic flow pattern is from the victim to the rogue AP to the attacking node and finally to the Internet. Therefore, a wide variety of attacks is possible such as traffic analysis, information gathering, X.509 stripping, DNS spoofing, credential harvesting, etc. For this research, DNS spoofing of the regent.edu domain was utilized to redirect the victim to a web page instructing the user to download a flash player update file, which is malicious. See Figure 3. **Note:** DNS spoofing of the target domain does not affect any user not connected to the rogue AP. As of the time of writing, the legitimate resolution of the regent.edu domain is 70.175.9.91.

D. Malicious Binary Generation and Anti-virus Detection

Veil-Evasion is an open source project that can be used to generate an executable payload that may bypass some, if not all, Anti-virus software via selectable payload encryption [7]. Veil-Evasion is installable on many Linux distributions including Kali Linux which the attacking node in this research uses. As of writing the software includes 51 different payloads. An attacker can also add their custom shellcode as well. This study used the Ruby Base64 payload to generate a binary, that when executed opens a connection back to the attacking node for further exploitation. The Ruby Base64 offered the lowest detection rate [8] of the large Anti-virus vendors flagging the binary, as of writing [9].

E. Binary Delivery

Utilizing DNS Spoofing, as outlined earlier, the regent.edu domain is the target that resolves to the Pineapple, which is running a minimal web server. The web server delivers to the client a page that states the client's flash player plugin needs to be updated. See Figure 4. A convincing web page can be crafted by hand or used from other projects such as the open source Wi-Fi Phisher, which is another way of creating a rogue AP (see: https://github.com/wifiphisher/wifiphisher).

IV. Post Exploitation

A. Introduction

The post exploitation stage in an attack is where the attempt to exploit the user, technology or systems was successful. The attacker now has a foothold in the system/network and may leverage further exploits, networking mapping, monitoring, etc. One tool that is open source and readily available in the Kali Linux penetration testing distribution is Metasploit which includes meterpreter which can be used for post-exploitation.

B. Opening a Reverse Shell and Privilege Escalation

Once the binary executes, meterpreter, opens a shell, connecting to the attacking node on the port specified during binary generation. See Figure 5. The process of opening the shell starts with the stager executable, which is the binary generated earlier. The stager prompts for UAC access (if enabled) and when granted downloads a DLL from the attacking node, which contains the reverse shell payload that loads into memory [10]. Once the shell opens, the attacker has the same permission as for when the stager executed, in this case, Administrator privileges. The next step the attacker may take is to obtain system privileges which will allow just about any modification desired. See Figure 6. Running netstat on the victim node will also show the connection from the ephemeral port of 49641 to port 8080. See Figure 7.

C. Maintaining Access

Once the shell is opened; the attacker will likely want to maintain access to the system. Metasploit has a built-in function to upload a Visual Basic Script (VBS) to the victim that will open a shell every time the user logs into the computer. However, many Anti-Virus flags the VBS as malicious. Another route may be to hide the stager in a legitimate executable. If an attacker chooses this path, the first step might be to determine what programs are installed on the victim node and find a program that is likely to run on a regular basis. See Figure 8. A candidate is PuTTY, the popular SSH and Telnet client.

Veil-Evasion, discussed earlier when generating the stager executable, has a payload called Backdoor Factory that was created by Joshua Pitts. Backdoor Factory (BDF) exploits the nature of the Portable Executable (PE) format. The PE format is a data encapsulation method that "provide[s] the best way for the Windows Operating System to execute code and also to store the essential data which is needed to run a program" [11].

BDF relies in part on the concept of code caves, which is the process redirecting program execution (like a function) to separate code and then returning for normal execution [12]. So, in other words, the newly generated binary from Veil - Evasion includes the stager and original PuTTY program, the stager code will run inside of PuTTY and PuTTY will run as normal. See Figure 9. Anti-Virus detection rates are high for this method but may be allowed anyway as PuTTY is a legitimate program.

D. Implications

Once an attacker has access to a node, meterpreter offers many commands such as those used for data exfiltration and infiltration, keylogging (See Figure 10), screenshots, remote desktop access, further system exploitation, webcam recording, timestamp and log manipulation and so on. If the user allowed the stager through UAC, the shell could likely get system authority which has more access to the system than even an Administrator account and is often used for system services; otherwise the attacker may try to exploit the system further depending on the attacker's overall goal.

V. Prevention

A. Introduction

In many cases, there may be little recourse for attacks occurring in a public location. Many coffee shops, for example, only provide Wi-Fi as a service enhancement for patrons. Users may use the service at their own risk own risk and the network may be unmanaged by an administrator. However, there are some remediation tactics users can employ, including education and technical controls.

B. Least Privilege

It is not uncommon for the average user to be logged in as a local administrator, which can be a bad idea for a variety of reasons. Programs that are executed as the Administrator retain those permissions. One option is to create a second unprivileged account for everyday use. The user would still need to enter Administrator credentials when installing a program, but such a delay might give the user a chance to stop the attack from progressing as the UAC is often disabled or not fully understood.

C. Remove any Saved Networks

It is standard practice in most operating systems to maintain a list of remembered networks the device associates. It may be good hygiene to clear this list, turn off this function or leave only one SSID on the list that the user trusts, which must be an encrypted network. Unencrypted networks, as demonstrated, are easy to spoof.

D. No Unencrypted Wi-Fi

As stated earlier, there is a cost-benefit tradeoff many users are seeking when they connect to open (and free) networks. Users have an allotted amount of data on cellular plans or pay per the megabyte or gigabyte. By using free Wi-Fi, the user offsets the cost of the mobile data plan which may include tethering for laptops or other devices.

However, the potential dangers of using unsecured and untrusted networks can far outweigh the perceived benefit which is why it might be better to stay off such networks or at a minimum, build a Virtual Private Network (VPN) tunnel to ensure the confidentiality and data integrity of the session. **Note:** encrypted Wi-Fi is still susceptible to attacks such as DNS Spoofing if the proper technical controls are absent such as client isolation.

E. Turn off Wi-Fi

By turning off Wi-Fi on the device when unneeded, the user negates the effect of the auto association behavior built into modern operating systems. When Wi-Fi is in the off state, probes for remembered networks will not send and may be helpful when the device is in a public space, but the user does not need or want Wi-Fi access.

F. Check What the Device is Doing

One indication of a rogue AP to a user may be that their device associates with an AP but the physical location of the business that offers the access is far away or not in the same place as the device. For example, a device associated with the SSID of "SouthwestWi-Fi" but the device may not be in an airport or on an airplane.

However, most operating systems have their Wi-Fi settings pane in such a way that it requires the users to manually navigate to the panel to see what network the device is associated. Fortunately, programs like NetStumbler exist for Windows to quickly scan the Wi-Fi space and will display the details of every network in range and band (2.4, 5 GHz or both depending in the Network Interface Card).

VI. Conclusion

In conclusion, it was discussed the growing trend for mobile and computer users to offload some of their data use to Wi-Fi networks due partly to the fact the mobile data are often allotted a finite amount of bandwidth or charged per use. Although the trend for encrypted Wi-Fi networks is on the rise, there are still many networks to remain unencrypted. These unencrypted networks can be found in coffee shops, malls, and businesses, etc. and often offered free of charge.

However, there are several attack vectors that users may be subject to and unaware of such as rogue access points, DNS spoofing, and malware delivery. Mitigation techniques may include the installation of software on the device to do a minimal site survey for legitimate access points, turning off Wi-Fi when unneeded, clearing out the list of saved networks, creating an unprivileged account on computers, and veering away from unencrypted networks, if possible.

References

[1] Lee, K., Lee, J., Yi, Y., Rhee, I., & Chong, S. (2013). Mobile data offloading: How much can WiFi deliver?. IEEE/ACM Transactions on Networking (TON), 21(2), 536-550.

[2] WiGLE Statistics. (n.d.). Retrieved February 21, 2017, from https://www.wigle.net/stats

[3] IEEE standard association. (2012, March 29). Retrieved February 21, 2017, from http://standards.ieee.org/getieee802/download/802.11-2012.pdf

[4] IEEE standard association. (2012, March 29). Retrieved February 21, 2017, from http://standards.ieee.org/getieee802/download/802.11-2012.pdf

[5] Dormann, W. (2015, August 11). Instant KARMA Might Still Get You. Retrieved February 21, 2017, from https://insights.sei.cmu.edu/cert/2015/08/instant-karma-might-still-get-you.html

[6] The Next-Gen Rogue Access Point: PineAP. (2017, January 12). Retrieved February 21, 2017, from https://www.hak5.org/episodes/pineapple-university/the-next-gen-rogue-access-point-pincap

[7] Truncer, C. (2016a, February 16). February 2016 V-Day. Retrieved February 21, 2017, from https://www.veil-framework.com/february-2016-v-day/

[8] Truncer, C. (2016b, June 14). Retrieved February 21, 2017, from https://www.youtube.com/watch?v=xNIohkmja2M&index=9&list=PLNhlcxQZJSm9NT-zQ9jHdYRhtGOASAYT7#t=48m47s

[9] Update3.exe | 7/35 | NoDistribute. (2017, February 15). Retrieved February 21, 2017, from https://nodistribute.com/result/xgsFYuHoB6mWnAcEz8ORCkGPMt

[10] Wadner, K. (2014, October 7). An Analysis of Meterpreter during Post-Exploitation. Retrieved February 21, 2017, from https://www.sans.org/reading-room/whitepapers/forensics/analysis-meterpreter-post-exploitation-35537

[11] Danehkar, A. (2005, December 27). Inject your code to a Portable Executable file. Retrieved February 21, 2017, from https://www.codeproject.com/articles/12532/inject-your-code-to-a-portable-executable-file

[12] Pitts, J. (2013, March 16). Backdooring Win32 Portable Executables. Retrieved February 21, 2017, from https://www.youtube.com/watch?v=SXaoVo_U7kA#t=2m0s

```
IEEE 802.11 Deauthentication, Flags: ........
  Type/Subtype: Deauthentication (0x000c)
▸ Frame Control Field: 0xc000
  .000 0001 0011 1010 = Duration: 314 microseconds
  Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
  Transmitter address: 82:2a:a8:          (82:2a:a8:          )
  Source address: 82:2a:a8:          (82:2a:a8:          )
  BSS Id: 82:2a:a8:          (82:2a:a8:          )
  .... .... .... 0000 = Fragment number: 0
  0000 0001 0111 .... = Sequence number: 23
```

Fig. 1. Deauthentication Frames sent to Broadcast

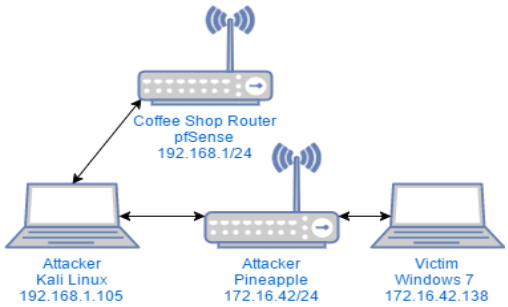

Fig. 2. Lab Network

```
Name:    regent.edu
Address:  172.16.42.1
```

Fig. 3. DNS Spoofing of a domain

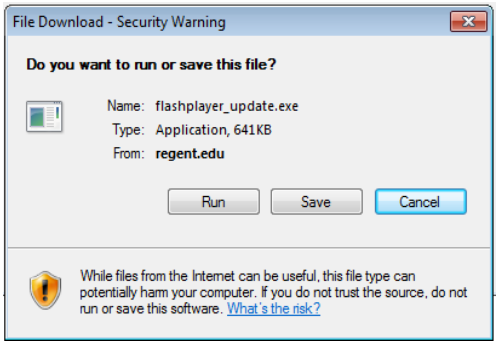

Fig. 4. Downloading the Binary in Internet Explorer

```
[*] Sending stage (957487 bytes) to 172.16.42.238
[*] Meterpreter session 2 opened (192.168.1.105:8080 -> 172.16.42.238:49641) at 2017-02-15 13:48:46 -0500
```

Fig. 5. The Stager binary sending the DLL and opening a session

```
meterpreter > getuid
Server username: user-win7\user
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Fig. 6. Privilege escalation

```
C:\Users\user>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    172.16.42.238:49641    192.168.1.105:8080     ESTABLISHED
```

Fig. 7. Showing the connection to the attacker on the victim node

```
[*] Enumerating applications installed on USER-WIN7

Installed Applications
======================

 Name                                          Version
 ----                                          -------
 NETGEAR A6210 Genie                           1.0.0.35
 NETGEAR A6210 Genie                           1.0.0.35
 NETGEAR A6210 Genie                           1.0.0.35
 NETGEAR A6210 Genie                           1.0.0.35
 PuTTY                                         0.67.0.0
```

Fig. 8. Enumerating installed applications in meterpreter

```
##################################################
[*] Cave 1 length as int: 705
[*] Available caves:
1. Section Name: None; Section Begin: None End: None; Cave begin: 0x29c End: 0xffc; Cave Size: 3424
2. Section Name: .rdata; Section Begin: 0x5d000 End: 0x7b000; Cave begin: 0x7a47c End: 0x7b000; Cave Size: 2948
3. Section Name: None; Section Begin: None End: None; Cave begin: 0x7c400 End: 0x7d00a; Cave Size: 3082
**************************************************
[!] Enter your selection: 3
[!] Using selection: 3
[*] Patching initial entry instructions
[*] Creating win32 resume execution stub
[*] Looking for and setting selected shellcode
File putty.exe is in the 'backdoored' directory
```

Fig. 9. Binary modification with BDF to include the meterpreter stager

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
google.com <Return> miketyson <Return> thisismypassword! <Return>
```

Fig. 10. Keylogging with meterpreter keyscan