WiFi Security Assessments

Robert Dooling - Dooling Information Security Defenders (DISD)
December, 2009

Table of Contents

Introduction

This paper is intended for anyone with an interest in the security of home to small-to-medium business (SMB) level wireless access points and WiFi networks – particularly, the types of authentication and encryption provided by these devices.

Readers will find it helpful to be familiar with basic TCP/IP and WiFi (802.11) concepts, but expertise is not required.

The purpose of this paper is to demonstrate the ease with which some default or poorly-configured wireless access point security configurations can be circumvented using readily-available free software. This paper does not describe any new, previously-undisclosed, or innovative techniques in wireless security and analysis. This should emphasize the point that these type of attacks can be performed by a modestly skilled attacker.

Unless otherwise specified, all of the tools mentioned in this paper are free, open source software (FOSS). This applies also to the DD-WRT firmware used for the target access point. Additionally, the attacking system and sample target access point described in this paper are each more than five years old. Again, the minimal costs associated with these tools and hardware emphasizes that an attacker need not be well-funded to compromise a network.

Note that some of the tools and techniques discussed in this paper may be illegal in your jurisdiction if used against others' systems. All demonstrations in this paper were run entirely with the author's personal systems.

Potentially-sensitive information (IP addresses, usernames) has been redacted from some illustrations contained herein, but all images otherwise represent the actual occurrence of events.

Default configuration overview

Default system settings for a DD-WRT firmware-installed wireless access point (AP) are shown below, as seen through the web interface. Note in particular, the LAN IP address and SSID values.

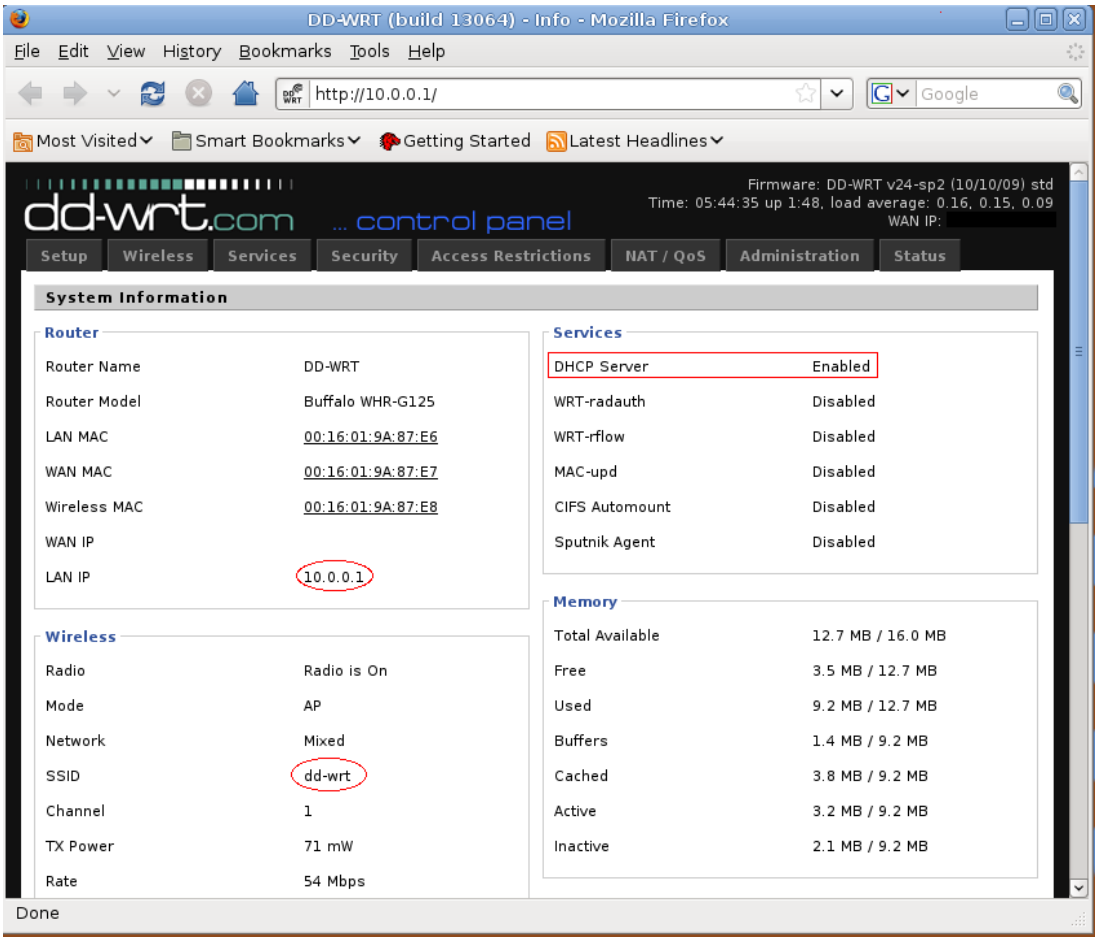

Illustration 1: DD-WRT firmware default system settings

DD-WRT default settings related to wireless functionality are shown in Illustration 2.

Illustration 2: Default wireless settings

As shown in Illustration 3, the wireless MAC address filter is disabled by default.

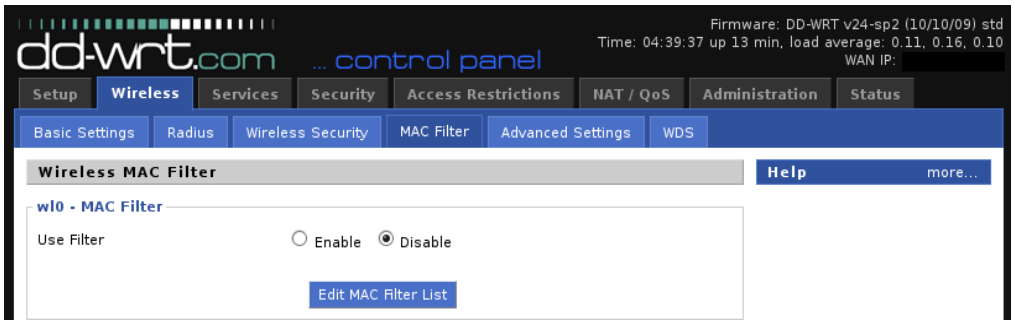

Illustration 3: Default MAC filter setting

Wireless security (i.e., encryption) is also disabled by default, as shown in Illustration 4.

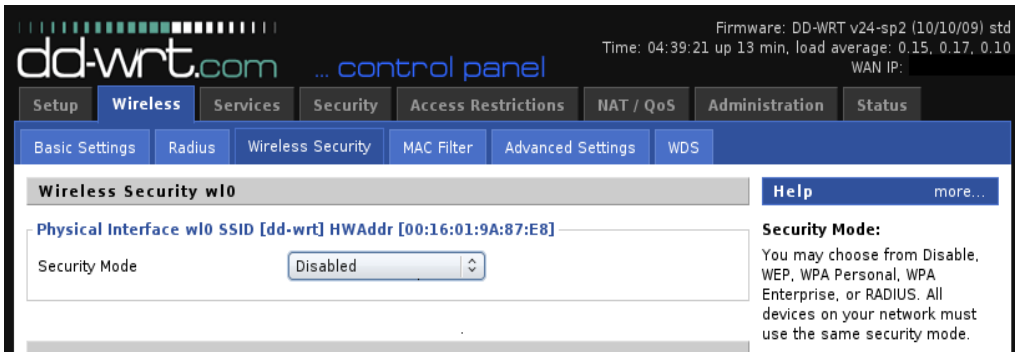

Illustration 4: Default wireless encryption setting

Default device management settings are shown in Illustration 5.

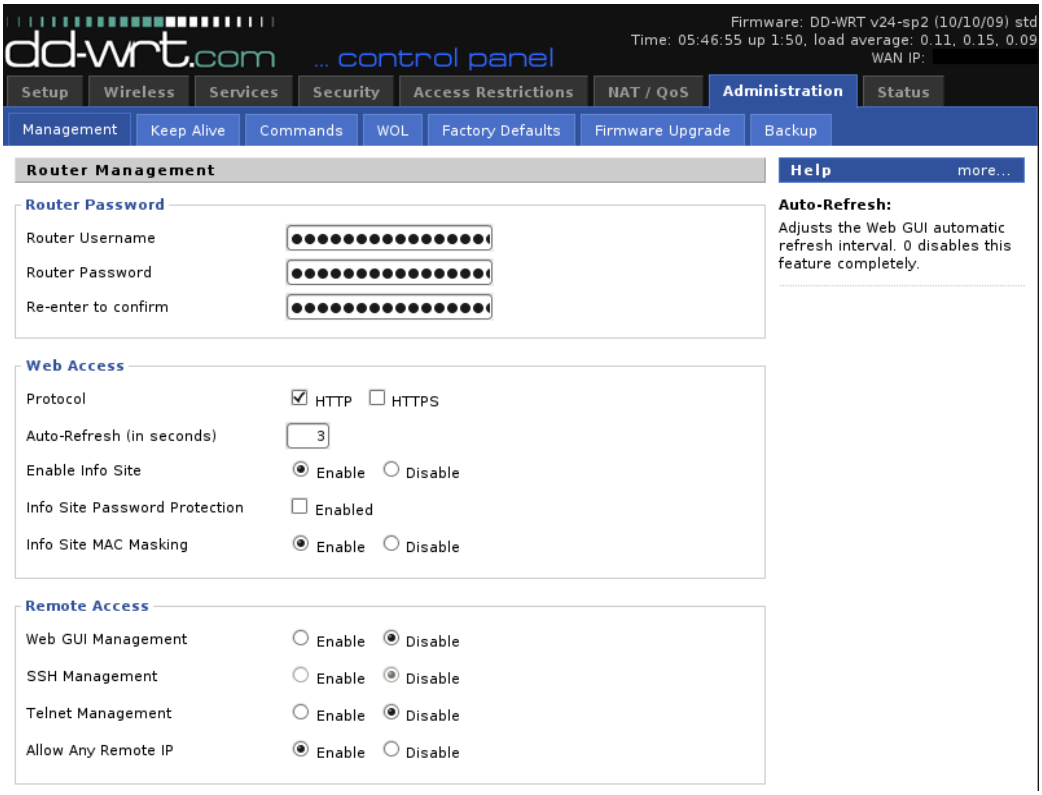

Illustration 5: Default device management settings

Default configuration attacks

The attacker can simply use a search engine to discover the default password settings for a DD-WRT AP. A search result is shown in Illustration 6.

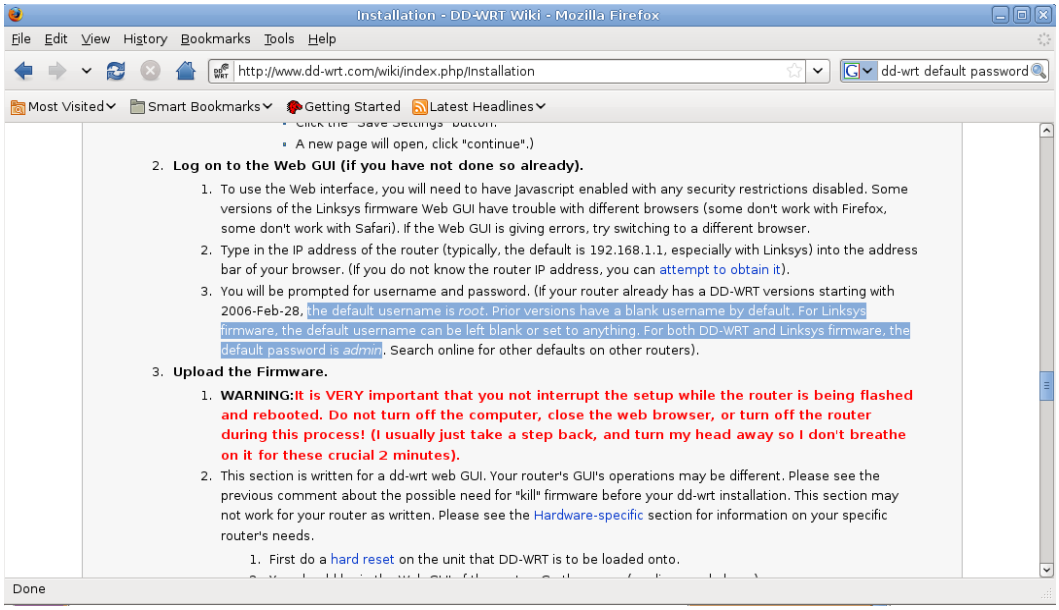

Illustration 6: Default password web search result

If the attacker is located on a local network segment[1], they can use a free tool such as ngrep [2] to capture device authentication credentials as they are sent across the network, such as when a legitimate administrator authenticates to the web interface. The use of this tool to capture in-transit authentication credentials is demonstrated in Illustration 7.

1 The attacker must be located either on a network within the same collision domain as the administrator and/or the AP (such as the wireless local network ('WLAN') or a hub-connected network), or perform CAM flooding or ARP spoofing attacks on a switched network (existing, freely-available tools can perform these attacks).

2 http://ngrep.sourceforge.net/

```
root@       ~# ngrep "Authorization: Basic"
interface: eth0 (10.0.0.0/255.255.255.0)
match: Authorization: Basic
#############
T 10.0.0.104:36205 -> 10.0.0.1:80 [AP]
  GET /index.asp HTTP/1.1..Host: 10.0.0.1..User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.18) Geck
  o/2010021501 Ubuntu/8.04 (hardy) Firefox/3.0.18..Accept: text/html,application/xhtml+xml,application/xml;q=0.9
  ,*/*;q=0.8..Accept-Language: en-us,en;q=0.5..Accept-Encoding: gzip,deflate..Accept-Charset: ISO-8859-1,utf-8;q
  =0.7,*;q=0.7..Keep-Alive: 300..Connection: keep-alive..Authorization: Basic cm9vdDphZG1pbg==...
##############################################################################################################
##############################################################################################################
############################
```

Illustration 7: ngrep authentication credentials capture

The authentication credentials captured in the example above are encoded as a Base-64 string. This encoding can be easily decoded using freely-available utilities, such as the web form[3] shown in Illustration 8.

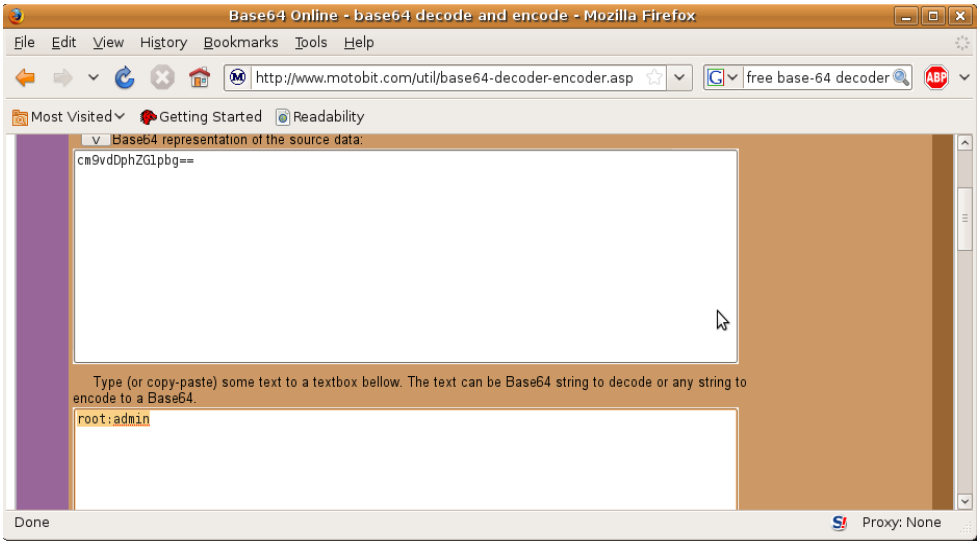

Illustration 8: Base-64 decoding utility

Alternatively, a local attacker can use the network protocol analyzing tool Wireshark[4] to capture and automatically decode the authentication credentials. Refer to Illustration 9.

3 http://www.motobit.com/util/base64-decoder-encoder.asp
4 http://www.wireshark.org/

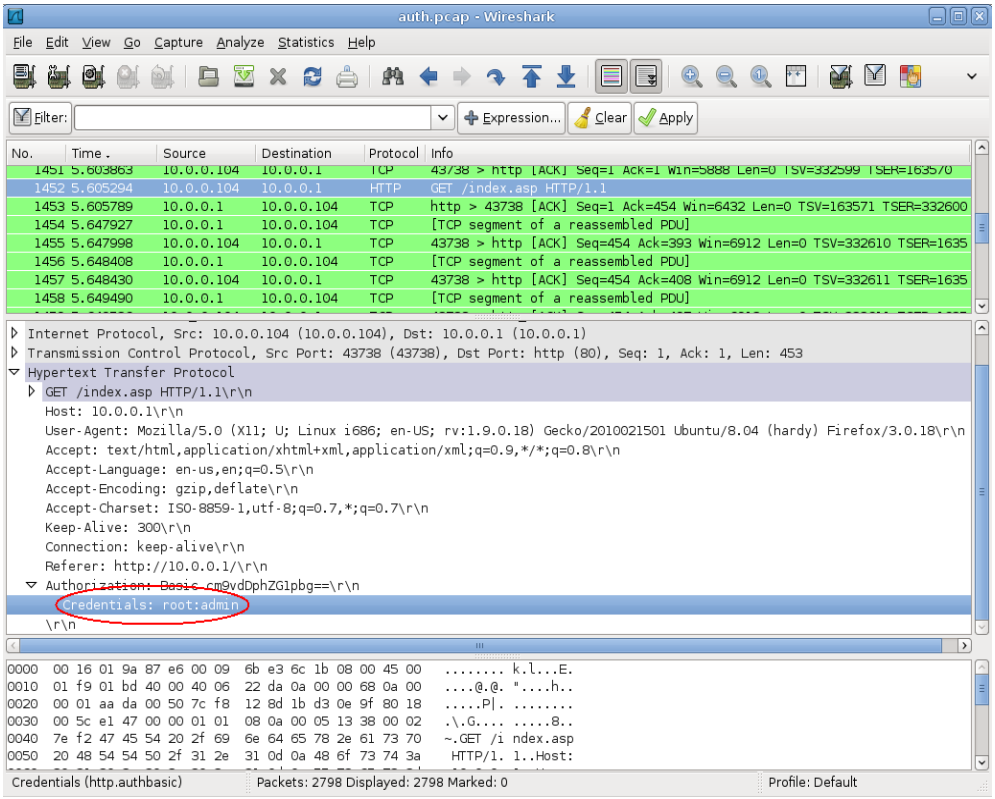

Illustration 9: Wireshark packet capture and decode

An attacker may exploit another opportunity on an open, unencrypted wireless network by placing their wireless interface into 'monitor mode'[5] to take advantage of the capability to sniff all wireless traffic, to and from all associated clients. This is done using commands similar to those shown in Illustration 10.

```
root@        ~# iwconfig wlan3 mode monitor channel 1
root@        ~# iwconfig wlan3
wlan3     802.11b/g  Mode:Monitor  Channel=1  Bit Rate=54 Mb/s
          Retry:on   Fragment thr:off
          Link Quality=0/100  Signal level=0 dBm  Noise level=0 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Illustration 10: Monitor mode syntax

The attacker can now use tcpdump[6] or a similar traffic-sniffing application to capture all wireless traffic, or only traffic matching certain parameters of interest. Sample syntax to capture all HTTP traffic (on port 80) to or from the client system at 10.0.0.116 is shown in Illustration 11.

5 http://en.wikipedia.org/wiki/Monitor_mode
6 http://www.tcpdump.org/

```
root@        ~# tcpdump -i 2 -nn -vv -X -s 0 -w traffic.pcap host 10.0.0.116 and port 80
```

Illustration 11: tcpdump capture syntax

The attacker can open the resulting packet capture file in Wireshark for easier scrutiny. This tool can compile potentially interesting traffic flows into a single window, as demonstrated by the email message shown in Illustration 12.

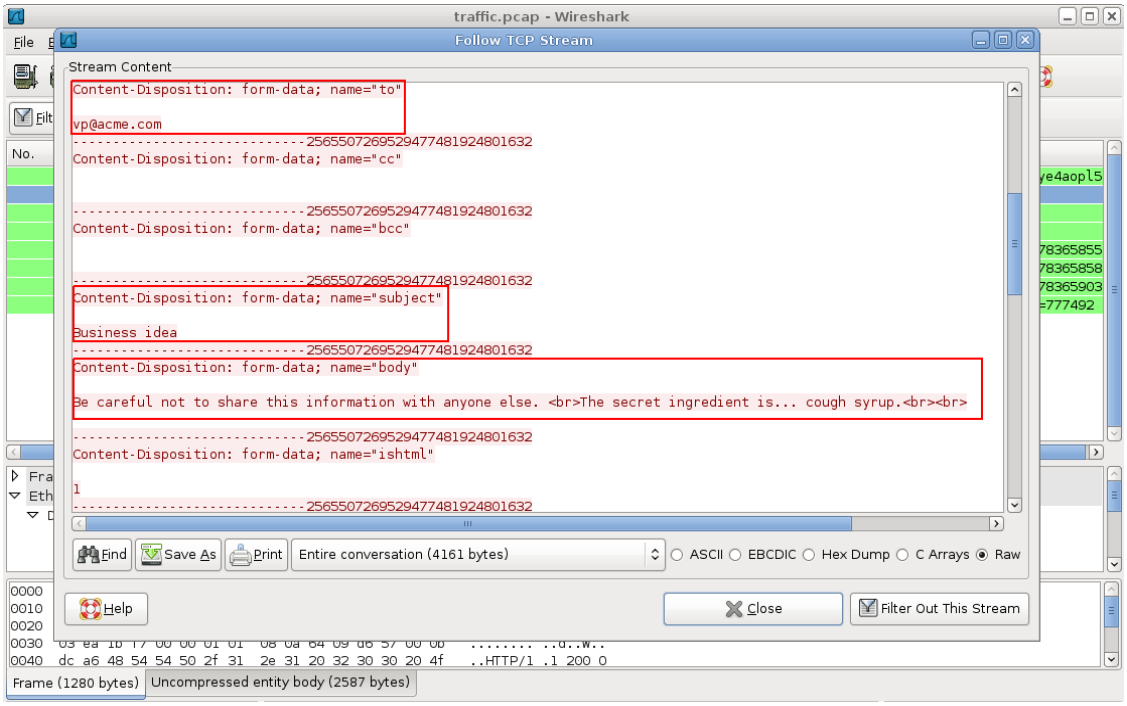

Illustration 12: Wireshark TCP stream view

WEP configuration overview

Basic wireless encryption can be enabled in DD-WRT on the 'Wireless Security' tab. The oldest, simplest mode of encryption is WEP[7]. The configuration shown in Illustration 13 enables 64-bit WEP encryption based on the passphrase '*ourweppassphrase*', resulting in a primary WEP key consisting of 10 hexadecimal characters: '*636166050E*'.

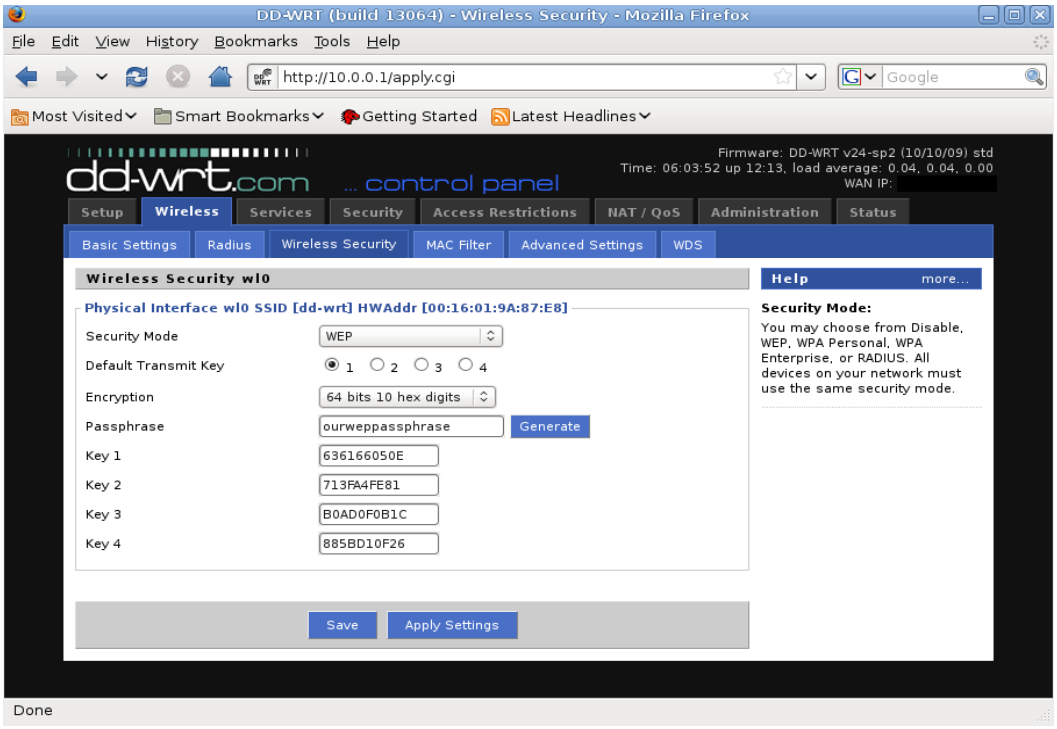

Illustration 13: WEP encryption configuration

7 http://en.wikipedia.org/wiki/Wired_Equivalent_Privacy

WEP configuration attacks

Attackers often first change the MAC (hardware) address of their network interface prior to conducting an attack, and re-change the value afterwards, in order to obscure the source of the attack. The default MAC address of the 'attacker' system for this demonstration is highlighted in Illustration 14.

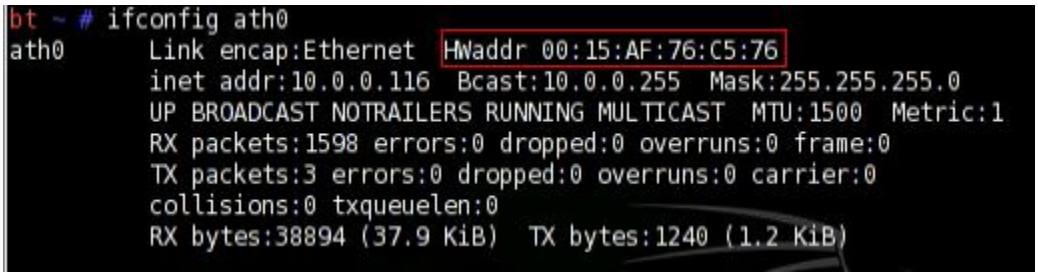

Illustration 14: Default MAC address of attacking system

The attacker can use the macchanger utility[8] to easily change the MAC address, as demonstrated in Illustration 15.

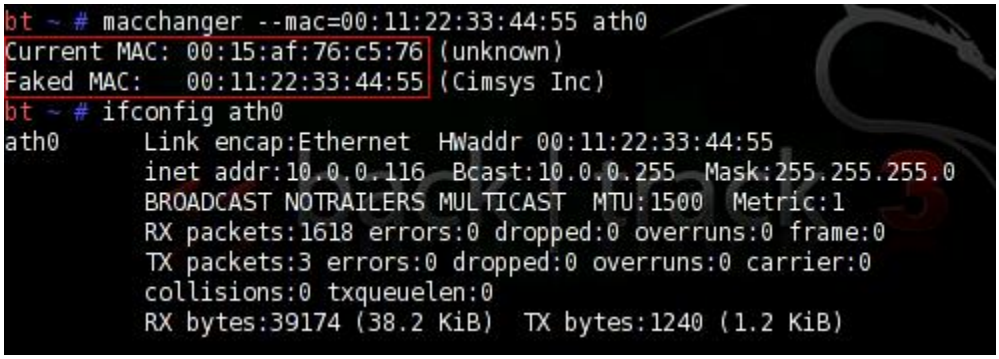

Illustration 15: macchanger usage

The attacker next places the wireless network interface into monitor mode, as confirmed by the output of the iwconfig[9] utility in Illustration 16.

8 http://www.alobbs.com/macchanger/
9 http://linux.die.net/man/8/iwconfig

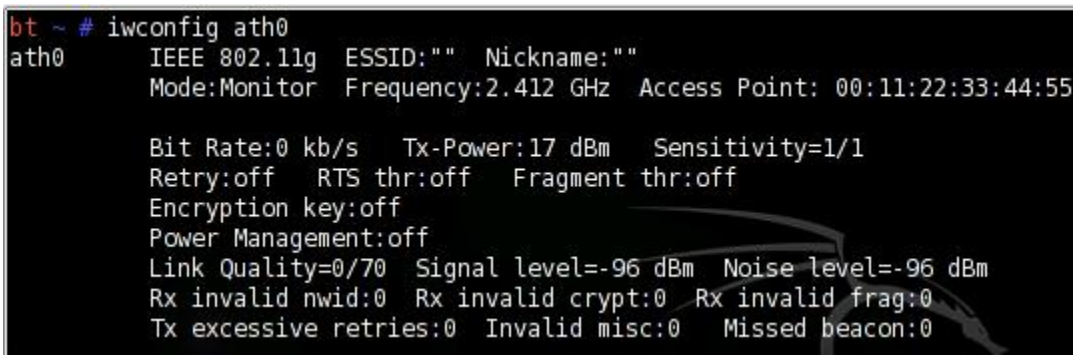

Illustration 16: iwconfig usage

Next, the attacker executes the Kismet tool[10] to detect and enumerate nearby wireless networks, as demonstrated in Illustration 17.

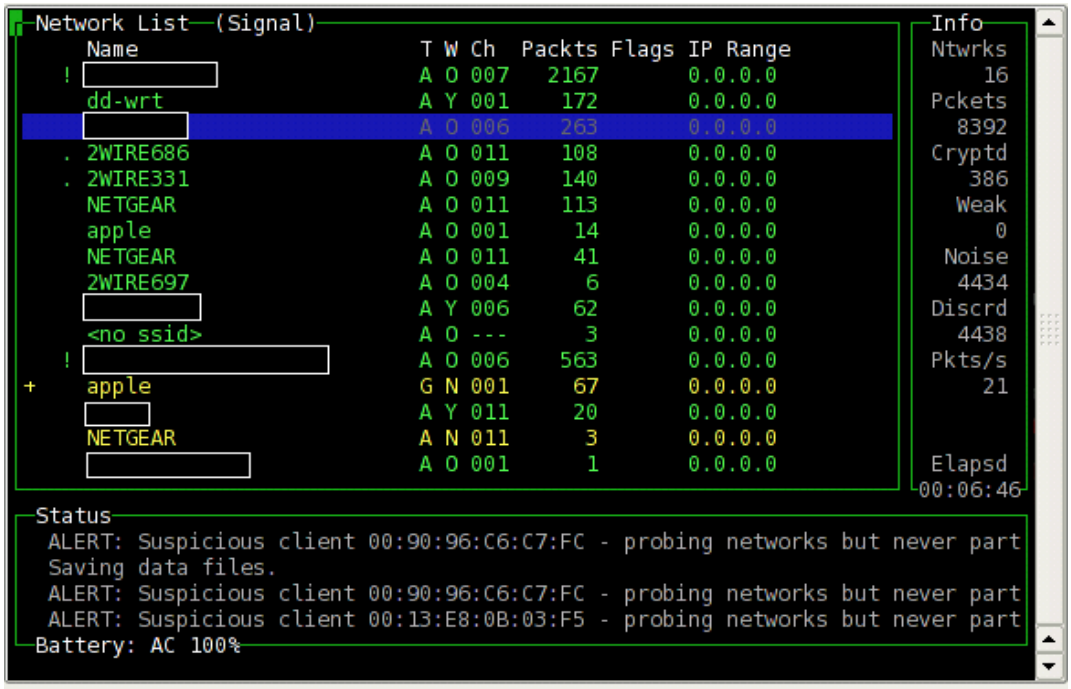

Illustration 17: Kismet Network list

Once the attacker has detected the target network of interest, they can select it to view additional details, as shown in Illustration 18. Note, in particular, the SSID, BSSID, Channel, and Encrypt values.

10 http://www.kismetwireless.net/

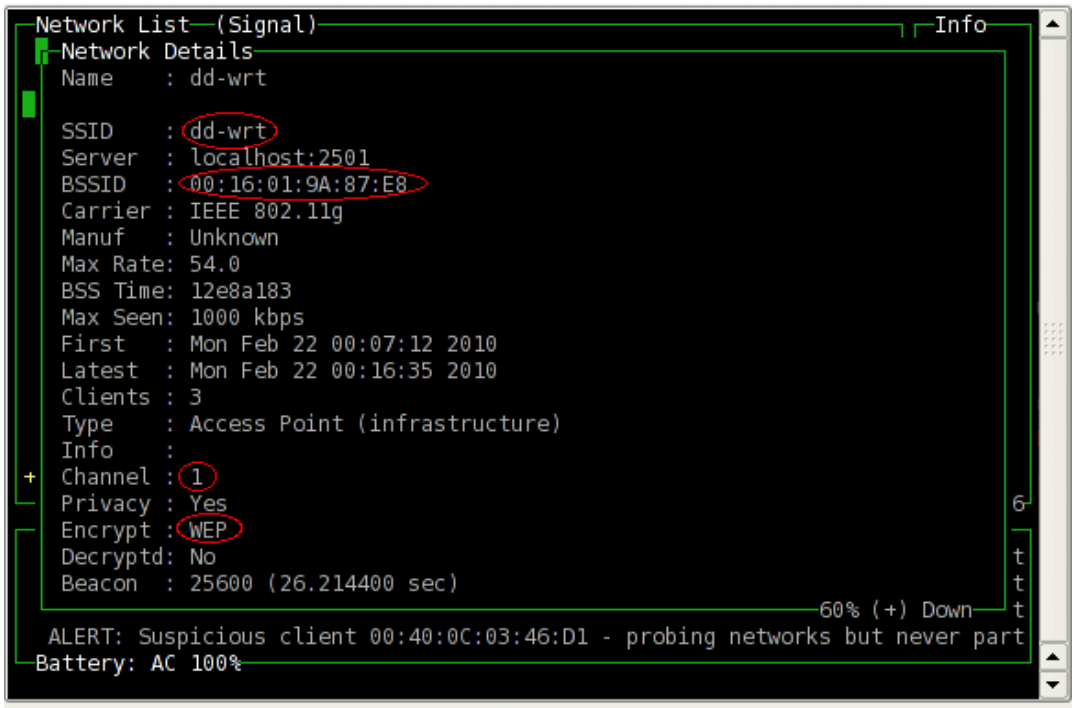

Illustration 18: Kismet target Network Details

The attacker then configures and runs airodump-ng[11] to sniff and record network traffic on the specified channel (-c 1) and wireless network (based on the BSSID – the AP's wireless interface MAC address). Example syntax is shown in Illustration 19.

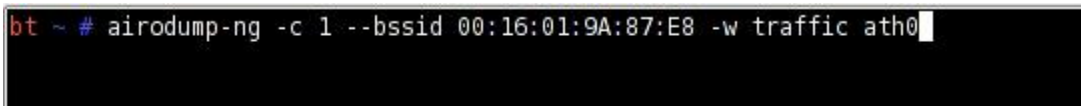

Illustration 19: Airodump traffic capture syntax

During the traffic capturing session, airodump displays status information regarding clients connected to the target AP, and the number of beacon and data packets captured, as shown in Illustration 20.

11 http://aircrack-ng.org/doku.php?id=airodump-ng

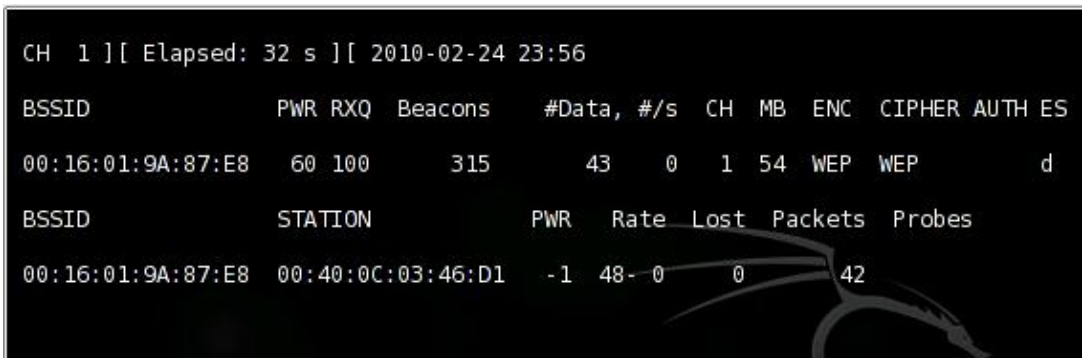

Illustration 20: Airodump status display

Next, the attacker tests the packet injection capabilities of their wireless network interface and the target AP using aireplay-ng[12]. The target network SSID ('dd-wrt') and BSSID are optional parameters for this test, as demonstrated in the successful test shown in Illustration 21.

Illustration 21: Aireplay packet injection test

In order to begin the active portion of a WEP attack, the attacker next uses aireplay to fake authentication to the target AP. 'Fake authentication' refers to the process of sending authentication and association requests to the AP, as though the client will subsequently send the WEP key in order to join the network. However, since the WEP key is not yet known, the attacker instead repeatedly sends these authentication and association requests, using the intervening time periods to conduct the attack. This ongoing state of being temporarily authenticated, pending submission of the WEP key, is known as fake authentication. Sample aireplay syntax and output for a fake authentication attack to the target AP every 30 seconds, from the client's recently-changed MAC address (00:11:22:33:44:55) is shown in Illustration 22.

12 http://www.aircrack-ng.org/doku.php?id=aireplay-ng

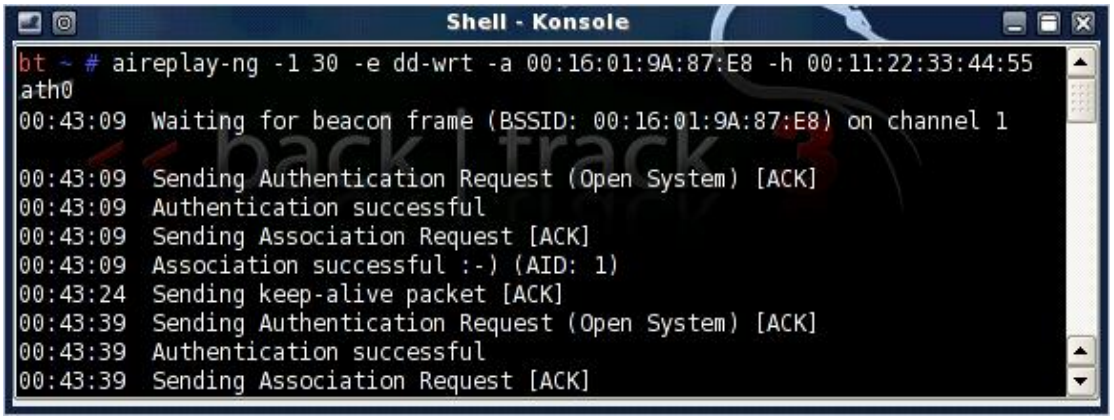

Illustration 22: Aireplay fake authentication attack

While the fake authentication attack continues to run in the background, the attacker begins an ARP request replay attack, also using aireplay. This attack listens for an ARP request on the target network, saves it, and then continuously resends it to the AP to be rebroadcasted. Meanwhile, the airodump session captures these broadcasted ARP packets, each with a new initialization vector ('IV'); these will be used to crack the WEP key. The syntax for such an attack is shown in Illustration 23.

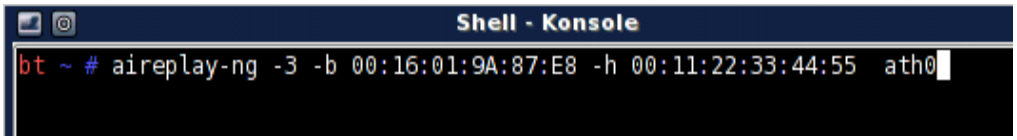

Illustration 23: Aireplay ARP replay attack syntax

The initial ARP request broadcast packet is identifiable as such, although encrypted, because of a fixed length and destination address (broadcast); it can be rebroadcasted continuously because WEP does not provide protection against such replay attacks.

An aireplay ARP replay attack is shown in action in Illustration 24.

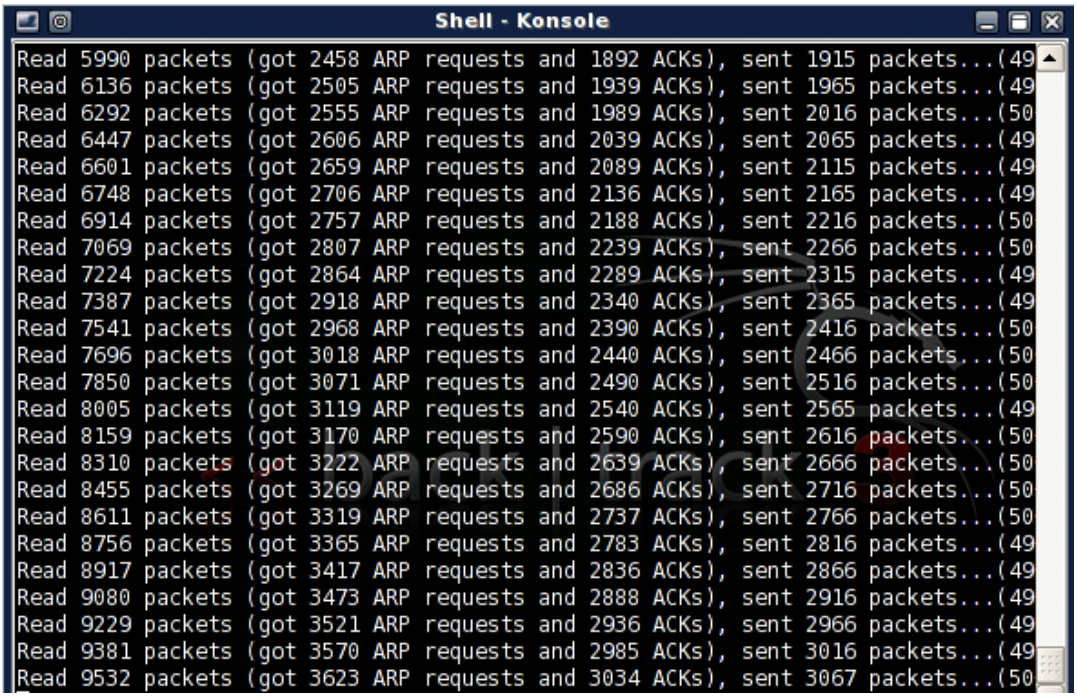

Illustration 24: Aireplay ARP replay attack in progress

Meanwhile, the airodump screen reports over 41,000 data packets captured after four minutes of the ARP replay attack, as shown in Illustration 25.

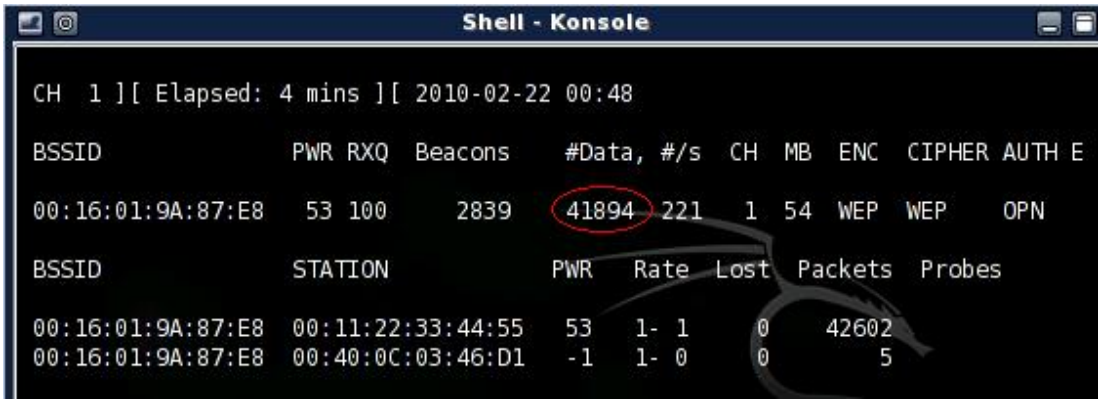

Illustration 25: Airodump capture progress

The attacker feeds the airodump packet capture file ('traffic.out-01.cap') and the AP BSSID as parameters to the aircrack-ng tool[13] as demonstrated in Illustration 26. With 47,178 IVs captured by airodump, this tool determined the WEP key quickly.

13 http://www.aircrack-ng.org/doku.php?id=aircrack-ng

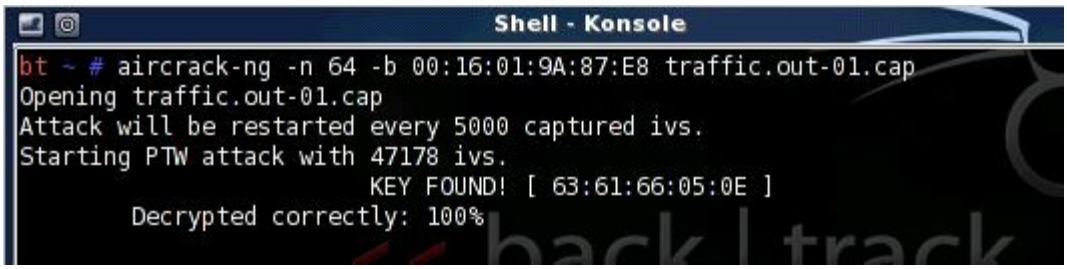

Illustration 26: Aircrack WEP key crack

Alternatively, the attacker could use the WEPCrack tool[14] to determine the WEP key. As shown in Illustration 27, using the same airodump capture file, this tool determined the WEP key in a fraction of a second.

Illustration 27: WEPcrack WEP key crack

The attacker now may use the newly-discovered WEP key with airdecap-ng[15] to retroactively decrypt the airodump capture file, containing WEP-encrypted network traffic. The decryption syntax is shown in Illustration 28.

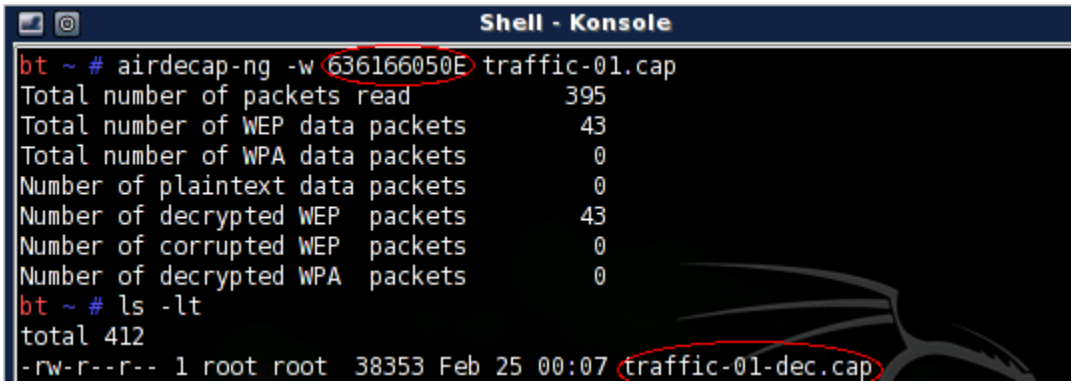

Illustration 28: Airdecap usage

The attacker can open the resulting output file, '*traffic-01-dec.cap*', in Wireshark to view the

14 http://wepcrack.sourceforge.net/
15 http://www.aircrack-ng.org/doku.php?id=airdecap-ng

unencrypted traffic, as shown in Illustration 29.

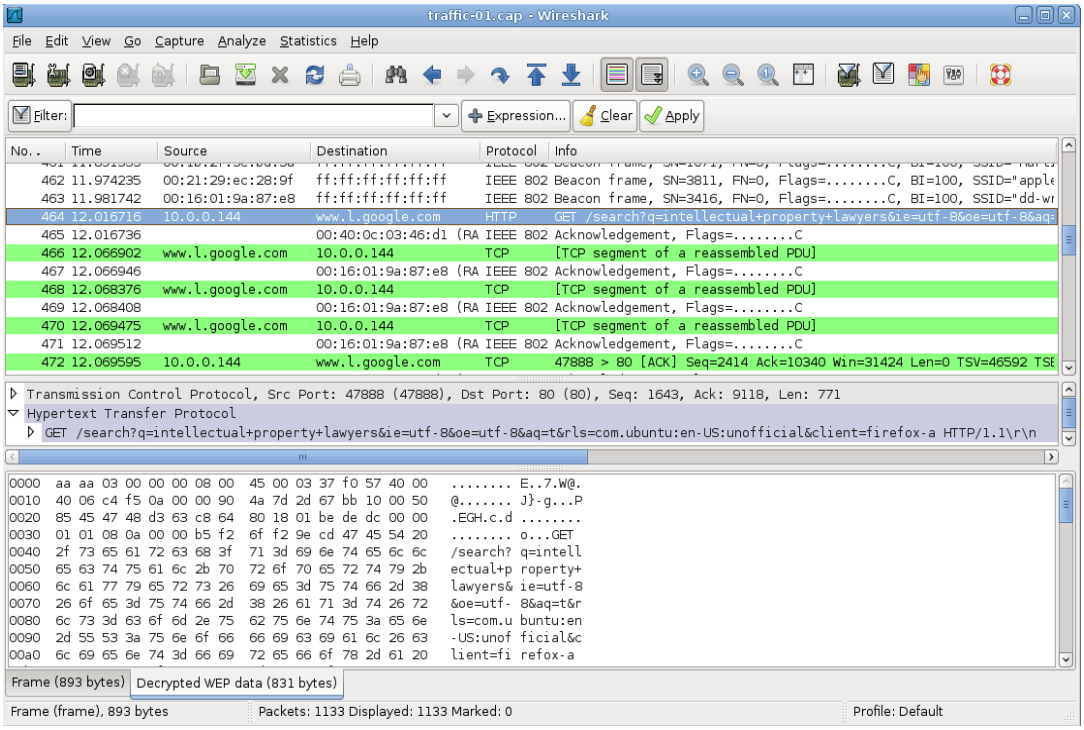

Illustration 29: Wireshark unencrypted traffic view

The 'Follow TCP stream' capability of Wireshark is used to better understand the contents of a TCP session, as demonstrated in Illustration 30.

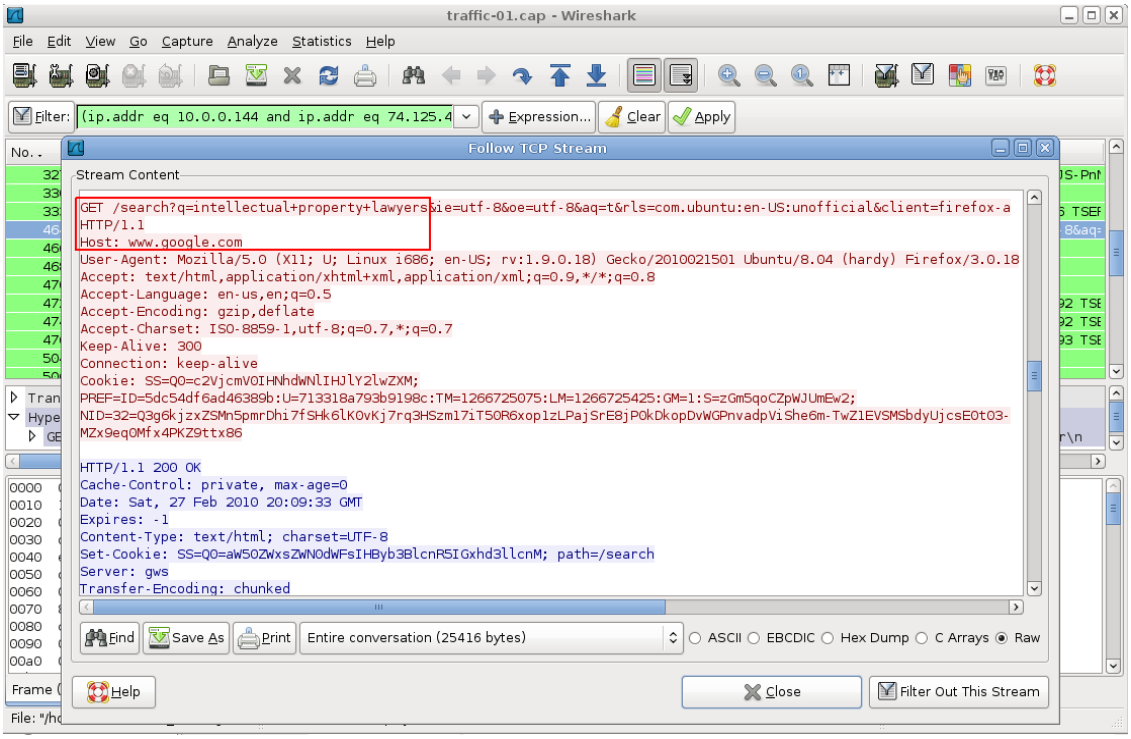

Illustration 30: Wireshark 'Follow TCP Stream' view

Alternatively, the attacker can decrypt the traffic within Wireshark itself, by providing the WEP key under the Preferences menu, Protocol section, IEEE 802.11, Key #1 field – shown in Illustration 31.

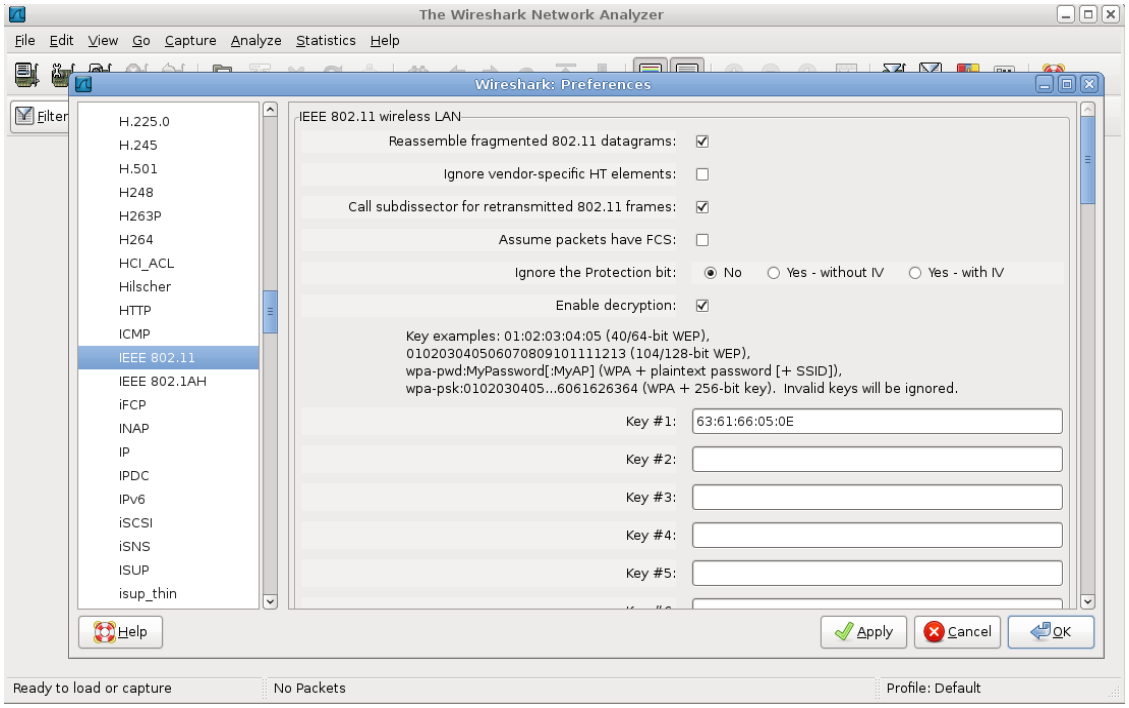

Illustration 31: Wireshark 802.11 decryption option

In order to get a clearer overall picture of what types of traffic has been captured, and to extract meaningful information from it, the attacker may use a tool such as Chaosreader[16] to parse the traffic capture file. The usage and output of this tool are shown in Illustration 32.

```
$ /opt/chaosreader0.94.pl -v ddwrt_traffic-dec.pcap

Chaosreader ver 0.94

Opening, ddwrt_traffic-dec.pcap

Reading file contents,
 100% (1724710/1724710)
Reassembling packets,
 100% (1823/1823)

Creating files...
   Num  Session (host:port <=> host:port)              Service
  0004  10.0.0.144:53922,[redacted]:993                imaps
  0003  10.0.0.144:53927,[redacted]:993                imaps
  0002  10.0.0.144:37807,[redacted]:25                 smtp
  0006  10.0.0.144:60857,[redacted]8:5190              aol
  0007  10.0.0.144:51481,74.125.47.125:5222            xmpp-client
  0005  10.0.0.144:52091,68.180.217.8:5050             mmcc
  0001  10.0.0.144:58520,10.0.0.1:53                   domain

index.html created.
```

Illustration 32: Chaosreader usage

16 http://chaosreader.sourceforge.net/

The report created by Chaosreader for this traffic capture file is shown in Illustration 33.

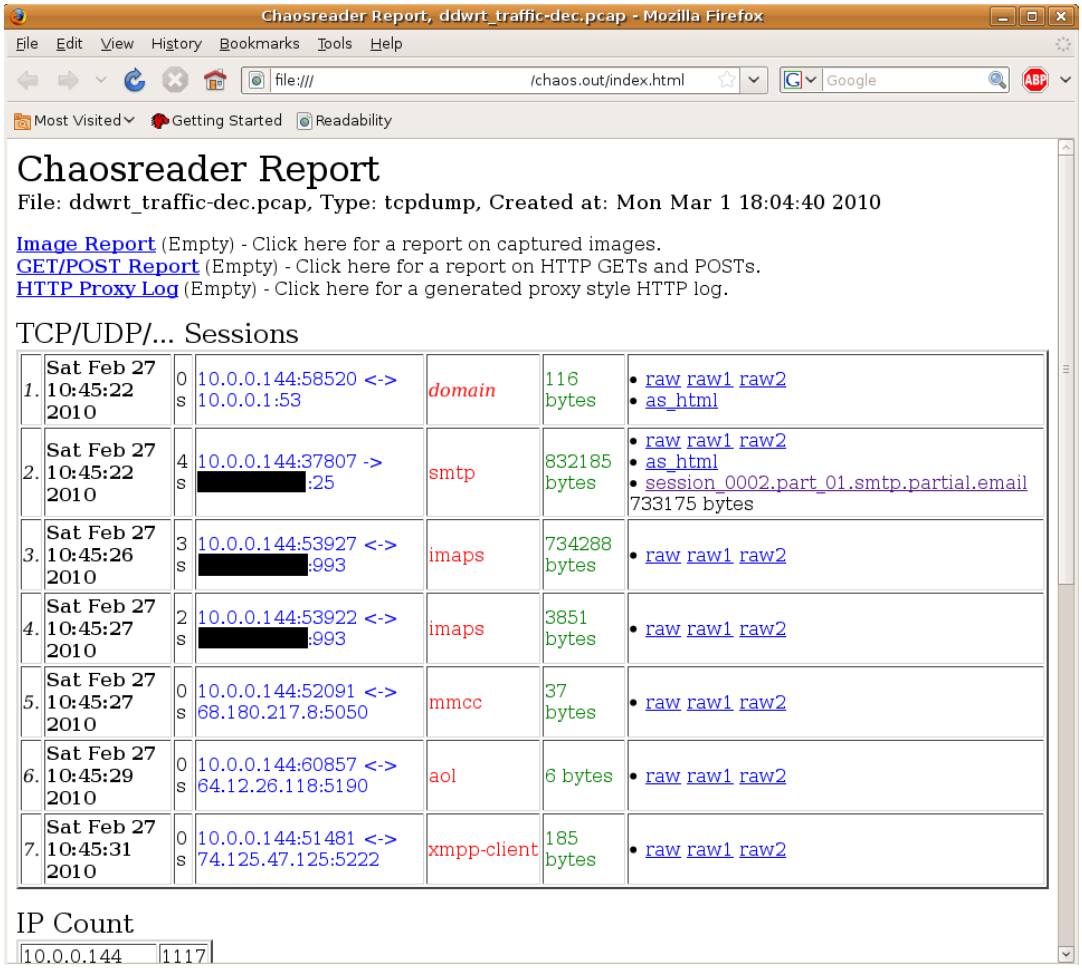

Illustration 33: Chaosreader report

In this instance, the tool reconstructed individual traffic sessions consisting of DNS, SMTP, IMAPS, and AOL IM protocols, among others.

The attacker can view the extracted contents of an email sent on the target network by clicking on the SMTP session link. This session is partially shown in Illustration 34.

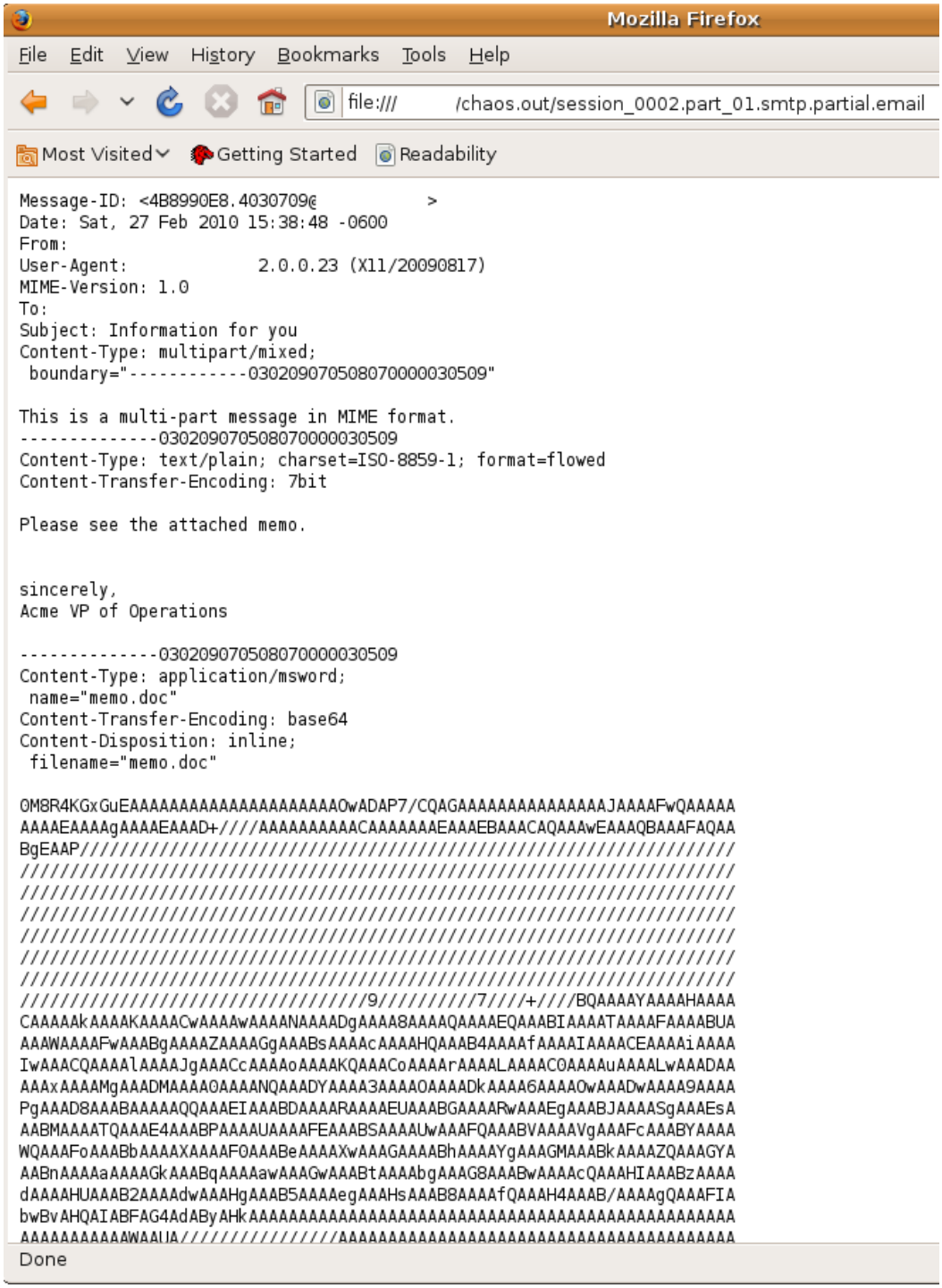

Illustration 34: SMTP session data from Chaosreader

Per the 'Content-Transfer-Encoding' parameter, the Microsoft Word attachment to this email is transferred in Base-64 encoding. The attacker can copy the long block of data following the

filename=”memo.doc” line and decode it using a freely-available Base-64 decoding tool, such as shown in Illustration 35.

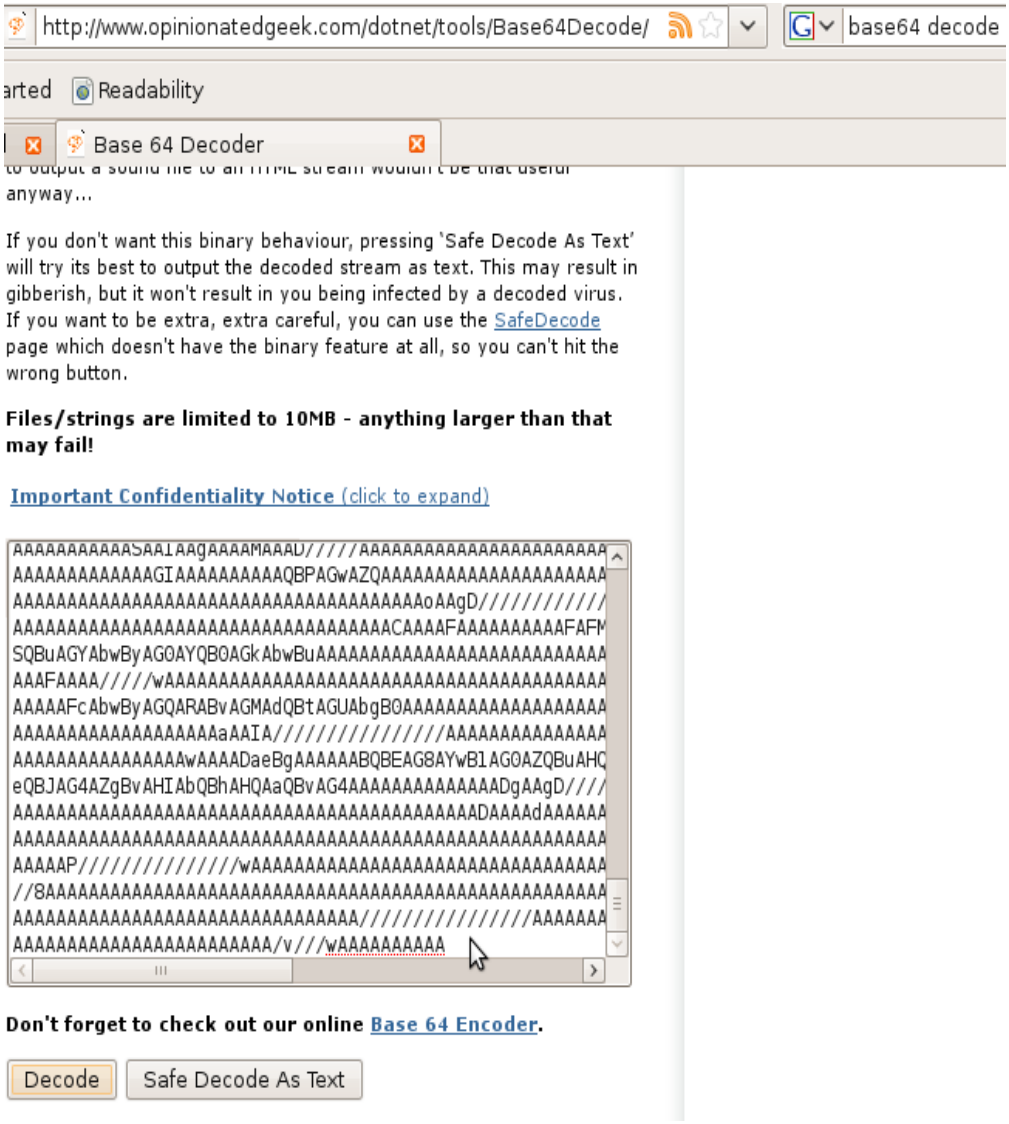

Illustration 35: Base-64 decoder form

The resulting output file can be opened as a Microsoft Word document using OpenOffice.org's Writer application, as demonstrated in Illustration 36.

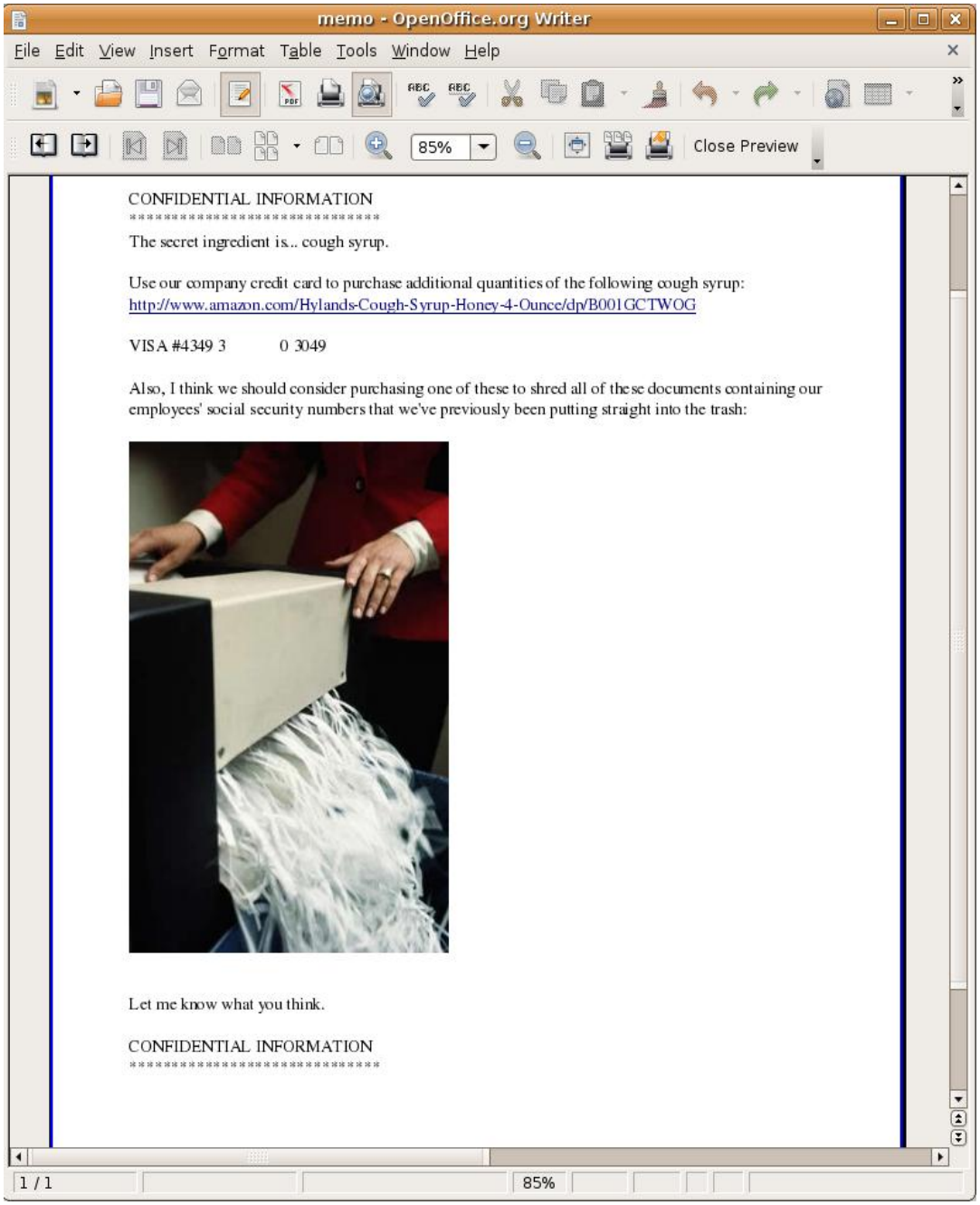

Illustration 36: Decoded Microsoft Word document

The attacker can also use ngrep to parse the decrypted network capture file for strings indicating authentication traffic, as demonstrated in Illustration 37.

```
                    $ ngrep AUTH -I traffic-dec.pcap
input: traffic-dec.pcap
match: AUTH
##########
T           :25 -> 10.0.0.144:58416 [AP]
  250-mail.               .com..250-PIPELINING..250-SIZE 71000000..250-ETRN..250-STARTTLS..250-AUTH L
  OGIN PLAIN..250-AUTH=LOGIN PLAIN..250-ENHANCEDSTATUSCODES..250 8BITMIME..
##
T 10.0.0.144:58416 ->            :25 [AP]
  AUTH PLAIN  mWFbilkMzdAZm1h3k89aWwuZmuyRejbPla1tK7d3
##########################################################################################################
##########################################################################################exit
```

Illustration 37: ngrep authentication search

The returned authentication credentials are encoded as a Base-64 string, which are easily decoded to reveal the SMTP (email) username and password, as shown in Illustration 38.

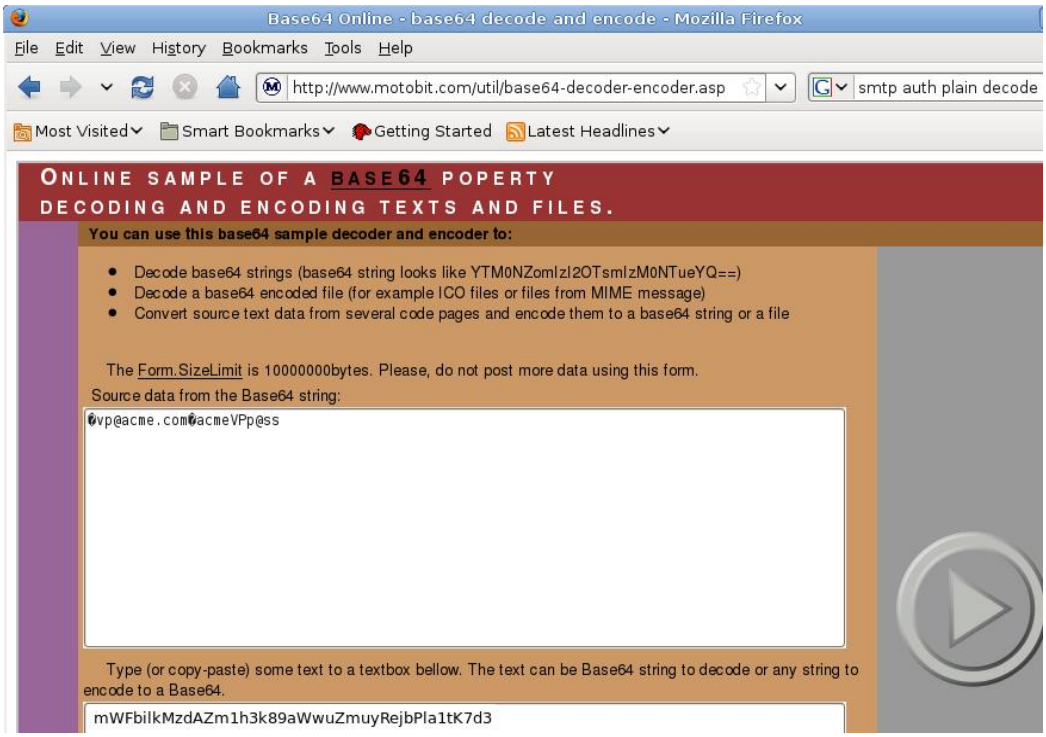

Illustration 38: Base-64 string decoding

Using the discovered WEP key to configure their wireless interface, the attacker can connect to the target network to interact with other clients. The attacker changes their wireless network interface from monitor mode to managed mode in order to associate with the target AP, as shown in Illustration 39.

```
bt ~ # wlanconfig ath0 create wlandev wifi0
ath0
bt ~ # ifconfig ath0 up
```

Illustration 39: Wireless interface managed mode syntax

The attacker next enters the WEP key into the configuration wizard prompt, as shown in Illustration 40.

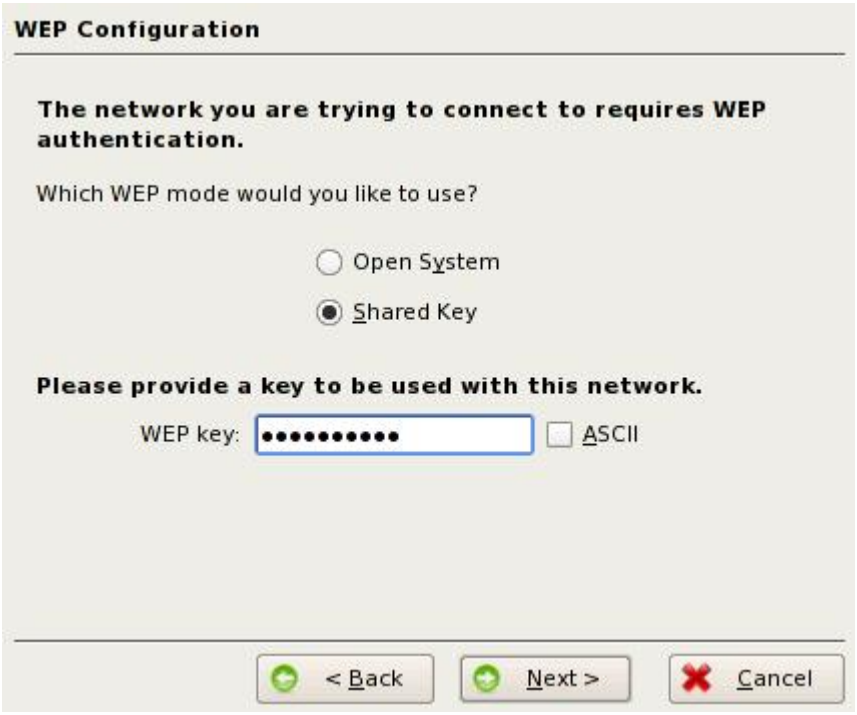

Illustration 40: WEP configuration wizard

As indicated in Illustration 41, the attacker's connection to the network has been successful.

Illustration 41: Successful connection message

The details of this network connection are revealed by issuing the `ifconfig` and `iwconfig` commands, as demonstrated in Illustration 42.

```
bt ~ # ifconfig ath0
ath0      Link encap:Ethernet  HWaddr 06:15:AF:76:C5:FB
          inet addr:10.0.0.124  Bcast:10.0.0.255  Mask:255.255.255.0
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1620 (1.5 KiB)  TX bytes:1240 (1.2 KiB)

bt ~ # iwconfig ath0
ath0      IEEE 802.11g  ESSID:"dd-wrt"  Nickname:""
          Mode:Managed  Frequency:2.412 GHz  Access Point: 00:16:01:9A:87:E8
          Bit Rate:54 Mb/s   Tx-Power:17 dBm   Sensitivity=1/1
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:6361-6605-0E   Security mode:restricted
          Power Management:off
          Link Quality=59/70  Signal level=-36 dBm  Noise level=-95 dBm
          Rx invalid nwid:700  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Illustration 42: Network connection details

The attacker, now wirelessly connected to the local network, can scan, enumerate, and attack other clients and servers on the network. They may also connect to the AP web interface as shown in Illustration 43 (the main page does not require authentication, by default).

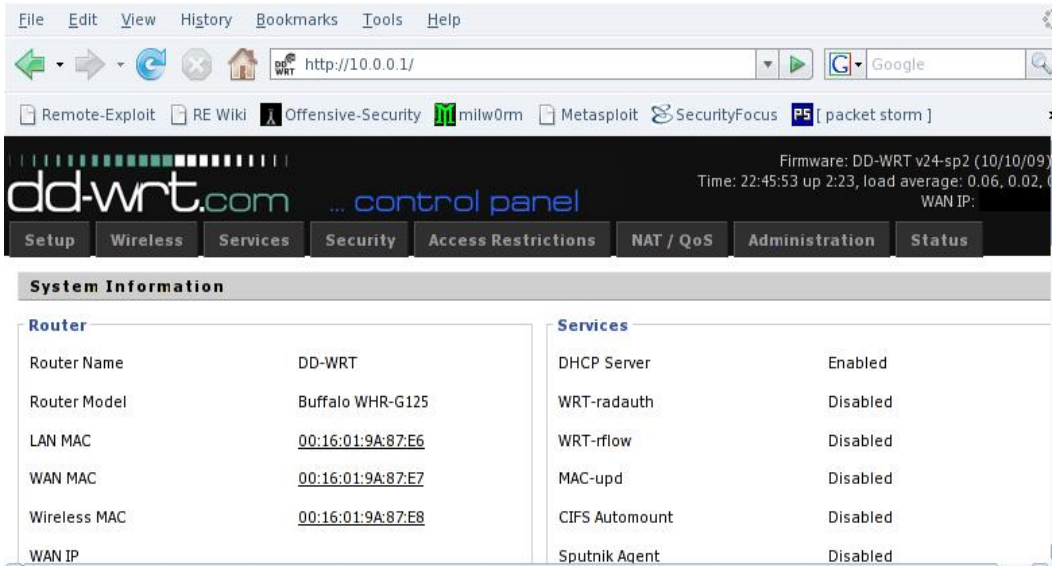

Illustration 43: AP web interface

WPA TKIP configuration overview

A more advanced mode of encryption, WPA ("Wi-Fi Protected Access")[17], is also configurable on the Wireless Security tab in the DD-WRT web interface. The configuration shown in Illustration 44 enables PSK ("pre-shared key")-based WPA 'personal' encryption utilizing the TKIP algorithm. The shared key is specified here as '*acmepass*'.

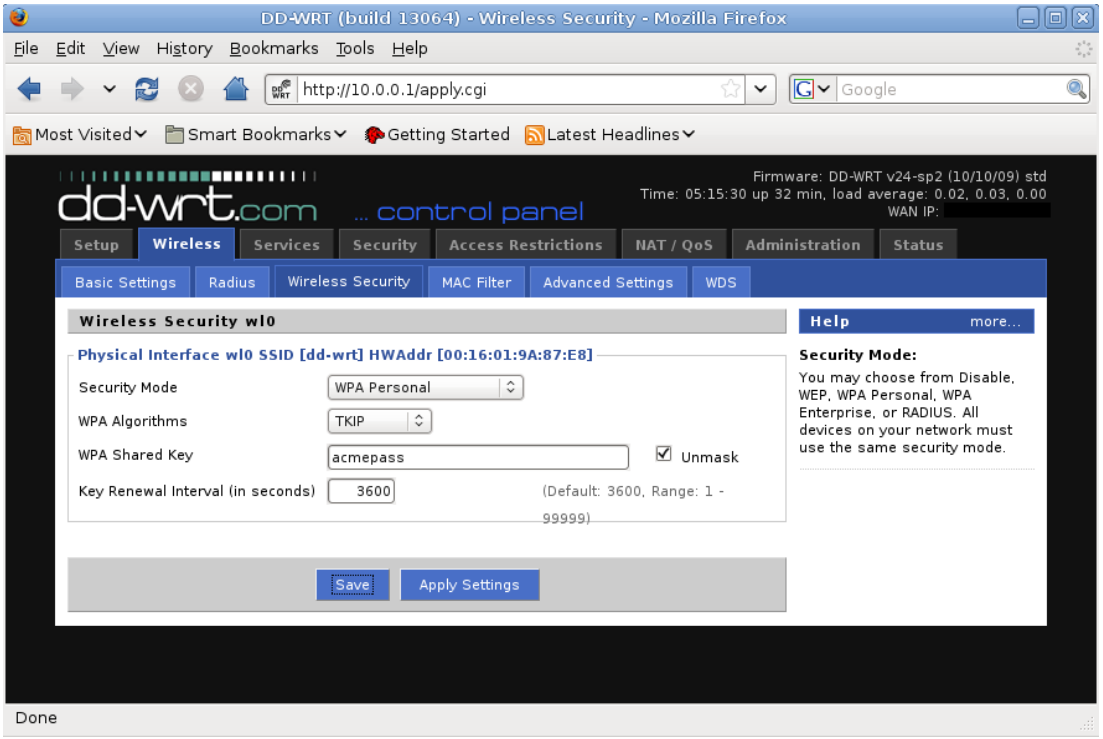

Illustration 44: WPA Personal encryption configuration

17 http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access

WPA TKIP configuration attacks

With their wireless network interface in monitor mode, the attacker executes Kismet to detect and enumerate the target wireless network. They can then select it in order to view additional details, as shown in Illustration 45.

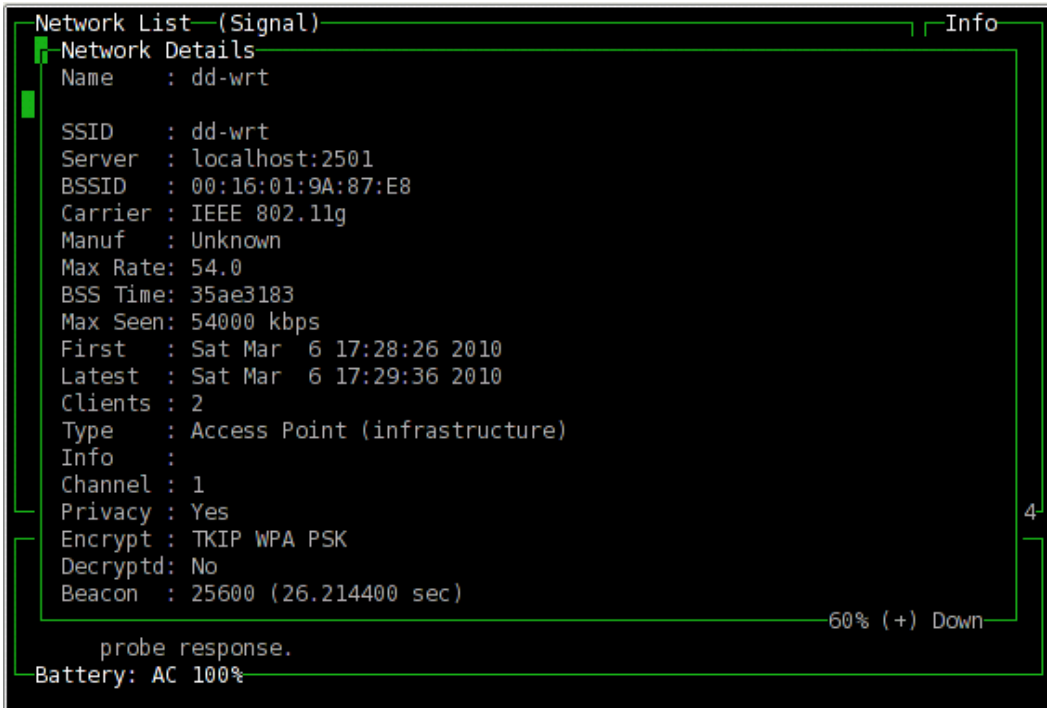

Illustration 45: Kismet target Network Details

Note the Encrypt values for the target network now specify TKIP (the algorithm), WPA (the security protocol), and PSK (pre-shared key).

The attacker then configures and runs airodump-ng to sniff and record network traffic on the target network. The airodump status display during this capture is shown in Illustration 46.

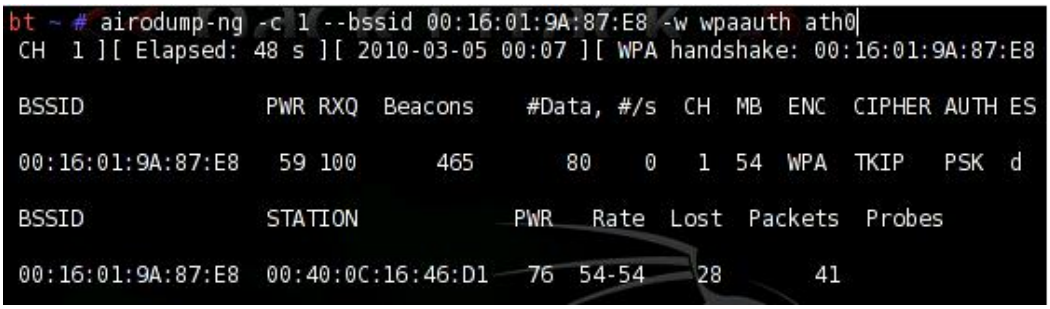

Illustration 46: Airodump syntax and status display

The WPA PSK decryption attack relies on capturing the authentication sequence between a client and AP. If these authentication sequences do not occur regularly enough on the target network, the attacker can use aireplay to selectively deauthenticate a known client (based on MAC address), or deauthenticate all users simultaneously. A deauthentication attack is demonstrated in Illustration 47.

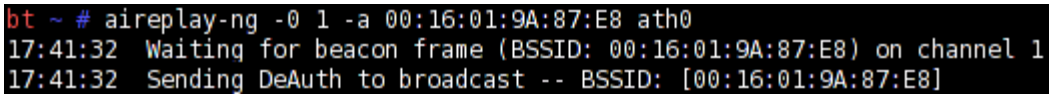

Illustration 47: Aireplay deauthentication attack

Shortly following this deauthentication attack, the attacker can terminate the airodump session, likely having captured the network traffic containing the targeted clients' (re-)authentication sequences. The attacker can use the coWPAtty tool[18] to attempt to crack the PSK using a dictionary attack method, based upon the packet capture file containing a client's authentication sequence ('*wpaauth-01.cap*') and the target AP's network name ('dd-wrt'), as demonstrated in Illustration 48.

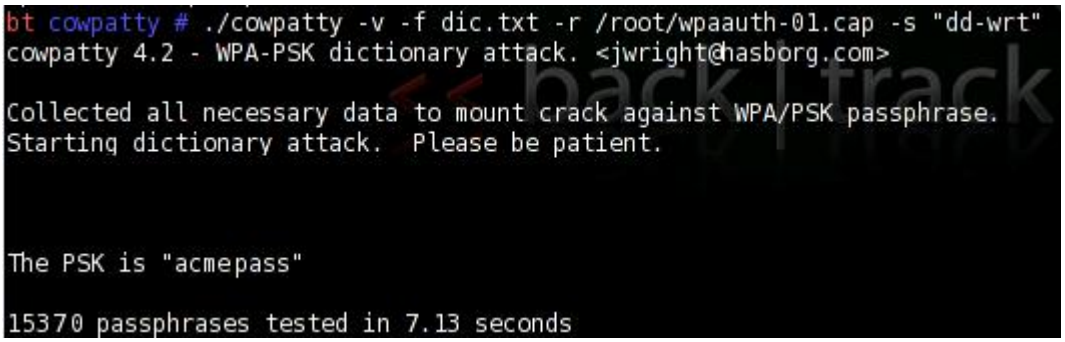

Illustration 48: CoWPAtty PSK cracking

Alternatively the attacker can use the aircrack-ng tool similarly to crack the PSK using a dictionary attack, as demonstrated in Illustration 49.

18 http://wirelessdefence.org/Contents/coWPAttyMain.htm

Illustration 49: Aircrack PSK cracking

A different approach to WPA PSK cracking involves precomputing hash files containing many PMK ('Pairwise Master Key') values, using the network name (SSID) combined with a list of possible password values. This approach is modeled after the 'rainbow table' technique[19]. The attacker can use the genpmk tool included with coWPAtty to create a hash table of PMK values, as demonstrated in Illustration 50.

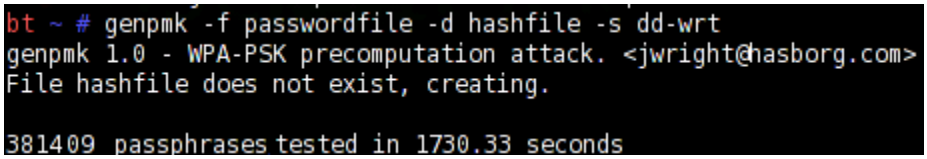

Illustration 50: Genpmk tool syntax

The attacker then runs coWPAtty, using the hash table as an input to attempt to determine the PSK, as demonstrated in Illustration 51.

19 http://en.wikipedia.org/wiki/Rainbow_table

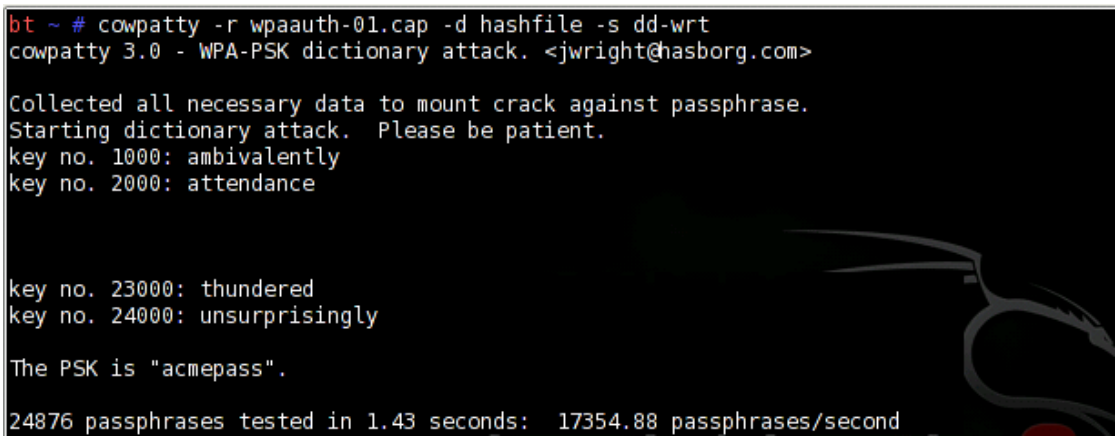

Illustration 51: CoWPAtty PSK cracking with hash table input

When successful, this cracking technique tends to be considerably faster than traditional password cracking methods.

Using the discovered WPA key to configure their wireless interface, the attacker can connect to the target network to interact with other clients or connect to the WPA-enabled access point management interface. The WPA network configuration wizard is shown in Illustration 52.

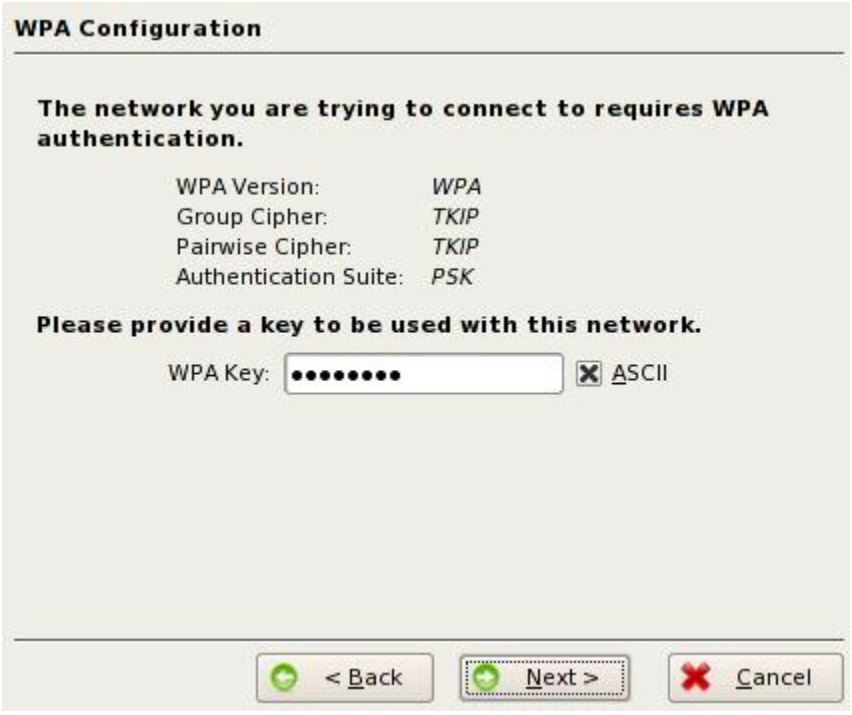

Illustration 52: WPA Configuration Wizard

As indicated in Illustration 53, the attacker's connection to the network has been successful.

Illustration 53: Successful connection message

The details of this network connection are shown in the output of the `ifconfig` and `iwconfig` commands, as shown in Illustration 54.

```
bt ~ # ifconfig ath0
ath0      Link encap:Ethernet  HWaddr 00:15:AF:76:C5:FB
          inet addr:10.0.0.116  Bcast:10.0.0.255  Mask:255.255.255.0
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2414 (2.3 KiB)  TX bytes:1603 (1.5 KiB)

bt ~ # iwconfig ath0
ath0      IEEE 802.11g  ESSID:"dd-wrt"  Nickname:""
          Mode:Managed  Frequency:2.412 GHz  Access Point: 00:16:01:9A:87:E8
          Bit Rate:36 Mb/s   Tx-Power:17 dBm   Sensitivity=1/1
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:5F94-9ABB-E8B4-2A1E-FB53-FC0F-D9AC-603F   Security mode
:restricted
          Power Management:off
          Link Quality=50/70  Signal level=-46 dBm  Noise level=-96 dBm
          Rx invalid nwid:6572  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Illustration 54: Network connection details

Note the 'Encryption key' field, which contains the PTK ('Pairwise Transient Key'). This value is derived from the PMK, which is in turn derived from the PSK and SSID.

The attacker can use the newly-discovered PSK with airdecap-ng to decrypt WPA-encrypted

network traffic from the airodump capture, between the AP and any clients whose four-way authentication handshake sequences were captured. The syntax to accomplish this is shown in Illustration 55, where 'acmepass' is the PSK.

```
          $ airdecap-ng -e dd-wrt -p acmepass wpaauth-01-II.cap
Total number of packets read           217
Total number of WEP data packets         0
Total number of WPA data packets        22
Number of plaintext data packets         0
Number of decrypted WEP  packets         0
Number of corrupted WEP  packets         0
Number of decrypted WPA  packets        22
```

Illustration 55: Airdecap usage

Alternatively, the attacker can decrypt the traffic within Wireshark itself, by providing the WPA PSK under the Preferences menu, Protocol section, IEEE 802.11, Key #1 field. This field is shown in Illustration 56.

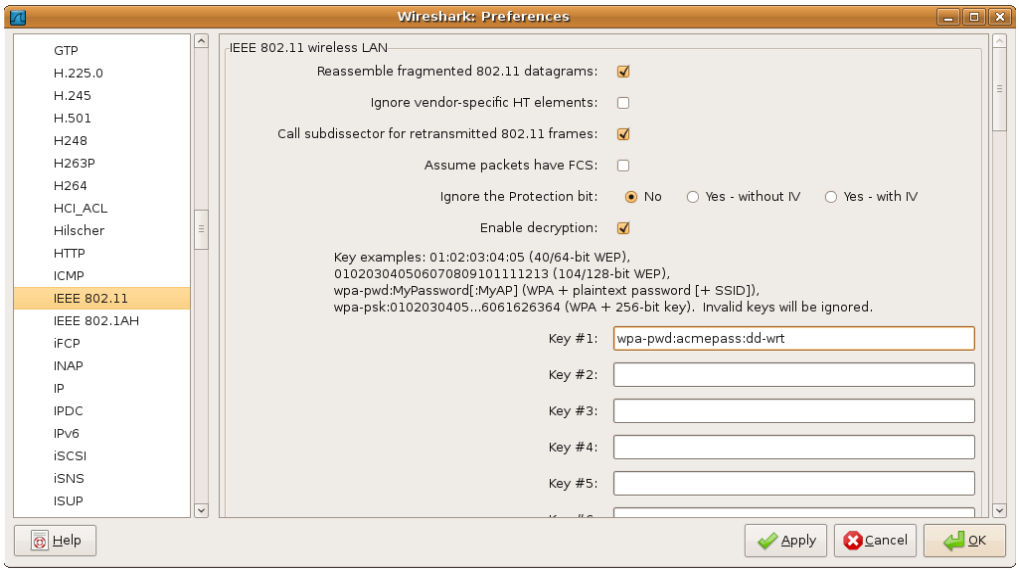

Illustration 56: Wireshark 802.11 WPA decryption option

Recommendations

Recommended settings for a wireless access point serving a home or small-to-medium-sized business network include:

- WPA2 Personal encryption utilizing the AES algorithm, based on a pre-shared key (PSK)
- Specify the pre-shared key as a non-dictionary-based passphrase exceeding 11 alphanumeric characters[20].
- Change the default SSID value to a non-common identifier.
- Regularly check for and apply firmware updates from the WAP vendor.
- Set the WAP administrative password to a difficult-to-guess value.
- Configure the WAP to allow web management via HTTPS, not HTTP.
- Enable the WAP firewall, if available.
- Configure the WAP to log network and administrative activities; periodically review the logs for suspicious activity.
- Power off the WAP and WiFi clients when not in use for extended periods of time.

The following settings tend to come at the cost of increased complexity of configuration, and provide only moderate increases in security, because they can all be bypassed with relative ease by an attacker. They may however deter casual intruders. As such, they are not recommended for most network implementations, but are provided here for information purposes, and for consideration on a network-by-network basis:

- Disable SSID broadcasts. An attacker can still discover 'hidden' SSIDs without difficulty; this setting only prevents them from being broadcasted as noticeably.
- Implement MAC address client filtering. An attacker can bypass these restrictions rather easily by cloning an authenticated client's MAC address.
- Disable the WAP DHCP server, and provide static IP assignments only to known clients. An attacker with knowledge of the network IP address range may still assign themselves a static IP address.
- Configure a non-standard internal IP address range. This will only prevent an attacker from guessing a commonly-used network address if they are unable to enumerate the range in use.

---

20 Depending on the security needs of a network, the PSK can be specified as anywhere from eight to 63 characters, or even a randomly-generated string of 64 hexadecimal characters. The guidance provided here to exceed 11 characters provides a reasonable level of security in conjunction with the other recommendations.