HTB{voicemail}

created.by{rjamison}

CONCEPT

SEND/RECEIVE OBSCURED MESSAGES USING DTMF SIGNALS AND DECIMAL ENCODING

DTMF Signaling

Dual-Tone Multi-Frequency (DTMF) Signals are the sounds your phone makes when you press a number. Each individual number or key press corresponds to a pair of frequencies - a low tone and high tone.

Button	Low Frequency (Hz)	High Frequency (Hz)
1	697	1209
2	697	1336
3	697	1477
4	770	1209
5	770	1336
6	770	1477
7	852	1209
8	852	1336
9	852	1477
0	941	1336
*	941	1209
#	941	1477

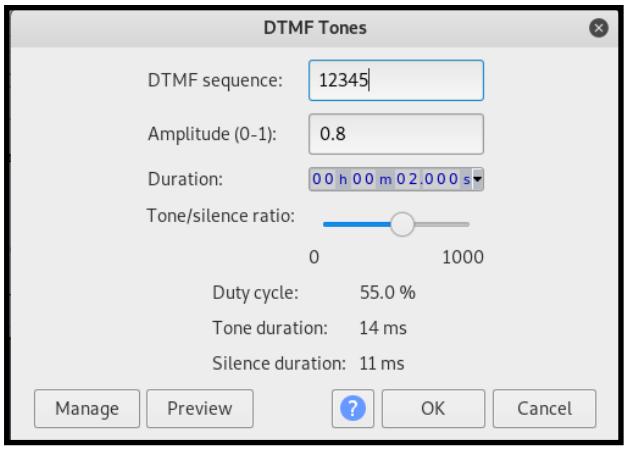
Table 1 - List of Dual-Tone Multi-Frequency Signals

Due to the proliferation of telephones across the globe, the sounds of DTMF signals are quite familiar to most people. As use of DTMF has become more widespread, many sound engineering tools and software can now generate and interpret DTMF.

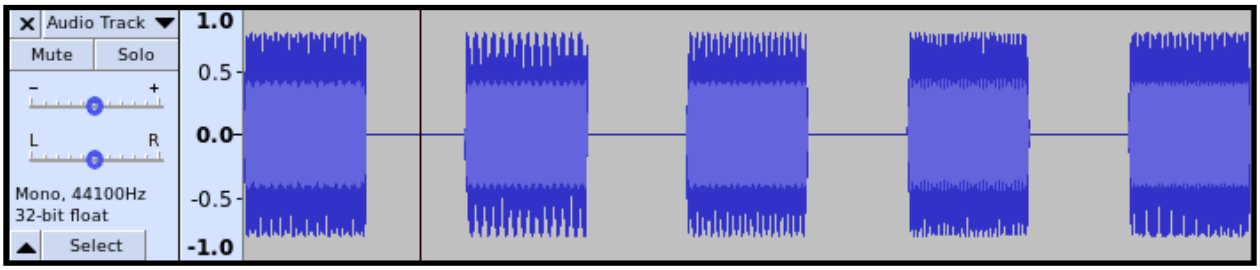
One of the best free tools available is Audacity, an open-source sound editor that allows users to generate DTMF using a GUI. As an example, we will generate the tones for **12345** using Audacity, but

you can input any of the following characters: ***1234567890*#***

In the menu, go to **Generate > DTMF**. You should see a pop up window that looks similar to our image on the right. In **DTMF Sequence**, type **12345** numbers. Depending on how much time you will have to transmit your data, we can also adjust our tone series **Duration**. In this case, we set it to **2 seconds**. Hit **OK** to generate.



As you can see below, what we end up with is an audio track. If you click **Play**, you'll hear what sounds like a telephone dialing.



Now let's save this as a file we can actually open. Go to **File > Export > Export as WAV**. Once you select your save location, you can even add metadata to your file, just in case you need to find it on your computer later. Remember, the file won't save until after you hit **OK** on the metadata page.

Decimal Encoding

Now that we have a means to transmit a message, we need to make sure our message can conform to the transmission requirements. Since DTMF restricts us to ***1234567890*#***, we need to use a number-based encoder to translate complex characters into numbers.

Converting from ASCII to Decimal (or Charcode base10) is our best option. There are several websites that are capable of transcoding ASCII characters to Decimal. Here's a few:

- CyberChef - https://gchq.github.io
- DCode - https://www.dcode.fr/en
- Branah - https://www.branah.com/ascii-converter

Here's a perfect example of how complex characters can be turned into just numbers and spaces:

- ASCI: **HTB{rjamison}**
- Decimal or Charcode Base10: **72 84 66 123 114 106 97 109 105 115 111 110 125**

DTMF is unable to process spaces, but you can fix this by turning **spaces** into **asterisks** and **pounds**. You can do this using **Find and Replace** in any text editor.

Result: **72*84*66*123*114*106*97*109*105*115*111*110*125**
Paste this into the **DTMF generator** in **Audacity**, and you've created your obscured message.

WRITE-UP

DECODE A MESSAGE USING DTMF SIGNALS AND BASE64/DECIMAL

The E-Mail

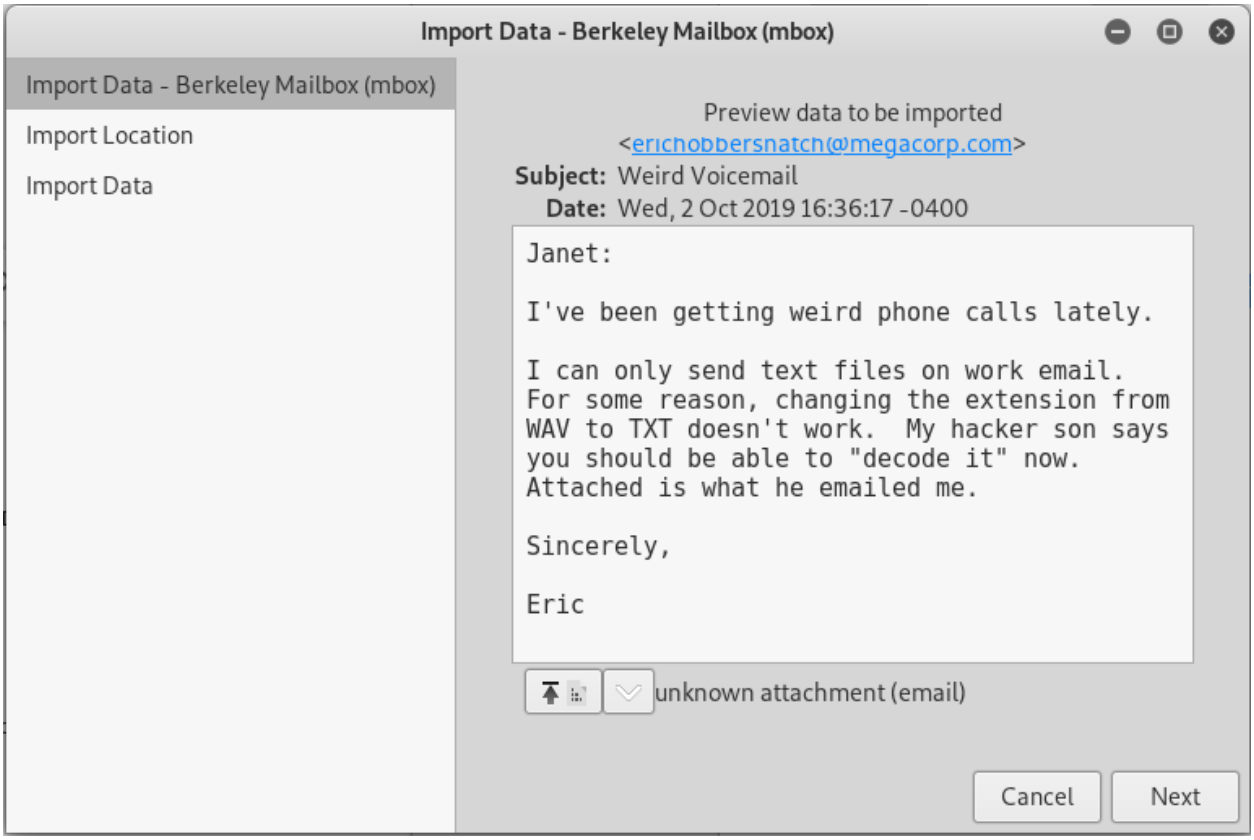
As soon as we open the zip, we find a file called “Weird Voicemail.eml”. EML files are essentially e-mail messages (text), so let’s see what the first few lines are.

```
rjamison@kali:~$  less “Weird Voicemail.eml”
MIME-Version: 1.0
Date: Wed, 2 Oct 2019 16:36:17 -0400
Message-ID: <19928393292304jskdfjl@megacorp.com>
Subject: Weird Voicemail
From: Eric Hobbersnatch <erichobbersnatch@megacorp.com>
To: Eric Hobbersnatch <erichobbersnatch@megacorp.com>
Content-Type: multipart/mixed;
boundary="000000000000c2453b0593f36b65"

--000000000000c2453b0593f36b65
Content-Type: multipart/alternative;
boundary="000000000000c245370593f36b63"

--000000000000c245370593f36b63
Content-Type: text/plain; charset="UTF-8"

Janet:

I've been getting weird phone calls lately.

I can only send text files on work email.  For some reason,
changing the extension from WAV to TXT doesn't work.  My hacker
son says you should be able to “decode it" now.  Attached is
what he emailed me.

Sincerely,

Eric.
```

It sounds like Eric is the victim of a telemarketer or a prankster. Additionally, it looks like simply **changing the file extension did not work**. Also of note, his hacker son likely **encoded** it somehow. Let’s check out the rest of the message.

```
--000000000000c2453b0593f36b65
Content-Type: application/octet-stream; name=email
Content-Disposition: attachment; filename=email
Content-Transfer-Encoding: base64
X-Attachment-Id: f_k19qdr8u0
Content-ID: <f_k19qdr8u0>

VWtsR1JzQU5SUUJYUVZaRlptMTBJQkFBQUFBQkFBSUFSS3dBQUJDeEFnQUVBQkFB
...
```

Let's extract this attachment. If you are in a GUI based operating system, you should be able to download the data directly using **Evolution** or another mail management application.



ALTERNATIVE: You can **create a copy** of the email and use **nano** to **remove everything but the base64 attachment data**.

```
rjamison@kali:~$  cp "Weird Voicemail.eml" attachment.txt
rjamison@kali:~$  nano attachment.txt
```

Base64 Decoding

It looks like the attachment is pretty lengthy (not shown), and the content was transferred using base64. Let's see what we get when we decode just the attachment data.

```
rjamison@kali:~$  cat attachment.txt | base64 -d > voicemail
```

Now we have the raw file dump from base64. Let's see if we can identify it.

```
rjamison@kali:~$  file voicemail
voicemail: RIFF (little-endian) data, WAVE audio, Microsoft PCM,
16 bit, stereo 44100 Hz
```

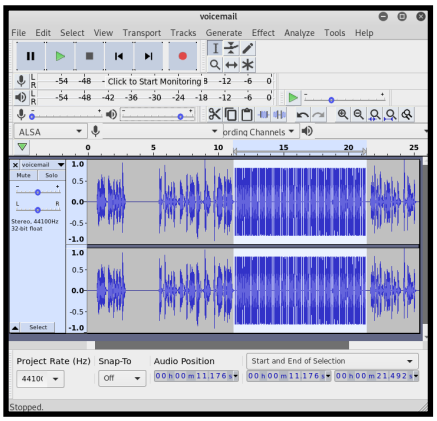
We've identified it as a WAV file! Let's add the extension for posterity.

```
rjamison@kali:~$  mv voicemail voicemail.wav
```

If you're in a GUI-based operating system, just double click and **listen to the audio**. It should sound like familiar voice mailbox playback followed by lots of beeps. You will need to install Audacity for the next steps.

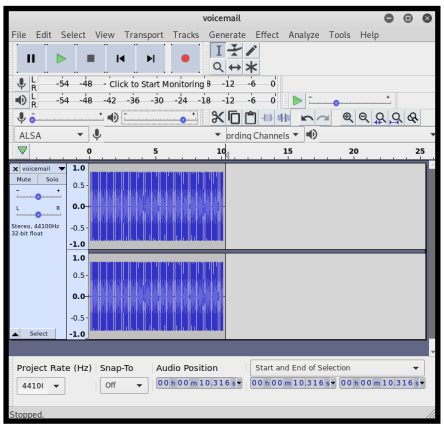
```
rjamison@kali:~$  sudo apt-get install audacity
rjamison@kali:~$  audacity
```

DTMF Signal Decoding



Now that we have the file open in Audacity, let's trim this down to just the DTMF signal portion.

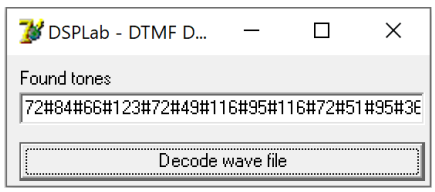
Use the cursor to select just the DTMF portion of the audio and click **Edit > Clip Boundaries > Split. Double click** the audio we aren't using and delete it.



Go to **File > Export > Export as WAV**. Call the new WAV file "dtmf.wav".

Next, we need to decode this WAV file. Windows users can download DSPLab from http://www.teworks.com/dtmf.htm to decode.

Linux and Mac users can go to http://dialabc.com/sound/detect/ for an online decoder.



As you can see, grabbing the values is as easy as copy/paste once you have a decoder:
72#84#66#123#72#49#116#95#116#72#51#95#36#48#117#78#68#95#87#64#118#51#36#125#10

Decimal Decoding

If you're using a GUI based operating system, you'll need to use a text editor to **Find and Replace** the hash symbols for spaces. Next, go to the decoder website of your choosing, and decode from decimal to ascii. http://dcode.fr/en is always the best choice.

Original:
72#84#66#123#72#49#116#95#116#72#51#95#36#48#117#78#68#95#87#64#118#51#36#125#10

Adjusted:
72 84 66 123 72 49 116 95 116 72 51 95 36 48 117 78 68 95 87 64 118 51 36 125 10

Decoded:
HTB{H1t_tH3_$0uND_W@v3$}