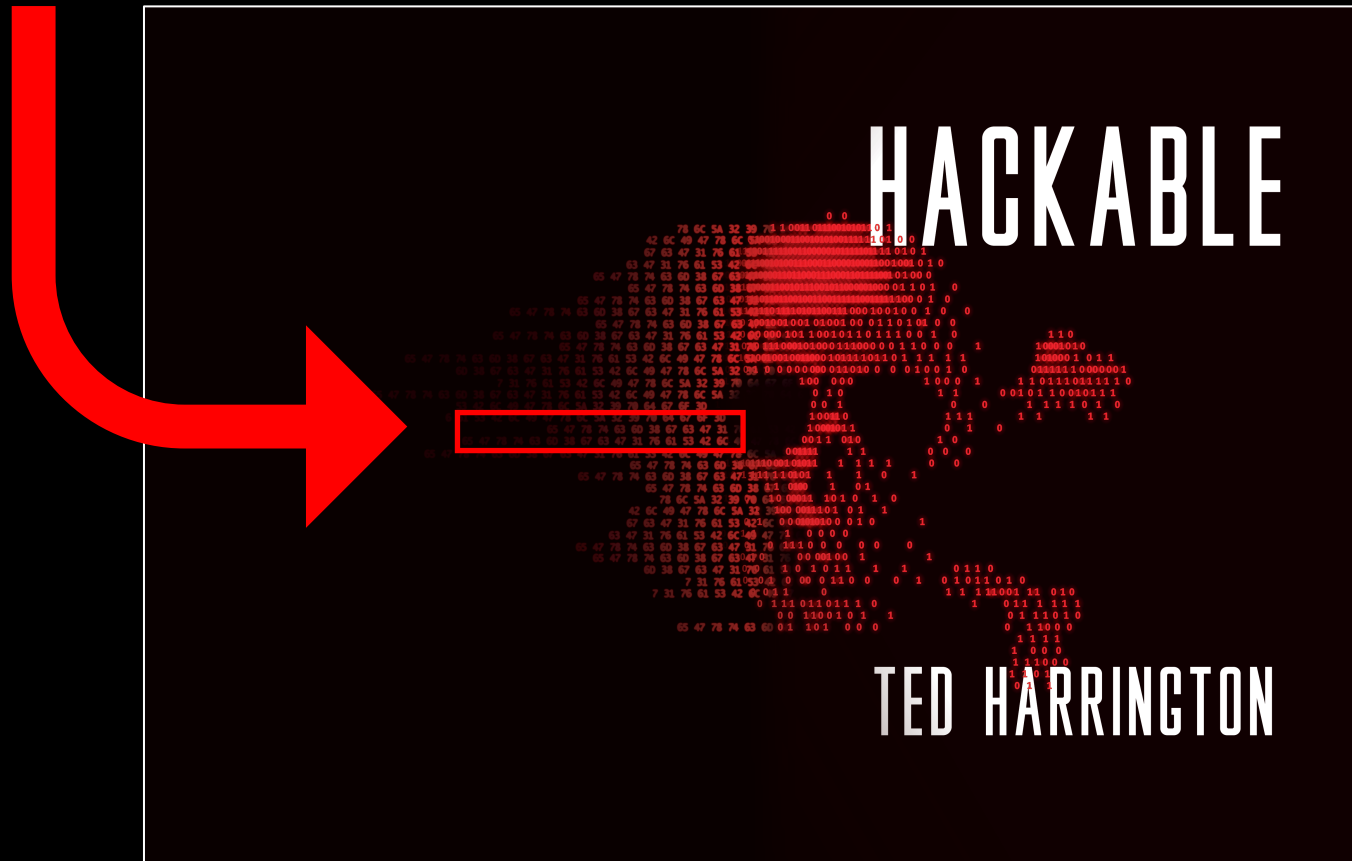


```
0 0
11001010110 1
10100111111 01 0 0
00101111101111 010 1
110000100011001001 0 1 0
110001101111001 0 1 0 0
1011000001000 01 1 0 1 0
001111110011111100 0 1 0
00111000100100 1 0 0 0
100100 01 10101 0 0
010110 11100 1 0
0011100000 110 0 0 1
010111101101 1 1 1 1
0011010 0 0 0100 1 0
000 1000 1 110111011111 0
1 0 1 1 001011001011 1
0 1 0 0 1 1 1 0 1 0
00110 1 1 1 1 1 1
001011 0 1 0
1 010 1 0
1 1 0 0 0
1 1 1 1 1 0 0
1 1 0 1
1 0 1
1 0 1 0
100 0 1 0 1
0 0 0 0 0
0100 1 1
0 1 1 1 0 1 0 1 1 0
0 1 1 0 0 0 1 0 1 0 1 0 1 0
0
0 1 1 1 0
0 0 1 0 1 1
1 0 0 0
1 1 1 0 0 1 1 0 1 0
0 1 1 1 0
0 0 1 0 1 1
1 0 0 0
1 1 1 1
1 0 0 0
1 1 1 0 0 0
1 1 0 1
0 1 1
```

# Easter Egg: Code Hidden in the Back Cover

Reversing walkthrough

# So, you found the easter egg.

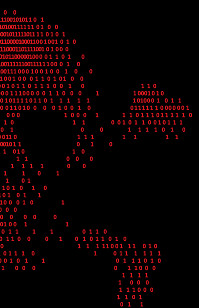


# Congratulations!

Now let's guide you through how to **decipher** it.

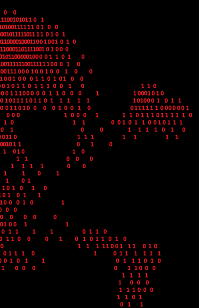
For ease of reference, here's the code we'll be working on:

65 47 78 74 63 6d 38 67 63 47 31 76 61 53 42 6c 49 47 78 6c 5a 32 39 70 64 67 6f 3d



# STEP 0: ESTABLISH YOUR GOAL

The objective is to decipher this code into something you can read. To do that, you'll need to reverse it.



# STEP 1: IDENTIFY THE ENCODING

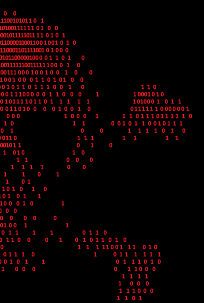
What do you notice about the code? A few things might jump out to you:

- It uses digits and letters
- The digit and letters are in pairs

What encoding is this?

(Try to reason through this first. When you're ready, advance to the next page for the answer.)

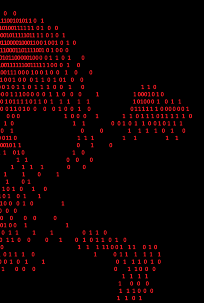
65 47 78 74 63 6d 38 67 63 47 31 76 61 53 42 6c 49 47 78 6c 5a 32 39 70 64 67 6f 3d



# STEP 1: IDENTIFY THE ENCODING

If you're a mathematics or computing nerd, you might recognize those as qualities of **hexadecimal**, a positional system that uses sixteen distinct symbols (usually "0"–"9" to represent values zero to nine, and "a"–"f" to represent values ten to fifteen).

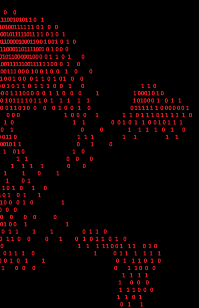
65 47 78 74 63 6d 38 67 63 47 31 76 61 53 42 6c 49 47 78 6c 5a 32 39 70 64 67 6f 3d



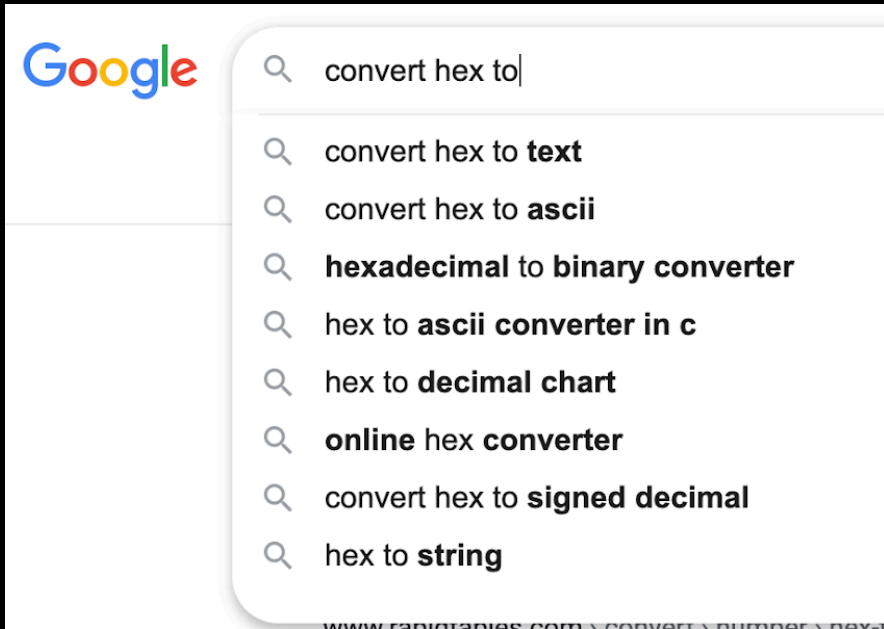
# STEP 2: CONVERT HEX TO...?

Although you've identified it as Hex, this alone isn't something you can read. So we'll need to convert it into something else.

How might you go about that?



# STEP 2: CONVERT HEX TO...?



Here's a tip: research things you can convert Hex into. A good place to start might be to hit up your friend Google, and start typing "convert Hex to..." and then try the different translators that come up.

If you want to try some of these out yourself, go for it! I'll wait. If you want a shortcut, advance to the next page.



# STEP 2: CONVERT HEX TO ASCII

That's right, you need to convert to ASCII. To convert Hex to ASCII, input the hex into a converter like this one: <https://www.rapidtables.com/convert/number/hex-to-ascii.html>

Hex to ASCII Text Converter

Enter hex bytes with any prefix / postfix / delimiter and press the *Convert* button  
(e.g. 45 78 61 6d 70 6C 65 21):

Paste hex numbers or drop file

```
65 47 78 74 63 6d 38 67 63 47 31 76 61 53 42 6c 49 47 78 6c 5a 32
39 70 64 67 6f 3d
```

Character encoding

ASCII

```
eGxtcm8gcG1vaSB1IGx1Z29pdgo=
```

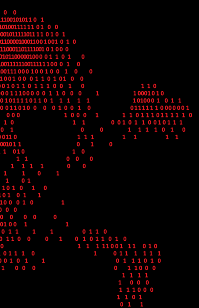
# STEP 2: CONVERT HEX TO ASCII

**Congratulations!**

You've successfully converted hexadecimal to ASCII!

Here's what you're working with now:

eGxtcm8gcG1vaSBllGxlZ29pdgo=

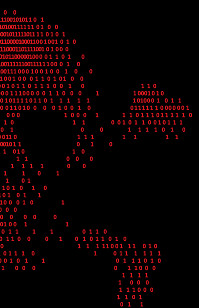


# STEP 3: DECODE THE MESSAGE

Ok, so you've made progress, but you still can't read it. You need to decode the message.

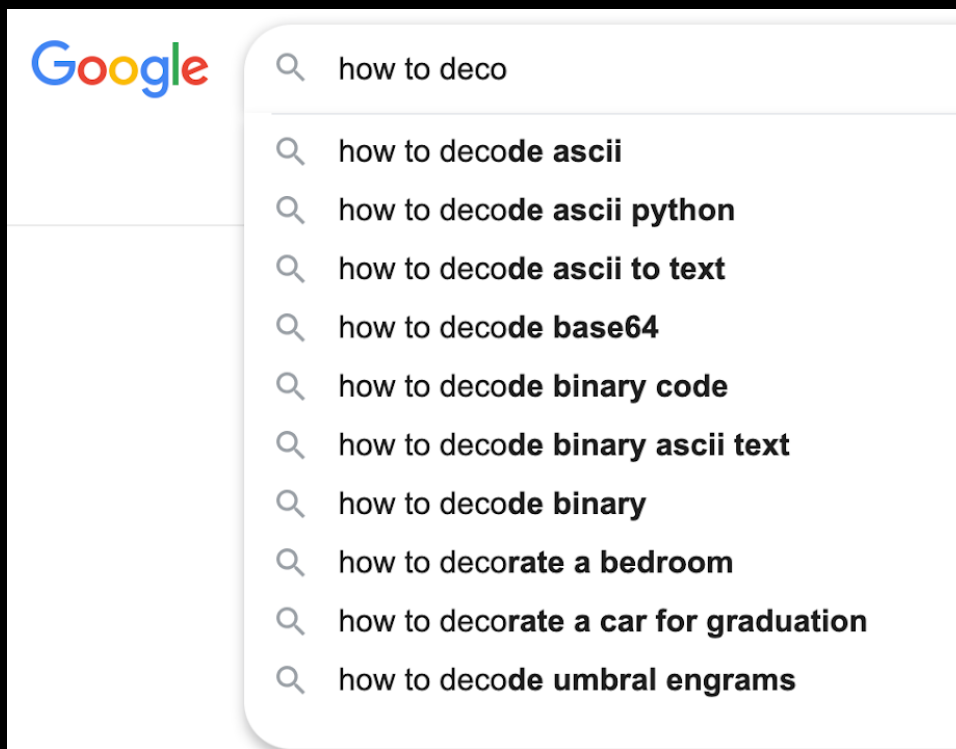
What should you do next?

**How would an attacker approach this?**



# STEP 3: DECODE THE MESSAGE

For this, you can turn to trusty friend again:



Which are you most likely to select first?

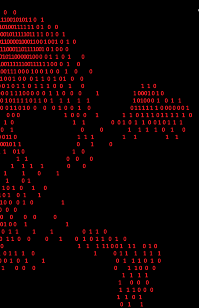
That's right, almost anyone is going to go for the **ASCII to text decoder**. Go ahead, give it a try! If you want a shortcut, advance to the next page.

# STEP 3: DECODE THE MESSAGE

Ah, ASCII-to-text didn't work did it?

You've just learned what it's like to reverse engineer: **you will constantly run into dead ends and red herrings.**

It would take most people a lot of trial and error (or being either lucky or experienced) to figure out what to convert it to, so in the interest of your time, I'll give you the shortcut: you need to decode from Base64 format.



# STEP 3: DECODE THE MESSAGE

Now that you've narrowed in on decoding from Base64, you need another converter. Again, let's go to Google! That's going to take you somewhere like this:

<https://www.base64decode.org/>

So, grab your output from the Hex to ASCII conversion, and use it to decode from Base64, like this:

eGxtcm8gcG1vaSBllGxlZ29pdgo=

The screenshot shows the 'BASE64 Decode and Encode' website. At the top, there are two tabs: 'Decode' (selected) and 'Encode'. Below the tabs, a green banner reads: 'Have to deal with Base64 format? Then this site is made for you! Use our super handy online tool to decode or encode your...'. The main section is titled 'Decode from Base64 format' and includes the instruction: 'Simply enter your data then push the decode button.' A large text input field contains the Base64 string 'eGxtcm8gcG1vaSBllGxlZ29pdgo='. Below the input field, there is an information icon and a note: 'For encoded binaries (like images, documents, etc.) use the file upload form a bit further down on this page.' There are two dropdown menus: 'Source character set' set to 'UTF-8' and 'Live mode' set to 'OFF'. A green button labeled '< DECODE >' is visible. At the bottom, the decoded output 'xlmro pmoi e legoiv' is displayed in a light gray box.

# STEP 3: DECODE THE MESSAGE

**Congratulations!**

You've successfully decoded Base64!

Here's what you're working with now:

```
xlmro pmoi e legoiv
```

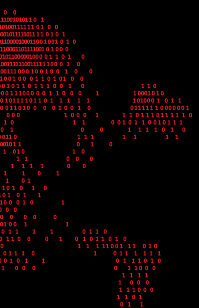


---

# STEP 4: DECIPHER

---

Ok, now we're getting somewhere! That output still isn't real words, but what do you notice about it?



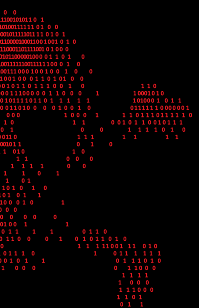


# STEP 4: DECIPHER

Kinda looks like it could be a sentence, doesn't it?

Reverse engineering is all about problem solving and trial & error. So go ahead and try different ways to decipher that sentence.

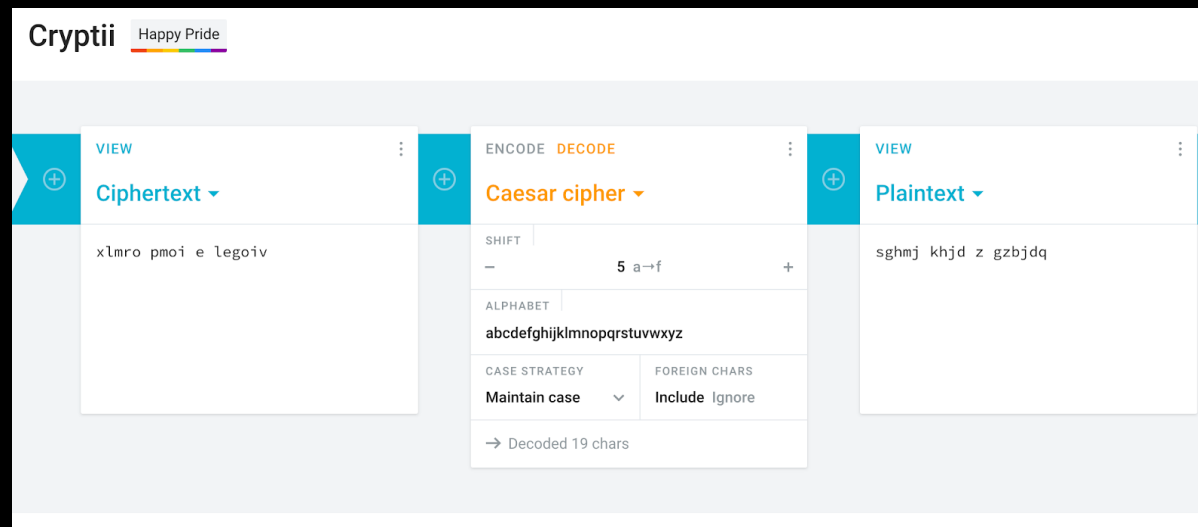
If you want a shortcut, jump ahead to the next page.



# STEP 4: DECIPHER

Good job! You've correctly guessed that this is probably encoded with a Caesar Cipher (also known as a shift cipher), one of the most common techniques in cryptography. To decipher using a Caesar Cipher, find a tool online to help you, like this one:

<https://cryptii.com/pipes/caesar-cipher-decoder>



The screenshot shows the Cryptii website interface for a Caesar cipher decoder. The page title is "Cryptii Happy Pride". The interface is divided into three main sections: "Ciphertext", "Caesar cipher", and "Plaintext".

- Ciphertext:** Contains the input text "xlmro pmoi e legoiv".
- Caesar cipher:** Contains the following settings:
  - SHIFT: 5 a→f
  - ALPHABET: abcdefghijklmnopqrstuvwxyz
  - CASE STRATEGY: Maintain case
  - FOREIGN CHARS: Include Ignore
  - Decoded 19 chars
- Plaintext:** Contains the output text "sghmj khjd z gzbjdg".

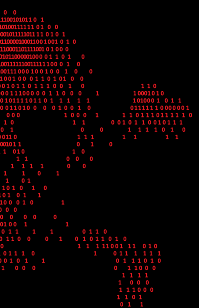
---

# STEP 4: DECIPHER

---

You're almost there! Now you just need to try different shifts to decode the message. The wrong shift returns unreadable text, but the correct shift returns something you can read!

**Play around with the shift until you can read the message in plaintext!**



0 0  
11001010110 1  
10100111111 01 0 0  
00101111101111 01 0 1  
110000100011001001 0 1 0  
110001101111001 0 1 0 0  
01011000001000 01 1 0 1 0  
001111110011111100 0 1 0  
00111000100100 1 0 0 0  
100100 01 10101 0 0  
010110 11100 1 0  
0011100000 1 10 0 0 1  
010111101101 1 1 1 1  
0011010 0 0 0100 1 0  
000 1000 1 110111011111 0  
1 0 1 1 00101 10010111  
0 1 0 0 1 1 1 1 0 1 0  
00110 1 1 1 1 1 1  
001011 0 1 0  
1 010 1 0  
1 1 0 0 0  
1 1 1 1 1  
1 1 0 1  
1 01  
101 0 1 0  
101 01 1  
100 01 0 1  
0 0  
0 0 0 0 0  
0100 1 1  
0 1 1 1 1 0 1 1 0  
0 1 1 0 0 0 1 0 1 0 1 0 1 0  
0 1 1 1 1 1 0 1 1 0 1 0  
0 1 1 1 1 0 1 1 1 1 1 1  
0 0 1 0 1 1 0 1 1 0 1 0  
1 0 0 0 0 1 1 0 0 0  
1 1 1 1  
1 0 0 0  
1 1 1 0 0 0  
1 1 0 1  
0 1 1

Thank you for reading  
*Hackable*, and please contact  
me if you need help with security  
assessments, speaking  
engagements, or really any  
security challenge you're facing!

Happy Hacking!  
  
Much love,  
Ted



[ted@tedharrington.com](mailto:ted@tedharrington.com)