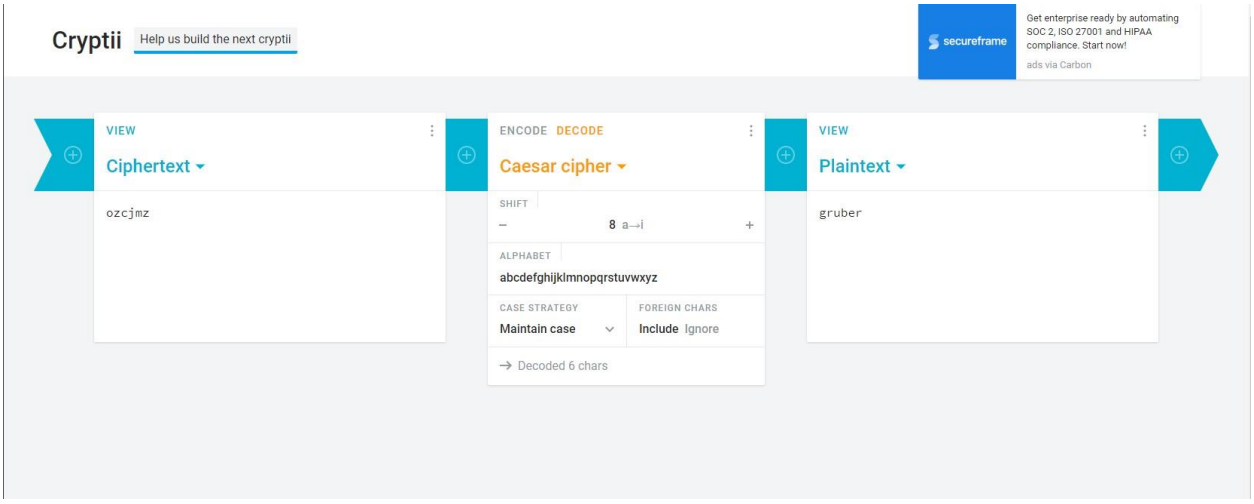
Rayshaun McIntosh

Homework Assignment 10

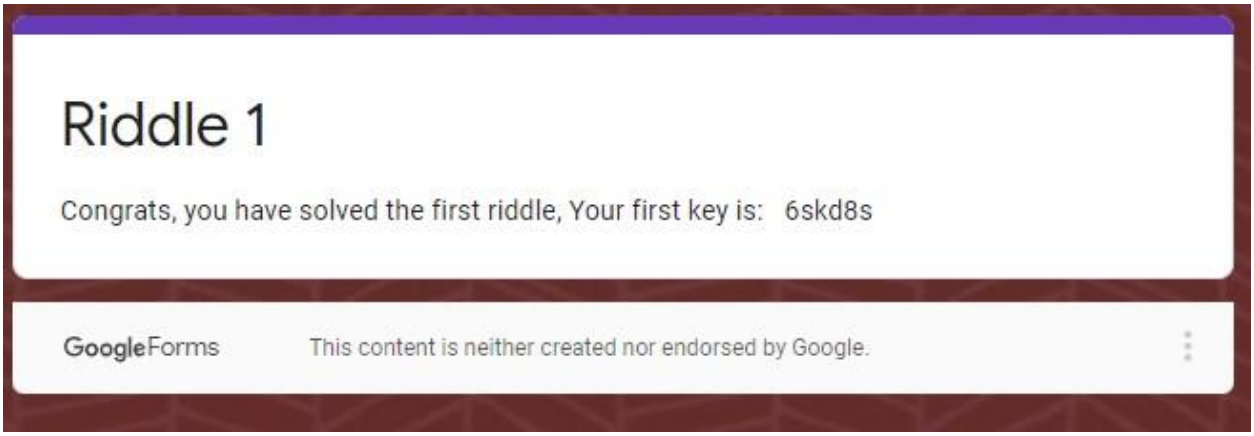
## Riddle 1

Roses are red violets are blue, Caesar would be 8 is your first clue. Decrypt ozcjmz and enter it below, and maybe a key then might just show.

After reading the first riddle, I went to https://cryptii.com/pipes/caesar-cipher and used the decode option and shifted the letters to 8 spaces. Doing that gave me the plaintext answer of gruber.




Once I entered the plaintext code into the site, I was greeted with this message: 6skd8s



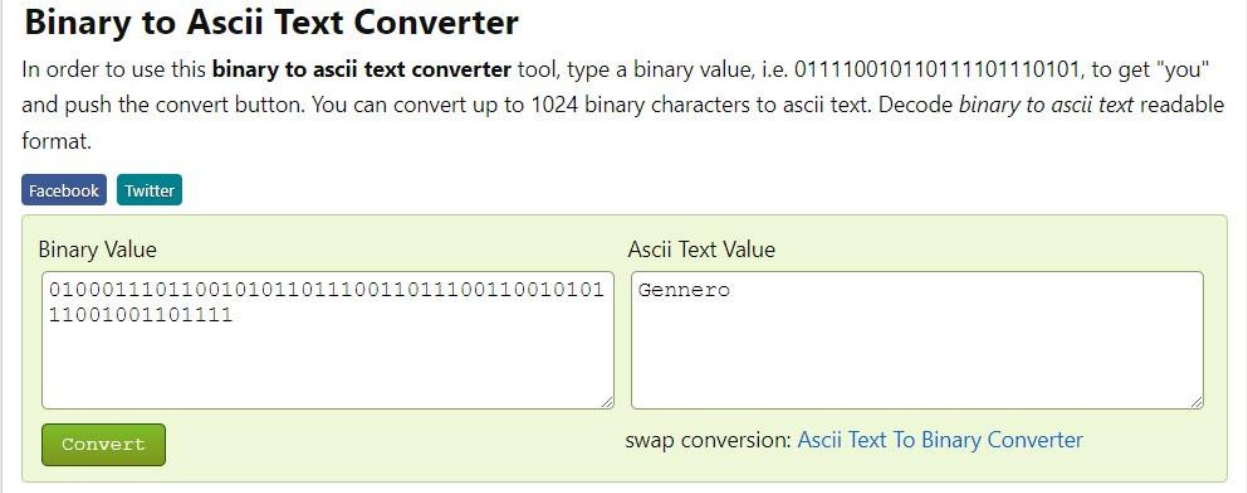

## Riddle 2

Humpty Dumpty sat on a wall, humpty dumpty had a great fall.

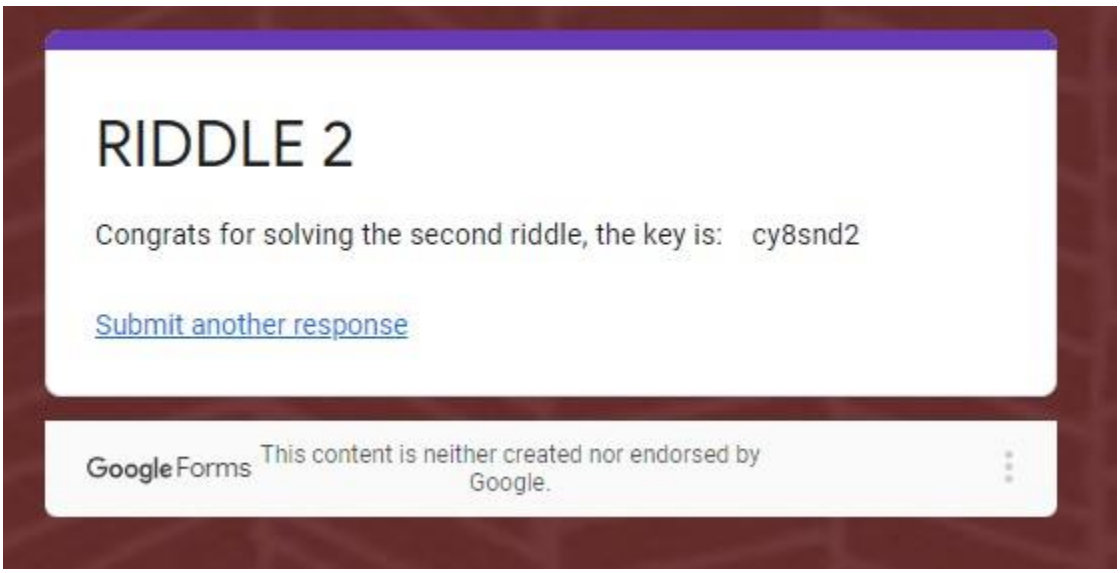
All the kings horses and all the kings men, couldn't decode this message for him:

01000111 01100101 01101110 01101110 01100101 01110010 01101111

I was given a binary code as the second riddle. I used https://www.binaryhexconverter.com/binary-to-ascii-text-converter to decode the binary to plaintext. When I put the binary code in, it translated to this: Gennero




Once I entered the plaintext as the passcode I was greeted with: cy8snd2




Riddle 3

A little cypher text, short and sweet: 4qMOIvwEGXzvkMvRE2bNbg==

Here is my vector, and also my key:
5284A3B154D99487D9D8D8508461A478C7BEB67081A64AD9A15147906E8E8564

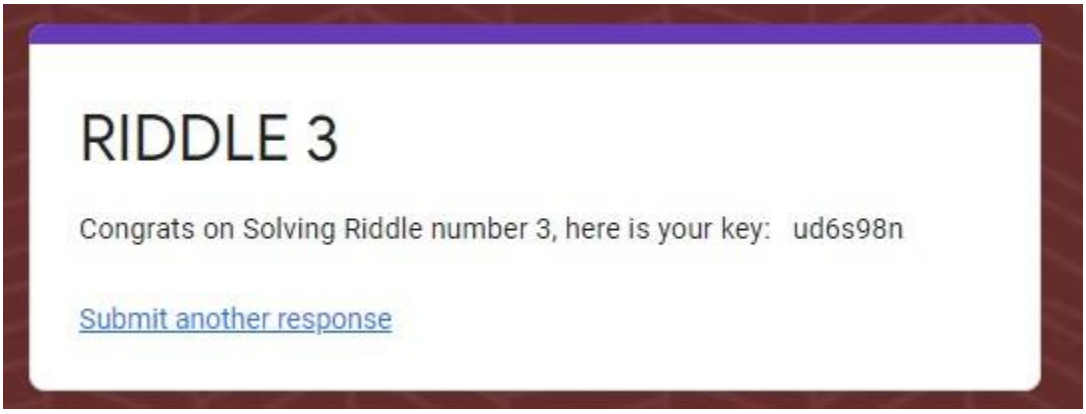
IV: 1907C5E255F7FC9A6B47B0E789847AED

Just OpenSSL to figure me out

Riddle 3 was unique. I had to do a lot of research to find out what would be the right direction to figure this out. After coming across this site https://askubuntu.com/questions/178521/how-can-i-decode-a-base64-string-from-the-command-line, it help me understand that I could use echo and pipe the message into openssl syntax, so I did using this command [echo "4qMOIvwEGXzvkMvRE2bNbg==" | openssl enc -pbkdf2 -nosalt -aes-256-cbc -d -base64 -K 5284A3B154D99487D9D8D8508461A478C7BEB67081A64AD9A15147906E8E8564 -iv 1907C5E255F7FC9A6B47B0E789847AED]. After that, I was able to decode the cipher text and found this: takagi

```
sysadmin@UbuntuDesktop:~$ echo "4qMOIvwEGXzvkMvRE2bNbg==" | openssl enc -pbkdf2 -nosalt -aes-256-cbc
-d -base64 -K 5284A3B154D99487D9D8D8508461A478C7BEB67081A64AD9A15147906E8E8564 -iv 1907C5E255F7FC9A6B
47B0E789847AED
takagi
```

Once I entered the decoded text, I was greeted with the 3rd key: ud6s98n




Riddle 4

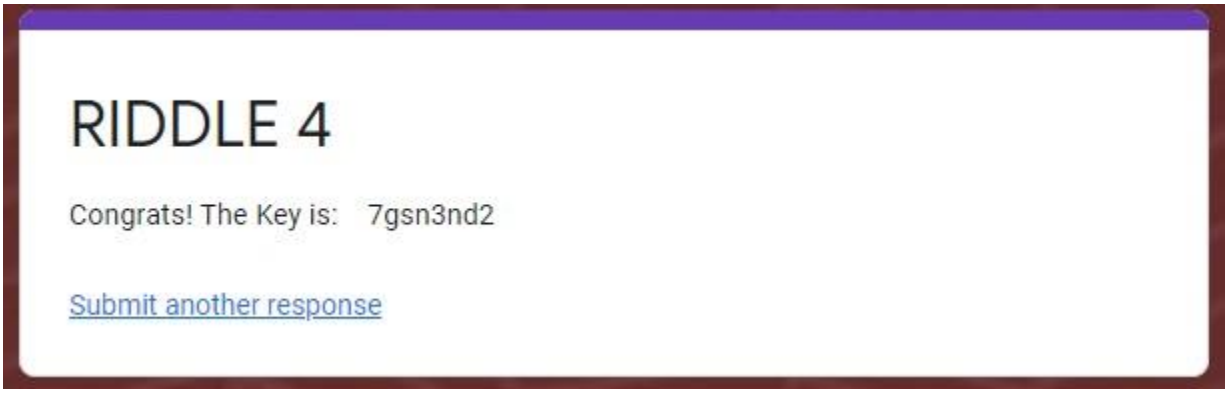
Jack and Jill went up the hill to use their public keys.

Jack had 2, and Jill did too to exchange their messages with ease.

What would Jack use to send an encrypted message to Jill?

Riddle 4 is a series of questions; the first question was What would Jack use to send an encrypted message to Jill? My answer was Jill's Public Key. The second question;

What would Jill use to decrypt Jack's message? Jill's Private Key. The third question; Jack and Jill invited Bob, Alice, Tim, and Peter along to exchange some messages. How many keys would they all need for asymmetric vs symmetric encryption? The answer for symmentric is (6 x 5) /2 = 15. 6 is the number of people and the 5 is number of people minus 1 divided by 2 is the equation. For Asymmetric is simple, (6 x 2) = 12. You multiply the number of people by 2. The fourth question; Tim just sent an encrypted message to one of his friends, which of the following keys did he likely use to encrypt the message? The answer is Alice's Public Key. The reason it was Alice was she was the only option that had a public key. For Tim to send a message, he would need her public key to do so for the message to be encrypted. After answering the final question, I was greeted with: 7gsn3nd2




Riddle 5

Hey diddle diddle, the cat and the fiddle, the cow jumped over the moon.

The little dog laughed when it found this MD5 hash, and the dish ran away with the spoon.

Hash: 3b75cdd826a16f5bba0076690f644dc7

For riddle 5 we were given an MD5 hash that needs to be decrypted. First thing I did was echo the hash into a file [echo "3b75cdd826a16f5bba0076690f644dc7" > riddle5] so I would be able to decrypt the hash.

```
sysadmin@UbuntuDesktop:~$ echo "3b75cdd826a16f5bba0076690f644dc7" > riddle5
```

After doing that I decided to use hashcat and use the wordlist rockyou.txt to see if I could decrypt the hash. I wanted a new file for the output, so I chose riddle5solved.txt.

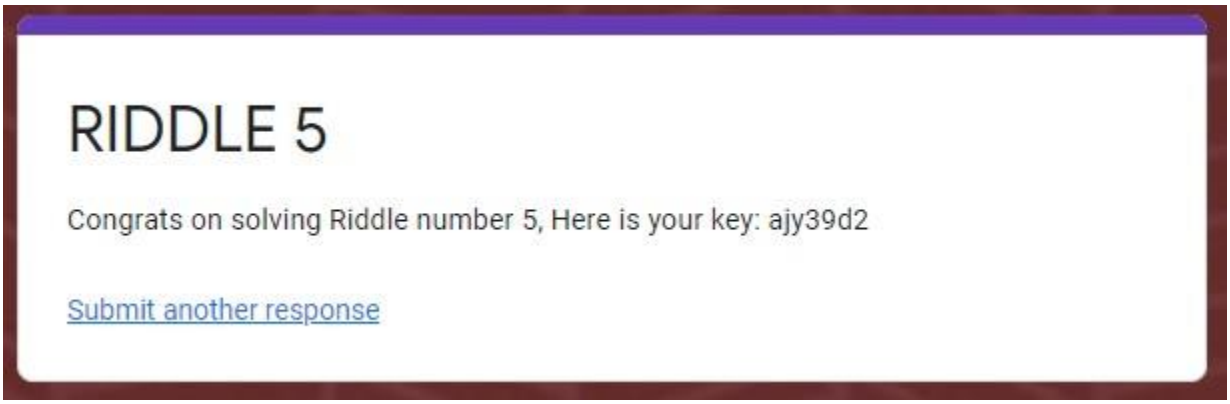
[hashcat -m 0 -a 0 -o riddle5solved.txt riddle5 /usr/share/wordlists/rockyou.txt --force]

```
sysadmin@UbuntuDesktop:~$ hashcat -m 0 -a 0 -o riddle5solved.txt riddle5 /usr/share/wordlists/rockyou
.txt --force
hashcat (v4.0.1) starting...

OpenCL Platform #1: The pocl project
====================================
* Device #1: pthread-Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, 1024/2956 MB allocatable, 2MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
```

After running this command, the hash was decrypted. I cat the new file I created so I would be able to see the results [cat riddle5solved.txt]. The plaintext was argyle.

```
sysadmin@UbuntuDesktop:~$ cat riddle5solved.txt
3b75cdd826a16f5bba0076690f644dc7:argyle
```

I use the plaintext and submit for riddle 5 and was greeted with this key: ajy39d2




Riddle 6

Mary has a secret code, hidden in a photo, and everywhere that photo went, the code was sure to go. She wrote the passphrase on the book, to access the code you just need to use some stego tricks and the secret will be showed.

For the last riddle we had to download a photo and extract the message hidden in the photo. Once I downloaded the photo I moved it from my vagrant folder to my home directory by using [mv /vagrant/mary-lamb.jpg ./].

```
sysadmin@UbuntuDesktop:~$ mv /vagrant/mary-lamb.jpg ./
```

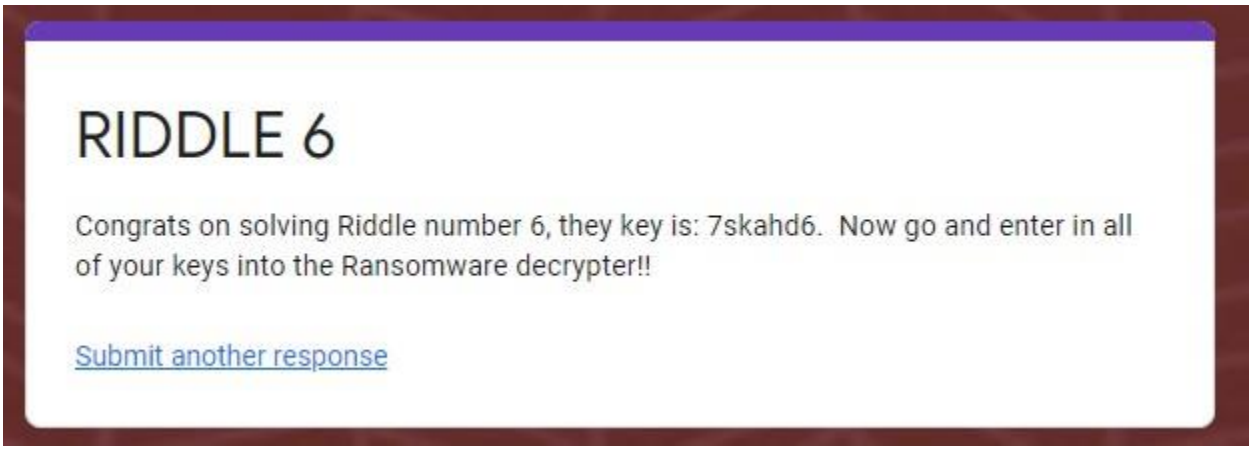
Now that the photo is in my home directory I used steghide to extract the message within the photo[steghide extract -sf mary-lamb.jpg]. After that a file was downloaded to my directory "code_is_inside_this_file.txt".

```
sysadmin@UbuntuDesktop:~$ steghide extract -sf mary-lamb.jpg
Enter passphrase:
wrote extracted data to "code_is_inside_this_file.txt".
```

Now that we have extracted the data we can take a look at the file. To see the contents of the file I used [cat code_is_inside_this_file.txt].

```
sysadmin@UbuntuDesktop:~$ cat code_is_inside_this_file.txt
mcclane
```

Doing that I found the password for the final riddle to get the last key, which was mcclane. Once I entered this password the final key was revealed to me. 7skahd6




After entering all the keys into the Ransomware Decrypter I was greeted with this below in the screenshot.


RANSOMWARE DECRYPTER
Congratulations! You have decrypted the Ransomware! All the Nakatomi Hospital Records are now Decrypted! Please take a screenshot of this message and submit as your homework!
Submit another response