Question 1

In this project you will experiment with a *padding oracle attack* against a toy web site hosted at crypto-class.appspot.com. Padding oracle vulnerabilities affect a wide variety of products, including secure tokens. This project will show how they can be exploited. We discussed CBC padding oracle attacks in Lecture 7.6, but if you want to read more about them, please seeVaudenay's paper.

Now to business. Suppose an attacker wishes to steal secret information from our target web site crypto-class.appspot.com. The attacker suspects that the web site embeds encrypted customer data in URL parameters such as this:

```
http://crypto-class.appspot.com/po?er=f20bdba6ff29eed7b046d1df9fb7000058b1ffb
4210a580f748b4ac714c001bd4a61044426fb515dad3f21f18aa577c0bdf302936266926ff37d
bf7035d5eeb4
```

That is, when customer Alice interacts with the site, the site embeds a URL like this in web pages it sends to Alice. The attacker intercepts the URL listed above and guesses that the ciphertext following the "`po?er=`" is a hex encoded AES CBC encryption with a random IV of some secret data about Alice's session.

After some experimentation the attacker discovers that the web site is vulnerable to a CBC padding oracle attack. In particular, when a decrypted CBC ciphertext ends in an invalid pad the web server returns a 403 error code (forbidden request). When the CBC padding is valid, but the message is malformed, the web server returns a 404 error code (URL not found).

Armed with this information your goal is to decrypt the ciphertext listed above. To do so you can send arbitrary HTTP requests to the web site of the form

```
http://crypto-class.appspot.com/po?er="your ciphertext here"
```

and observe the resulting error code. The padding oracle will let you decrypt the given ciphertext one byte at a time. To decrypt a single byte you will need to send up to 256

HTTP requests to the site. Keep in mind that the first ciphertext block is the random IV. The decrypted message is ASCII encoded.

To get you started here is a short Python script that sends a ciphertext supplied on the command line to the site and prints the resulting error code. You can extend this script (or write one from scratch) to implement the padding oracle attack. Once you decrypt the given ciphertext, please enter the decrypted message in the box below.

This project shows that when using encryption you must prevent padding oracle attacks by either using encrypt-then-MAC as in EAX or GCM, or if you must use MAC-then-encrypt then ensure that the site treats padding errors the same way it treats MAC errors.

Answer:

```
The Magic Words are Squeamish Ossifrage
```