ECE435: Network Engineering – Homework 3

Encryption

Due: Thursday, 8 February 2024, 12:30pm

For this homework short answers will suffice.

To submit, create a document with your answers (text, pdf, libreoffice, MS Office if you must) and e-mail them to *vincent.weaver@maine.edu* by the homework deadline. Title your e-mail "ECE435 Homework 3" and be sure your name is included in the document.

1. **Cryptographic Hash Functions**

   (a) **md5sum/sha256 (3pts)**

   i. Download the file `hw3_test.txt` from the website:
   `https://web.eece.maine.edu/~vweaver/classes/ece435/hw3_test.txt`
   and calculate the md5sum.
   On Linux you can run something like `md5sum test.txt`
   If you aren't running Linux, you can try using a website for this,
   `https://emn178.github.io/online-tools/md5.html` might work.

   Report the md5sum that you get.

   ii. Make a copy of the file, and then make a small change (for example change the homework number). Re-run the md5sum.

   Report the resulting md5sum. How does the result compare to the unmodified file?

   iii. Also generate the SHA-256 sum for the original `hw3_test.txt` file. (SHA-256 is the 256-bit variant of SHA-2). On Linux you can use the `sha256sum` program for this.

   Report the resulting sha256 sum. How is it different from the md5sum?

2. **PGP/GPG (5pts)**

   On Linux use the `gpg` program for these tasks (if not installed, you can install it, something like `apt-get install gpg` or equivalent). You can also download GPG software for Windows/OSX from `https://gnupg.org/download/`.

   (a) Validating Signature

   i. The file `hw3_test.txt.signed` is a file that has been PGP/GPG signed by me.
   Verify that it was actually me that signed it.
   First download the signed file:
   `http://web.eece.maine.edu/~vweaver/classes/ece435/hw3_test.txt.signed`

   Then download my public key:
   `http://web.eece.maine.edu/~vweaver/classes/ece435/weaver.public_key`

   You will have to add this key to your keystore:

```
gpg --import weaver.public_key
```

Validate the `hw3_test.txt.signed` file:

```
gpg --verify ./hw3_test.txt.signed
```

Was it signed by me?

Now change something in the `hw3_test.txt.signed` file.
Reverify. **Does it still pass?**

ii. You have validated the document using the public key I linked to, but how can you know it was really *me* who signed things and not an imposter?
GPG might have complained about this.

Describe one technique used to authenticate that a public key belongs to who it says it does.

(b) Encrypt a message using gpg and using my public key.

You can use the public key you imported earlier.

Create a text file `secret_message.txt` with your message.

Then run something like this:

```
gpg --output secret_message.gpg --encrypt \
--recipient vincent.weaver@maine.edu secret_message.txt
```

Attach this `secret_message.gpg` when submitting your assignment.

3. **HTTPS and Certificate Authorities (1pt)**

   (a) Connect a web browser to `https://umaine.edu`

   (b) **What certificate authority is used by this site? Can you view the certificate? What type of hash was used for signing things?**

   (c) Hint: on most desktop browsers you can find this info by clicking on the padlock icon next to the URL and the clicking on a few menu items.

4. **Short Answer Question (1pt)**

   (a) The git SCM tool used to use SHA-1 to uniquely identify files. They are now transitioning to using SHA-256 instead. Why?

5. **Extra Credit (optional)**

   (a) If you are looking for an extra challenge, see if you can create a file that has the same md5sum as the hw3_test.txt file. If you are able, attach it to your submission (assuming it's less than a few megabytes in size). (Note: it might not be possible to do this in a reasonable amount of time)