

Introduction to Computer Security

Lab 3 - Exploring Identity & Authentication

Report
of
Andreas Wilhelm

February 10, 2019

You can find the code and everything else also on GitHub under the link <https://github.com/awilsee/CSec>. Maybe more convenient for you.

1 Exploring Certificate Authority Public Key Infrastructure (CA-PKI) with OpenSSL

1.1 A

Beside your descriptions I also have to change the following two options to get the web server in chapter 2 running. Additionally I have generated the ca-key with 2048 bits.

```
1 default_bits          = 1024
2 CipherString = DEFAULT@SECLEVEL=1
```

1.2 B

What do you observe? What fields are there?

It shows the data of the certificate with all the inputs from before. Additionally you can see that *sha256WithRSAEncryption* is used for encryption. Further more the RSA public key is included.

1.3 C

Open the key file your just created with a text editor and note your observations.

You can see the 3DES encrypted private key, modulus, the private/public exponent, both primes and exponents. All encrypted with your previously choosen keyphrase.

What do you observe in this certificate? Try to connect your observations to the earlier outputs. For example, where did the the serial number of the certificate come from?

The certificate includes the modulus and exponent of the certificate which is needed for the RSA decryption. Additionally it contains also the *X509v3 Authority Key Identifier* which is the identification from my created CA and its corresponding key. So it's now signed from the CA.

What capabilities have you just given an entity by signing their public key? What prevents anyone from impersonating Google or other wellknown service in this way?

I give the entity now the possibility to use their certificate which is signed and therefore trustful which means it can noe basically used by him. The user wouldn't recognize that this CA wasn't real, because he doesn't check if all the certificates are from a trustful certification. The reason for that is that all not anybody can create a CA and sign certificates. Only certain CA can sign them and browser only accept this ones, otherwise they block the content or at least warn the user.

2 Using CA PKI to Secure and Authenticate Web Sites

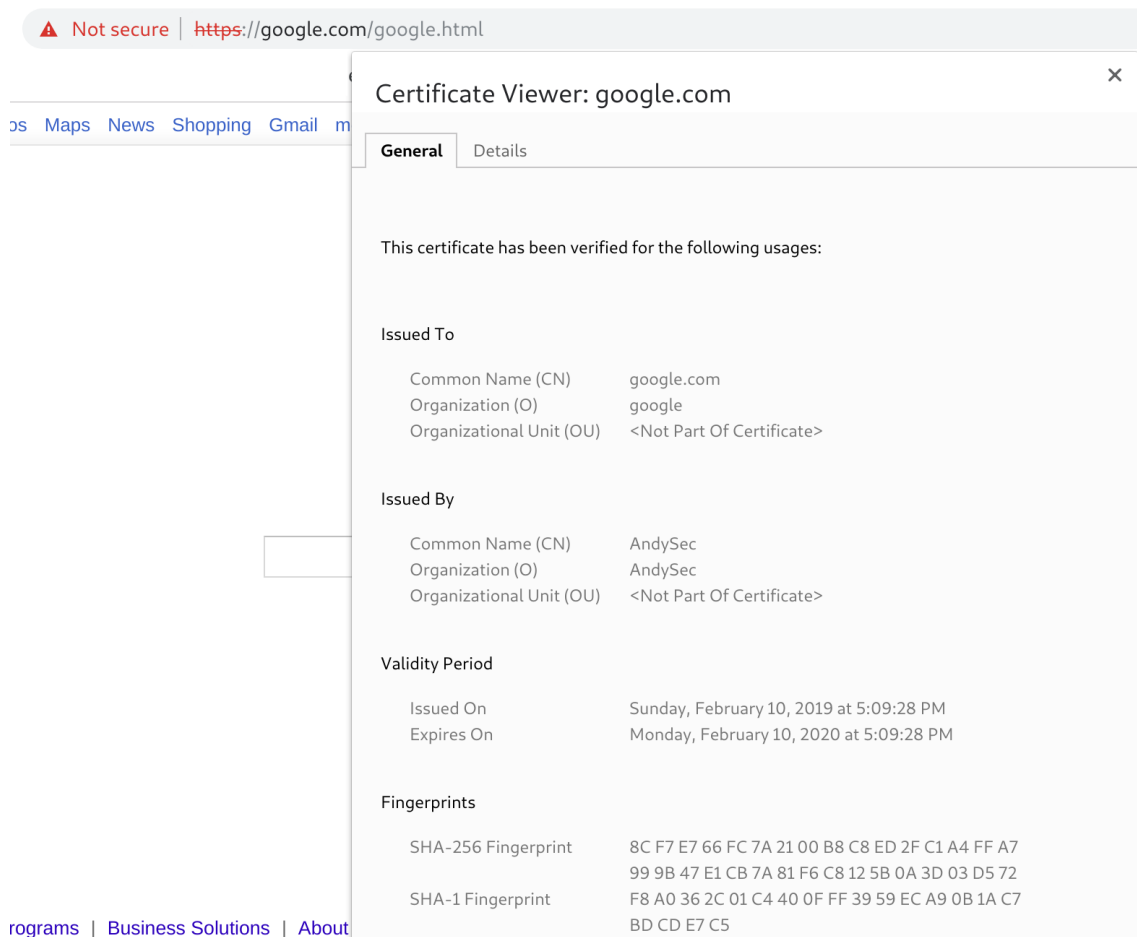
Can you verify this?

Yes, I can confirm that the browser doesn't accept the certificate. Though in chrome you can go to advanced and manually trust this certificate. In firefox you have to import the CA-key at first as described.

Is there any way for the user or the web browser to know you're not at the real google.com

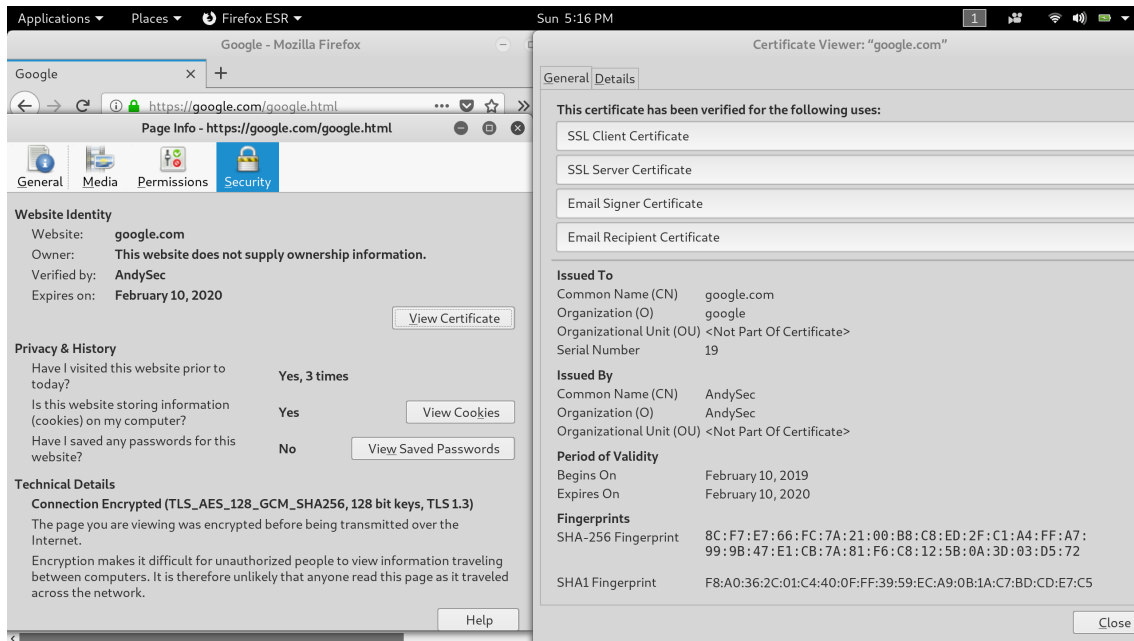
In Chrome you get still warned that this site is insecure, as you can see in figure 1. If you click on the details, you can see for whom the certificate is, that it was issued by the CA-organisation, the validity and the fingerprints.

Figure 1: Screenshot of Chrome



After you imported the CA to the trustful CAs. You don't get any warning and a green locked key lock sign, as you can see in figure 2. So for the first view anybody can't see any difference compared to the original site except that the faked one is out of date.. Only if you click to see the details you see the same detailed informations about it as in chrome respectively described before.

Figure 2: Screenshot of Firefox



3 Password Files

What do you observe?

What is the hash of your password? What is the date of your last password change?

What algorithm was used to protect your password? What is the value of the salt used to protect your password?

4 Password Cracking

4.1 A

By just looking at the hashes, what can you learn about an individual user's passwords? Is there anything you can learn about the passwords of all the users from this password file?

4.2 B

If your program completes (i.e. hashes all the words in the dictionary) without finding a match, what can you assume about the passwords? Which passwords were you able to crack?

4.3 C

Which additional passwords did you crack?

Were you able to crack all the passwords? Why or why not? If not, what characteristics do you think the un-cracked password has?

For those passwords you were unable to crack, what would be required to crack them and how do you think it would take?

Using each of the cracked passwords, go to the user's respective web page. What is the secret word for each user?