Cryptography

Willard Rafnsson

IT University of Copenhagen

In this assignment, we put ourselves in the shoes of cryptographic engineers. The goal is to gain experience with the uses of cryptography in practise. We'll use standard tools, and finish securing PayBud by adding SSL/TLS, hashing, and secure password storage.

Problem 1 : Cryptosystems

The gpg tool is the OpenPGP part of GNU Privacy Guard, which implements the PGP cryptosystem. Consult its manual (`man gpg`), or online tutorials, for the following:

Part 1 Do the gpg tasks listed under Assignment 4 in LearnIT. Include `symmetric.txt`, `mysymmetric.txt.gpg`, `mypublic.txt.gpg`, and your public key, in your submission.

The `openssl` tool is a cryptographic toolkit that implements the SSL/TLS cryptosystem.

Part 2 Create a public/private key pair (pardon the squished text; it's a long command):

```
openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout paybud.key -out paybud.crt -passout stdin
```

Type "`password`" at the silent password prompt, and hit ENTER. Bundle these keys into a key container format for PayBud. Type "`password`" as the password.

```
openssl pkcs12 -export -in paybud.crt -inkey paybud.key -out paybud.p12 -name paybud -caname root
```

PGP and SSL/TLS rely on fundamentally different trust models.

Part 3 (PGP) Go to `https://keyserver.ubuntu.com/`, and look up a friend of mine there, `freysteinn@freysteinn.com`. Click the top result. Explain what you see: Who vouches for whom? What does `[selfsig]` mean? What does `revok` mean? Explain Web of Trust in the context of this example. (~4 sentences)

Part 4 (SSL/TLS) Run the following command.

```
echo | openssl s_client -connect www.itu.dk:443 | openssl x509 -text
```

Explain what you see: What does depth 0, 1, and 2 mean? What does "Root CA" mean? What is that Hex-encoded "Modulus"? Is ITU a certificate authority? Explain Chain of Trust in the context of this example. (~6 sentences) Now run

```
openssl x509 -text -in paybud.crt
```

Who vouches for this key? Is this a root CA? (~2 sentences)

Not all of the protocols in the SSL/TLS cryptosystem are secure.

Part 5 (rating) Go to `https://www.ssllabs.com/ssltest/`. Type `web.whatsapp.com`. What are the stated reasons that the scanned server is capped at a B-rating? Are those reasons justified? (What kind of risk is the user at?) (~5 sentences)

Problem 2 : SSL/TLS

Assignment 4 ships with an updated version of PayBud; it now uses the TLS cryptosystem. Import the changes you made to A3-PayBud, into this A4-PayBud, by following the instructions given in the A4 section of the course page.

Part 1 Write the two-factor authentication protocol that PayBud now implements, in protocol design syntax. Let U, C and S denote the user, client, and server, respectively. Explain (in English) each step of the protocol. (~5 sentences)
Why does this stop the man-in-the-middle attack from Assignment 2? (~1 sentence)

Note: if you did not implement two-factor authentication in A3, then write the authentication protocol without a second factor of authentication.

Note: instead of including the TLS handshake at the start of your protocol, please assume that a handshake has already been made, and that the browser & server have arrived at a shared secret K. as a shorthand, you can write $\{m\}_K$ for message m encrypted with key K.

Place your `paybud.p12` into the PayBud root directory. Start the server, and browse to `https://localhost:5000` (note `https`). Your browser will issue a warning; in Chrome: you'll see "Your connection is not private". That's your browser doing its job.

Part 2 Why do you see this window? Why should you worry about this in general, but be okay with this in our scenario? (~2 sentences)

Hint: `https://letsencrypt.org/docs/certificates-for-localhost/`

Bypass the warning (Chrome: *Advanced* → *Proceed to localhost*). Once PayBud is running, download `https://github.com/drwetter/testssl.sh`, and run it on PayBud:

```
[path/to/]testssl.sh https://localhost:5000
```

Part 3 `testssl.sh` detected two known vulnerabilities in PayBud. Name and describe each of the two vulnerabilities. (~2 sentences)

Hint: Look for `VULNERABLE` under `Testing vulnerabilities`, disregarding the DoS vulnerability.

Bonus: What other weaknesses does `testssl.sh` reveal in PayBud?

Part 4 Fix those two vulnerabilities. Show (with `testssl.sh`) that your fix was successful. Explain how you modified the code (which methods did you add/modify & how?).

Hint: Configure the Java Secure Sockets Extension[a] (JSSE) by updating the appropriate properties in `java.security.Security` or `java.lang.System`, by invoking the `setProperty` method on these classes.

[a] `https://docs.oracle.com/javase/8/docs/technotes/guides/security/jsse/JSSERefGuide.html#InstallationAndCustomization`

Problem 3 : Hashing

Sally: "This thing is still being hacked? How?!"
Alice: "Can't be the connection; we secured that... how are we using this connection?"

This assignment ships with a video demonstration of yet another hack on PayBud.

Part 1 What vulnerability (in PayBud) is this hack exploiting? (~1 sentence)

Part 2 Show how the logging you implemented in Assignment 3 can reveal that this vulnerability is being exploited. (~3 sentences)

Part 3 Fix this vulnerability, by protecting the integrity of the session cookie. Explain how you modified the code (which methods did you add/modify & how?). (~3 sentences)

Hint: `https://stackoverflow.com/a/44110982` and `java.util.Base64.getUrlEncoder`. The cookie is formatted correctly; put the hash in it, and use it to verify integrity upon receiving requests. First use a hard-coded string constant as key. Once that works, generate & store a key in `paybud.p12`, and `KeyStore.getKey`.
`keytool -genseckey -keystore paybud.p12 -storetype pkcs12 -keyalg HMacSHA512 -keysize 2048 -alias HMACkey -keypass password`

Problem 4 : Password Storage

An attacker that gains access to the PayBud database (e.g. via. SQL injection) obtains all passwords, and thus gains access to all user accounts.

Part 1 Fix this problem, by storing passwords securely. Follow the instructions in Schneider, Section 5.1.2, to store only salted hashed passwords. Explain how you modified the code (which methods did you add/modify & how?).

Hint: `https://www.baeldung.com/java-password-hashing#2-implementing-pbkdf2-in-java`. Use SHA256, 256-bit salt & hash, 100.000 iterations, and `java.util.Base64.getEncoder`. You'll need a `salt` column in the `users` table, and to replace the `password` column with `hashword`; see the included database file.

Alice: "Joana leaked the password database when she quit! We must act now!"
Sally: "Relax! The passwords were all stored hashed. ... ... Alice? Why that look?"

Part 2 PayBud is vulnerable to an Offline Dictionary Attack. Exploit this vulnerability by following the instructions provided alongside this assignment. (~4 sentences)

Note: Explain every step of your attack; include each command, explain what it does, explain why you're doing it (i.e. to what end), and show that your attack is successful.

Why would the attack have been even easier if we had only hashed, and not salted, the passwords? (i.e. what is the purpose of salt?) (~1 sentence)

Part 3 Fix this vulnerability, by enforcing a password policy, to disallow weak passwords. Define a policy; what is a valid password according to your policy? (~4 sentences)

Note: We'll test your policy against the RockYou (pass)word list. Try to ensure that your policy prevents any password contained therein, without making creation of valid passwords too difficult.

Implement your policy in PayBud. (which methods did you add/modify & how?).

Hint: We recommend using Passay to implement your password policy. `http://www.passay.org`

Part 4 What can you do to prevent **Offline** Dictionary / Brute-Force Attacks when the passwords are weak by nature (PIN numbers)? (~1 sentence) What can you do (besides 2FA) to prevent **Online** Dictionary / Brute-Force Attacks? (~1 sentence)