Research Document ITS

Main Research Question: Which hashing algorithm should be used for securing sensitive data, such as passwords?

Subquestions:

1. Why is hashing important?
2. What are the main differences between various hashing algorithms?
3. Which one is most suitable for the project?

Research Methods:

Literature study, Available Product Analysis → Security Test → Prototyping(DOT methodology,2021)

Research Question 1: Why is hashing important?

Hashing is an industry standard in securing sensitive information. Without it, all data would be stored in plain text, thus making it easily accessible to an attacker who would want to maliciously use it. Another important part is that hashing is one-way, meaning that once hashed, it should not be possible to hash the original input back into plain text.

Research Question 2: What are the main differences between various hashing algorithms?

Different hashing algorithms are more suited to protect against certain kinds of attack. For instance, the most popular on the market right now, Argon2, is secure against both side-channel attacks and GPU attacks(Moritz Halbritter). In comparison, BCrypt, another popular option, is only secure against side channel attacks, given the right parameters, which in some cases, can be very costly in regard to performance and memory consumption(Alex Biryukov et.al, 2017).

Research Question 3: Which one is most suitable for the project?

Coming down to a particular choice depends on measuring what threats are the most concerning in the current project. This also means that research should also be done on the different kinds of potential attacks or threats that could come up in securing the application.

In this case, hashing would be needed for passwords and/or emails. So to protect against them, both side-channel attacks and GPU attacks must be taken into consideration(Wikipedia, 2022). As mentioned in Question 2, there are two options, BCrypt or Argon2 and with the noted performance loss in BCrypt, Argon2 would be the most suitable tool.

Main Research Question: Which hashing algorithm should be used for securing sensitive data, such as passwords?

With the answers portrayed in the aforementioned research questions, a conclusion can be drawn that Argon2 is the best available option for solving the problem at hand. Being significantly faster than BCrypt and also producing secure hashes with significantly less memory-consuming parameters(Alex Biryukov et.al,2017), it would definitely get the job done.

References

Halbritter, M. (2016). Argon2 Binding for the JVM. GitHub repository README on https://github.com/phxql/argon2-jvm.

Alex Biryukov et.al(2017). Argon2: the memory-hard function for password hashing and other applications. Whitepaper on https://github.com/P-H-C/phc-winner-argon2/blob/master/argon2-specs.pdf at **Problem of existing Schemes**.

Alex Biryukov et.al(2017). Argon2: the memory-hard function for password hashing and other applications. Whitepaper on https://github.com/P-H-C/phc-winner-argon2/blob/master/argon2-specs.pdf at **Our Solution**.

Wikipedia(2022). Side-channel attacks. Wikipedia page on https://en.wikipedia.org/wiki/Side-channel_attack.

DOT Research Methodology(2021). Research Methods on https://ictresearchmethods.nl/Methods at (Literature study, Available Product Analysis, Security Test, Prototyping)