Security Tools Lab 1 Project 4

– Passwords (46 pts)

This can be an individual project as well as a group project (2 students)

Despite the well-established problems facing password-based authentication, it continues to be the dominant form of authentication used on the web. Complex passwords that are difficult for an attacker to guess are also hard for users to remember which leads to users creating weaker passwords to avoid the burden of recalling them. In fact, with the increase in the number of passwords users are required to store, they often reuse passwords across websites.

Password managers can help users more effectively manage their passwords. They reduce the cognitive burden placed upon the user by generating strong passwords, storing those passwords, and then filling in the appropriate password when a site is visited. The user is now able to follow the latest security advice regarding passwords without placing a high cognitive burden on themselves. But password managers are still vulnerable to other forms of attack. For instance, some browser-based password managers like LastPass and RoboForm has been shown to be vulnerable to cross site scripting attacks and network injection attacks because of their autofill features. Another vulnerability lies in password generation stage which some password managers offer, where the generated password is not strong and can be easily guessed through various forms of password cracking like dictionary attack, rainbow table attack or rule-based attack. In this project you will evaluate various password generators and the strength of the password they generate – not the bit quality but the resistance to guessability through various password cracking mechanisms.

This is a non-exhaustive list of freely available password generators:

- RoboForm: https://www.roboform.com/password-generator
- Passwords Generator: https://passwordsgenerator.net/
- LastPass: https://www.lastpass.com/features/password-generator
- DashLane: https://www.dashlane.com/features/password-generator

On top of online ones, you can also write your own python script based on a random number generator like Linux's /dev/random for instance. In your report, compare at least 3 different sources.

Generation must be parameterized by character classes—letters, letters and digits and letters, digits and symbols — and password length — 8, 12, and 20 characters long—in order to determine if these options had any effect on the randomness and ultimately guessability of generated passwords. Generate large enough samples across the various generators and across the various parameters to make your comparison statistically significant.

To analyse the quality of passwords generated in terms of predictability you can either:

1. Create your own tool based on the various mechanisms we learned in Module 2
2. Use a readily available tool such as Dropbox's zxcvbn : https://github.com/dropbox/zxcvbn

If you choose 1, it is considered a group project and 2 will constitute an individual project. Across generators and parameters, determine the strength of password generation with aim of doing a holistic comparison and analysis.

Password generation implementation – 12 (8 if group)

Password quality implementation – 10 (14 if group)

Complexity of work – 10

Results, Evaluation & Discussion – 8

Report Quality & Presentation of work – 6