Research Proposal
Protecting Password Managers against infected computers.

Abstract :
In this paper, we propose methodologies that password managers (PMs) can employ in order to be safe against the attacks generated from inside the system. Here we survey the working of some highly used password managers and examine their usage of system memory. Based on the examination we plan demonstrate how an already infected system can be used to exploit the memory usage of the PMs. We will also suggest some countermeasures to reduce the attack surface and harden the PMs.

Introduction :
At the highest level, attacks on password managers can be classified into two types : 1. Attacks generated from outside the system, 2. Attacks generated from inside the system.

PMs use various strategies to perform an autofill on a webpage. Autofill can be automatic, manual or hybrid. The choice of which kind of autofill should be used on a particular webpage is made after evaluating various parameters. For instance, an automatic autofill can only be performed on HTTPS webpages. The autofill functionality can be exploited by an attacker who is in control of the users network by injecting invisible iframes into the requested web page by a user. We have mentioned this example just to highlight the first category of attacks. But, we majorly focus on the second category where the attacks are generated from an infected system. Narrowly speaking, we will focus on protecting PMs from attacks on systems memory.

Jeffery Goldberg, 1Password's Chief Defender Against the Dark Arts, said, an attacker who is in a position to exploit the information in memory is already in a very powerful position. No password manager (or anything else) can promise to run securely on a compromised computer. We agree with the statement, but, with this proposal we are not striving to develop a completely secure system. Our motive is to reduce the attack surface and harden the currently available systems.

Basic Working of Password Managers :

1. Users enter relevant information (password, security questions) into the PM.
2. PM encrypts the information using an encrypted master password.
3. The information is decrypted whenever needed for use.

Tradeoff :
One master password to unlock a password manager data store.

Memory Usage :
Different designs of password managers use the memory differently. The basic functionality, as explained above, of all the PMs remains the same. It is the design that makes all the difference in terms of security.

The contents of memory(associated with the PM) differs according to the state the PM is in. PMs can be in two states Running and Not Running. When the PM is not running all the data resides on the disk. Now, the security on the disk depends on how aptly does the PM follow the NIST cryptographic specifications.

The major problem lies when the PM is running. In this state the PM uses systems memory and at some stage it has to decrypt the user data. Also, the encrypted master password resides in memory. Security of the data also depends on how effectively the data in memory is cleaned when the PM transitions from locked to unlocked state.

Topic of focus (Survey and vulnerabilities) :

1. Can we provide a cryptographic solution to the above mentioned trade-off?
2. How can we make the use of SGX to reduce the attack surface?
   (Traditional password storage and retrieval **VS** password storage and retrieval based on SGX)
3. What can we do to protect the passwords against Keylogging and Clipboard Sniffing.
4. With continuing research on this topic we can come up with some additional protective measures.

References:

https://crypto.stanford.edu/~dabo/pubs/papers/pwdmgrBrowser.pdf
https://www.ise.io/casestudies/password-manager-hacking/
https://www.schneier.com/blog/archives/2019/02/on_the_security_1.html
https://www.zdnet.com/article/critical-vulnerabilities-uncovered-in-popular-password-managers/