110550126 曾家祐 critique1

D. Silver, S. Jana, E. Chen, C. Jackson, and D. Boneh, “Password managers: Attacks and Defenses,” in Proceedings of USENIX Security, 2014.

1. Summary

這篇論文主要是要解決因為網站數目的增加，在不同網站上使用不同組帳號密碼的數量也隨之增加，為了更方便記住密碼，用戶會使用密碼管理器來輔助管理密碼，但若密碼管理器的自動填充策略不當(ex: 填寫時機錯誤)則會使得密碼在用戶不知情的情況下被攻擊者竊取(ex: 流氓路由器)，而密碼管理器所提供的雲端同步更讓用戶在其他裝置上所使用的密碼有被竊取的風險。這使得密碼管理器很容易成為一個嚴重的資安漏洞。

而隨著科技的進步，生活中越來越多東西都電子化，包括個人資料和營行帳戶等。而我們的帳號密碼則是保護這些資料的關鍵，但由於我們擁有的帳號密碼數量很多，因此我們不得不使用密碼管理器幫助我們更方便登入系統，但若是這些密碼管理器有漏洞，則讓攻擊者可以竊取我們的帳號密碼，而且將不是單一個網頁的帳號密碼，而是所有我們紀錄在密碼管理器上的帳號密碼，從而造成個資外流或金錢上的損失。因此研究如何強化這些密碼管理器而保護我們的帳號密碼是很重要的。

研究者先是分析了十個市面上常見的密碼管理器面對不同種填寫密碼情況時會如何自動填寫密碼(ex:當網路協定和首次填寫密碼不同時會拒絕自動填充等)，並運用三種 Sweep Attacks、Injection Techniques、Password Exfiltration 常見的網路攻擊方式和研究者自己設計的“clickjacking” attack 來觀察攻擊方式和前面密碼管理器面對不同情況時反應的情況做對比後，總結出攻擊方式和密碼管理器漏洞之間的關係。

了解到了這些攻擊的原理方式後，針對這些攻擊背後的方式(例如植入 javascript 等)進行防禦，但因為現在瀏覽器技術的原因，瀏覽器很難檢測到 JavaScript 的注入，(研究中的所有攻擊都需要 JavaScript 注入)，所以不在有 JavaScript 的網頁自動填寫密碼是難以實現的，而直接封禁 iFrame 又會對正常使用 iFrame 的網站造成不便。

所以研究者透過其他方法研究，提出了 Forcing user interaction、Secure Filling、Server-side defenses 兩種用戶端、一種 server 端可以加強密碼管理器安全性的設定和防禦方法，並向這些密碼供應商表明他們的研究並促使這些網路供應商針對自己的弱點修補和增加安全性。

2. Strength(s) of the paper

在這篇論文中分析了十種市面上常見的密碼管理器，並詳細分析了它們在面對不同種情況時(同網域不同網址的網頁、傳送的網路協議改變、表單元素不同)會如何自動填寫密碼(不填寫、完全自動、需與用戶交互)，隨後統計出了最普遍最簡單的攻擊方式如何攻擊，並提出了簡單卻能夠有效防護這些攻擊的方法。

除了尋常的網路攻擊之外，研究者還觀察了透過嵌入式設備來攻擊竊取密碼，這種一般情況下比較不容易注意到的攻擊方式讓密碼管理器的安全性，在用戶不會注意到的地方也能夠有所提升。

在提出解決方法來防禦攻擊者的攻擊時，也不只根據理想情況提出防禦方法，而是綜合起其他技術原因提出不影響網頁運作和實用性的情況下提出了解決方法而可以真正的解決問題。

最後還將它們研究到某些密碼管理器的漏洞通知了密碼管理器廠商，提高了當時密碼管理器的安全性強度。

3. Weakness(es) of the paper

在文中探討的攻擊方式是屬於攻擊者不在場，只透過修改封包數據等相對簡單的攻擊來進行測試，雖然也有效提高了密碼管理器的安全性，但若攻擊者使用的技術更高，或者是文中並未考慮到的情況來竊取使用者的密碼，密碼管理器還是有一定的資安風險。

此外，文中討論的防禦方式，主要都是在客戶端進行被動防禦，當攻擊者發動了攻擊了之後，我們才在透過 Forcing user interaction、Secure Filling

等方法進行防禦，若是可以多提出一點在 server 端從源頭就想辦法阻攔攻擊者不讓攻擊者找到漏洞攻擊或者是攻擊失效則更可以更好的保證用戶各方面包含使用密碼管理器的安全。

最後，則是這篇論文是在 2014 年發表的，相對於現在的 2023 年許多攻擊方法已經過時，且有更高級的攻擊方式了，且文中所提到的有些問題現在也因為網路協議的更新而很少出現了，但這篇研究還是給予了當時密碼管理器的安全性很大的提升。

4. Your own reflection

在這篇論文中我了解到了密碼管理器的基本運作原理和最基本的網路攻擊方式，以及如何應對這些攻擊方式，此外還讓我更加了結如何做一個研究，除了解決現在已經出現的網路攻擊外，我們還可以自己創造攻擊，讓研究結果更為全面更能普遍解決問題，應用到現在的科技上。

正如同我前面所述，若我是作者，既然無法檢測到 javascript 的注入，那是否可以檢查網站的 javascript 代碼，是否有問題，例如可能對用戶的安全性造成危脅的屬性設定，或者是表單傳送目的地與預期不一致等問題，在用戶接觸到輸入密碼前就對用戶發出警告並建議不要使用自動填充等方式來加強安全性。