密碼工程 Critique1

分組名單：

姓名	學號
陳以晴	111550178
蔡昀錚	111550035
莊婷馨	111550057
謝詠晴	111550113
鄭芯薇	111550064

1. Name of the paper:

Password Managers: Attacks and Defenses

David Silver, Suman Jana, and Dan Boneh, Stanford University; Eric Chen and Collin Jackson, Carnegie Mellon University.

2. Summary:

這篇文獻比較四種瀏覽器(Google Chrome 34, Microsoft Internet Explorer 11, Mozilla Firefox 29, and Apple Safari 7)內建的密碼管理器（browser built-in password managers, mobile password managers, 3rd party managers）自動填寫密碼（autofill）的策略，提出這些密碼管理器常見的安全性漏洞以及攻擊者如何獲取使用者資訊，如果採用鬆散的安全措施，例如無視網頁安全狀態自動填寫密碼，則用戶可能遭遇極高的安全風險。

首先，作者接著展示「咖啡店攻擊者」（melicious network）如何在無需任何用戶互動的情況下竊取已儲存的密碼，例如放置透明的 iframe，或是更改密碼送往的目標網站，使密碼管理器自動填入的密碼外洩。另外，Cloud-based password syncing 技術會讓使用者暴露在密碼外洩的風險，因為攻擊者可以輕易竊取他們未在該設備上存取過的密碼，甚至未曾登錄過的網站密碼均有可能被洩露。

接著，作者提出了改進密碼管理器安全性的方法，比如需要「用戶互動」(user interaction) 才能自動填寫密碼，告知用戶目前自動填寫的網站為何，或是在 HTTPS 憑證驗證失敗時絕不自動填寫密碼。作者也提出「安全填寫」(secure filling) 機制，利用限制 Javascript 的讀取，來保證透過密碼管理器自動填寫的密碼可以安全送出，在某些情況下甚至比手動輸入密碼更安全（例如：透過 HTTP 獲取，但由 HTTPS 提交的登入頁面）。

此外，論文還提到了密碼管理器在現實世界的弱點，例如使用 iFrame 進行攻擊或在登入表單中插入 JavaScript，從而在用戶無意識的情況下竊取密碼。在文獻中，作者使用 sweep 攻擊並實驗各種加強密碼管理器安全性的技術，展示他們如何應用在現有的密碼管理器中。由於作者的發現，LastPass 不再在 iFrames 自動填寫密碼，1Password 也不再提供從 HTTPS 填入密碼到 HTTP 頁面，這些策略皆提高密碼管理器的安全性。

3. **Strength(s) of the paper:**

1) 完備性強：對於當時常見的 password manager 都有涵蓋到，列出的 password manager 種類多樣，分成三類包含 Desktop Browser PMs, 3rd Party PMs, and iOS PMs。
2) 文字淺顯易懂：文章用詞直白且會附上舉例，提升讀者的易讀和理解性。
3) 表格化彙整：使用表格詳細列出與比較不同 password manager 間對於自動填寫密碼的方針，讀者可以清楚觀察到其優缺點。
4) 多方面分析：關於自動填寫密碼的方式，考量到了多種情況下的可能性，如設置 auto-complete 參數或是連接網路時 HTTPS 的穩定性。
5) 嚴謹的驗證：針對不同狀況下的自動填寫密碼方式(如同網域不同網址的網頁、需與用戶互動、不同裝置的同步或連線不佳時)，都有進行攻擊的實作。
6) 攻擊方式分類清晰：對於攻擊的步驟及方式有明確且邏輯性的分類，Sweep Attacks、Injection 和 Password Exfiltration 三步驟下，分別再有子細項的攻擊方式，如 Sweep Attacks 下有再分成 iFrame, Window 及 Redirect sweep attack。
7) 提供改善建議：不僅列出攻擊的原理方式及結果，也提供該如何改善防禦及進步的建議，且考量了用戶使用方便性及網頁運作上等實務性上的因素。

4. Weakness(es) of the paper:

1) 圖像化不足：內容以大量文字為主，雖然有表格但過於擁擠，且缺乏數據分析，可以考慮加圖表來增加不同管理器各方面的可視化。
2) 時代侷限：如現今新興瀏覽器（例如：arc）未在論文中出現，反之內文所提及之 Windows 8.1 pro 已在 2023 年 1 月終止服務，在實用性上不免有所侷限。另外現今攻擊手段日新月異，此篇論文所提出的理論逐漸缺乏參考價值。
3) 實踐展示缺乏：缺乏列出實踐方式的程式碼以及明確的試驗過程呈現，也未提供詳細實驗數據與結論，僅憑口述結果方式較為缺乏說服力。
4) 安全威脅的多面向：文中對於其他潛在的安全威脅，如密碼管理器數據庫的加密強度、後端雲同步的安全性等，探討可能不夠全面。
5) 理想與現實的差距：實驗驗證主要在特定的設置和環境下進行，可能無法完全反映所有真實世界的攻擊情境。這意味著實際攻擊成功率可能與研究結果有所差異。

5. Your own reflection:

A. What did you learn from this paper?

從這篇論文中，我們了解密碼管理器存在不少安全隱患，不論是自動填入或是手動填寫均有密碼洩漏的風險，儘管沒有被駭客入侵，也可能在無意中因為連接到惡意網路而被攻擊者取得所有存取過的密碼。

透過文獻使用的攻擊手法，我們也更加了解密碼存取的原理以及他們如何被洩漏。此外，我們也學到其在安全設計上面臨的複雜挑戰，以及如何透過深入分析和創新方法來提升這些工具的安全性。而在實際驗證各平台的密碼管理器的安全性時，也要盡可能地考量到各種狀況下的情境，以及確保應用到現實層面時是可運作的。

B. How would you improve/extend the work if you were the author?

1) 首先可以考慮更廣泛的覆蓋範圍，探索更多不同類型的瀏覽器和不同平台上的密碼管理器，並將使用頻率列入考慮來得出大眾更加在乎什麼部分（例：也許對一般民眾，便捷性遠比安全性重要)，也可包括地區性的密碼管理工具，以獲得更全面的安全性評估。
2) 進行深入的用戶研究，了解用戶對於密碼管理器的使用習慣、對安全提示的反應以及對改進措施的接受度，從而設計出既安全又便於用戶接受的解決方案。
3) 增加實際模擬，可能包含自製或嘗試攻擊，倘若能夠有確切效益的作品，也可長期評估和追蹤加上自行改進措施實施後的影響，包括安全性提升的效果以及可能出現的新問題或挑戰。
4) 結合 2014 年以後出現或投入實用的技能，例如：密碼生成評估、網站漏洞監控、多因素驗證（MFA）/ 二次驗證等。

C. What are the unsolved questions that you want to investigate?

1) 密碼管理器在面對先進持續性威脅（APT）攻擊時的表現如何？
2) 在多設備和跨平台使用環境中，如何有效同步和保護密碼數據？
3) 密碼管理器如何更好地融入未來的身份驗證框架，例如結合生物識別技術或零信任安全模型？
4) 對於現在廣泛出現的二次認證(包含學校 e3 也有要求)，其效益和密碼管理器相比如何?是否因為已知密碼管理器終究不夠可靠才有相應的大量二次認證出現？
5) 如何從一開始就阻攔攻擊者，使其無法抓到漏洞進行攻擊，而非等到被攻擊後才透過 Secure Filling 等方式進行防禦？

D. What are the broader impacts of this proposed technology?

1) 提升大眾對網絡安全的意識：閱讀完這篇論文後，我們更加了解到大眾在登錄頁面或連接公共網路時，通常會忽略跳出的安全性警訊，從而增加了使自身暴露於個人資訊被盜的風險中。此外，使用者為了使用上的安全性，往往選擇自動填寫密碼，而忽略了密碼管理器有安全性的疑慮，此篇論文則使大眾在選擇密碼管理器時能更為謹慎。
2) 影響政策和標準制定：文中對於各瀏覽器平台的自動填寫密碼有完整性的研究及攻擊測試，這些研究成果有助於促進相關平台加強自身的安全性，並讓政府或資訊管理協會更新安全標準和政策。
3) 對於網絡開發者來說，他們需要時常更新並檢查瀏覽器的設定以確保網站密碼管理器的安全性，例如：可能需要修改現有的登錄流程以適應新的安全協定；還有很重要的一點是，開發者需要在加強密碼管理的安全性與最佳化使用者體驗間找到平衡，比如若跳出太多次密碼驗證的畫面可能會令使用者感到十分煩躁。

E. Else?

綜上所述，這篇論文不僅加深了我們對密碼管理器安全性的理解，讓我們更加了解各種攻擊手段的原理，也激發了對未來研究方向和實踐應用的思考，對於提升整體網絡安全生態與大眾網路安全意識有著深遠的影響。

Extra Credit：簡易密碼管理器實作

這段 code 實現了一個具有圖形使用者介面（GUI）的簡單密碼管理器原型，使用Python的Tkinter庫來創建。它允許使用者通過圖形介面添加、存儲以及在安全條件下自動填充網站登錄憑證。以下是代碼的詳細解釋和使用步驟：

代碼作用

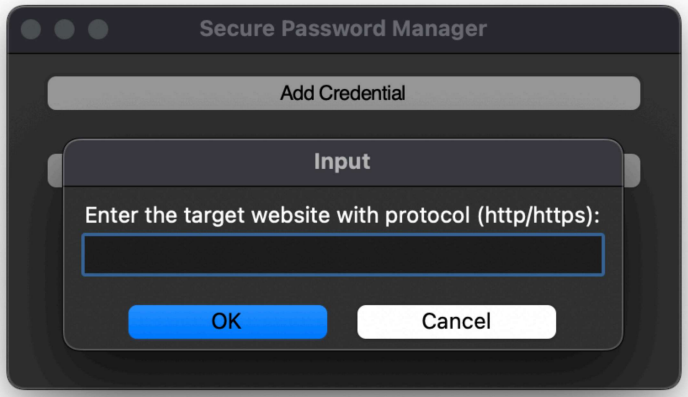
1. **初始化密碼管理器**：定義了一個SecurePasswordManager類，用於管理使用者的登錄憑證。這些憑證以網站域名（包括協議，如http或https）為鍵，使用者名稱和密碼對作為值，存儲在一個字典中。
2. **添加憑證**：使用者可以通過輸入網站的完整地址（包括協議），使用者名和密碼來添加新的憑證。這些資訊將被存儲在密碼管理器中。
3. **自動填充憑證**：當使用者嘗試自動填充一個網站的憑證時，密碼管理器會檢查提供的域名是否以http://開頭。如果是，出於安全考慮，自動填充將被禁用，並提示使用者。如果是安全的https://連接，且憑證存在，密碼管理器將返回使用者名和密碼。

使用步驟

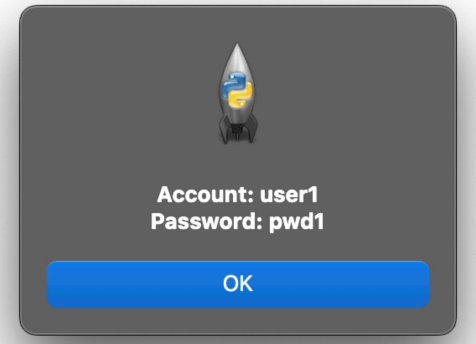
1. **啟動程序**：執行代碼後，會出現一個帶有兩個按鈕的窗口：Add Credential（添加憑證）和Autofill Credential（自動填充憑證）。

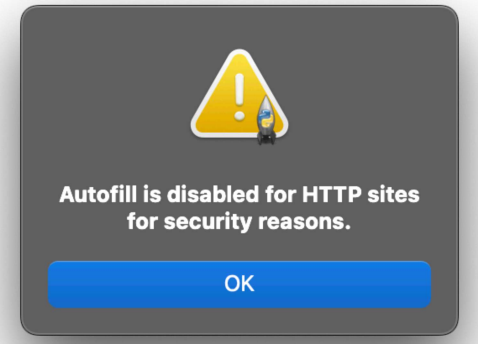
2. **添加憑證**：
    1. 點擊Add Credential按鈕。
    2. 在彈出的對話框中依次輸入目標網站的地址（包括http或https）、帳戶名和密碼。
    3. 提交後，資訊將被儲存，同時彈出一個提示框，告知使用者憑證已成功添加。

3. **嘗試自動填充憑證**：
    1. 點擊Autofill Credential按鈕。
    2. 在彈出的對話框中輸入要自動填充的網站地址（包括http或https）。



    3. 程序會檢查輸入的地址是否符合安全填充的條件：
        1. 如果是http://開頭，將顯示一個警告，告知使用者不能自動填充。
        2. 如果是https://且憑證存在，將顯示帳戶名和密碼。
        3. 如果域名不匹配或使用HTTP，將提示錯誤。





注意事項

1. **安全性**：出於演示目的，密碼在此原型中未進行加密處理。在實際應用中，應確保對敏感資訊進行加密。
2. **HTTP和HTTPS**：自動填充功能只在使用HTTPS的網站上啟用，以保護使用者數據的安全。
3. **GUI介面**：通過Tkinter創建，包括基本的文字輸入和提示框，使使用者操作直觀簡單。

這個原型提供了一個基礎框架，展示了如何使用Python和Tkinter創建一個具有基本功能的密碼管理器GUI應用。在開發完整的應用時，可能需要添加更多高級功能和安全措施，如密碼加密、多因素認證等。