Critique 1

姓名	學號
朱驛庭	111550093
王睿宇	111550085
邱振源	111550100
黃佑得	111550160
陳 燁	111550124

Name of the paper

> Password Managers: Attacks and Defenses

Summary

在這篇文章裡，作者討論了四種瀏覽器(Google Chrome 34, Microsoft Internet Explorer 11, Mozilla Firefox 29, and Apple Safari 7)下密碼管理器 (password managers, PMs) 的潛在安全漏洞，可以對其造成攻擊的方式，以及應對這些攻擊所可以採取的防禦措施。在文中，作者在一開始提到了密碼管理器的概念與原理，同時，也敍述其重要性，並強調密碼管理器的本身可能成為攻擊者的目標，從而使密碼管理器本身成為潛在的安全漏洞。

作者先提及了「咖啡店攻擊」，可以藉由放置透明的iframe或是改變密碼可能填寫並傳輸到的網站目的地，便可以在使用者不知情的情況下成功竊取密碼，從而造成密碼外洩。再者，如果密碼管理器是存在雲端的資料庫(Cloud-based password syncing)可能會抗攻擊找直接攻擊雲端資料庫，使使用者不知道自己的密碼已被竊取。

而後，作者在文中提到了其他針對密碼管理器的許多不同攻擊方法，例如：藉由密碼管理器的擴充套件，或其他程序指令來竊取其中的密碼；抑或是透過存在於背景或縮小使人不易發現的視窗，來誘導密碼管理器自動填充貯存其中的密碼，從而洩漏其登錄憑證；或是針對密碼管理器本身的設計缺陷攻擊，進而使密碼洩漏。

為了防禦這些攻擊模式，作者也提出了一些防禦的策略，例如：讓密碼管理器在某些條件下不自動填充(像是錯誤的HTTPS驗證憑證)，並讓密碼管理器在自動填充密碼之前先行與信任的瀏覽器UI進行互動；或是提出一種「安全填充(secure filling)」的方法，由於有的時候自動填充還是可能比手動填充更加安全，作者希望持續改進安全填充的方法，並藉此提升密碼管理器的安全性。

Strengths of the paper

1. 這篇論文針對了市面上各種不同的密碼管理器 (password managers, PMs) 進行全面的評估，包括深入分析它們的自動填充政策（autofill policy，詳見表格1）並評估面對不同類型掃描攻擊（sweap attack）的韌性。這些評估的結果被系統性地整理成表格，使讀者能夠輕鬆獲取和理解這些數據。

2. 完成測試後，他們將這些研究結果分享給了測試的密碼管理器的開發者。他們告訴這些開發者測試階段發現的某些攻擊類型的漏洞，並提出可能的改進策略以增強存儲密碼的安全性。這種反饋對於密碼管理器安全性的持續改進非常有幫助，能確保用戶憑證能夠抵禦不斷變化的威脅。

3. 這篇論文的脈絡十分清楚，從淺到深的介紹讀者應該要會的各種名詞與知識，讓讀者從基本概念無縫過渡到更高級的話題。一開始講述了密碼管理器所採取的不同自動填充模式和政策，為讀者提供整篇論文的基礎。接著，將焦點轉移到各種 sweap attack 模式，解釋這些攻擊是如何運作的以及它們對密碼安全可能構成的風險。最後則討論如何防禦這些攻擊手段，提供開發者和用戶可以採用的策略來減輕這些威脅。這種方法確保了每個人都能獲得對主題的全面理解，增強他們保護敏感信息的能力。

Weaknesses of the paper

1. <Password Managers: Attacks and Defenses>發表於2014年，但與2024年相比，許多文章中提到的攻擊方法已經過時。然而，這篇論文仍然幫助了一些密碼管理器修正了他們發現的一些錯誤。隨著科技的進步，技術和攻擊策略不斷進化，許多在論文中提到的攻擊手段已不再是主流或有效的威脅。儘管如此，這篇論文的價值並未完全消失。它為密碼管理器的開發者提供了一個寶貴的學習機會，讓他們能夠檢視和改進自己的安全措施。

2. 在這篇論文中，提到了許多不同的攻擊方法，但是這篇論文並未提供這些attack如何被實際實施。我認為，如果能夠包含一個GitHub頁面來提供這些攻擊的程式碼，對於那些有興趣的人或對實作感興趣的人來說會非常有幫助。這樣能促進學術和實務界之間的合作與交流。提供程式碼可以使閱讀這篇文章德人理解這些攻擊的具體實施方式，並有可能發現新的防禦策略或改進現有方法。

3. 文中討論的防禦方法主要是在客戶端進行的被動防禦。論文主要關注的方法，如user interaction 和 secure filling。如果我們能夠在伺服器端提出更多的解決方案，從源頭上阻止攻擊者，防止他們找到漏洞來利用或使攻擊無效，這將進一步保障用戶在所有方面的安全，包括使用密碼管理器。透過加強在伺服器端的安全措施，例如實施更複雜的身份驗證機制、加密通訊協議，以及及時修補已知的安全漏洞，我們可以有效地降低攻擊者成功入侵系統的機會。

Reflection

在這篇論文中，我們學到了很多的密碼管理器，像是 LastPass 還有 Apple Safari 7，這篇論文也詳細描述了在不同的設定下這些密碼管理器會有哪些不同的行為，像是在 HTTPS/HTTP ，

autocomplete 這個 attribute 欄位、html 中 form 裡面 action 的 attribute；作者之後也提供了一個基於中間人攻擊的威脅模型，基於這個模型，他們提出了幾種攻擊模式：sweep attack、injection、password exfiltration、Attacks That Need User Interaction。最後，這篇論文提供了兩個加強密碼管理器的解決方案：Forcing User Interaction 和 Secure Filling.

不過，因為這篇論文缺少一些相關實作。雖然作者詳細描述了很多攻擊模式還有幾個加強策略，但是他們卻沒有展示出來究竟他們是怎麼完成的，我們認為說如果作者可以提供一些實作範例到網路上或是一些相關平台會讓讀者們對於這篇論文有更好的理解。

其中一個我們認為還沒有被解決的問題是，我們該如何開發新的密碼管理器，這種密碼管理器不僅需要保持用戶習慣的使用體驗，同時還需要提升安全性。我們覺得一個可行的方法是將密碼管理器和生物辨識技術結合起來。例如，利用像是現在流行的 FaceID 和 TouchID 這樣的技術。另外一個待解決的問題是，我們該如何實現通過一個不安全的路由器來注入 JavaScript 的技巧。

最後，這篇論文所帶來的顯著影響之一是，密碼管理器的供應商們已經開始嘗試解決他們的自動填充政策問題。文章裡提到，LastPass 決定不再自動填充位於 iFrames 中的密碼欄位，而 1Password 則不再允許從 HTTPS 頁面自動填充密碼到 HTTP 頁面。