隨著用戶因網站數目的增加，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，而使用密碼管理器。 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。

在調查中的十個密碼管理器中大致上可分為「需與用戶交互的手動自動填充」和「不需與用戶交互的自動自動填充」

在這十個密碼管理器中，每個管理器都支援將同一個網域的網站視為同一個網站，同時也接受填寫密碼的表單方式改變而自動填寫密碼，這使得攻擊者可以從該網域中最弱的網站來攻擊竊取密碼。而使用的網路協定不同（ex:http,https）則有些管理器會拒絕自動填充，而若不注意協議的改動無論是否有手動交互都不安全，因為一種攻擊方式就是將表單中的操作改為受到攻擊者的控制。

接下來考慮兩種情況，第一種情況是不同的url，有些管理器會與用戶進行互動表示與首次操作的頁面為不同來源，但另一種情況：相同的url網站，但修改表單使目的地不同，則所有密碼管理器都允許這種情況下自動填寫。

有些網站可以修改「自動完成」屬性 來增強來簡單防禦一些攻擊，但也因為實用性的降低而使得相對不受歡迎。

此外有一些密碼管理器額外具有以下值得一提的安全功能：

1. 對網頁iframe 中表單的自動填寫的限制
2. 不填寫能見度為0的表單（但會填寫能見度不為0的因此並無增加安全性）
3. 一鍵完成填寫帳密的一系列操做
4. 一鍵完成除了填寫帳密加上提交密碼，讓用戶無須與表單交互

接下來研究者模擬攻擊者使用了不直接觀察到用戶填寫資料的過程，只透過監控或修改網路封包的方式來進行相對較弱的攻擊方式，分別試驗了以下幾種攻擊方法：

1. Sweep attacks(掃描式攻擊)：

Sweep attack主要依賴於修改網路封包並有iFrame、Window和Redirect三種方式對用戶進行攻擊：

* iFrame Sweep attack是透過不可見的iFrame 讓密碼密碼管理器加載iFrame 時自動填充相對應的密碼，而攻擊者可以透過對iFrame加註的javascript代碼來竊取用戶資料。
* Window Sweep attack 則是透過⽤⼾禁⽤彈出窗⼝阻⽌程序，讓javascript可以偽裝成彈出式視窗來竊取資料。
* Redirect sweet attack 則是攻擊者透過將用戶網頁導向較脆弱的網頁，竊取資料後再導回原網站。

1. Injection Techniques(注入技術)：  
    因密碼管理器會將同一網域的帳密相關聯，所以攻擊者可以透過同一網域中的http來竊取原本使用https提交方式。

攻擊者還可以透過嵌入式裝置來竊取資料，因為許多嵌入式裝置因為加設該通道安全而默認使用http，而使用https者攻擊者也可以透過與嵌入式裝置網站的有效證書來進行前面的掃瞄攻擊。

當訪問一個有問題的https網站時(例如：https驗證錯誤)，瀏覽器通常會給予警告，但較算用戶不輸入任何訊息，還是會因為密碼管理器的自動填寫而遭竊取資料。

而通過前面幾種攻擊方式所綜合起來的攻擊方式則更難防禦，此外跨網站的腳本若在任一網頁上有XSS漏洞則會讓腳本有被竄改的風險進而將JavaScript加到登入表單中。

1. Password Exfiltration(密碼洩漏)：

* 攻擊者可以透過網頁中添加的不可見iFrame來加載密碼並盜取，
* 因有些密碼管理器在表單操作與首次不同時依然會自動填寫並送出，因此攻擊者可以直接修改表單內的元素使用戶主動洩漏密碼。

因為前面所有的攻擊方式都與JavaScript 注⼊有關，因此防⽌密碼管理器在易受 JavaScript 注⼊攻擊的⾴⾯上⾃ 動填充密碼是一個解決方案，但難以實現，另一個方法是完全阻止iFrame 中的⾃動填充。但此方法不能完全防護，也會給正常使用iFrame的網站造成不便。

研究者提出了幾種可行方法來保護用戶

1. Forcing user interaction(強制⽤⼾交互)：

5.1 Forcing user interaction

5.2 Secure Filling

5.2.1 Limitations of secure filling

5.3 Server-side defenses

6  Related work

7 Conclusions