USENIX'14 - Silver - Password Managers: Attacks and Defenses

概述: (S)由于口令管理器提供了隐私数据的存储，研究其安全性是必要的；(T)本文重点关注于流行的PM的autofill策略，(A)通过在4个平台上的10个流行PM进行测试实验，给出了不同PM的自动填充策略和特征，指出PM容易遭受sweep attacks，攻击可以通过注入JS在无用户互动时拿到用户的口令，(R)本文对不同的攻击方式进行了详细描述，并给出了相对的对抗方法。

优点: ①系统调研了PM的autofill policy，提出了sweep攻击，并给出了改进方法
②参考文献介绍较好
③在攻击能力较弱的情况下即可完成攻击(许多研究攻击能力过强)

问题: ①本文内容集中于autofill，而标题为PM的攻防，标题应更集中于研究主题
②本文在第五章给出的部分建议不够实际，如要求用户观察action是否匹配，由TLS警告可知，用户通常会忽略提示；
③secure filling中，攻击如果可以访问 PasswordInfo 对象也可以拿到口令，有研究提出在生物识别时使用真实口令；应简要讨论获取 PasswordInfo 的可能性

1. 本文重点研究桌面浏览器的口令管理器，第三方PM，iOS PM和Android PM的auto-fill policy
autofill策略分为自动填充和手动填充(IE11在HTTPS页面自动填充，在HTTP页面需用户交互)

2. Threat Model (攻击目标是在用户不操作的情况下获取PM的口令) ↳一些PM在特定场景中也需人工交互
主动的MITM网络攻击者：当用户登录正常网站时，攻击不能窃听或介入登录过程，当用户连接到攻击控制的网络时，攻击可以注入、拦截或篡改传送的包(evil coffee shop attacker)

3. Autofill Policies (当登录页面与保存凭证的页面不同时，是否应autofill) 可改 lect-secure page
1> domain and path: 相同domain，相同header的多个页面以及重新设计页面可以自动填充
2> Protocol: HTTPS vs. HTTPS：登录与存储的协议不同，一个为HTTP，而另一个为HTTPS
3> Modified form action: 登录与存储的表单的action不同
4> Autocomplete attribute: 当input不提供该属性时，PM是否会autofill(若攻击控制网络，可改属性)
5> Broken HTTPS behavior: 登录时原本的HTTPS连接被破坏，用户可能会无视证书的问题
6> Modified password field name: 登录和存储时的input字段名不同(导致self-exfiltration)

▲额外的PM特征
① iFrame autofill: 是否会填充页面中的iFrame部分
② Visibility: 当输入框不可见(display:none或opacity为0时，是否autofill
③ Autofill method: 是否会根据字段名填充
④ Autofill and Submit: autofill之后是否会自动提交表单

4. 攻击流程(多数情况用户不与网站通信而且不知道口令正在被攻击获取)
1> Sweep attacks (支持自动填充的PM)
3步：攻击者让用户的浏览器访问恶意网站；攻击者篡改流量注入JS代码；JS代码可导出口令
▲比如用户连接到攻击者的热点，用户会重定向到一个页面，但其中有攻击设置一些不可见元素(见下)

①iFrame sweep attack: 热点页面存在许多iFrames，攻击者向其中注入登录表单和JS，当iFrame加载时，PM会自动填充相应的字段，注入的JS可获取口令。
②Window sweep attack: 使用户关闭popup blocker，运用不可见的blocker进行上述的攻击
③Redirect sweep attack: 用户访问a.com，攻击者将其重定向到b.com(空页面)，注入表单和JS生成口令，以此类推，最终重定向回a.com(认为历史记录用户不会在意，可通过形似的名称)
▲当PM在多个设备间同步时，攻击危害会扩大。

2> Injection技术
攻击需篡改流量，只要与其登录页面same origin即可，可能的技术：
① HTTP login page (登录页为HTTP，提交页为HTTP或HTTPS，攻击者很容易注入JS代码
② Embedded devices I: 许多嵌入式的设备的登录页为HTTP，因此可以注入JS代码
③ Embedded devices II: 一些家庭路由器虽使用HTTPS，但为自签名证书，攻击可伪造证书发动攻击
④ Broken HTTPS: 利用用户通常会忽视TLS警告，可实现redirect sweep attack
⑤ Active Mixed Content: 在HTTPS页面包含基于HTTP的active content，例如脚本，也可用攻击
⑥ XSS Injection: 即使在HTTPS网站攻击者可以通过XSS攻击注入脚本
⑦ Leftover Passwords: 攻击利用在PM中保存的过时网站的口令，入侵过时网站获取口令。

3> 口令泄漏(Password Exfiltration)
① stealth: 当表单填充后，攻击者通过iFrame加载已控制的网站，以参数的形式传递凭证
② Action: 攻击者可以通过修改action的指向拿到用户的口令
▲即使在需交互的场景，也可以完成攻击，例如通过clickjacking攻击

5. 增强方案
1> 强制要求用户交互：交互过程应在可信的浏览器UI中进行，不允许JS修改；应展示出要自动填充的domain；在broken https场景，应不提供填充口令的服务
2> Secure filling (在攻击者将恶意JS注入登录表单时，通过HTTPS提交表单的autofill仍是安全的)
① PM在保存时会存储相应表单的action ② 当表单自动填充时口令字段是JS不可读的
③一旦口令或用户名字段被修改，则autofill失效 ④需检查action匹配之时才可以完成自动填充
▲使用不可打印字符填充字段，将action和口令字段值存入一个对象，提交时再替换。
▲局限性
①AJAX时不使用表单的action，且需要JS获取值，不改动很难实现secure filling
②当口令字段的name与明文字段的name相同时可能导致self-exfiltration攻击
③在用户手动输入时无法解决上述问题；在注册时无法使用与登录表单相同的方法。
3> 服务器正常的防护
①在登录和提交页面使用HSTS ②使用CSP防止恶意JS的注入
③将登录界面放在subdomain，限制同源数且减小攻击面
▲给PM供应商改进3问题，有的进行了改进