Usenix Sec'14 - Zhiwei Li - The Emperor's New Password Manager: Security Analysis of Web-based Password Managers

优点：①整个分析了5个web-based PM的实现上的漏洞，指出了4种漏洞，另外指出5个PM中的4个可能被攻击者直接获取凭证

②给出了本文中web attacker的攻击能力，并从UDS框架中给出了PM的安全目标

③安全分析时的用例图很好

问题：①由于本文从技术报告得来，行文更多地分析技术，连贯性不够好（特别是引言）。除此之外，为什么分析这5个PM，以及为什么要分析PM的这些功能而不考虑其他功能应给出解释

②可以从密码学的角度考虑PM的安全性，例如协议和为口令提供的保护

③应给出分析PM还可能存在什么类型的漏洞（即本文找出的漏洞是否全面）

▲引言的逻辑（STAR）.

1）由于在不同网站之间选择随机、安全的口令带来的认知负担，用户倾向于选择简单口令或重用口令

2）PM可以在网站中自动生成和填充口令，且可以在极低的部署代价下提升安全性和可用性（抵抗钓鱼网站）

3）PM才被宣传，LostPass在2011年已经有了超过100万帐户.

4）尽管理想的PM有许多优点，但实现的错误将导致安全问题，本文旨在指出安全问题的根源，并给出best-practice和anti-patterns来指导设计

5）不安全的实现将引入新的漏洞，如单点故障将使攻击者拿到所有口令.

6）对5个web-based PM做安全分析，发现了4种漏洞，包括bookmarklet，web，logic和UI. 其中4个PM在受到攻击后会泄漏用户的凭证；且发现漏洞的原因不同：例如logic或对安全模型的误解.

→ 由于PM的代码经过了混淆和压缩，且各项功能的实现不同，无公开的安全架构，故分析受困难

R 7）认识了PM的攻击面、安全目标和漏洞，揭示了实际中PM通常不安全这一事实，进一步提供了指导和建议避免安全实践，提出使用defense-in-depth来增加PM的安全性.

▲Password Manager（LastPass, RoboForm, MyIlogin, PasswordBox, NeedMyPassword）

1）core：作为一个DB存储不同网站的用户名和口令，

2）PM通过master username和password进行访问控制，用户只需要记忆master password

entry point（登录页），username和password. → credential.

3）在使用过程中，通常要求PM执行代码，故通常需要授权的浏览器扩展或bookmarklet.

4）特定的功能（文中表1提供了不同PM的功能）.

①Collaboration：允许用户与他人共享口令，双方均有帐户访问PM，且口令拥有方可以设置读写权限.

②Credential Encryption：尽可能减少其中存储的明文数据，在web-based PM中，可以使用JS对凭证进行加密（使用KDF生成的master key，例如从master username和password生成）.

③Login Bookmarklets：除了extension，PM还提供了更便携的bookmarklet，允许在当前页执行JS代码

▲Threat Model（本文重点考虑web attacker）

1）web attacker可以控制任意多个web server和DNS domain，且可以使用户登录他控制的网站

2）用户可能在web attacker控制的网站上注册帐户并用PM管理，用户可以依赖于扩展、bookmarklet和网站管理口令，攻击者也可以使用PM服务.

3）PM的代码非恶意，不会盗窃web应用的隐私数据，且不会主动发起未授权共享.

4）用户使用唯一的master password，且不会与其它应用共享.

▲Security Model（high-level：确保存储的凭证只会被授权的用户和对应的网站访问

1）Master Account Security：保证完整性，攻击者不能伪装成用户向PM认证，PM需保证master account的安全性和凭证的安全性（如cookie的安全性），加密凭证的key应一直在client-side

2）Credential Database Security：确保存储的凭证的CIA，攻击者不能获取、修改或删除凭证.

3）Collaborator Integrity：确保凭证共享功能的安全性（应提供访问控制列表）

4）Unlinkability：不允许通过多个网站跟踪1个用户也不应使web应用可关联一个用户的两个帐户

▲Attack Surface.

1）Bookmarklet Vulnerabilities：由于bookmarklet的代码可能在攻击者的JS上下文中执行，执行的过程可能被攻击获取

2）Web Vulnerabilities：应考虑传统的web漏洞，例如CSRF和XSS

3）Authorization Vulnerabilities：需确保所有的凭证共享和更新操作都是经过授权的，漏洞通常是由服务器端的错误检查造成的

4）User Interface Vulnerabilities：考虑直接对PM的钓鱼攻击.

▲安全性分析.（Case Study）.

1）Bookmarklet Vulnerabilities（LastPass, RoboForm及MyIlogin通过bookmarklet提供凭证访问和填充）

▲许多移动设备无法安装extension，只能使用bookmarklet.

•利用的漏洞（由于bookmarklet需要用户的匿名标识符，故三者均可能遭受Linkability attack）

①LastPass：图3的第6步发送的资源为bml.php?x=dropbox.com，该uri是可预测的，且其中包含了敏感的数据（key_rand_encrypted）和dropbox.com的凭证，由同源策略允许攻击者在页面插入并执行脚本.

②RoboForm：第8步使用postMessage发送凭证，但没有进行origin check. 而且由于代码在攻击者的页面运行，因此攻击者可以修改window.referer_url来对应Alice想登录的web应用.

③MyIlogin（click then fill）：在第1步中除了发送加密的用户名和口令之外，还发送了解密密钥；另外MyIlogin在第6步不检查url而是默认将最近点击的url作为该值.

2）Web Vulnerabilities（CSRF：LastPass, RoboForm, NeedMyPassword；XSS：NeedMyPassword）

▲NeedMyPassword的XSS漏洞允许攻击者完全控制帐户，RoboForm的CSRF漏洞允许攻击者向用户的凭证数据库中任意更新、删除和添加凭证.

▲利用的漏洞

①LastPass One Time Password（OTP必须可以向LastPass认证Alice，且可以恢复出master key）→ 且不会向LastPass server泄漏任何信息（包括.com）

设置OTP的过程易遭受CSRF攻击，因为只通过cookie认证，使LastPass无法验证rand_encrypted_key是否与master key的加密值关联。→ OTP attack将可能导致隐私泄漏，离线猜测和DoS攻击

②RoboForm Extension：删除、更新和添加凭证的过程均不包含CSRF token，因此攻击者可以向Alice的数据库中添加、删除、修改凭证.

③NeedMyPassword：可能遭受XSS和CSRF攻击

3）Authorization Vulnerabilities（MyIlogin, PasswordBox，可以在Alice不访问攻击者网站时完成攻击）

•利用的漏洞：

①MyLogin Sharing Credentials (通过公钥来实现共享)
MyLogin在共享web card之前只认证了Alice，但未检查Alice是否有权限共享wcid标识的web card

②PasswordBox Sharing Credentials.
在sharing logic中，PasswordBox不检查Alice是否拥有它共享的asset-id. 使得攻击者可以共享自己不拥有的asset.
→ PasswordBox只加密了口令，因此泄漏用户名和创建时间等会有隐私问题

4> User Interface Vulnerabilities. (RoboForm, LastPass)
用户登录到PM时可能遇到的钓鱼攻击，特别是bookmarklet应要求用户手动新建1个tab并登录，但只有MyLogin遵守了这个实践
•利用的漏洞：通过替换展示给用户的页面进行钓鱼攻击

▲Lessons and Mitigations (concrete guidence & defense-in-depth).
→包括架构、协议或浏览器的改进，PM开发人员在设计时应避免的anti-pattern，指导PM的安全性设计

1> Bookmarklet Vulnerabilities. (根源是代码在不可信环境执行)
由于web浏览器确保了iframe的安全、独立的环境，建议使用iframe构建bookmarklet，例如在iframe中加载bookmarklet通过同源策略分隔代码执行并保证DOM API的完整性.

2> Web Vulnerabilities (defense-in-depth例如CSP，指出开发者注意的anti-pattern)
①XSS. 除了验证输入并清除输出，还应提供Content Security Policy (CSP)作为抵抗XSS的第一道防线 √
②CSRF: PM应在所有的页面和表单中通过token提供CSRF保护.
defense in depth: 应用应检查所有请求的Referers和Origin headers.
若不能提供token保护，可以依赖于特定的header(如: X-CSRF-Token)来进行CSRF防御.
③Secrets in Javascript files: 在HTML或响应的JSON中不存储secret值.

3> Authorization Vulnerabilities. (logic flaws)
①Insufficient authorization: 使用简单的共享模型，每个凭证只有一个拥有者，且只有该拥有者可以更改凭证及共享列表
②Predictable Identifier: 推荐使用密码学意义上的安全随机数作为identifier.

4> User Interface Vulnerabilities
建议仿照MyLogin的方式要求用户打开新tab登录.