Usenix sec'06 - Chiasson - A Usability Study and Critique of Two Password Managers

(S) PwdHash和Password Multiplier在真实用户当时能提供期待的安全性吗？用户愿意使用这两个PM吗？
(T) 对24名参与者做用户调研测试，测试两个工具的可用性（设计十实验，让用户使用两个工具完成特定的任务（工具声称的功能）并且填写问卷。(R) 发现两种工具的可用性都不好，用户的mental model存在问题，对于是否正确完成任务的认识存在问题，且用户通常不信任这个工具，不愿把控制权交给工具。跟随USENIX'99的研究，认为糟糕的可用性会导致安全性问题。最后给出了一系列的建议。

优点：①对于可用性研究的总体介绍很好。②指出之前许多可用性研究的问题，如数不够，以及只填写问卷的缺点

问题：① 设计远程登录任务时应给用户明确的说明（是否输入④⑤）而不是作为用户认知的问题来测试
② 本文提出的①②③点用户需进行的操作与前述的①-⑤点重合，实际上由②④点包含
③ 有认知操作可能会影响用户的判断和行为，应添加limitation指出本文因许可研究目的不足

---

1. 研究问题：Hash-based PM被期待能提升安全性，但用户当时是否能保证安全性，用户是否愿意使用这些方案仍然未知，且安全性很少是用户的第一目标。
→ Why Johnny Can't Encrypt? → Usability in Security是否得到了改善？

2. 对PwdHash和Password Multiplier (P-Multiplier) 做可用性评估 (website passwords)

▲ 调研方法（可用性研究通常可以分为两类）→ 确定系统是否适合特定的用户群体和目标。

1) 分为两类：
① usability inspection methods：不涉及终端用户，需要执行者对可用性研究非常专业，包括cognitive walkthrough和heuristic evaluations等。此类方法可以找到一定的可用性问题，但无法替代真实的用户调研。一般情况下，此类方法通常用于不同阶段系统的设计
② user studies：通常用于确定设计的决策，并找到忽略的问题，包括控制型实验（用于测试特定的猜想）以及field studies（系统部署在真实场景，系统日志和interview可以用于评估可用性）。
→ 可能是用户参与到实验室设计的n个任务中，也可能是以突发奇想的方式观察用户的行为，并找到其中出乎意料的问题。

2) 特定的要求与规范（揭示用户在完成特定任务时遇到的困难）
① 任务应可以反映出真实的应用场景，部署的环境应根据技术细节和给定的任务尽可能符合真实的场景。
② 若希望用户不需要专人训练就可以使用系统，则对系统的训练不应包含在测试中
③ observer的角色是观察用户的行为并记录具体的现象，不应影响用户的决策
可提供script保证所有参与者接收的信息是相同的
④ 应使用户觉得他们在改善系统而不是有人在评估他们的行为（防止用户已知测试的目的而迎合这些目的），例如使用think aloud.
⑤ 应选择正确的用户来调研，且应保证数量足够显示可用性问题。

▲ 权的方案（评估的是已实现的版本而不是正在改进的版本）
包含4类任务：①迁移用户的帐户，使用给定的PM ②从主要使用的计算机上登录受保护的帐户
③ 更改帐户的口令 ④ 远程（即不是自己主要使用的机器）访问帐户
共n名参与者参与一小时，完成5个相关的任务（分别完成PwdHash和P-Multiplier）

▲ 21名参与者，多数为大学中的学生（其中一名由于语言障碍被去除），未根据使用Firefox的经验选择用户（使用过Firefox的用户对于技术的经验更丰富）

▲ 调研过程

1) pre-task questionnaire：获取用户对web安全和口令的态度
① 重用口令 96%(25/26)。 ② 担忧口令的安全性 58%(15/26)
③ 选择口令的策略 易记67%(18/26)，难猜54%(14/26)，系统生成0%，与其它口令相同62%(16/26)

2) Tasks（使用两台机器完成任务）→ 不包含删除口令的任务，因为两个工具都不提供该功能
① Login In：使用工具生成的口令登录帐户。
② Migrate Pwd：使用自己的口令登录并更改为工具生成的口令。
③ Remote Login：在一个未安装插件的机器上登录网站 → P-Multiplier无此任务
④ Update Pwd：使用工具生成的口令登录并更改为另一个同样为工具生成的口令
⑤ Second Login：使用更改后的口令登录。
顺序可变。顺序固定，中间可插入其它任务
▲ 由于两个插件初始都不具有用户界面，故提供了说明书。

3) post-task questionaire：参与者比较这两个工具

▲ 数据收集

1) 通过observer的observation：提示用户think-aloud，但不提供影响用户操作的提示、每个任务通过观察可给出如下的结论：（可能造成口令泄漏）
① Successful. ② Dangerous success ③ Failed. ④ False completion（未完成但认为自己完成了） ⑤ Failed due to previous.（④⑤中）由于前置任务失败导致此任务失败

2) 接下来进行secondary measures，给参与者几个Likert-scale问题（5分）
通过16个问题的问卷（4类×4个问题）→ 乱序的，并非分组出现
一半的问题能形成对立的，避免偏见
① Perceived Security ② Comfort Level with Giving Control of Passwords to a Program
③ Perceived Ease of use ④ Perceived Necessity and Acceptance.

▲ 结果（两者在可用性方面表现得不够好）

1) 整体来看PwdHash比P-Multiplier的效果更好，但仍可能存在漏洞

2) 用户可能在未完成任务的情况下认为自己成功了（可能未完成激活但未得到反馈）

3) 通过运算Likert-scale所对应的平均值，并且通过t-tests评估两个工具的差异。
t-test表明PwdHash更容易使用且更加安全。但两者均让用户：不愿意放弃对口令的控制权，不认为需要这样的工具。

4) 在index card中，用户可能会在失败完成任务后在表格中选择任务易完成，此类结果称误差性

3. 对结果的分析与解释

1) 两个工具都存在的问题

用户的mental model与real实现之间存在偏差，包括不知道何时，也不知道如何激活一个功能，不理解激活后可维持多久，不知道是否正确完成了一个任务。

①不知道是否激活成功，希望找到一定的反馈和线索；认为一次会话只需激活一次即可，所有激活的方式都需要用户的鼠标在口令字段上，用户可能忘记误以为口令已受到保护。

②部分用户认为每次登录都会生成一个新的口令。（在认为保护时，原始口令可能更弱）

③部分可用性问题来源于网站的设计，例如无法识别接口，给出的字段名称不一致。

④用户因为某些问题而沮丧，甚至错后会选择重置口令；用户不知道真实的口令；用户不想把对口令的控制权交给一个计算机程序。

2.7 与原始PwdHash用户调研的对比

虽然比P-Multiplier好，但PwdHash仍存在较大的可用性问题。

①认为只输入一次@@就可以 ②对于是否激活的反馈感到沮丧 ③认为一次激活后就不需要激活

④在远程登录时，对是否输入@@感到困惑（这一点应在说明书中给出）

⑤部分用户抱怨完全不知道PwdHash的工作原理

3.7 对Password Multiplier的可用性的评价。

①用户在其它机器可能无权力安装软件（或插件），故可移植性不好。

②用户记忆负担较重，需记忆许多用户名以及用于修改口令的用户名后缀。

③给用户的反馈过少，导致用户的mental model与实际不一致。

④认为软件可以保护用户，可能采取更弱的行为

⑤其将PM可生成较强的口令，但一些生成口令在网站的PSM上显示medium，令用户沮丧。

⑥更改口令的方式可能令用户困惑，需记住为name或更改master password

用户可能对使用什么master password保护口令感觉不安全。

4.7 可用性问题导致的安全问题。

▲不恰当的可用性机制将导致用户选择绕过安全机制

①用户在注册、登录和更改口令时可能忘记激活导致任务失败。→当用户失败时，他们可能会尝试所有的口令，可能攻击者通过钓鱼网站或JS攻击获取大量的口令。

②用户在使用PM等安全工具时，会认为自己的口令被加强，会选择一个简单依旧的口令

4. 可用性问题。

1.7 mental model：用户对于软件是如何运行的没有很好的理解，这一点在可用性研究非常重要，但在安全社区却忽略了。

应提供一致的信息保证用户可以知道自己行为的结果，应提供及时且明确的反馈。

对本文测试的PM的建议：（本文作者的方案专注于透明度，忽略了这一点）

①当口令受到保护时应给出明显的提示。

②当激活工具时应给用户明显的提示（否则用户会不知道自己的口令是否受到了保护）

③对于现存的口令如何迁移的提示应更明显。

④当出现问题时，应给出短的易理解的和如何改正问题的提示。☆（问题可能来源于网站而非PM）

⑤有一种方法帮助用户检查自己的帐户是否受到保护。

（若可集成到网页中，可以更好地给用户反馈，例如问题来自网页还是PM）。

2.7 User Acceptance and View of Necessity

从可用性角度，让用户满意和接受是非常重要的，对于两个软件，用户不理解，因此会不信任，推荐在安装插件 告知用户保护口令的重要性，以及PM如何实现了这个目标。

▲Criteria for Security Software to be Usable

为了使安全的软件可用，用户需要

①be reliably made aware of the security tasks they must perform.

②be able to figure out how to successfully perform those tasks

③not make dangerous errors.

④be sufficiently comfortable with the interface to continue using it.

（1999 UENIX See）

⑤be able to tell when their task has been completed

⑥have sufficient feedback to accurately determine the current state of the system.

不提供反馈容易加剧用户对PM的不信任

Formative usability tests：在系统开发时发现潜在的问题并进行改正

Summative usability tests：获取性能数据评估可用性，比较不同人群中的效果。