Proc of Passwords'14 - Stajano - Password-manager friendly (PMF): Semantic annotations to improve the effectiveness of password managers

本质述：由于不同网站的口令认证细节存在差异，PM通常采用启发式的编程方法，存在许多问题，例如PM变得复杂且在某些网站中不可用；(1)本文旨在定义一种语义标识使网站为PM提供标准的接口。(A)本文采用HTML的class属性进行标识，给出34种表单下5种输入的class值，基于standford口令策略提出当口令长度超出一定阈值就不检测策略，另外还为表单提供了表单提交完成的反馈信息。(R)认为PMF对用户、网站、PM及开发者均有益，特别是使PM代码更精简更易维护。

优点：①提出了通过网站添加标识字段的方式为PM提供标准接口，使其更易实现与维护。
②对相关工作以及不同网站实现细节的差异给PM实现和应用带来的阻碍的分析较好。

问题：①本文缺少可行性分析，应提供严谨的调研和可用性分析来增加说服力
②本文方案需各个网站改动，无法兼容老版本网站
③虽然口令超出阈值就accept可增强可用性，但可能使用户选择易记忆的口令，应提供一定的约束；另外未给出推荐阈值(未说明推荐standford的20个字符)

1、不同网站要求的口令输入方式不同，使得不同的口令管理器只能启发式地实现，但此类实现易受攻击，应长期维护；本文提出一系列网站可以立即采用的语义标签。

1> 从用户与网站的角度，基于口令的认证是相对一致的，填写表单，提交服务器，服务器返回结果，细微的实现差别使PM等软件不能解析和自动识别表单：①可能存储输入错误的口令(PM不知登录是否成功)，②JS的使用可能带来许多问题 ③PM通常从表单中提取额外信息(可能错误)...

2> 研究动机
，PM的优点：增加可用性和安全性，减少手动输入且可以得到更随机的口令，可以帮助用户存储口令而无需记忆，

2、PMF语义标记(semantic markup)

1> 添加PMF语义标记到与创建、访问和管理帐户的表单，简化以下工作
①找到表单，确定表单类型或目的　　③解析口令策略 生成有效口令
②找到表单中的重要输入字段　　④检测错误。
autocomplete 实现了类似功能

2> 采用HTML中的class属性，采用pmf前缀，进而使用semantic class name语义，不同类型的表单采用不同的class标记，如Login为pmf-login(另有注册、更改口令重置口令)

3> 输入表单
①Username：输入用户名，采用pmf-username标记。
由于用户在一个网站可能有多个帐户，口令重置和修改功能可能无法得知要变更哪个用户的口令，可以使用hidden字段输入用户名(使用pmf-username标记)

④Passwords：
在注册时无此义，登录和更改时可以使用pmf-password和pmf-new-password区分

③Stay signed in：让用户在一个网站登录的复选框，pmf-stay-signed-in(PM可更改该设置，使其默认不生效，保护用户隐私安全)

④Hidden inputs：由于PM仿照用户行为，用户不可见hidden表单，故不提供该标记

4> 口令创建策略
随机生成的口令可能不满足网站的口令策略，但当前PM无法直接读取网站策略
▲提出网站添加一个可供机器读取的口令策略的JSON格式的单即可
→认为网站开发较烦琐，进一步提出：→ 2014 Standford password policy
▲当口令长度超过一定阈值直接accept，若小于该值，则需要符合网站口令策略。

5> Errors
为PM提供表单提交试的反馈信息 pmf-error.