SOUPS'06-ka-Ping Yee-Passpet: Convenient Password Management and Phishing Protection

(S)基于四种技术改善口令认证方案，减轻用户记忆负担，避免用户被欺骗遭受钓鱼攻击，(A)提出口令管理方案应满足的16条可用性和安全性目标，给出具体设计，使用password hashing从一个可记忆的口令生成网站唯一的口令，petnames让用户为网站自定义别名帮助识别假冒网站，password strengthening使用password multiplier的技术抗击离线字典攻击，使用UI customization防止故手通过用户界面欺骗用户，并使用remote storage提供额外的安全性保护。(R)对Passpet进行了安全性和可用性分析，并给出了未来的评估方法。

优点：①对方案的设计的描述较好，分为用户的角度和实现原理的角度 ②对未来评估计划的描述较好

问题：①没有提供较好的适应网站口令策略的方法。

②对master secret等待时间越长，强度越高的描述不清晰，应给出更多解释

③可用性和安全性目标不够合理，基本都是自己方案的特征，例如⑫，本文方案也引入了remote storage；另外，没有与商用PM的比较。

---

1. Passpet采用4种技术改善方案，减轻记忆负担，避免被欺骗遭受钓鱼攻击。(更加安全方便)

1> password hashing：通过1个可记忆的口令做哈希得到每个帐户的口令。

2> petnames：用户为自己的标签有助于用户识别假冒的网站(类似于电话本中定义的别名)

3> password strengthing：抵抗字典攻击。

4> UI customization：防止故意采用用户界面欺骗

2、除了方案的设计，还引入了secure interaction design，可应用于其它领域

1> 使用用户指定的标签用于口令合并

2> 在用户输入口令时可评估字典攻击的时间

3> 用户可自己置password strengthing的强度。

4> 可将安全工具与不同用户的帐户相关联

5> 定制化或自己安全工具的按钮防止被欺骗

3. 问题与目标

1> Usability Goals

①提高登录网站的便利性 ②可以与现存的表单和网站兼容

③允许使用工具完成帐户迁移 ④允许用户修改单独1个网站的口令

⑤允许用户在多台机器上实现登录(表达随时随地) ⑥用户只需记忆1个secret

⑦允许用户更改master secret

2> Security Goals.

⑧每个网站的口令唯一 ⑨用户选择的secret可抵抗离线字典攻击

⑩可以适应性地随计算机速度提升增强安全级别 ⑪避免强口令的长期存储

⑫避免引入集中式的依赖。 ⑬抵抗基于伪造表单的攻击

⑭抵抗基于模仿UI的攻击 ⑮帮助用户识别认证网站 ⑯改变在网页输入口令的习惯。

4. 设计

▲从用户使用的角度(Passpet Firefox extension)

1> 设置，初始化

① 用户在表单中输入master address， username@hostname (代表Passpet server)

② 从一组动物图标中选择一个随机图标，给动物一个随机名称，名称和图标代表Passpet的角色。用于与用户交互(当名称、label不同时，生成的口令不同)

③ 用户选择master secret，且Passpet会给出改好改破该master secret所需要的离线攻击的时间(年份+机器+攻击时间)

2> Everyday Use (Passpet为toolbar button).

① 浏览器启动时，Passpet处于未激活状态

② 点击图标并输入master secret后可激活，persona会维护master secret

③ 当焦点在口令字段，且点击了Passpet button，口令字段会被自动填充

3> Managing Relationships.

Passpet会为用户展示site label，若未赋予label则为unknown。

对于non-SSL site会给出标识与非SSL网站的提示。

▲当label重复时，给出具体的提示信息，供用户选择

▲用户在注册和迁移帐户后都可以利用Passpet自动填充口令，当需要修改passpet生成的口令时，可以通过修改site_label来更改一个网站的口令。

▲生成一个新的Passpet persona，并逐步从旧的Passpet persona把各个网站的口令迁移过来，当迁移完成时，旧的key可以丢弃。

▲本方案的实现机制。

1> Passpet的icon和label是多样的，不同用户之间一般不同

2> Site Identification.

Passpet将用户输入的site label与site identifier相关联 → (root_key, field_name, field_value)

▲对于SSL网站，root_key为根CA公钥的指纹。

field_name和field_value为SSL证书中的信息名称及字段值，按选择顺序依次为O, CN

因此只有另一个网站可以由相同CA签署了同一证书才能显示出同一个label

▲对于非SSL网站，root_key为空，field_name为D(代表domain name)，site identifier的值为(n+1)级的domain，例：www.subdomain.example.com → example.com.

www.gov.mb.ca → gov.mb.ca.

Passpet使用两级的TLD来确定值，site identifier指购买或拥有域名的特定级别的域名，此时多个site可能会得到相同的值(如：www.example.net example.net)

▲Site label可以使用用户自身的语言。

3> Password Generation (采用Password Multiplier的思路)

首次输入master secret后，计算 $V=H^{k_1}$(master_address||"\0"||master_secret)，缓存在计算机中

接下来计算 $P=H^{k_2}(site\_label||"\backslash 0"||master\_secret||"\backslash 0"||V)$

▲master_address 用于防止彩虹表攻击(即预计算的攻击).

▲如何满足网站的口令策略? Passpet 生成的口令至少包含一个数字、一个小写字母和一个大写字母. 若需要特殊符号, 则需点击特定的按钮键.

▲将 P 的 base62 位表示形式的数字的 n 个最低有效位(在 [0-9a-zA-Z] 中作为口令. 若不包含上述3种类型的字符, 则执行 $P'=H(P)$, $P''=H(H(P))\cdots$

▲master secret 允许 Unicode 字符, 此时 hash 运算中, master secret 和 site label 编码为 UTF8

4>Variable Strengthening.

k_1 值可以由用户指定并修改, k_2 是固定的. 在 Setup 过程中, Passpet 可以通过 k_1 和敌手的算力计算成功进行字典攻击的平均时间.

5>Local Storage

当 Passet persona 被映射时, 将维护一个 site_list 的列表, 存储加密且 MAC 的文件中.

site_label_file = E_{W_1}(site_label_list)||M_{W_2}(site_label_list)

其中: $W=H^{k_2}$(master_secret||"\0"||V), W_1=high 128 bits of W, W_2=low 128bits of W.

▲每个 persona 维护一个 cache, 包括 master address, index, V 和 site_label_file.

6>Remote Storage

由于 master_address 和 master_secret 由用户输入, k_2 固定, 需要使用远程服务器存储 k_1 和 site label file 的副本.

▲master_address 为邮件地址格式相同, 允许用户选择存储文件的位置, 通过 SRP 认证.

维护的内容为: (username, index, k_1, salt, verifier, site_label_file).

index: 记录文件的索引; salt, verifier: 用于SRP协议

▲支持两个未认证的命令.

① create(username, k_1, salt, verifier) 若不存在文件则创建空文件并返回 index

② list(username) 对于给定的 username, 返回 (index, k_1) 的列表.

▲在认证会话中, 客户端发送 (username, index), 获取 record, 接下来进行 SRP 协议, server 可接受如下的命令.

① delete() 服务器删除记录. ② read() 服务器返回 record 中的 site_label_file.

③ write(old_mac, site_label_file). 若文件的 MAC 与 old_mac 相关, 则 server 自动替换文件, 否则返回错误信息.

▲流程: ① 初始化 persona 阶段, 用户选择新的 master secret, 并在 strengthening 过程中选择 k_1.(得以)

② Passpet 使用 SRP 的 salt 和 verifier, 使用 username 作为 SRP 用户名, W 作为口令. 发送 create 命令给 server, 获得 index, persona 初始化完成

③ 激活过程中, 用户根据缓存的 V 和输入的 master_secret 计算 W 并与 server 认证, 发送 read 命令获取 site_label_file 并检查 file 的 MAC 值.

▲当在另一台机器上使用 Passpet 时, 用户可直接创建未初始化的 persona, 在激活过程中, Passpet 执行 list 并对所有的 (index, k_1) 依次登录尝试, 直到成功. 接下来缓存 index 和 V 并完成初始化.

▲当 site_label 发生改变时, Passpet 加密更新的 site label 并通过 write 命令写回 storage server. 由于数据库较小(一个网站占 40-80 字节), 故可以在一次传输中完成.

5. 安全性分析 (phishing 代表模仿网站, spoofing 代表模仿用户按口令的一部分)

1>Dictionary attacks → user's passwords.

根据 Password Multiplier 的不同情况的分析.

但本文 Passpet 需通过 SRP 协议访问 storage server, 即使有 V 和部分网站口令, 敌手缺少 master secret 无法访问 storage server. (若可得文件, 也可执行离线猜测攻击, 与得到网站口令相同)

通过与 password Multiplier 类似的分析, 表明 Passpet 在 k_1 相等时可提供与 Password Multiplier 同等强度的抵抗离线字典攻击的能力

2>Attacks on Storage Server → site label file

该文件包括用户的浏览习惯, 可能带来隐私风险,

使用 MAC, 保证在未知 K 的情况无法对文件进行有效的修改.

使 SRP 做认证, 防止未授权的敌手删除或破坏文件造成 DoS 攻击

▲当 storage server 不可用时, 用户可以依靠 site_label_file 的副本和缓存的 V 作, 但文件可能非最新

3>Phishing Attacks on Site Passwords.

▲在 Passpet 中, 用户通过点击按钮生成网站口令, 不需要手动输入, 无法通过这种方式给敌手

故除非让 Passpet 生成一个口令, 包含以下三种情况 (十分很难完成)

① 更改 site label file 使得 site label 与敌手的 SSL 证书一致

② 获取一个与 site label 一致的数字证书 ③ 欺骗用户生成与敌手网站一致的 site label.

4>Spoofing Attacks on the Master Password. 敌手可能通过伪造 Passpet icon.

对抗措施: ① icon 是随机选取的, 用户之间不同, 敌手不太可能产生一样的

为用户与 Passpet Tool 提供验证的信任关联 ② Passpet 在请求 master secret 时使用自定义的别名 site label, 敌手难以产生相同的

③ 用户只在点击按钮后输入一次 master secret, 不会在外部提示下输入.

5>Cross-Site Issues.: 认为 label 代表当前页而非目标页更安全, 防止跳转注入 → 与PwdHash类似

6. 可用性分析.

1>最大的优势为在一次会话中只需输入一次 master secret, 后续只需点击按钮即可登录.

2>另外生成新口令很方便 (输入 site label 即可), 所以更改口令也很方便.