2017-Jang-Forensically-Sound Analysis of Security Risks of using Local Password Managers

优点：① 对论文研究背景较好，例如可记忆记住的什么因素

② 从使用规范和实际设计的角度研究本地PM的安全性

问题：① 对于RAM为1GB的场景，应检查PM应用推荐的硬件的要求

② 可对其它功能、OS和PM进一步检测和分析，并通过用户调研了解用户的PM使用习惯

③ 可建立威胁模型，对攻击者能力目标建模

▲ Forensically-Sound：指数字证据是以法律可接受的方式收集、分析、处理和存储的，且有合理的证据证明，可以保证数字证据在调查过程中没有被破坏或毁坏（无论是故意破坏还是意外）

▲ 研究背景

1> 能否记忆口令取决于以下什么因素：

① 口令复杂度：长度、字符、大小写以及是否包含字典单词 ③ 口令匹配：口令与帐户的对应

② 使用频率：使用频率低的口令较难记忆 ④ 更新效率：最佳实践推荐定期更新并检查口令历史

2> 为了对抗记忆和匹配口令，用户使用许多不安全的方法，口令管理器是权衡了安全性和可用性的最好方法之一，逐渐成为个体、中小型企业的最佳实践之一，但仍可能成为单点故障

3> 口令管理器可以分为三类：① 基于云，例如LastPass和1Password，需安装在本地，但可以同步到云，用户可以随时访问。② 基于浏览器，Chrome、Firefox等都提供了内置的PM，加密的方式根据OS而存在不同。③ 本地PM，本文分析的均为开源，允许导出数据库。

4> 过去研究更多讨论基于云和浏览器的PM的安全性，本文关注本地PM的安全性，关注实现的细节，不关注加密算法，研究的本地PM包括 KeePass 2.28、RoboForm 7.9.12、Password Safe 3.35.1

▲ 研究问题：采用forensic研究方法探索本地PM的安全性（之前的研究多通过理论或模拟攻击）

▲ 研究设计与方法

1> 漏洞存在的可能：PM在运行中，操作可能导致瞬态存储的内容发生改变，随后在OS交换内存时使得持久存储发生改变，通常出现在系统负载较大时，此时交换到外部存储甚至是明文

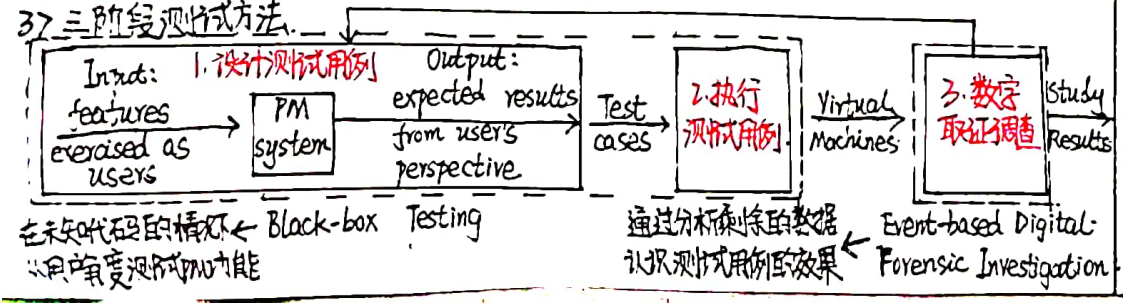
本文使用EnCase工具完成取证分析

2> 取证过程的需求：

① 从数据源获取的证据不应由调研人员的活动而被篡改。

② 证据应该是可重复的，需进行多次重复的实验

3> 三阶段测试方法：



① 对3个PM设计了19个测试用例（包括正常使用和特定的场景中），测试用户用于身份认证的不同方法，导出、打印，复制/粘贴，自动完成（auto-complete）和卸载。

若在其中的测试用例中有新发现，则进一步设计测试用例。

② 第2阶段包含创建虚拟机和执行测试用例。

首先创建 baseline 的VM，然后保存到独立的文件夹中，防止被意外更改数据

③ 根据基于事件的研究框架，使用EnCase进行阶段3的取证检测。

(i) System Preservation and Documentation，验证收集取证据是否被篡改或有错误并记录

(ii) System Evidence Searching and Documentation，包括4步骤，目标识别；数据提取与相关解释；数据比较（与预期结果对比）；知识更新（认识漏洞及漏洞产生的原因）

(iii) Digital Evidence Reconstruction and Documentation：测试之前的假设，即将一个测试用例在新克隆的虚拟机上重复执行，确保结果可重复可信赖。

▲ 实验结果

1> KeePass

① K1：当PC内存容量较小时，主口令可能会存储在页文件中，通过EnCase搜索可找到明文主口令

② K5, K6：导出时不需要重新输入主口令，且导出时以明文形式显示，且删除后仍存在于Recycle Bin文件

③ K7：在打印时将未加密的数据库输出到临时文件中，若正确打印，文件被删除；否则，文件仍存在

2> Password Safe

① P3：不恰当的关闭会导致存储在粘贴板上的隐私数据泄漏（存储在页文件中）

一种方法是采用RoboForm方法，使用safe-folder临时存储加密的复制到粘贴板的数据。

3> RoboForm：与KeePass的K类似。

▲ 讨论

1> 尽管有的漏洞局限于特定的场景，但这些场景实际中很常见，例如关闭PM就关机。使用户处于危险当中

2> 可进行的未来工作

进一步扩展测试用例的规模，例如其它的功能，其它类别的PM和OS。