USBKey 远程劫持技术分析①

潘建军[1]，王加阳[1]，罗海波[2]

[1](中南大学 信息科学与工程学院，长沙 410084)
[2](中南大学 湘雅三医院，长沙 410083)

摘　要：身份认证技术对于开放环境中的信息系统具有极其重要的作用，因此人们在实际应用中不断探索,研发了一系列身份认证技术。伴随密码学，智能卡等技术的发展和成熟，一种结合现代密码学、智能芯片、USB 技术的认证方式终于出现在我们面前，这就是基于 PKI（公共密钥基础设施）的 USBKey 认证方式。寻求一种基于 API Hook 的认证安全检测技术，用于验证基于 USBKey 的身份认证是否存在缺陷，并且提出相应的改进方案。

关键字：CrytoAPI；CrypSPI；API Hook；身份认证；USBKey

USBKey Remote Hijacking Technical Analysis

PAN Jian-Jun[1], WANG Jia-Yang[1], LUO Hai-Bo[2]

[1](School of Information Science and Engineering, Central South University, Changsha 410083, China)
[2](Xiangya Third Hospital, Central South University, Changsha 410083, China)

Abstract: Authentication technology plays an important role in the open environment for information system, so people continue to explore practical applications, to find a series of authentication technology. With cryptography, smart card technology developing and maturing, a combination of modern cryptography, smart chip, USB authentication technology has finally appeared in front of us. In this paper we find a detection technique based on API Hook, used to verify the identity authentication based on USBKey.

Key word: CryptoAPI; CryptoSPI; API Hook; authentication technology; UsbKey

身份认证对于开放环境中的信息系统具有极其重要的作用，是网络安全服务及其他服务的安全基础。近几年来，随着 Internet 服务，尤其是电子商务、电子政务、网上支付等业务的发展，信息安全问题已经日益成为人们关注的热点问题，特别是在网络国际化,全球互联的大背景下，信息安全已经成为事关国家安全，社会稳定的关键问题。

目前主要的身份认证技术概括起来有四类:基于口令的认证方式、基于硬件令牌的认证方式、基于生物特征的认证方式以及基于密码技术的认证方式。出于安全等因素的考虑，大多数认证系统都综合采用了其中的多个因素来确认一个对象。

身份认证对于开放环境中的信息系统具有极其重要的作用，是网络安全服务及其他服务的安全基础。近几年来，随着 Internet 服务，尤其是电子商务、电子政务、网上支付等业务的发展，信息安全问题已经日益成为人们关注的热点问题，特别是在网络国际化,全球互联的大背景下，信息安全已经成为事关国家安全，社会稳定的关键问题。

目前主要的身份认证技术概括起来有四类:基于口令的认证方式、基于硬件令牌的认证方式、基于生物特征的认证方式以及基于密码技术的认证方式。出于安全等因素的考虑，大多数认证系统都综合采用了其中的多个因素来确认一个对象。

2　API Hook 技术

API Hook 是一种高级编程技术，很多编程问题的实现和解决都需要用到这种技术。本文定义的 API 是

① 基金项目:安徽省教育厅自然科学基金(2005KJ004ZD);
收稿时间:2011-02-24;收到修改稿时间:2011-04-11

一种广义的 API,既包括 Windows SDK ,也包括中断向量表 IDT,系统服务描述表 SSDT 等。API Hook 技术是一种改变和扩充现有程序功能的有效手段，它通过拦截程序调用的函数来实现[1]。

3 USBKey远程劫持技术实现

本文讨论的 USB 远程劫持技术主要通过用户层(ring3)API Hook 技术实现。现在使用的 USBKey 安全认证技术主要是建立在公开密钥基础设施 PKI 之上的。PKI 通过延伸本地用户的接口为各种应用提供安全服务，包括身份认证、加/解密、签名服务等。目前应用程序开发接口国际标准主要有 CDSA,RSA PKSC#11 和 Mircrosoft CryptoAPI 等[2]。由微软视窗操作系统已经成为实际上的客户机操作系统主流，因此 CryptoAPI 接口也成为最引人瞩目技术体系之一。因此实施 USBKey 远程劫持的最重要的基础就是通过分析，了解密码服务供应商 CSP 的工作流程，从而对 CryptoAPI 进行挂钩，以达到劫持 USBKey 的目的。

3.1 CryptoAPI 和 CSP

应用开发者可以使用 Microsoft CryptoAPI 提供的加密功能开发基于 Windows 的应用程序，调用 CryptoAPI 不需要了解操作系统底层的工作原理。前文提到密码服务是由 CSP 模块实现的,但 CryptoAP I 函数并不直接与 CSP 通信。操作系统会过滤这些函数调用,再通过 CryptoSP I 函数传递给相应的 CSP。操作系统 Cryp toAP I 和 CryptoSP I 的关系如图 1 所示。在这种架构里，多个应用程序可以安全地使用不同 CSP 提供的加密签名服务。CSP 是一个功能上相互间独立的模块,例如有的 CSP 运行时需要用户输入个人身份认证 PIN (Personal Identification Number) ,有的 CSP 基于智能卡等专用密码设备。可见 CSP 的安全性不依赖于操作系统,而是 CSP 本身的设计参量,从而允许同一个应用程序能适合在不同的安全环境下运行。

3.2 USBKey 签名工作原理

USBKey 的 CSP 模块将公/私密钥对保存在称为容器的数据对象里面。每个客户持有一个容器用来存放他的一个交换密钥对和一个签名密钥对。公私钥对和容器都需要永久保存在存储介质中，因此属于永久数据对象[3]。

CSP 工作流程为：

(1) 获取 CSP 名称和 CSP 类型。

CSP 都有一个名字和一个类型，每个名字都是唯一的，而类型则不是，不同的 CSP 可能类型一样。

(2) 应用程序获取 CSP 句柄。

使用 CryptoAPI 中的函数 CryptAcquireContext,参数为 CSP 的名称，类型参数以及密钥容器参数，该函数返回一个指向被选择 CSP 的句柄。

(3) 获取 CSP 哈希句柄 。

使用 CryptoAPI 中的函数 CryptCreateHash 创建散列值哈希句柄，该句柄用于对待签名数据进行摘要，以获取长度一致的待签名数据。

(4) 计算待签名数据 HASH 值。

使用 CryptoAPI 中的函数 CryptHashData 对待签名数据进行 HASH 计算，操作分两次进行，第一次调用该函数时，需讲待签名数据值设为 NULL,通过函数参数返回数据长度。以该长度为大小，分配内存空间。然后继续调用该函数以获得哈希值结果。

(5) 对哈希数据进行签名并返回。

使用 CryptoAPI 中的函数 CryptSignHash 对待签名数据摘要信息进行签名并返回签名结果和状态[4][5]。

至此，经过以上操作，我们已经成功的使用 USBKey 的 CSP 对关键数据进行签名。

3.3 USBKey 远程劫持。

USBKey 进行客户身份认证的基本原理基本相同，因此我们选择网上银行系统作为本次分析的场景。

USBKey 远程劫持定义：远程用户在不具有 USBKey 硬件设备的条件下通过软件破解的方式，从而达到通过服务器的身份认证的条件。

由应用程序开发原理可知 USBKey 对于关键数据的签名一般由签名控件发起。安装于浏览器签名控件会读取待签名数据，并通过选择合适的证书获取该证书 CSP 名称及类型。因此我们可以基于上文对 CSP 软件工作流程的分析，使用 API Hook 技术对签名过程进行模拟。 以下为我们假设实验网络环境如下图 1 所示：

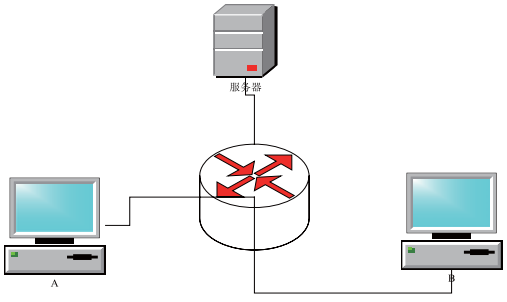

图 1 网络环境图

假设 A 用户为持有 USBKey 的合法用户，B 为远程劫持者。

USBKey 远程劫持软件由服务器端和客户端两部

分构成，服务器端主要安装于被劫持用户计算机中，客户端通过 CreateRemoteThread 方式，远程注入到 B 用户的 IE 浏览器中。

服务器和客户端初始化工作流程：

(1) 服务器端枚举 A 机器所安装的所有有效证书。具体过程如下：

① 使用 CertOpenStore 打开 Windows 证书存储机构。

② 调用 CertEnumCertificationInStore 枚举存储机构里安装的有效证书。

③ 使用 CertGetCertificationContextProperty 获取证书属性。包括 CSP 名称，密钥容器名称，证书类型等。

④ 应用程序服务器端将证书文件保存为 XML 格式并在 80 端口进行监听。

(2) 客户端初始化主要用于连接服务器端，并且从服务器端获取证书信息。主要工作流程如下：

① 客户端根据 IP 地址获取向 A 客户机的 80 端口进行连接。

② 若连接成功则接受服务器发送的证书数据，并安装到计算机中。

③ 初始化工作完成，等待进一步操作。

证书获取的具体情况如图 2 所示：

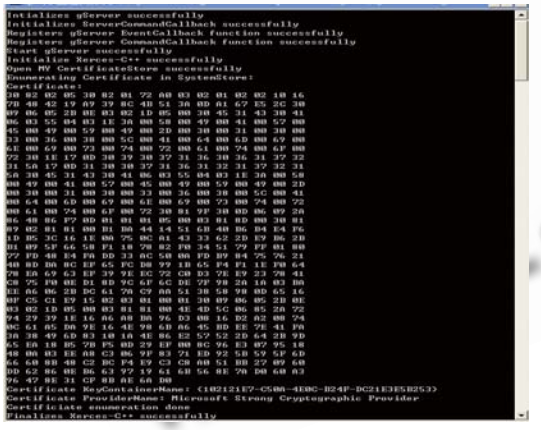

图 2　软件服务端显示结果

客户端 B 劫持服务端 A 的 USBKey 具体工作流程如下：

(1) 客户端与服务端保持已经建立的连接。

(2) 若服务器端 USBKey 已经插入计算机 USB 接口，则服务端向客户端发送上线通知。

(3) 服务器端使用预先获得的账号和静态密码进行登录（实际应用中可以使用内核级键盘过滤程序突破网上银行系统客户端密码保护控件的保护）。

(4) 转账操作过程需要静态密码和 USBKey 对关键数据进行授权和签名以验证用户是否为具有权限的合法用户。根据以上分析的 USBKey 的 CSP 工作原理，签名控件会首先调用 CryptoAPI 函数 CryptAcquireContext，此步骤指定 CSP 名称及类型，该函数继续调用 CryptoSPI 的函数 CPAcquireContext。我们在此过程中使用 API Hook 技术对 CPAcquireContext 进行挂钩，并且比较 CSP 名称，若为指定服务端 CSP 的名称，则将该请求通过网络连接发送到 B 机器，获得该机器上的一个 CSP 句柄，并且返回客户端。

(5) 若成功获取 CSP 句柄，则继续进行签名数据的以后步骤，并以 CSP 句柄为关键字，对以后所有的 CryptoSPI 的调用进行 Hook，将所需要的参数传递给远程机器进行运算，并且返回结果和状态。

(6) 当进行 CPSignHash 步骤时，远程服务器所在机器将弹出 PIN 码输入窗口，我们输入事先准备好的 PIN 码。

(7) 若签名成功，则服务器会通过本次交易请求，若失败，则交易失败。

经测试，基于 API Hook 的 USBKey 远程劫持技术，对现在使用的大多数 USBKey 都成功的发挥了作用。

4　结语

通过以上问题分析和实现，我们得出这样的结论：现在广泛使用的基于 USBKey 的身份认证机制，由于 Key 厂商本身的疏忽以及系统设计上存在的缺陷仍然不能完全保证信息系统的安全。因此软件开发商在开发应用系统，尤其是涉及人们核心利益的应用系统过程中，应该充分考虑现在 USBKey 本身固有的问题，适当采用比较完善的客户端保护，从而为信息系统合法用户的权益提供多一层的保护。

参考文献

1 程建明.网络金融.北京:清华大学出版社, 2005.

2 周天虹.浅析网上银行业务的安全规划和实施措施.信息网络安全,2007.

3 Pietrek M. Peering Inside the PE:A Tour of the Win32Portable Executable File Format.Microsoft Systems Journal, 1994,9: 27–30.

4 Richter J. Windows 高级编程指南.王书洪,刘光明译.北京:清华大学出版社,1999.

5 Richter J, Nasarre C. Windows Via C/C++. Microsoft Press, 2008.9,1:98–131.