Vulnerability of DPAPI data protection in Win2K, Win2K3, Windows Server 2008, and Windows Server 2012

Passcape Software

1 漏洞的简要描述

1.1 问题所在

我们之前的一篇文章描述了DPAPI系统的运行原理，其卓越的可靠性、功能价值和抗破解性。不久前，我们偶尔发现，Windows 2003中的一些DPAPI blobs有解密问题。在确定了问题的原因后，发现了一个相当有趣的DPAPI安全漏洞，它可以在所有的服务器操作系统中重现，从Win2K开始，到Windows Server 2012结束。

简而言之，它的本质是以下几点。默认情况下，所有具有交互式登录权限的域用户的主密钥（内置账户除外）都是在Windows 2000兼容模式下创建的；因此，用DPAPI加密的数据解密不需要所有者的登录密码。

1.2 受影响的软件

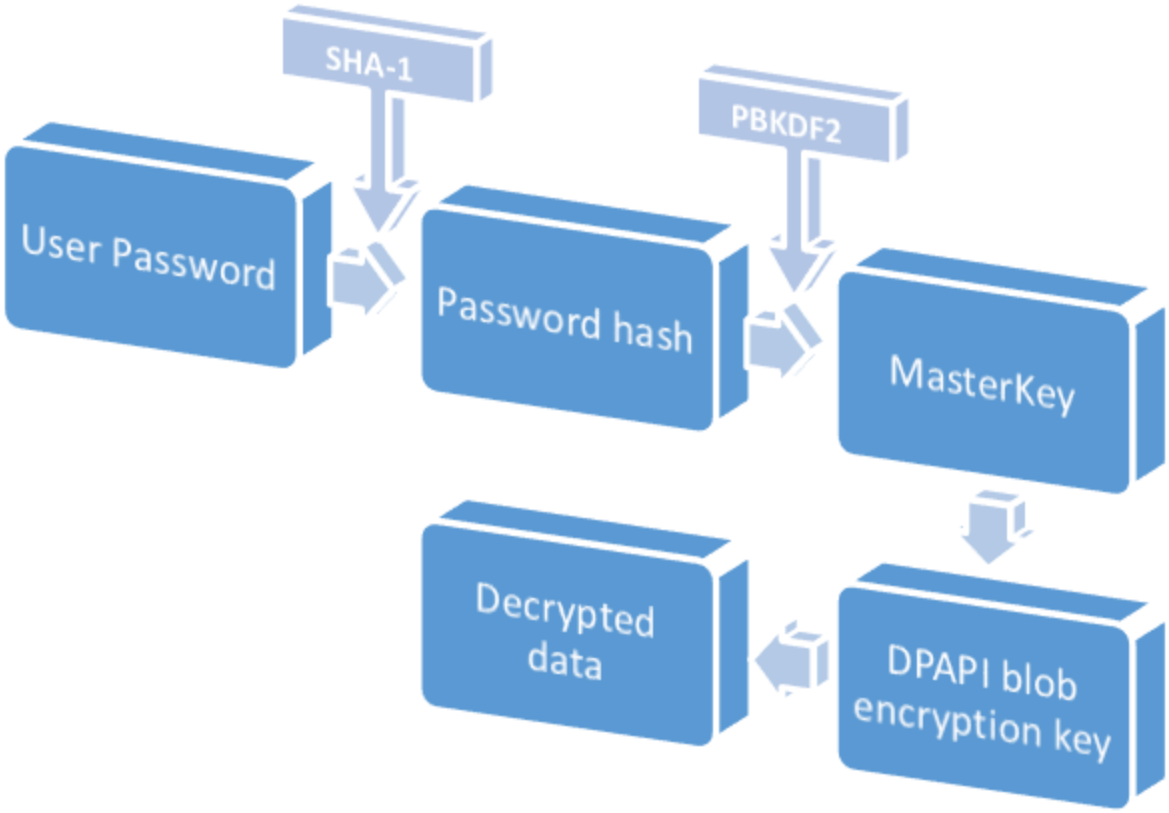
这意味着，任何管理员或任何其他对服务器有物理访问权的用户都可以解密脆弱用户账户的以下个人数据：

- 流行浏览器中的密码和表格自动完成数据。Internet Explorer, Google Chrome, Opera Browser等。
- Outlook、Windows Mail、Windows Live Mail等的电子邮件账户密码。
- Windows FTP管理器的账户密码
- 共享文件夹和资源的访问密码
- 无线网络的密钥和密码
- Windows CardSpace中的加密密钥
- Windows Vault中的加密密钥
- 远程桌面连接密码
- .NET Passport的密码
- Windows Live ID的个人数据
- 加密文件系统（EFS）中的私钥
- S-MIME邮件中的加密密钥
- 用户的证书
- 互联网信息服务中的私人数据
- EAP/TLS和802.1x认证
- 凭证管理器中的网络密码
- 任何应用程序中的个人数据，通过Windows API函数CryptProtectData进行编程保护，如Skype、Windows权利管理服务、Windows Media、MSN信使、Google Talk等。

2 技术细节

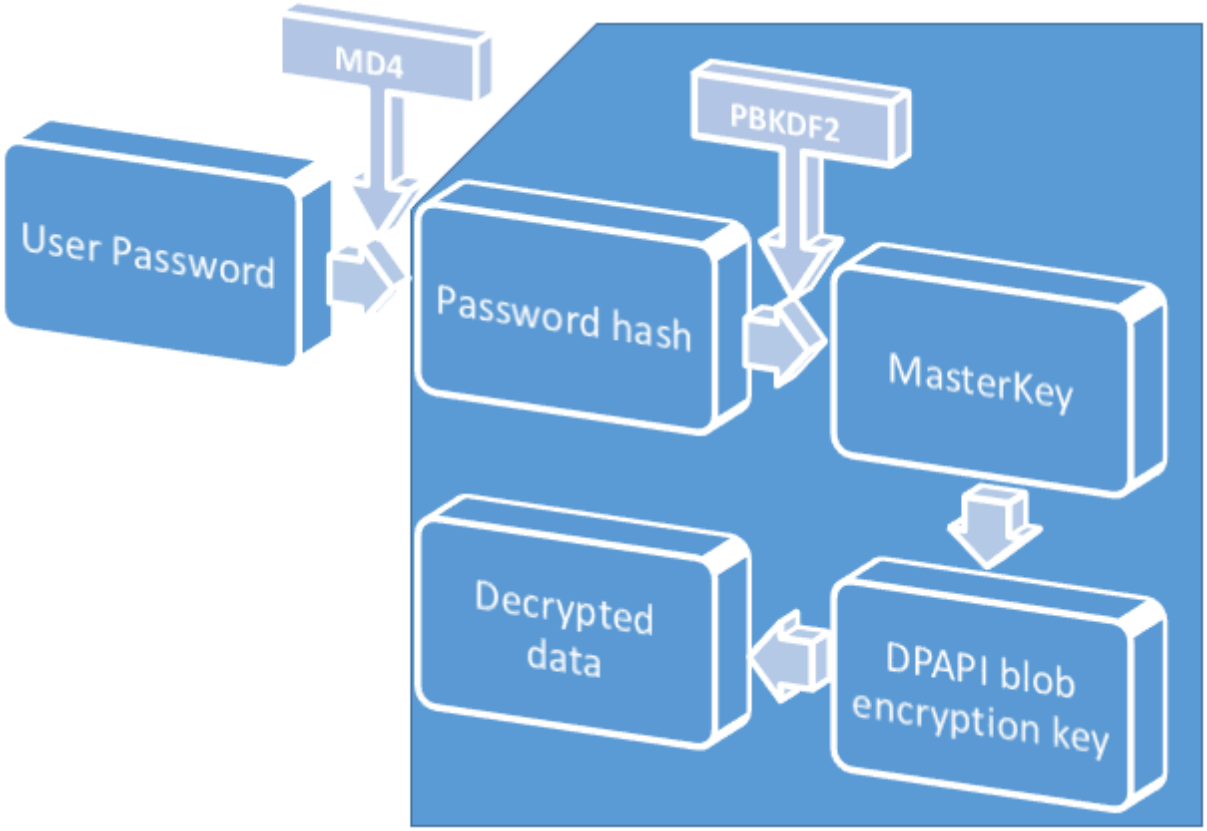
2.1 在Windows XP和更高的操作系统中的DPAPI加密

这就是用DPAPI加密的私人数据的解密过程(省略了一些细节)。最初，数据所有者的登录密码通过SHA-1得到密码哈希值；然后，密码哈希值和所有者的SID被送入PBKDF2函数。在输出端，产生一个预密钥，它参与主密钥的解密。解密后的主密钥反过来被用于解密实际的DPAPI blobs。以下是图表中的情况：



2.2 操作系统Windows 2000中的DPAPI加密

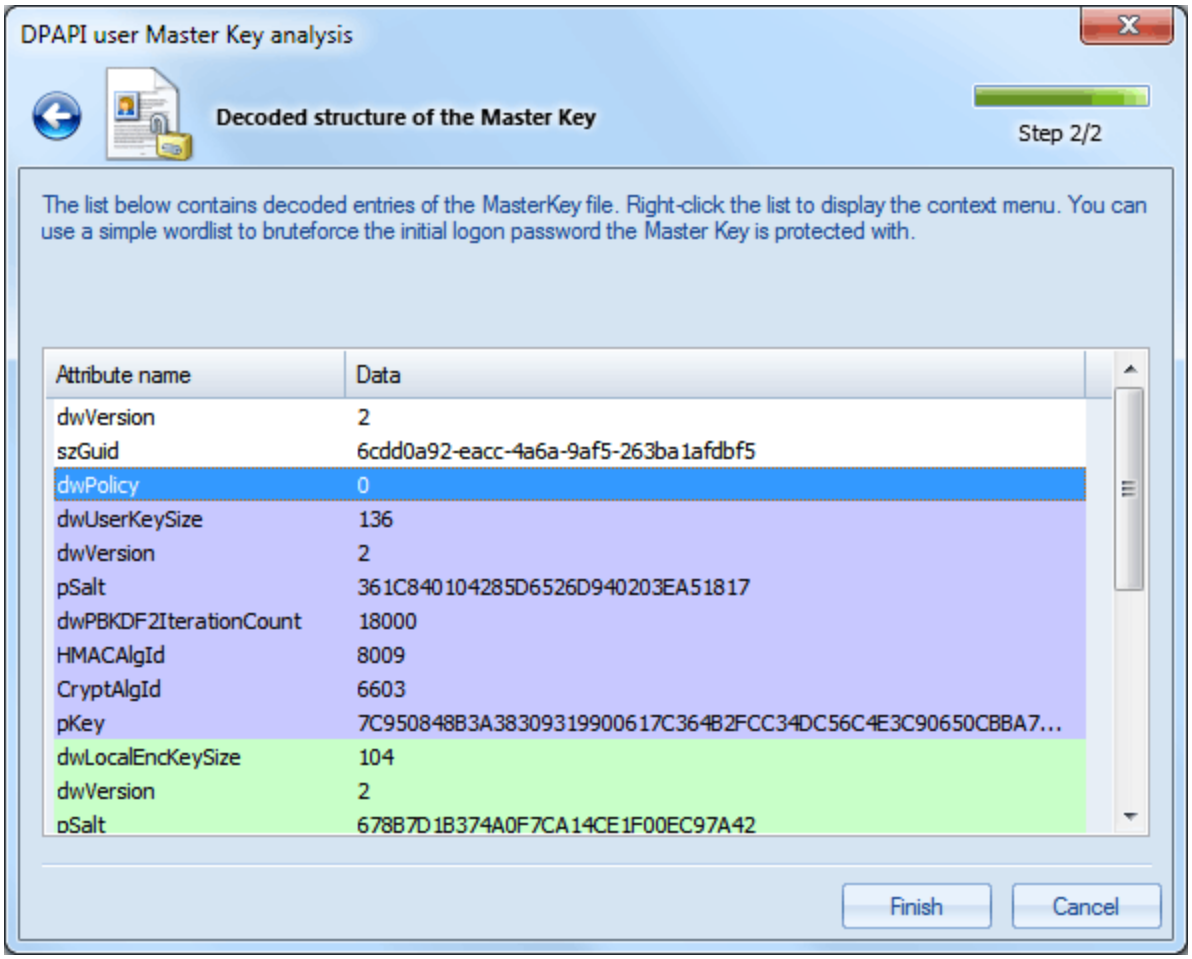
在Windows 2000中使用的DPAPI的第一个实现有其他主密钥加密算法。但这并不是使它变得极其脆弱的原因；而是为了获得用户的密码散列，它使用了MD4散列函数而不是SHA-1。它看起来是这样的：



有趣的是，同样的hash函数也被用于验证用户的登录密码，这些密码的散列值被存储在SAM注册表或活动目录中（对于服务器操作系统）。因此，用户的明文密码对于解密主密钥并不是真正必要的。只需从SAM或NTDS.DIT中获取相应用户的现有MD4哈希值并将其作为数据输入即可。其余的都已经知道了。

2.3 Windows 2003、2008、2012服务器操作系统中的DPAPI加密缺陷

使用哪种算法- SHA-1或MD4 -在主密钥的头中指定。dwPolicy标志位的第4位表示主密钥使用SHA-1算法。在Windows服务器操作系统中，新创建的具有交互式登录权限的用户在默认情况下没有设置此标志。分别地，解密他们的私有数据不需要登录密码。



3 利用漏洞

3.1 在Windows 2012服务器中创建交互式域用户

让我们从理论转向实践，尝试向Windows Server 2012域添加一个新用户，然后创建一些DPAPI机密，然后在不需要用户登录密码的情况下脱机解密。

打开“Active Directory用户和计算机”控制台，创建一个名为Test的新域用户。授予该用户交互式登录权限。为此，只需将该用户添加到本地管理员组。

3.2 为新用户创建主密钥和DPAPI秘钥

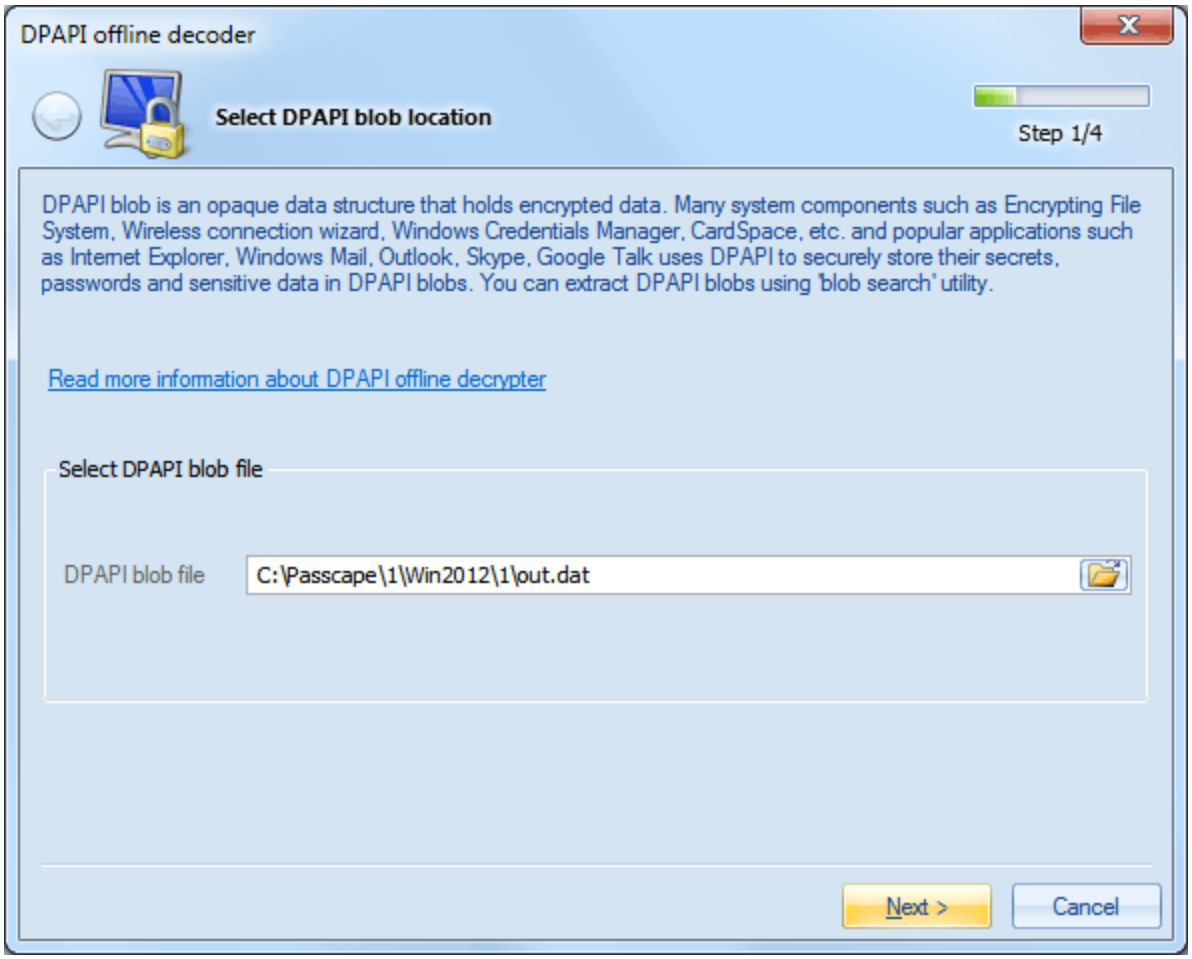
现在我们需要注销系统，然后在这个帐户下登录。新账户还没有万能钥匙。它将在第一次调用CryptProtectData函数时创建。我们将通过强制调用各自的函数来加快这一过程。为此目的，我们有一个同名的实用程序 CryptProtectData.exe，它简单地调用API函数CryptProtectData与命令行参数(实用程序源代码 可以在网站上找到)。使用以下参数启动:CryptProtectData mysupersecret out.dat。在输出中，我们将得到out.dat文件，其中包含一个DPAPI blob，其中包含我们加密的文本(mysupersecret)。

因此，主密钥已经创建好，并保存在C:\Users\test\AppData\Roaming\Microsoft\Protect\\
其中 -所有者的SID。我们仍然需要它，所以要么记住它，要么复制整个目录。
- DPAPI主键名;例如 6cdd0a92-eacc-4a6a-9af5-263ba1afdbf5

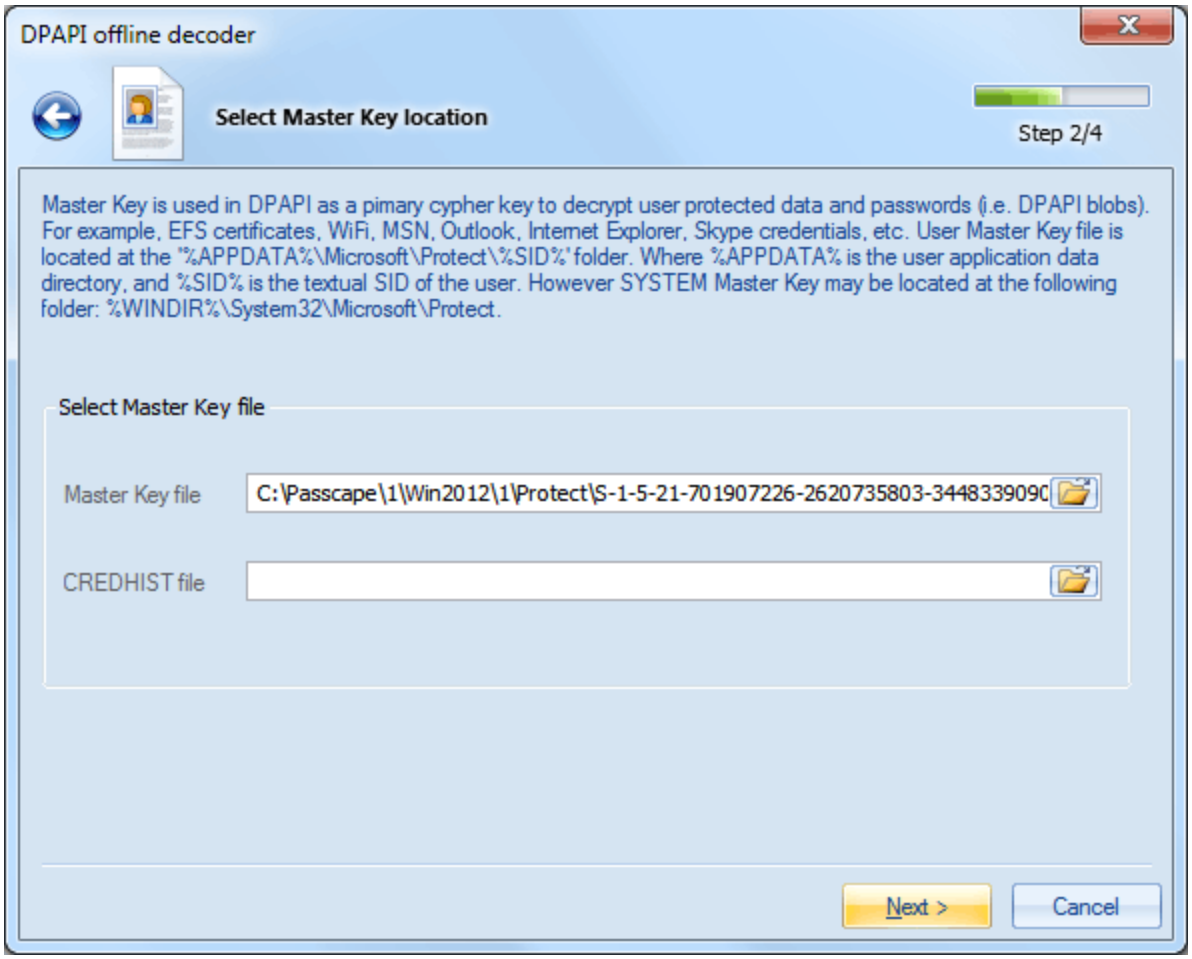
除此之外，对于out.dat文件的脱机解密，我们将需要数据所有者的MD4 hash，该散列存储在活动目录中。为了获得它，我们将利用我们的实用程序 并复制一份NTDS.DIT 文件，以及获取散列所需的SYSTEM注册表副本。

3.3 在不知道所有者登录密码的情况下解密用户DPAPI秘钥

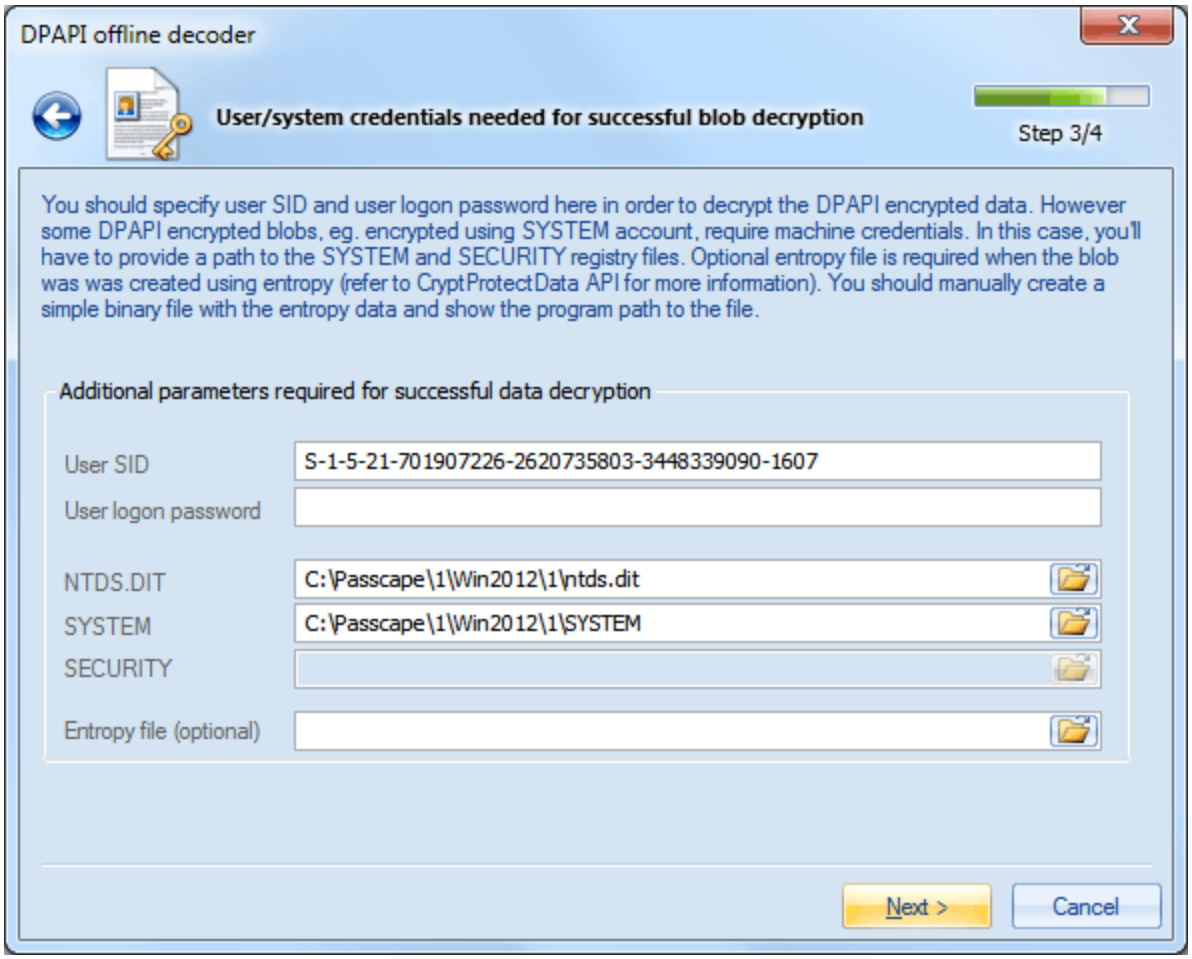
结果，我们会得到:
- 带加密机密的out.dat文件，我们需要解密
- 数据所有者的文本SID
- 他的万能钥匙
- 所有者的散列(NTDS.DIT文件和系统)。



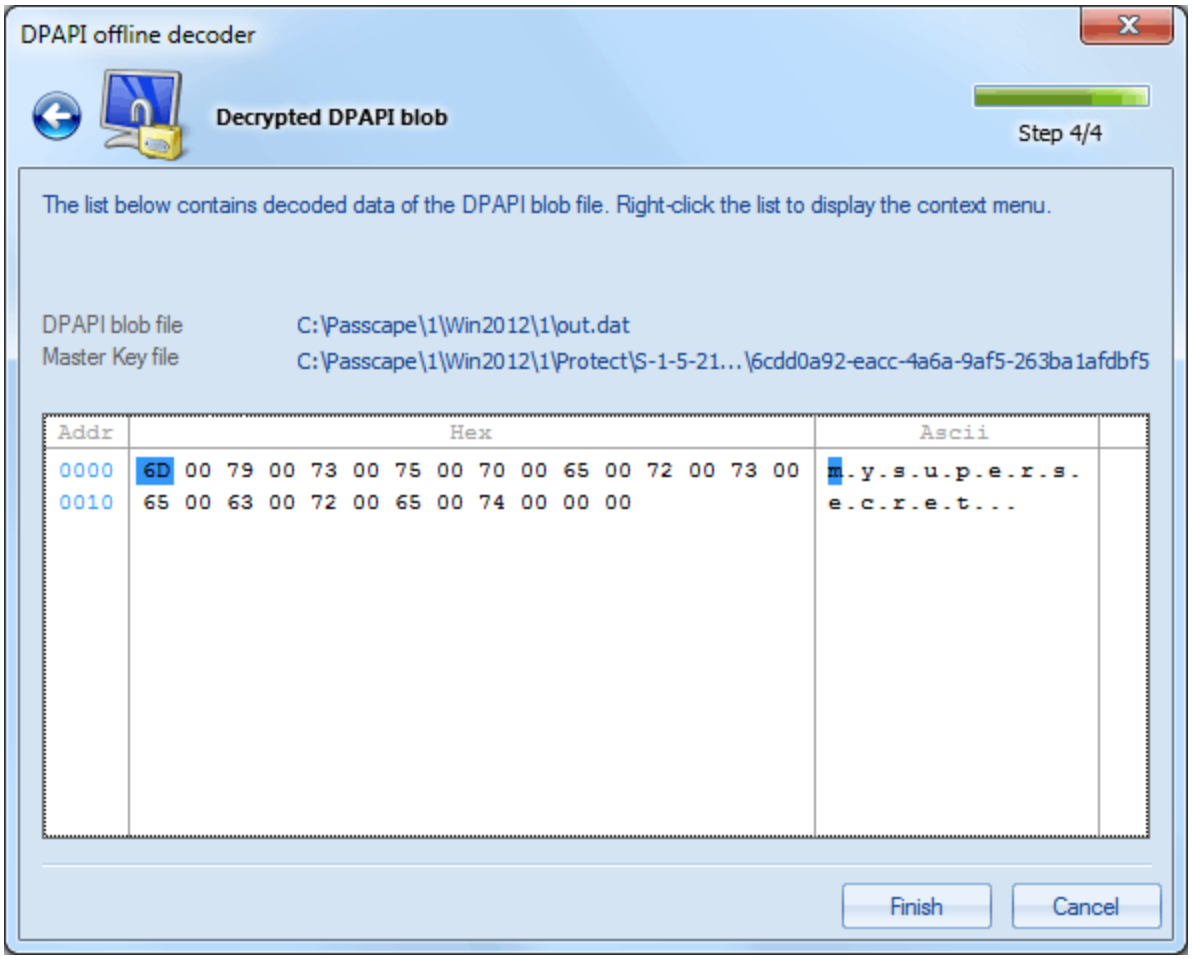
Now, launch the DPAPI offline decryption utility and tell it the path to the out.dat file.



On the second step of the Wizad, specify the path to the data owner's Master Key: select the path and then click Next. The program pops a warning that a vulnerability is found in the Master Key, and therefore the decryption can be carried out two ways: with the user's password and without it. We, certainly, want the latter.



So, on the next step of the Wizard, enter the owner's SID and the path to the NTDS.DIT and SYSTEM files, leaving the password field blank.



单击Next，我们就得到了解密的秘密。如您所见，不需要所有者的密码。

4 总结

目前尚不清楚的是，交互用户中的这一缺陷是否与错误有关，或者不应将其视为错误。也许这是服务器操作系统中DPAPI实现的一个特性；例如以提供向后兼容性。另一方面，这似乎是极不可能的，因为用户的机密数据的安全没有得到适当的保证，而DPAPI是专门用于基于所有者密码的数据保护系统。有趣的是，事实证明，桌面电脑比运行服务器操作系统的电脑更能抵抗离线密码恢复。不管怎样，系统管理员应该意识到这个漏洞，因为预先警告是预先准备好的。