विंडोज़ में LSA रहस्य

पास्केप सोफ्टवेर (हिन्दी अनुवाद : धीरेन कुमार)

Table contents

LSA रहस्य क्या हैं?

1 LSA रहस्य क्या हैं?

LSA रहस्य विंडोज़ में स्थानीय सुरक्षा प्राधिकरण (LSA) द्वारा उपयोग किए जाने वाले महत्वपूर्ण डेटा के लिए एक विशेष प्रोटेक्टेड स्टोरेज है। LSA को सिस्टम की स्थानीय सुरक्षा नीति के प्रबंधन, ऑडिटिंग, प्रमाणीकरण, सिस्टम में यूजर्स को लॉग इन करने, निजी डेटा संग्रहीत करने के लिए डिज़ाइन किया गया है। यूजर्स और सिस्टम के संवेदनशील डेटा को गुप्त रूप से संग्रहीत किया जाता है। सभी गुप्त डेटा तक पहुंच केवल सिस्टम के लिए उपलब्ध है। हालांकि, जैसा कि नीचे दिखाया गया है, कुछ प्रोग्राम, विशेष रूप से विंडोज पासवर्ड रिकवरी, इस प्रतिबंध को ओवरराइड करने की अनुमति देते हैं।

LSA रहस्यों में क्या संग्रहीत है?

2 LSA रहस्यों में क्या संग्रहीत है?

मूल रूप से, रहस्यों में कैश्ड डोमेन रिकॉर्ड शामिल थे। बाद में, विंडोज डेवलपर्स ने स्टोरेज के लिए एप्लिकेशन क्षेत्र का विस्तार किया। इस समय, वे पीसी यूजर्स के टेक्स्ट पासवर्ड, सर्विस अकाउन्ट पासवर्ड (उदाहरण के लिए, जिन्हें कुछ कार्यों को करने के लिए एक निश्चित यूजर द्वारा चलाया जाना चाहिए), इंटरनेट एक्सप्लोरर पासवर्ड, RAS कनेक्शन पासवर्ड, SQL और CISCO पासवर्ड, सिस्टम अकाउन्ट स्टोर कर सकते हैं। पासवर्ड, निजी उपयोगकर्ता डेटा जैसे EFS एन्क्रिप्शन की, और भी बहुत कुछ।

उदाहरण के लिए, **NL$KM** सीक्रेट में कैश्ड डोमेन पासवर्ड एन्क्रिप्शन की होती है। **L$RTMTIMEBOMB** विंडोज की एक निष्क्रिय कॉपी की समाप्ति तक बचे हुए समय को संग्रहीत करता है। **L$HYDRAENCKEY** रिमोट डेस्कटॉप प्रोटोकॉल में उपयोग की जाने वाली सार्वजनिक RSA2 की को संग्रहीत करता है। संयोग से, इस तथ्य के बावजूद कि स्वचालित लॉगिन सेट नहीं है, विंडोज 7 के कुछ वर्जनो में रहस्यों में एडमिनिस्ट्रेटर अकाउन्ट पासवर्ड का प्लेन टेक्स्ट हो सकता है, इस प्रकार संपूर्ण लक्ष्य प्रणाली से समझौता किया जा सकता है।

LSA रहस्य कहाँ संग्रहीत हैं?

3 LSA रहस्य कहाँ संग्रहीत हैं?

LSA रहस्यों को Windows रजिस्ट्री में **HKEY_LOCAL_MACHINE/Security/Policy/Secrets** की में एन्क्रिप्टेड रूप में संग्रहीत किया जाता है। मूल की, **HKEY_LOCAL_MACHINE/Security/Policy**, में अतिरिक्त डेटा होता है, जो रहस्यों तक पहुँचने और उन्हें डिक्रिप्ट करने के लिए आवश्यक होता है। यहाँ इस की के कुछ मूल्यों का विवरण दिया गया है।

की: **HKEY_LOCAL_MACHINE/Security/Policy/SecDesc**
वेल्यु का नाम:
डेटा का प्रकार: **REG_BINARY**
विवरण: गोपनीयता के साथ रजिस्ट्री ट्री तक पहुँचने के लिए सुरक्षा विवरणक।

की: **HKEY_LOCAL_MACHINE/Security/Policy/PolState**
वेल्यु का नाम:
डेटा का प्रकार: **REG_BINARY**
विवरण: रहस्य सबसिस्टम की वर्तमान स्थिति।

की: **HKEY_LOCAL_MACHINE/Security/Policy/PolRevesion**
वेल्यु का नाम:
डेटा का प्रकार: **REG_BINARY**
विवरण: सबसिस्टम का वर्जन शामिल है।

की: **HKEY_LOCAL_MACHINE/Security/Policy/PolPrDmS**
वेल्यु का नाम:
डेटा का प्रकार: **REG_BINARY**
विवरण: डोमेन SID.

की: **HKEY_LOCAL_MACHINE/Security/PolicyPolPrDmN**
वेल्यु का नाम:
डेटा का प्रकार: **REG_BINARY**
विवरण: डोमेन नाम।

की: **HKEY_LOCAL_MACHINE/Security/PolicyPolEKList**
वेल्यु का नाम:
डेटा का प्रकार: **REG_BINARY**
विवरण: LSA रहस्यों के लिए एन्क्रिप्शन की की सूची शामिल है.

PolRevesion में वेल्यु 1.1 NT ऑपरेशन सिस्टम से मेल खाता है, 1.5 - Windows 2000, 1.7 - Windows XP और Win2K3, 1.9 - Windows Vista, 1.10 - Windows 7| Windows Vista से पहले, **PolSecretEncryptionKey** वेल्यु में रजिस्ट्री में केवल एक एन्क्रिप्शन की संग्रहीत की गई थी, Windows Vista से शुरू होकर, **PolEKList** में कई एन्क्रिप्शन कीज हो सकती हैं।

LSA रहस्य विस्तार से

4 LSA रहस्य विस्तार से

भौतिक स्तर पर, रहस्य एक बाइनरी रजिस्ट्री फ़ाइल सुरक्षा में की के गुप्त नाम के साथ संग्रहीत किए जाते हैं। उदाहरण के लिए, **Security/Policy/Secrets/$MACHINE.ACC** | रजिस्ट्री में प्रत्येक रहस्य को पाँच मानों द्वारा दर्शाया गया है:

1. **CurrVal** - रहस्य का वर्तमान एन्क्रिप्टेड मूल्य।
2. **CupdTime** - अंतिम अद्यतन समय, 8-बाइट FILETIME संरचना के रूप में।
3. **OldVal** - रहस्य का पिछला मूल्य।
4. **OupdTime** - पिछला अद्यतन समय।
5. **SecDesc** - सुरक्षा डिस्क्रिप्टर, यानी कौन से यूजर्स रहस्य तक पहुँच सकते हैं, और जिन्हें इसे एक्सेस करने से प्रतिबंधित किया गया है।

यदि सिस्टम रहस्यों में से किसी एक को पढ़ने/डिक्रिप्ट करने में असमर्थ है, तो वह इसमें छठा वेल्यु लिखता है, **PolMod**, जो इंगित करता है कि रहस्य क्षतिग्रस्त है। उदाहरण के लिए, यदि पावर आउटेज या रजिस्ट्री फ़ाइल के क्षतिग्रस्त होने के कारण LSA डेटाबेस में लेन-देन पूरा नहीं हुआ था।

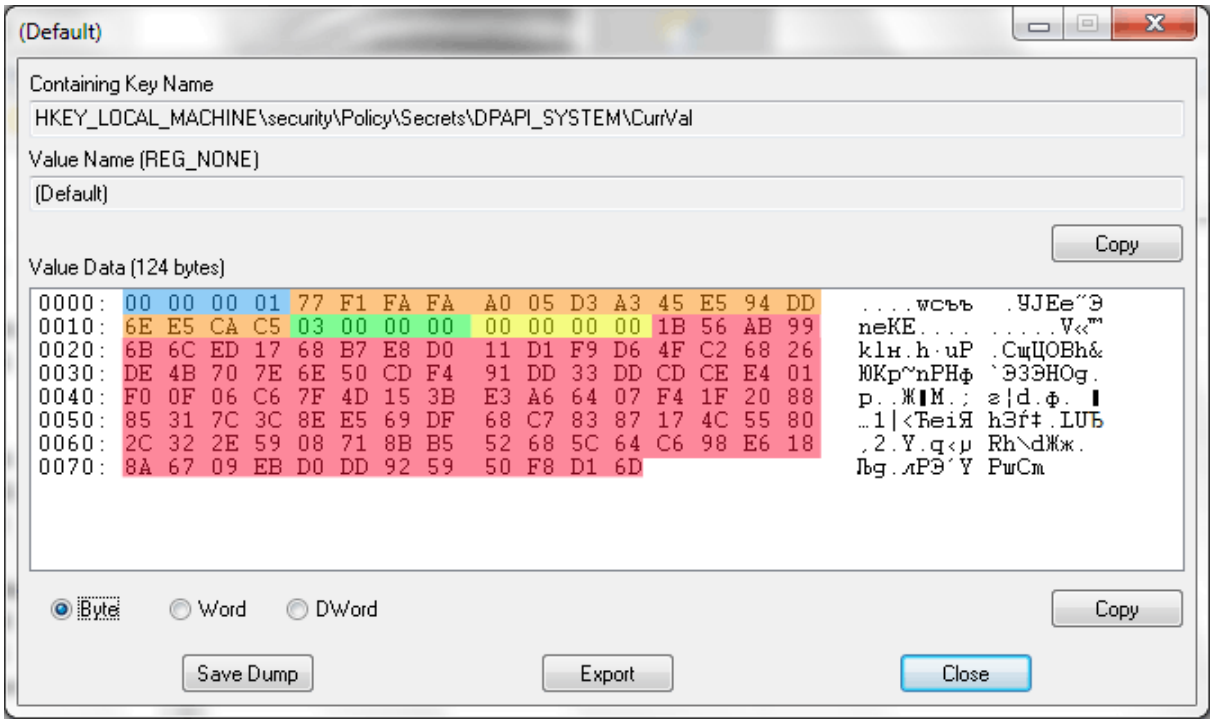
CurrVal और OldVal डेटा संरचना

5 CurrVal और OldVal डेटा संरचना

वर्जन 1.9 से शुरू होकर, रहस्यों की संरचना नाटकीय रूप से बदल गई है; इसलिए, हम पुराने प्रारूप को कवर नहीं करने जा रहे हैं। एक एन्क्रिप्शन की के बजाय, अब आप एन्क्रिप्शन की सूची (**PolEKList**) पर प्रत्येक रहस्य को किसी भी वेल्यु से बाँध सकते हैं।

एन्क्रिप्शन एल्गोरिथम का चयन करने का विकल्प भी है! तो, डेटा संरचना में पहले 4 बाइट्स डेटा का वर्जन है; फिर सूची में आवश्यक की का पता लगाने के लिए 16-बाइट एन्क्रिप्शन की पहचानकर्ता का अनुसरण करता है। इसके बाद एन्क्रिप्शन एल्गोरिदम की सूची के लिए एक पहचानकर्ता के साथ एक DWORD है जिसके साथ गुप्त एन्क्रिप्ट किया गया है।

उदाहरण के लिए, मान 3 SHA-256 हैशिंग एल्गोरिथम और AES-256 ब्लॉक एन्क्रिप्शन एल्गोरिथम के बंडल से मेल खाता है। एल्गोरिथम पहचानकर्ता के बाद डिक्रिप्शन के दौरान उपयोग किए जाने वाले विभिन्न फलेग के साथ 4-बाइट वेल्यु होता है। और, अंत में, एन्क्रिप्टेड डेटा चला जाता है। आंकड़ा देखें।



Windows 2000, XP, 2003 में LSA रहस्य एक्सप्लोइट

6 Windows 2000, XP, 2003 में LSA रहस्य एन्क्रिप्शन

Windows Vista तक, रहस्यों का डिक्रिप्शन बल्कि तुच्छ लग रहा था। सबसे पहले, किसी को गुप्त एन्क्रिप्शन की को डिक्रिप्ट करने की आवश्यकता होती है। यहाँ यह कैसा दिखता था:

```
BOOL CSecrets::DecryptPrimaryKey()
{
   BYTE rc4key[0x10];

   MD5Init();
   MD5Update(m_pSyskey,0x10);
   for ( int i=0; i<1000; i++)
      MD5Update(((LPBYTE)m_pCypherKey)+0x3C,0x10);
   MD5Final(rc4key);

   RC4SetKey(rc4key,0x10);
   RC4Decrypt(((LPBYTE)m_pCypherKey)+0xC,0x30);

                                    return ( memcmp(((LPBYTE)m_pCypherKey)
+0xC,CYPHERKEY_AUTHENTIFICATOR,0x10)==0 );
}
```

जहाँ m_pSyskey - 16-बाइट SYSKEY वेल्यु;
m_pCypherKey - रजिस्ट्री की
HKEY_LOCAL_MACHINE/Security/Policy/PolSecretEncryptionKey से वेल्यु

एक बार गुप्त एन्क्रिप्शन की प्राप्त हो जाने के बाद, कोई भी रहस्यों के डिक्रिप्शन के लिए आगे बढ़ सकता है। DES एल्गोरिथम का उपयोग करके रहस्यों को एन्क्रिप्ट किया गया था।

Windows Vista और बाद के OSes में Lsa रहस्य एक्सेसिंग

7 Windows Vista और बाद के OSes में Lsa रहस्य एन्क्रिप्शन

Windows Vista (और उच्चतर OSes) में, एन्क्रिप्शन एल्गोरिथ्म, जैसा कि पहले उल्लेख किया गया था, बहुत अधिक परिष्कृत हो गया है। सबसे पहले, किसी को अभी भी एन्क्रिप्शन कीज की सूची को डिक्रिप्ट करने की आवश्यकता है (हाँ, अब कई कीज की अनुमति है), **HKEY_LOCAL_MACHINE/Security/Policy/PolEKList** में संग्रहीत। फिर वास्तविक रहस्यों के लिए आगे बढ़ें। प्रत्येक रहस्य अब एक प्रमुख पहचानकर्ता, एन्क्रिप्शन एल्गोरिथम पहचानकर्ता और वास्तविक एन्क्रिप्टेड डेटा संग्रहीत करता है। की को डिक्रिप्ट करने के लिए एक कार्यशील एल्गोरिथ्म इस तरह दिखता है:

- की वेल्यु पढ़ें और एन्क्रिप्शन की पहचानकर्ता खोजें।
- एन्क्रिप्शन कीज (PolEKList) की सूची में, आपके द्वारा पहले प्राप्त किए गए पहचानकर्ता का उपयोग करके आवश्यक की खोजें।
- एल्गोरिथ्म पहचानकर्ता और मिली की का उपयोग करके रहस्य को डिक्रिप्ट करें।

इस प्रकार, LSA डेटाबेस में रहस्यों को न केवल विभिन्न एल्गोरिदम के साथ एन्क्रिप्ट किया जा सकता है, बल्कि अलग-अलग मूल संदर्भ भी हो सकते हैं। उदाहरण के लिए, अन्य पीसी से SYSKEY का उपयोग करें।

रहस्यों को पढ़ना और संपादित करना

8 रहस्यों को पढ़ना और संपादित करना

सॉफ्टवेयर डेवलपर्स के लिए उपलब्ध रहस्यों को संभालने के लिए API का एक सेट है। इस प्रकार, कोई भी विंडोज़ एप्लिकेशन अपने स्वयं के रहस्य बना और पढ़ सकता है, लेकिन केवल वर्तमान यूजर संदर्भ की सीमाओं के भीतर। रहस्य पढ़ने के लिए स्रोत कोड के साथ परिशिष्ट 1 देखें।

यदि आपको LSA रहस्य देखने या संपादित करने की आवश्यकता है, उदाहरण के लिए, अपने अकाउन्ट के टेक्स्ट पासवर्ड को हटाने के लिए, आप विंडोज पासवर्ड रिकवरी टूल का लाभ उठा सकते हैं, जिसमें LSA रहस्यों को संभालने के लिए एक सुविधाजनक प्लगइन है। वैसे, यह प्लगइन वर्तमान ऑपरेटिंग सिस्टम के रहस्यों और बाहरी रजिस्ट्री फ़ाइलों दोनों के साथ काम करता है।

परिशिष्ट

9 परिशिष्ट

LSA रहस्य पढ़ने के लिए कार्यक्रम का स्रोत कोड। ध्यान दें कि सभी रहस्यों को यूजर के संदर्भ में नहीं पढ़ा जा सकता है। इसके अलावा, एडमिनिस्ट्रेटर विशेषाधिकारों की आवश्यकता है। कार्यक्रम के निष्पादन योग्य को निम्न लिंक पर डाउनलोड किया जा सकता है।

```
// LsaSecretReader.cpp : Defines the entry point for the console application.
#include "stdafx.h"
#include <windows.h>
#include <stdio.h>
#include <ntsecapi.h>


#pragma comment (lib, "Advapi32")

PLSA_UNICODE_STRING InitLsaString(LPWSTR wszString,
PLSA_UNICODE_STRING lsastr)
{
	if ( !lsastr )
		return NULL;

	if ( wszString )
	{
		lsastr->Buffer=wszString;
		lsastr->Length=(USHORT)lstrlenW(wszString)*sizeof(WCHAR);
		lsastr->MaximumLength=lsastr->Length+2;
	}
	else
	{
		lsastr->Buffer=L"";
		lsastr->Length=0;
		lsastr->MaximumLength=2;
	}

	return lsastr;
}


int _tmain(int argc, _TCHAR* argv[])
{
	NTSTATUS status;
	LSA_OBJECT_ATTRIBUTES att;
	LSA_HANDLE pol;
	LSA_UNICODE_STRING secret, *data=NULL;

	if ( argc!=2 )
	{
		_tprintf(TEXT("Syntax: %s secretnamen"),argv[0]);
```

```
		return 1;
	}

	memset(&att,0,sizeof(att));

	status=LsaOpenPolicy(NULL,&att,0,&pol);
	if ( status!=ERROR_SUCCESS )
	{
		_tprintf(TEXT("LsaOpenPolicy error: %lXn"),status);
		return 2;
	}

	InitLsaString(argv[1],&secret);
	status=LsaRetrievePrivateData(pol,&secret,&data);
	if ( status!=ERROR_SUCCESS )
	{
		_tprintf(TEXT("LsaRetrievePrivateData error: %lXn"),status);
		return 3;
	}
	LsaClose(pol);

	if ( data && data->Buffer && data->Length )
	{
		for ( USHORT i=0; i<data->Length; i+=16 )
		{
			_tprintf(TEXT("%04X: "),i);
			LPBYTE ptr=(LPBYTE)data->Buffer;
			ptr+=i;
			for ( int j=0; j<min(16,data->Length-i); j++ )
				_tprintf(TEXT("%02X "),ptr[j]);
			_tprintf(TEXT("n"));
		}
	}
	else
	{
		_tprintf(TEXT("No data"));
	}

	return 0;
}
```

आउटपुट का उदाहरण

```
C:>LsaSecretReader.exe DPAPI_SYSTEM
0000: 01 00 00 00 73 4F 19 CF 6B B7 6C 8A BC 6D 35 EF
0010: 19 9C A6 3E 9A 80 A7 0C 9D D4 FD B1 20 C6 B1 A5
0020: 7A 87 5F 2B 51 3E 1D E0 45 9B 99 B2
```