

# Analysis of Malicious Security Support Provider DLLs

Matt Graeber  
October 7, 2014

## Introduction

- Matt Graeber
  - FireEye Labs Advanced Reverse Engineering (FLARE) Team
    - Malware Analyst
    - Instructor
  - Past
    - Researcher
    - US Army Red Team
    - 中文翻译
  - Reformed certification hoarder
  - For fun
    - PowerShell!
    - PowerSploit
  - Twitter - @mattifestation



## Goals

- What are security support providers (SSP)
- Local security authority (LSA)/SSP architecture
- SSPs from an attacker's perspective
- Legitimate SSPs
- SSP internals
- Installation
- Detection
- Mitigation
- Obligatory IDA screenshot
- Obligatory PowerShell screenshot

## Background

- A malicious security support provider (SSP) DLL was found recently during a recent IR engagement.
- Searching for 'SpLsaModelInitialize' – a required SSP DLL export, yielded only two unique hits in our internal malware database.
- The uniqueness of this type of malware warranted additional investigation...

## Definitions

- A security support provider (SSP) – a.k.a security package:
  - A user-mode security extension used to perform authentication during a client/server exchange.
  - e.g. schannel (SSL)
- An authentication package (AP)
  - Used to extend interactive logon authentication
  - e.g. Enable RSA token authentication
- SSP/AP
  - Can serve the tasks of SSPs and APs. Loaded in lsass.
  - e.g. kerberos and msv1\_0 (NTLM)

## LSA Extensible Architecture

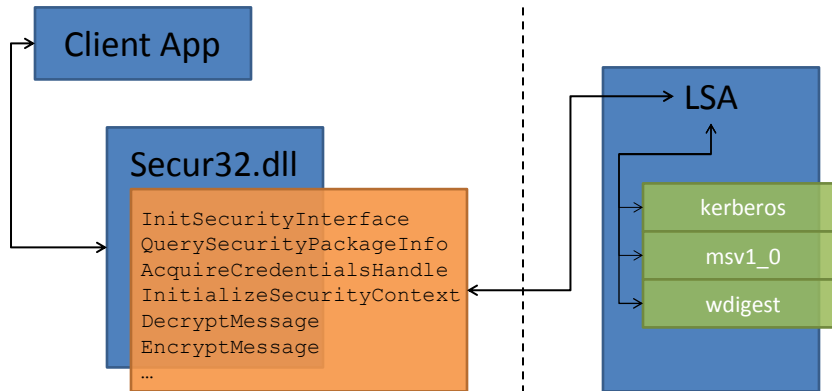
The Local Security Authority (LSA) is responsible for nearly all aspects of local security on a system

- Authenticate and log on users
- Manage credentials – SAM/NTDS/etc.
- Built-in support for message privacy and integrity

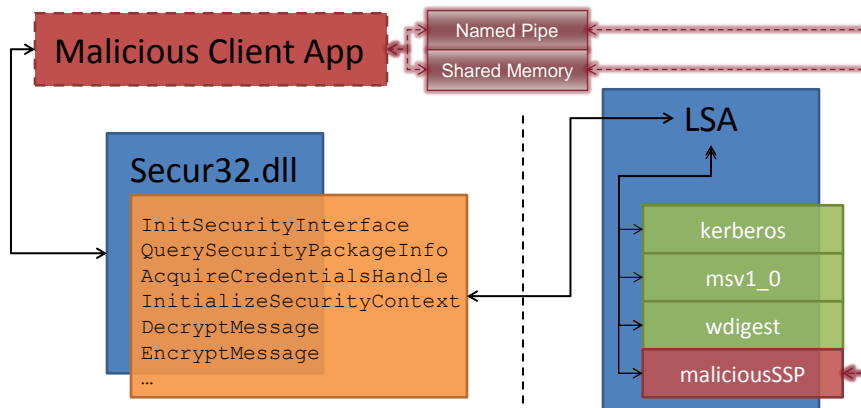
LSA is extensible

- SSP/APs are loaded into LSA (lsass.exe) at boot
- Custom SSP/APs can either replace or proxy existing providers.

## SSPI Architecture - Legitimate



## SSPI Architecture - Malicious



## SSP Benefits from an Attacker's Perspective

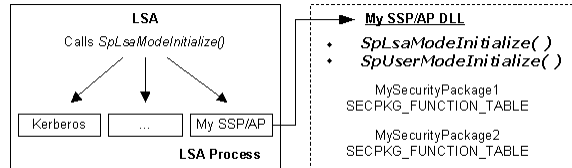
- Once installed, your DLL is loaded into lsass.exe!
  - i.e. no need to inject into lsass.exe
- Not a well-known persistence mechanism
- Once loaded into lsass, you are handed an officially supported “credential capture API”.
- i.e. officially supported, Mimikatz-like functionality without needing Mimikatz

## Common Legitimate SSPs

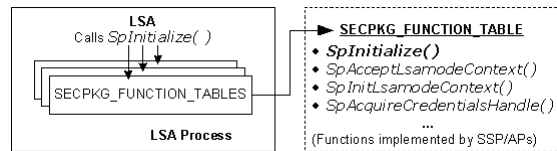
- Microsoft
  - msv1\_0.dll
  - kerberos.dll
  - negoexts.dll
  - wsauth.dll
  - schannel.dll
  - TSpkg.dll
  - msoidssp.dll
  - pku2u.dll
  - etc.
- 3<sup>rd</sup> Party
  - wsauth.dll – VMWare Horizon View
  - CTXAUTH.dll – Citrix
  - phonefactorlsa.dll – PhoneFactor

# LSA SSP Initialization Procedure

1. Inform LSA of SSP implemented functions



2. Inform SSP of available LSA support functions



<http://msdn.microsoft.com/library/windows/desktop/aa378339.aspx>

# SSP Development - Requirements

```

Administration Developer Command Prompt for VS2013
C:\>dumpbin /NOLOGO /EXPORTS %windir%\System32\kerberos.dll
Dump of file C:\windows\System32\kerberos.dll
File Type: DLL

Section contains the following exports for Kerberos.dll
00000000 characteristics
5315937D time date stamp Tue Mar 04 03:49:01 2014
0.00 version
1 ordinal base
32 number of functions
10 number of names

ordinal hint RVA      name
5 0 00022EE4 DllMain
6 1 00067468 KerbCreateTokenFromTicket
7 2 00064FC8 KerbDomainChangeCallback
8 3 0005C1EC KerbIsInitialized
9 4 0005C1FC KerbKdcCallback
10 5 000629D0 KerbMakeKdcCall
11 6 000216EC SpInitialize
12 7 0002323C SpInstanceInit
13 8 00021434 SpLsaModeInitialize
14 9 0002311C SpUserModeInitialize

Summary
8000 .data
6000 .pdata
F000 .rdata
1000 .reloc
3000 .rsrc
93000 .text
    
```

Minimum required functions\*

1. SpInitialize
2. SpShutDown
3. SpGetInfo

Required Export

## SSP Development - Implementation

```
NTSTATUS NTAPI SpLsaModeInitialize(
    _In_    ULONG LsaVersion,
    _Out_   PULONG PackageVersion,
    _Out_   PSECPKG_FUNCTION_TABLE *ppTables,
    _Out_   PULONG pcTables
);
```

- Called by LSA when your SSP DLL is loaded.
- Only required export function
- Informs LSA of the functions your SSP DLL implements via PSECPKG\_FUNCTION\_TABLE
- LSA expects at a minimum, the following in PSECPKG\_FUNCTION\_TABLE:

▪ SpInitialize	typedef struct SECPKG_FUNCTION_TABLE {	
▪ SpShutDown	...	
▪ SpGetInfo	SpInitializeFn	*Initialize;
	SpShutdownFn	*Shutdown;
	SpGetInfoFn	*GetInfo;
	SpAcceptCredentialsFn	*AcceptCredentials;
	SpGetCredentialsFn	*GetCredentials;
	SpGetUserInfoFn	*GetUserInfo;
	SpAddCredentialsFn	*AddCredentials;
	SpSetExtendedInformationFn	*SpChangeAccountPasswordFn;
	...	
	} SECPKG_FUNCTION_TABLE, *PSECPKG_FUNCTION_TABLE;	

## SSP Development - Implementation

```
NTSTATUS SpInitialize(
    _In_    ULONG_PTR PackageId,
    _In_    PSECPKG_PARAMETERS Parameters,
    _In_    PLSA_SECPKG_FUNCTION_TABLE
    FunctionTable
);
```

- Called by LSA after SpLsaInitialize
- Informs your SSP DLL the available LSA functions via PLSA\_SECPKG\_FUNCTION\_TABLE

```
typedef struct _LSA_SECPKG_FUNCTION_TABLE {
    ...
    PLSA_GET_CREDENTIALS           GetCredentials;
    PLSA_OPEN_SAM_USER             OpenSamUser;
    PLSA_GET_USER_CREDENTIALS      GetUserCredentials;
    PLSA_GET_USER_AUTH_DATA        GetUserAuthData;
    PLSA_UPDATE_PRIMARY_CREDENTIALS UpdateCredentials;
    PLSA_GET_AUTH_DATA_FOR_USER     GetAuthDataForUser;
    CredReadDomainCredentialsFn     *CredReadDomainCredentials;
    PLSA_PROTECT_MEMORY            LsaProtectMemory;
    PLSA_UNPROTECT_MEMORY          LsaUnprotectMemory;
    PLSA_GET_SERVICE_ACCOUNT_PASSWORD GetServiceAccountPassword;
    ...
} LSA_SECPKG_FUNCTION_TABLE, *PLSA_SECPKG_FUNCTION_TABLE;
```

## SSP Development - Implementation

```
NTSTATUS SpShutDown(void);
```

- Called at system shutdown
- Can simply return NULL
- Isass.exe will crash if this is not implemented

## SSP Development - Implementation

```
NTSTATUS SpGetInfo(_Out_ PSecPkgInfo PackageInfo);
```

- Provides general information about a security package
- Can return the following info:
  - Name
  - Description
  - Capabilities
  - etc.
- Must be implemented but it doesn't need to do anything.



## SSP Installation

1. Copy the SSP DLL to %windir%\System32
  - Note: Because the DLL is loaded into lsass, it must be compiled for the same architecture as lsass.exe
2. Add the file name (without extension) to:
  - HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Security Packages
  - HKLM\SYSTEM\CurrentControlSet\Control\Lsa\OSConfig\Security Packages
3. Optional: Load it into lsass immediately by calling secur32!AddSecurityPackage
4. Reboot

## Malicious SSP PoC – mimilib SSP

- Benjamin Delpy (@gentilkiwi) recently added SSP functionality to mimilib.dll. He has yet to document or heavily advertise this functionality.
- Once installed and loaded into lsass.exe, it captures passwords in plaintext.
- This is achieved with the SpAcceptCredential callback function.

## Malicious SSP PoC – mimilib SSP

```

Administrator: Windows PowerShell
PS C:\> Get-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\Lsa -Name 'Security Packages'
Select-Object -Exp
kerberos
msv1_0
schannel
wdigest
lsass
pku2u
mimilib
PS C:\> Get-SecurityPackages | Ft Name, Comment, Capabilities

Name                                Comment                                Capabilities
----                                -
Negotiate                           Microsoft Package Negotiator          ...COMPATIBLE, LOGON, RESTRICTED_TOKENS
Negotiate                           NegoExtender Security Package          ...E, LOGON, MUTUAL_AUTH, NEGOTIATOR
Kerberos                             Microsoft Kerberos V1.0                ...NLY_WITH_CHECKSUM, RESTRICTED_TOKENS
NTLM                                 NTLM Security Package                  ...NEGOTIABLE, LOGON, RESTRICTED_TOKENS
Schannel                             Schannel Security Package              ...CEPT_WIN32_NAME, STREAM, MUTUAL_AUTH
Microsoft Unified Security Protocol ... Schannel Security Package              ...CEPT_WIN32_NAME, STREAM, MUTUAL_AUTH
wdigest                             Digest Authentication for Windows      ...LY, IMPERSONATION, ACCEPT_WIN32_NAME
TS SSP                               TS Service Security Package            ...IRED, ACCEPT_WIN32_NAME, MUTUAL_AUTH
pku2u                                PKU2U Security Package                 ...COMPATIBLE, MUTUAL_AUTH, NEGOTIABLE
KiwiSSP                              Kiwi Security Support Provider          CONNECTION, ACCEPT_WIN32_NAME

PS C:\> ls C:\Windows\System32\mimilib.dll

Directory: C:\Windows\System32

Mode                LastWriteTime         Length Name
----                -
-a---             6/2/2014   8:03 AM        112128 mimilib.dll

PS C:\> ls C:\Windows\System32\kiwissp.log

Directory: C:\Windows\System32

Mode                LastWriteTime         Length Name
----                -
-a---            10/1/2014   4:14 PM         2773 kiwissp.log
  
```

2014 19

## Malicious SSP PoC – mimilib SSP

%windir%\System32\kiwissp.log

```

[00000000:000003e7] [00000005] WORKGROUP\WIN-LOI4CUIDKP1$ (SYSTEM)
[00000000:000003e4] [00000005] WORKGROUP\WIN-LOI4CUIDKP1$ (NETWORK SERVICE)
[00000000:000003e4] [00000005] WORKGROUP\WIN-LOI4CUIDKP1$ (NETWORK SERVICE)
[00000000:000003e7] [00000005] WORKGROUP\WIN-LOI4CUIDKP1$ (SYSTEM)
[00000000:000003e7] [00000005] WORKGROUP\WIN-LOI4CUIDKP1$ (SYSTEM)
[00000000:000527ee] [00000002] WIN-LOI4CUIDKP1\anonymous (anonymous) badpassword
[00000000:00052807] [00000002] WIN-LOI4CUIDKP1\anonymous (anonymous) badpassword
[00000000:00065d64] [00000002] WIN-LOI4CUIDKP1\anonymous (anonymous) badpassword
[00000000:00065d7a] [00000002] WIN-LOI4CUIDKP1\anonymous (anonymous) badpassword
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:000003e4] [00000005] WORKGROUP\WIN-LOI4CUIDKP1$ (NETWORK SERVICE)
[00000000:000003e4] [00000005] WORKGROUP\WIN-LOI4CUIDKP1$ (NETWORK SERVICE)
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:000003e4] [00000005] WORKGROUP\WIN-LOI4CUIDKP1$ (NETWORK SERVICE)
[00000000:000003e4] [00000005] WORKGROUP\WIN-LOI4CUIDKP1$ (NETWORK SERVICE)
  
```

## Malicious SSP Mitigations

### Prevention

- Windows 8.1/Server 2012 R2 running Secure Boot with UEFI:
  - `HKLM\SYSTEM\CurrentControlSet\Control\Lsa\RunAsPPL (DWORD) = 1`
- Makes lsass a protected process. Forces SSP DLLs to be co-signed by Microsoft.
- With Secure Boot (w/ UEFI) enabled, RunAsPPL is set as a UEFI secure variable and cannot be deleted.

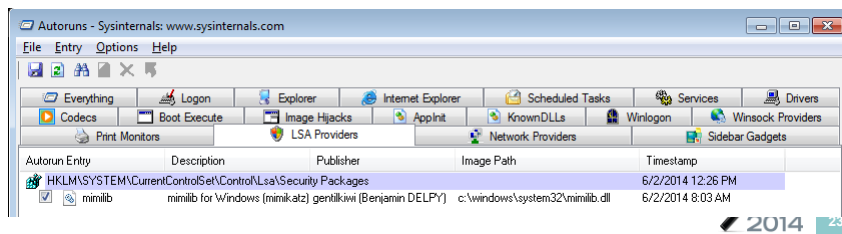
## Malicious SSP Detection

- Windows 8.1/Server 2012 R2 only
- Generate event logs upon loading of an unsigned lsass module:
  - `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe`
  - `AuditLevel = 8 (REG_DWORD)`
  - Reboot
- When an unsigned SSP is loaded, either of the following events will trigger:
  - 3033
  - 3066

## Malicious SSP Detection

### Detection

- Whitelist legitimate SSP DLLs.
- Alert when  
HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Security Packages or  
HKLM\SYSTEM\CurrentControlSet\Control\Lsa\OSConfig\security Packages is modified to contain an SSP not in the whitelist.
- Alert on any DLLs that export SpLsaModelInitialize that are not in the whitelist.
- MIGHT be present under 'LSA Providers' in Sysinternals Autoruns



## Malicious SSP Mitigations

### Removal

- Remove the SSP from the following reg keys:
  - HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Security Packages
  - HKLM\SYSTEM\CurrentControlSet\Control\Lsa\OSConfig\Security Packages
- Delete the DLL from %windir%\System32
- Call secur32!DeleteSecurityPackage?
  - Oops! MS forgot to implement that function.
- Reboot ☹

# Malicious SSP Mitigations

```
; Exported entry 19. DeleteSecurityPackageA
; Exported entry 20. DeleteSecurityPackageW

; SECURITY_STATUS __stdcall NegImportSecurityContext(LPMSTR pszPackageName)
public ?NegImportSecurityContext@00YA.JPEAU_SecBuffer@00PEAXPEA_K0Z
?NegImportSecurityContext@00YA.JPEAU_SecBuffer@00PEAXPEA_K0Z proc near
mov     eax, 80090302h ; DeleteSecurityPackageA
ret
?NegImportSecurityContext@00YA.JPEAU_SecBuffer@00PEAXPEA_K0Z endp
```

# References

- [Registering SSP/AP DLLs](#)
- [Configuring Additional LSA Protection](#)
- [LSA Mode Initialization](#)
- [Mimikatz PoC SSP](#)

## Merci!!!

Thank you Benjamin Delpy (@gentilkiwi) for the following:

1. Performing all the original research on malicious SSPs
2. Writing a PoC malicious SSP
3. Writing Mimikatz! <3
4. Patiently and enthusiastically answering all my dumb questions.



MIRcon.  
2014

27

## QUESTIONS?

MIRcon.  
2014

28