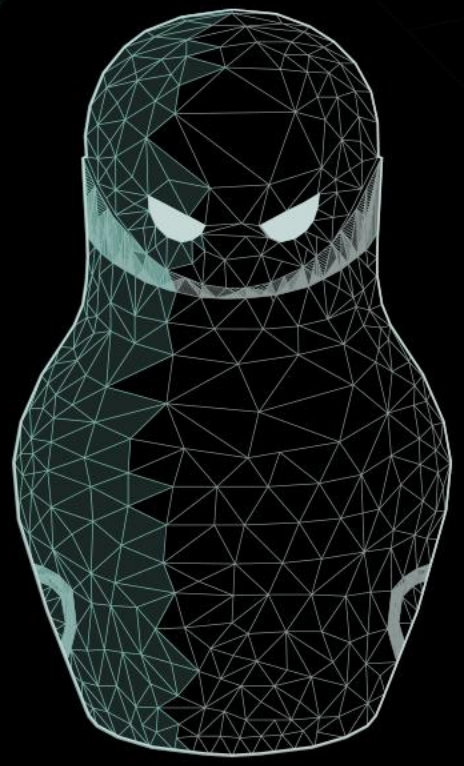


ZERO NIGHTS 2015

Did You Get Your Token?

Daniel and Azure (Keen Team)

ABOUT US

Daniel King (金龙) @long123king
• Keen Team Security Researcher
• 3/5 years working experience, former TrendMicro employee
• Windows Security Research, keen on uncovering secrets Under The Hood

Azure (杨杰韬) @Hoshizoranoaoi
• Keen Team Intern Security Researcher
• Senior student at China University of Petroleum
• Sandbox Bypass, keen on pwning programs and devices

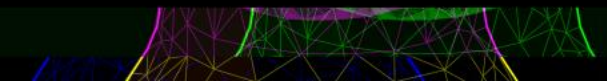
Keen Team @K33nTeam
• 5 Champions in Pwn2Own
• 2 Nominations for Pwnies Awards 2015
• Hosting GeekPwn 2014, 2015
• 10%+ foreign team members
• Peter Hlavaty and Marco Grassi were ZeroNights speakers

OUTLINE

1. Windows Security Model
2. Access Check
3. Token
4. Object and Security Descriptor
5. Protected Process
6. Sandbox
7. Browser Sandbox details
8. A story about sandbox bypass
9. How to make use of sandbox in Windows

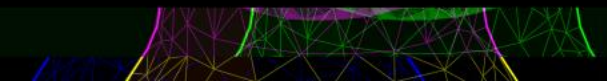
Windows Security Model

1. Securable resources are referenced as **Objects**
2. Each object has its own **Security Descriptor**
3. Each process has a **Primary Token** and zero or more **Impersonation Tokens**
4. **Access Check** happens whenever an object is created or opened
5. **Effective Token** is checked against the object's **Security Descriptor**
6. Results of **Access Check** are cached to each host process's **Handle Table**
7. Objects and Processes are all hierarchical, so **Security Descriptors** and **Tokens** are inheritable

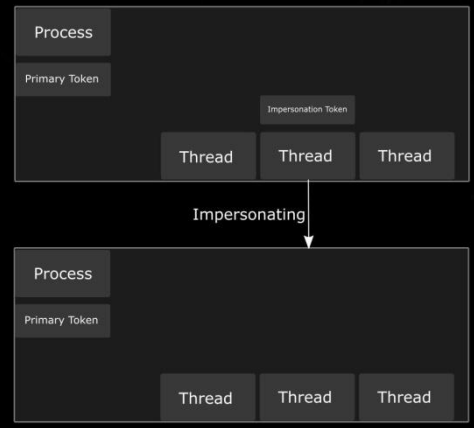


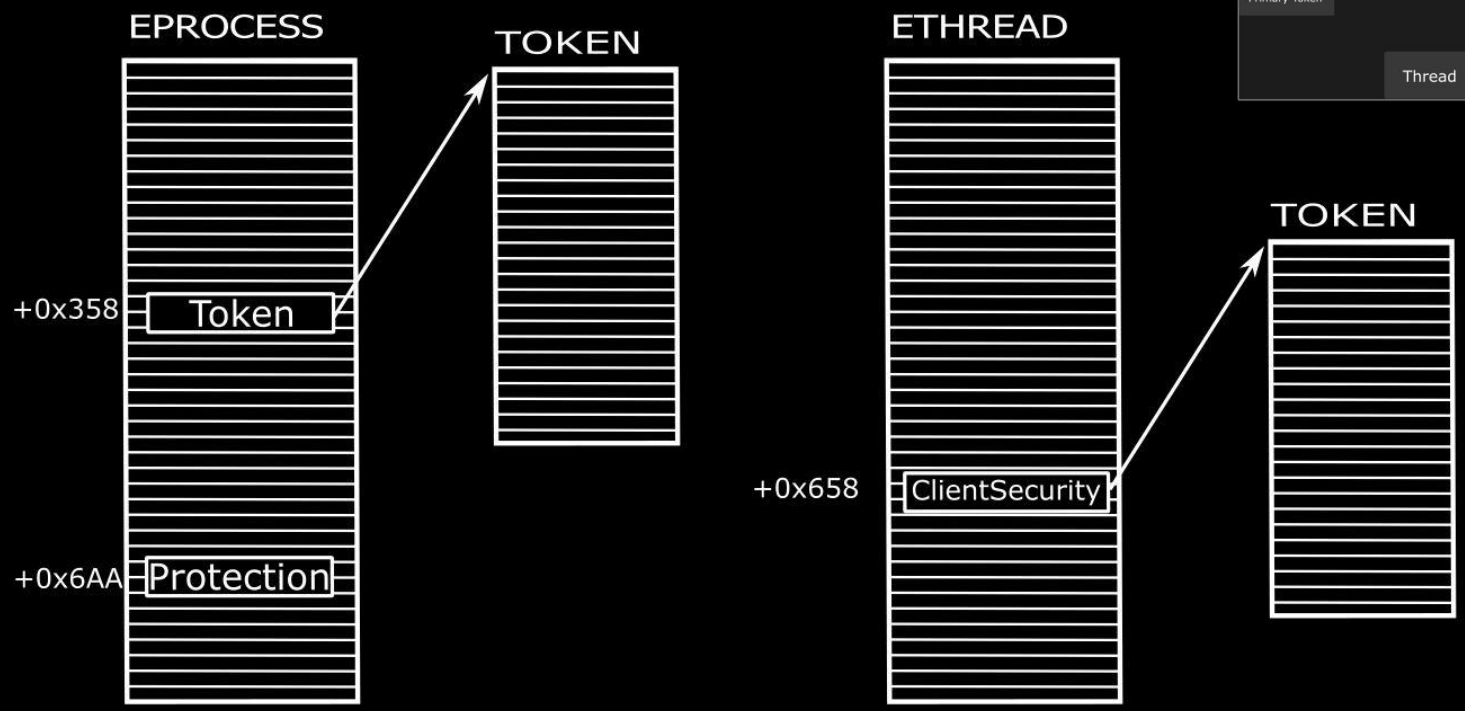
Access Check

1. **D**iscretionary **A**ccess **C**ontrol **L**ist Check
2. **Privileges** and **Super Privileges** Check
3. **Integrity Level** and **Mandatory Policy** Check
4. **Restricted Token**'s Access Check
5. **AppContainer**'s Capabilities Check
6. **Trust Level** Check

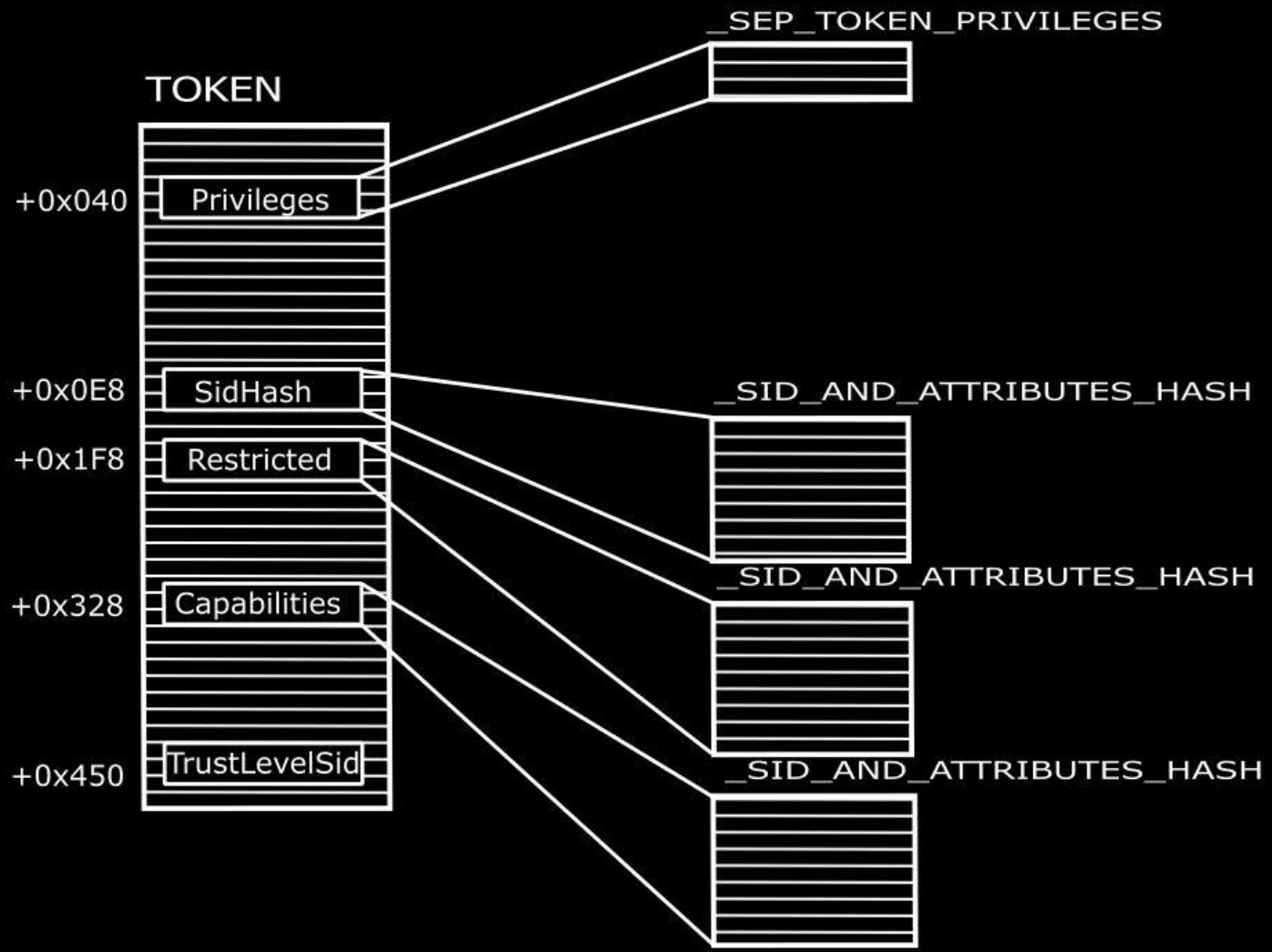


Token


Process
Primary Token
Impersonation Token
Thread
Thread
Thread
Impersonating
Process
Primary Token
Thread
Thread
Thread


EPROCESS
TOKEN
+0x358
Token
+0x6AA
Protection
ETHREAD
TOKEN
+0x658
ClientSecurity

Token Layout



```
CurrentToken                : 0x
 [+]User                    : f0 1f 47 86 cb 00 00 00 00 00 00 00 00 00 00 01 05 00 00 00 00 00 05 15 00 00 00 52 a4 e2 86 ...
    [+]User                 : f0 1f 47 86 cb 00 00 00 00 00 00 00 00 00 00 01 05 00 00 00 00 00 05 15 00 00 00 52 a4 e2 86 ...
        [+]Sid              : S-1-5-21-2263000146-343837727-1826472087-1001                                                  : DESKTOP-SN4JMIA/Daniel King
        [+]Attributes       : 0x00000000
 [+]Owner                   : 68 dc 46 86 cb 00 00 00 01 05 00 00 00 00 00 05 15 00 00 00 52 a4 e2 86 1f 8c 7e 14 97 c0 dd 6c ...
    [+]Owner                : S-1-5-21-2263000146-343837727-1826472087-1001                                                  : DESKTOP-SN4JMIA/Daniel King
 [+]Groups                  : 0e 00 00 00 00 00 00 00 48 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00 64 9c 46 86 cb 00 00 00 ...
    [+]GroupCount           : 0x0000000e
    [+]Groups[0]            : 48 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00
        [+]Sid              : S-1-5-21-2263000146-343837727-1826472087-513                                                   : DESKTOP-SN4JMIA/None
        [+]Attributes       : 0x00000007                                                                                     : mandatory,default,enabled,
    [+]Groups[1]            : 64 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00
        [+]Sid              : S-1-1-0                                                                                        : Everyone
        [+]Attributes       : 0x00000007                                                                                     : mandatory,default,enabled,
    [+]Groups[2]            : 70 9c 46 86 cb 00 00 00 10 00 00 00 00 00 00 00
        [+]Sid              : S-1-5-114                                                                                      : NT AUTHORITY/本地帐户和管理员组成员
        [+]Attributes       : 0x00000010                                                                                     : deny-only,
    [+]Groups[3]            : 7c 9c 46 86 cb 00 00 00 10 00 00 00 00 00 00 00
        [+]Sid              : S-1-5-32-544                                                                                   : BUILTIN/Administrators
        [+]Attributes       : 0x00000010                                                                                     : deny-only,
    [+]Groups[4]            : 8c 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00
        [+]Sid              : S-1-5-32-545                                                                                   : BUILTIN/Users
        [+]Attributes       : 0x00000007                                                                                     : mandatory,default,enabled,
    [+]Groups[5]            : 9c 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00
        [+]Sid              : S-1-5-4                                                                                        : NT AUTHORITY/INTERACTIVE
        [+]Attributes       : 0x00000007                                                                                     : mandatory,default,enabled,
    [+]Groups[6]            : a8 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00
        [+]Sid              : S-1-2-1                                                                                        : Console Logon
        [+]Attributes       : 0x00000007                                                                                     : mandatory,default,enabled,
    [+]Groups[7]            : b4 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00
        [+]Sid              : S-1-5-11                                                                                       : NT AUTHORITY/Authenticated Users
        [+]Attributes       : 0x00000007                                                                                     : mandatory,default,enabled,
    [+]Groups[8]            : c0 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00
        [+]Sid              : S-1-5-15                                                                                       : NT AUTHORITY/This Organization
        [+]Attributes       : 0x00000007                                                                                     : mandatory,default,enabled,
    [+]Groups[9]            : cc 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00
        [+]Sid              : S-1-5-113                                                                                      : NT AUTHORITY/本地帐户
        [+]Attributes       : 0x00000007                                                                                     : mandatory,default,enabled,
    [+]Groups[10]           : d8 9c 46 86 cb 00 00 00 07 00 00 c0 00 00 00 00
        [+]Sid              : S-1-5-5-0-131376                                                                               : Logon Session
        [+]Attributes       : 0xc0000007                                                                                     : mandatory,default,enabled,logon-id,
    [+]Groups[11]           : ec 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00
        [+]Sid              : S-1-2-0                                                                                        : Local
        [+]Attributes       : 0x00000007                                                                                     : mandatory,default,enabled,
    [+]Groups[12]           : f8 9c 46 86 cb 00 00 00 07 00 00 00 00 00 00 00
        [+]Sid              : S-1-5-64-10                                                                                    : NT AUTHORITY/NTLM Authentication
        [+]Attributes       : 0x00000007                                                                                     : mandatory,default,enabled,
    [+]Groups[13]           : 08 9d 46 86 cb 00 00 00 60 00 00 00 00 00 00 00
        [+]Sid              : S-1-16-8192                                                                                    : Mandatory Label/Medium Mandatory Level
        [+]Attributes       : 0x00000060                                                                                     : integrity,integrity-enabled,
 [+]Privileges              : 05 00 00 00 13 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 00 00 00 03 00 00 00 19 00 00 00 ...
    [+]PrivilegeCount       : 0x00000005
    [+]Privileges[0]        : 13 00 00 00 00 00 00 00 00 00 00 00
        [+]Luid             : 13 00 00 00 00 00 00 00                                                                        : SeShutdownPrivilege
            [+]HighPart     : 0x00000000
            [+]LowPart      : 0x00000013
        [+]Attributes       : 0x00000000
    [+]Privileges[1]        : 17 00 00 00 00 00 00 00 03 00 00 00
        [+]Luid             : 17 00 00 00 00 00 00 00                                                                        : SeChangeNotifyPrivilege
```

TokenInsight

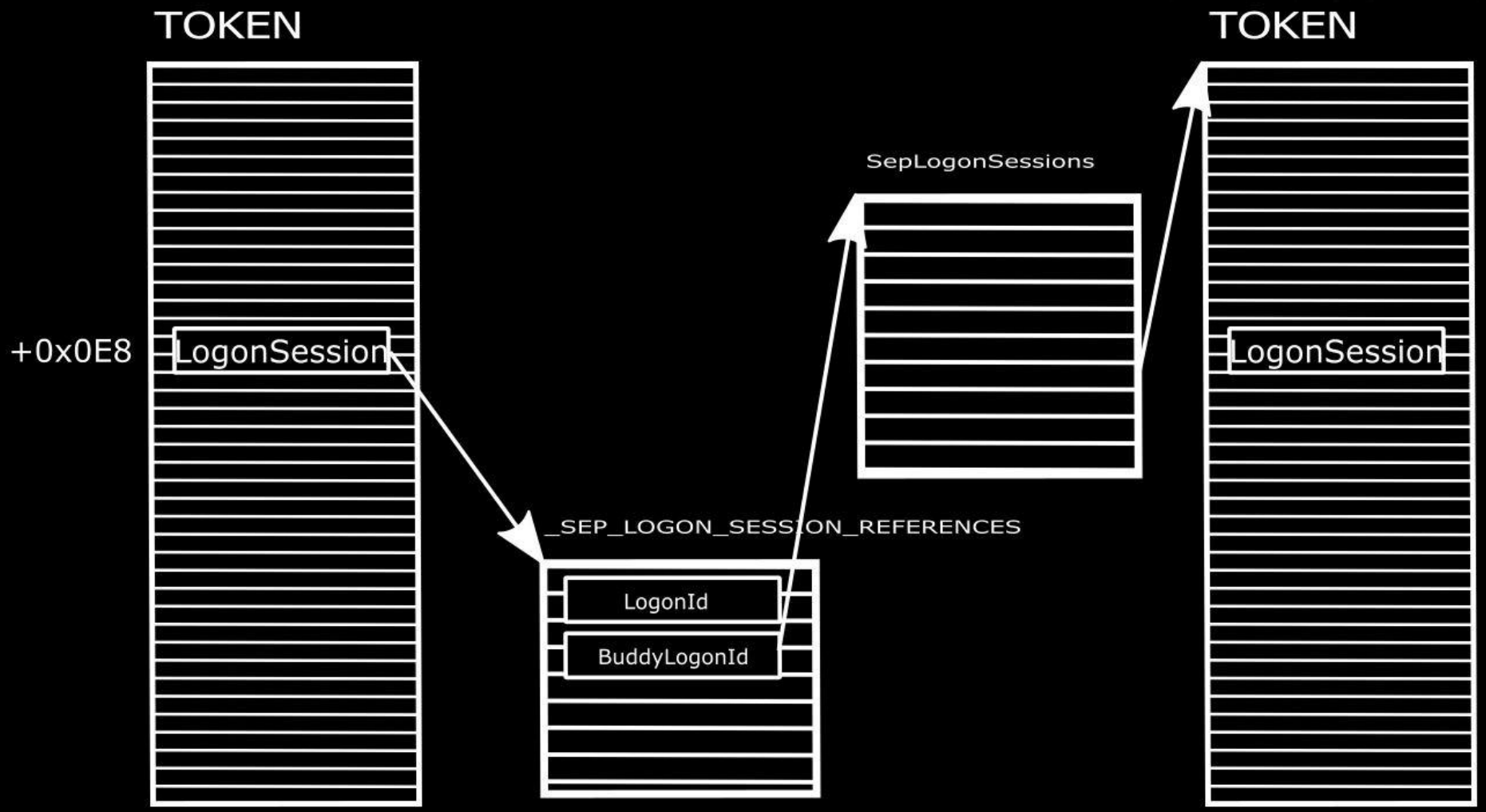
https://github.com/long123king/TokenInsight

An application for obtaining, dumping and modifying token from user land.

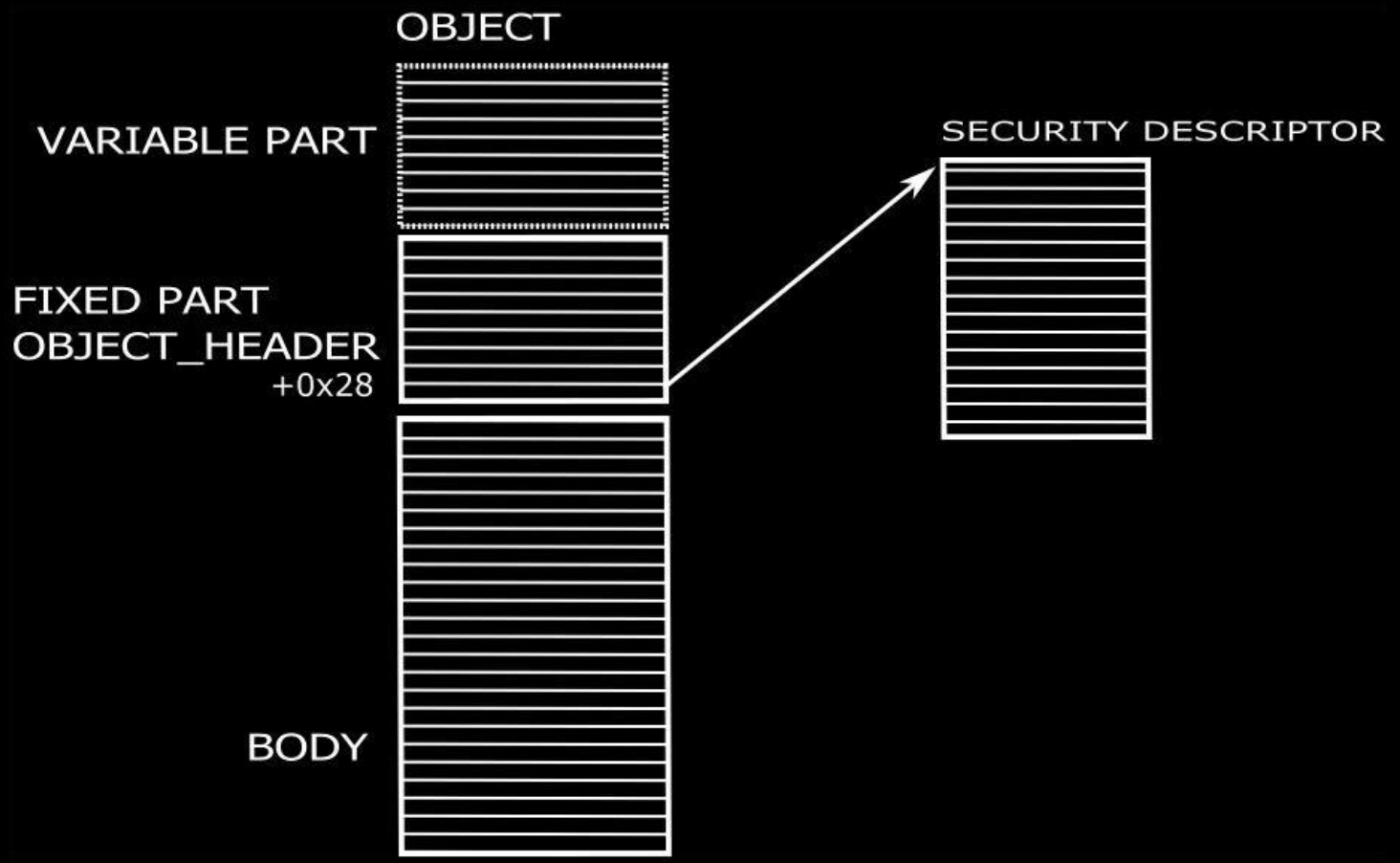
Calculate Hash of Sid Groups

```
static
NTSTATUS
RtlSidHashInitialize
(
    __in PSID_AND_ATTRIBUTES Groups,
    __in size_t GroupsCount,
    __inout PSID_AND_ATTRIBUTES_HASH HashBuffer
)
{
    if (NULL == HashBuffer)
        return 0xC000000D;

    memset(HashBuffer, 0, 0x110);
    if (0 == GroupsCount || NULL == Groups)
        return 0;

    HashBuffer->SidCount = GroupsCount;
    HashBuffer->SidAttr = Groups;

    if (GroupsCount > 0x40)
        GroupsCount = 0x40;
    if (0 == GroupsCount)
        return 0;

    size_t bit_pos = 1;

    for (size_t i = 0; i < GroupsCount; i++)
    {
        PISID sid = reinterpret_cast<PISID>((Groups + i)->Sid);

        size_t sub_authority_count = sid->SubAuthorityCount;

        DWORD sub_authority = sid->SubAuthority[sub_authority_count - 1];

        *(size_t*)(&HashBuffer->Hash[(sub_authority & 0x0000000F)]) |= bit_pos;
        *(size_t*)(&HashBuffer->Hash[((sub_authority & 0x000000F0) >> 4) + 0x10]) |= bit_pos;

        bit_pos <<= 1;
    }

    return 0;
}
```

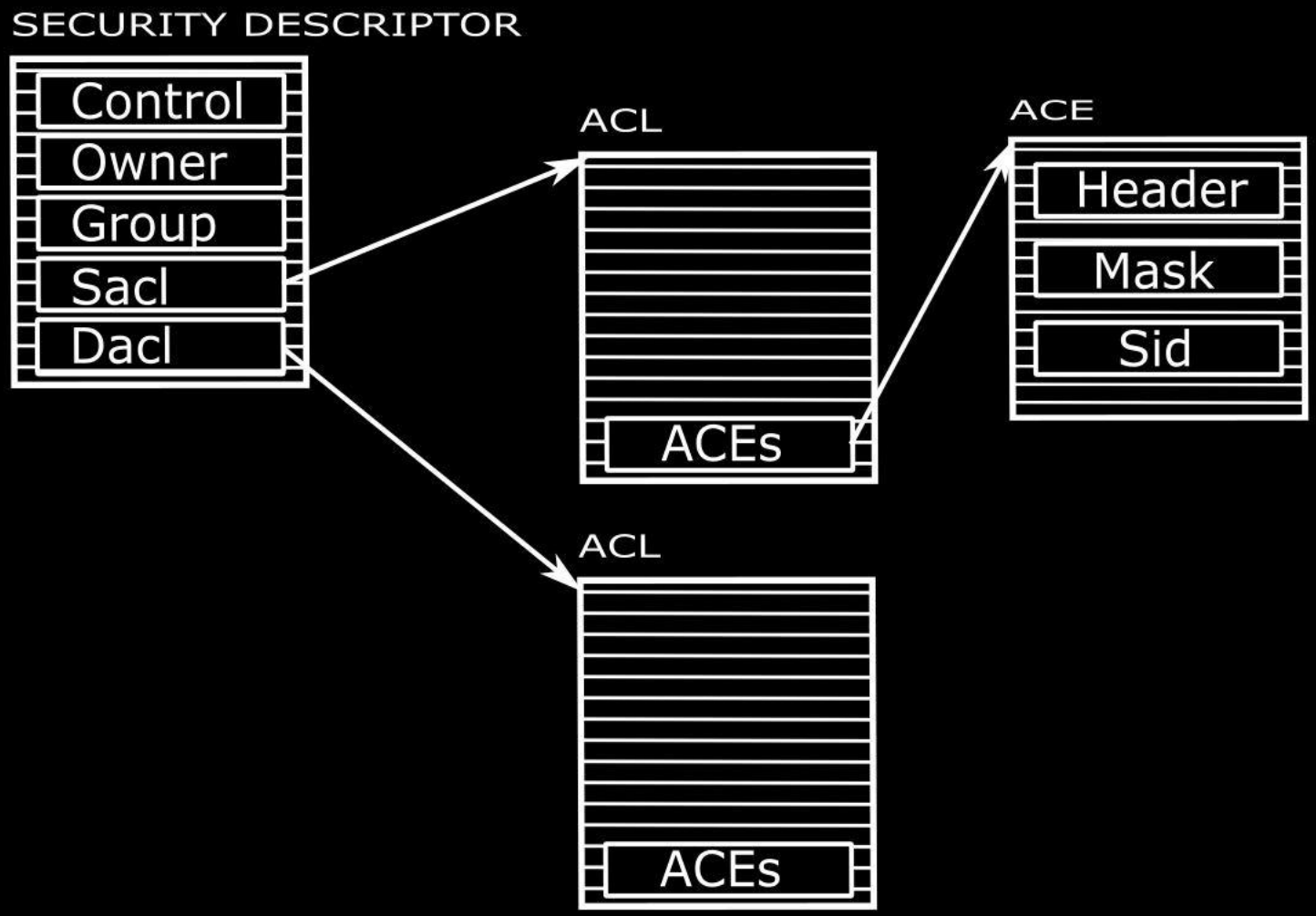
Linked Token and Session



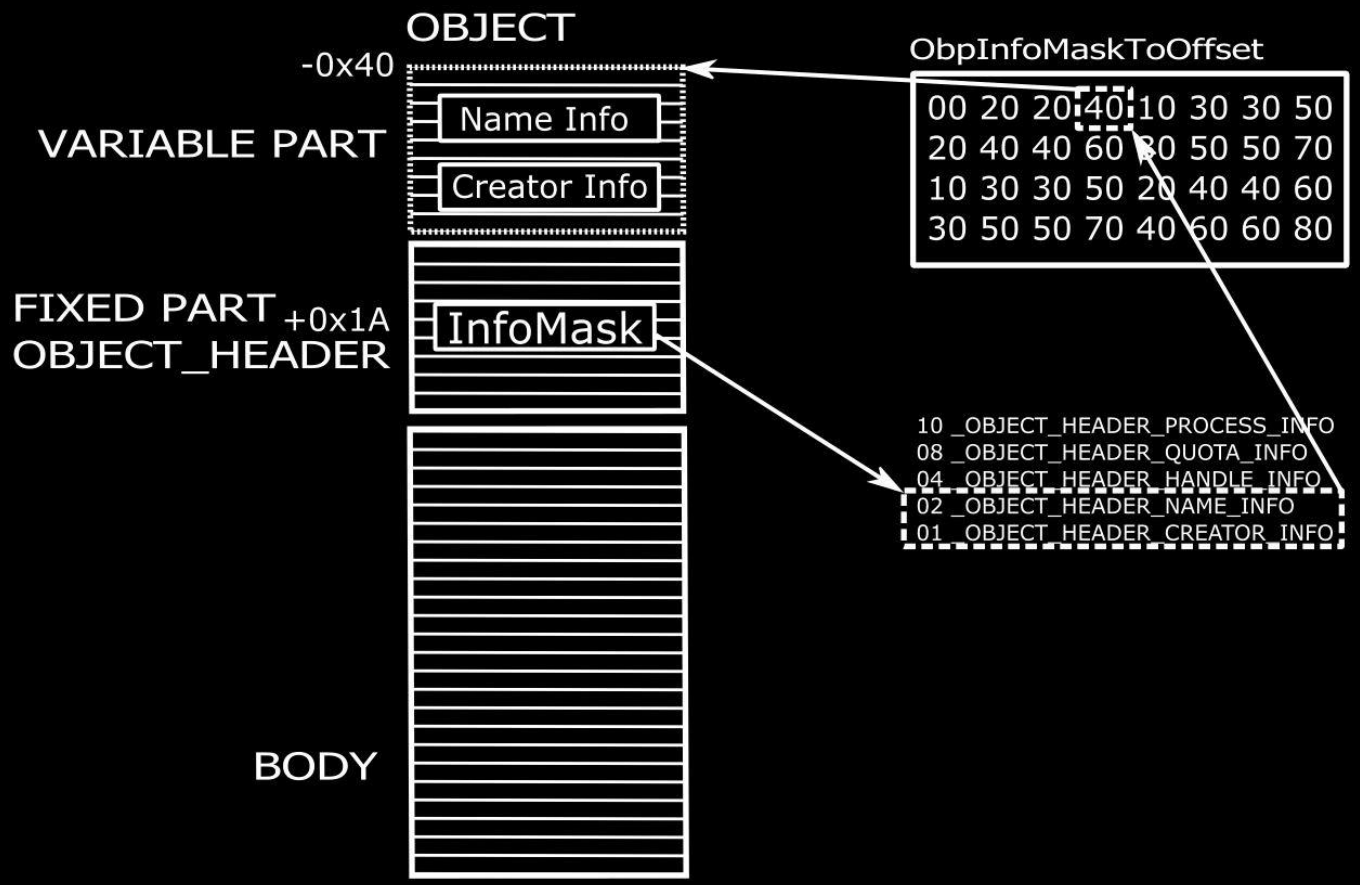
Object Layout



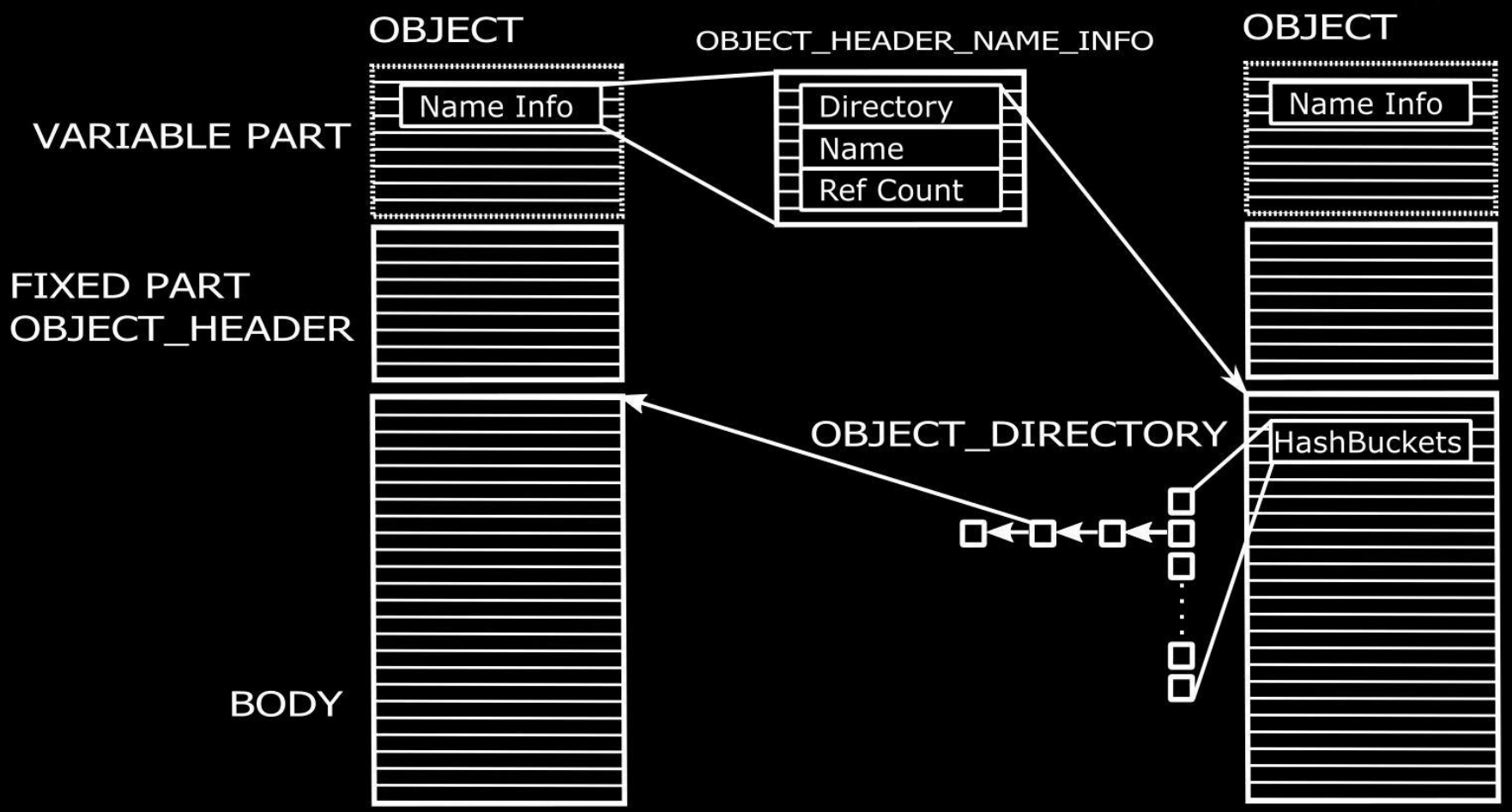
Security Descriptor Layout



Object Layout

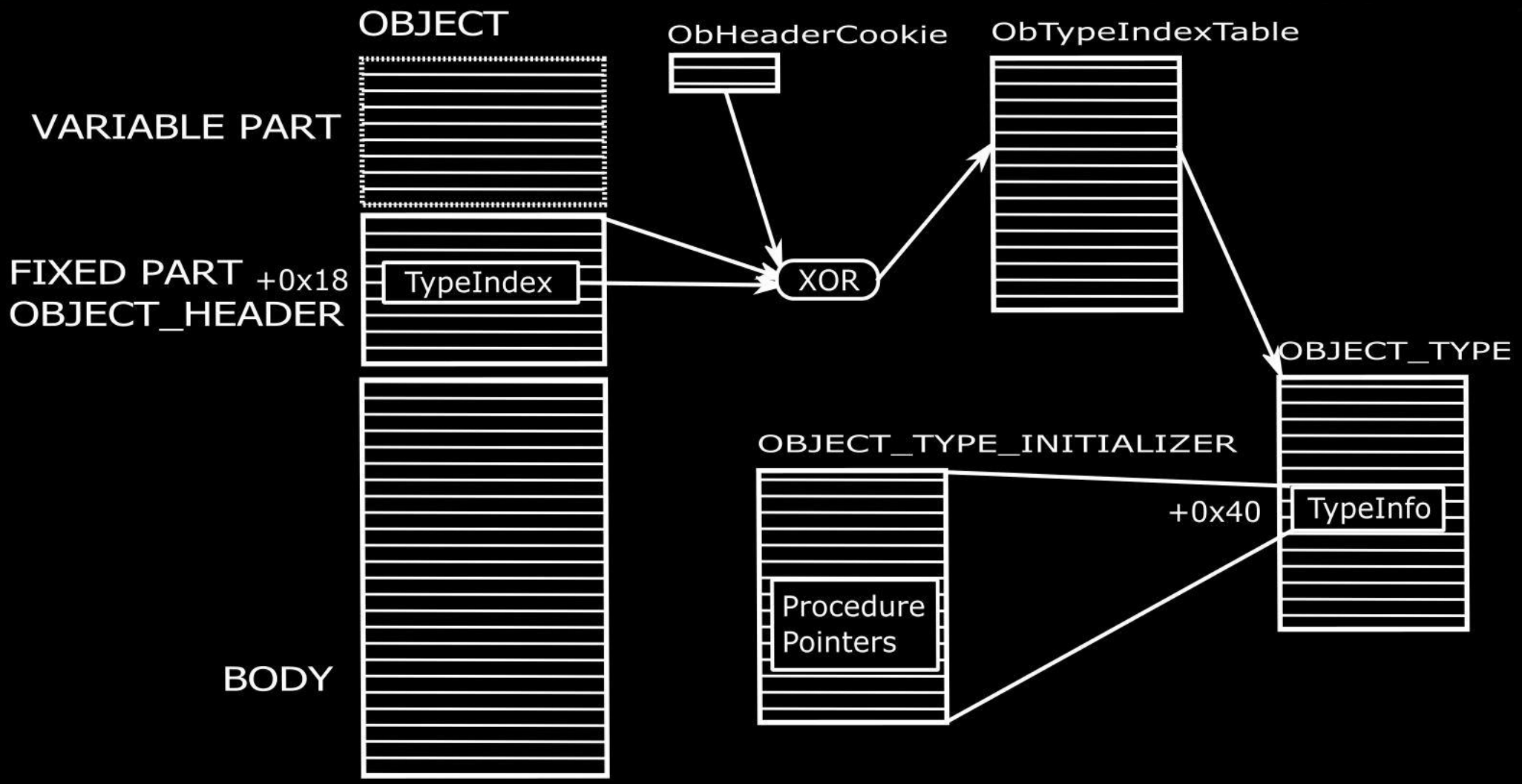


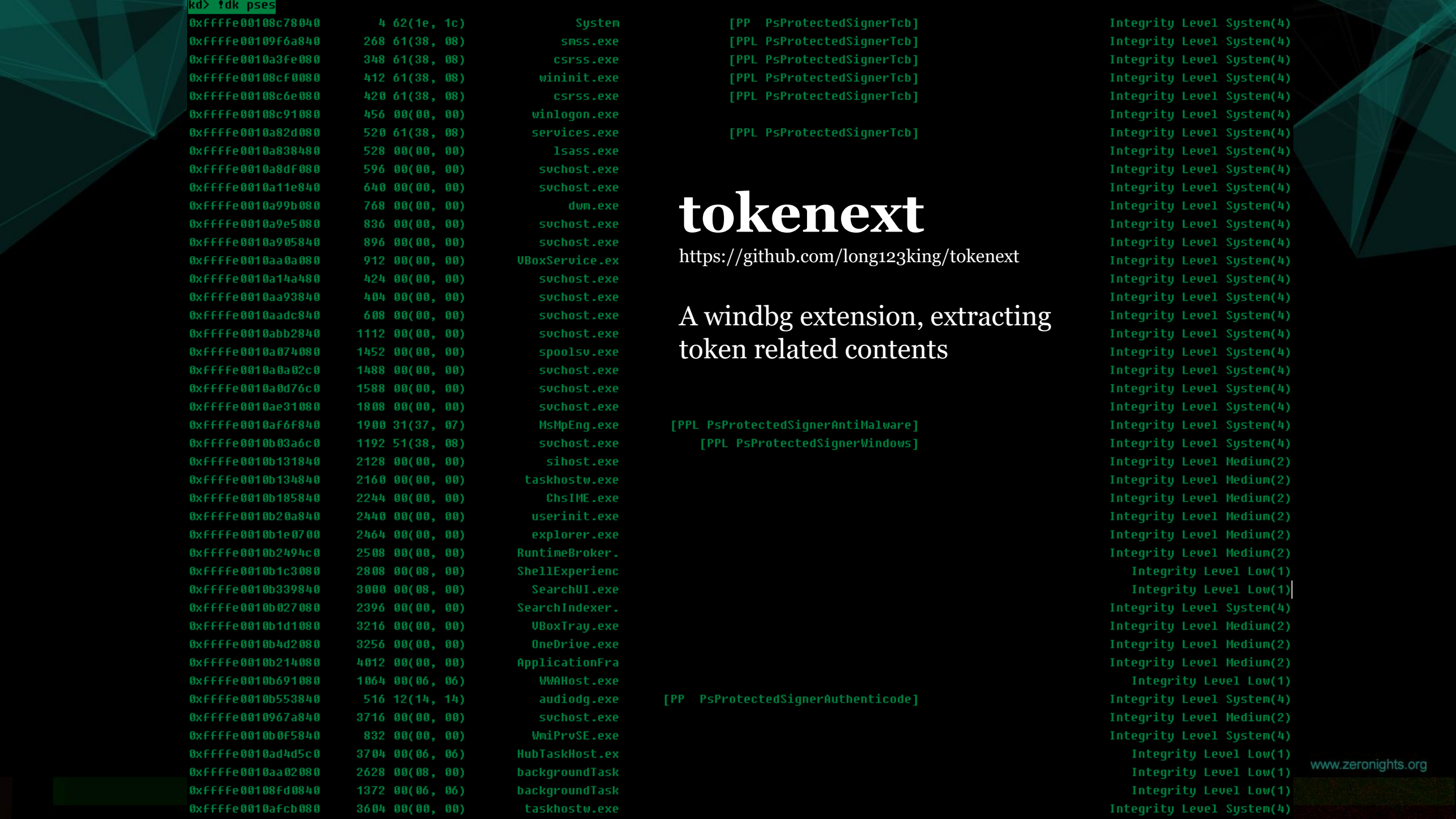
Object Directory Layout



Object Type

```
uint8_t CTokenExt::realIndex(size_t type_index, size_t obj_hdr_addr)
{
    uint8_t byte_2nd_addr = obj_hdr_addr >> 8;
    return type_index ^ m_ob_header_cookie ^ byte_2nd_addr;
}
```





tokenext

https://github.com/long123king/tokenext

A windbg extension, extracting token related contents

Protected Process

```
RTL_PROTECTED_ACCESS RtlProtectedAccess[] =
{
//   Domination,       Process,         Thread,
//         Mask,  Restrictions,   Restrictions,
    {         0,             0,             0}, //PsProtectedSignerNone             Subject To Restriction Type
    {         2,    0x000fc7fe,    0x000fe3fd}, //PsProtectedSignerAuthenticode      0y00000010
    {         4,    0x000fc7fe,    0x000fe3fd}, //PsProtectedSignerCodeGen           0y00000100
    {         8,    0x000fc7ff,    0x000fe3ff}, //PsProtectedSignerAntimalware       0y00001000
    {      0x10,    0x000fc7ff,    0x000fe3ff}, //PsProtectedSignerLsa               0y00010000
    {      0x3e,    0x000fc7fe,    0x000fe3fd}, //PsProtectedSignerWindows           0y00111110
    {      0x7e,    0x000fc7ff,    0x000fe3ff}, //PsProtectedSignerTcb               0y01111110
};
```

PspCheckForInvalidAccessByProtection

If the Host should be subject to Target's Restrictions?

- ☐ Kernel Mode Host
- ☐ Target Not Protected
- ☐ PP Host
- ☐ PPL Host, PPL Target
  Host Signer Dominates Guest Signer
- ☑ Others

	RESTRICTIONS	PASSES	ALLOWED ACCESS
PROCESS	0x000fc7fe	0x00003801	PROCESS_SET_LIMITED_INFORMATION PROCESS_QUERY_LIMITED_INFORMATION PROCESS_SUSPEND_RESUME PROCESS_TERMINATE
	0x000fc7ff	0x00003800	PROCESS_SET_LIMITED_INFORMATION PROCESS_QUERY_LIMITED_INFORMATION PROCESS_SUSPEND_RESUME
THREAD	0x000fe3fd	0x00001c02	THREAD_RESUME THREAD_QUERY_LIMITED_INFORMATION THREAD_SET_LIMITED_INFORMATION THREAD_SUSPEND_RESUME
	0x000fe3ff	0x00001c00	THREAD_RESUME THREAD_QUERY_LIMITED_INFORMATION THREAD_SET_LIMITED_INFORMATION

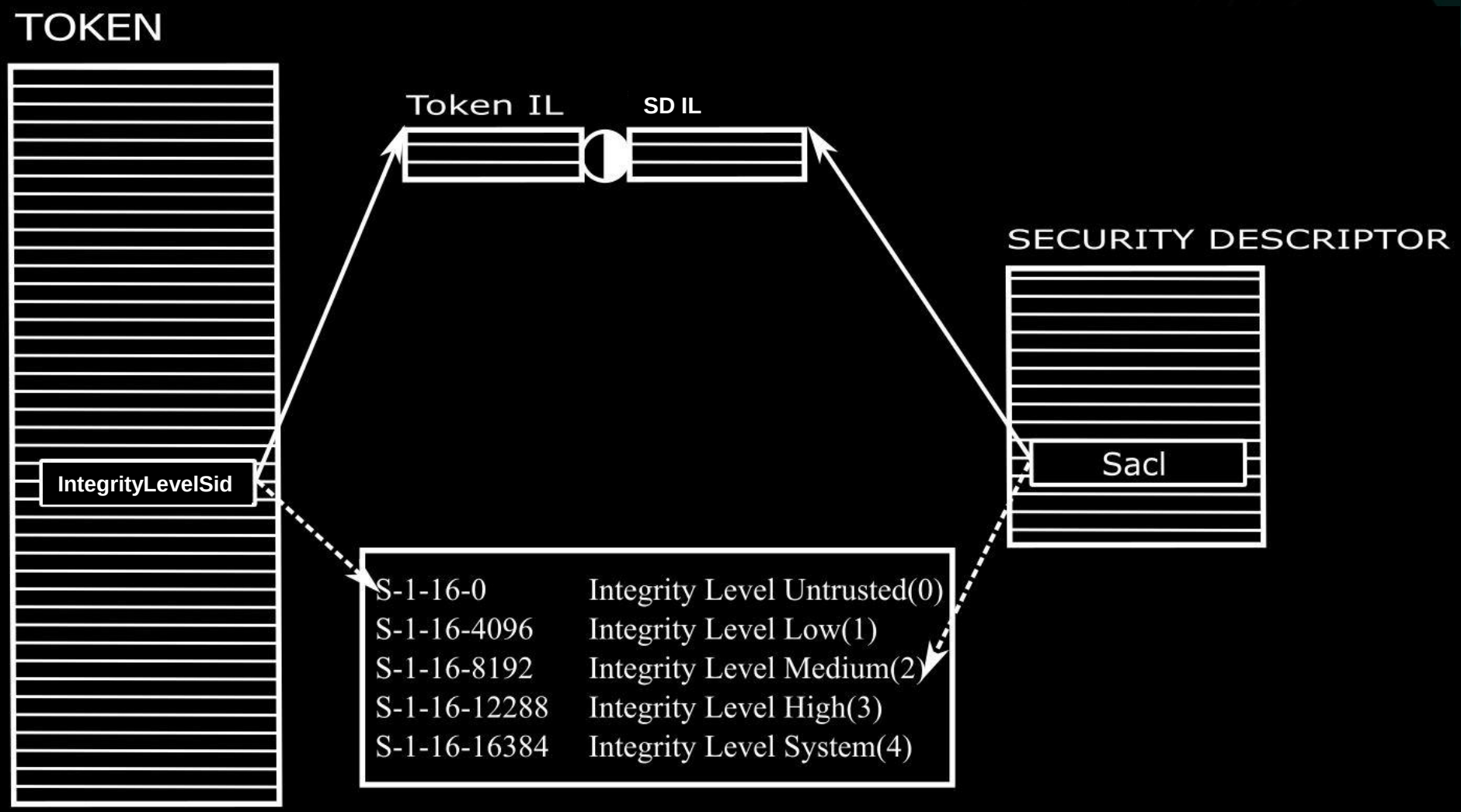
Protected Process

```
static
NTSTATUS
NtDebugActiveProcess(
    __in HANDLE ProcessHandle,
    __in HANDLE DebugObjectHandle
    )
{
    PEPROCESS target_process = nullptr;
    NTSTATUS result = ObReferenceObjectByHandleWithTag(ProcessHandle, &target_process);
    if (!NT_SUCCESS(result))
        return result;

    PEPROCESS host_process = PsGetCurrentProcess();

    if (host_process == target_process)
        return 0xC0000022;

    if (PsInitialSystemProcess == target_process)
        return 0xC0000022;

    if (PspCheckForInvalidAccessByProtection(PreviousMode, host_process->Protection, target_process->Protection))
        return 0xC0000712;

    // ......
}
```

Protected Process

```
static
NTSTATUS
NtCreateUserProcess(
    __out PHANDLE ProcessHandle,
    __out PHANDLE ThreadHandle,
    __in  ACCESS_MASK ProcessDesiredAccess,
    __in  ACCESS_MASK ThreadDesiredAccess,
    __in  POBJECT_ATTRIBUTES ProcessObjectAttributes OPTIONAL,
    __in  POBJECT_ATTRIBUTES ThreadObjectAttributes OPTIONAL,
    __in  ULONG CreateProcessFlags,
    __in  ULONG CreateThreadFlags,
    __in  PRTL_USER_PROCESS_PARAMETERS ProcessParameters,
    __in  PVOID Parameter9,
    __in  PNT_PROC_THREAD_ATTRIBUTE_LIST AttributeList
    )
{
    ACCESS_MASK allowed_process_access = ProcessDesiredAccess;
    ACCESS_MASK allowed_thread_access = ThreadDesiredAccess;
    PS_PROTECTION protection = ProcessContext.member_at_0x20;
    if (PspCheckForInvalidAccessByProtection(PreviousMode, host_process->Protection, target_process->Protection))
    {
        // 1 << 0x19 = 0x80000,  WRITE_OWNER
        if (MAXIMUM_ALLOWED == ProcessDesiredAccess)
            allowed_process_access = (((~RtlProtectedAccess[protection.Signer].DeniedProcessAccess) & 0x1FFFFF) | ProcessDesiredAccess) & (~(1 << 0x19));

        if (MAXIMUM_ALLOWED == ThreadDesiredAccess)
            allowed_thread_access = (((~RtlProtectedAccess[protection.Signer].DeniedThreadAccess) & 0x1FFFFF) | ThreadDesiredAccess) & (~(1 << 0x19));
    }
    //PspInsertProcess(..., allowed_process_access, ...);
    //PspInsertThread(..., allowed_thread_access, ...);
}
```

Sandbox


TOKEN
Token IL
SD IL
SECURITY DESCRIPTOR
IntegrityLevelSid
Sacl
S-1-16-0 Integrity Level Untrusted(0)
S-1-16-4096 Integrity Level Low(1)
S-1-16-8192 Integrity Level Medium(2)
S-1-16-12288 Integrity Level High(3)
S-1-16-16384 Integrity Level System(4)

Another 2 kinds of sandbox

1. Sandbox based on **AppContainer** and its **Capabilities Sid**

   Windows Apps and **IE Enhanced Protected Mode** are built upon this kind of sandbox

2. Sandbox based on **Trust Level**

   \Windows\SharedSection [0x61: Trust Label Lite(PPL) PsProtectedSignerTcb(6)]

   \KnownDlls32* [0x61: Trust Label Lite(PPL) PsProtectedSignerTcb(6)]

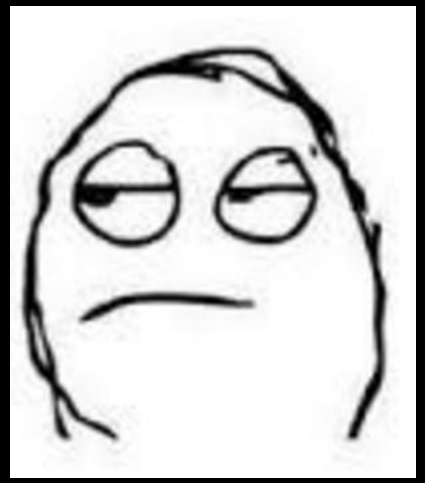
   \KnownDlls* [0x61: Trust Label Lite(PPL) PsProtectedSignerTcb(6)]

   Token Object of System Process [0x62: Trust Label Protected(PP) PsProtectedSignerTcb(6)]

Browser Sandbox

iexplore.exe	Medium
iexplore.exe	Low

IE Protected Mode



RuntimeBroker.exe	Medium
MicrosoftEdgeCP....	AppContainer
MicrosoftEdgeCP....	AppContainer
MicrosoftEdge.exe	AppContainer
browser_broker.exe	Medium

Edge

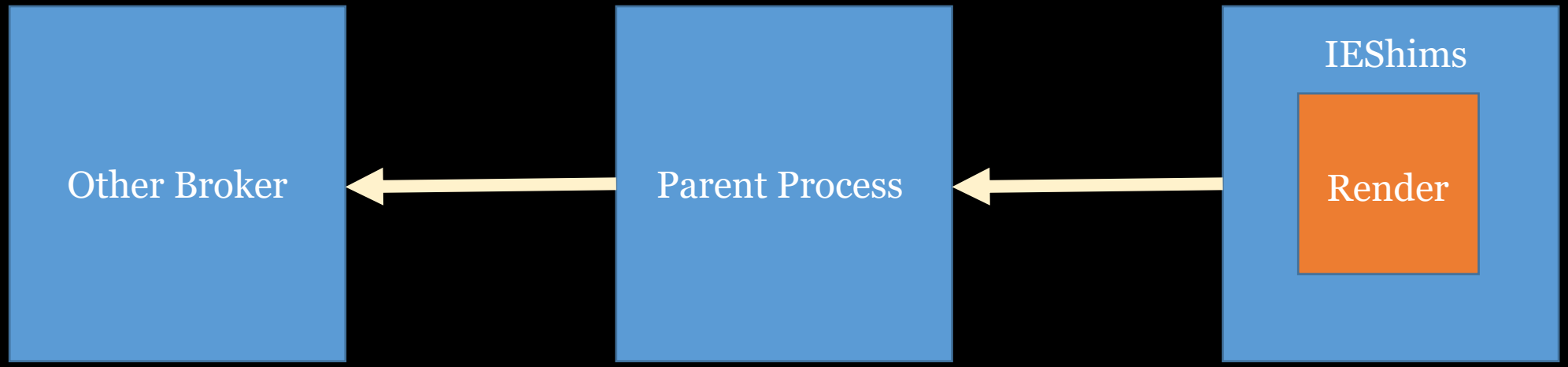
iexplore.exe	Medium
iexplore.exe	AppContainer

IE Enhanced Protected Mode

chrome.exe	Medium
chrome.exe	Low
chrome.exe	Untrusted

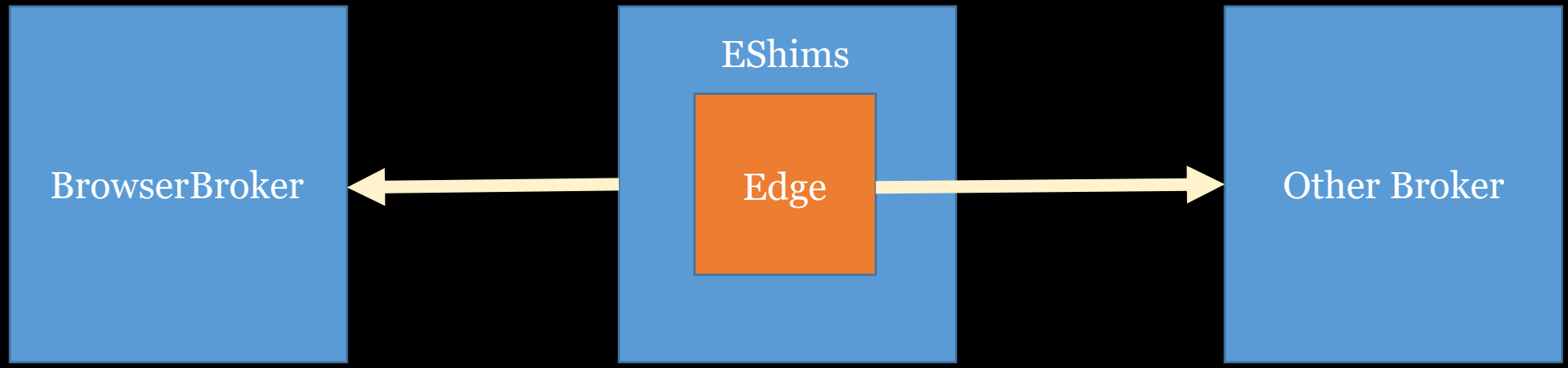
Chrome

The way a token from broker to render



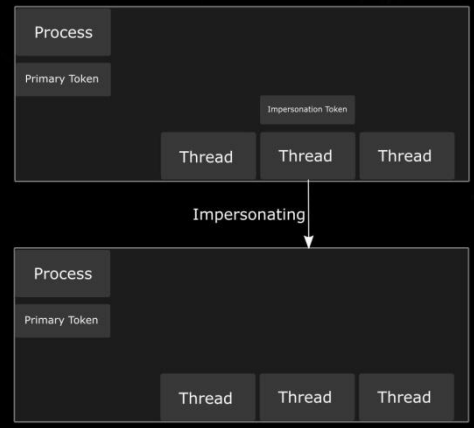
Internet Explorer 11

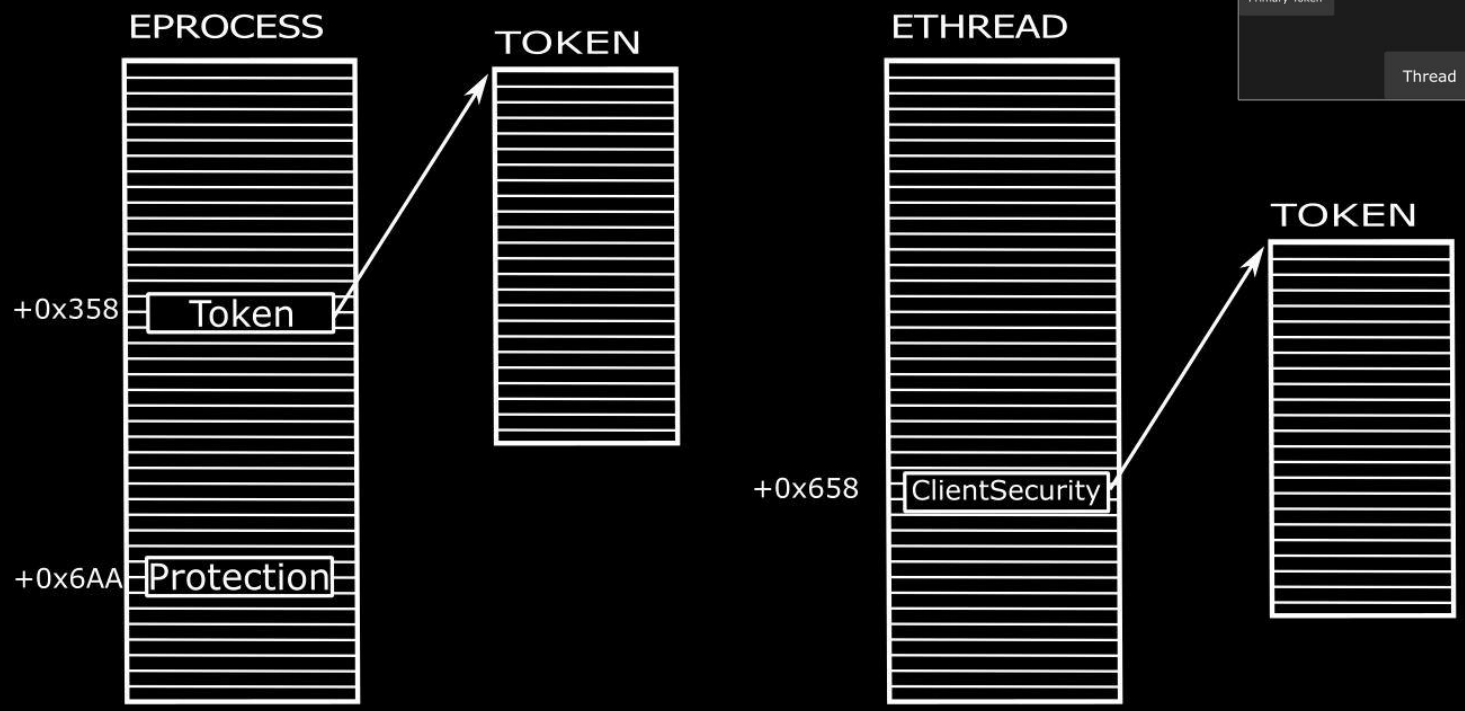
The way a token from broker to render



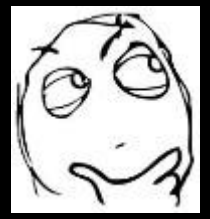
Edge

Token


Process
Primary Token
Impersonation Token
Thread
Thread
Thread
Impersonating
Process
Primary Token
Thread
Thread
Thread


EPROCESS
TOKEN
+0x358
Token
+0x6AA
Protection
ETHREAD
TOKEN
+0x658
ClientSecurity

Is there any way to escape sandbox logically?



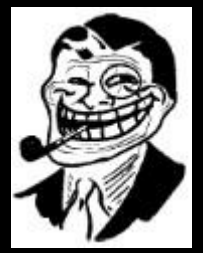
Symlink?

Fixed in APSB-15-09

```
for ( i = StrRStrIW(&Source, 0, L"\\"); ; i = StrRStrIW(&Source, lpLasta, L"\\") )
{
  lpLasta = i;
  if ( !i )
    break;
  *i = 0;
  sub_1002921C((int)&FileName, (int)&pLinkName, (int)&Source, 0, 0);
  *lpLasta = 92;
  if ( GetFileAttributesW(&FileName) & FILE_ATTRIBUTE_REPARSE_POINT )
  {
    hObject = CreateFileW(&FileName, GENERIC_READ, 0, 0, OPEN_EXISTING, 0x2200000u, 0);
    if ( hObject != (HANDLE)-1 )
    {
      v23 = lstrlenW(&lpPureFileName);
      lpFirstb = (LPCWSTR)(2 * (lstrlenW(&lpExtension) + v23) + 0x4000);
      if ( (unsigned int)lpFirstb >= 0xFFFFFFFF )
        sub_100290F8();
      sub_10029633();
      v17 = v24;
      if ( v24 )
      {
        v25 = GetRepasePoint(hObject, v24);// \??\C:\Users\Azure\AppData\Local\Temp\Low\dtpmicueigsemwng
        v18 = (int)v25;
        if ( v25 )
        {
          sub_10011934((int)v25, (int)lpFirstb, (int)lpLasta);
          sub_10011934(v18, (int)lpFirstb, (int)&lpPureFileName);
          sub_10011934(v18, (int)lpFirstb, (int)&lpExtension);
        }
      }
      CloseHandle(hObject);
      goto LABEL_26;
    }
  }
}
```

Did you really get your token?

NO, this is my file!



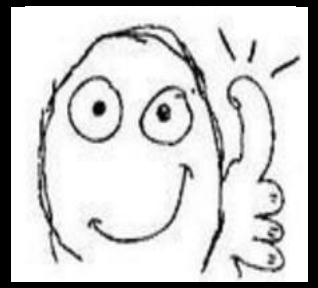
Code after fix

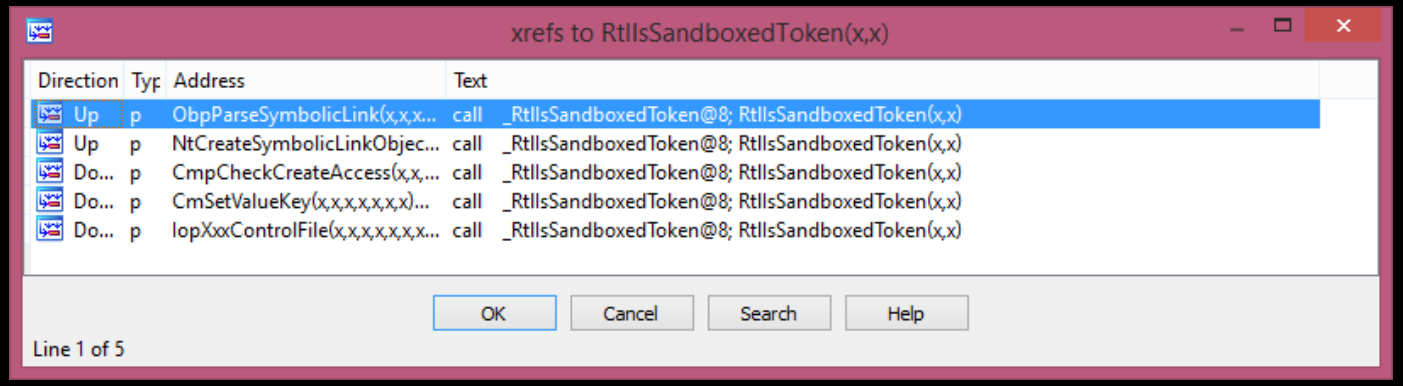
You need more elegant way?

```
hFile = CreateFile(argv[1],               // file to open
    GENERIC_READ,          // open for reading
    FILE_SHARE_READ,       // share for reading
    NULL,                  // default security
    OPEN_EXISTING,         // existing file only
    FILE_ATTRIBUTE_NORMAL, // normal file
    NULL);                 // no attr. template

if (hFile == INVALID_HANDLE_VALUE)
{
    printf("Could not open file (error %d\n)", GetLastError());
    return;
}

dwRet = GetFinalPathNameByHandle(hFile, Path, BUFSIZE, VOLUME_NAME_NT);
if (dwRet < BUFSIZE)
{
    _tprintf(TEXT("\nThe final path is: %s\n"), Path);
}
else printf("\nThe required buffer size is %d.\n", dwRet);

CloseHandle(hFile);
```

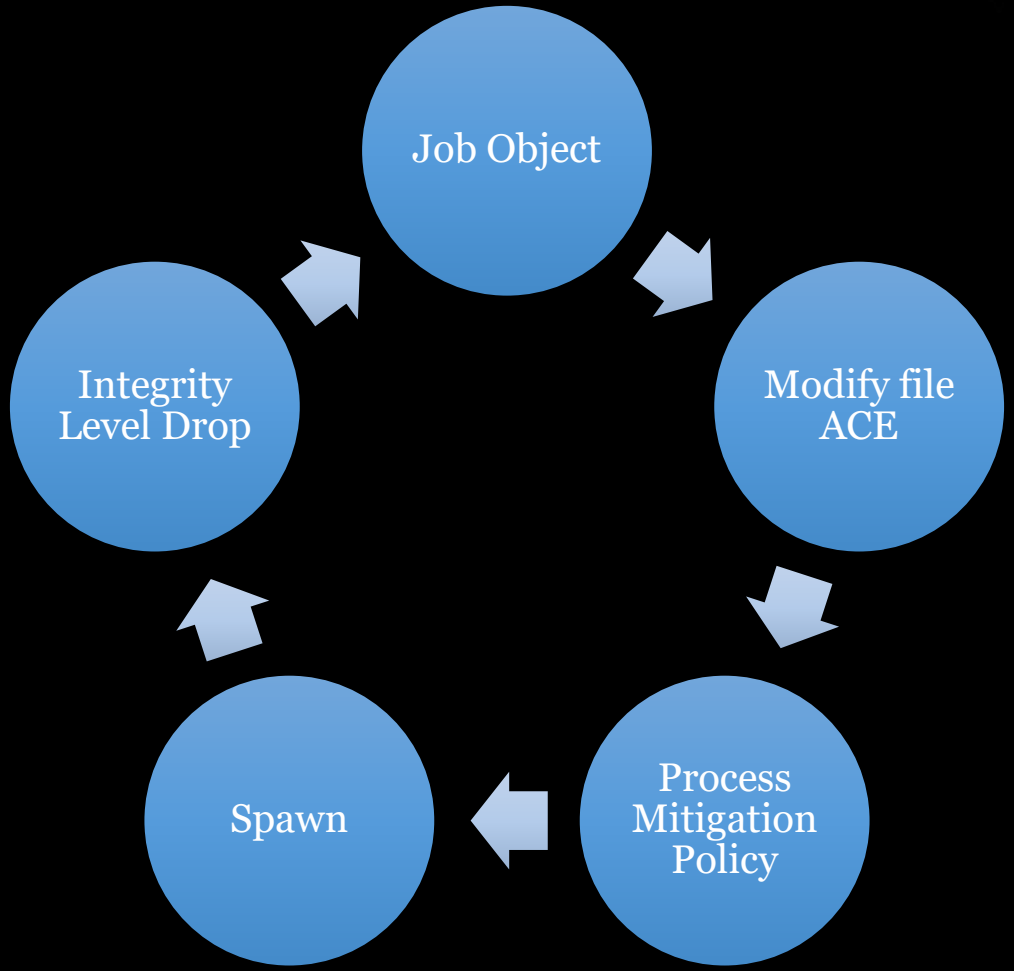
Mitigations about sandbox bypass

Finally fixed in MS15-090

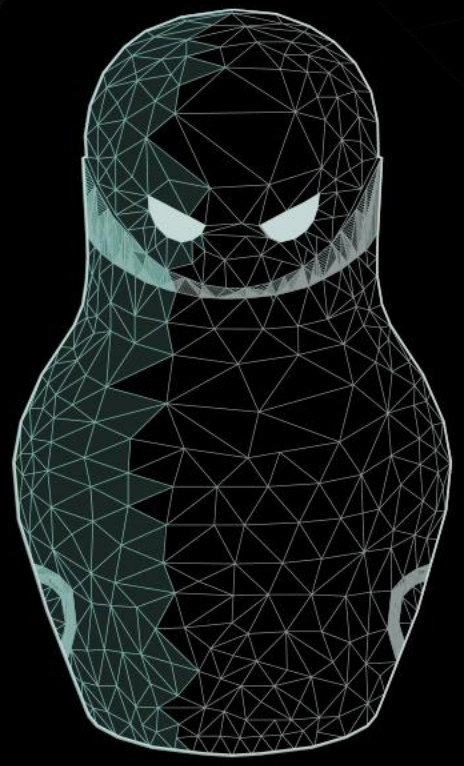




How to make use of Windows sandbox?



https://github.com/trailofbits/AppJailLauncher



ZERO NIGHTS 2015

Questions?

Special thanks

Jihui Lu (@promised_lu)
Alex Ionescu(@aionescu)
James Forshaw (@tiraniddo)
Peter Hlavaty (@zeromem)
Liang Chen (@chenliang0817)

All KeenTeam members and you