



THE WINDOWS
Sandbox Paradox
FLASHBACK

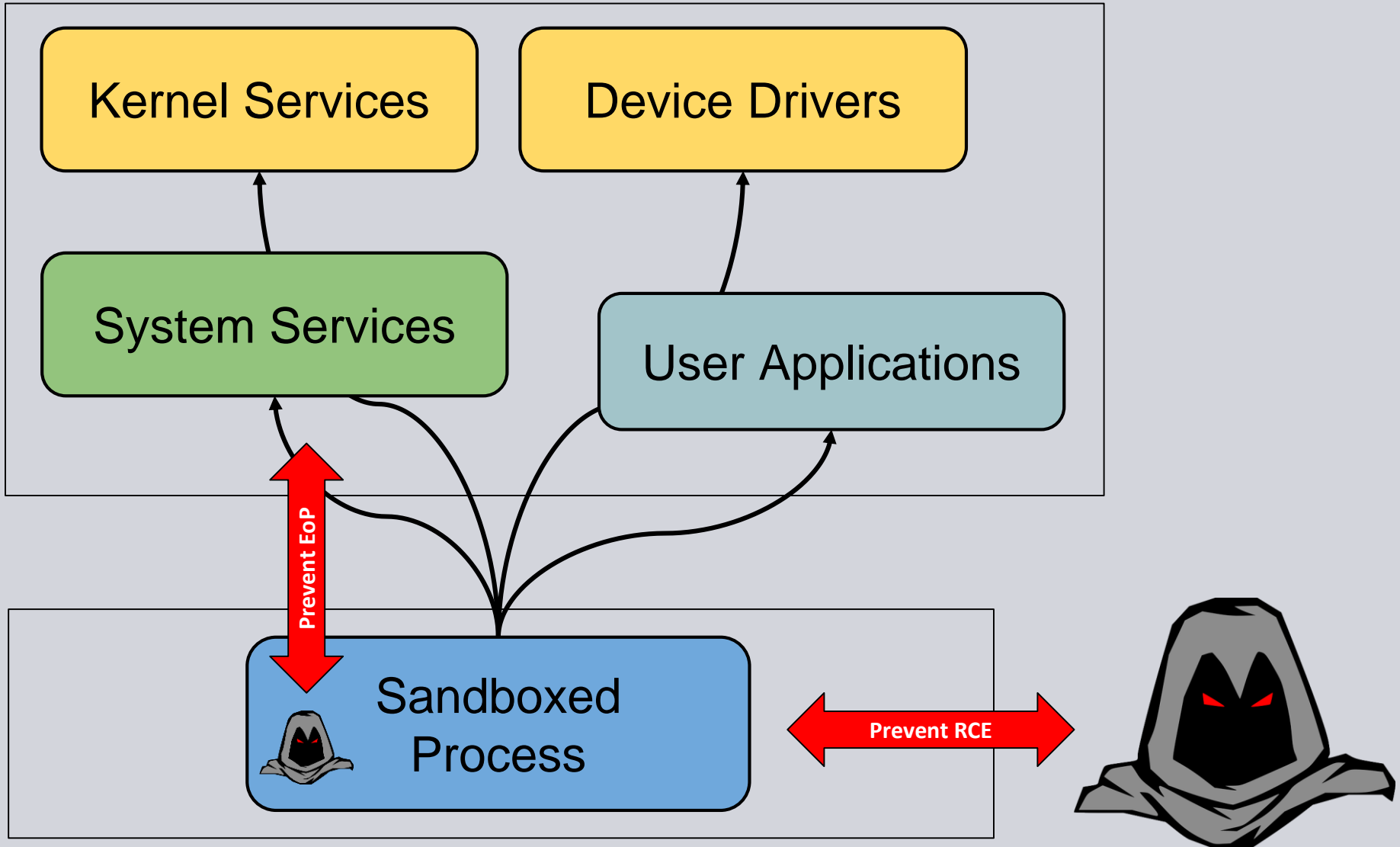
Nullcon 2019

James Forshaw @tiraniddo

Obligatory Background Slide

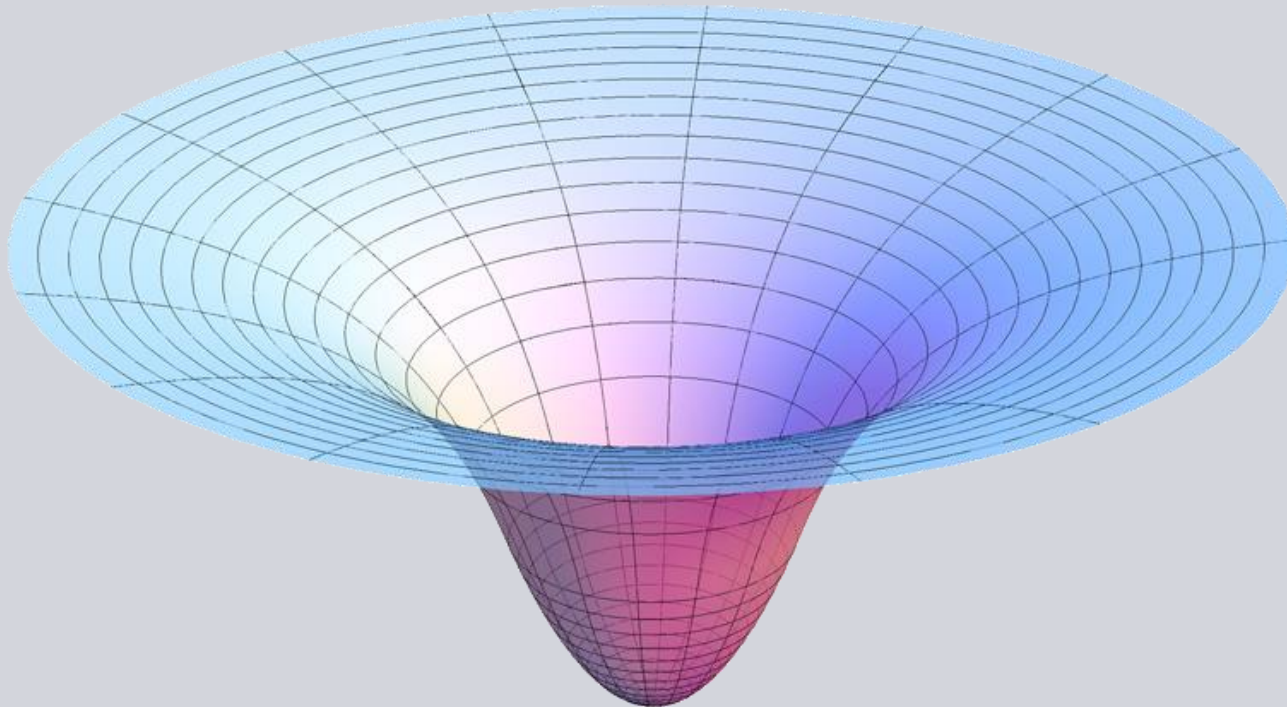
- Founder Member of Google's Project Zero
- 10+ Years of Windows Security Research
- Logical vulnerability specialist
- Chromium Windows Sandbox Owner
- "Attacking Network Protocols" Author
- @tiraniddo on Twitter.

What I'm Going to Talk About



Sandboxing Requirement #1

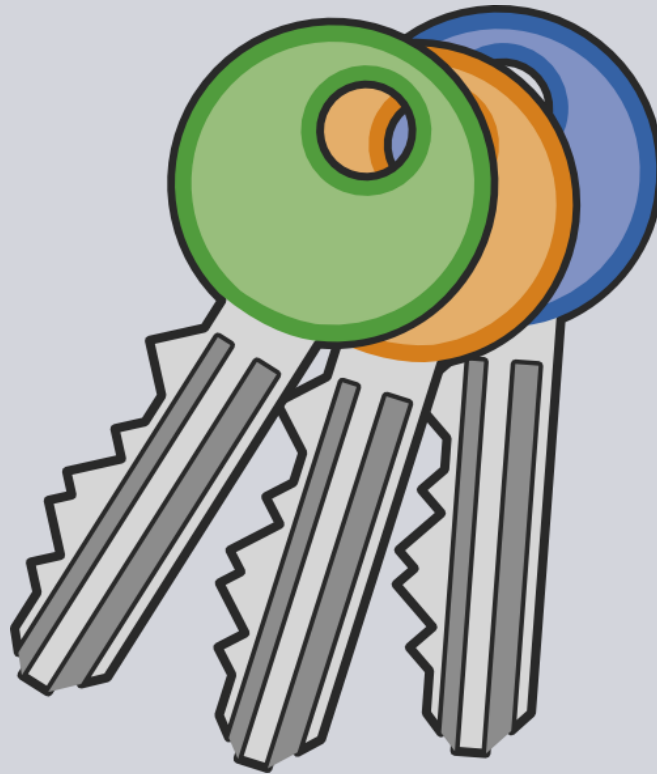
- Easy to get in, hard to get out



<http://upload.wikimedia.org/wikipedia/commons/d/d9/GravityPotential.jpg>

Sandboxing Requirement #2

- Protects the user's data from disclosure



<https://openclipart.org/detail/190821/cles-de-serrure---lock-keys-by-enolynn-190821>

Sandboxing Requirement #3

- Work within the limits of the OS



http://upload.wikimedia.org/wikipedia/commons/8/8b/MUTCD_R2-1.svg

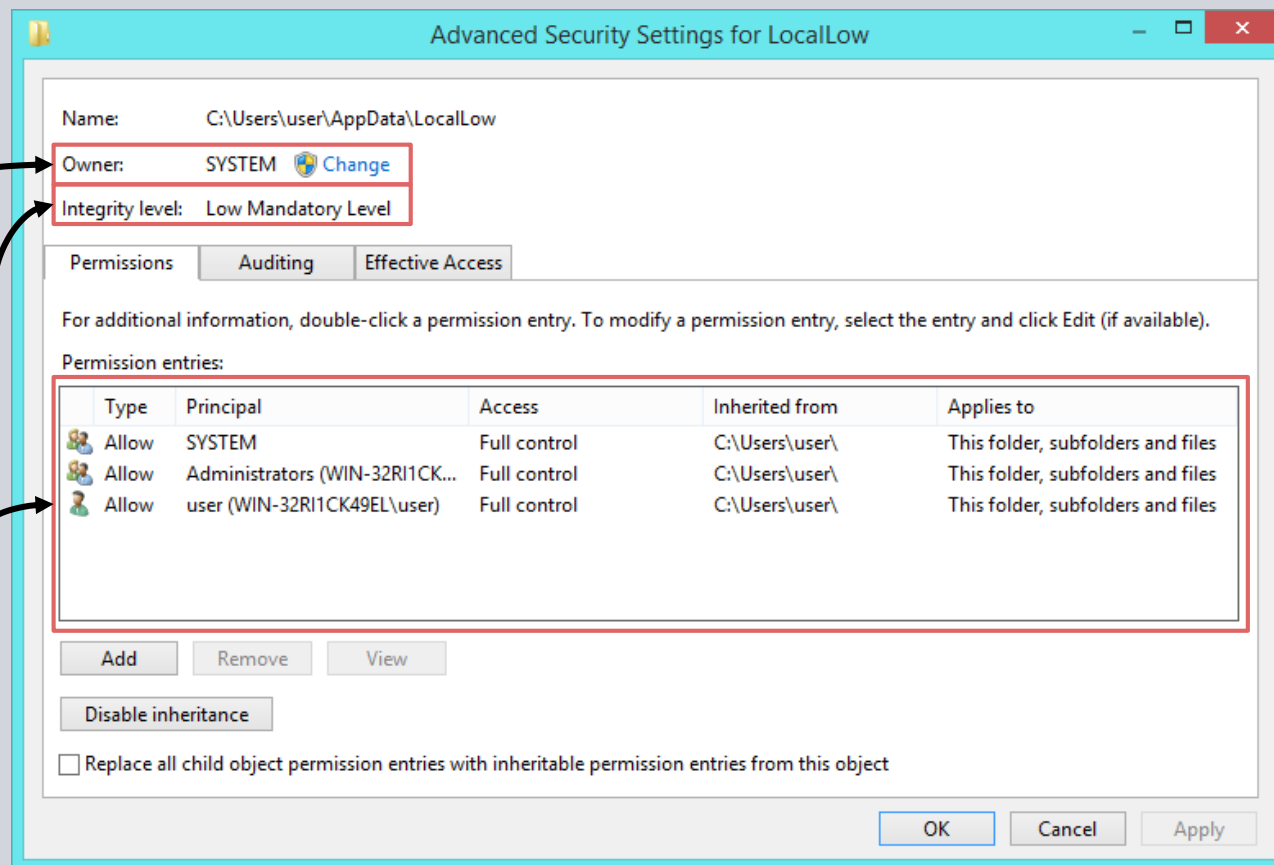
Sandboxing Requirement #4

- Sandboxed application is usable



<http://pixabay.com/p-305189/>

Resource Security Descriptor



Owner of Secured Resource

Mandatory Integrity Label

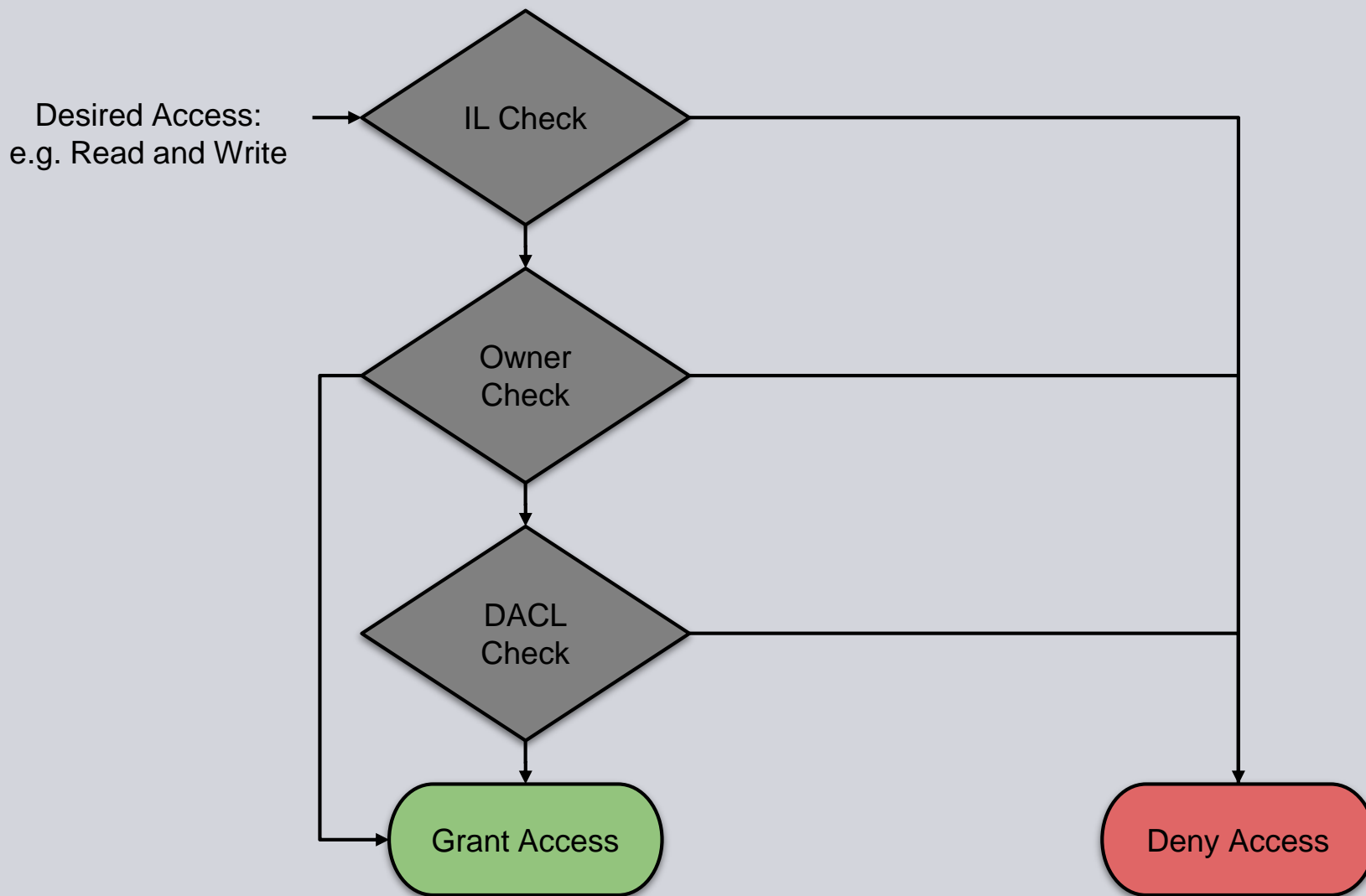
Discretionary Access Control List (DACL)

Access Tokens

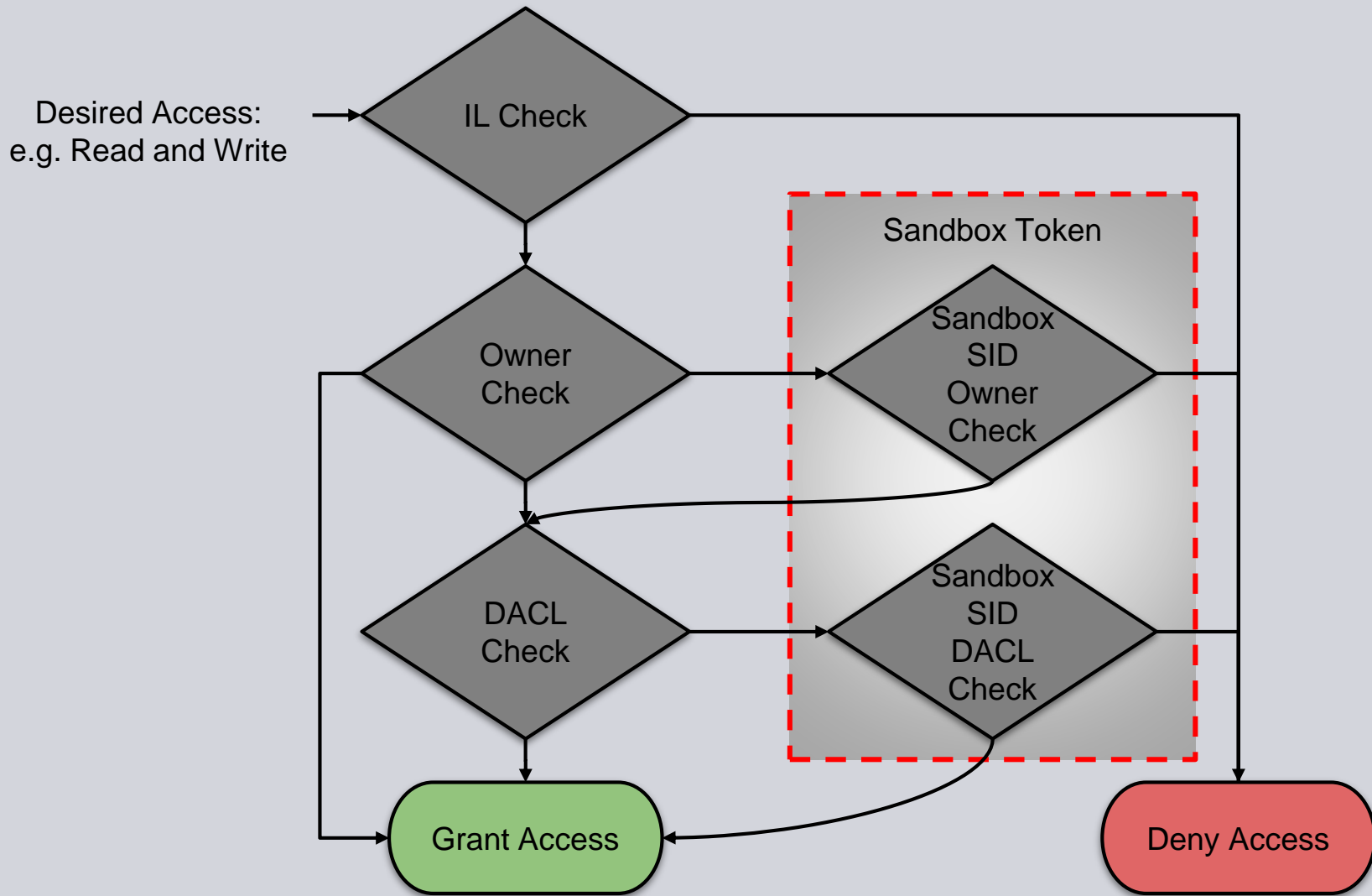
The screenshot shows the 'chrome.exe:2716 Properties' dialog box with the 'Security' tab selected. The 'User' field is 'WIN-32RI1CK49EL\user' and the 'SID' is 'S-1-5-21-3711643808-3202222375-1035956708-1001'. The 'Groups' list includes 'BUILTIN\Administrators', 'BUILTIN\Users', 'CONSOLE LOGON', 'Everyone', 'LOCAL', 'Logon SID (S-1-5-5-0-5071873)', 'Mandatory Label\Medium Mandatory Level', and 'NT AUTHORITY\Authenticated Users'. The 'Privilege' list includes 'SeChangeNotifyPrivilege', 'SeIncreaseWorkingSetPrivilege', 'SeShutdownPrivilege', 'SeTimeZonePrivilege', and 'SeUndockPrivilege'. Annotations on the left side of the image point to these specific fields with the following labels:

- User Security Identifier
- Groups
- Mandatory Label
- Privileges

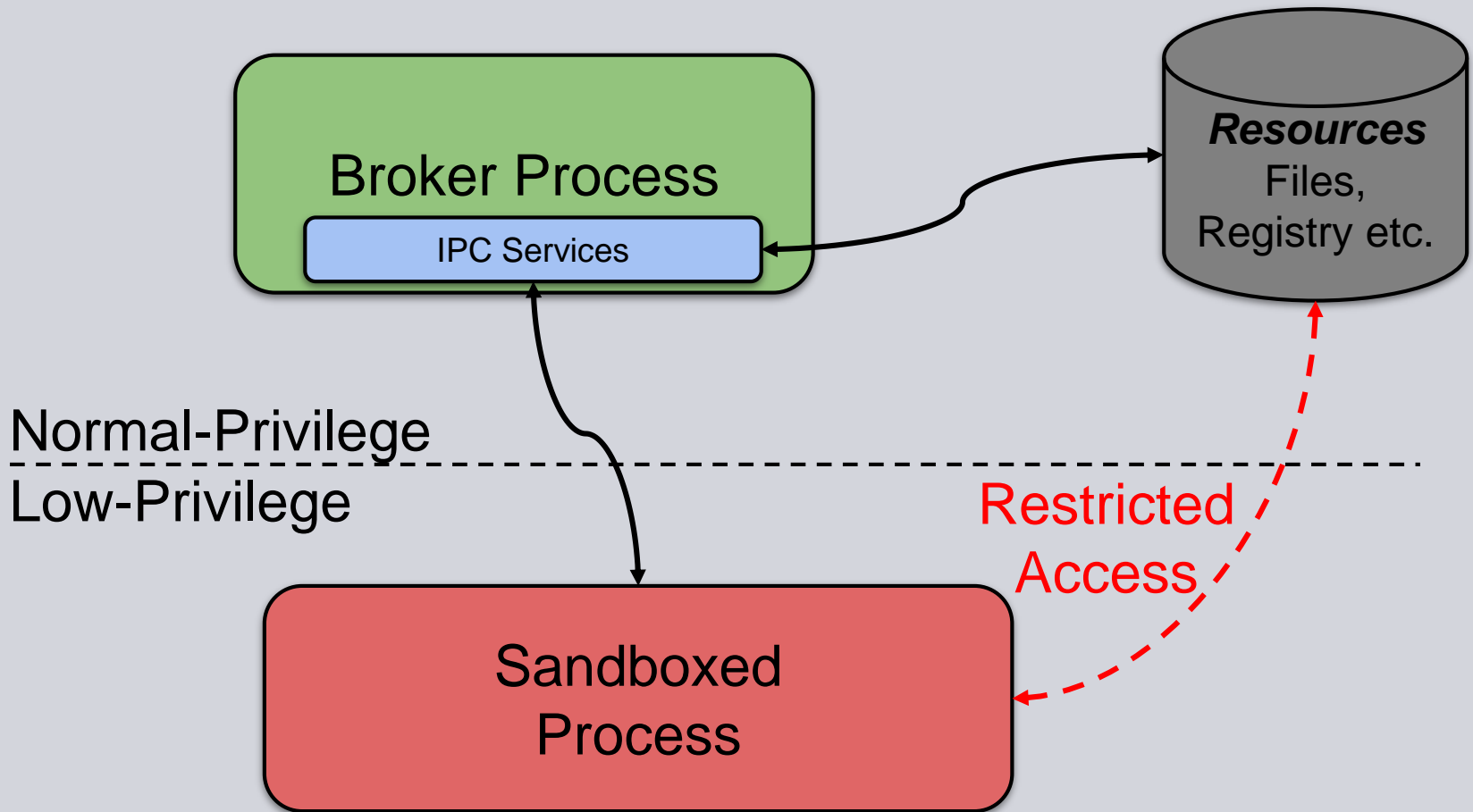
Access Check



Sandbox Access Check



Typical User-Mode Approach



The Windows Sandbox Paradox

Kernel Attack Surface

~300 Syscalls ~400 Syscalls + ~1000 Win32k

http://nullcon.net the neXt security thing! nju CON

Crash!

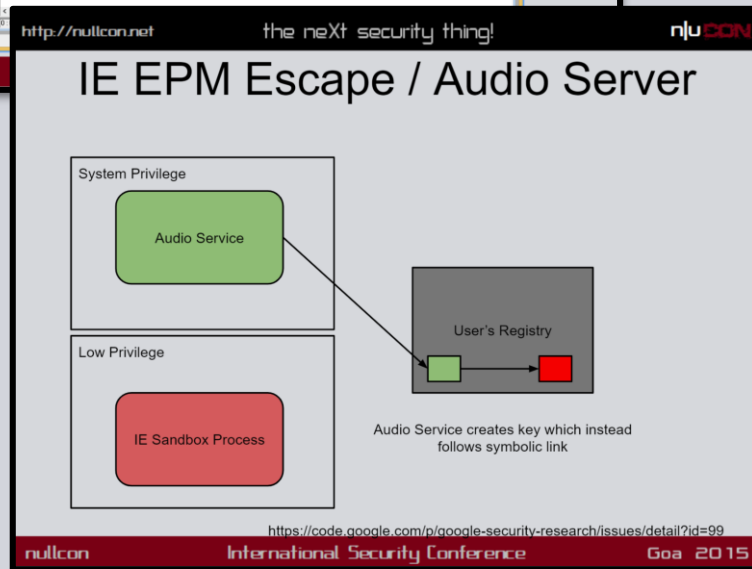
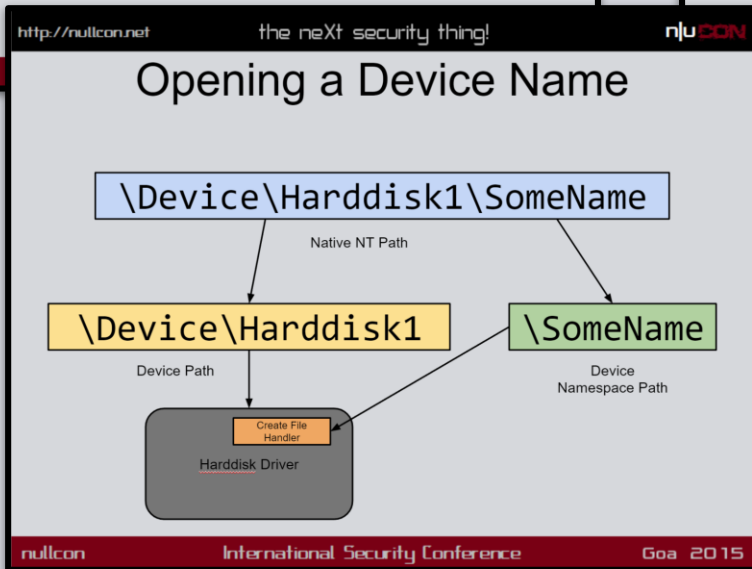
```

Command
WARNING: Whitespace at start of path element
***** Symbol Path validation summary *****
Response: Time (ms) Location:
Deferred: Time (ms) srv*
***** Symbol Path validation summary *****
Response: Time (ms) Location:
Deferred: Time (ms) srv*
Microsoft (R) Windows Debugger Version 6.3.9600.16384 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.

*** wait with pending attach

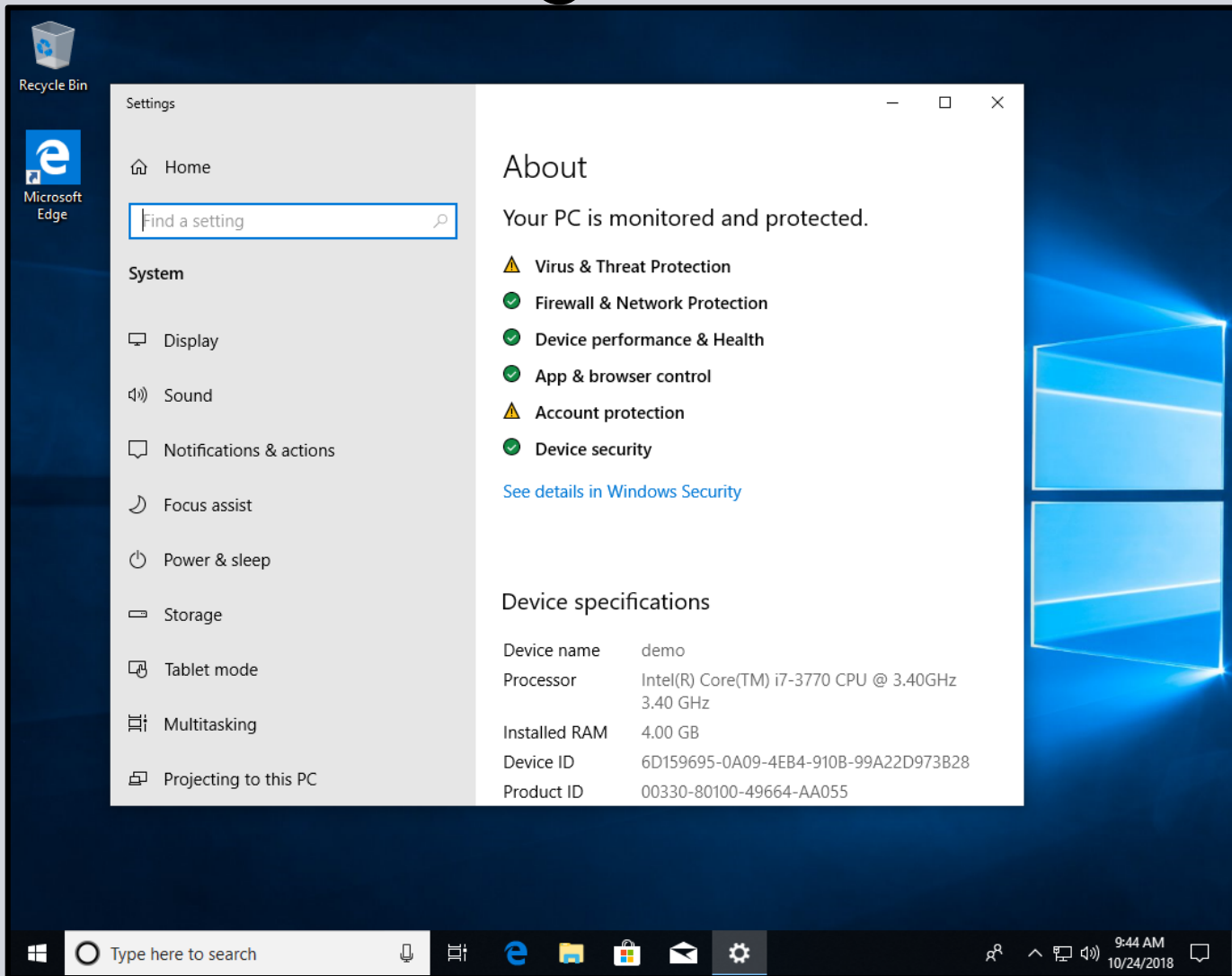
***** Symbol Path validation summary *****
Response: Time (ms) Location:
Deferred: Time (ms) srv*
WARNING: Whitespace at start of path element
***** Symbol Path validation summary *****
Response: Time (ms) Location:
Deferred: Time (ms) srv*
Symbol search path is http://chromium-browser-sysrv.com\code\storage.googleapis.com:srv*
Executable search path is:
ModLoad: 0007f9f' 1429000 0007f9f' 14276000 c:\windows\system32\calc.exe
ModLoad: 0007f9f' 3040100 0007f9f' 3040100 c:\windows\system32\ntdll.dll
Breakpoint hit: waiting 30 seconds
WARNING: Break-in timed out: suspending
This is usually caused by another thread holding the loader lock
(3d0 188) Wake debugger = code 00000007 (first chance)
0007f9f: 30e2c04 48b8b8000000 xov rax.qword ptr [rbp+08b8] ss: 00000004'540af98+00007f9f30e2c04
0 00000000
Child-SP: 00000004'540af98 0007f9f: 30d8336 atdll!RtlTimeStatus+0x20
00000004'540af98 0007f9f: 30d8336 atdll!_LdrInitializeThunk
00000004'540af98 00000000 00000000 atdll!LdrInitializeThunk
  
```

http://nullcon.net the neXt security thing! nju CON

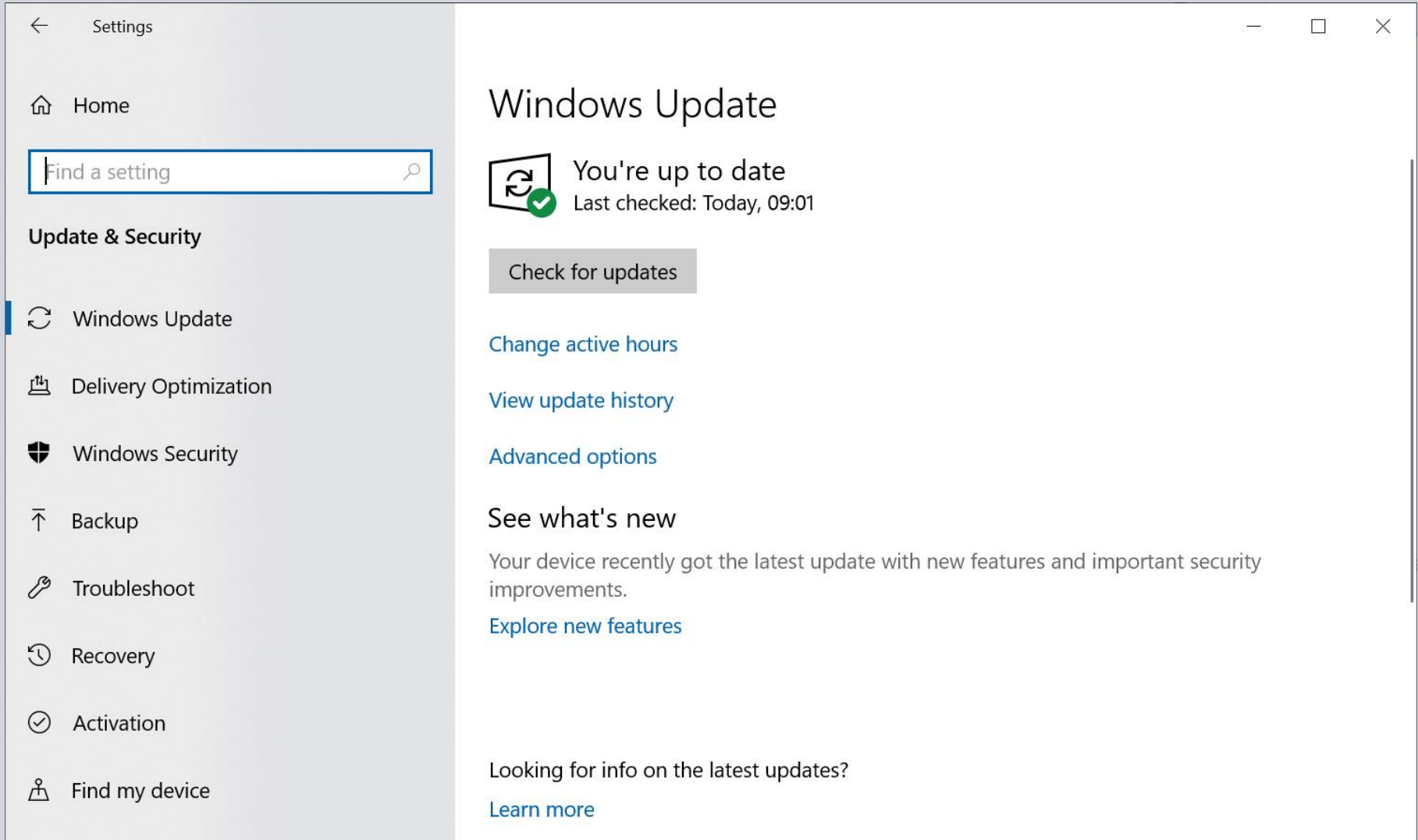


Welcome to 2019

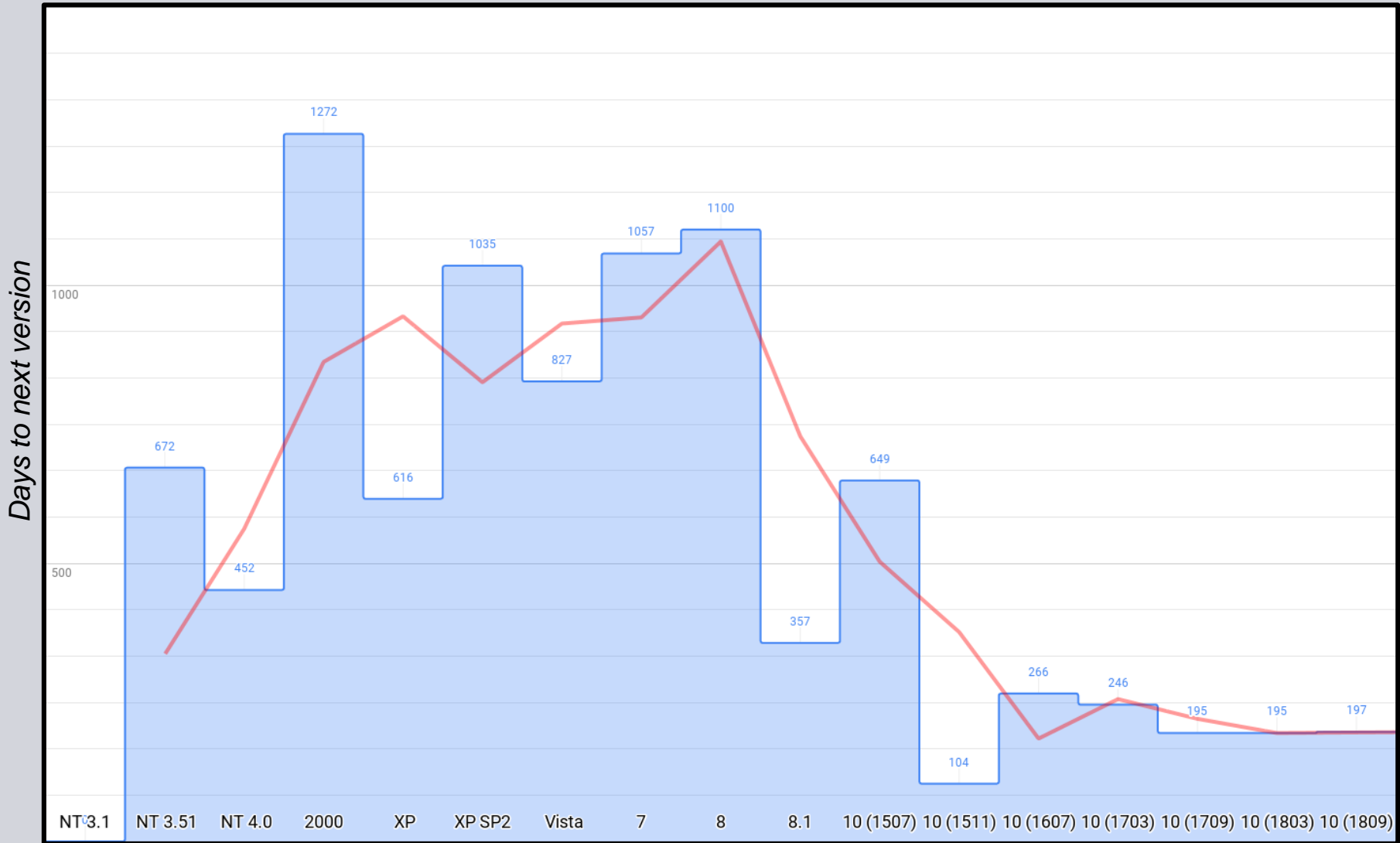
Introducing Windows 10



You're Going to Update



Time to Release is Shorter



Microsoft Edge

The screenshot shows the Microsoft Edge Tips website in a browser window. The address bar displays `https://microsoftedgetips.microsoft.com/en-gb/`. A cookie notice is visible at the top. The main content area features a 'Your progress' section with '0% complete', a 'Microsoft Edge Tips' title, and a 'Here's what's new!' section with a green leaf icon. Two featured tips are shown in a green background:

- Quickly get to stuff**: A tip with a 'NEW!' badge. Below the text is a browser toolbar where the menu icon (three horizontal lines) is circled in orange.
- Stop videos from playing automatically**: A tip with a 'NEW!' badge. Below the text is an illustration of a laptop displaying a video player.

App Container Features

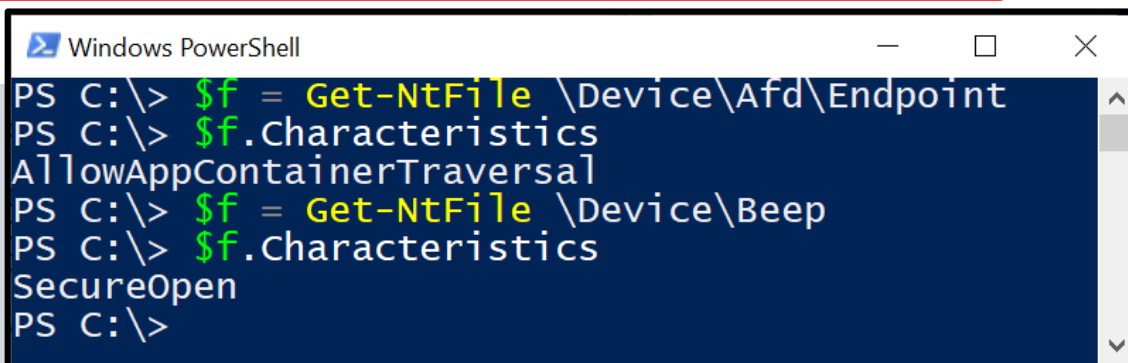
AC Device Attack Surface

```
BOOLEAN IopDoFullTraverseCheck(PDEVICE_OBJECT Device,  
    PSECURITY_SUBJECT_CONTEXT SubjectSecurityContext) {
```

```
    if (Device->Characteristics &  
        (FILE_DEVICE_ALLOW_APPCONTAINER_TRAVERSAL |  
         FILE_DEVICE_SECURE_OPEN)  
        == FILE_DEVICE_ALLOW_APPCONTAINER_TRAVERSAL) {  
        return FALSE;  
    }  
    If AC Traversal flag in device then allow
```

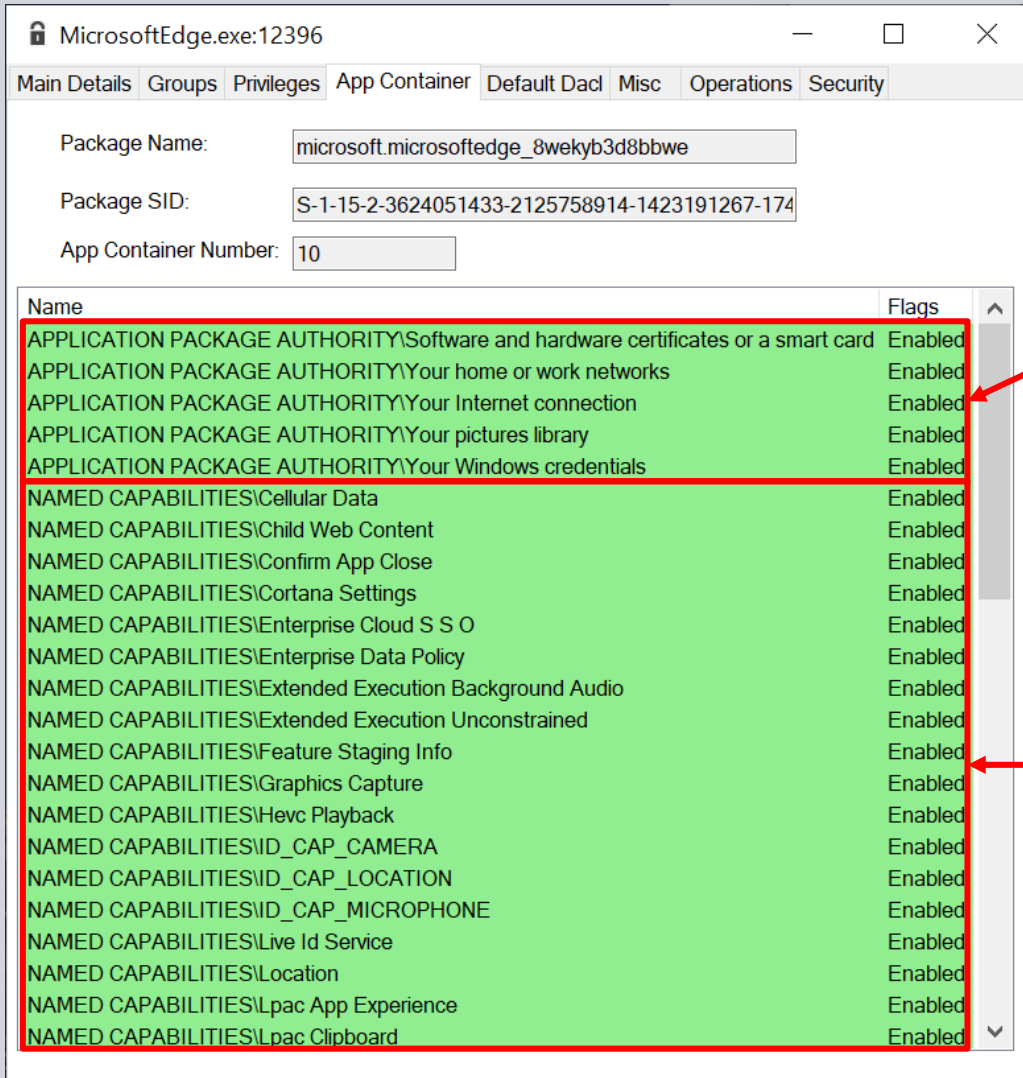
```
    BOOLEAN IsAppContainer;  
    SeIsAppContainerOrIdentifyLevelContext(SubjectSecurityContext,  
        &IsAppContainer);  
    return IsAppContainer;  
    Only allow if not AC.
```

```
}
```



```
Windows PowerShell  
PS C:\> $f = Get-NtFile \Device\Afd\Endpoint  
PS C:\> $f.Characteristics  
AllowAppContainerTraversal  
PS C:\> $f = Get-NtFile \Device\Beep  
PS C:\> $f.Characteristics  
SecureOpen  
PS C:\>
```

Generic AC Capabilities



MicrosoftEdge.exe:12396

Main Details Groups Privileges App Container Default Dacl Misc Operations Security

Package Name: microsoft.microsoftedge_8wekyb3d8bbwe

Package SID: S-1-15-2-3624051433-2125758914-1423191267-174

App Container Number: 10

Name	Flags
APPLICATION PACKAGE AUTHORITY\Software and hardware certificates or a smart card	Enabled
APPLICATION PACKAGE AUTHORITY>Your home or work networks	Enabled
APPLICATION PACKAGE AUTHORITY>Your Internet connection	Enabled
APPLICATION PACKAGE AUTHORITY>Your pictures library	Enabled
APPLICATION PACKAGE AUTHORITY>Your Windows credentials	Enabled
NAMED CAPABILITIES\Cellular Data	Enabled
NAMED CAPABILITIES\Child Web Content	Enabled
NAMED CAPABILITIES\Confirm App Close	Enabled
NAMED CAPABILITIES\Cortana Settings	Enabled
NAMED CAPABILITIES\Enterprise Cloud S S O	Enabled
NAMED CAPABILITIES\Enterprise Data Policy	Enabled
NAMED CAPABILITIES\Extended Execution Background Audio	Enabled
NAMED CAPABILITIES\Extended Execution Unconstrained	Enabled
NAMED CAPABILITIES\Feature Staging Info	Enabled
NAMED CAPABILITIES\Graphics Capture	Enabled
NAMED CAPABILITIES\Hvc Playback	Enabled
NAMED CAPABILITIES\ID_CAP_CAMERA	Enabled
NAMED CAPABILITIES\ID_CAP_LOCATION	Enabled
NAMED CAPABILITIES\ID_CAP_MICROPHONE	Enabled
NAMED CAPABILITIES\Live Id Service	Enabled
NAMED CAPABILITIES\Location	Enabled
NAMED CAPABILITIES\Lpac App Experience	Enabled
NAMED CAPABILITIES\Lpac Clipboard	Enabled

Fixed capabilities,
introduced in
Windows 8

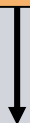
Generic capabilities,
introduced in
Windows 10

Capability String to SID

```
BOOL DeriveCapabilitySidsFromName(LPWSTR CapName, PSID **Sids)
```

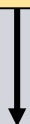
MyFantasticCapability

Arbitrary name



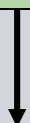
myfantasticcapability

Lower Case



SHA256

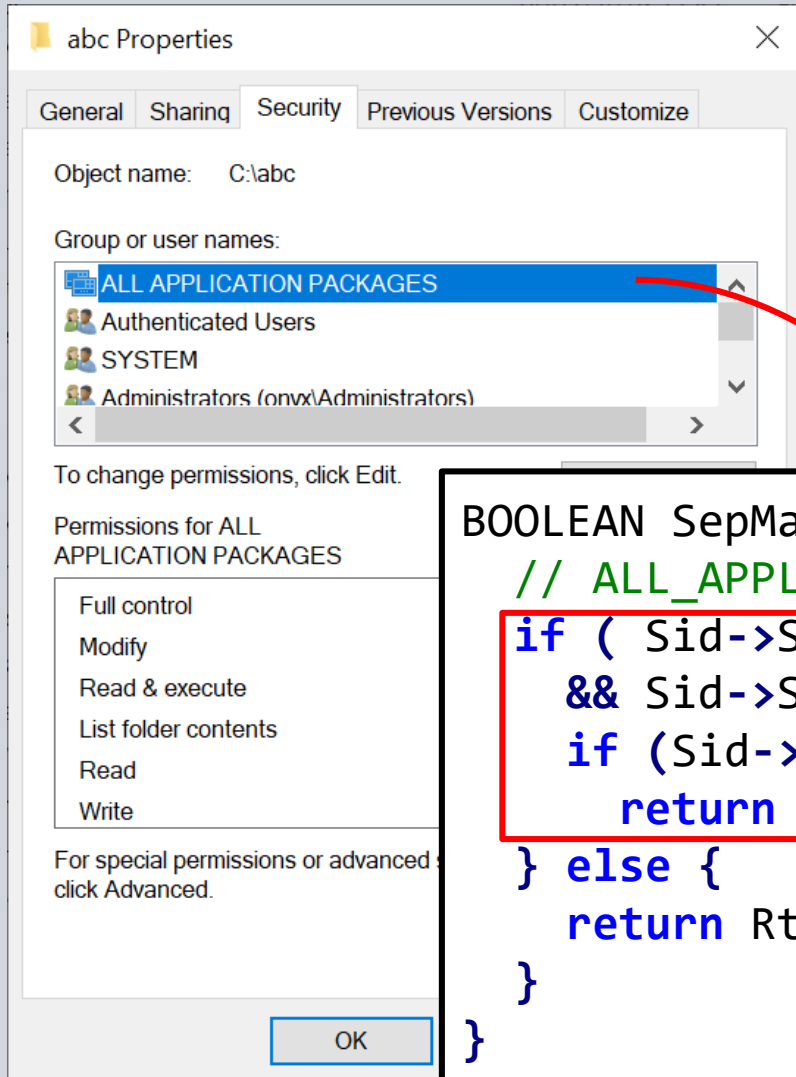
Hash Unicode String



S-1-15-3-%d-%d-%d-%d-...

Set RIDs to 32 bit Hash Values

All Application Packages



Hardcoded Group
Check if an App
Container

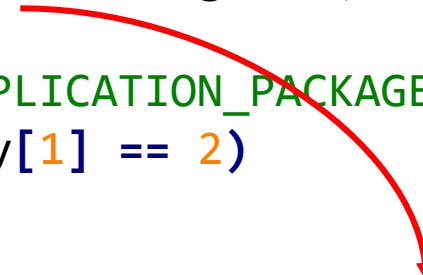
```
BOOLEAN SepMatchPackage(PTOKEN Token, PSID Sid) {  
    // ALL_APPLICATION_PACKAGES is S-1-15-2-1  
    if ( Sid->SubAuthority[0] == 2  
        && Sid->SubAuthorityCount == 2 ) {  
        if (Sid->SubAuthority[1] == 1)  
            return TRUE;  
    } else {  
        return RtlEqualSid(Token->Package, Sid);  
    }  
}
```

Low Privilege App Container

```
BOOLEAN SepMatchPackage(PTOKEN Token, PSID Sid) {
    if ( Sid->SubAuthority[0] == 2 &&
        Sid->SubAuthorityCount == 2 ) {
        if (Sid->SubAuthority[1] == 1 &&
            SepCanTokenMatchAllPackageSid(Token))
            return TRUE;
        // ALL_RESTRICTED_APPLICATION_PACKAGES is S-1-15-2-2
        if (Sid->SubAuthority[1] == 2)
            return TRUE;
    } else {
        return R
    }
}

BOOLEAN SepCanTokenMatchAllPackageSid(PTOKEN Token) {
    int Policy;
    AuthzBasepQuerySecurityAttributeAndValues(
        L"WIN://NOALLAPPPKG", &Policy)

    return Policy == 0;
}
```



Child App Containers

MicrosoftEdge.exe:18384 - User onyx\tyranid - TokenId ...

Main Details Groups Privileges App Container Default Dacl Misc Operations Security

Package Name: microsoft.microsoftedge_8wekyb3d8bbwe

Package SID: 7-1740899205-1073925389-3782572162-737981194

App Container Number: 4

Name

- APPLICATION PACKAGE AUTHORITY\Software and hardware certifi
- APPLICATION PACKAGE AUTHORITY>Your home or work networks
- APPLICATION PACKAGE AUTHORITY>Your Internet connection
- APPLICATION PACKAGE AUTHORITY>Your pictures library
- APPLICATION PACKAGE AUTHORITY>Your Windows credentials
- NAMED CAPABILITIES\Cellular Data
- NAMED CAPABILITIES\Child Web Content
- NAMED CAPABILITIES\Confirm App Close
- NAMED CAPABILITIES\Cortana Settings
- NAMED CAPABILITIES\Enterprise Cloud S S O
- NAMED CAPABILITIES\Enterprise Data Policy
- NAMED CAPABILITIES\Extended Execution Background Audio
- NAMED CAPABILITIES\Extended Execution Unconstrained
- NAMED CAPABILITIES\Feature Staging Info
- NAMED CAPABILITIES\Graphics Capture
- NAMED CAPABILITIES\Hvc Playback
- NAMED CAPABILITIES\ID_CAP_CAMERA
- NAMED CAPABILITIES\ID_CAP_LOCATION
- NAMED CAPABILITIES\ID_CAP_MICROPHONE
- NAMED CAPABILITIES\Live Id Service
- NAMED CAPABILITIES\Location

MicrosoftEdgeCP.exe:196 - User onyx\tyranid - TokenId ...

Main Details Groups Privileges App Container Default Dacl Misc Operations Security

Package Name: microsoft.microsoftedge_8wekyb3d8bbwe/002

Package SID: 3513710562-3729412521-1863153555-1462103995

App Container Number: 10

- APPLICATION PACKAGE AUTHORITY\Software and hardware certificates or a smart card
- APPLICATION PACKAGE AUTHORITY>Your Internet connection
- NAMED CAPABILITIES\Child Web Content
- NAMED CAPABILITIES\Graphics Capture
- NAMED CAPABILITIES\ID_CAP_CAMERA
- NAMED CAPABILITIES\ID_CAP_LOCATION
- NAMED CAPABILITIES\ID_CAP_MICROPHONE
- NAMED CAPABILITIES\Lpac App Experience
- NAMED CAPABILITIES\Lpac Clipboard
- NAMED CAPABILITIES\Lpac Com
- NAMED CAPABILITIES\Lpac Crypto Services
- NAMED CAPABILITIES\Lpac Enterprise Policy Change Notifications
- NAMED CAPABILITIES\Lpac Identity Services
- NAMED CAPABILITIES\Lpac Instrumentation
- NAMED CAPABILITIES\Lpac Media
- NAMED CAPABILITIES\Lpac Payments
- NAMED CAPABILITIES\Lpac Pn P Notifications
- NAMED CAPABILITIES\Lpac Printing
- NAMED CAPABILITIES\Lpac Services Management
- NAMED CAPABILITIES\Lpac Session Management
- NAMED CAPABILITIES\Lpac Web Platform

Writable

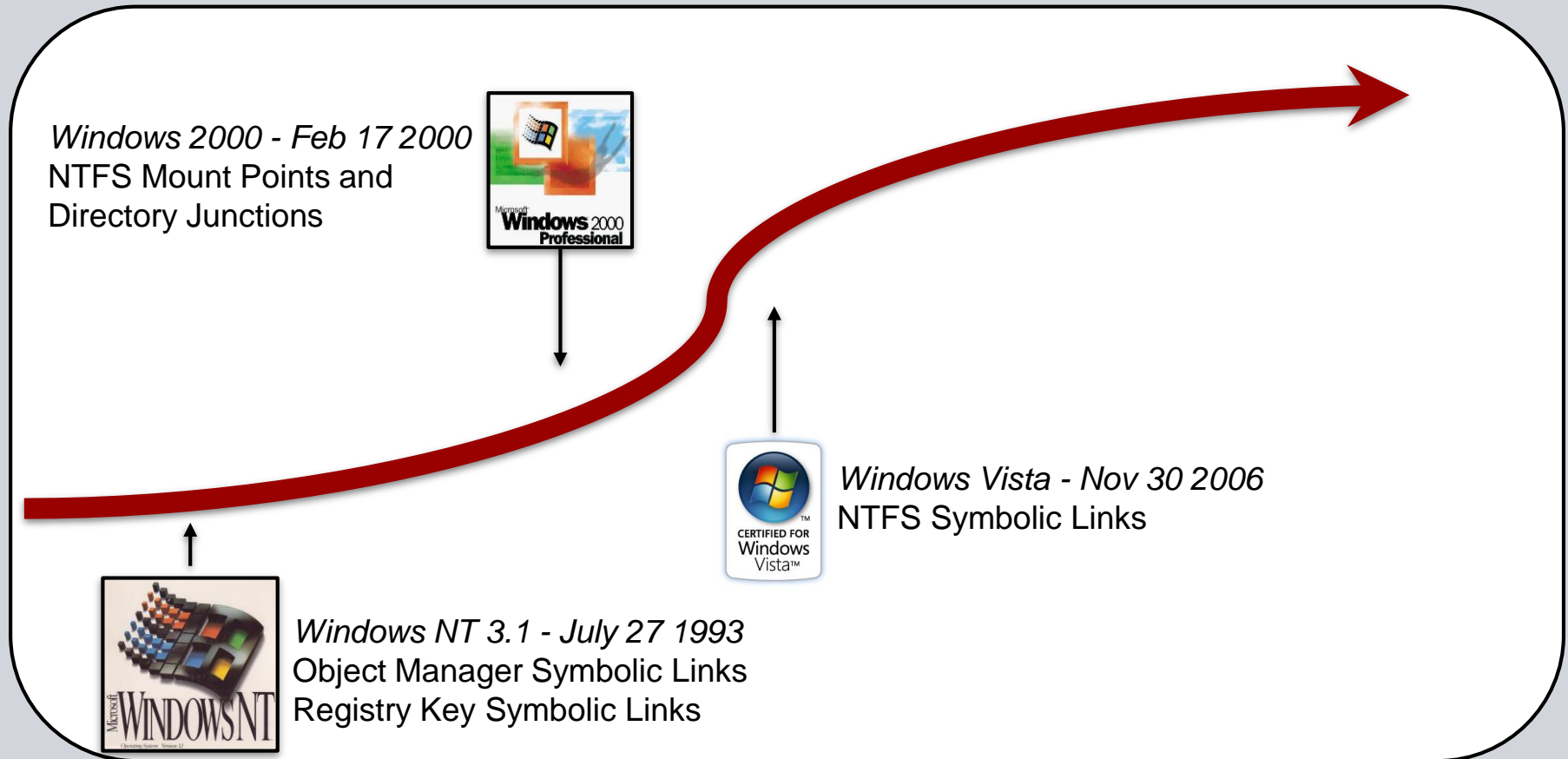
Read-Only

S-1-15-2-PARENT-RIDS

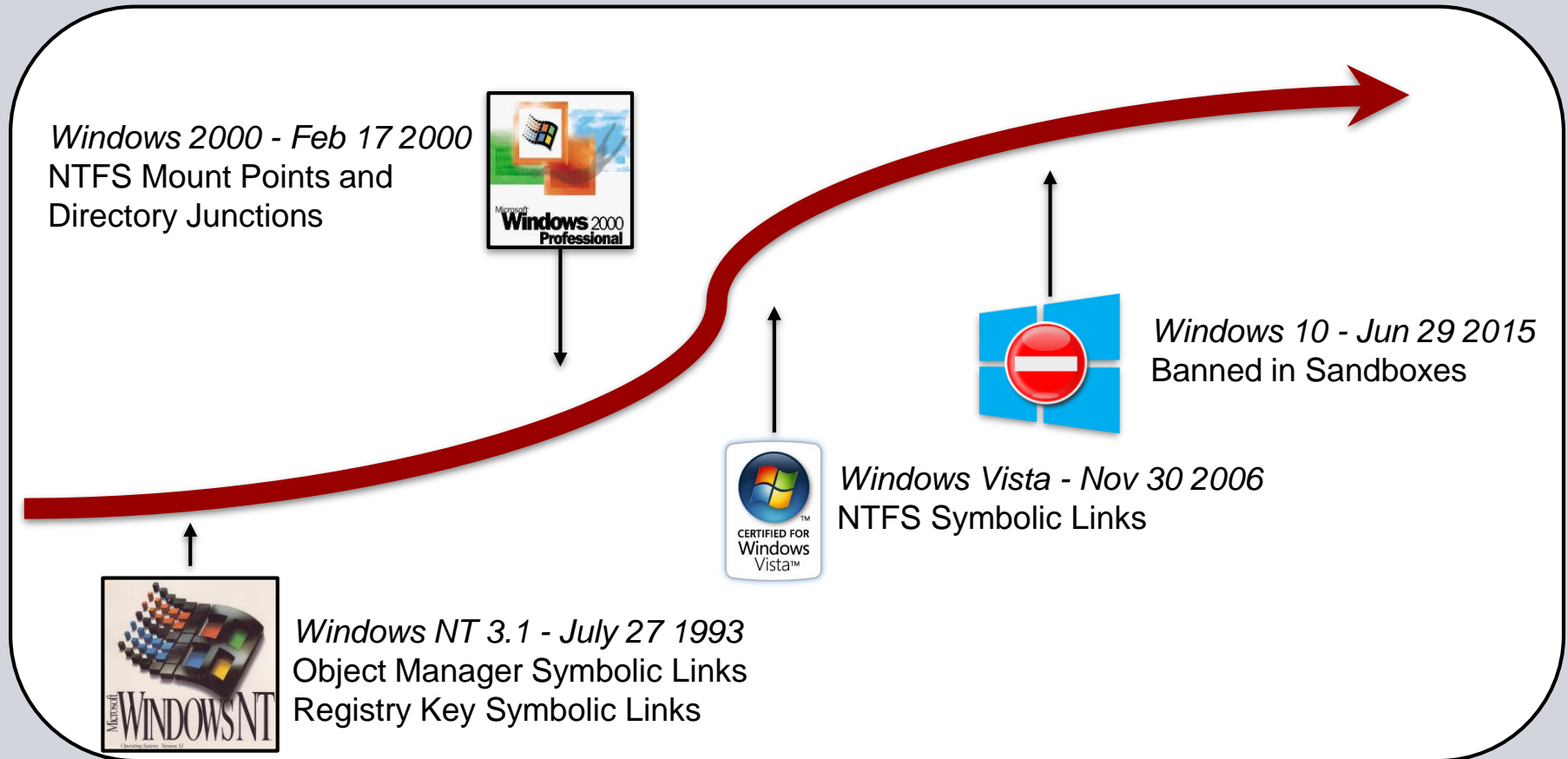
S-1-15-2-PARENT-RIDS-CHILD-RIDS

Mitigations

History of Symbolic Links



History of Symbolic Links




RtlIsSandboxedToken

- Introduced in Windows 10 but backported to Windows 7

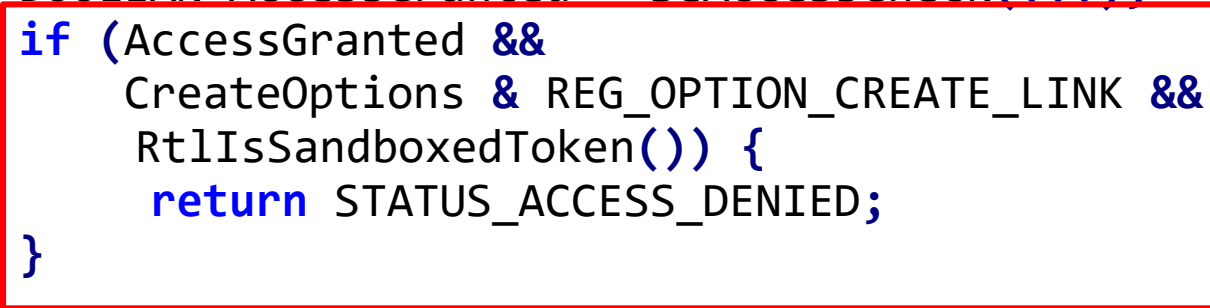
```
BOOLEAN RtlIsSandboxedToken() {  
    SECURITY_SUBJECT_CONTEXT SecurityContext;  
  
    SeCaptureSubjectContext(&SecurityContext);  
    return !SeAccessCheck(  
        SeMediumDaclSd,  
        SubjectSecurityContext,  
        READ_CONTROL);  
}
```

Must pass
security
check



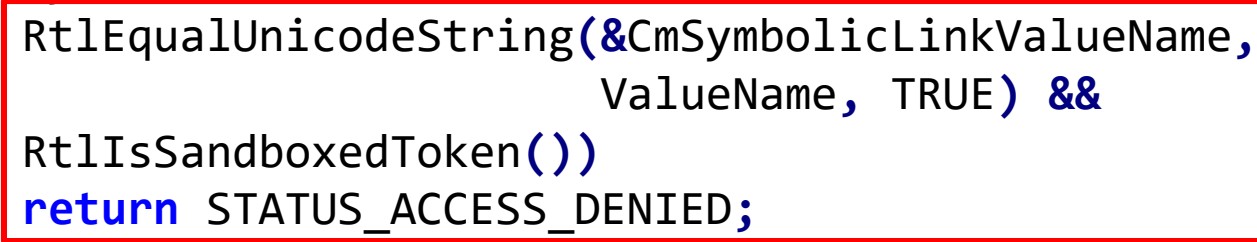
Registry Key Symbolic Links

```
NTSTATUS CmpCheckCreateAccess(...) {
    BOOLEAN AccessGranted = SeAccessCheck(...);
    if (AccessGranted &&
        CreateOptions & REG_OPTION_CREATE_LINK &&
        RtlIsSandboxedToken()) {
        return STATUS_ACCESS_DENIED;
    }
}
```



Hard
Ban!

```
NTSTATUS CmSetValueKey(...) {
    if (Type == REG_LINK &&
        RtlEqualUnicodeString(&CmSymbolicLinkValueName,
                               ValueName, TRUE) &&
        RtlIsSandboxedToken())
        return STATUS_ACCESS_DENIED;
}
```



Blocking NTFS Mount Points

```
NTSTATUS IopXxxControlFile(...) {  
    if (ControlCode == FSCTL_SET_REPARSE_POINT &&  
        RtlIsSandboxedToken()) {  
        if (buffer.ReparseTag == IO_REPARSE_TAG_MOUNT_POINT) {  
            InitializeObjectAttributes(&ObjAttr, buffer.PathBuffer);  
            status = ZwOpenFile(&FileHandle, FILE_GENERIC_WRITE,  
                &ObjAttr, ..., FILE_DIRECTORY_FILE);  
            if (status < 0)  
                return status;  
            // Continue.  
        }  
    }  
}
```

Checks target is a directory and writable

Bypassing the Mitigation

Issue 486: Windows: Sandboxed Mount Reparse Point

[Code](#)[◀ Prev](#)

7 of 23

[Next ▶](#)

Creation Mitigation Bypass

[Back to list](#)Reported by forshaw@google.com, Jul 22 2015**Project Member**

Windows: Sandboxed Mount Reparse Point Creation Mitigation Bypass

Platform: Windows 10 (build 10240), earlier versions do not have the functionality

Class: Security Feature Bypass

Summary:

A mitigation added to Windows 10 to prevent NTFS Mount Reparse Points being created at integrity levels below medium can be bypassed.

Description:

Windows 10 has added some new mitigations to block the creation or change the behaviour of certain symbolic links when issued by a low integrity/sandboxed process. The presumed aim to to make it harder to abuse these types of tricks to break out of a sandbox.

<https://bugs.chromium.org/p/project-zero/issues/detail?id=486>

Bypassing the Mitigation

Issue 486: Windows: Sandboxed Mount Reparse Point



Code

[< Prev](#)

7 of 23

[Next >](#)

Creation Mitigation Bypass

[Back to list](#)Reported by forshaw@google.com, Jul 22 2015

Project Member

Windows: Sandboxed Mount Reparse Point Creation Mitigation Bypass

Platform: Windows 10 (build 10240), earlier versions do not have the functionality

Class

Summ

A miti

mediu

Descr

Windo

when

tricks

```
NTSTATUS NtSetInformationProcess(...) {  
    // ...  
  
    case ProcessDeviceMap:  
        HANDLE hDir = *(HANDLE*)Data;  
        if (RtlIsSandboxedToken())  
            return STATUS_ACCESS_DENIED;  
        return ObSetDeviceMap(ProcessObject, hDir);  
  
    // ...  
}
```

Use Hardlinks

Issue 531: Windows: Creating Hardlinks Doesn't Require Write Permissions to the Target



Code

1 of 4

[Back to list](#)Reported by forshaw@google.com, Sep 14 2015

Project Member

Microsoft requested I removed information from a public presentation that you can create NTFS hardlinks without needing write permissions on the target file. Their view is they want to fix this, at the least to prevent its abuse in sandboxed applications so a case has been set up to track the issue. It's still under the normal 90 day SLA.

This bug is subject to a 90 day disclosure deadline. If 90 days elapse without a broadly available patch, then the bug report will automatically become visible to the public.

<https://bugs.chromium.org/p/project-zero/issues/detail?id=531>

Use Hardlinks

Issue 531: Windows: Creating Hardlinks Doesn't Require Write Permissions to the Target

[Code](#)

1 of 4

[Back to list](#)Reported by forshaw@google.com, Sep 14 2015

Project Member

Microsoft requested I removed information from a public presentation that you can create NTFS hardlinks without needing write permissions on the target file. Their view is they want to fix this, at the

```
NTSTATUS NtSetInformationFile(...) {
```

```
  case FileLinkInformation:
```

```
    ACCESS_MASK RequiredAccess = 0;
```

```
    if(RtlIsSandboxedToken()) {
```

```
      RequiredAccess |= FILE_WRITE_ATTRIBUTES;
```

```
    }
```

```
    ObReferenceObjectByHandle(FileHandle, RequiredAccess);
```

```
  }
```

Setting an Mitigation Policy

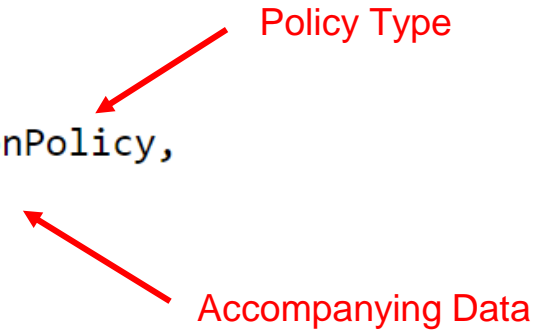
SetProcessMitigationPolicy function

Sets the mitigation policy for the calling process.

Syntax

C++

```
BOOL WINAPI SetProcessMitigationPolicy(  
    _In_ PROCESS_MITIGATION_POLICY MitigationPolicy,  
    _In_ PVOID lpBuffer,  
    _In_ SIZE_T dwLength  
);
```

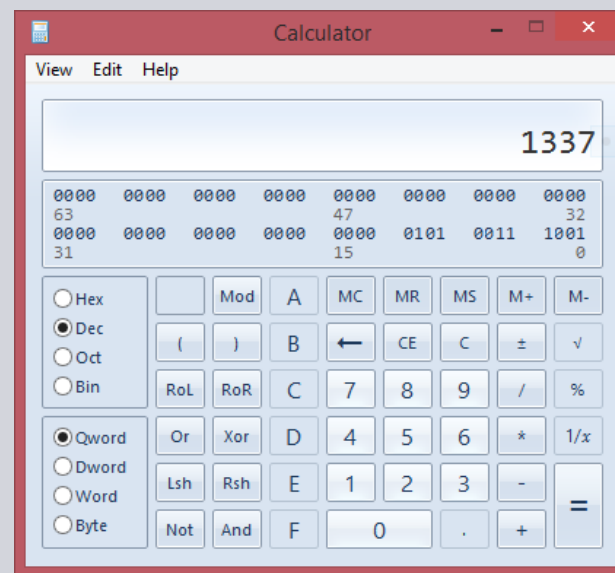
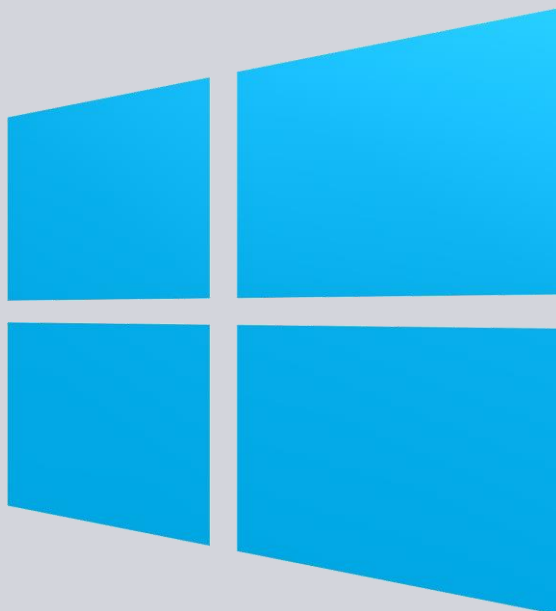
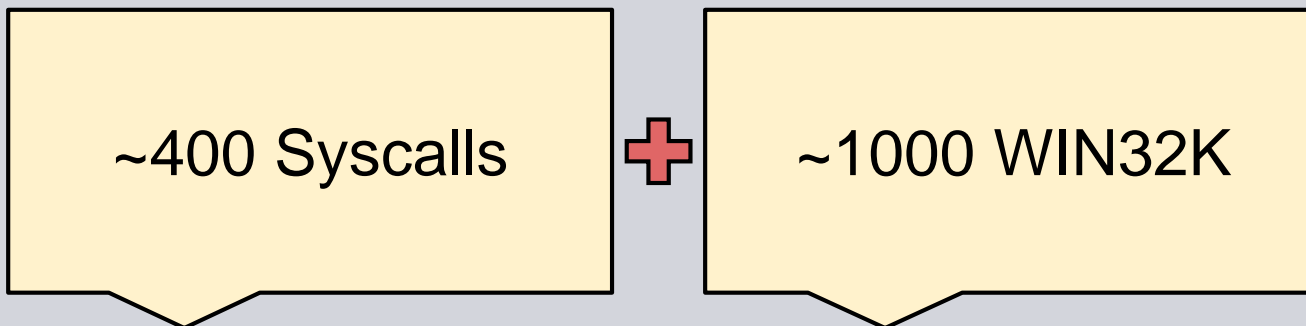


Available Policies

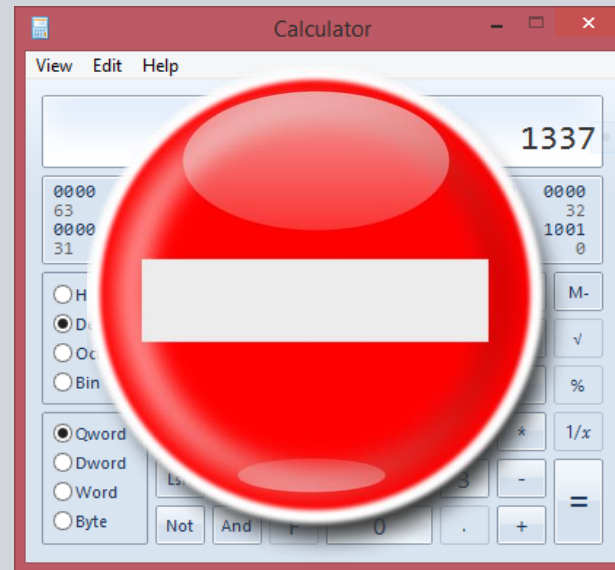
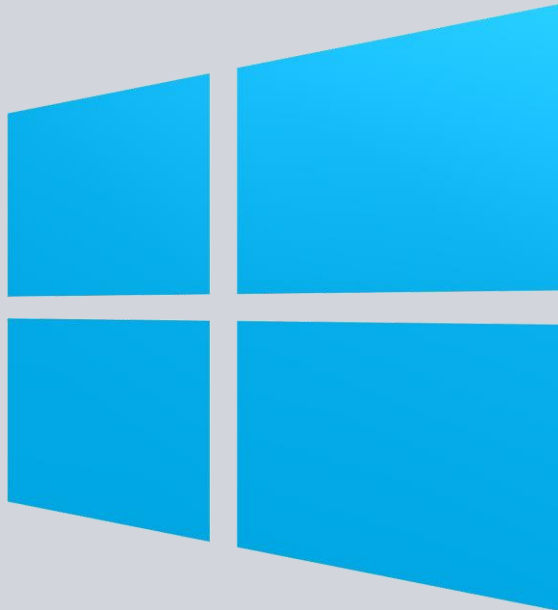
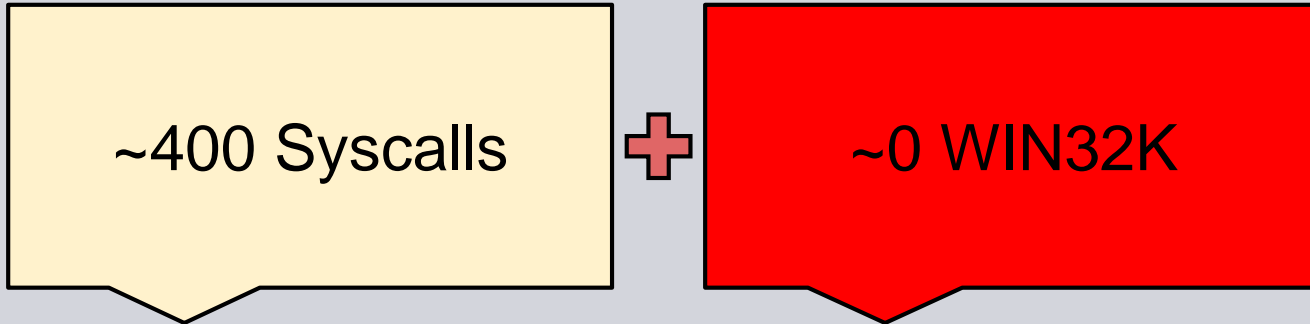
<i>Policy</i>	<i>Supported Win8.1 Update 2</i>	<i>Supported Win10 TH2</i>
ProcessDEPPolicy	Yes	Yes
ProcessASLRPolicy	Yes	Yes
ProcessDynamicCodePolicy	Yes	Yes
ProcessStrictHandleCheckPolicy	Yes	Yes
ProcessSystemCallDisablePolicy	Yes	Yes
ProcessMitigationOptionsMask	Invalid	Invalid
ProcessExtensionPointDisablePolicy	Yes	Yes
ProcessControlFlowGuardPolicy	Invalid	Invalid
ProcessSignaturePolicy	Yes*	Yes
ProcessFontDisablePolicy	No	Yes
ProcessImageLoadPolicy	No	Yes

* Not supported through SetProcessMitigationPolicy

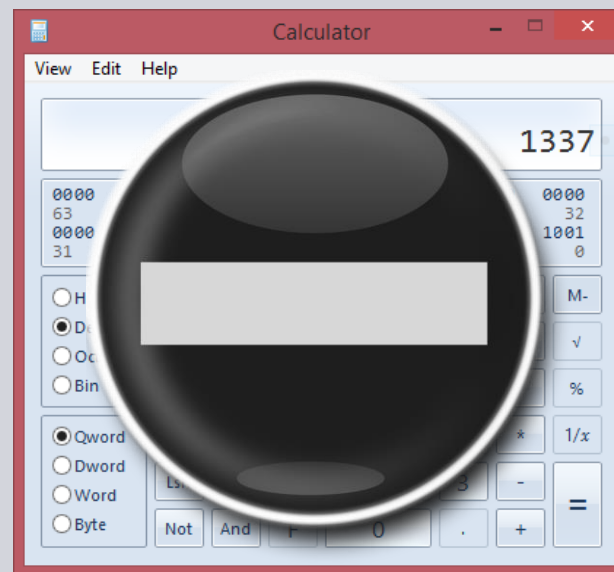
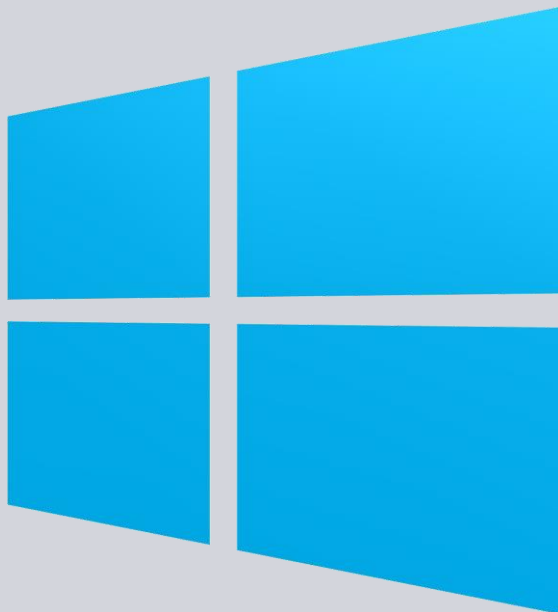
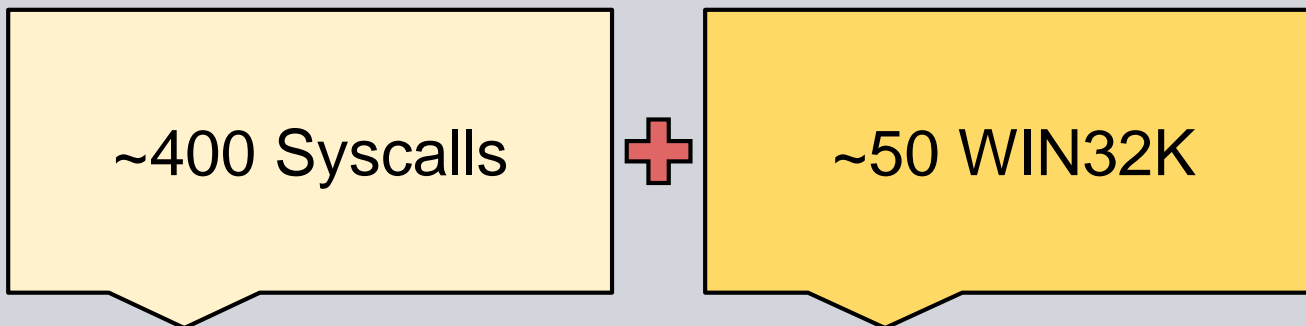
Kernel Attack Surface



WIN32K System Call Disable



WIN32K System Call Filter



Font Disable Policy

```
struct PROCESS_MITIGATION_FONT_DISABLE_POLICY
{
    DWORD DisableNonSystemFonts           : 1;
    DWORD AuditNonSystemFontLoading      : 1;
    DWORD ReservedFlags                   : 30;
};
```

Disable fonts loaded from memory or outside
of %WINDIR%\Fonts

Bypassable Mitigation

Issue 779: Windows: Custom Font Disable

[Code](#)[< Prev](#) 2 of 2[Back to list](#)

Policy Bypass

Reported by forshaw@google.com, Mar 24 2016**Project Member**

Windows: Custom Font Disable Policy Bypass

Platform: Windows 10 Only

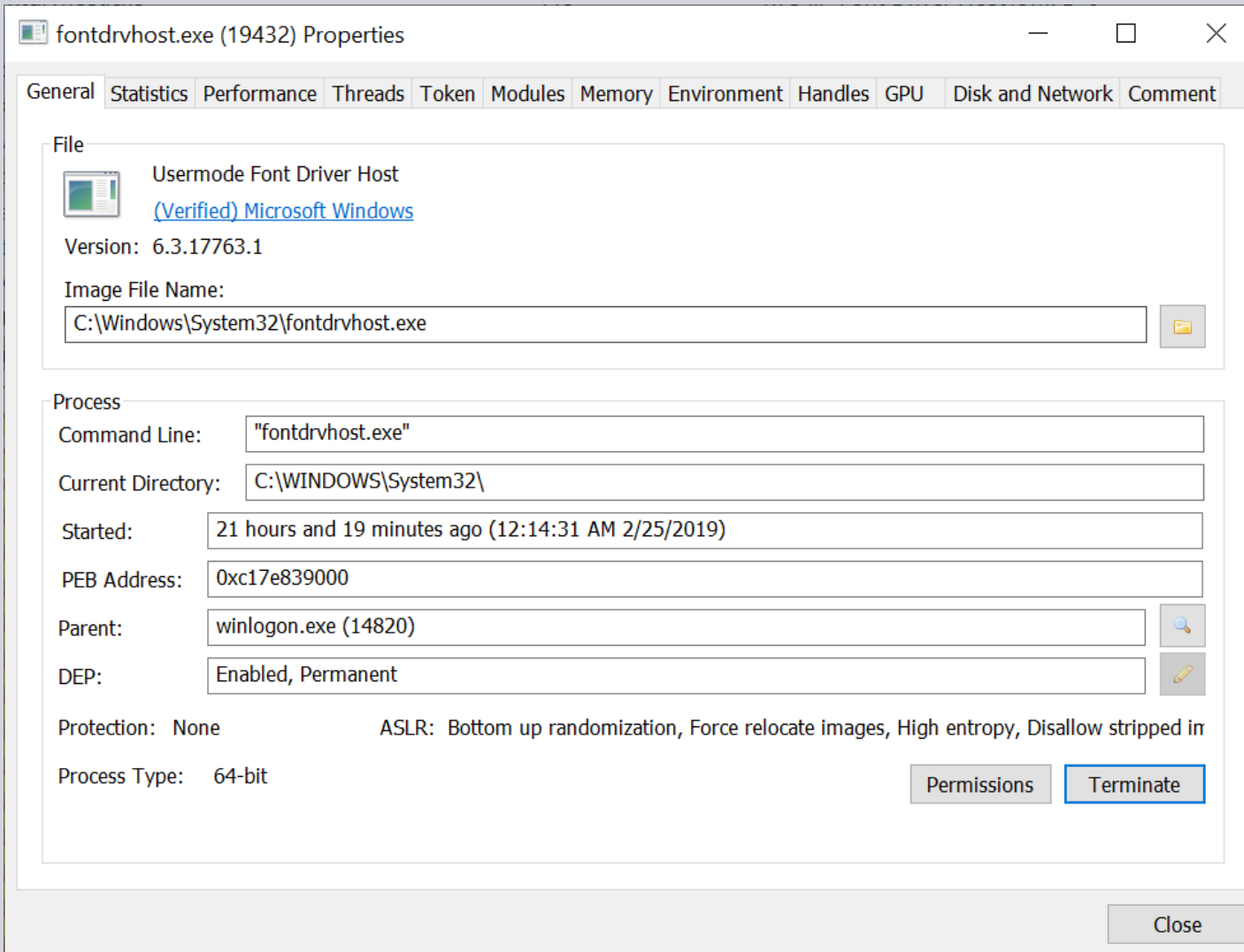
Class: Security Feature Bypass

Summary:

It's possible to bypass the ProcessFontDisablePolicy check in win32k to load a custom font from an arbitrary file on disk even in a sandbox. This might be used as part of a chain to elevate privileges. If anything this is really a useful demonstration that you probably really want to shutdown the object manager directory shadowing as part of the sandbox mitigations, even if you don't fix the explicit bypass.

<https://bugs.chromium.org/p/project-zero/issues/detail?id=779>

User Mode Font Driver



fontdrvhost.exe (19432) Properties

General Statistics Performance Threads Token Modules Memory Environment Handles GPU Disk and Network Comment

File

Usermode Font Driver Host
[\(Verified\) Microsoft Windows](#)

Version: 6.3.17763.1

Image File Name:
C:\Windows\System32\fontdrvhost.exe

Process

Command Line: "fontdrvhost.exe"

Current Directory: C:\WINDOWS\System32\

Started: 21 hours and 19 minutes ago (12:14:31 AM 2/25/2019)

PEB Address: 0xc17e839000

Parent: winlogon.exe (14820)

DEP: Enabled, Permanent

Protection: None ASLR: Bottom up randomization, Force relocate images, High entropy, Disallow stripped in

Process Type: 64-bit

Permissions Terminate

Close

Bugs Bugs Bugs

Issue 468: Windows: User Mode Font Driver



Code

1 of 2 [Next >](#)

Thread Permissions EoP

[Back to list](#)

Reported by forshaw@google.com, Jun 30 2015

Project Member

Windows: User Mode Font Driver Thread Permissions EoP

Platform: Windows 10 Build 10130

Class: Elevation of Privilege

Summary:

The host process for the UMFD runs as a normal user but with a heavily restrictive process DACL. It's possible to execute arbitrary code within the context of the process because it's possible to access the process's threads leading to local EoP.

Description:

NOTE: This was tested on the latest available build on Windows 10. I don't know if the final version will change the functionality to fix this vulnerability.

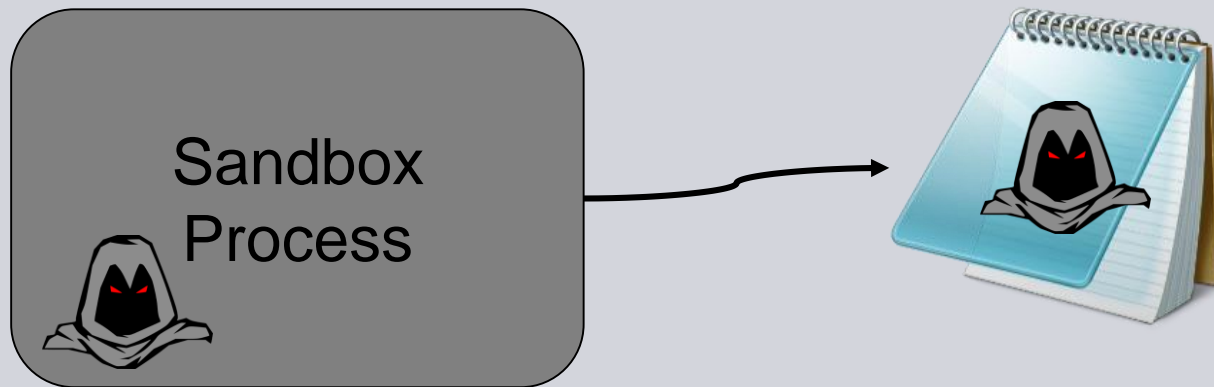
<https://bugs.chromium.org/p/project-zero/issues/detail?id=468>

Process Mitigations Inheritance

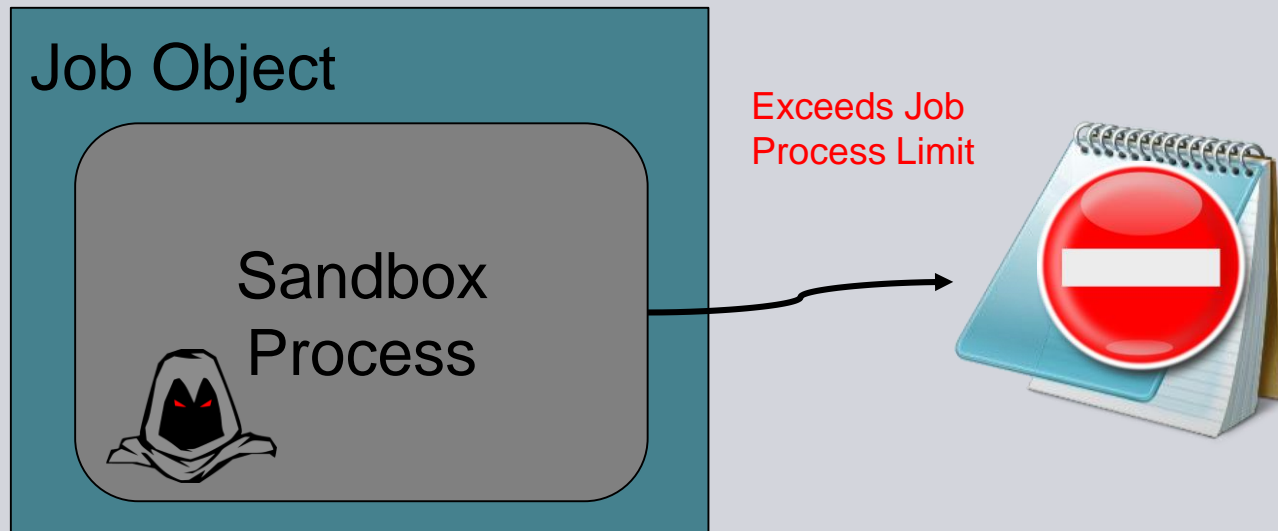
- No policies can be disabled once set in-process.
- However only a small subset of mitigations are inherited

<i>Policy</i>	<i>Inherited</i>
Dynamic Code	No
System Call Disable	Yes
Signature	No
Font Disable	No
Image Load	Yes

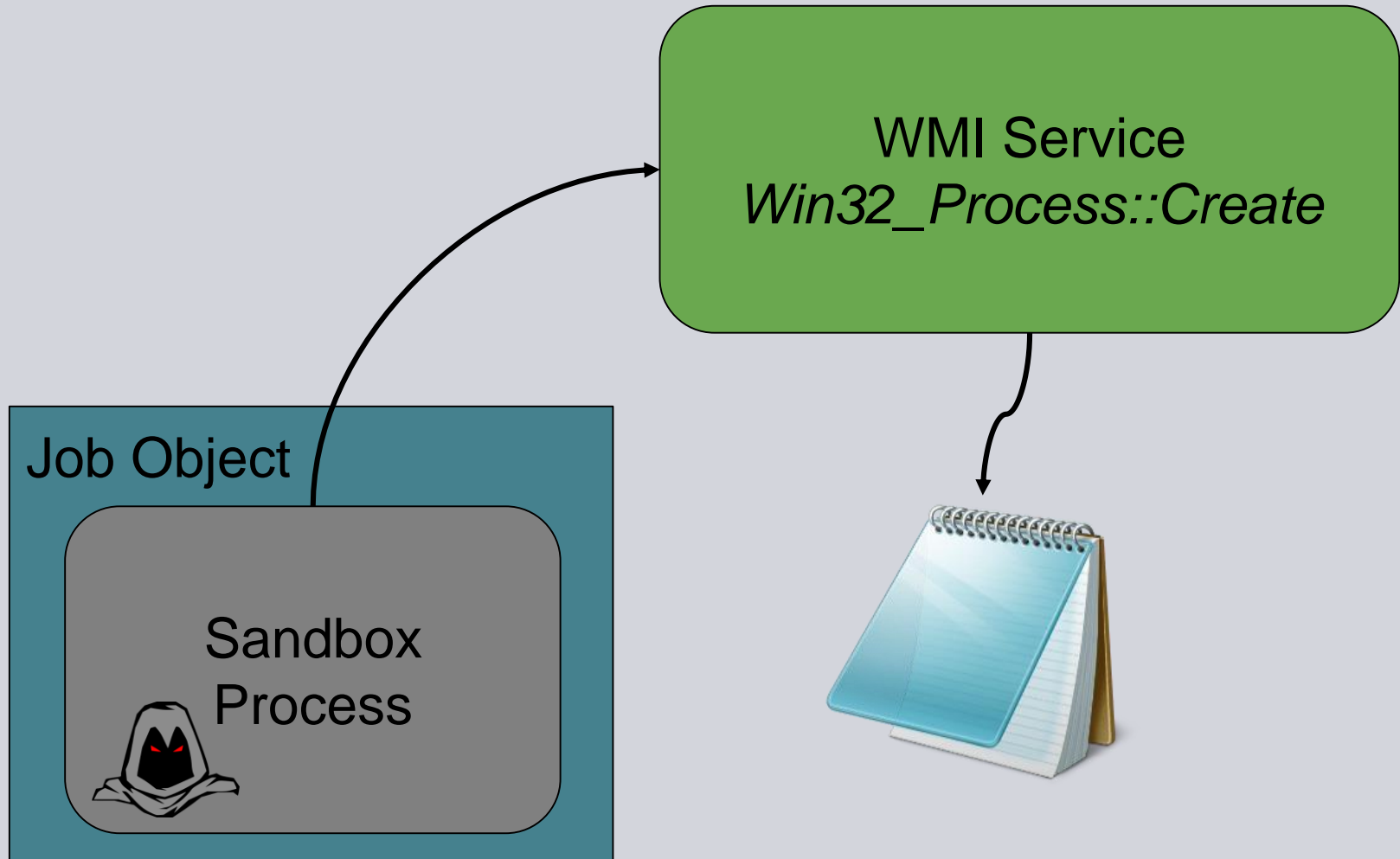
Migrate to a New Process



Restrict With Job Objects



Restrict With Job Objects

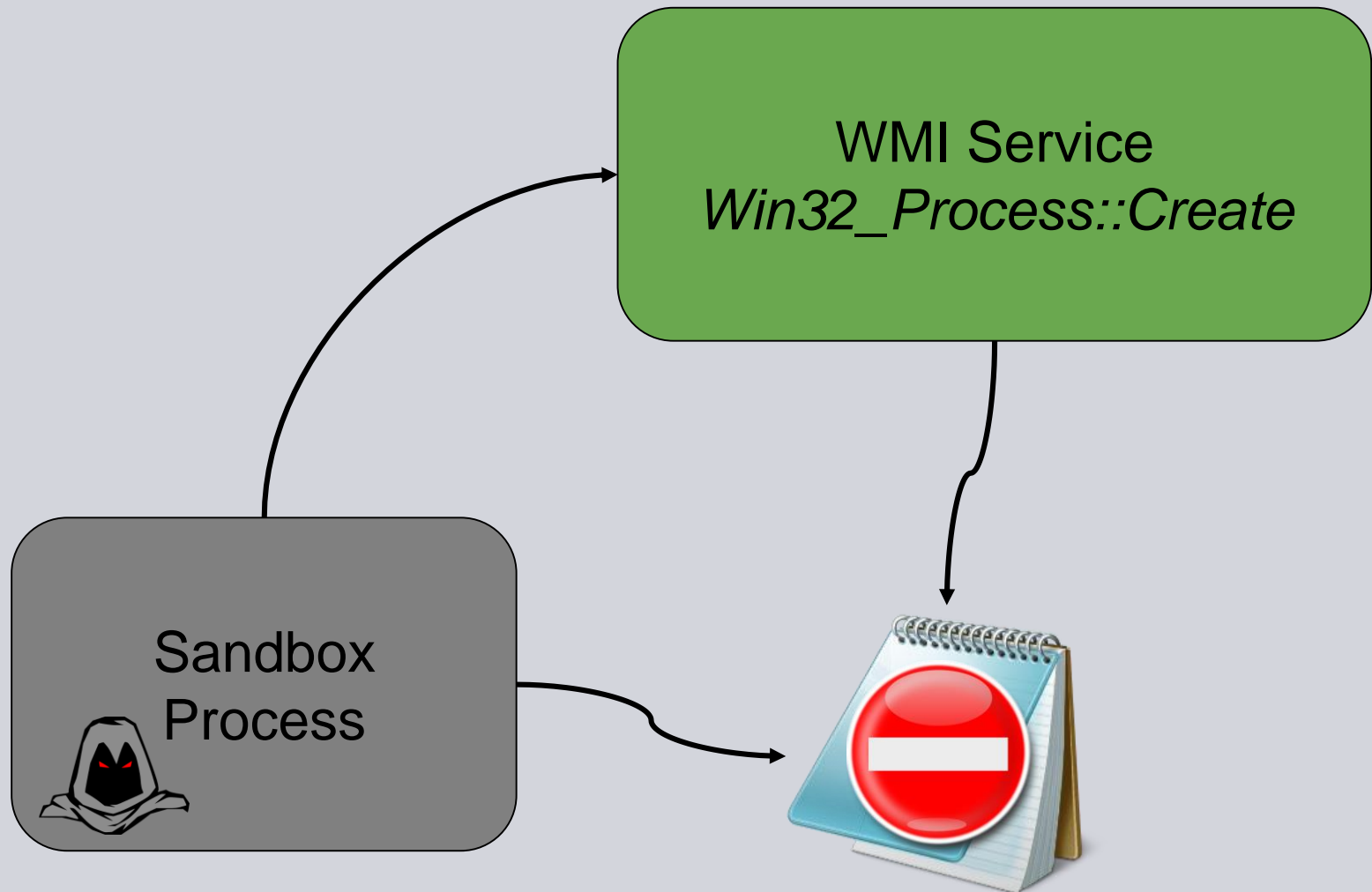


Inside NtCreateUserProcess

```
DWORD ChildProcessPolicyFlag = // From process attribute.  
BOOLEAN ChildProcessAllowed = TokenObject->TokenFlags &  
                                CHILD_PROCESS_RESTRICTED;  
if (!ChildProcessAllowed) {  
    if (!ChildProcessPolicyFlag &                                Block process with  
        PROCESS_CREATION_CHILD_PROCESS_OVERRIDE)                Token flag.  
        || !SeSinglePrivilegeCheck(SeTcbPrivilege))  
        return STATUS_ACCESS_DENIED;  
}
```

```
SepDuplicateToken(TokenObject, ..., &NewTokenObject);  
if (ChildProcessPolicyFlag &  
    PROCESS_CREATION_CHILD_PROCESS_RESTRICTED)                Set the flag on new  
    NewTokenObject->TokenFlags |= CHILD_PROCESS_RESTRICTED;    Token
```

Effective Mitigation



Except When It's Not!

Issue 1544: Windows: Child Process Restriction Mitigation Bypass



Code

[◀ Prev](#)

79 of 94

[Next ▶](#)

[Back to list](#)

Reported by forshaw@google.com, Mar 4 2018

Project Member

Windows: Child Process Restriction Mitigation Bypass
Platform: Windows 10 1709 (not tested other versions)
Class: Security Feature Bypass

Summary:

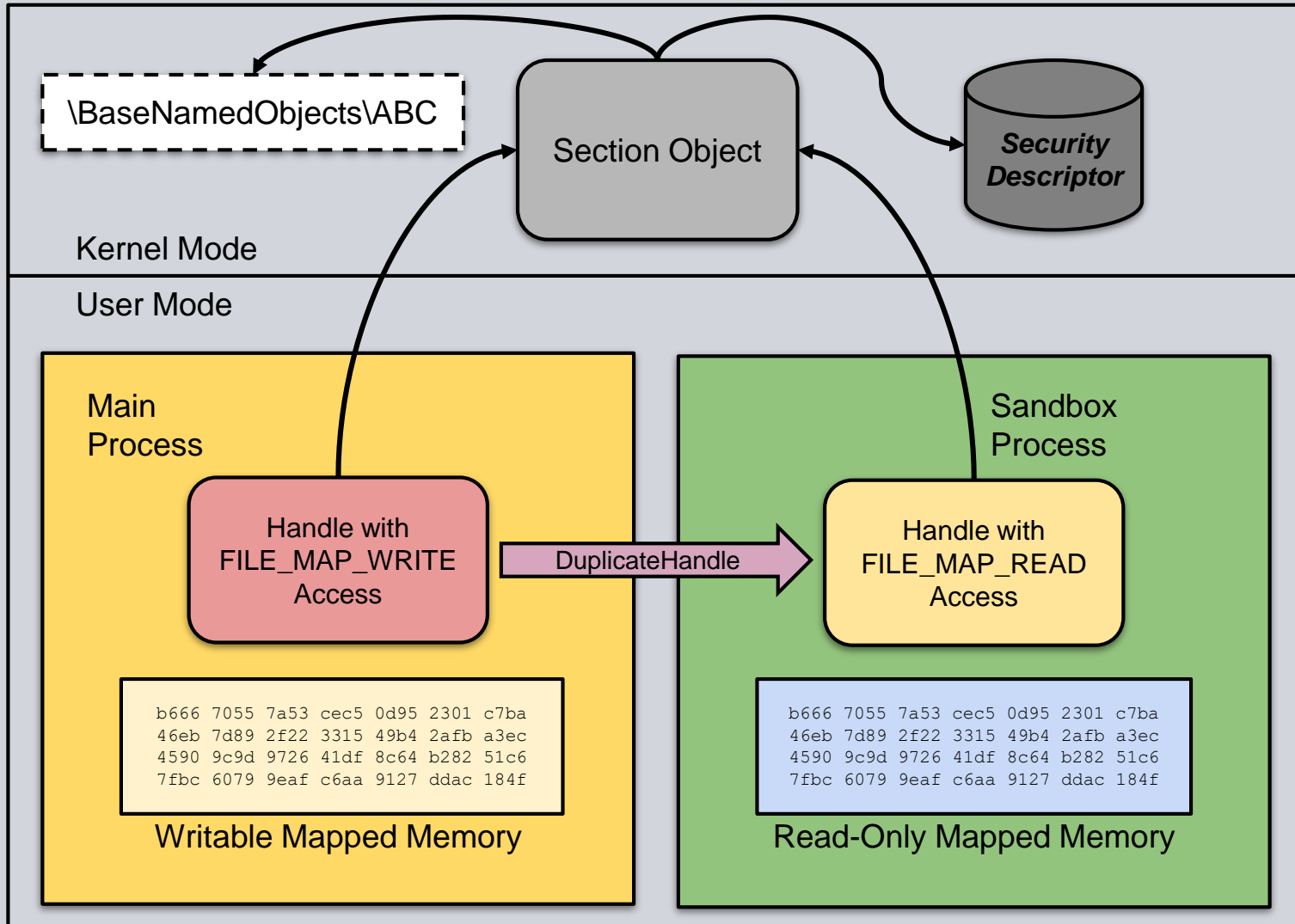
It's possible to bypass the child process restriction mitigation policy by impersonating the anonymous token leading to a security feature bypass.

Description:

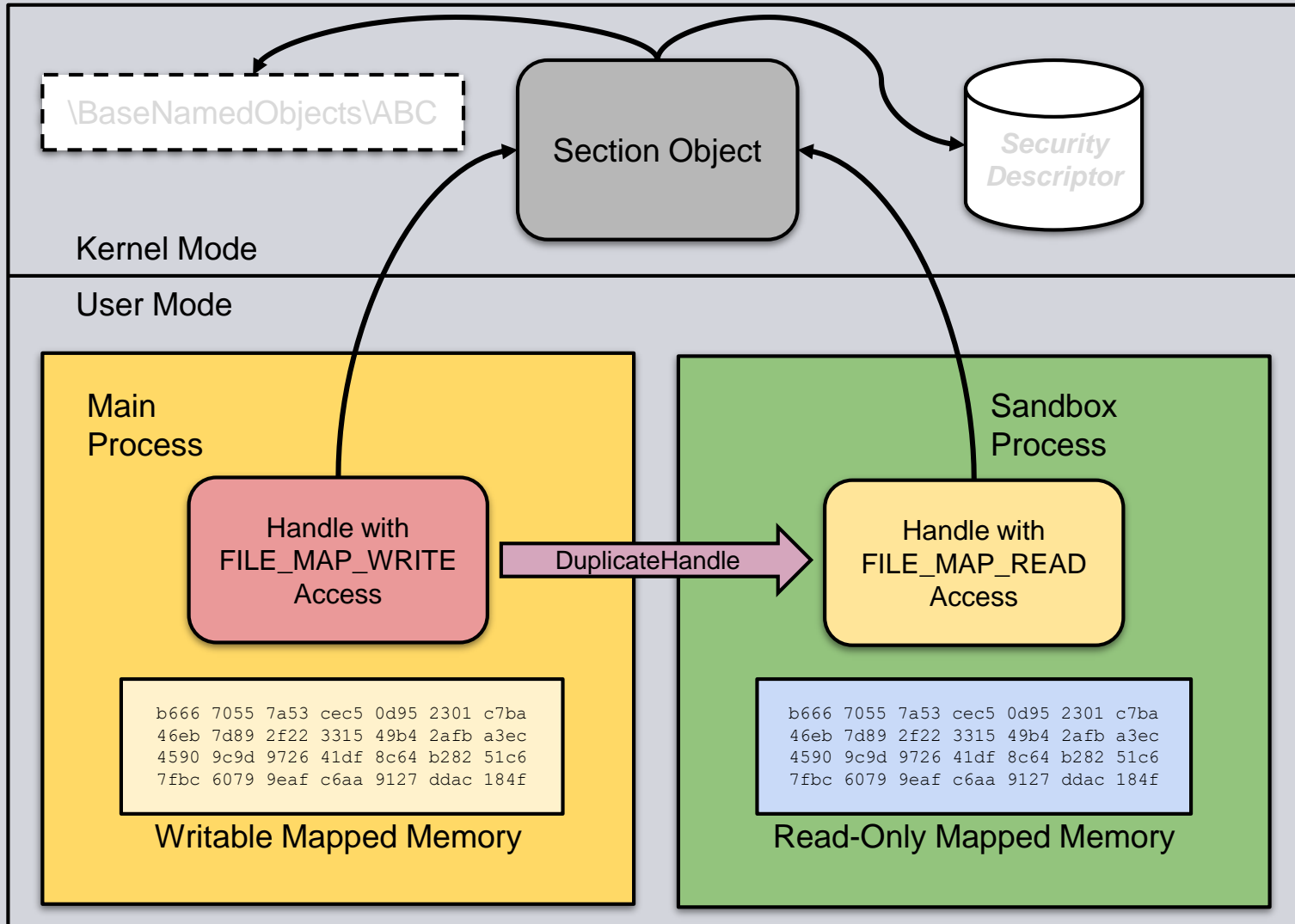
Windows 10 has a mitigation policy to restrict a process creating new child processes. I believe the main rationale is to prevent escaping some of the other mitigations which are not inherited across to new child processes as well as bugs which can only be exploiting from a fresh process. The policy is enforced as a flag in the token rather than on the process which allows the restriction to be passed across process boundaries during impersonation, which would also kill abusing WMI Win32_Process and similar.

<https://bugs.chromium.org/p/project-zero/issues/detail?id=1544>

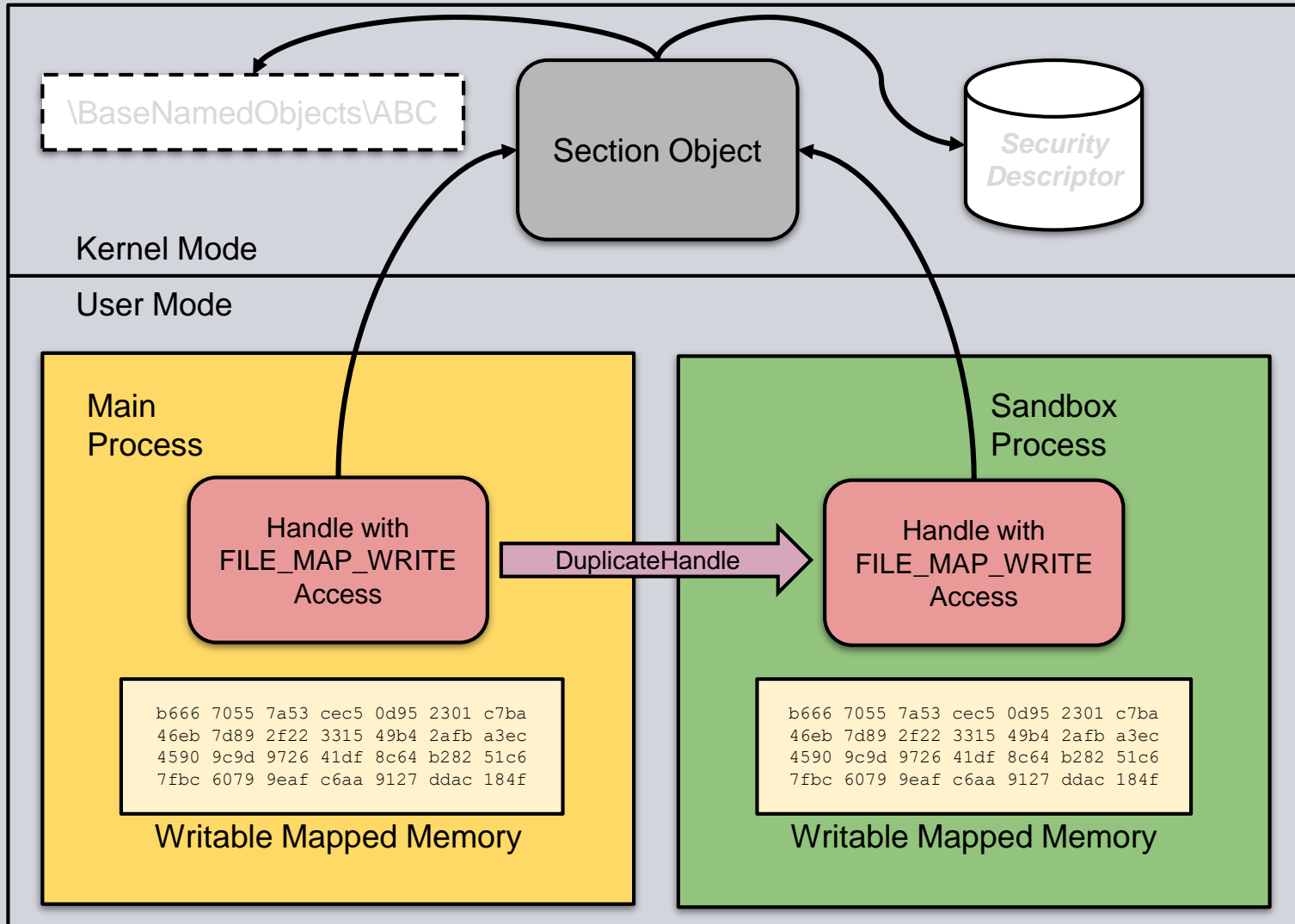
Sharing Sections



Sharing Sections



Sharing Sections



Name No Longer Needed

No Name - No SD

```
Windows PowerShell
PS C:\> $s = New-NtSection -Size 10000
PS C:\> $s.Name

PS C:\> $s.SecurityDescriptor.Dacl

PS C:\> $s = New-NtSection -Size 10000
>> -SecurityDescriptor "D:(A;;;GA;;;WD)"
PS C:\> $s.Name

PS C:\> $s.SecurityDescriptor.Dacl
```

Type	User	Flags	Mask
Allowed	Everyone	None	000F001F

Specify SD to Create

DEMOS

All of the Above

Hyper-V Everywhere

- Windows Defender Application Guard
- Windows Sandbox (coming soon!)



**Microsoft
Hyper-v**

PICO Process Available

```
root@onyx: ~  
BITS 64  
  
%define SYS_execve 59  
  
_start:  
    mov rax, SYS_execve  
; Load the executable path  
    lea rdi, [rel _exec_path]  
; Load the executable path  
    lea rsi, [rel _argument]  
; Build argument array on stack = { _exec_path, _argument, NULL }  
execve.asm 1,1 Top  
#include <stdio.h>  
#include <stdlib.h>  
#include <unistd.h>  
#include <string.h>  
#include <sys/socket.h>  
#include <sys/un.h>  
  
int main(int argc, char** argv) {  
    struct sockaddr_un addr = {};  
    int sock = -1;  
afunix.c 1,1 Top  
"afunix.c" 36L, 712C
```

Conclusions

- Introduction of Windows 10 Had Effect on Sandboxes
 - Edge gave Microsoft an excuse to innovate
 - Fast release cycle meant new mitigations could ship sooner
- Still plenty of things I'd like to see
 - Better system call filtering, NTOS as well as WIN32K
 - Improvements to Eliminate long standing warmup problems.

Thanks for Listening *Questions?*