



DEV✓CORE

LeakLess? Another Leak Way in Windows Kernel DFSC

Angelboy

戴夫寇爾股份有限公司
angelboy@devco.re

DEVCORE CONFERENCE 2024 | 2024.03.16



Angelboy

DEV✓CORE

DEV✓*CORE*

Introduction

Why do we need Elevation of
Privilege ?

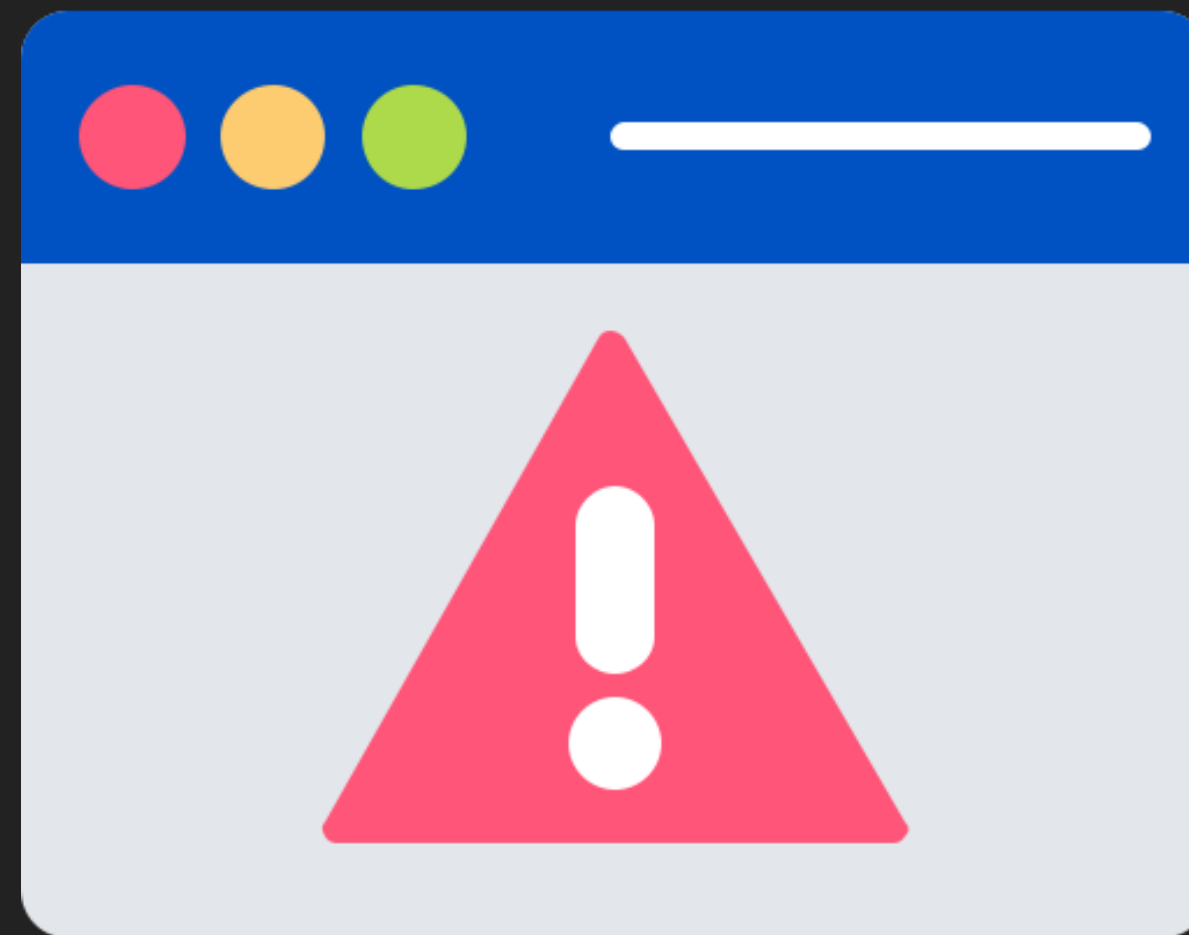
```
struct group_info_init_groups = { .usage = ATOMIC_INIT(2) };
struct group_info *groups_alloc(int gidsetsize){
    struct group_info *group_info;
    int nblocks;
    int i;

    nblocks = (gidsetsize + NGROUPS_PER_BLOCK - 1) / NGROUPS_PER_BLOCK;
    /* Make sure we always allocate at least one indirect block pointer */
    nblocks = nblocks ? : 1;
    group_info = kmalloc(sizeof(*group_info) + nblocks*sizeof(gid_t *), GFP_USER);
    if
gro
gro
ato
    if (gidsetsize <= NGROUPS_SMALL)
        group_info->blocks[0] = group_info->small_block;
    else {
        for (i = 0; i < nblocks; i++) {
            gid_t *b;
            b = (gid_t *) __get_free_page(GFP_USER);
            if (!b)
                goto out_undo_partial_alloc;
        }
    }
}
```



Why?

- 在執行紅隊過程中，經常取得一台機器控制權後，**權限過低無法獲取機密資訊**
- 現階段已知提權方法**越來越難用**，條件越來越**嚴苛**



Why?

- 在執行紅隊過程中，經常取得一台機器控制權後，**權限過低無法獲取機密資訊**
 - 現階段已知提權方法**越來越難用**，條件越來越**嚴苛**
- Pwn2Own
 - 有機會的話還是會想要能在 Pwn2Own 上能攻下 Windows

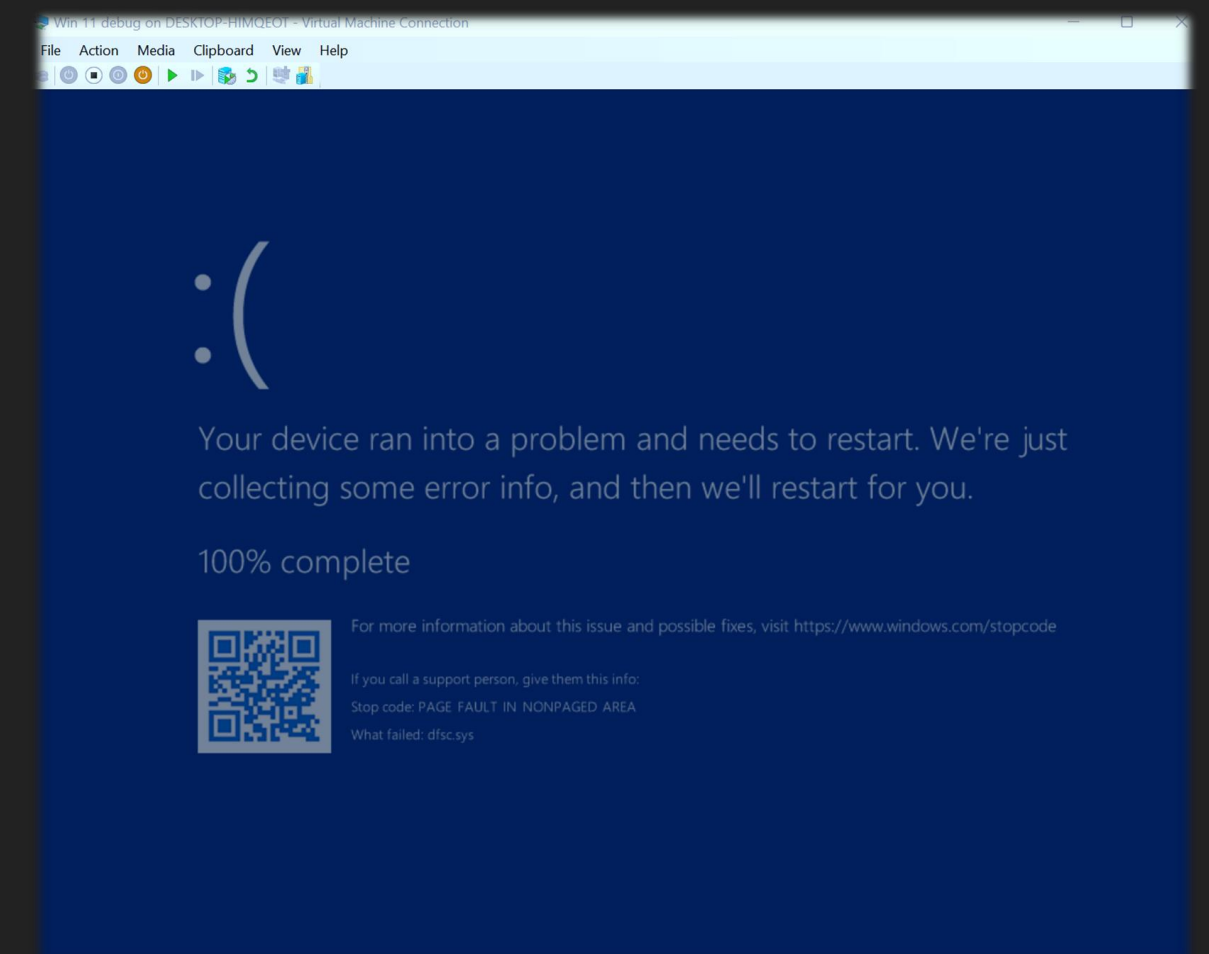


Why?

- 在執行紅隊過程中，經常取得一台機器控制權後，**權限過低無法獲取機密資訊**
 - 現階段已知提權方法**越來越難用**，條件越來越**嚴苛**
- Pwn2Own
 - 有機會的話還是會想要能在 Pwn2Own 上能攻下 Windows
- 具有很大的**影響度**

Why?

- 在執行紅隊過程中，經常取得一台機器控制權後，**權限過低無法獲取機密資訊**
- 現階段已知提權方法**越來越難用**，條件越來越**嚴苛**
- Pwn2Own
 - 有機會的話還是會想要能在 Pwn2Own 上能攻下 Windows
- 具有很大的**影響度**
- 個人興趣

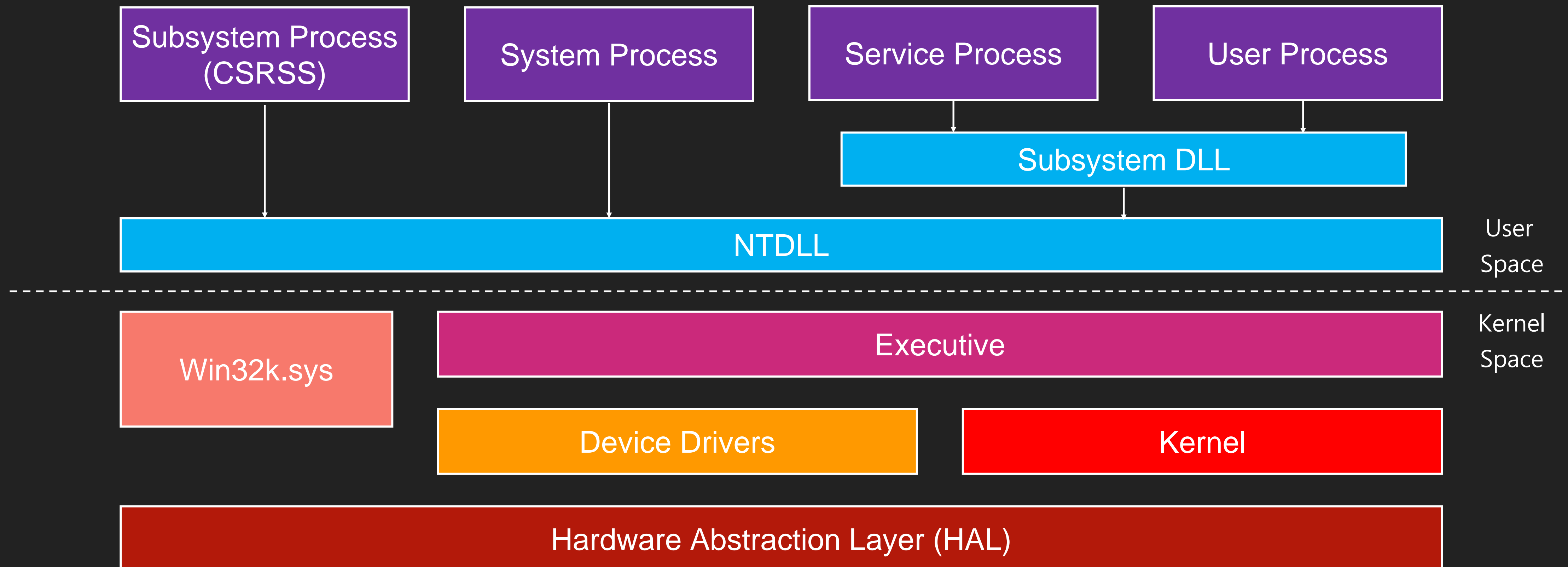




How to choose a target ?

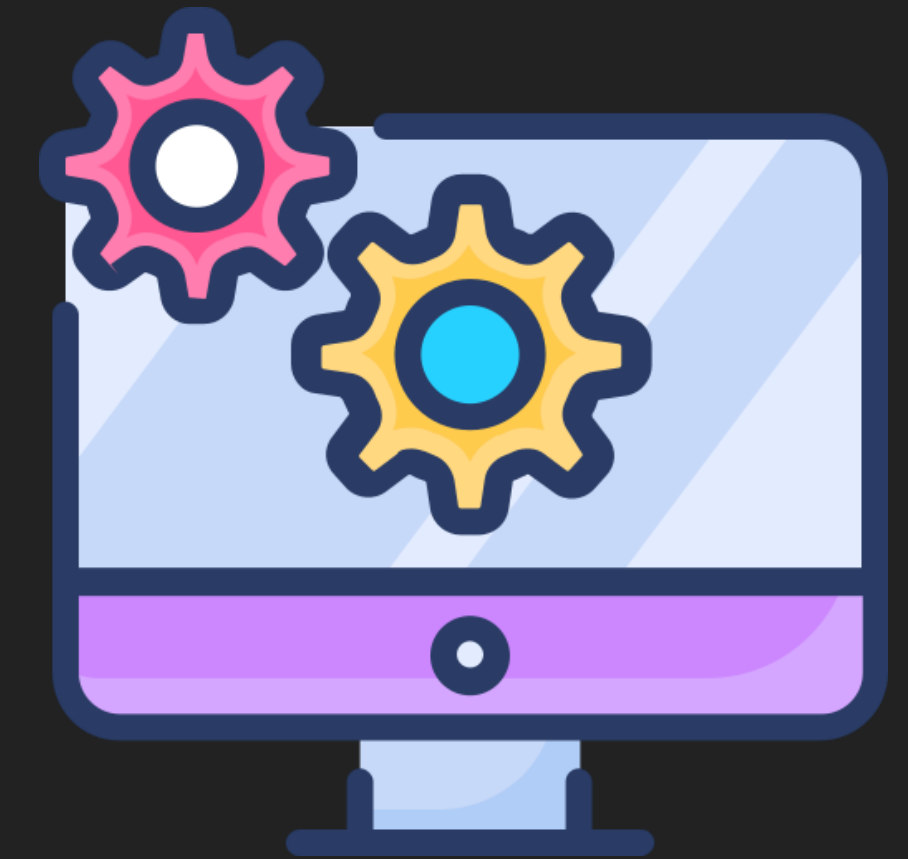
How to choose a target ?

Windows 架構圖



How to choose a target ?

- 提權主流
 - Service
 - Potato 系列
 - Windows Kernel



📖 README 📄 GPL-3.0 license

Juicy Potato (abusing the golden privileges)

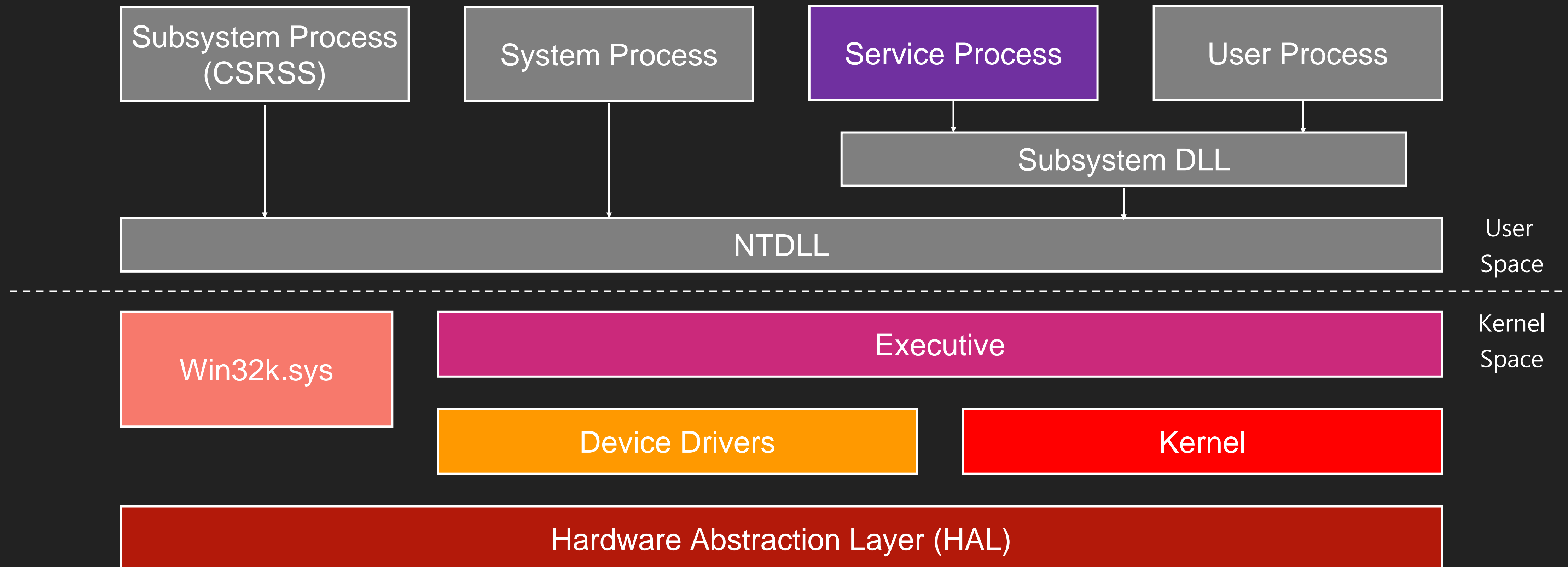
A sugared version of [RottenPotatoNG](#), with a bit of juice, i.e. another Local Privilege Escalation tool, from a Windows Service Accounts to NT AUTHORITY\SYSTEM

How to choose a target ?

- 提權主流
 - Service
 - ~~Potato~~ 系列
 - Windows Kernel

How to choose a target ?

Windows 架構圖



How to choose a target ?

- 提權主流
 - ~~Service~~
 - Windows Kernel
 - Pwn2Own 中只允許 Windows Kernel 的漏洞

Local Escalation of Privilege Category

An attempt in this category must be launched from within the target under test from a non-admin and non-root account. In this category, the entry must leverage a kernel vulnerability to escalate privileges.

Targets:

Target	Prize	Master of Pwn Points
Ubuntu Desktop	\$20,000	2
Microsoft Windows 11	\$30,000	3
Apple macOS	\$40,000	4

DEV✓*CORE*

**Windows Kernel
Elevation of Privilege**

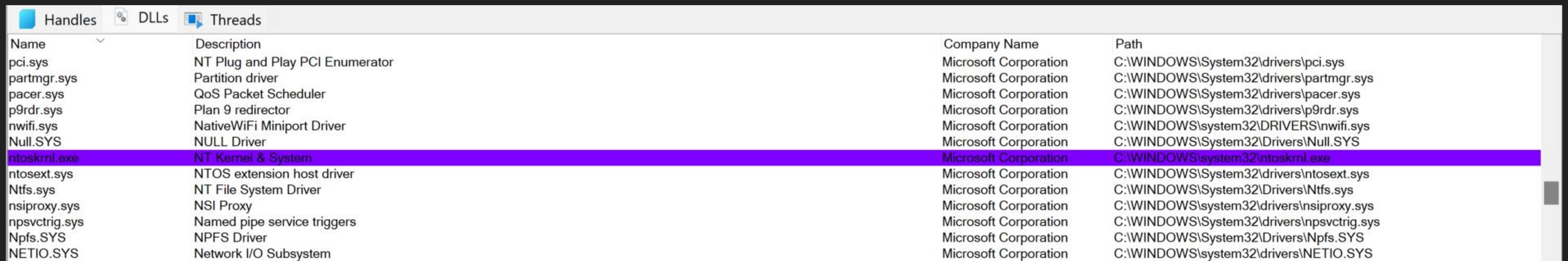


Find Vulnerability in Windows Kernel

Windows Kernel EoP

Find Vulnerability in Windows Kernel

- Ntoskrnl.exe
- Device driver



Name	Description	Company Name	Path
pci.sys	NT Plug and Play PCI Enumerator	Microsoft Corporation	C:\WINDOWS\System32\drivers\pci.sys
partmgr.sys	Partition driver	Microsoft Corporation	C:\WINDOWS\System32\drivers\partmgr.sys
pacer.sys	QoS Packet Scheduler	Microsoft Corporation	C:\WINDOWS\System32\drivers\pacer.sys
p9rdr.sys	Plan 9 redirector	Microsoft Corporation	C:\WINDOWS\System32\drivers\p9rdr.sys
nwifi.sys	NativeWiFi Miniport Driver	Microsoft Corporation	C:\WINDOWS\system32\DRIVERS\nwifi.sys
Null.SYS	NULL Driver	Microsoft Corporation	C:\WINDOWS\System32\Drivers\Null.SYS
ntoskrnl.exe	NT Kernel & System	Microsoft Corporation	C:\WINDOWS\system32\ntoskrnl.exe
ntosext.sys	NTOS extension host driver	Microsoft Corporation	C:\WINDOWS\System32\drivers\ntosext.sys
Ntfs.sys	NT File System Driver	Microsoft Corporation	C:\WINDOWS\System32\Drivers\Ntfs.sys
nsiproxy.sys	NSI Proxy	Microsoft Corporation	C:\WINDOWS\system32\drivers\nsiproxy.sys
npsvctrig.sys	Named pipe service triggers	Microsoft Corporation	C:\WINDOWS\System32\drivers\npsvctrig.sys
Npfs.SYS	NPFS Driver	Microsoft Corporation	C:\WINDOWS\System32\Drivers\Npfs.SYS
NETIO.SYS	Network I/O Subsystem	Microsoft Corporation	C:\WINDOWS\system32\drivers\NETIO.SYS

Windows Kernel EoP

Find Vulnerability in Windows Kernel

- 第三方 driver
 - AMD
 - NVIDIA
 - Dell
 - ...

新聞

Dell修補存在12年的驅動程式高風險漏洞

安全業者SentinelOne揭露Dell驅動程式DBUtil (dbutil_2_3.sys) 含有影響Windows裝置的安全瑕疵，Dell判定屬於存取控管不足漏洞，可讓具本機非管理員權限的攻擊者，取得核心模式執行權限

文/ 林妍臻 | 2021-05-05 發表

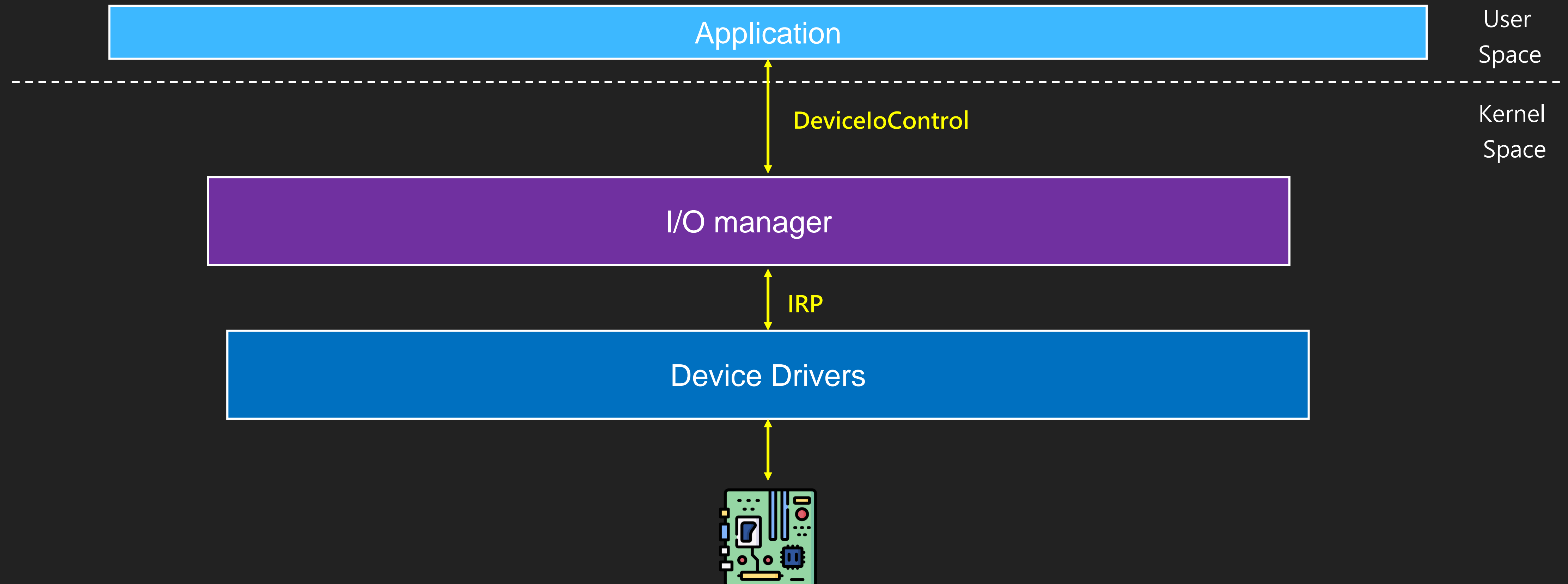
👍 讚 254

分享

Windows Kernel EoP

Find Vulnerability in Windows Kernel

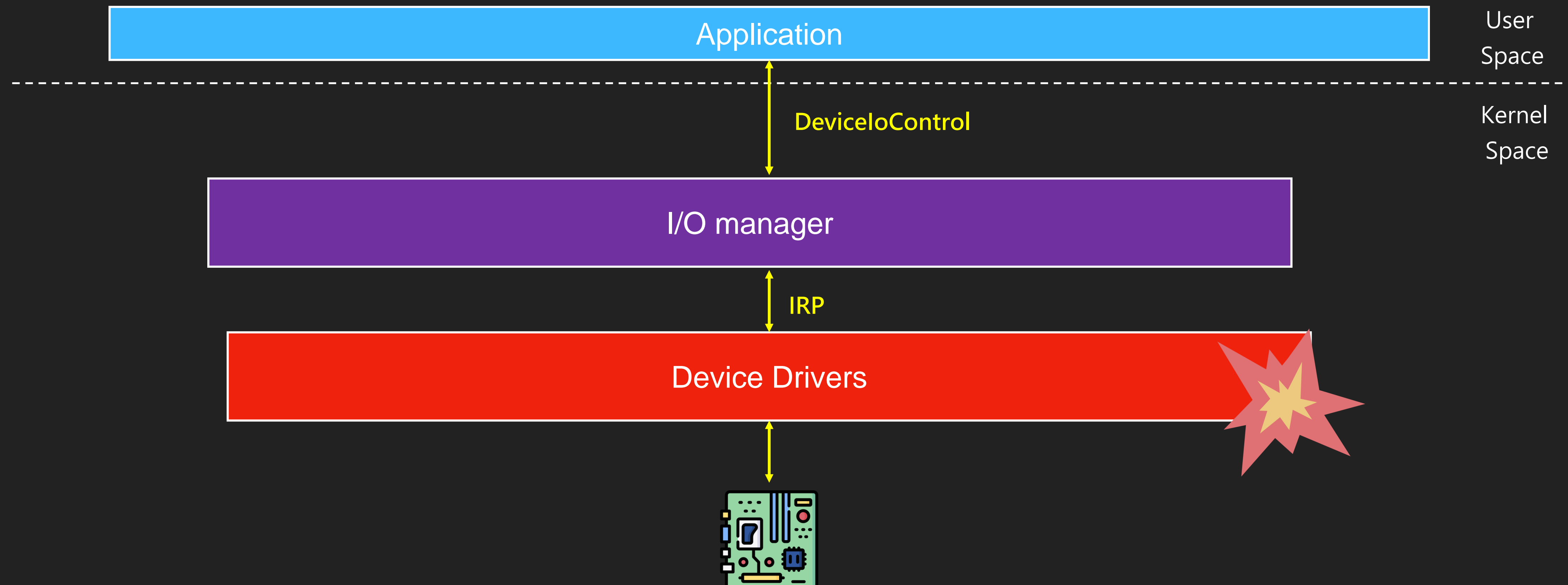
- User 透過 **DeviceIoControl/NtFsControlFile** 等 API 與 kernel 互動

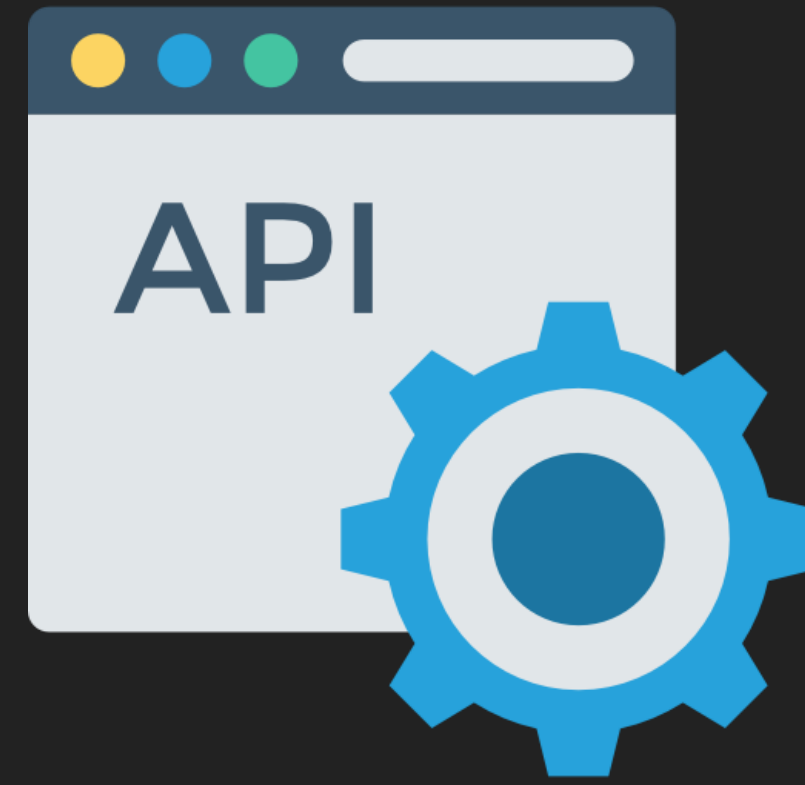


Windows Kernel EoP

Find Vulnerability in Windows Kernel

- User 透過 **DeviceIoControl/NtFsControlFile** 等 API 與 kernel 互動





Use Windows API to leak kernel address

Windows Kernel EoP

Use Windows API to leak kernel address

- Use `NtQuerySystemInformation` with `SystemModuleInformation` can get `ntoskrnl` address.

C++

Copy

```
__kernel_entry NTSTATUS NtQuerySystemInformation(  
    [in]          SYSTEM_INFORMATION_CLASS SystemInformationClass,  
    [in, out]    PVOID SystemInformation,  
    [in]          ULONG SystemInformationLength,  
    [out, optional] PULONG ReturnLength  
);
```



But ...

DEV✓*CORE*

Windows Kernel EoP

KASLR Leak Restriction

- Windows NtQuerySystemInformation 系列的 leak kernel pointer 方法已無法在沒有 **SeDebugPrivilege** 情況下使用 (24H2)



Yarden Shafir
@yarden_shafir

One example: if the caller should not receive kernel addresses, calling NtQuerySystemInformation with SystemModuleInformation will not leak kernel image addresses.

[翻譯貼文](#)

```
Modules = ProcessModules->Modules;
for ( i = (_KLDLDR_DATA_TABLE_ENTRY *)PsLoadedModuleList;
      i != (_KLDLDR_DATA_TABLE_ENTRY *)&PsLoadedModuleList;
      i = (_KLDLDR_DATA_TABLE_ENTRY *)i->InLoadOrderLinks.Flink )
{
    length = neededLength + sizeof(_RTL_PROCESS_MODULE_INFORMATION);
    if ( neededLength + sizeof(_RTL_PROCESS_MODULE_INFORMATION) < neededLength )
        return STATUS_UNSUCCESSFUL;
    neededLength += sizeof(_RTL_PROCESS_MODULE_INFORMATION);
    if ( ModuleInformationLength >= length )
    {
        imageBase = 0i64;
        if ( !IsRestrictedByFeatureFlag )
            imageBase = i->DllBase;
        Modules->ImageBase = imageBase;
        Modules->ImageSize = i->SizeOfImage;
        Modules->Flags = i->Flags;
        Modules->LoadCount = i->LoadCount;
        Modules->LoadOrderIndex = numberOfModules;
        Modules->InitOrderIndex = 0;
        pathName.Buffer = (char *)Modules->FullPathName;
    }
}
```

Windows kernel leak 漏洞
開始變得不可或缺



Use Vulnerability to create arbitrary
memory write primitive

Windows Kernel EoP

Use Vulnerability to create arbitrary memory write primitive

- 想辦法使用漏洞做到任意記憶體寫入
- Overwrite kernel object
 - IoRing
 - Windows Notification Framework
 - Previous mode
-

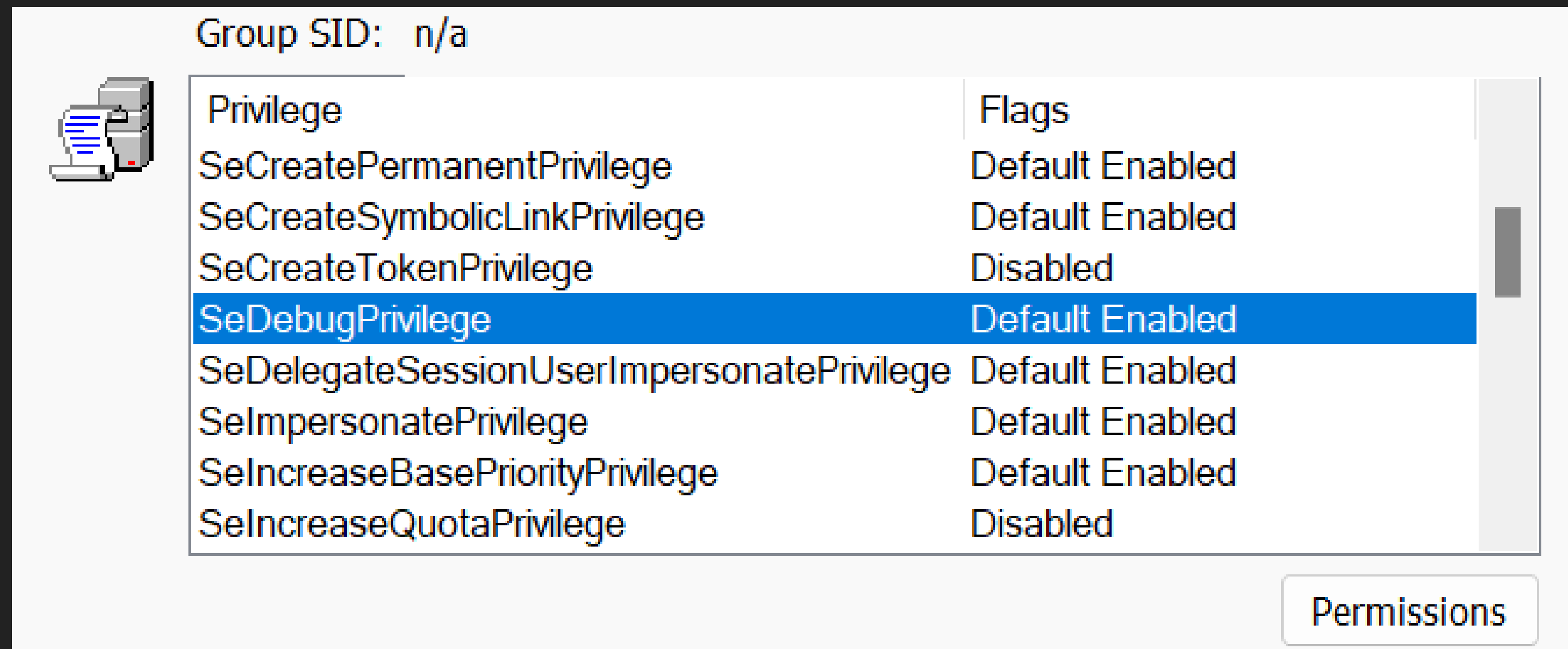


Modify or create a high privilege Token

Windows Kernel EoP

Modify or create a high privilege Token

- 置換或修改當前 Process Token
- 變成具有高權限的 Privilege 或直接變成 System Token



Group SID: n/a

Privilege	Flags
SeCreatePermanentPrivilege	Default Enabled
SeCreateSymbolicLinkPrivilege	Default Enabled
SeCreateTokenPrivilege	Disabled
SeDebugPrivilege	Default Enabled
SeDelegateSessionUserImpersonatePrivilege	Default Enabled
SeImpersonatePrivilege	Default Enabled
SeIncreaseBasePriorityPrivilege	Default Enabled
SeIncreaseQuotaPrivilege	Disabled

Permissions

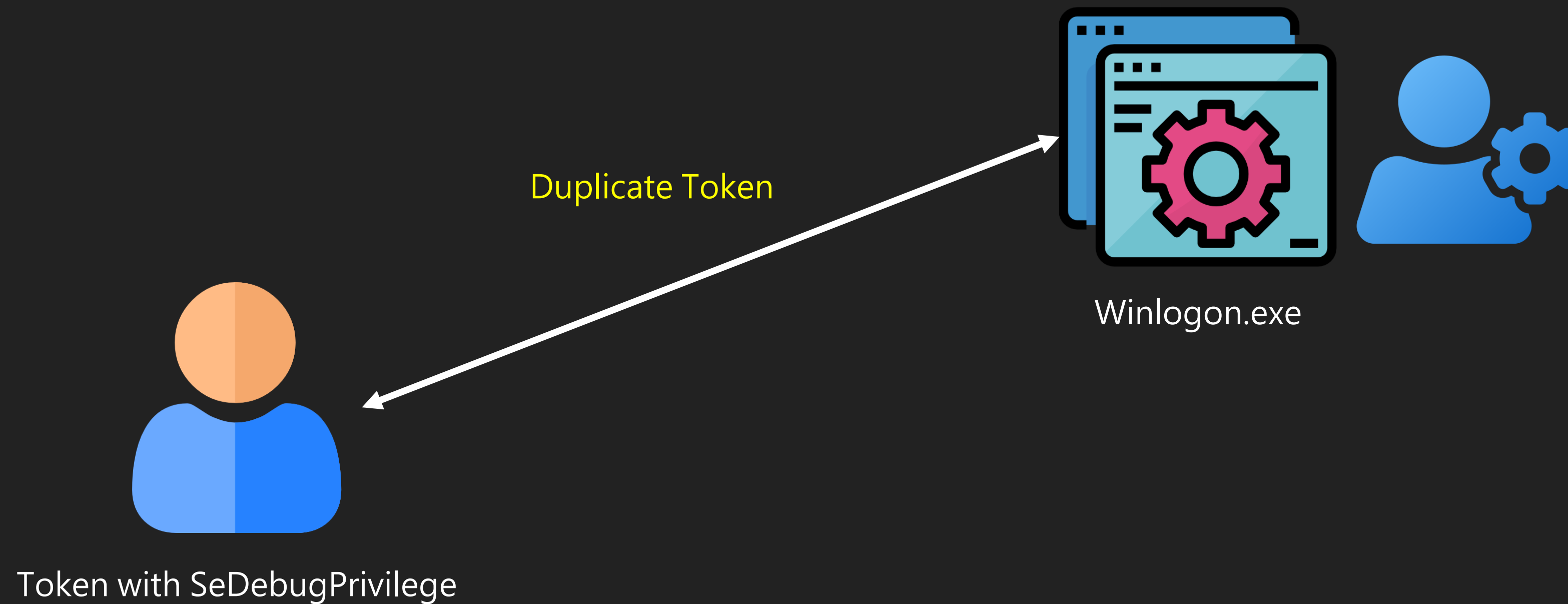


Create Process with Token

Windows Kernel EoP

Create Process with Token

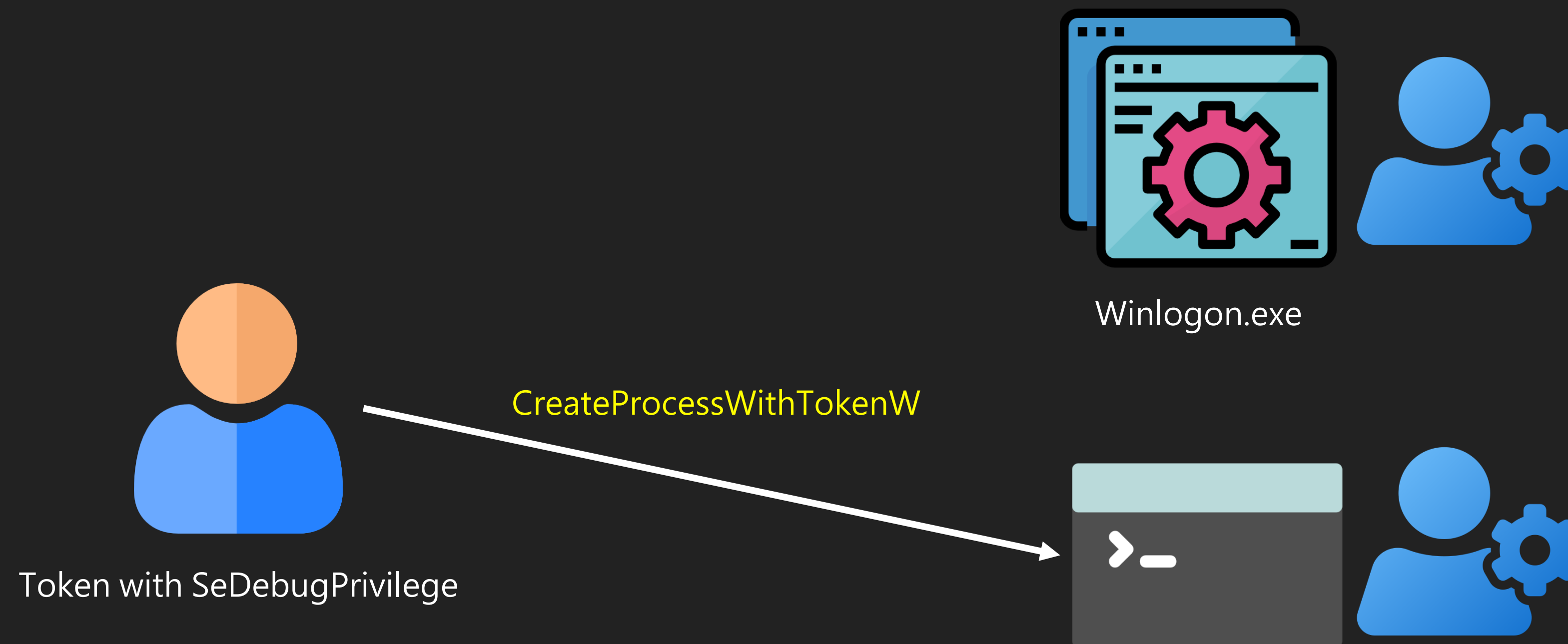
- 利用 DuplicateToken 去 Duplicate 高權限 Process 的 Token



Windows Kernel EoP

Create Process with Token

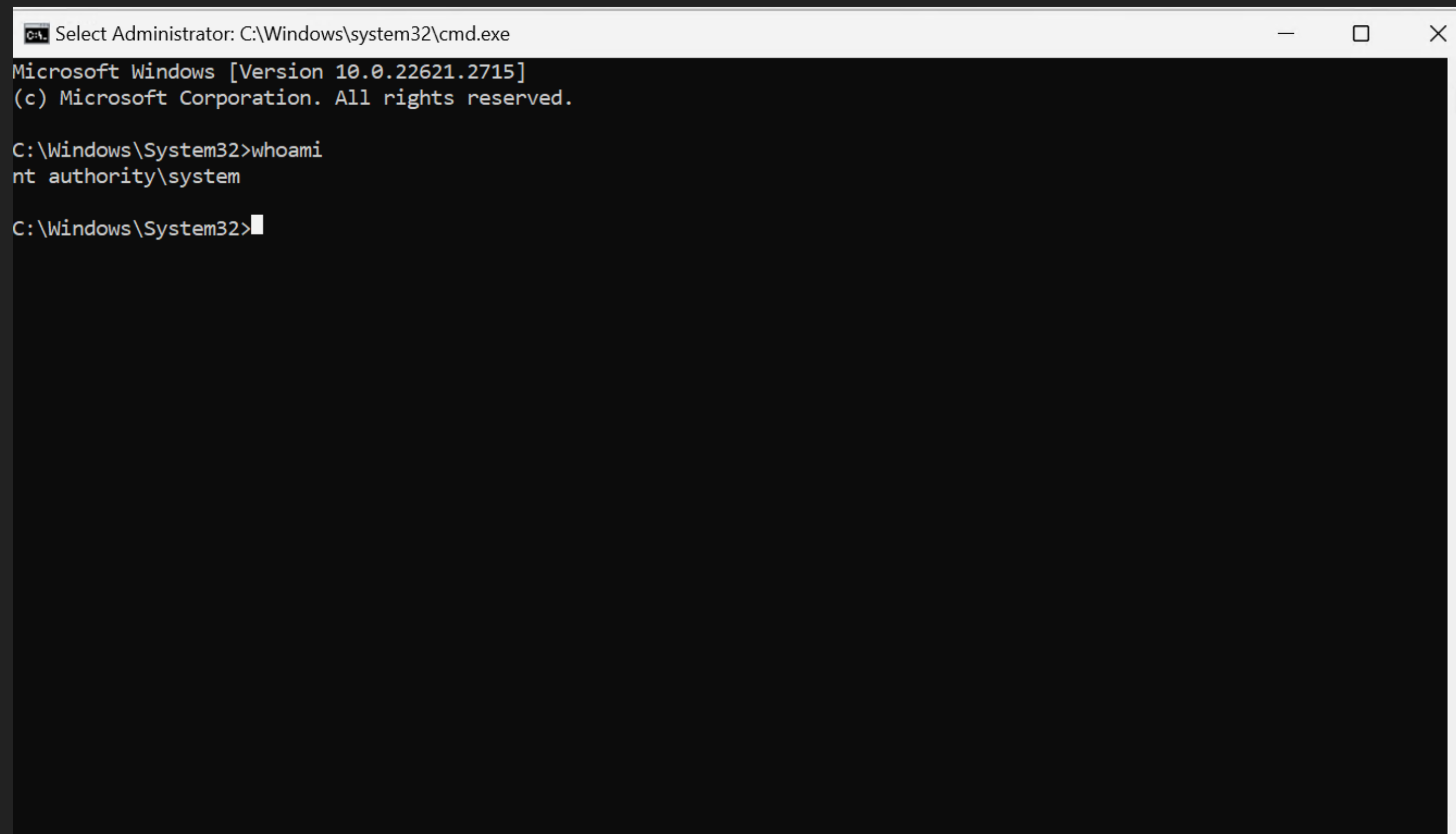
- 利用 `CreateProcessWithTokenW` 創建高權限的 Process



Windows Kernel EoP

Create Process with Token

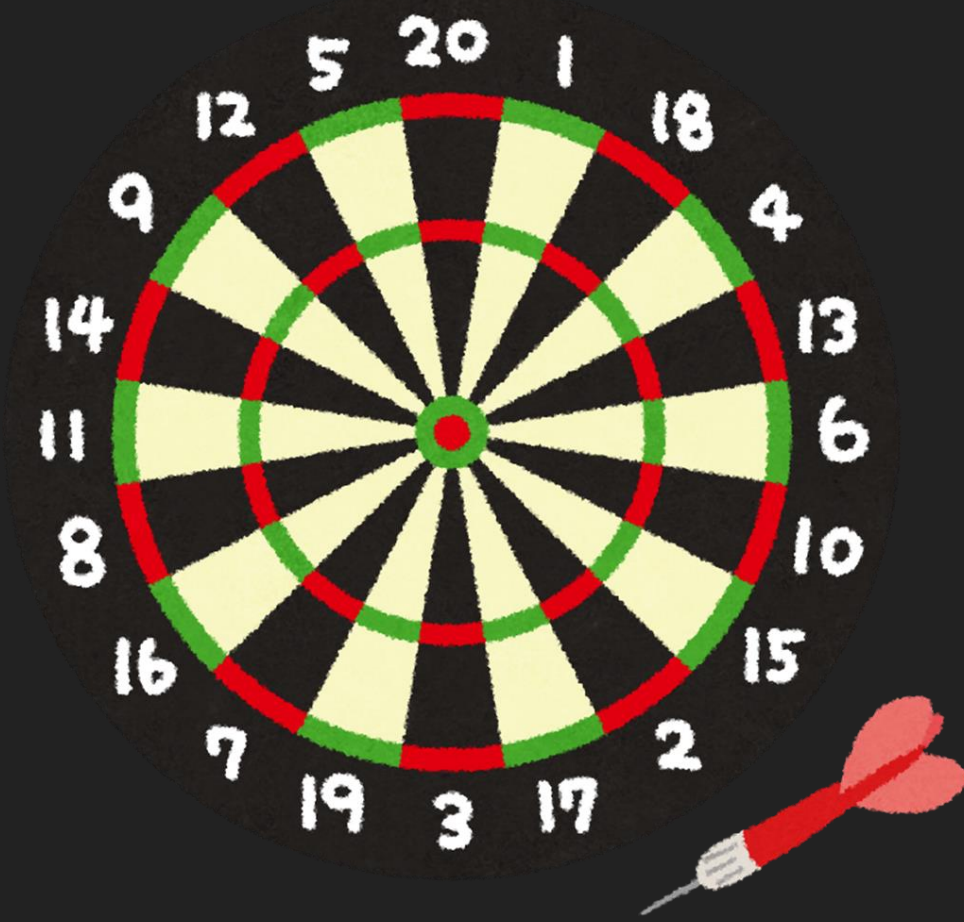
- 利用 CreateProcessWithTokenW 創建高權限的 Process



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>whoami
nt authority\system

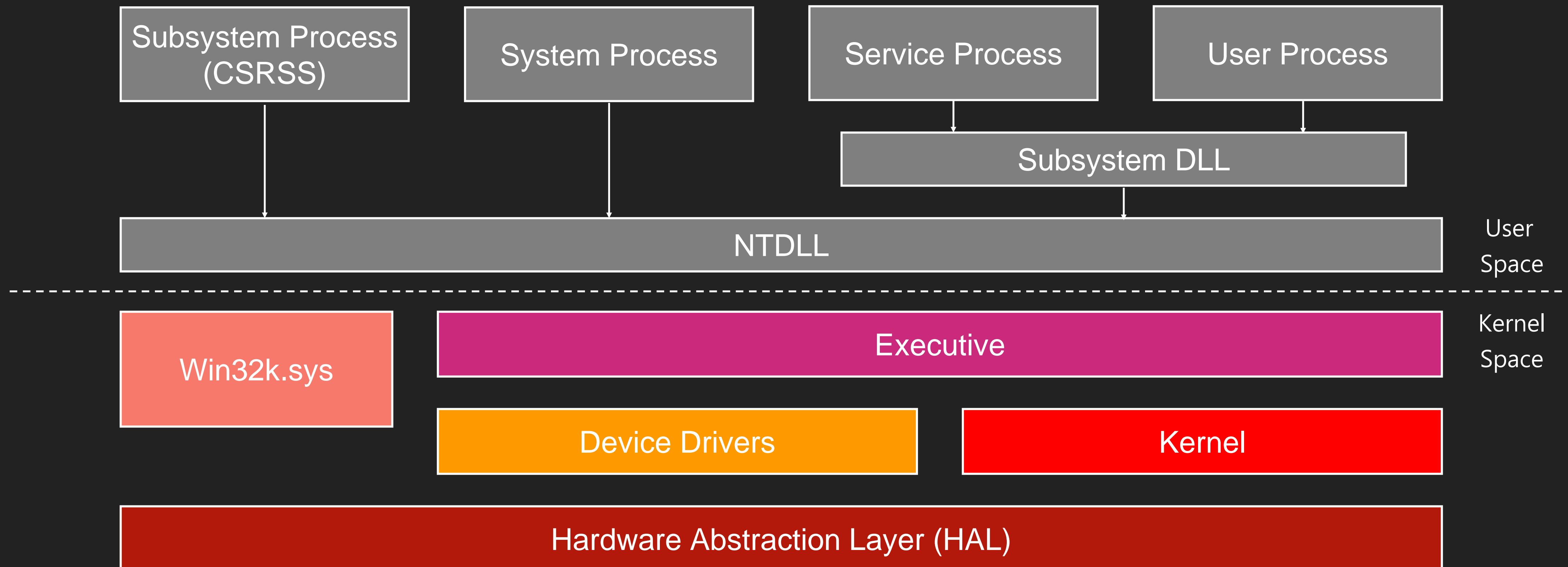
C:\Windows\System32>
```



Pick up a target

Pick up a target

Windows 架構圖



Pick up a target

- 過去漏洞多不多
- 程式碼品質如何

Name	
CVE-2024-21346	Win32k Elevation of Privilege Vulnerability
CVE-2024-20686	Win32k Elevation of Privilege Vulnerability
CVE-2024-20683	Win32k Elevation of Privilege Vulnerability
CVE-2023-41772	Win32k Elevation of Privilege Vulnerability
CVE-2023-36776	Win32k Elevation of Privilege Vulnerability
CVE-2023-36743	Win32k Elevation of Privilege Vulnerability
CVE-2023-36732	Win32k Elevation of Privilege Vulnerability
CVE-2023-36731	Win32k Elevation of Privilege Vulnerability
CVE-2023-36011	Win32k Elevation of Privilege Vulnerability
CVE-2023-35631	Win32k Elevation of Privilege Vulnerability
CVE-2023-35337	Win32k Elevation of Privilege Vulnerability
CVE-2023-29336	Win32k Elevation of Privilege Vulnerability
CVE-2023-28274	Windows Win32k Elevation of Privilege Vulnerability

Pick up a target

DEV✓CORE

- 過去漏洞多不多
- 太多人看的目標不選
 - 容易撞洞

The image shows a promotional banner for DEVCORE. At the top, there is a blue patterned border. Below it, an orange pill-shaped button contains the word "COLLISION". The center of the banner features the word "DEVCORE" in large white letters. Below this, a blue pill-shaped button contains the word "TARGETING". Underneath, the text "Lexmark MC3224 in the Printer category" is displayed. At the bottom left, there is a yellow pill-shaped button with "PRIZE \$" above it and "\$5,000" inside. At the bottom right, there is a blue pill-shaped button with "POINTS" above it and "1" inside.

PRIZE \$	Lexmark MC3224 in the Printer category	POINTS
\$5,000		1

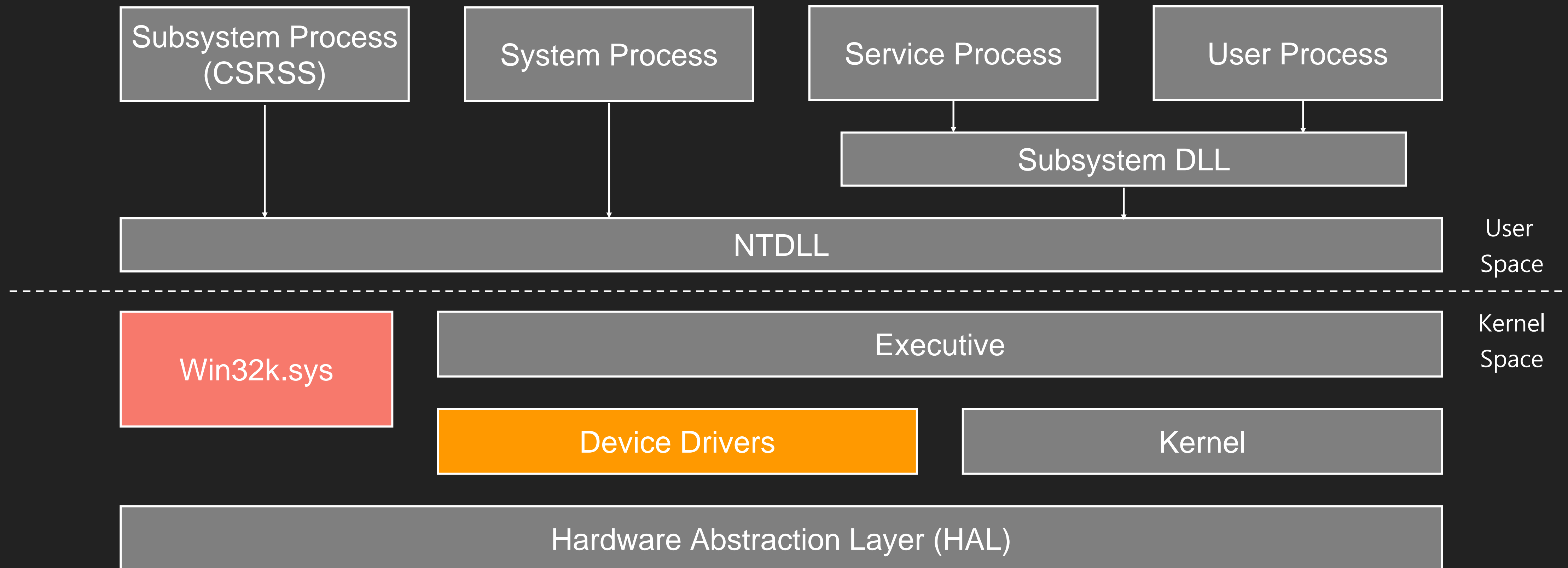
Pick up a target

- 過去漏洞多不多
- 太多人看的目標不選
 - 容易撞洞
- 有興趣的目標



Pick up a target

Windows 架構圖



Pick up a target

- Win32k
 - 每個月都有洞，但需要花不少時間搞懂機制，資源很多



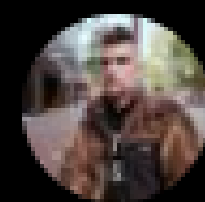
Pick up a target

- Win32k
 - 每個月都有洞，但需要花不少時間搞懂機制，資源很多
 - 但以目標為 Pwn2Own 來說，可能會來不及



Pick up a target

- Win32k
 - 每個月都有洞，但需要花不少時間搞懂機制，資源很多
 - 但以目標為 Pwn2Own 來說，可能會來不及
 - 之後會改 Rust 寫



David Weston (DWIZZZLE) ✓

@dwizzzleMSFT

My presentation slides for "Windows 11: security by-default" from @BlueHatIL covering:

Rust in win32k, Adminless Windows, Token Binding, Sandboxing win32, and more!

Pick up a target

- Device Driver
 - 相對單純很多
 - 大部分都出現過不少漏洞
- Afd.sys
- Ntfs.sys
- cldflt.sys
- clfs.sys
- ...



<https://securityintelligence.com/x-force/patch-tuesday-exploit-wednesday-pwning-windows-ancillary-function-driver-winsock/>

Pick up a target

- 可以從每個月的 Patch Tuesday 看看哪個比較軟

The screenshot shows a webpage for the Zero Day Initiative. The header includes the logo and navigation links: PRIVACY, WHO WE ARE, HOW IT WORKS, BLOG, ADVISORIES, LOG IN, and SIGN UP. A yellow 'SUBSCRIBE' button is visible on the right. The main content area features the title 'THE FEBRUARY 2024 SECURITY UPDATE REVIEW' and the author 'February 12, 2024 | Dustin Childs'. Below the title, there is a section titled '< BACK TO THE BLOG' with two circular icons (a heart and a left arrow). The text below this section reads: 'It's the second patch Tuesday of the year, and Adobe and Microsoft have released a fresh crop of security updates just in time to be our Valentine. Take a break from your other activities and join us as we review the details of their latest advisories. For those interested in the Microsoft 0-day discovered by the ZDI Threat Hunting Team, you can watch this special edition of the Patch Report:'

File System 系列好像不錯看

Pick up a target

File System Driver

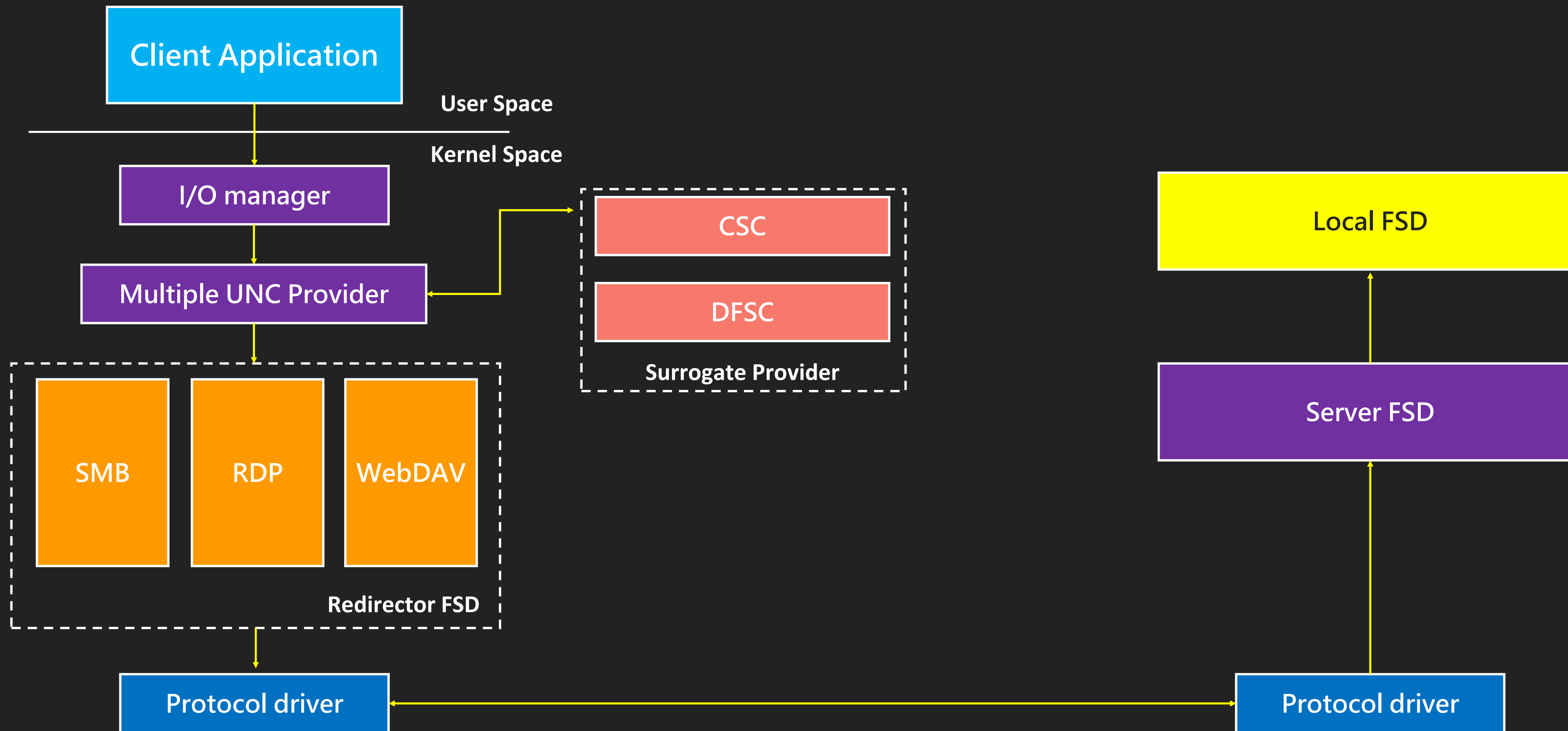
- Local
 - Ntfs
 - Fastfat
 - ...
- Remote (Server 、 Client)
 - RDP
 - SMB
 - ...



```
Windows PowerShell
PS C:\Users\angelboy> type \\server\test\test.txt
```

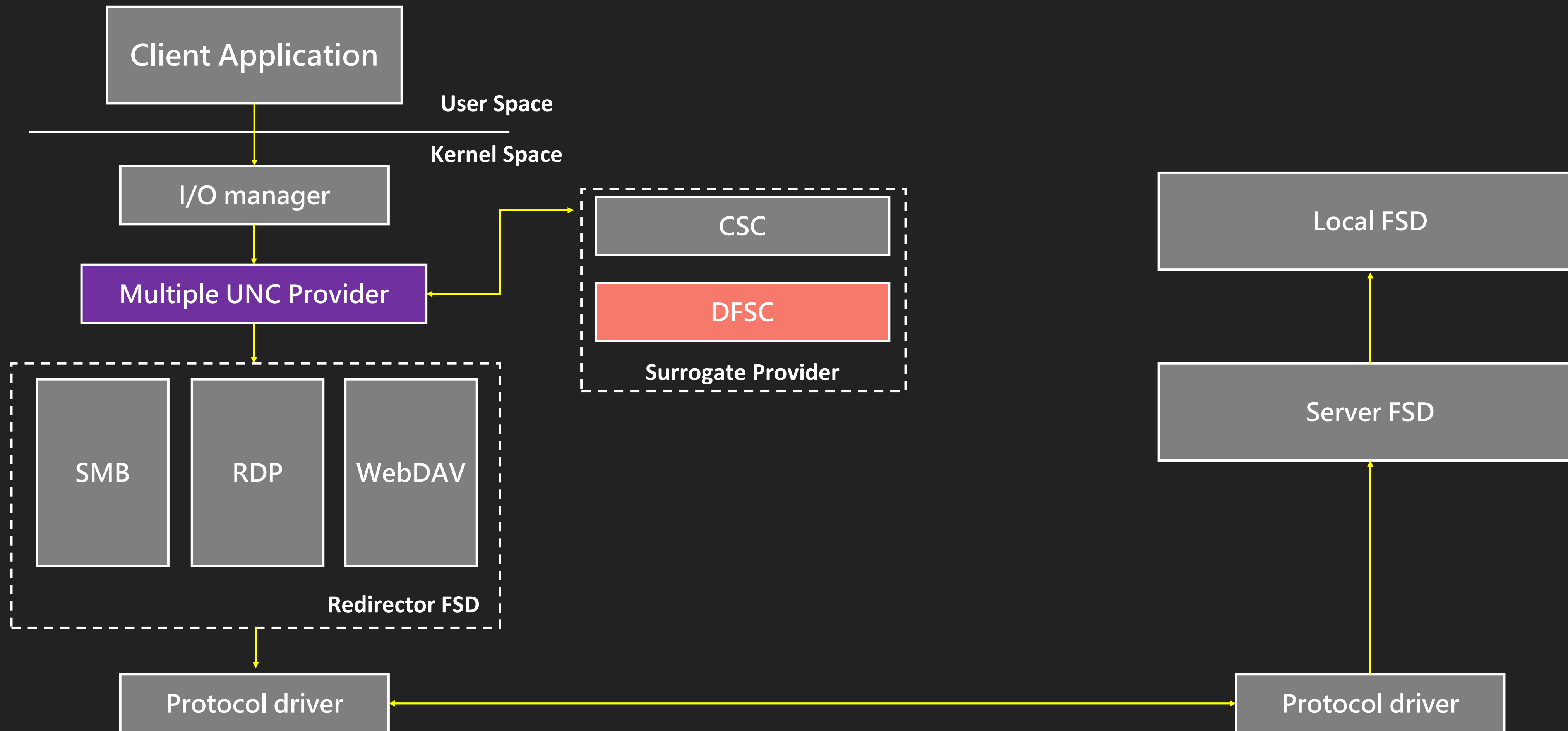

Pick up a target

File System Driver



Pick up a target

File System Driver

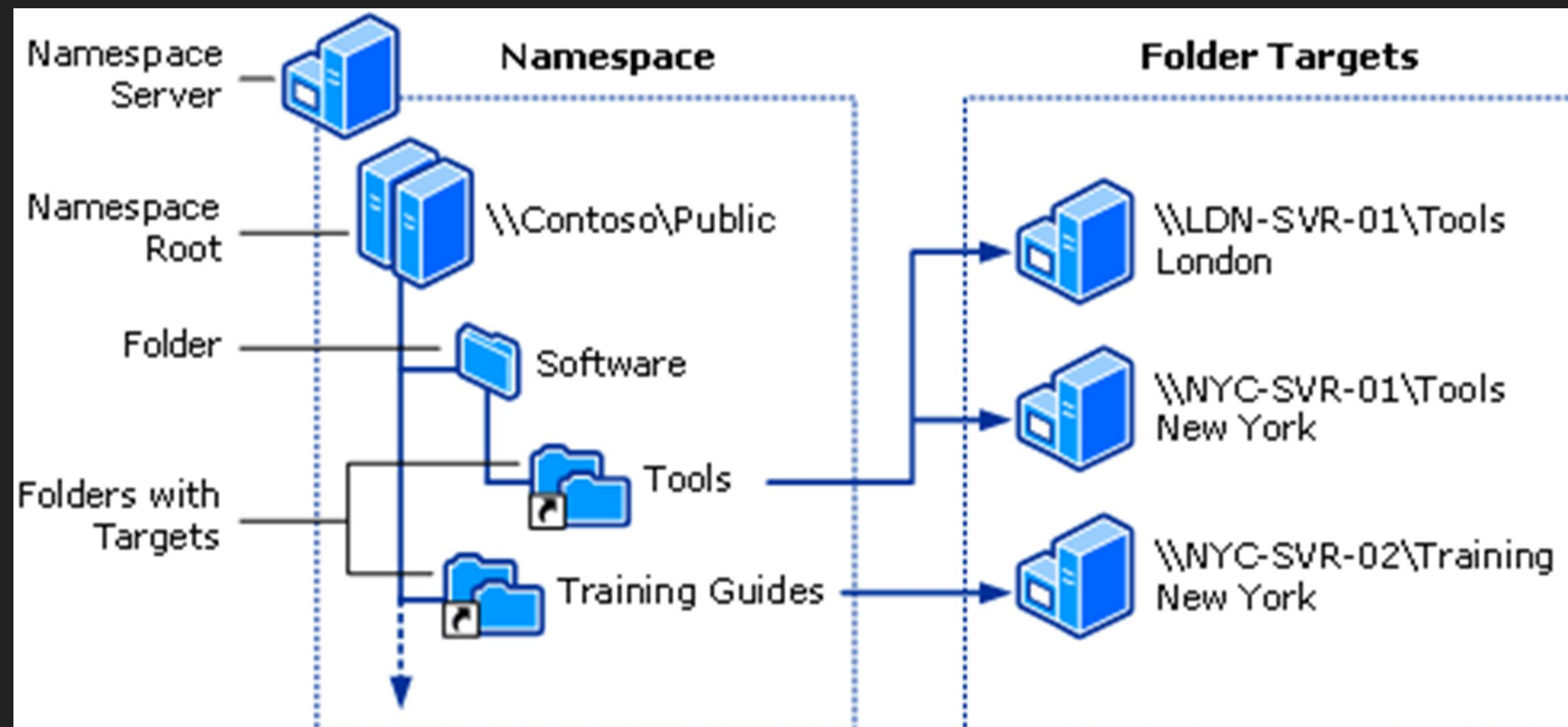


DEV✓*CORE*

**Distributed File
System**

Distributed File System

- 當我們在域中 access 一個 UNC 路徑時，DFS server 會將 domain 解出真正的位置後，再讓他去 access 真正位置，可用來做為 Load balance 用



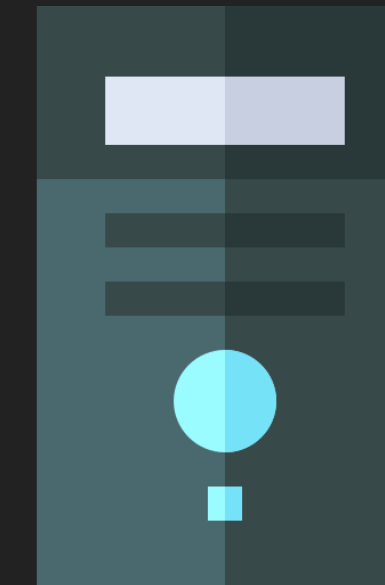
Distributed File System

- Communication



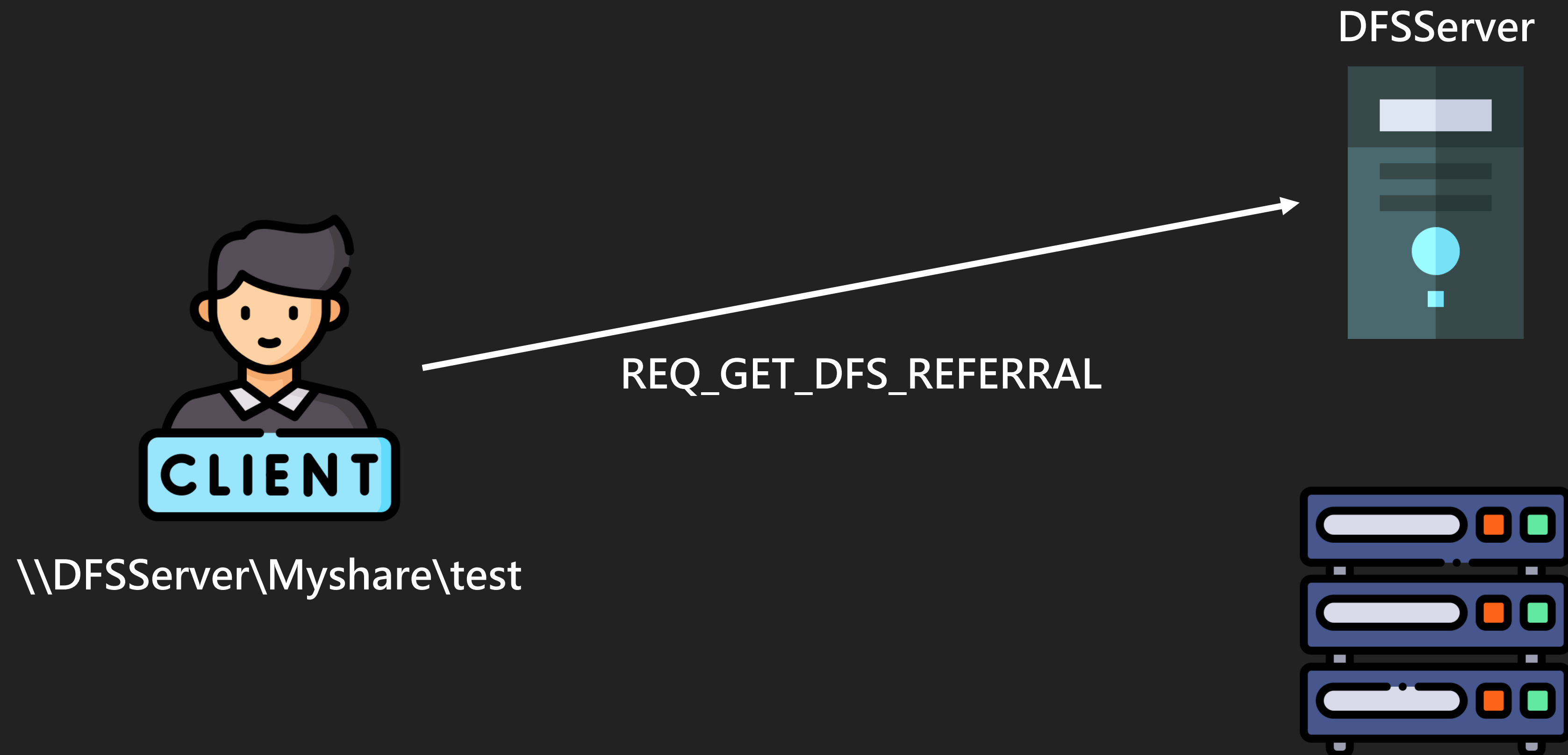
\\DFSServer\Myshare\test

DFSServer



Distributed File System

- Communication



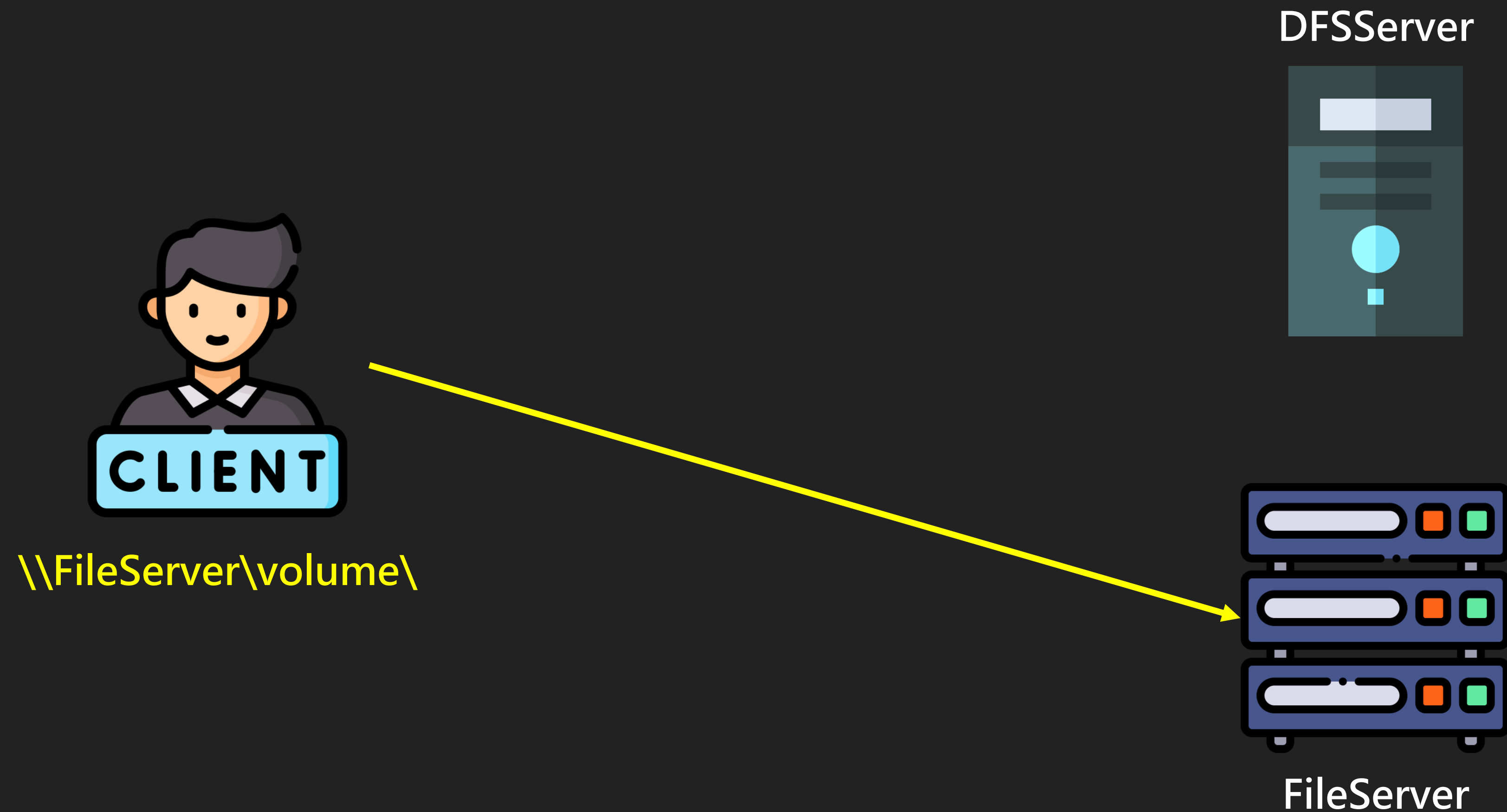
Distributed File System

- Communication



Distributed File System

- Communication



Distributed File System

- Component
 - dfs.sys
 - Server 端，通常只會存在 Windows Server
 - dfsc.sys
 - Client 端，一般的 Windows 電腦環境中都會有

Pick up a target

過去漏洞

- CVE-2016-7185
 - Found by James Forshaw
 - DFS Client Driver **Arbitrary Drive Mapping EoP**
 - The DFS Client driver and running by default insecurely **creates and deletes drive letter symbolic links in the current user context** leading to EoP

Distributed File System

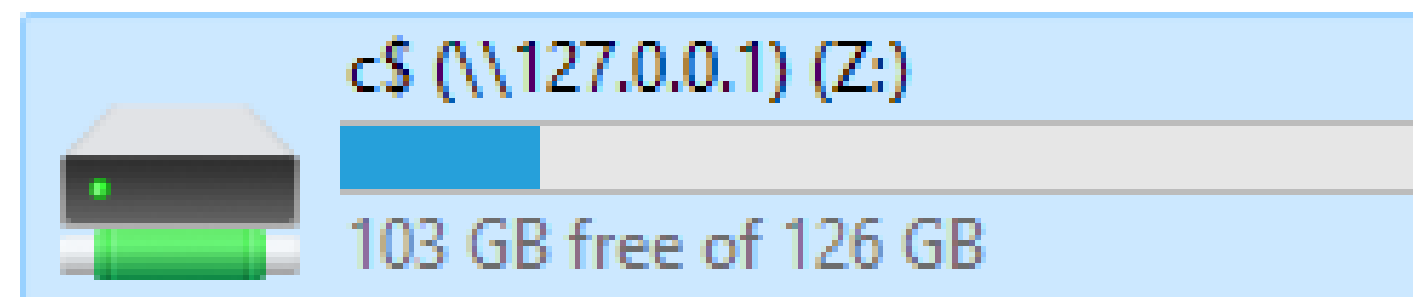
Create Drive Letter

- `net use` 網路磁碟機功能

```
C:\Users\angelboy>net use z: \\127.0.0.1\c$  
The command completed successfully.
```



Network locations



Distributed File System

Create Drive Letter

- DfscFsctrlCreateDriveLetter
 - FSCTL - 0x601E0
 - FILE_ANY_ACCESS
- 直接對 `\\Device\DfsClient` 送 FSCTL 及相對應的 input 就可以使用該功能

Distributed File System

Create Drive Letter

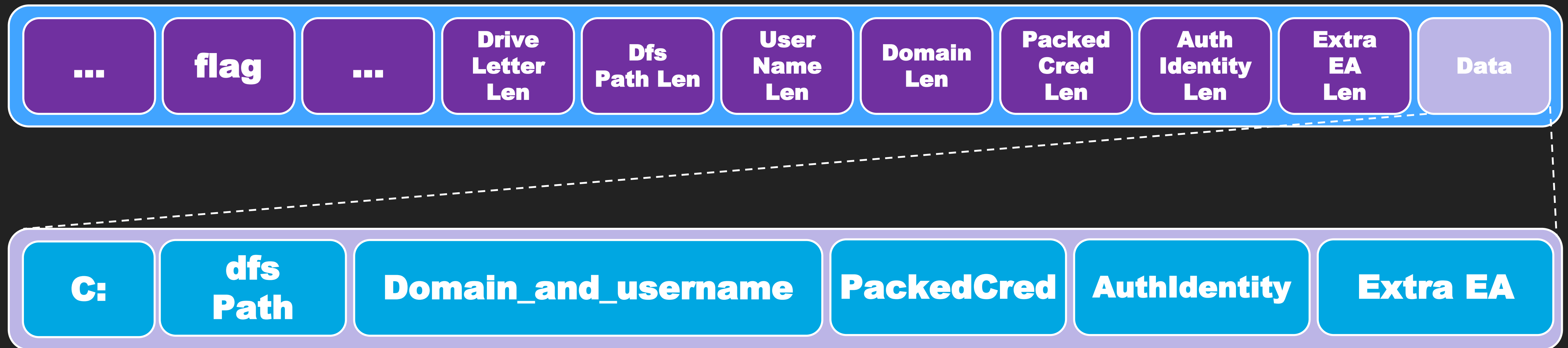
- Create Drive Letter Request (Input Buffer)

```
struct createdrive_req
{
    char w[2];
    unsigned short flag;
    int dword4;
    unsigned short driveLetter_len;
    unsigned short dfs_path_len;
    unsigned short username_len;
    unsigned short domain_len;
    unsigned short PackedCredentialsString_len;
    unsigned short AuthIdentity_len;
    unsigned short extra_ea_len;
    wchar_t data[0x10];
}
```

Distributed File System

Create Drive Letter

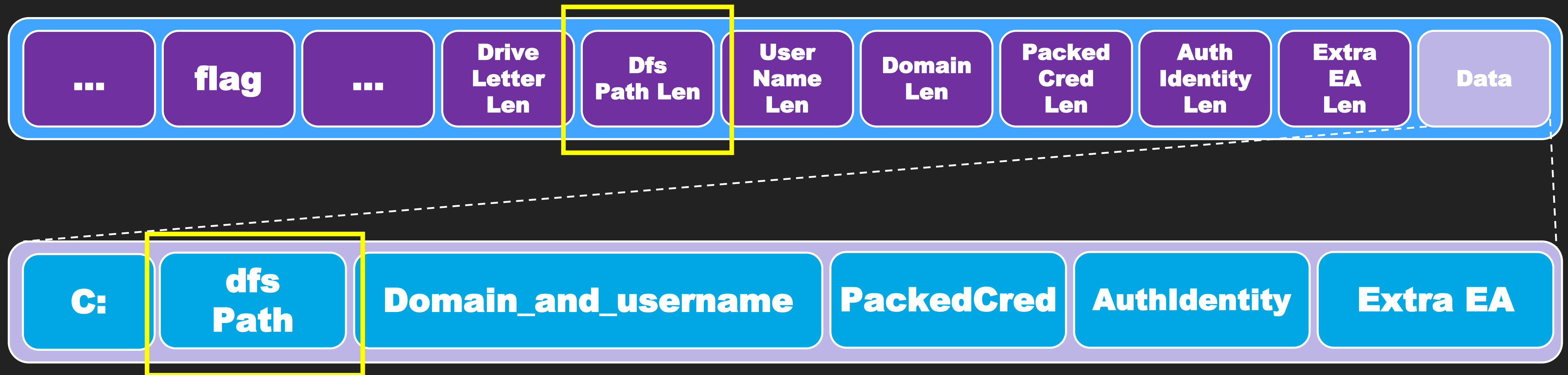
- Create Drive Letter Request (Input Buffer)



Distributed File System

Create Drive Letter

- Create Drive Letter Request (Input Buffer)



Distributed File System

Create Drive Letter

- 當 dfsc driver 收到 **Create Drive Letter** Request 後，會直接與 target 嘗試建立 Symbolic Link 及建立連線
 - 根據你 LUID/Driver Letter/dfs path 來串接 Symbolic Link 及要連去 remote 的路經
 - $Z \leftrightarrow \backslash\text{Device}\backslash\text{Mup}\backslash\text{DfsClient}\backslash;c:\text{LogonId}\backslash\text{FileServer}\backslash\text{xxx}$

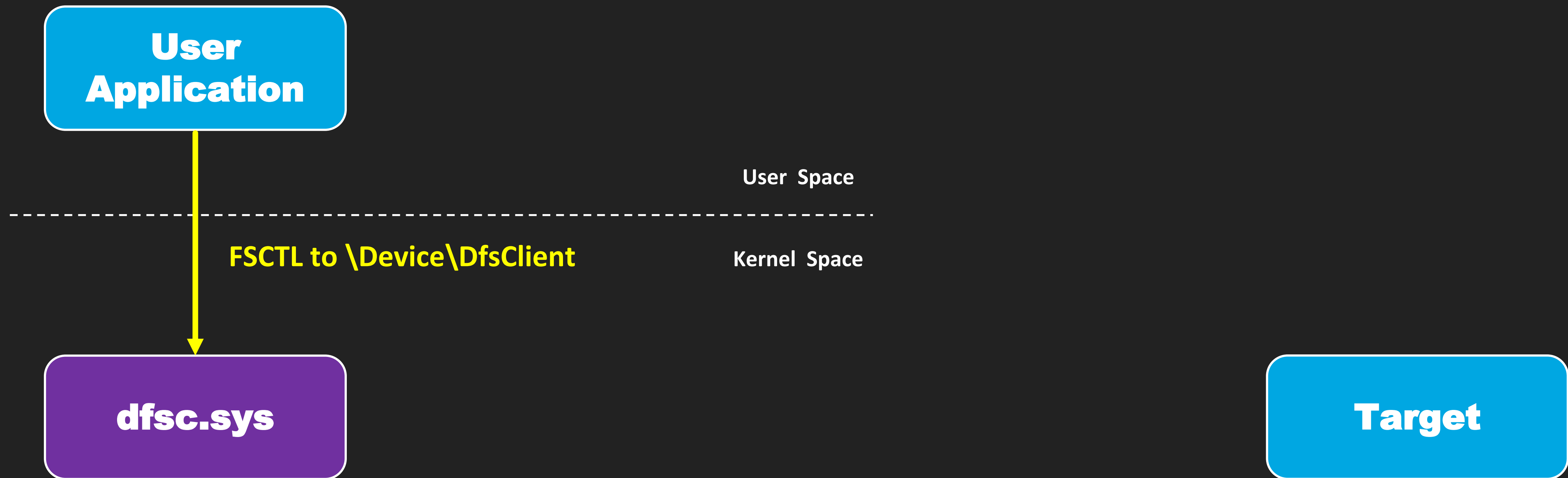
Distributed File System

Create Drive Letter

- DfscCreateTreeConnectName
 - 後續就會丟給 CreateFile 來去跟 Server 建立連線
 - IPC\$

Distributed File System

Create Drive Letter



Distributed File System

Create Drive Letter

**User
Application**

User Space

Kernel Space

dfsc.sys

Target

Create Symbolic Link to Target Path

Z: -> \Device\Mup\DfsClient\;z:LogonId\Target\xxx

Distributed File System

Create Drive Letter

**User
Application**

User Space

Kernel Space

dfsc.sys

Target

CreateFile over SMB

Z: -> \Device\Mup\DfsClient\;z:LogonId\Target\xxx

Distributed File System

Create Drive Letter

**User
Application**

User Space

Kernel Space

dfsc.sys

Target

if Error

It would delete Symobilc Link

Z: -> \Device\Mup\DfsClient\;z:LogonId\Target\xxx

DEV✓*CORE*

Useless Vulnerability

Useless Vulnerability

CVE-2022-34719

Windows Distributed File System (DFS) Elevation of Privilege Vulnerability

CVE-2022-34719

Security Vulnerability

Released: Sep 13, 2022

Useless Vulnerability

CVE-2022-34719

- 在 DfscCreateTreeConnectName 中
 - 並沒有對 UNICODE_STRING 的長度做檢查且使用 USHORT 來分配字符串的空間，造成 integer overflow

```
totalLen = LogonIdString.Length + DevicePath.Length + original_path->Length + 0x1E;
dest->MaximumLength = totalLen;
Pool2 = ExAllocatePool2(0x102i64, totalLen, 0x74436644i64); // totalLen is USHORT
dest->Buffer = (PWSTR)Pool2;
if ( !Pool2 )
    return -1073741670;
dest->Length = 0x16;
*( _OWORD * )Pool2 = *( _OWORD * )L"\\Device\\Mup"; // overflow if totalLen = 0
*( _DWORD * )(Pool2 + 0x10) = *( _DWORD * )L"Mup";
*( _WORD * )(Pool2 + 0x14) = SourceString[0xA];
```


Useless Vulnerability

CVE-2022-34719

- 在 `DfscCreateTreeConnectName` 中
 - 後續則直接將字串寫到分配出來的 `buffer` 中，造成 `heap overflow`

```
totalLen = LogonIdString.Length + DevicePath.Length + original_path->Length + 0x1E;
dest->MaximumLength = totalLen;
Pool2 = ExAllocatePool2(0x102i64, totalLen, 0x74436644i64); // totalLen is USHORT
dest->Buffer = (PWSTR)Pool2;
if ( !Pool2 )
    return -1073741670;
dest->Length = 0x16;
*(_OWORD *)Pool2 = *(_OWORD *)L"\\Device\\Mup"; // overflow if totalLen = 0
*(_DWORD *) (Pool2 + 0x10) = *(_DWORD *)L"Mup";
*(_WORD *) (Pool2 + 0x14) = SourceString[0xA];
```

Useless Vulnerability

CVE-2022-34719

- 在 `DfscCreateTreeConnectName` 中
 - Windows 中字符串大多都以 `UNICODE_STRING` 存
 - 長度都是 `USHORT`，常常有 integer overflow 發生

```
typedef struct _UNICODE_STRING {
    USHORT Length;
    USHORT MaximumLength;
    PWSTR Buffer;
} UNICODE_STRING, *PUNICODE_STRING;
```

Useless Vulnerability

CVE-2022-34719

- Exploitation?
 - 只能蓋 Pool Header 四個 Bytes

Useless Vulnerability

CVE-2022-34719

- Exploitation?
 - 只能蓋 Pool Header 四個 Bytes
 - 蓋的內容不可控

Useless Vulnerability

CVE-2022-34719

- Exploitation?
 - 只能蓋 Pool Header 四個 Bytes
 - 蓋的內容不可控
 - Useless

“排 heap 就是浪費時間，不如再找一個洞”

- Angelboy 10.17.2022

DEVCORE SECURITY
CONSULTING

- Meh @ DEVCORE CONF 2023

DEVCORE

DEV✓*CORE*

LeakLess
Vulnerability

LeakLess Vulnerability

DEV✓CORE

CVE-2022-38025

Windows Distributed File System (DFS) Information Disclosure Vulnerability

CVE-2022-38025

Security Vulnerability

Released: Oct 11, 2022

LeakLess Vulnerability

CVE-2022-38025

- 漏洞又是在 Create Drive Letter 功能中
 - FSCTL 0x601e0 Buffer 傳遞方式

Enter the IOCTL value to decode in the box below

IOCTL VALUE (hex)

That IOCTL decodes to:

Device:

Function:

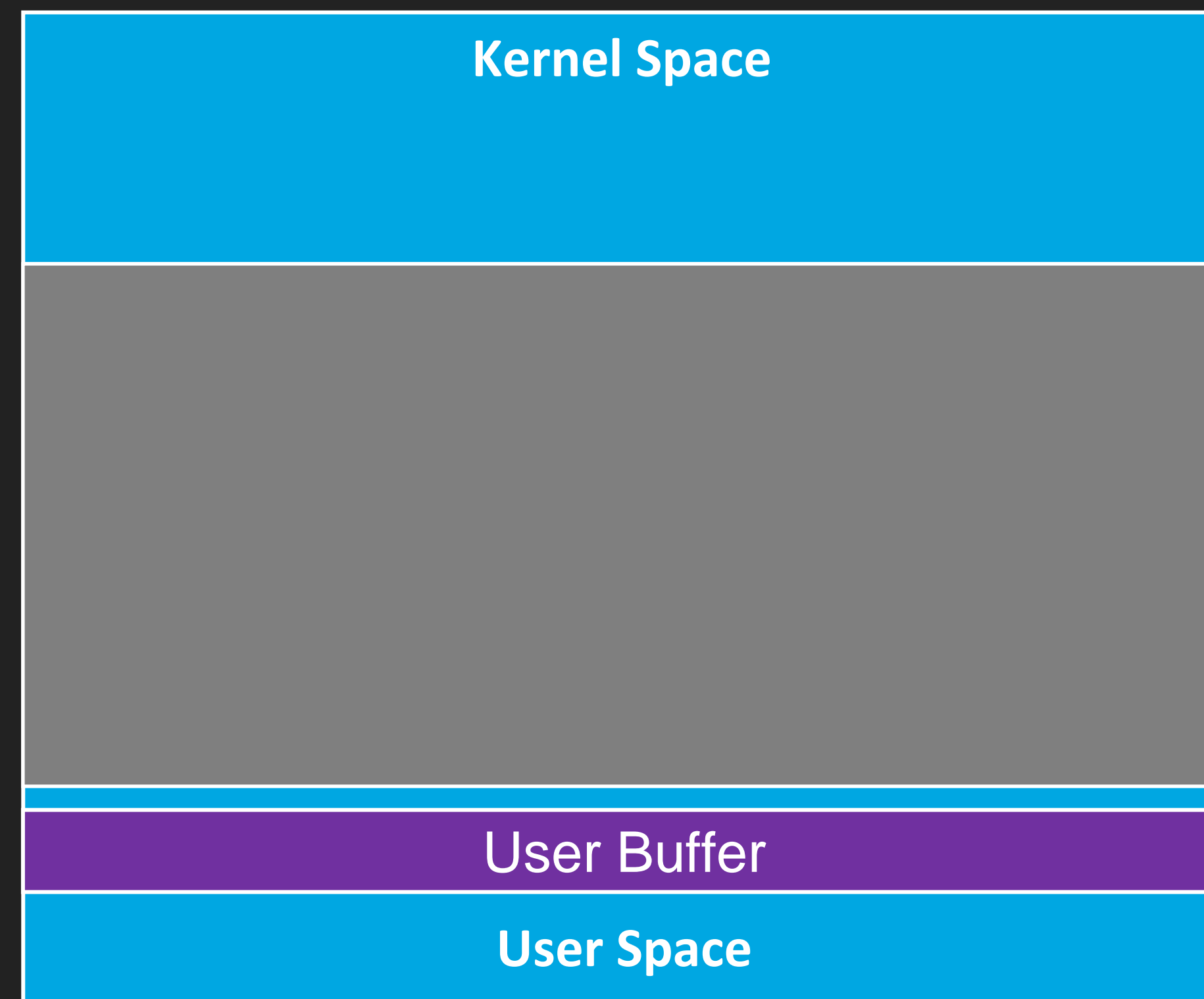
Access:

Method:

LeakLess Vulnerability

CVE-2022-38025

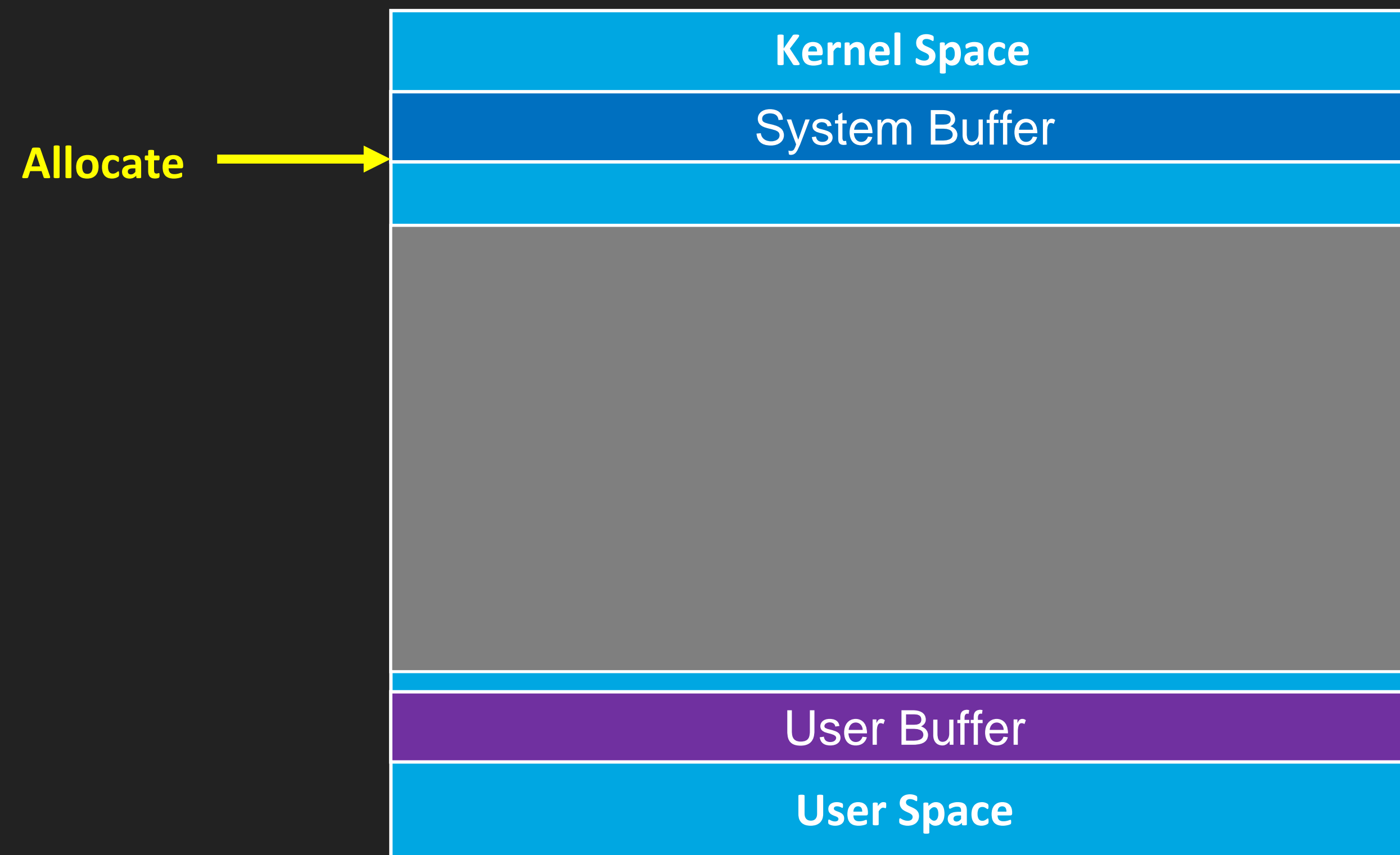
- Buffered I/O



LeakLess Vulnerability

CVE-2022-38025

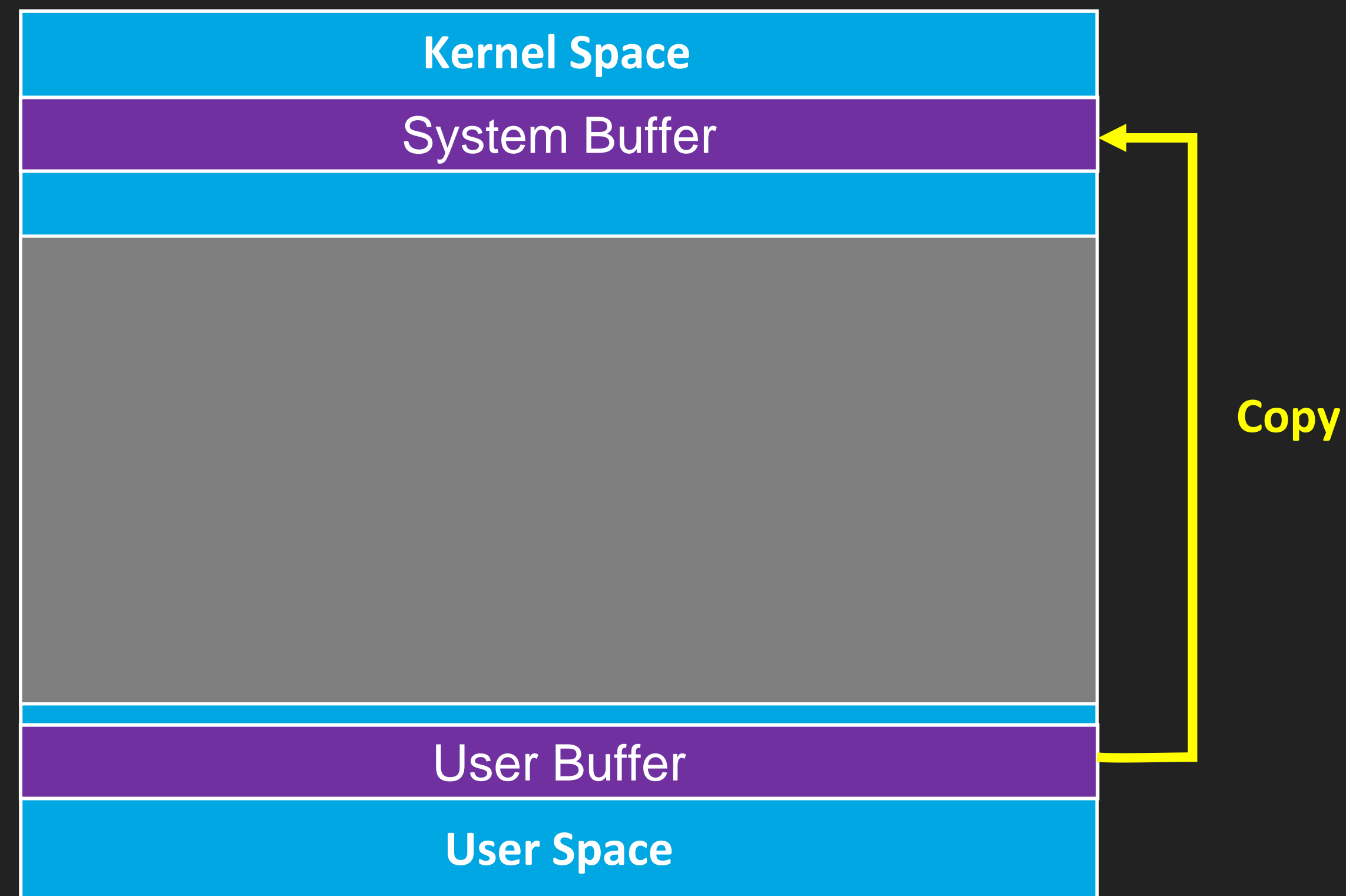
- Buffered I/O



LeakLess Vulnerability

CVE-2022-38025

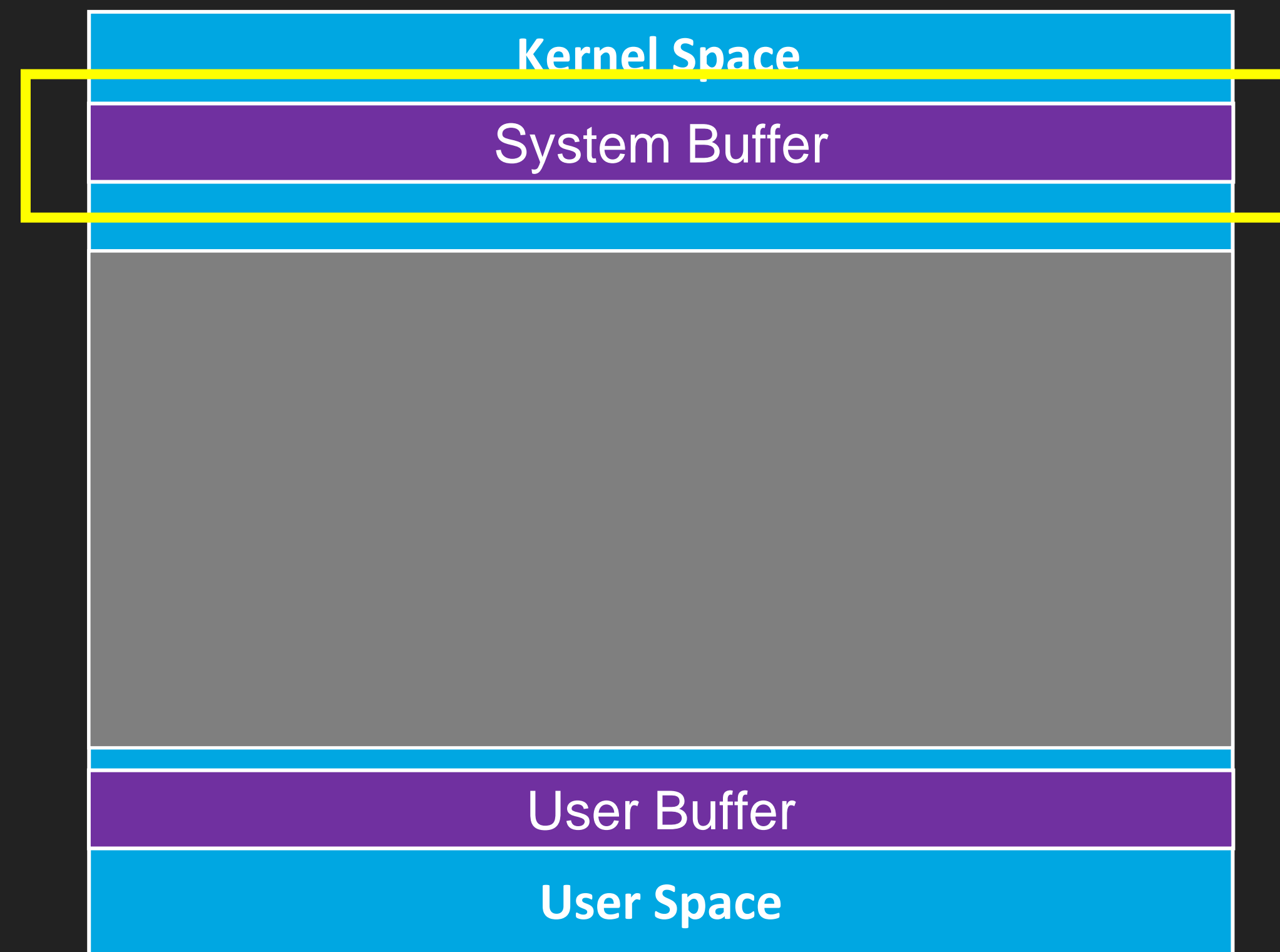
- Buffered I/O



LeakLess Vulnerability

CVE-2022-38025

- Buffered I/O



LeakLess Vulnerability

CVE-2022-38025

- Buffered I/O

```
__kernel_entry NTSYSCALLAPI NTSTATUS NtFsControlFile(  
    [in]          HANDLE          FileHandle,  
    [in, optional] HANDLE        Event,  
    [in, optional] PIO_APC_ROUTINE ApcRoutine,  
    [in, optional] PVOID          ApcContext,  
    [out]         PIO_STATUS_BLOCK IoStatusBlock,  
    [in]          ULONG           FsControlCode,  
    [in, optional] PVOID          InputBuffer,  
    [in]          ULONG           InputBufferLength,  
    [out, optional] PVOID          OutputBuffer,  
    [in]          ULONG           OutputBufferLength  
);
```

Systembuffer = ExAllocatePool(...,InputBufferLength,...)

LeakLess Vulnerability

CVE-2022-38025

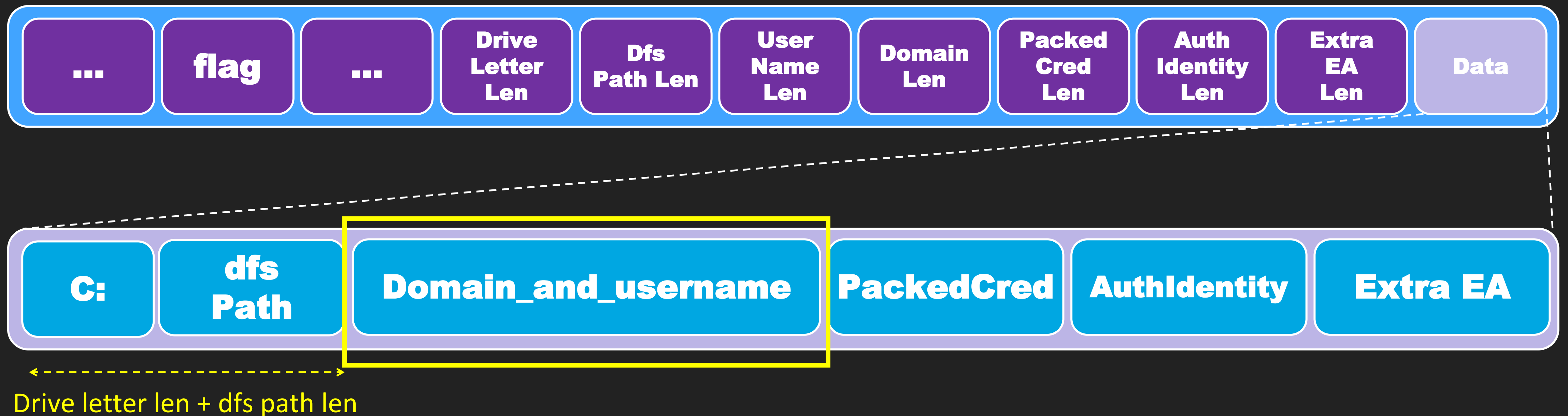
- DfscFsctrlCreateDriveLetter 會根據 Request 給的 Length 來決定要從 buffer 的哪邊來取 data 如 : Username 、 Domain 等

```
driverletter_dfspath_len = driverletter_len + dfs_path_len_1;
unamelen = sysbuf->unamelen;
v21 = &data[driverletterLen];
v60 = v21;
if ( (_WORD)unamelen )
    UserName = &data[(unsigned __int64)driverletter_dfspath_len >> 1];
```

LeakLess Vulnerability

CVE-2022-38025

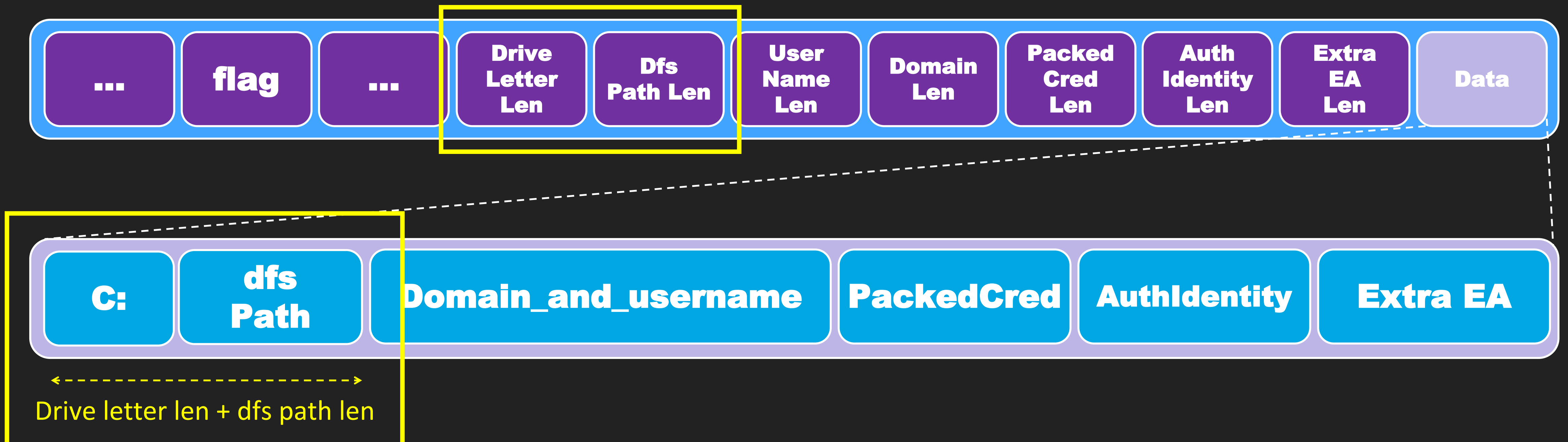
- Username = data[DriverLetterLen + DfsPathLength]



LeakLess Vulnerability

CVE-2022-38025

- Username = data[DriverLetterLen + DfsPathLength]



LeakLess Vulnerability

CVE-2022-38025

- 直覺上會直接看有沒有 Out of bounds ，但實際上對所有的長度都有相對應檢查，乍看下並沒有直接的 Out of bounds

```
DataLen = dfs_path_len
          + driveLetter_len
          + usernamelen
          + PackedCredentialsStringlen
          + sysbuf->extra_ea_len
          + sysbuf->AuthIdentity_len;
if ( DataLen > iputbuflen - 0x16 || (_WORD)domanlen && domanlen + 2 >= usernamelen )
    return (unsigned int)STATUS_INVALID_PARAMETER;
```

LeakLess Vulnerability

CVE-2022-38025

- 直覺上會直接看有沒有 Out of bounds ，但實際上對所有的長度都有相對應檢查，乍看下並沒有直接的 Out of bounds

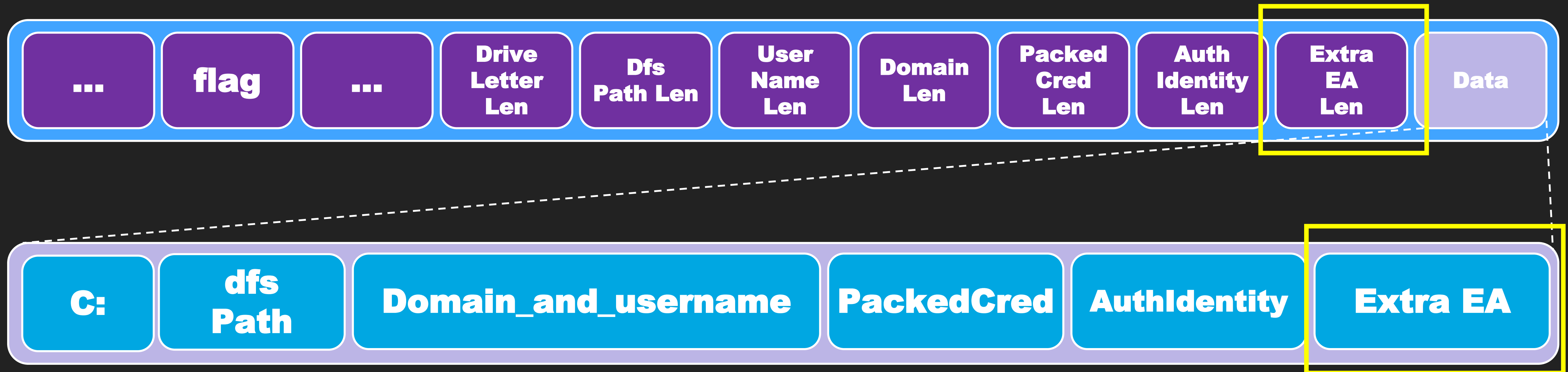
```
DataLen = dfs_path_len
         + driveLetter_len
         + usernamelen
         + PackedCredentialsStringlen
         + sysbuf->extra_ea_len
         + sysbuf->AuthIdentity_len;
if ( DataLen > iputbuflen - 0x16 || (_WORD)domanlen && domanlen + 2 >= usernamelen )
    return (unsigned int)STATUS_INVALID_PARAMETER;
```

Sizeof(struct create req)

LeakLess Vulnerability

CVE-2022-38025

- Extra EA (Extra Extend Attribute)



LeakLess Vulnerability

CVE-2022-38025

- Extra EA (Extra Extend Attribute)
- 對於檔案操作會根據不同 File System 會有不同的處理，可額外給定其它屬性

```
__kernel_entry NTSTATUS NtCreateFile(  
    [out]          PHANDLE          FileHandle,  
    [in]           ACCESS_MASK      DesiredAccess,  
    [in]           POBJECT_ATTRIBUTES ObjectAttributes,  
    [out]          PIO_STATUS_BLOCK IoStatusBlock,  
    [in, optional] PLARGE_INTEGER   AllocationSize,  
    [in]           ULONG             FileAttributes,  
    [in]           ULONG             ShareAccess,  
    [in]           ULONG             CreateDisposition,  
    [in]           ULONG             CreateOptions,  
    [in]           PVOID             EaBuffer,  
    [in]           ULONG             EaLength  
);
```

LeakLess Vulnerability

CVE-2022-38025

- Extra EA (Extra Extend Attribute)
- FILE_FULL_EA_INFORMATION
 - 以 Key & Value 型式儲存

```
typedef struct _FILE_FULL_EA_INFORMATION {
    ULONG    NextEntryOffset;
    UCHAR    Flags;
    UCHAR    EaNameLength;
    USHORT   EaValueLength;
    CHAR     EaName[1];
} FILE_FULL_EA_INFORMATION, *PFILE_FULL_EA_INFORMATION;
```

LeakLess Vulnerability

CVE-2022-38025

- Extra EA (Extra Extend Attribute)
- 可提供而外的 Extend Attribute，後續 `CreateFile` 建立連線時，會將該 `Extend Attribute` 給 `CreateFile`，`dfsc.sys` 會根據 `extra_ea_len` 來取得該結構，不過這邊是給 `extra_ea` 如下這個結構

```
struct extra_ea {  
    ULONG offset;  
    ULONG size;  
    int createoption;  
    int fileattributte;  
};
```

LeakLess Vulnerability

CVE-2022-38025

- Extra EA (Extra Extend Attribute)
- 再根據這個 extra_ea 結構的 offset 來指向，真正 EA 結構

```
struct extra_ea {  
    ULONG offset;  
    ULONG size;  
    int createoption;  
    int fileattributte;  
};
```



LeakLess Vulnerability

CVE-2022-38025

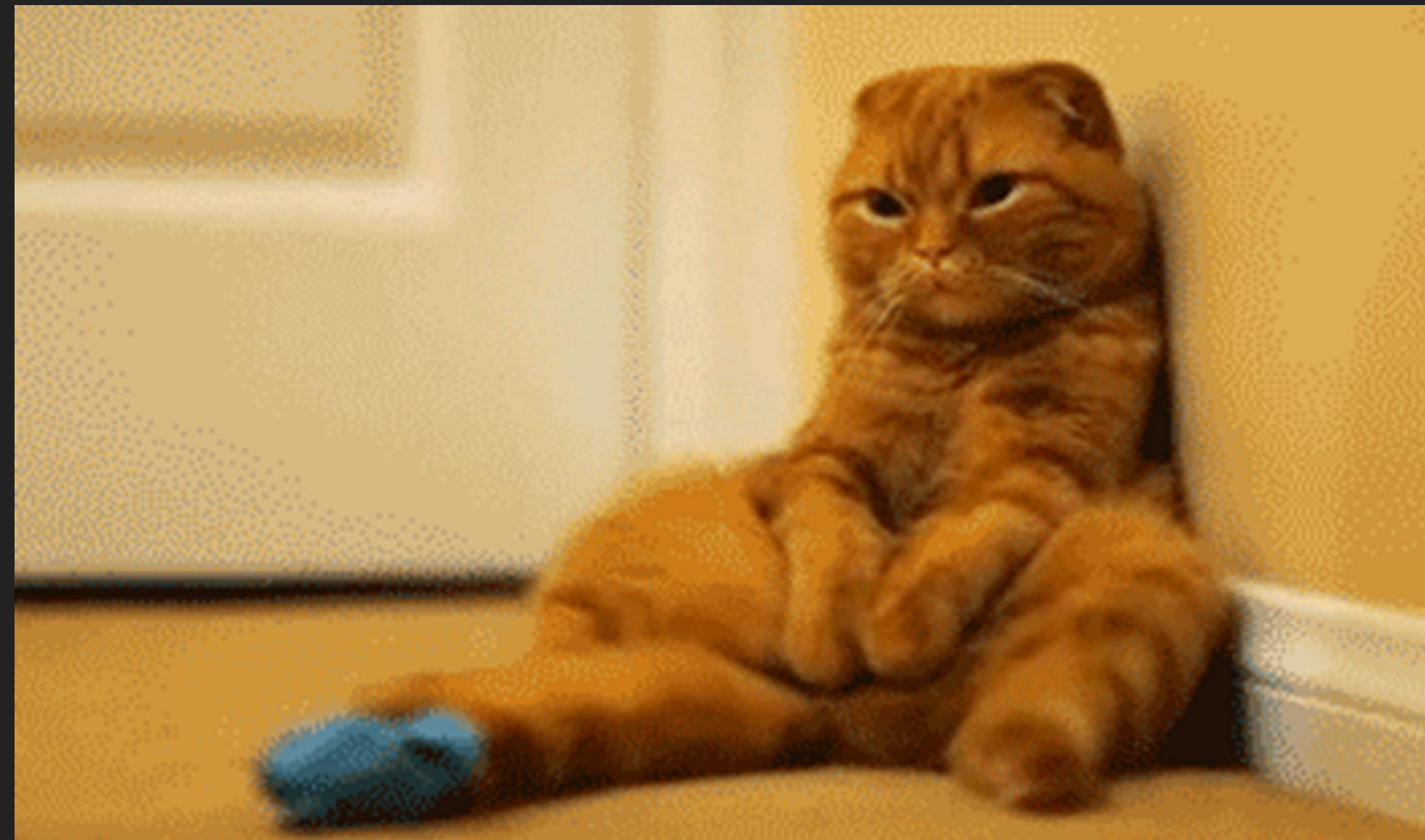
- Extra EA (Extra Extend Attribute)
 - 有好好檢查

```
if ( extra_eaoff > 0x10 && ULongAdd(DataLen, extra_eaoff - 0x10, &DataLen) < 0 )
{
    KeymgrCredentials = STATUS_INTEGER_OVERFLOW;
    v34 = WPP_GLOBAL_Control;
    if...
    v35 = 20i64;
    goto LABEL_56;
}
if ( ULongAdd(DataLen, dfsaddea_ ->size, &DataLen) < 0 )
{
    KeymgrCredentials = STATUS_INTEGER_OVERFLOW;
```

```
if ( DataLen <= maxlen )
{
    LABEL_66:
    v68.Buffer = UserName;
    v68.MaximumLength = unamelen;
```



這時我們一度以為沒洞了



LeakLess Vulnerability

CVE-2022-38025

- Extra EA (Extra Extend Attribute)
 - 來看看怎麼取得 extra_ea 結構
 - 有沒有問題?

```
AuthIdentity_len_25h = AuthIdentity_len + 0x25;
extra_ea_len = sysbuf->extra_ea_len;
extra_ea_offset = ((cred_uname_driverletter_dfspath_len + AuthIdentity_len_25h) & 0xFFFFFFFF0) - 0x16;
if...
if...
dfsaddea_ = (extra_ea *)&data[extra_ea_offset >> 1];
extra_eaoff = dfsaddea_->offset;
```

LeakLess Vulnerability

CVE-2022-38025

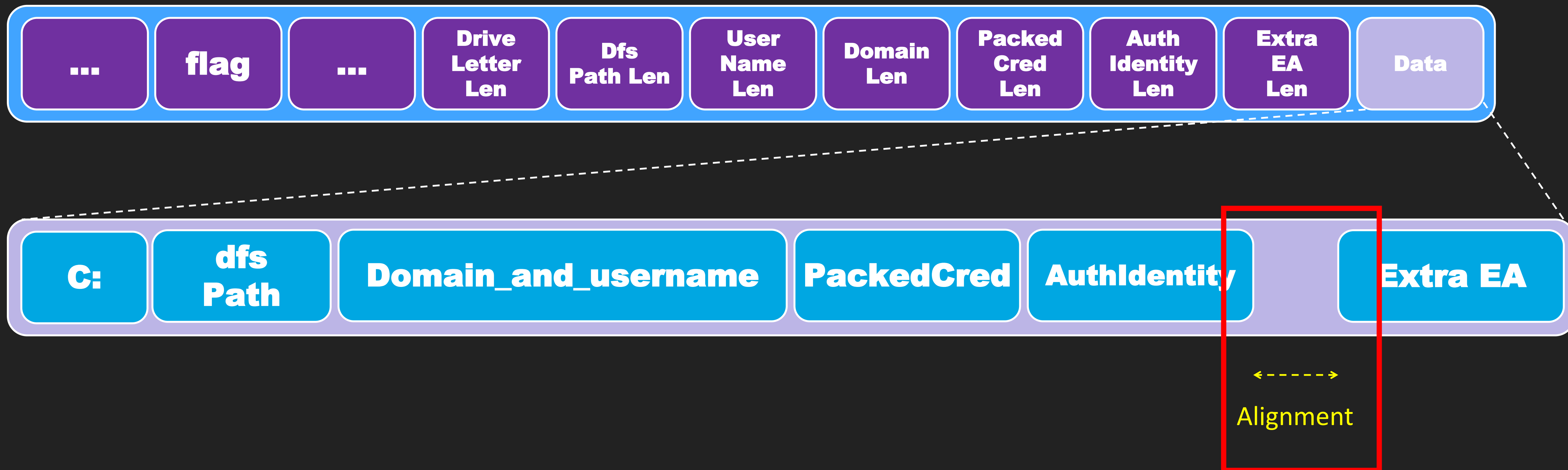
- Extra EA (Extra Extend Attribute)
 - 來看看怎麼取得 extra_ea 結構
 - 有沒有問題?

```
AuthIdentity_len_25h = AuthIdentity_len + 0x25;
extra_ea_len = sysbuf->extra_ea_len;
extra_ea_offset = ((cred_uname_driverletter_dfspath_len + AuthIdentity_len_25h) & 0xFFFFFFFF0) - 0x16;
if...
if...
dfsaddea_ = (extra_ea *)&data[extra_ea_offset >> 1];
extra_eaoff = dfsaddea_->offset;
```

LeakLess Vulnerability

CVE-2022-38025

- Extra EA (Extra Extend Attribute)

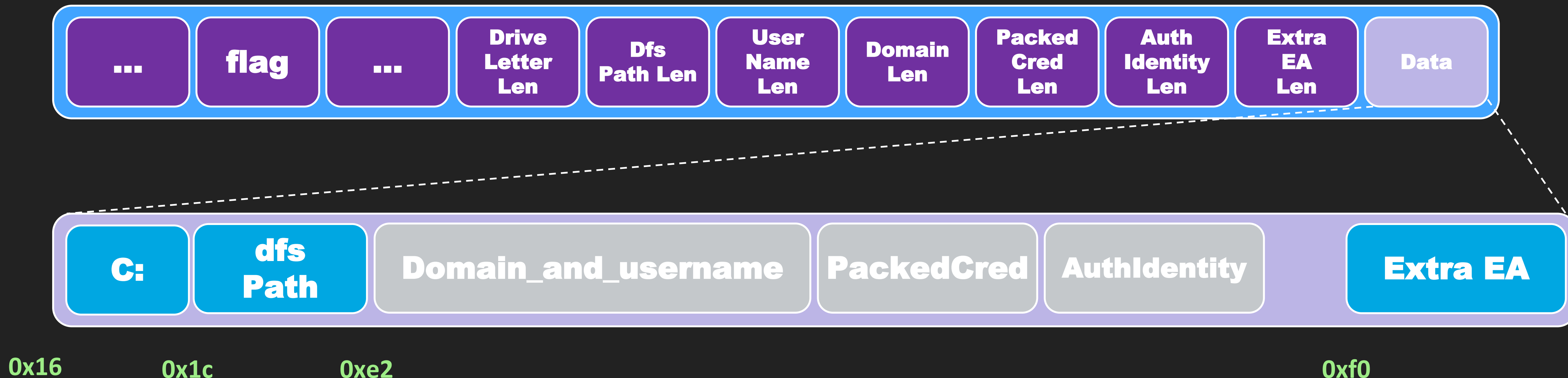


LeakLess Vulnerability

CVE-2022-38025

- Extra EA (Extra Extend Attribute)

Drive letter len = 6
Dfs Path = 0xc6

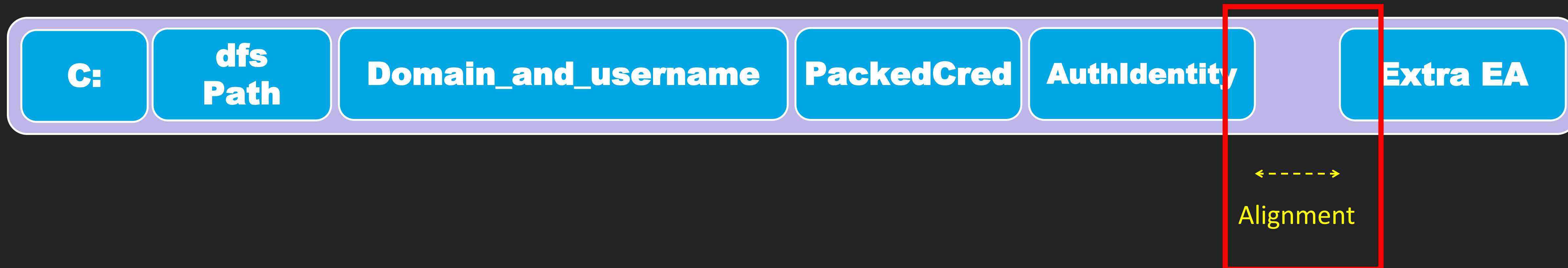


這邊為了 alignment 會多補 0xe bytes

LeakLess Vulnerability

CVE-2022-38025

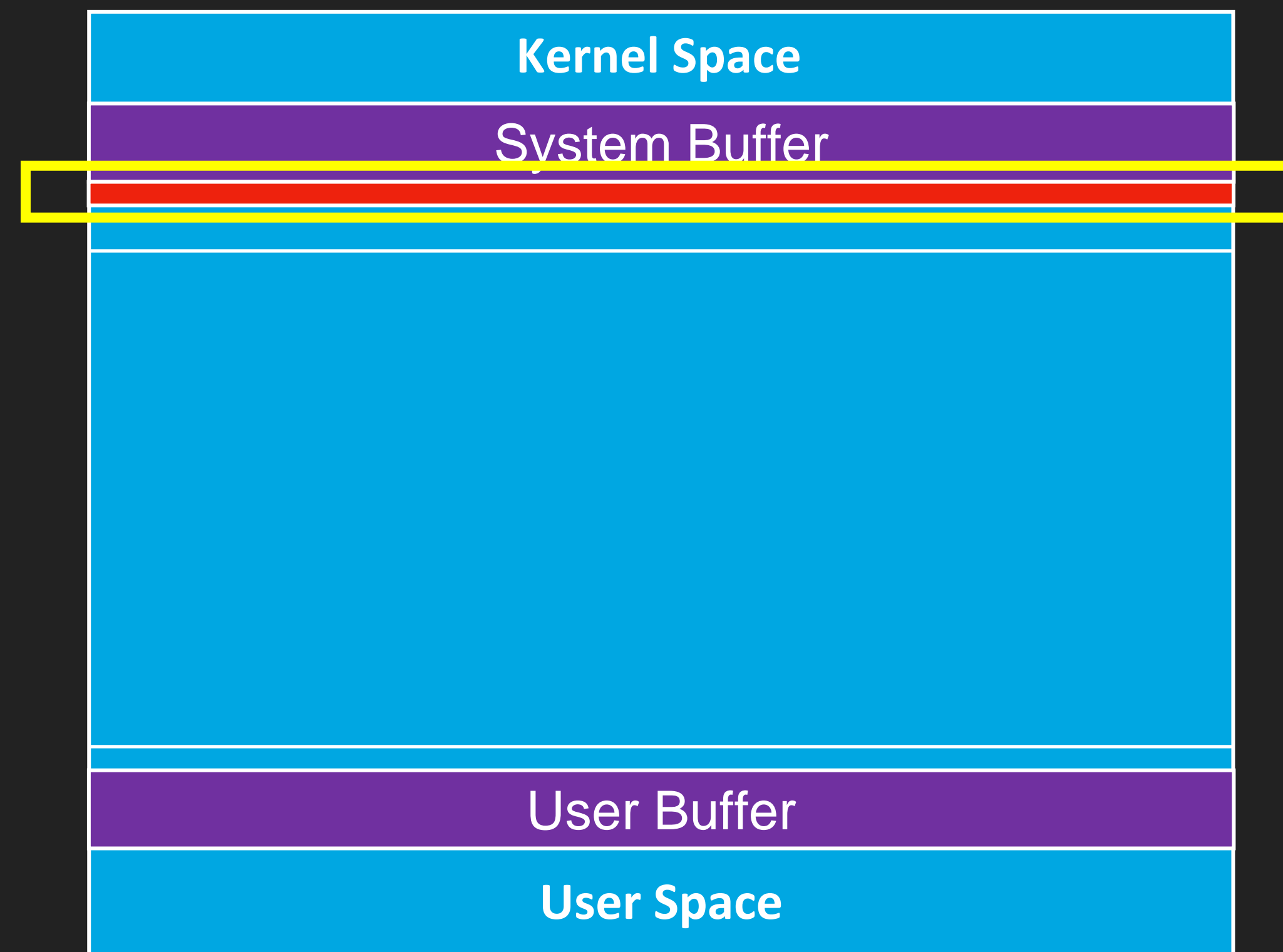
- Extra EA (Extra Extend Attribute)
 - 長度檢查並沒有包含 **alignment** 部分!
 - Out of bounds Read !
 - 最多能越界讀 0xe bytes



LeakLess Vulnerability

CVE-2022-38025

- Buffered I/O



LeakLess Vulnerability

CVE-2022-38025

- Extra EA (Extra Extend Attribute)
 - 長度檢查並沒有包含 **alignment** 部分!
 - Out of bounds Read!
 - 最多能越界讀 0xe bytes
 - 微軟很常忘記算 **alignment** 長度，但也不大容易發現就是了

LeakLess Vulnerability

CVE-2022-38025

- Exploitation ?
 - 只能越界讀 **0xe** bytes

LeakLess Vulnerability

CVE-2022-38025

- Exploitation ?
 - 只能越界讀 **0xe** bytes
 - 後面多數都是 heap chunk header 沒有位置資訊

LeakLess Vulnerability

CVE-2022-38025

- Exploitation ?
 - 只能越界讀 **0xe** bytes
 - 後面多數都是 heap chunk header 沒有位置資訊
 - 廢洞?
 - 近一兩年如果沒有可以證明 kernel pointer leak 很常被 MSRC 算成 DoS



Windows Kernel Heap

Part 1: Segment heap in windows kernel

Angelboy



angelboy@chroot.org



[@scwuaptx](https://twitter.com/scwuaptx)



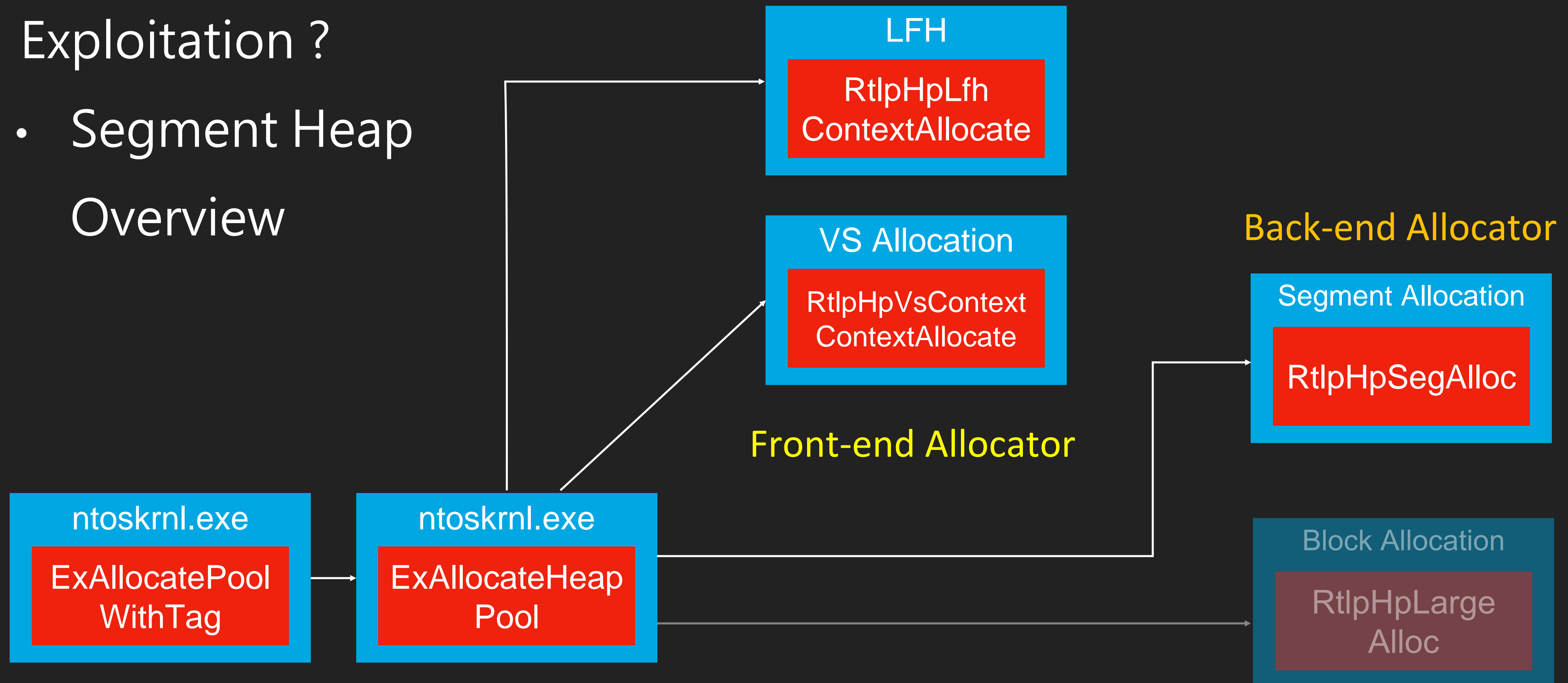
回顧一下 Windows Kernel Segment Heap

DEV✓CORE

LeakLess Vulnerability

CVE-2022-38025

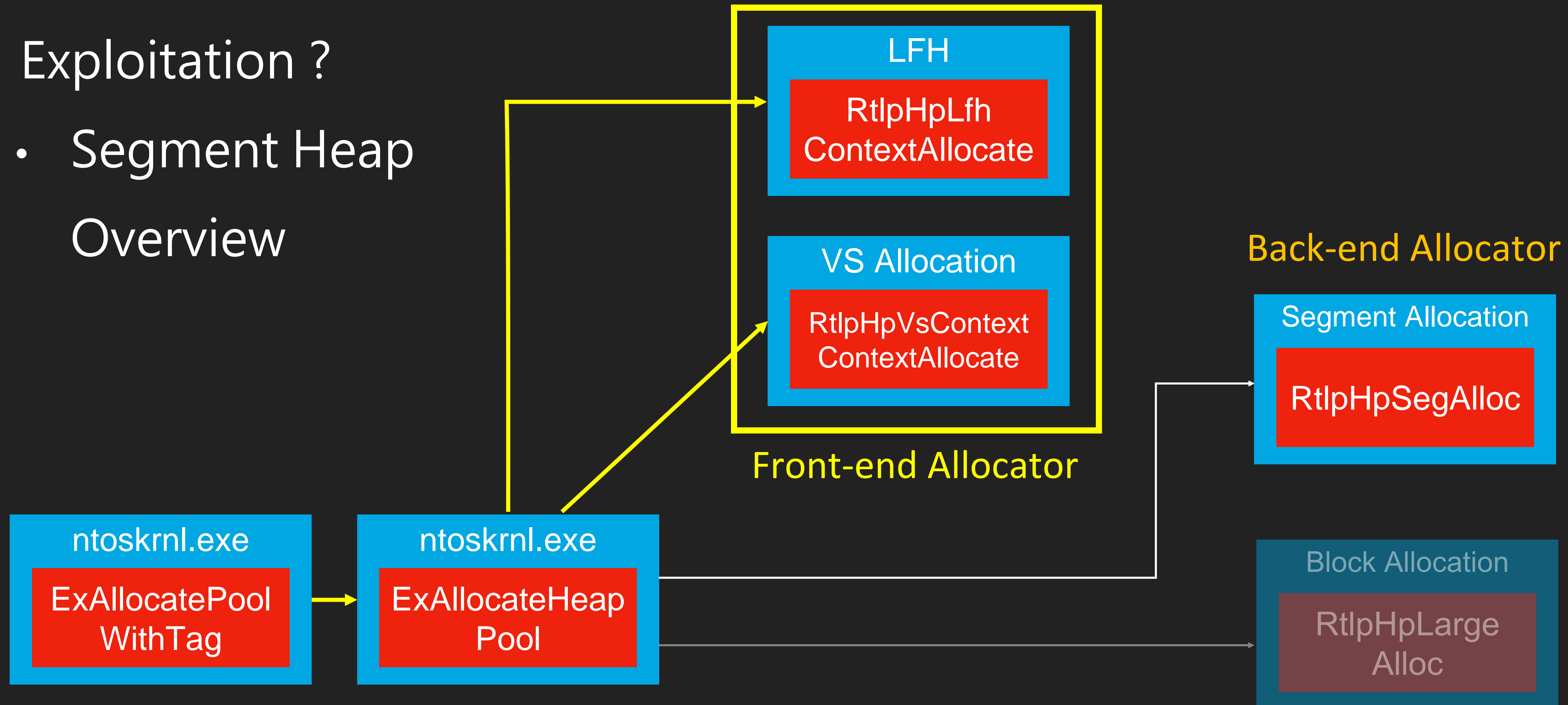
- Exploitation ?
- Segment Heap Overview



LeakLess Vulnerability

CVE-2022-38025

- Exploitation ?
- Segment Heap Overview



LeakLess Vulnerability

CVE-2022-38025

- LFH and Variable Size Allocation Pool Chunk



LeakLess Vulnerability

CVE-2022-38025

- Exploitation ?
 - 兩大 Frontend Allocation 都只能
 - Leak Encode 過的 Header 資訊
 - 只少要有 `RtlpHpHeapGlobals.HeapKey` 才有可能獲得 Kernel Pointer Address

LeakLess Vulnerability

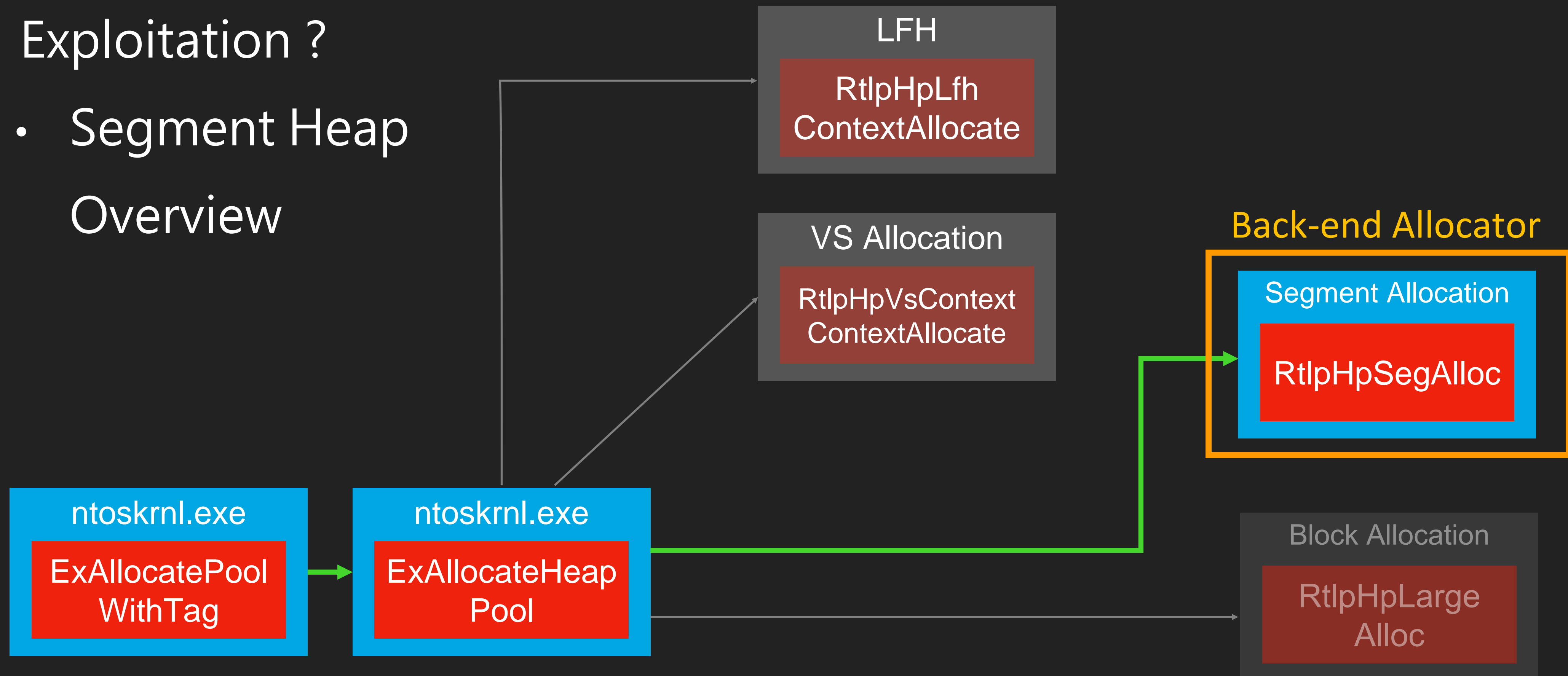
CVE-2022-38025

- Exploitation ?
 - 兩大 Frontend Allocation 都只能
 - Leak Encode 過的 Header 資訊
 - 只少要有 `RtlpHpHeapGlobals.HeapKey` 才有可能獲得 Kernel Pointer Address
 - How about backend Allocation ?

LeakLess Vulnerability

CVE-2022-38025

- Exploitation ?
- Segment Heap Overview



LeakLess Vulnerability

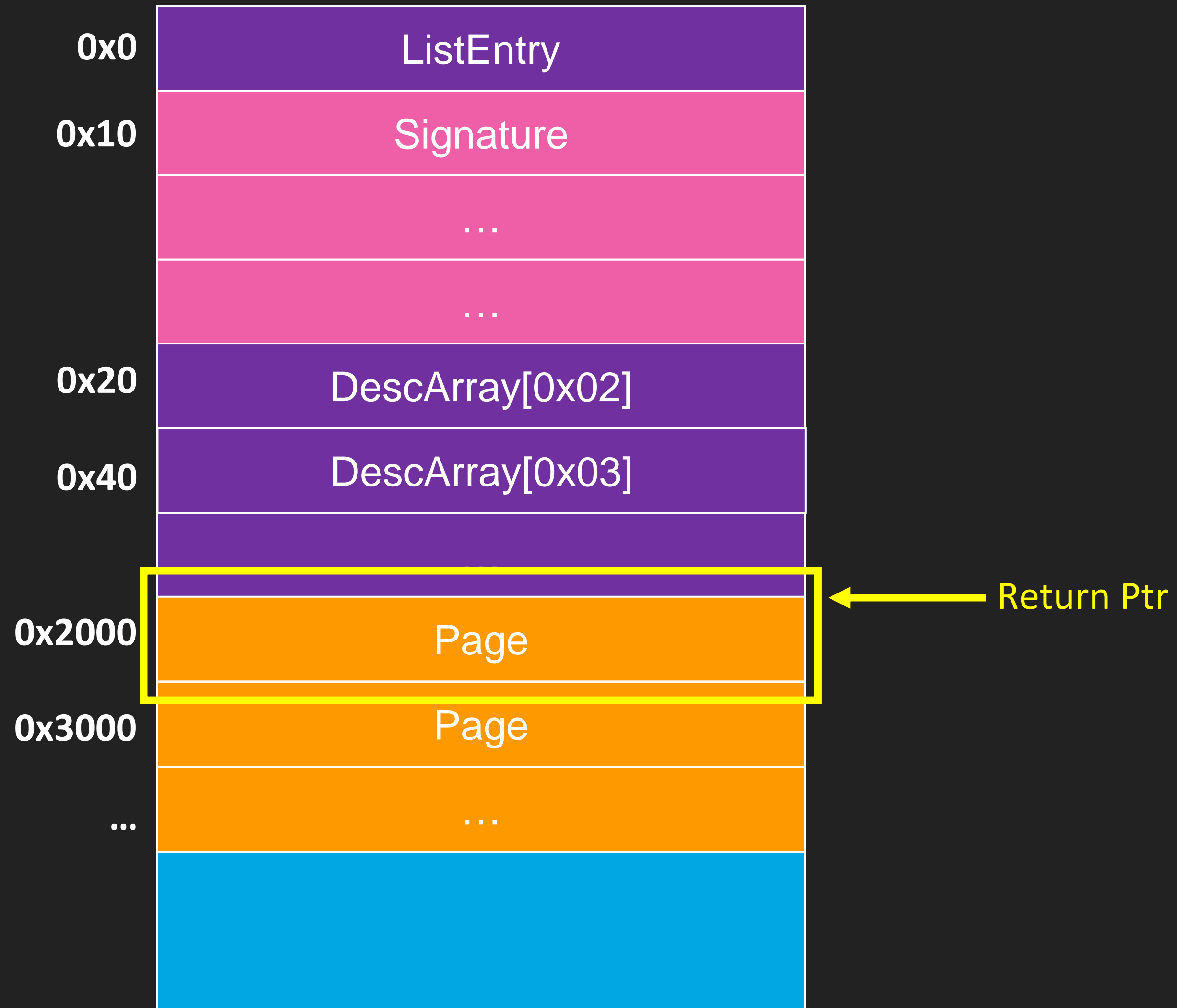
CVE-2022-38025

- Exploitation?
- Segment Allocation
 - $0x20000 < \text{Size} < \text{SegContexts}[1].\text{MaxAllocationSize} (0x7f0000)$
 - $\text{Size} \& 0\text{fff} == 0$

LeakLess Vulnerability

CVE-2022-38025

- Exploitation?



LeakLess Vulnerability

CVE-2022-38025

- Exploitation?
- Segment Allocation
 - Page
 - No any meta data !

LeakLess Vulnerability

CVE-2022-38025

- Exploitation?
- Segment Allocation
 - Page
 - No any meta data !
 - 可以放一些 Kernel Object 在 input buffer 後面，就有機會 leak kernel 資訊

LeakLess Vulnerability

CVE-2022-38025

- Exploitation?
- There are many good object for it.
 - Named Pipe
 - LFH Subsegment
 - ...

DECEMBER 29, 2014 BY IONESCU007

Sheep Year Kernel Heap Fengshui: Spraying
in the Big Kids' Pool

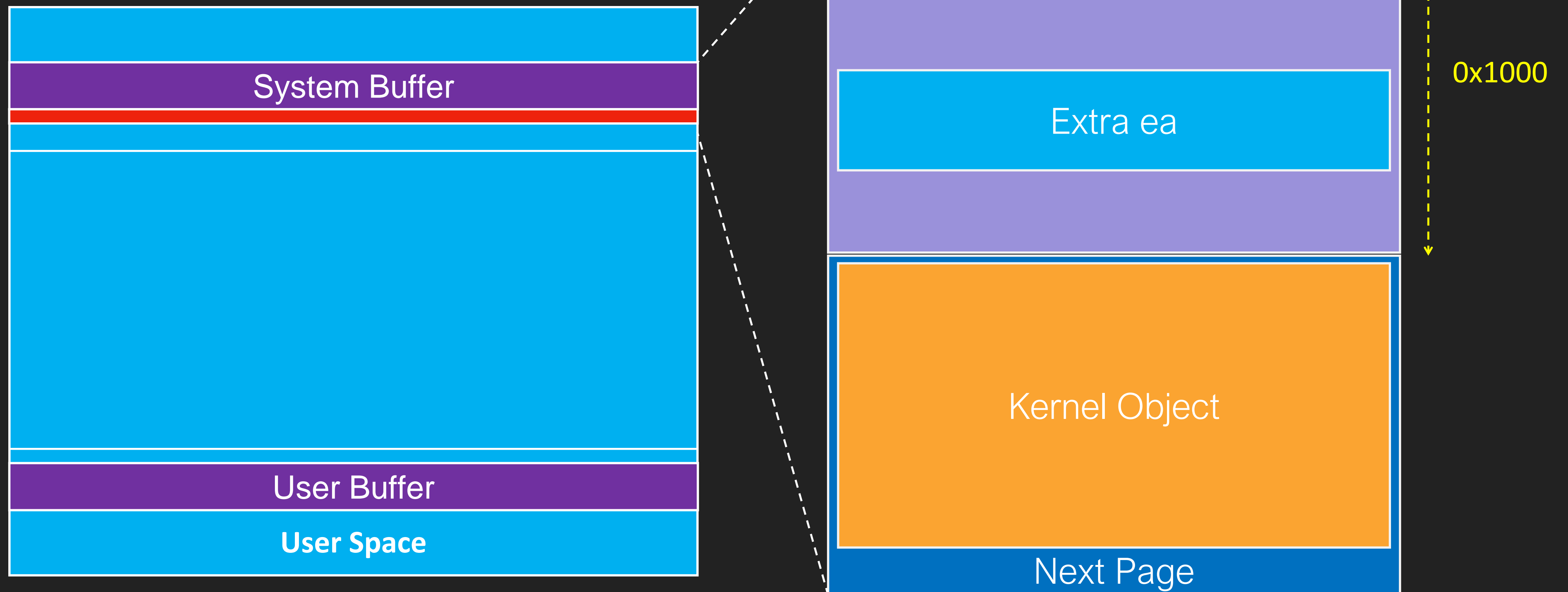
The State of Kernel Exploitation

<https://www.alex-ionescu.com/kernel-heap-spraying-like-its-2015-swimming-in-the-big-kids-pool/>

LeakLess Vulnerability

CVE-2022-38025

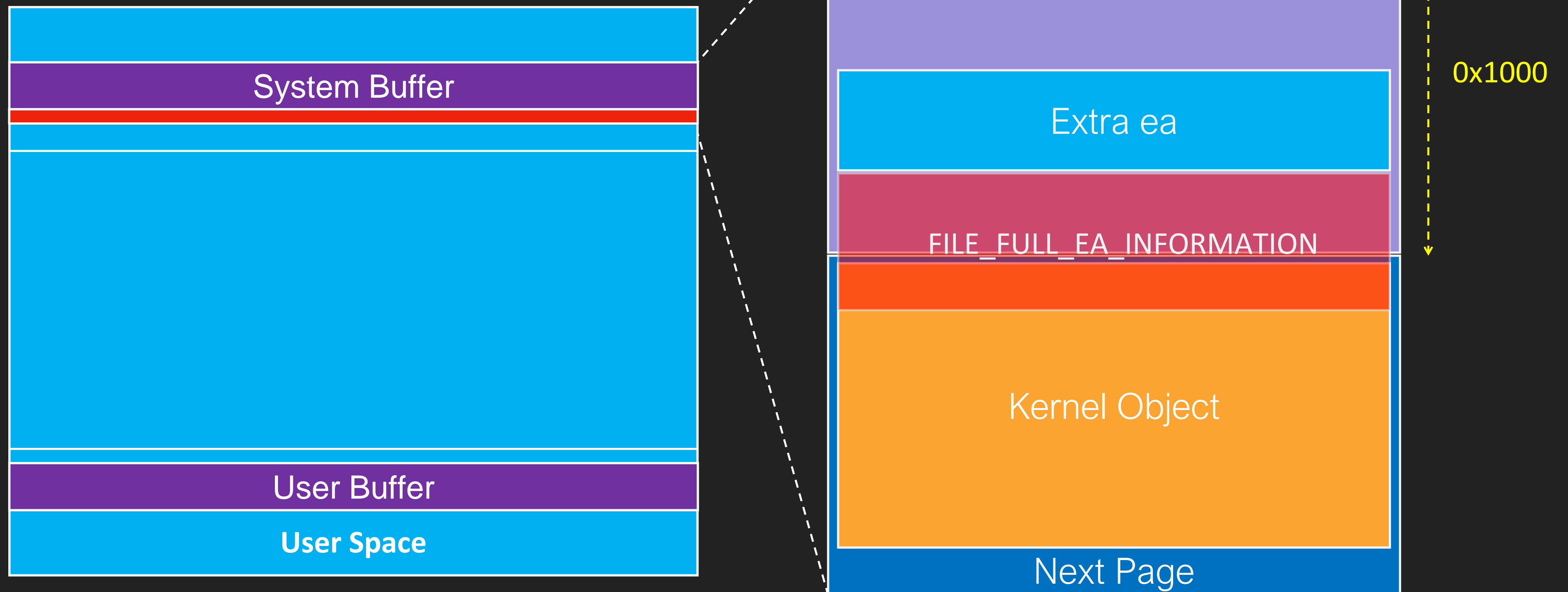
- Exploitation?



LeakLess Vulnerability

CVE-2022-38025

- Exploitation?



LeakLess Vulnerability

CVE-2022-38025

- Exploitation?
 - 可以越界讀把資料讀到 Extend Attribute 裡面了
 - 那麼要怎麼讀出來?



LeakLess Vulnerability

CVE-2022-38025

- Exploitation
 - Extra EA (Extra Extend Attribute) in DfscFsctrlCreateDriveLetter
 - 可提供而外的 Extend Attribute，後續 **CreateFile** 建立 **SMB** 連線時，會將該 Extend Attribute 給 CreateFile

LeakLess Vulnerability

CVE-2022-38025

- Exploitation
 - AuthIdentity Extend Attribute
 - 主要儲存認證資訊
 - 當 CreateFile 用 smb 向 Remote server 溝通，此時如有給 AuthIdentity Extend Attribute 就會用 AuthIdentity 中的 Cred 來向 Remote Server 認證

LeakLess Vulnerability

CVE-2022-38025

- Exploitation
 - **AuthIdentity** Extend Attribute structure

```
Offset Size struct authstruct
      {
0000 0004  ULONG version;
0004 0004  ULONG xxxoffset;
0008 0004  ULONG useroffset;
000C 0004  ULONG usernamelen;
0010 0004  ULONG domainoffset;
0014 0004  ULONG domainlen;
0018 0004  ULONG packoffset;
001C 0004  ULONG packlen;
0020 0004  ULONG flag;
0024 0004  ULONG PackageListOffset;
0028 0004  ULONG PackageListLen;
002C 0001  char data[1];
      0030 };
```

LeakLess Vulnerability

CVE-2022-38025

- Exploitation
 - AuthIdentity Extend Attribute structure
 - 這邊只要把 User offset 或 Domain offset 指向

越界讀的 offset 就可

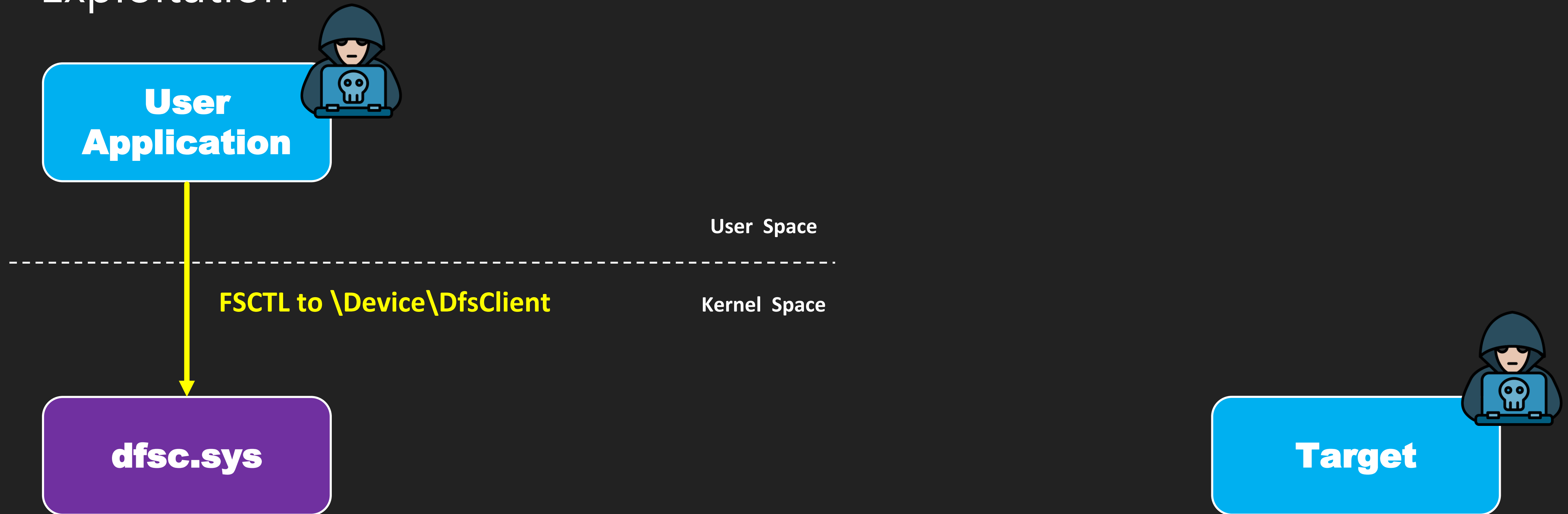
- 長度不可超過越界讀的大小

Offset	Size	struct authstruct
		{
0000	0004	ULONG version;
0004	0004	ULONG xxxoffset;
0008	0004	ULONG useroffset;
000C	0004	ULONG usernamelen;
0010	0004	ULONG domainoffset;
0014	0004	ULONG domainlen;
0018	0004	ULONG packoffset;
001C	0004	ULONG packlen;
0020	0004	ULONG flag;
0024	0004	ULONG PackageListOffset;
0028	0004	ULONG PackageListLen;
002C	0001	char data[1];
	0030	};

LeakLess Vulnerability

CVE-2022-38025

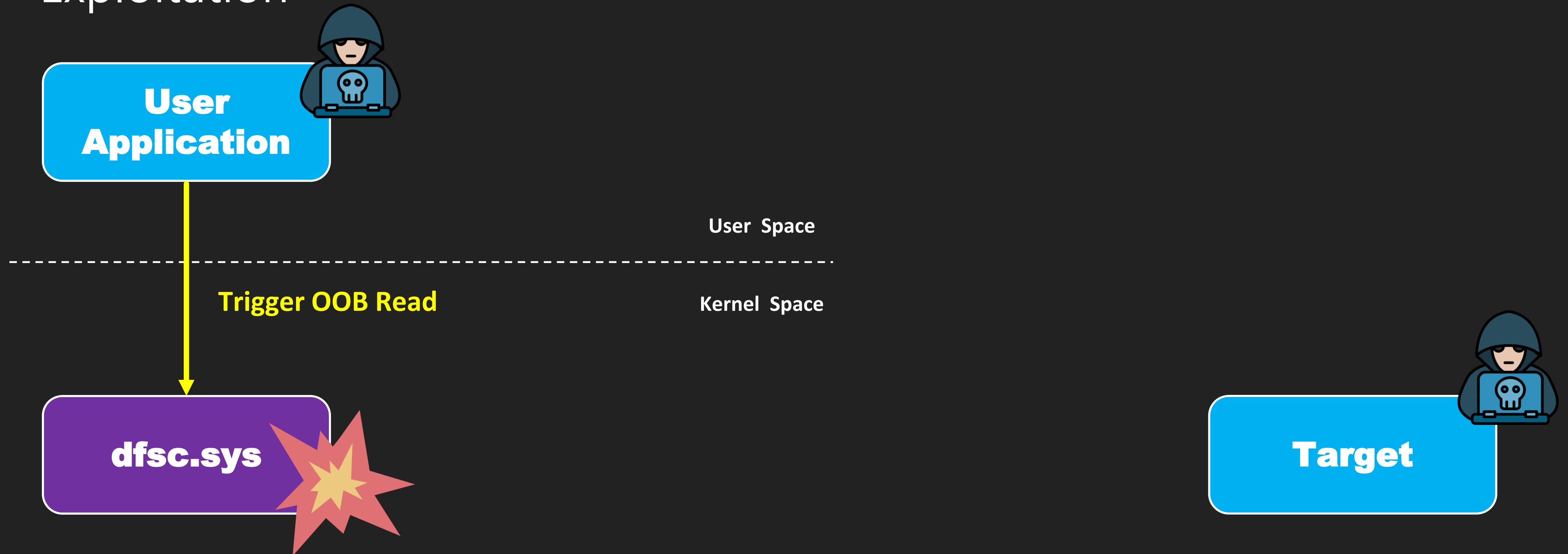
- Exploitation



LeakLess Vulnerability

CVE-2022-38025

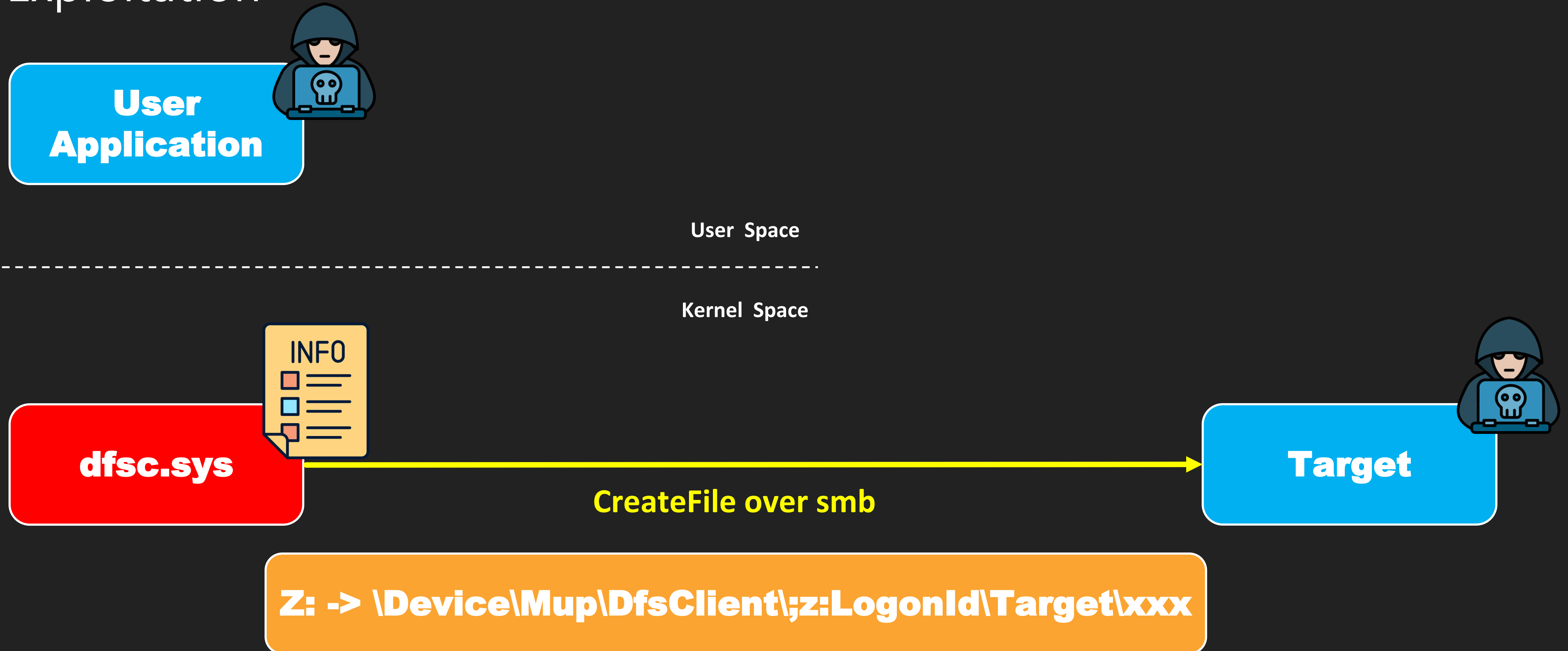
- Exploitation



LeakLess Vulnerability

CVE-2022-38025

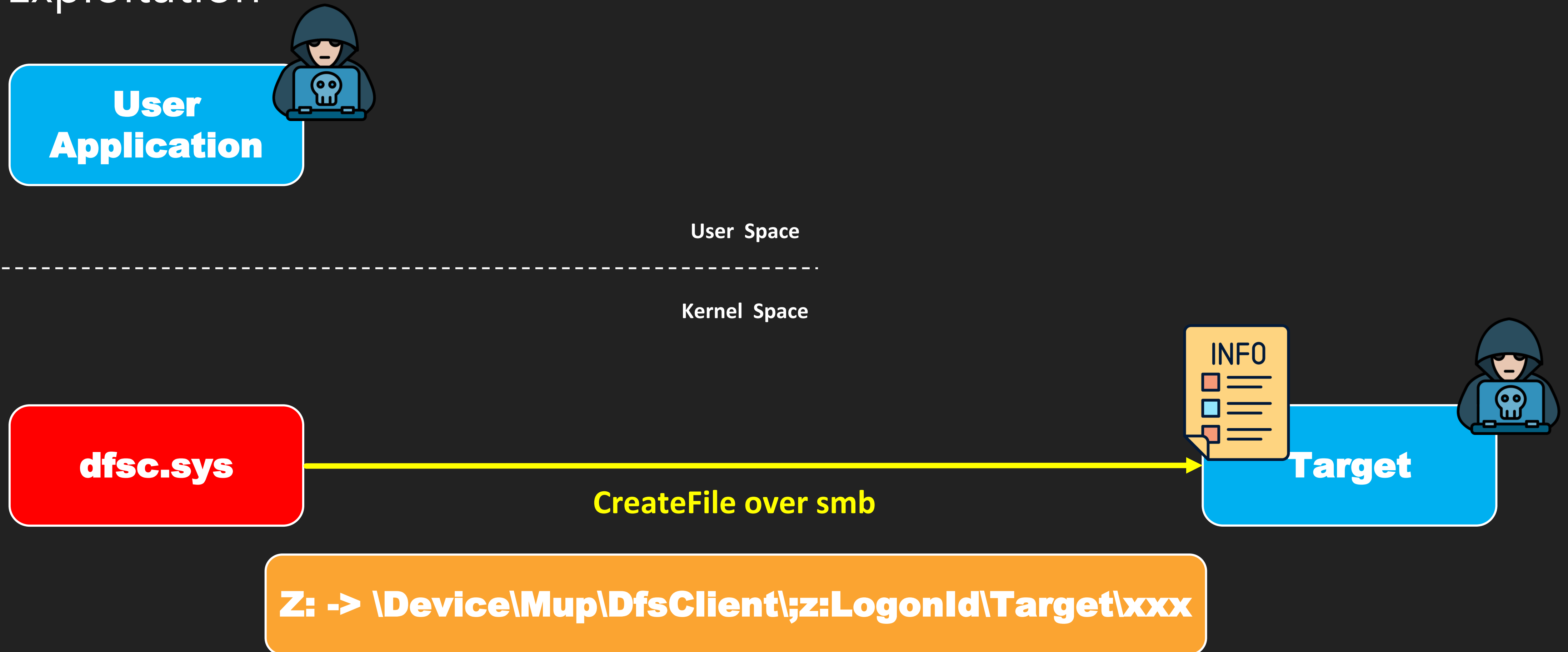
- Exploitation



LeakLess Vulnerability

CVE-2022-38025

- Exploitation



LeakLess Vulnerability

CVE-2022-38025

- Exploitation

The image shows a Wireshark network traffic capture for a connection to tcp.port == 445. The packet list pane shows the following traffic:

No.	Time	Source	Destination	Protocol	Length	Info
30	37.678572	172.28.73.222	172.28.66.231	SMB2	342	Negotiate Protocol Request
31	37.679993	172.28.66.231	172.28.73.222	SMB2	590	Negotiate Protocol Response
32	37.722022	172.28.73.222	172.28.66.231	TCP	60	50274 → 445 [ACK] Seq=362 Ack=989 Win=261632 Len=0
33	37.775481	172.28.73.222	172.28.66.231	SMB2	220	Session Setup Request, NTLMSSP_NEGOTIATE
34	37.776595	172.28.66.231	172.28.73.222	SMB2	401	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CH
35	37.781865	172.28.73.222	172.28.66.231	SMB2	693	Session Setup Request, NTLMSSP_AUTH, User: aaaaaaa\退詔蒞懐葑蒞
36	37.783138	172.28.66.231	172.28.73.222	SMB2	130	Session Setup Response, Error: STATUS_LOGON_FAILURE
37	37.784659	172.28.73.222	172.28.66.231	TCP	60	50274 → 445 [RST, ACK] Seq=1167 Ack=1412 Win=0 Len=0

The details pane shows the structure of the NTLMSSP_AUTH message (packet 35):

- NTLMSSP identifier: NTLMSSP
- NTLM Message Type: NTLMSSP_AUTH (0x00000003)
- Lan Manager Response: 00
- LMv2 Client Challenge: 0000000000000000
- NTLM Response: 012cefc6c563439a2c3e1b8d3385520c0101000000000007e409e930576d80134904ed4...
- Domain name: aaaaaaa
- User name: 退詔蒞懐葑蒞
- Host name: DESKTOP-EVLDFRK
- Session Key: 2b0b4b1c10b2db02c6b0fcb50505732

A red starburst highlights the Session Key field. Below the details pane, the raw packet bytes are shown in hexadecimal and ASCII. A red box highlights the following bytes: 00 90 1f ae 0f 85 ff ff 00 60 fd a8 0f 85. The ASCII representation shows '.....D.E.S'.

~~LeakLess?~~ Leak Vulnerability in DFSC



orange 🍊 1 year ago

排 heap 還是有用的(跑

DEV✓*CORE*

Summary

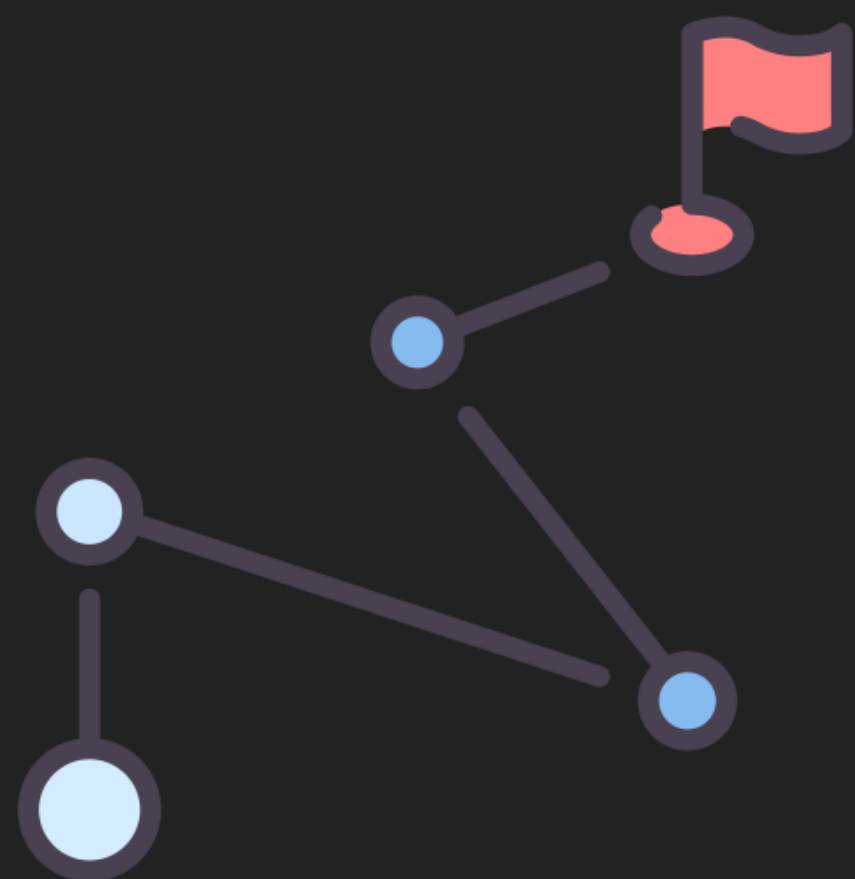
- 在做 Alignment 時也要注意 buffer 大小計算

Summary

- 在做 Alignment 時也要注意 buffer 大小計算
- 任何微小的漏洞都可能造成危害

- 在做 Alignment 時也要注意 buffer 大小計算
- 任何微小的漏洞都可能造成危害
- 也許可以繼續研究的方向
 - [mup.sys](#)
 - [rdbss.sys](#)

當找不到洞時，不妨休息一下
也許哪天回頭看
你可能會有不同的新發現



DEV/CORE

DEV✓CORE

Thanks

戴夫寇爾股份有限公司
angelboy@devco.re

DEV✓CORE

Thanks

戴夫寇爾股份有限公司
angelboy@devco.re