

❄️ 看雪 · 第六届安全开发者峰会

猫鼠游戏 如何进行 Windows 平台在野 0day 狩猎

金权 安恒信息

```
#include <stdio.h>
int main()
{
    printf("Hello,World!");
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    printf("Hello,W
    return 0;
}
```

2022 SDC

自我介绍

➤ 金权 (@jq0904/银雁冰)

- 安恒信息猎影/卫兵实验室高级安全专家
- 研究方向
 - Windows平台漏洞挖掘与利用
 - Windows平台在野0day狩猎
- 演讲经历
 - Black Hat USA 2022
 - HITB Amsterdam 2021
 - Blue Hat Shanghai 2019
- 40+ Microsoft/Adobe CVE致谢 (7次在野0day致谢)
- 2020~2022 MSRC TOP 100



目录

1. 背景
2. 用不同的方法解决不同的问题
 - 为什么沙箱适合狩猎Office在野0day
 - 为什么YARA适合狩猎Windows本地提权在野0day
3. 近两年热门在野0day案例分析
 - Office在野0day案例分析
 - Windows本地提权在野0day案例分析
4. 如何培养在野0day狩猎人才

目录

1. 背景

2. 用不同的方法解决不同的问题

- 为什么沙箱适合狩猎Office在野0day
- 为什么YARA适合狩猎Windows本地提权在野0day

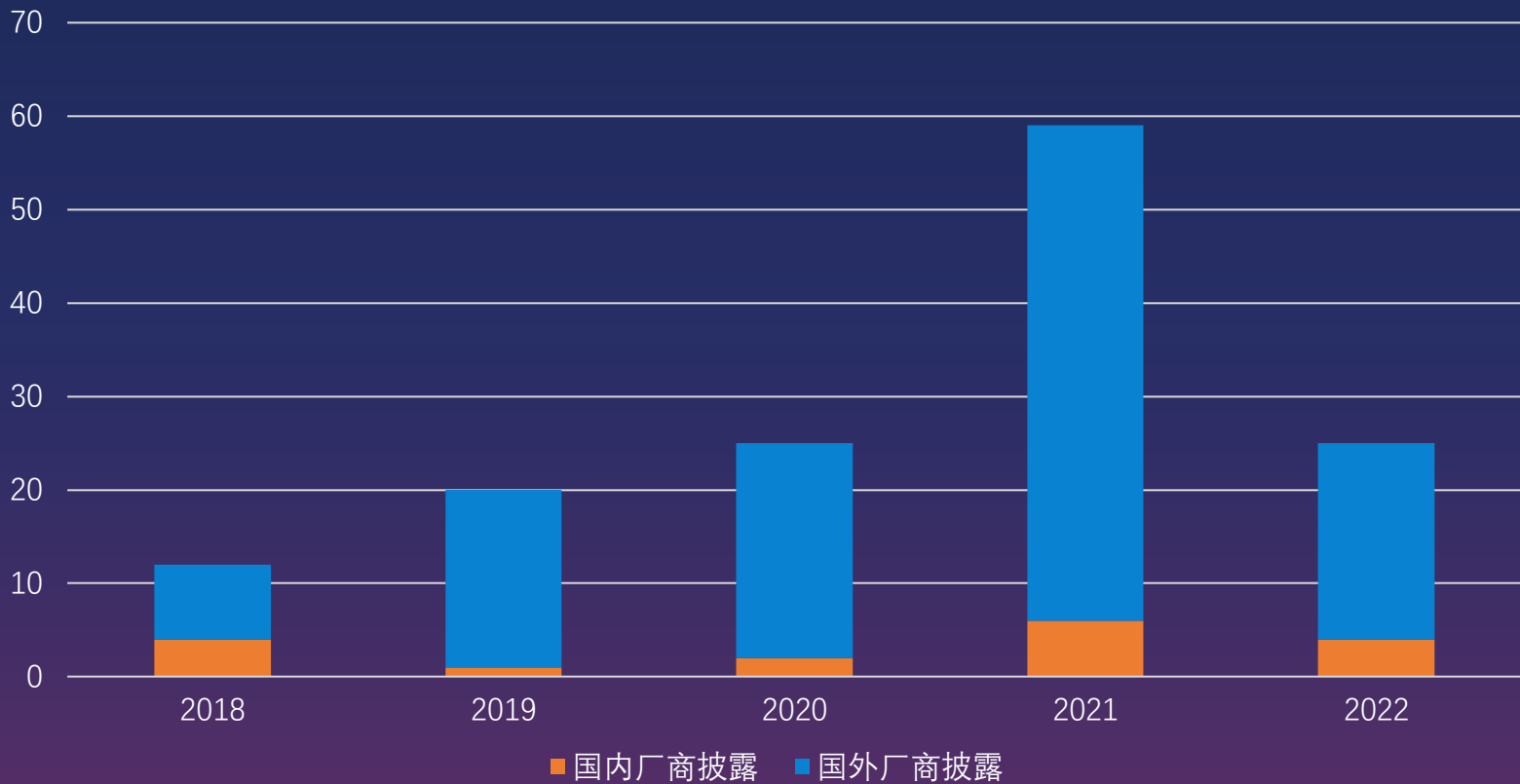
3. 近两年热门在野0day案例分析

- Office在野0day案例分析
- Windows本地提权在野0day案例分析

4. 如何培养在野0day狩猎人才

背景

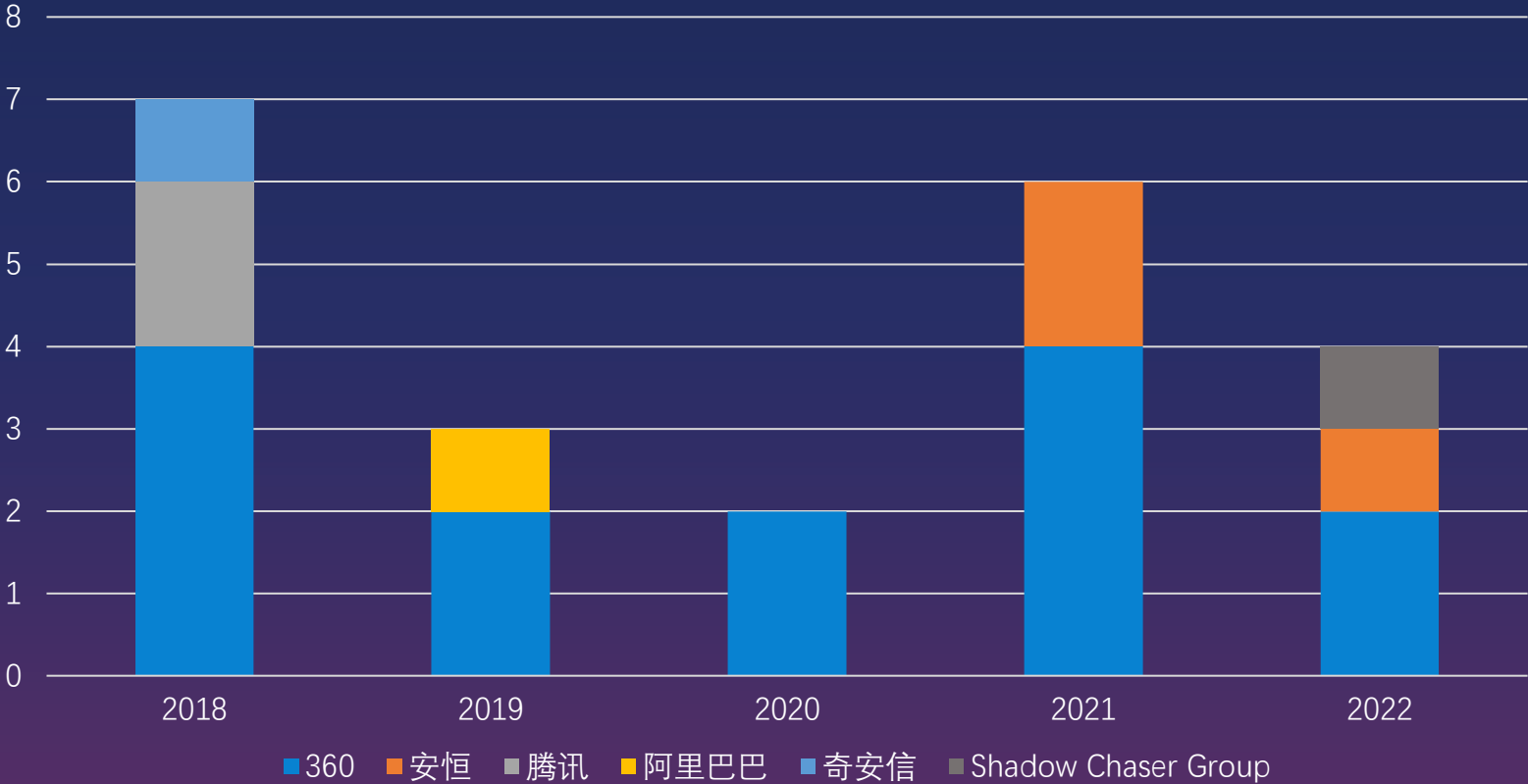
最近5年全球在野0day披露数量及国内外占比



主要统计数据来源: 0day "[In the Wild](#)" - Project Zero

背景

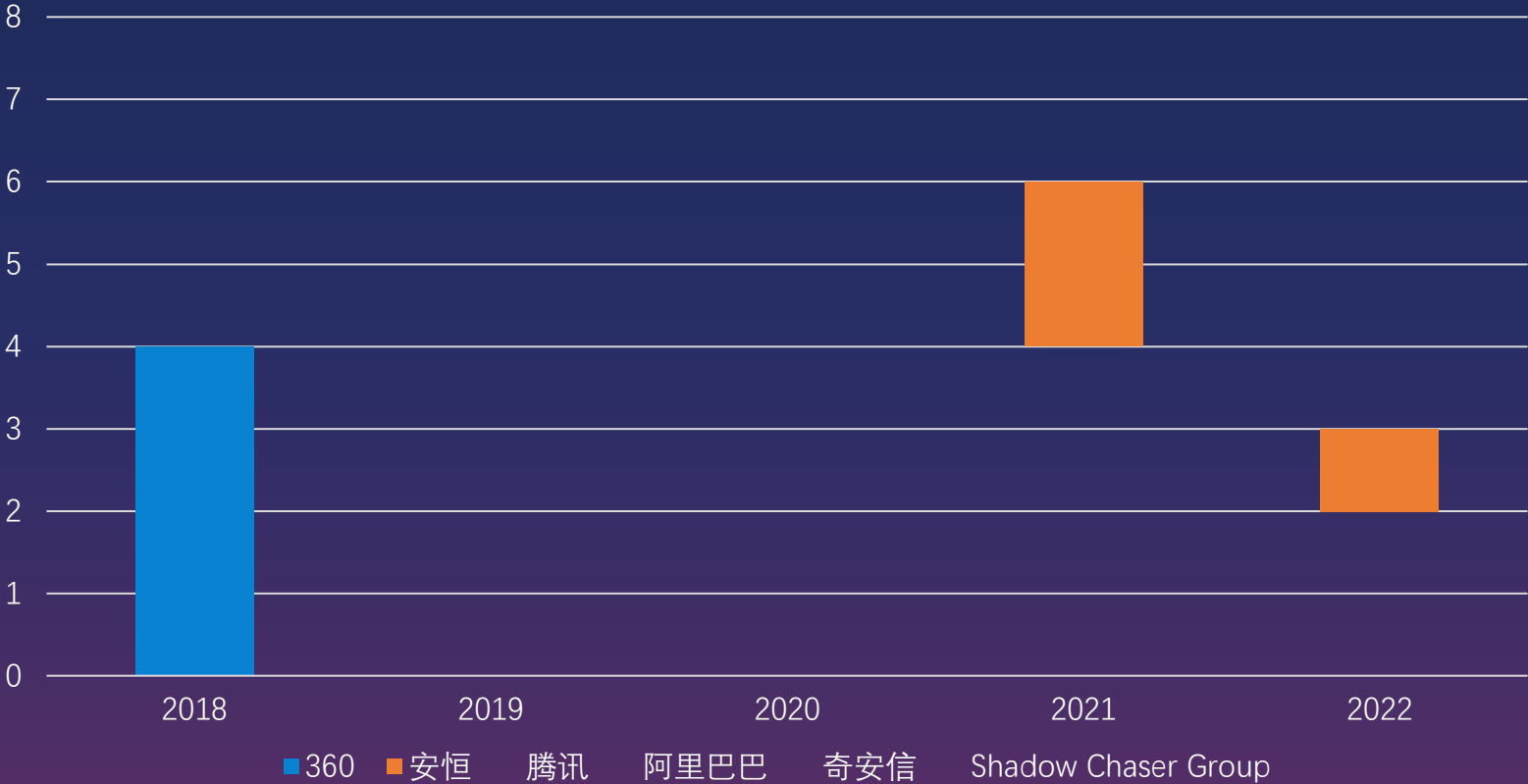
最近5年中国厂商披露在野0day统计



主要统计数据来源: [Oday "In the Wild" - Project Zero](#)

背景

我参与披露的在野0day



主要统计数据来源: [0day "In the Wild" - Project Zero](#)

目录

1. 背景
2. 用不同的方法解决不同的问题
 - 为什么沙箱适合狩猎Office在野0day
 - 为什么YARA适合狩猎Windows本地提权在野0day
3. 近两年热门在野0day案例分析
 - Office在野0day案例分析
 - Windows本地提权在野0day案例分析
4. 如何培养在野0day狩猎人才

Office在野0day漏洞分类

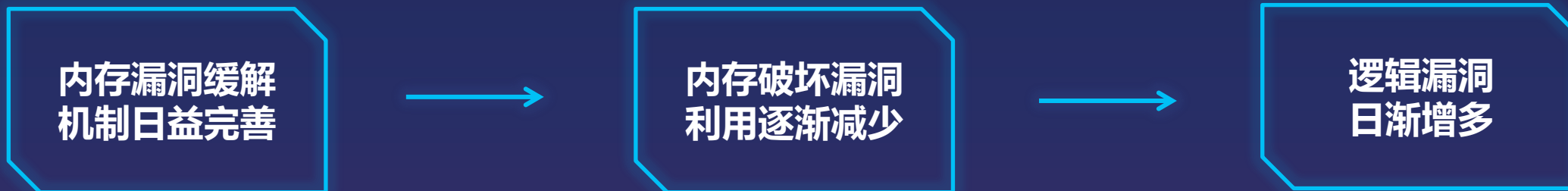
➤ 纯Office漏洞

- **内存破坏漏洞**：例如CVE-2015-1641，近几年此类漏洞攻击逐渐消失
- **逻辑漏洞**：沙虫漏洞（CVE-2014-4114），Moniker系列漏洞（CVE-2017-0199等）

➤ 以Office为载体的其他漏洞

- **Flash漏洞**：随着Flash退役而消失
- **IE浏览器漏洞**：随着IE退役逐渐减少，但攻击面仍在
- **.NET漏洞**：CVE-2018-8759，个例
- **Windows系统漏洞**：CVE-2014-4148等，不排除后面还会出现

趋势一：逻辑漏洞越来越多



趋势二：远程加载漏洞利用的手法越来越流行

2014年

CVE-2014-4114

2017年

CVE-2017-0199

CVE-2017-8759

2018年

CVE-2018-5002

CVE-2018-8174

CVE-2018-15982

2020年

CVE-2020-0674

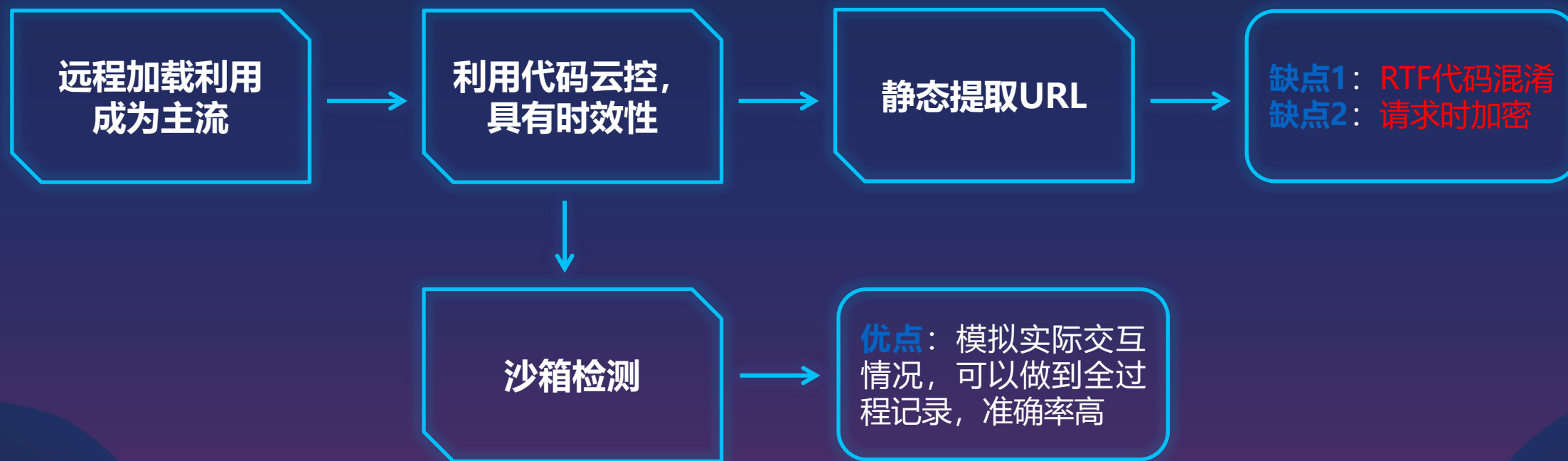
2021年

CVE-2021-40444

2022年

CVE-2022-30190

为何说沙箱适合狩猎Office在野0day



为何说沙箱适合狩猎Office在野0day

逻辑漏洞一般
影响较多版本



沙箱环境制作
工作量较小



只需维护一个常
见Office版本的
全补丁镜像

沙箱能覆盖的Office在野0day狩猎场景

所有以Office为载体的在野0day场景



适用沙箱
狩猎的场景

可行的思路

- 做一个全补丁Office环境镜像，将可疑样本投入沙箱，通过异常行为进行过滤
 - 环境制作：Windows 10 + Office 2019 (仅供参考)
 - 异常行为
 - Office进程启动可疑子进程（需过滤常见白名单）
 - 检测是否有异常地址处的指令访问敏感模块的导出表（即EAF）
- 做一些Office Nday环境镜像，在筛选0day时顺便标定一些被用于攻击的Nday（可选）
 - 需要对Office历史漏洞进行深入研究
 - 静态标定：需要对抗RTF代码混淆，容易遗漏，可参考开源的[quicksand](#)项目
 - 动态标定：需要挂钩常见漏洞模块进行漏洞点Check，技术难度相比静态稍大

目录

1. 背景
2. 用不同的方法解决不同的问题
 - 为什么沙箱适合狩猎Office在野0day
 - **为什么YARA适合狩猎Windows本地提权在野0day**
3. 近两年热门在野0day案例分析
 - Office在野0day案例分析
 - Windows本地提权在野0day案例分析
4. 如何培养在野0day狩猎人才

Windows本地提权在野0day漏洞分类

➤ 内核提权漏洞（本议题重点讨论这类漏洞）

- 由Windows操作系统内核驱动程序的内存破坏引发的提权漏洞，例如：
 - CVE-2021-1732 win32kfull.sys驱动内核提权漏洞

➤ 用户态提权漏洞

- 由内存破坏漏洞导致的打印机提权漏洞，例如：
 - CVE-2020-0986 打印机提权漏洞

➤ 逻辑提权漏洞

- 例如COM提权漏洞和打印机逻辑提权漏洞，例如：
 - CVE-2017-0213 COM提权漏洞

Windows本地提权在野0day使用场景

1. 作为完整漏洞链的一部分，与浏览器等漏洞配合使用

➤ 典型场景

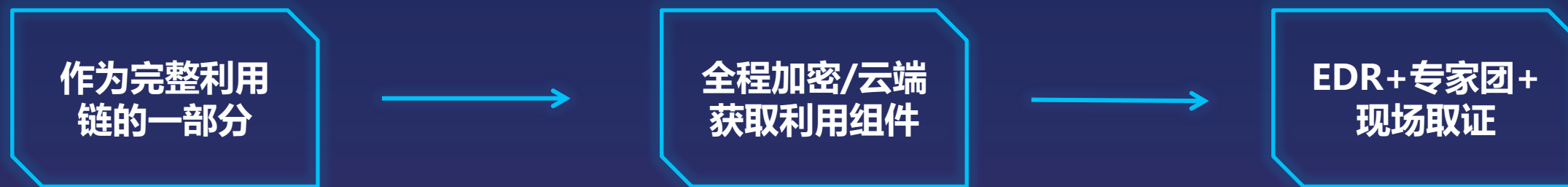
- 以反射型DLL的方式**内嵌在Office漏洞利用代码内**，例如：CVE-2017-0263
- 以反射型DLL的方式**内嵌在浏览器漏洞利用代码内**，例如：CVE-2021-31956
- 前置漏洞利用成功后再**从云端获取并解密使用**，例如：CVE-2018-8453

2. 作为独立组件使用

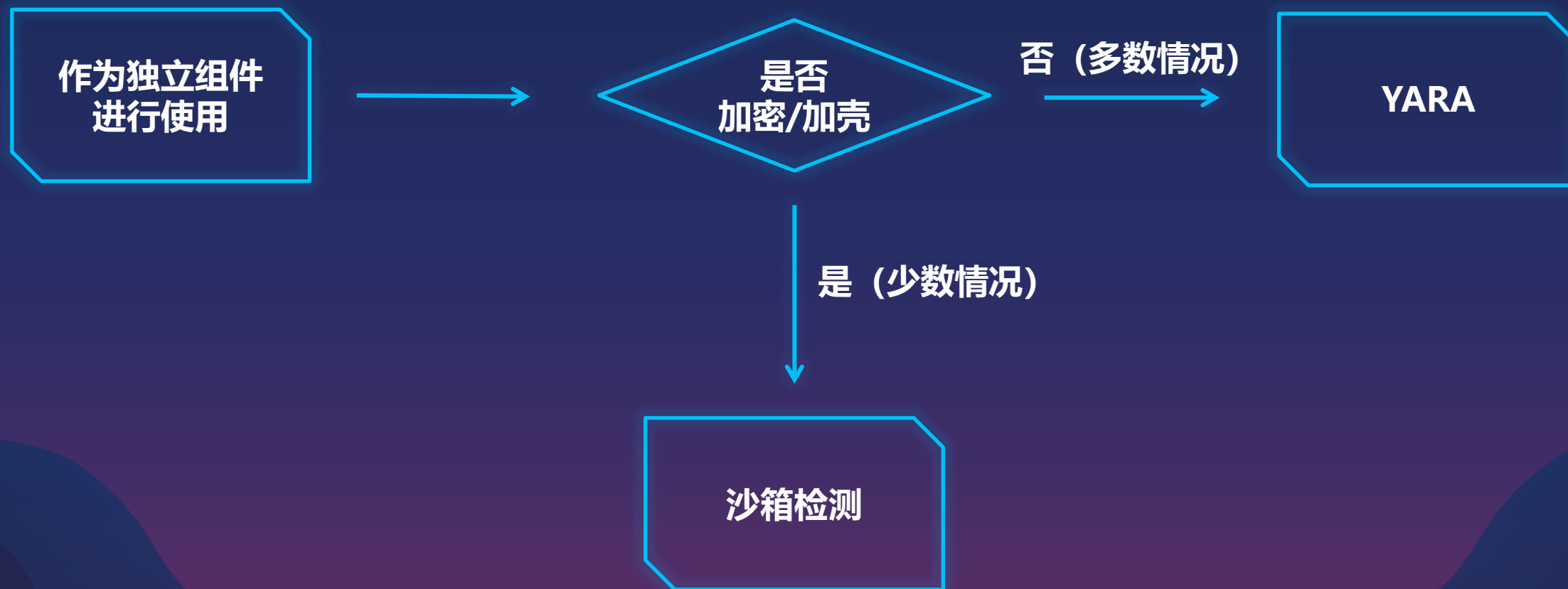
➤ 典型场景

- 下发单个提权组件，**接收参数**进行提权，例如：CVE-2021-1732
- 无需提供任何参数，**直接执行**当前程序提权，例如：CVE-2022-37969

不同场景需要不同的狩猎方法



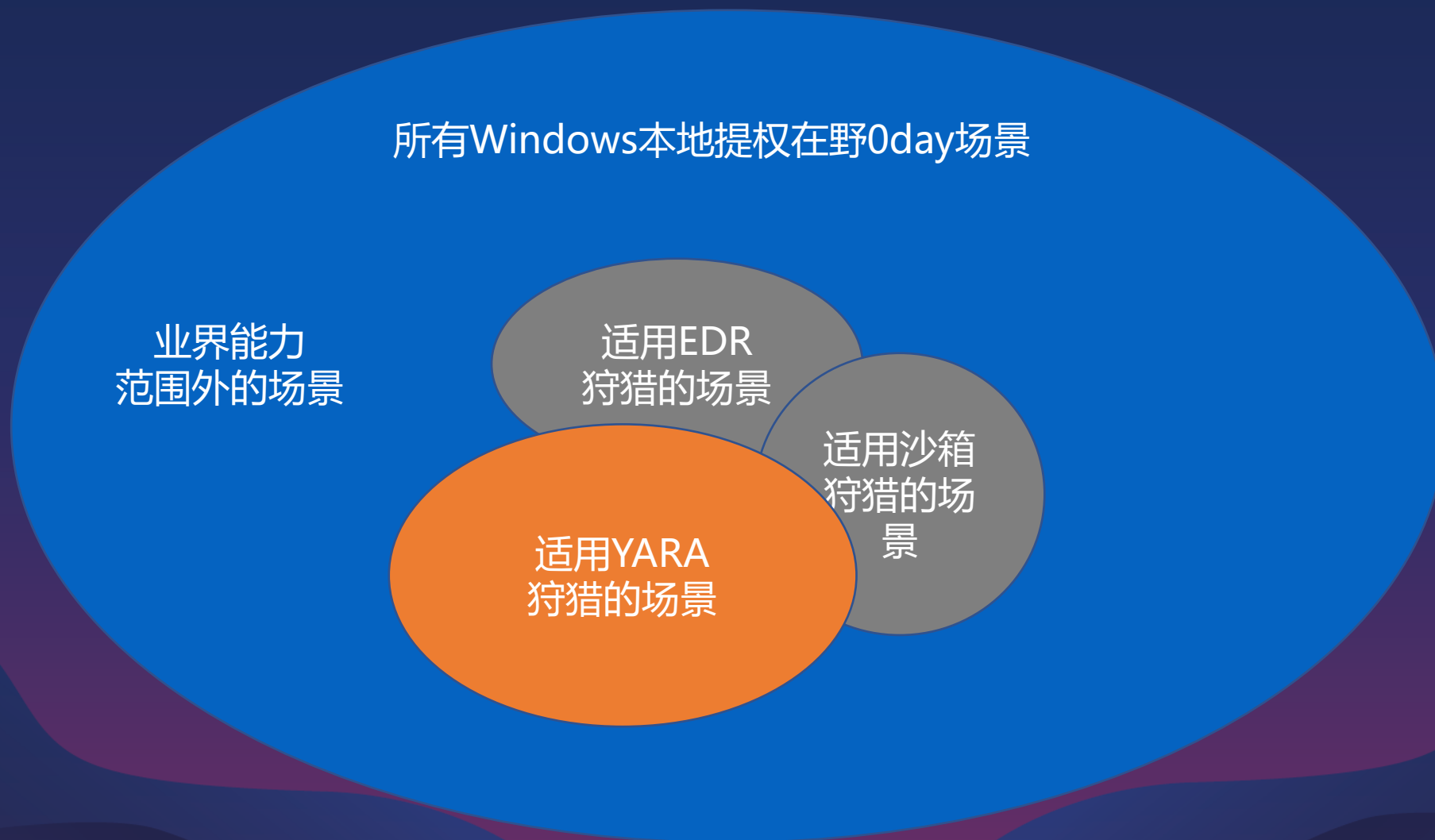
不同场景需要不同的狩猎方法



YARA狩猎本地提权样本的其他优势

1. 漏洞利用代码运行前往往会进行操作系统版本/杀毒软件/沙箱环境检测
 - 商业化提权组件会确保其可靠性，避免非预期的蓝屏或被检测到（规避动态检测）
2. 攻击者经常使用一些经典漏洞利用手法
 - 这些手法随时间流逝具有一定连续性（具备较强的静态特征）
 - HMValidateHandle、Previous Mode等手法
3. 攻击者经常瞄准较为热门的漏洞模块
 - 例如Win32k驱动和CLFS驱动（热门模块的规则具有通用性）

YARA能覆盖的Windows LPE在野0day狩猎场景



如何正确地编写YARA规则



1. 为漏洞利用各个阶段的特征编写规则

2. 为最新的漏洞和利用手法编写规则

3. 为最可能出现的漏洞编写规则

为漏洞利用各个阶段的特征编写规则

➤ 通常，一个Windows内核本地提权漏洞利用包含以下阶段

- ① 漏洞触发
- ② 堆风水
- ③ 内核信息泄露
- ④ 任意地址读写
- ⑤ 控制流劫持
- ⑥ 权限提升

➤ 基于每个阶段的通用特征写规则（详细可参考[我Black Hat USA 2022的演讲](#)）

为最新的漏洞和利用手法编写规则

➤ 借助Pipe Attribute的任意地址读取

- 2020年7月 “[Scoop the Windows 10 pool!](#)” Paul Fariello and Corentin Bayet of Synacktiv

➤ 借助WNF的任意地址读写

- 2021年6月 “[PuzzleMaker attacks with Chrome zero-day exploit chain](#)” Kaspersky
- 2021年7月 “[CVE-2021-31956 Exploiting the Windows Kernel \(NTFS with WNF\)](#)” Alex Plaskett
- 2021年7月 “[Windows Pool OverFlow Exploit](#)” YanZiShuang

➤ 我们基于此捕获了一些有价值的在野提权漏洞样本

- 2021年10月 我们捕获了一个在野CLFS 1day，并发现了一个补丁绕过漏洞：CVE-2022-24481
- 2022年4月 我们捕获了一个在野CVE-2021-31956样本，该样本可实现Chrome浏览器沙箱逃逸

为最可能出现的漏洞编写规则

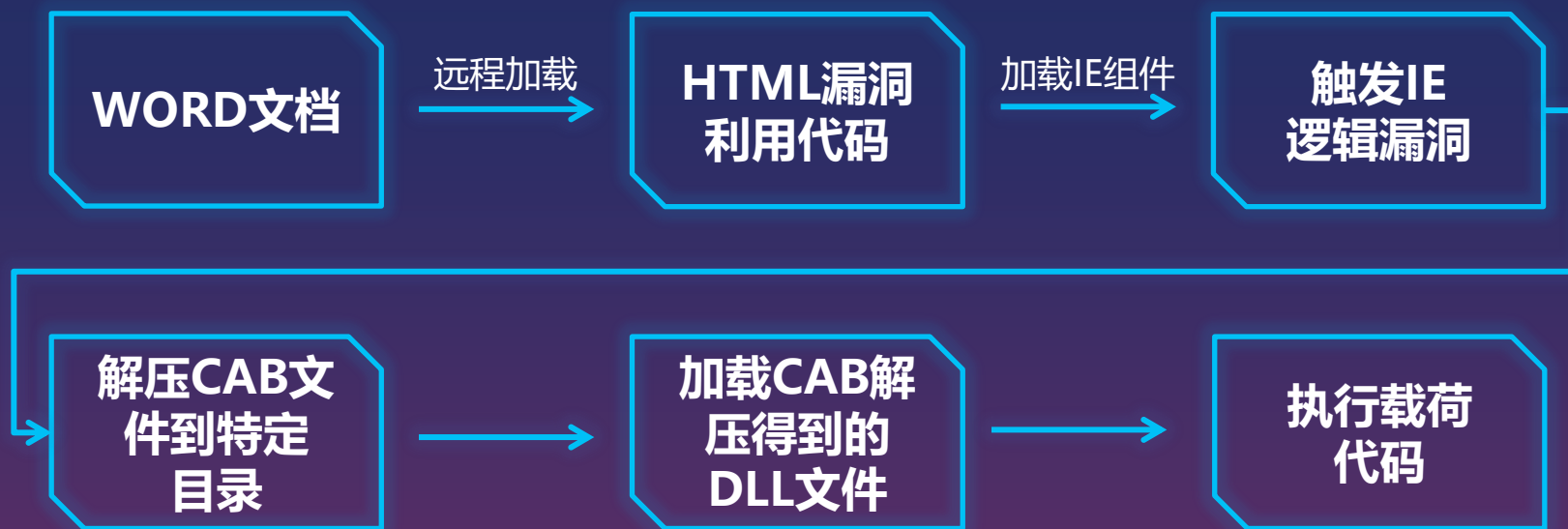
- 桌面窗口管理器 (Desktop Window Manager/DWM) 漏洞的例子
 - 2021年4月13日, 卡巴斯基写了一篇博客, 披露了CVE-2021-28310, 这是一个Windows DWM组件的在野0day
 - "Zero-day vulnerability in Desktop Window Manager (CVE-2021-28310) used in the wild"
 - 2021年5月3日, ZDI发布了一篇博客, 博客中披露了另一个漏洞CVE-2021-26900, 这也是一个Windows DWM组件的漏洞
 - "CVE-2021-26900: Privilege escalation via a use after free vulnerability in win32k"
 - 2021年5月22日, 我们捕获了一个Windows DWM组件的在野0day: CVE-2021-33739

目录

1. 背景
2. 用不同的方法解决不同的问题
 - 为什么沙箱适合狩猎Office在野0day
 - 为什么YARA适合狩猎Windows本地提权在野0day
3. 近两年热门在野0day案例分析
 - **Office在野0day案例分析**
 - Windows本地提权在野0day案例分析
4. 如何培养在野0day狩猎人才

案例分析：CVE-2021-40444

- 2021年9月微软披露的一个IE浏览器在野0day
 - 在野样本借助WORD文档加载远程漏洞利用代码
 - 发现厂商：Mandiant, EXPMON, MSTIC



在野样本攻击流程

案例分析：CVE-2021-40444

➤ 漏洞成因

- ① CAB文件解压路径穿越：在catDirAndFile函数内犯两次错误
- ② .CPL协议加载DLL

➤ 修复方案

- ① 针对CAB路径穿越问题的修补
- ② 针对.CPL协议加载DLL的修补

➤ 详细分析报告

- [《CVE-2021-40444 IE远程代码执行漏洞分析报告》](#)

路径穿越：第一次犯错

➤ 第一次犯错

- 早期版本的`catDirAndFile`函数 (某早期版本的代码示意)
 - 红框内代码完全没有考虑路径穿越问题

```
cbResult -= strlen(pszFile);           // Update remaining size
if (cbResult <= 0) {
    return FALSE;
}
lstrcat(pszResult,pszFile);            // Append file name

    /** Success
return TRUE;
}
```

路径穿越：第二次犯错

➤ 第二次犯错：PathCchCanonicalizeA的缺陷

➤ 某次安全更新后的catDirAndFile函数

- PathCchCanonicalizeA: 将当前路径转换为规范化路径
 - 只能处理 “..\” , 无法处理 “../”
- PathIsPrefixA: 检查第二参数的路径是否是以第一参数的路径作为前缀
 - BOOL PathIsPrefixA(LPCSTR pszPrefix, LPCSTR pszPath);

```
if ( (signed int)(v4 - strlen((const char *)a4)) > 0 )
{
    StringCchCatA(a4, v11, v13);
    if ( PathCchCanonicalizeA(&pszPath, v5, v5, v8, v9) >= 0
        && (!cchDest
            || PathCchCanonicalizeA(&pszPrefix, v5, (char *)cchDest, v12, v14) >= 0 && PathIsPrefixA(&pszPrefix, &pszPath)) )
    {
        return 1;
    }
}
return 0;
```

路径穿越：第二次犯错

➤ 检视PathIsPrefixA的校验情况

➤ pszPrefix: C:\Users\test\AppData\Local\Temp\Cab13F2

➤ 考虑pszPath的两种情况

➤ 如果传入的路径中有 “..\” , PathCchCanonicalizeA 会将其转化为当前路径的上级路径:

- 转换前: C:\Users\test\AppData\Local\Temp\Cab13F2\..\championship.inf
- 转换后: C:\Users\test\AppData\Local\Temp\championship.inf <- 校验失败, 无法穿越

➤ 如果传入的路径中有 “../” , PathCchCanonicalizeA 不会将其转化为当前路径的上级路径:

- 转换前: C:\Users\test\AppData\Local\Temp\Cab13F2\../championship.inf
- 转换后: C:\Users\test\AppData\Local\Temp\Cab13F2\../championship.inf <- 校验成功, 可以穿越

针对CAB路径穿越的完整修复

➤ CVE-2021-40444的补丁

➤ 第二次修复后的PathCchCanonicalizeA函数

- 加入了对 “../” 这一情况的处理

```
if ( *(_BYTE *)a4 && (signed int)(v6 - strlen((const char *)a4)) > 0 )
{
    StringCchCatA(a4, v13, v16);
    v11 = strlen(v4);
    if ( v11 )
    {
        do
        {
            if ( v4[v5] == '/' )
                v4[v5] = '\\';
            ++v5;
        }
        while ( v5 < v11 );
    }
    if ( PathCchCanonicalizeA(v4, v14, v17) >= 0
        && (!cchDest || PathCchCanonicalizeA((char *)cchDest, v15, v18) >= 0 && PathIsPrefixA(&pszPrefix, &pszPath)) )
    {
        return 1;
    }
}
return 0;
```

思考：还存在类似问题吗？



Omri Herscovici

@omriher



It turns out that by replacing `\\` with '/', you can bypass Microsoft's PathCchCanonicalize used to mitigate Path-Traversal.

By [@EyalItkin](#) and his ongoing saga of Reverse RDP Attack.

目录

1. 背景
2. 用不同的方法解决不同的问题
 - 为什么沙箱适合狩猎Office在野0day
 - 为什么YARA适合狩猎Windows本地提权在野0day
3. 近两年热门在野0day案例分析
 - Office在野0day案例分析
 - **Windows本地提权在野0day案例分析**
4. 如何培养在野0day狩猎人才

案例分析: CVE-2022-24521

- 2022年4月微软披露的一个CLFS驱动本地提权在野0day
 - 在野样本作为一个独立组件使用
 - 发现厂商: **CrowdStrike, NSA**
- 事后复盘: 其他厂商有没有可能发现这个在野0day?
 - 线索1: [“Cisco Talos shares insights related to recent cyber attack on Cisco”](#)
 - bb62138d173de997b36e9b07c20b2ca13ea15e9e6cd75ea0e8162e0d3ded83b7
 - 首次提交VirusTotal时间: **2022-03-18 16:21:46 UTC**
 - **加密**压缩包: bdo.zip

名称	创建时间	CRC	大小	压缩后大小	修改时间	访问时间
ex.exe	2022-02-25 19:46	50A9F3E7	296 944	296 956	2022-01-31 17:48	2022-02-25 19:46
ex64.exe	2022-02-25 19:53	20A6034A	302 064	302 076	2022-02-25 19:53	2022-02-25 19:53
exploit.exe	2022-02-25 19:53	C4396C9A	296 944	296 956	2022-02-25 19:53	2022-02-25 19:53
exploit_64.exe	2022-02-25 19:53	1A70AFBB	302 064	302 076	2022-02-25 19:53	2022-02-25 19:53

案例分析: CVE-2022-24521

➤ 事后复盘: 其他厂商有没有可能发现这个在野0day?

➤ 线索2: [“Cisco Talos shares insights related to recent cyber attack on Cisco”](#)

- eb3452c64970f805f1448b78cd3c05d851d758421896edd5dfbe68e08e783d18
- 首次提交VirusTotal时间: 2022-08-25 12:05:44 UTC <- 提交到VT的时间太晚
- 与压缩包内的 `ex.exe` 文件高度吻合:
 - 文件编译时间: 2021-10-28 22:05:33 UTC <- 样本在几个月前已编译完成
 - 数字签名时间: 2022-01-31 01:43:00 UTC <- 与压缩包内文件修改时间吻合
 - CRC32: 50A9F3E7 <- 与原始文件完全一致

➤ 结论: 有一定难度, 但样本就在眼皮底下

- 从2022年3月18日到2022年4月12日(一个月的窗口期), 只要能够解密该压缩包, 就有可能发现内含的CVE-2022-24521在野0day样本

漏洞成因

- CLFS驱动代码缺乏校验
 - SignatureOffset指向的区域与某个Container Context的pContainer指针重合
 - 调用ClfsEncodeBlock时导致pContainer被覆盖为用户伪造的fakeContainer

```
typedef struct _CLFS_LOG_BLOCK_HEADER
{
    UCHAR MajorVersion;
    UCHAR MinorVersion;
    UCHAR Usn;
    CLFS_CLIENT_ID ClientId;
    USHORT TotalSectorCount;
    USHORT ValidSectorCount;
    ULONG Padding;
    ULONG Checksum;
    ULONG Flags;
    CLFS_LSN CurrentLsn;
    CLFS_LSN NextLsn;
    ULONG RecordOffsets[16];
    ULONG SignaturesOffset;
} CLFS_LOG_BLOCK_HEADER, *PCLFS_LOG_BLOCK_HEADER;
```

```
typedef struct _CLFS_CONTAINER_CONTEXT
{
    CLFS_NODE_ID cidNode;
    ULONGLONG cbContainer;
    CLFS_CONTAINER_ID cidContainer;
    CLFS_CONTAINER_ID cidQueue;
    union
    {
        CClfsContainer* pContainer;
        ULONGLONG ullAlignment;
    };
    CLFS_USN usnCurrent;
    CLFS_CONTAINER_STATE eState;
    ULONG cbPrevOffset;
    ULONG cbNextOffset;
} CLFS_CONTAINER_CONTEXT, *PCLFS_CONTAINER_CONTEXT;
```

漏洞利用：借助漏洞伪造虚假对象

- 借助漏洞，内存中的pContainer指针被覆盖为伪造的fakeContainer指针，从而被替换为一个伪造的对象，该对象里面又伪造了一个虚函数表

```
pLogBlockHeader->SignaturesOffset = 0x1600  
pBaseLogRecord->rgContainers[0xA] = 0x1592  
pContainerCtx->pContainer = fakeContainer
```

```
1: kd> dq ffffe50a`9e7f8602 <- 内存中的CLFS_CONTAINER_CONTEXT对象  
ffffe50a`9e7f8602  00000000`00000000 00000000`00080000  
ffffe50a`9e7f8612  ffffffff`00000002 00000098`84eff990 <- 伪造的对象指针  
ffffe50a`9e7f8622  00000000`00000000 00000000`00000000  
...cut...  
  
1: kd> dq 00000098`84eff990  
00000098`84eff990  00000098`84eff890 00000000`00000000 <- 伪造的虚表指针  
...cut...
```

漏洞利用：任意地址写入

➤ 任意地址写入

➤ 借助伪造的虚函数表，劫持CCLfsContainer对象的两次虚函数调用，进行任意地址写入

- CLFS!CCLfsContainer::Release -> nt!HalpDmaPowerCriticalTransitionCallback
- CLFS!CCLfsContainer::Remove -> nt!XmXchgOp

```
/* CLFS!CCLfsBaseFilePersisted::RemoveContainer 代码片段*/
44 88 74 24 20      mov     [rsp+58h+var_38], r14b
48 8B 07            mov     rax, [rdi]
48 8B 40 18         mov     rax, [rax+18h]
48 8B CF           mov     rcx, rdi
FF 15 B2 52 FD FF  call     cs:__guard_dispatch_icall_fptr <- 第一次调用
48 8B 07            mov     rax, [rdi]
48 8B 40 08         mov     rax, [rax+8]
48 8B CF           mov     rcx, rdi
FF 15 A2 52 FD FF  call     cs:__guard_dispatch_icall_fptr <- 第二次调用
```


漏洞利用：任意地址读取

➤ 任意地址读取

- 使用 “[Scoop the Windows 10 pool!](#)” 这篇文章中提到的Pipe Attribute手法

```
struct PipeAttribute {  
    LIST_ENTRY list;  
    char * AttributeName;  
    uint64_t AttributeValueSize;    <- 待读取大小  
    char * AttributeValue;        <- 待读取地址  
    char data [0];  
};
```

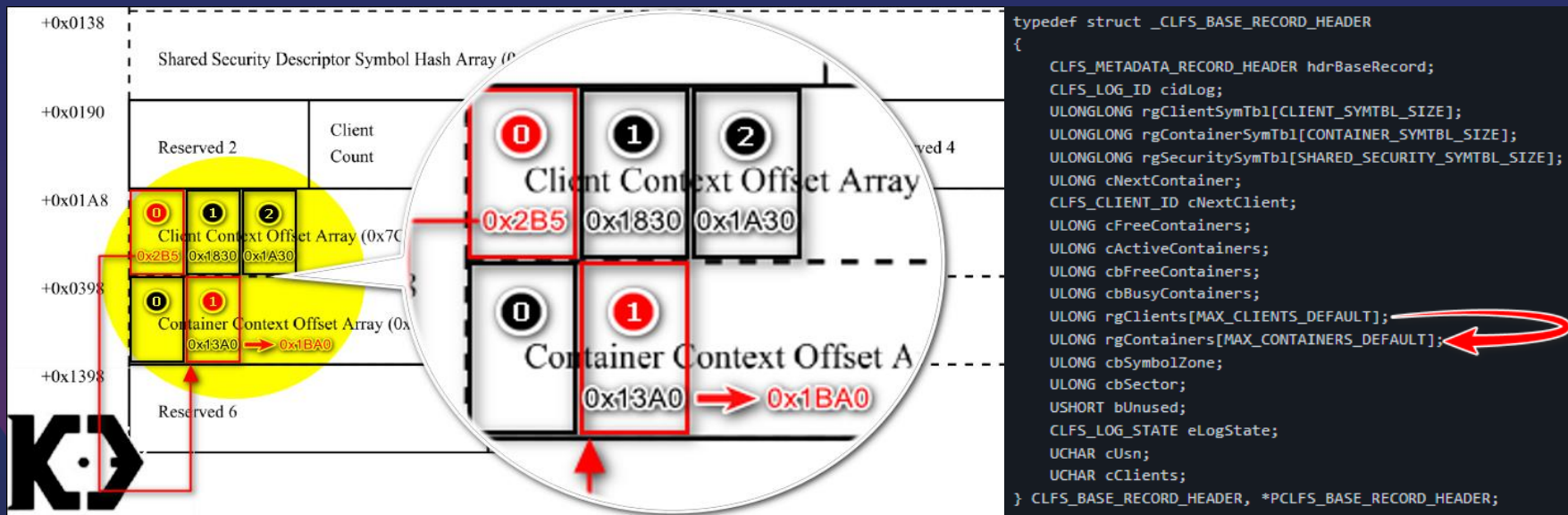
- 使用SystemBigPoolInformation来泄露内核堆块信息

```
hRes = NtQuerySystemInformation(SystemBigPoolInformation, pBuffer,  
    dwBufSize, &dwOutSize);
```

同源样本：CVE-2021-????? (1Day)

➤ 除了使用的漏洞不同，其他手法完全一致

- SHA256: 86a8f267cf0f51c032f7b1777eb1e51f7cd1badf3f3894e2557a3f571fca9f3d
- 文件编译时间: 2021-09-30 20:29:57 UTC
- 该漏洞的更多细节可以参考[我在Black Hat USA 2022的演讲](#)



更多CLFS漏洞在野样本

More ?

1Day

1Day

0Day

0Day

CVE-2021-?????

CVE-2022-24481

CVE-2022-24521

CVE-2022-37969

目录

1. 背景
2. 用不同的方法解决不同的问题
 - 为什么沙箱适合狩猎Office在野0day
 - 为什么YARA适合狩猎Windows本地提权在野0day
3. 近两年热门在野0day案例分析
 - Office在野0day案例分析
 - Windows本地提权在野0day案例分析
- 4. 如何培养在野0day狩猎人才**

如何培养在野0day狩猎人才

- 该领域做得比较好的团队/公司（基于公开统计数据，捕获在野0day数量 ≥ 3 ）
 - Google TAG
 - Mandiant（已被Google收购）
 - Kaspersky GReAT
 - Microsoft
 - 360 ATA
 - 安恒猎影实验室
 - CrowdStrike
 - Trend Micro
- 从统计结果看，都属于少数团队/公司多次捕获在野0day

培养此类人才的几点建议

1. 好苗子很重要（敏锐的嗅觉）
2. 对历史案例的仔细研究（严苛的训练）
3. 手握有价值的数据源（巧妇难为无米之炊）
4. 合适的平台和工具
 - 优质的EDR、沙箱、静态引擎
 - 用不同的工具解决不同的问题
5. 不断模拟演练（一次次复盘优化）
 - 缩短各环节时间消耗：发现、复现、归类、分析
 - 行动力：早发现、早防御、早修复
6. 未知攻，焉知防？
 - 逆向思维（站在攻的角度去思考防）



感谢聆听

Q&A

参考链接

1. <https://unit42.paloaltonetworks.com/unit42-slicing-dicing-cve-2018-5002-payloads-new-chainshot-malware/>
2. <https://bbs.pediy.com/thread-193443.htm>
3. <https://www.mandiant.com/resources/blog/cve-2017-0199-hta-handler>
4. <https://ti.dbappsecurity.com.cn/blog/index.php/2021/02/10/windows-kernel-zero-day-exploit-is-used-by-bitter-apt-in-targeted-attack/>
5. <https://securelist.com/operation-powerfall-cve-2020-0986-and-variants/98329/>
6. <https://bugs.chromium.org/p/project-zero/issues/detail?id=1107>
7. <https://securelist.com/cve-2018-8453-used-in-targeted-attacks/88151/>
8. <https://i.blackhat.com/USA-22/Thursday/us-22-Jin-The-Journey-Of-Hunting-ITW-Windows-LPE-0day.pdf>

参考链接

9. https://github.com/synacktiv/Windows-kernel-SegmentHeap-Aligned-Chunk-Confusion/blob/master/Scoop_The_Windows_10_pool.pdf
10. <https://research.nccgroup.com/2021/07/15/cve-2021-31956-exploiting-the-windows-kernel-ntfs-with-wnf-part-1/>
11. <https://securelist.com/zero-day-vulnerability-in-desktop-window-manager-cve-2021-28310-used-in-the-wild/101898/>
12. <https://www.zerodayinitiative.com/blog/2021/5/3/cve-2021-26900-privilege-escalation-via-a-use-after-free-vulnerability-in-win32k>
13. <https://bbs.pediy.com/thread-270017.htm>
14. <https://blog.talosintelligence.com/2022/08/recent-cyber-attack.html>
15. <https://www.pixiepointsecurity.com/blog/nday-cve-2022-24521.html>