CVE-2019-0863漏洞分析

angel010 / 2019-07-07 08:43:00 / 浏览数 4843 安全技术 漏洞分析 顶(0) 踩(0)

2018年12月，黑客SandboxEscaper公布了Windows Error Reporting (WER)组件的一个0 day漏洞。经过分析，研究人员发现了另外一个漏洞可以配合该漏洞来进行权限提升。根据微软发布的公告，在2019年5月发布补丁前，该漏洞都是0 day漏洞。那么该漏洞是如何工作的呢？

Microsoft WER

Windows Error Reporting 工具是一个灵活的基于事件的反馈基础设施，用来收集关于软硬件的问题，然后将信息报告给微软，然后微软提供对应的解决方案。比如，如果Windows系统奔溃了，那么就会生成错误报告，并保存在WER报告队列目录`C:\ProgramData\Microsoft\Windows\WER\ReportQueue`中，每个报告都会INI文件。为了让所有进程都报告错误情况，所有用户都有`ReportQueue`目录的写权限，如下图所示：

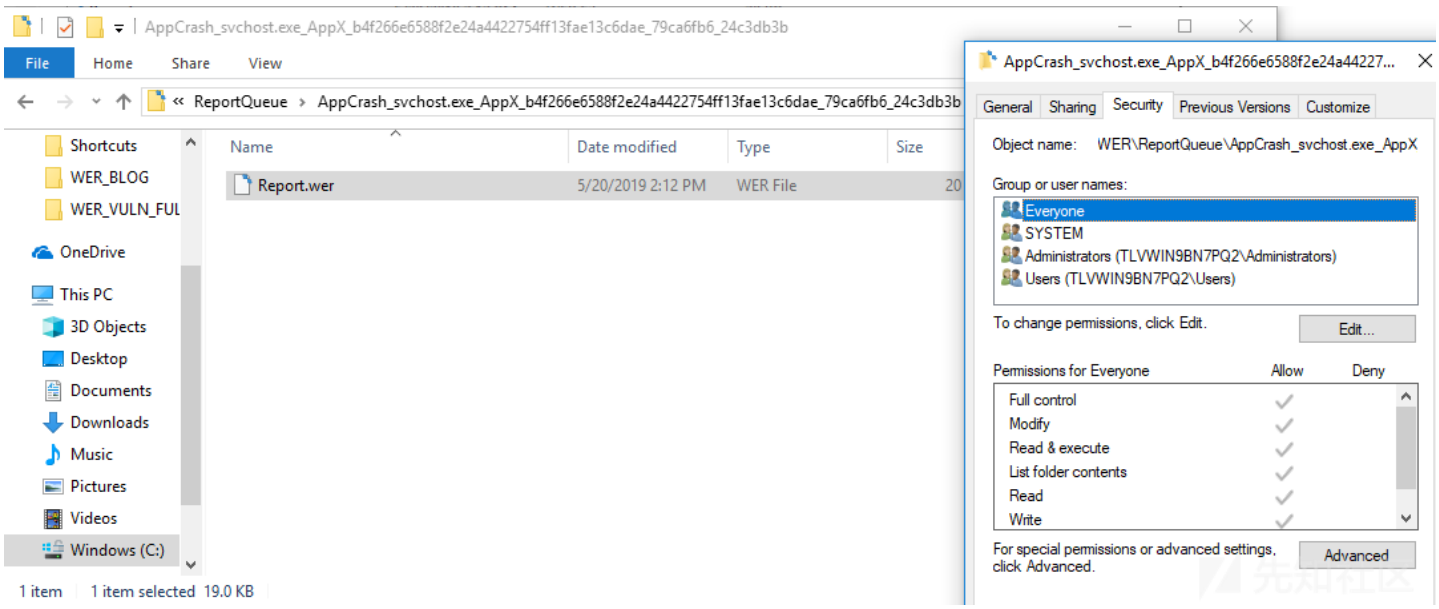

图1. Windows Error Reporting queue目录

报告生成后，就会发送给微软进行下一步分析。这种交互有很多种触发方式，其中一种方式就是使用`Windows Error Reporting\QueueReporting`计划任务。从安全的角度来分析，该任务很有意思，因为：

- 它是以System权限运行的
- 可以在需要时触发
- 用固定的命令行参数wermgr.exe -upload来运行特定的二进制代码。

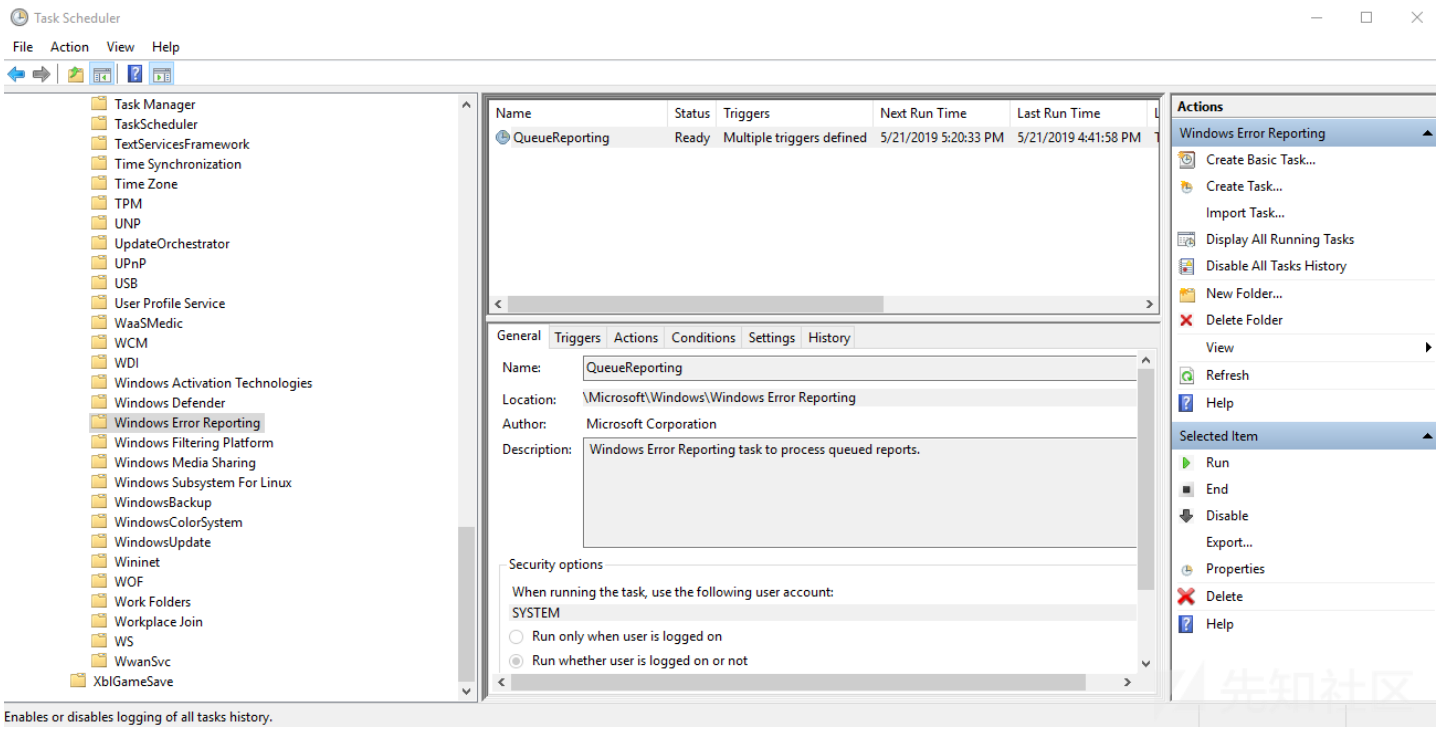

图2. Windows Error Reporting计划任务
执行后，wermgr.exe
会与暂停的报告文件和目录进行交互。读取文件、分析并复制到其他目录中，甚至会删除。因为任何用户都有写权限，如果不注意，就会产生一些安全漏洞。

滥用文件系统链接

Windows系统支持不同类型的文件系统链接，文件系统链接可以将文件和目录指向其他目标文件和目录。一旦链接被扫描或重解析后，就会将用户重定向到目标路径。从安

下面的例子解释了对`kernel32.dll`没有写权限的用户可以在`c:\temp\Dir\x.dll`和`C:\Windows\System32\kernel32.dll`之间创建一个链接。如果可以重定向到更

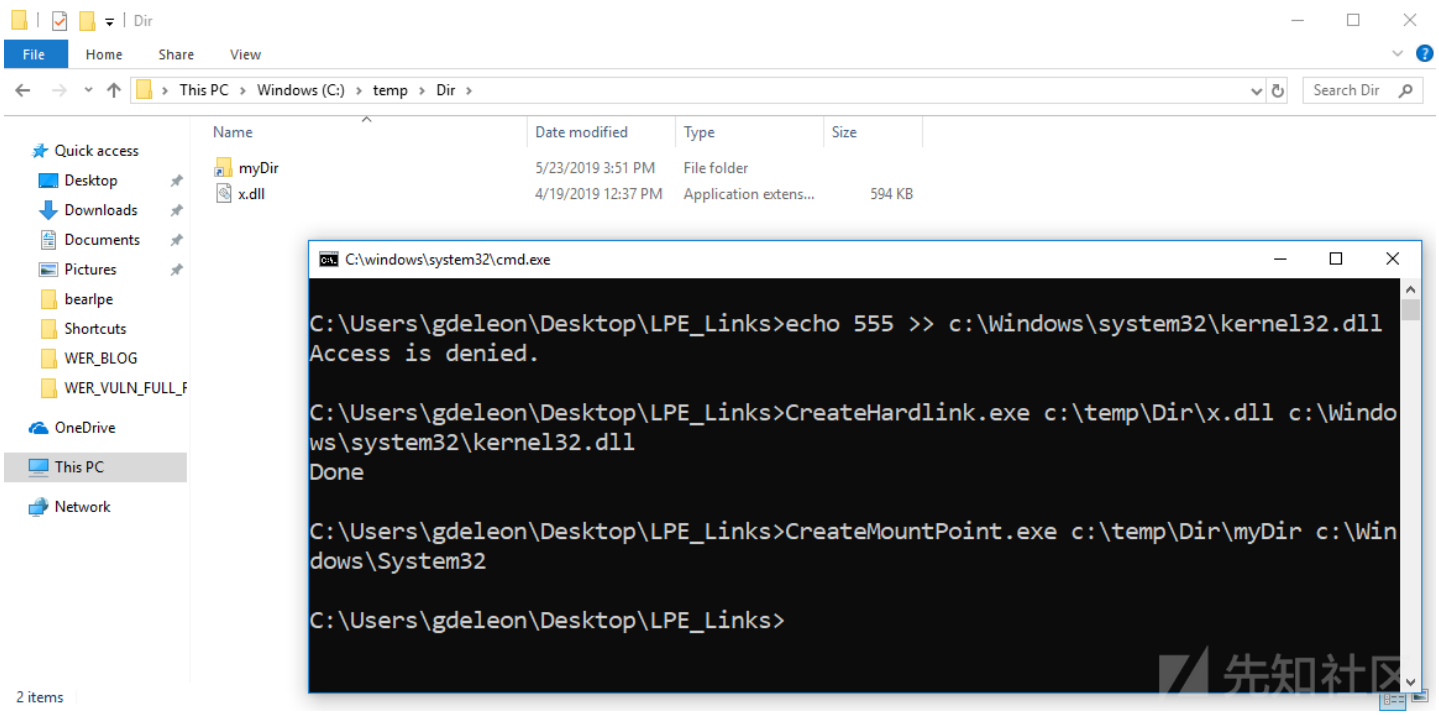

图3. 创建到用户本没有写权限的文件的硬链接

PoC

简而言之，黑客可以利用WER的能力来修改文件权限来对任意文件分配读、写、编辑和删除权限，具体来说就是使用前面提到的文件系统链接技术将report目录中的文件链
下面介绍一个完全的bug利用场景。

场景1：

Step 1: `wermger.exe`分析report目录中的所有文件，并提交给微软:

```
int64 DoCoreUpload(/* ... */) {
   /* ... */
   Ret = WerpSubmitReportFromStore(ReportPath, /* ... */);
   if (Ret >= 0) {
       /* Report successfully uploaded */
   } else {
       if (Ret == ERROR_FILE_CORRUPT) {
           DeleteCorruptedReportFromStore(ReportPath);
       }
   }
}
```

Step 2: 当 `wermger.exe`检测到损坏的`Report.werINI`文件，最终会删除它，但首先它要增加进程执行权限修改文件的DACL特征才能删除该文件。
漏洞利用：黑客利用`wermger.exe`读取文件DACL权限的这个小的窗口期，来增加对该文件的删除权限。如果攻击者创建了该文件与系统中其他文件的链接，DACL读取后，

场景2

Step 1:
首先， wermgr.exe -upload 调用`wermgr!DoCoreUpload`函数，该函数会列出`ReportQueue`下的所有子目录。读取错误报告并提交给微软。
Step 2:
如果 wermgr.exe 发现损坏的`Report.werINI`文件，就修改其DACL，之后再删除它。具体来看：

- 首先，wermgr!DeleteCorruptedReportFromStore列出所有的子目录的文件；

然后，wermgr!PreparePathForDeletion修改每个文件的权限。因为函数使用kernel32!GetFileSecurity读取了文件的安全描述符，并调用kernel32!SetFileSecurity来应

```
int64 PreparePathForDeletion(wchar_t* FileName) {
 PSECURITY_DESCRIPTOR SecurityDescriptor = NULL;
 DWORD BytesRead = 0;
 PDACL Dacl = NULL;
 /* ... */

 if ( !GetFileSecurity(FileName,
                DACL_SECURITY_INFORMATION,
                NULL, 0, &BytesRead) ) {
    /* ... */
    return;
 }

 SecurityDescriptor = new BYTE[BytesRead];

 if ( !GetFileSecurity(FileName,
                DACL_SECURITY_INFORMATION,
                SecurityDescriptor,
                BytesRead, &BytesRead) ) {
    /* ... */
    return;
 }

 if ( GetSecurityDescriptorDacl(SecurityDescriptor,
                     &DaclPresent,
                     &Dacl, &DaclDefaulted) )
 {
    /* ... */
    HANDLE TokenHandle = NULL;
    PACL NewAcl = NULL;
    EXPLICIT_ACCESS ExplicitAccess = {0};

    /* ... */
    LPVOID UserName = new BYTE[/* ... */];
    GetTokenInformation(TokenHandle, TokenUser,
                  UserName, &BytesRead);

    ExplicitAccess.Trustee.ptstrName = UserName;
    ExplicitAccess.Trustee.TrusteeType = TRUSTEE_IS_NAME;
    ExplicitAccess.grfAccessMode = GRANT_ACCESS;
    ExplicitAccess.grfAccessPermissions = DELETE | /* ... */;
    /* ... */

    SetEntriesInAcl(1, &ExplicitAccess, Dacl, &NewAcl);
    InitializeSecurityDescriptor(&SecurityDescriptor, 1);
    SetSecurityDescriptorDacl(&SecurityDescriptor, 1, NewAcl, 0);
    SetFileSecurity(FilePath, DACL_SECURITY_INFORMATION,
                &SecurityDescriptor);
 }
}
```

在正确的时间创建链接是非常困难的，黑客会不断地尝试直到成功为止。攻击者可能会攻击DLL、EXE和脚本等可执行文件，用恶意payload来覆盖他们，然后用System权

https://unit42.paloaltonetworks.com/tale-of-a-windows-error-reporting-zero-day-cve-2019-0863/

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板