gtworek / PSBits Public

Notifications Fork 525 Star 3.2k

Files

e97cbbb

Go to file

- AppLockerBypass
- CERTPL2Hosts
- CopyEAs
- DFIR
- DNS
- EnableAllParentPrivileges
- FMAPI
- FakeAMSI
- FakeCmdLine
- GPO
- GetWindowFlag
- HashSrv
- HideSnapshot
- IFilter
- IOCTL_VOLSNAP_SET_MAX_DIFF...
- LSASecretDumper
- LoLBIN
- MSI_Payload
- Misc
- NTDSdiff
- NTFSObjectID
- NetstatWithTimestamps
- NoDLP
  - bin2wav.ps1
  - bin2word.ps1
  - readme.md
  - wav2bin.ps1
  - word2bin.ps1
- NoRebootSvc
- NoRunDll
- NtPowerInformation
- OfflineSAM
- PasswordStealing
- ProcessMitigations
- RDPHoneyPot
- RegExport

PSBits / NoDLP / bin2wav.ps1

gtworek Update bin2wav.ps1 755522f · last year History

Code Blame 117 lines (96 loc) · 2.86 KB Raw

```powershell
# PoC only. Filenames are hardcoded.

$msPerBit = 500 # must be divisible by 10
$srcData = Get-Content -Path C:\temp\test1.txt -Encoding Byte

$startStopBits = $true

[byte[]]$wavHeader = @()

#WAWRIFFHEADER
$wavHeader += (0x52, 0x49, 0x46, 0x46) #RIFF
$wavHeader += (0,0,0,0) #size1. to be filled later. 36+size2. Offset 0x04.
$wavHeader += (0x57, 0x41, 0x56, 0x45) #WAVE

#FORMATCHUNK
$wavHeader += (0x66, 0x6d, 0x74, 0x20) #fmt
$wavHeader += (0x10, 0x00, 0x00, 0x00) #chunk size
$wavHeader += (0x01, 0x00) #FormatTag = PCM
$wavHeader += (0x01, 0x00) #channels
$wavHeader += (0x44, 0xAC, 0x00, 0x00) #samples per second
$wavHeader += (0x44, 0xAC, 0x00, 0x00) #bytes per second
$wavHeader += (0x01, 0x00) #block align
$wavHeader += (0x08, 0x00) #bits per sample

#DATACHUNK
$wavHeader += (0x64, 0x61, 0x74, 0x61) #data
$wavHeader += (0x00, 0x00, 0x00, 0x00) # size2. Number of samples Offset 0x28

#size2
$samplesCount = 44.100 * $srcData.Count * $msPerBit * 8

if ($startStopBits)
{
    $samplesCount += (44.100 * 2 * $msPerBit)
}

$wavHeader[0x28] = $samplesCount -band 0xFF
$wavHeader[0x29] = ($samplesCount -band 0xFF00) -shr 8
$wavHeader[0x2A] = ($samplesCount -band 0xFF0000) -shr 16
$wavHeader[0x2B] = ($samplesCount -band 0xFF000000) -shr 24

#totalsize
$wavHeader[0x04] = ($samplesCount + 36) -band 0xFF
$wavHeader[0x05] = (($samplesCount + 36) -band 0xFF00) -shr 8
$wavHeader[0x06] = (($samplesCount + 36) -band 0xFF0000) -shr 16
$wavHeader[0x07] = (($samplesCount + 36) -band 0xFF000000) -shr 24


$filename = 'C:\temp\test1.wav'
del $filename # for PoC
$fsw = new-object IO.FileStream($filename, [IO.FileMode]::CreateNew)
$writer = new-object IO.BinaryWriter($fsw)
$writer.Write($wavHeader)


$loFreq = 300
$hiFreq = 2000
```

- SIP
- ServerLevelPluginDll
- Services
- ShutdownPriority
- SuspendProcess
- Sysmon

```
$hiFreq = 2000

$samplesPerBit = 44.1 * $msPerBit
$body = New-Object byte[] $samplesPerBit

# a bit of data for a start.
if ($startStopBits)
{
    for ($k=0; $k -lt ($samplesPerBit); $k++)
    {
        $body[$k] = 0
        if ((($k+1) % 4410) -eq 0)
        {
            $body[$k] = 255
        }
    }
    $writer.Write($body)
}

for ($i = 0; $i -lt $srcData.Count; $i++)
{
    $srcByte = $srcData[$i]
    for ($j = 0; $j -lt 8; $j++)
    {
        if (($srcByte -band (1 -shl (7 - $j))) -ne 0)
        {
            $freq = $hiFreq
        }
        else
        {
            $freq = $loFreq
        }
        for ($k=0; $k -lt ($samplesPerBit); $k++)
        {
            $body[$k] = [byte](([System.Math]::Sin((2 * $k * [System.Math]::PI * $freq)
        }
        $writer.Write($body)
    }
}

# a bit of data for a stop.
if ($startStopBits)
{
    for ($k=0; $k -lt ($samplesPerBit); $k++)
    {
        $body[$k] = 0
        if ((($k+1) % 4410) -eq 0)
        {
            $body[$k] = 255
        }
    }
    $writer.Write($body)
}


$fsw.Close()

# and now play it
# $PlayWav = New-Object System.Media.SoundPlayer
# $PlayWav.SoundLocation = $filename
# $PlayWav.PlaySync()
```