



CYART

inquiry@cyart.io

www.cyart.io

Living-Off-The-Land Lab

Table of contents

1. Lab Objective	3
2. Tools	3
3. Lab Execution	3
4. Lab Logs	6
5. Summary	6

List of Figures

Figure 3.1 Shows AV bypassed and all process listed	4
Figure 3.2 Shows all local users on system	4
Figure 3.3 Shows current user	5
Figure 3.4 Harvesting credentials	5
Figure 3.5 Shows cleaning up of exported files	5

List of Tables

Table 2.1 Shows Tools	3
Table 4.1 Shows Logs	6



1. Lab Objective

- Demonstrate attacks using native Windows tools (PowerShell, WMI).
- Perform file-less execution to bypass antivirus.
- Harvest system credentials safely in a lab environment.

2. Tools

<i>Tool</i>	<i>Purpose</i>
PowerShell	Execute fileless attacks in memory
WMI	Enumerate users and harvest credentials
Windows VM	Target system for simulation (IP: 192.168.1.38)

Table 2.1 Shows Tools

3. Lab Execution

Step 1: Fileless PowerShell Execution

Encode a PowerShell command

\$command = 'Get-Process'

\$bytes = [System.Text.Encoding]::Unicode.GetBytes(\$command)

\$encoded = [Convert]::ToBase64String(\$bytes)

Execute encoded command (fileless)

*powershell.exe -NoProfile -ExecutionPolicy Bypass -EncodedCommand
\$encoded*

Observation: Processes listed without creating any files on disk. AV bypassed in lab simulation.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> $command = 'Get-Process'
PS C:\Windows\system32> $bytes = [System.Text.Encoding]::Unicode.GetBytes($command)
PS C:\Windows\system32> $encoded = [Convert]::ToBase64String($bytes)
PS C:\Windows\system32> powershell.exe -NoProfile -ExecutionPolicy Bypass -EncodedCommand $encoded
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
149	8	3152	8568	1.22	4696	0	AggregatorHost
304	18	5148	26116	1.59	8548	1	ApplicationFrameHost
183	11	6332	12076	0.05	3144	0	audiodg
271	14	4440	19000	0.59	5332	1	conhost
588	22	1956	5792	1.59	484	0	csrss
437	18	1956	5352	6.83	580	1	csrss
465	17	4616	20624	2.00	4704	1	ctfmon
360	18	3856	13840	0.28	5192	0	dashost
211	17	3744	10760	0.11	3628	0	dllhost
256	25	5964	14376	0.34	7772	1	dllhost
962	47	40912	75644	20.22	1120	1	dwm
3191	115	90604	177832	37.59	7804	1	explorer
50	7	1772	5060	0.14	880	1	fontdrvhost
50	6	1480	3888	0.02	884	0	fontdrvhost
0	0	60	8	0	0	0	Idle
141	9	1516	7240	0.05	6416	1	LocationNotificationWindows
1393	25	7072	18676	17.67	732	0	lsass
0	0	316	61116	19.41	1816	0	Memory Compression
473	16	8748	18008	1.53	2920	0	MpDefenderCoreService
211	17	15880	32212	0.13	1000	1	msedge
1350	46	48116	131800	4.20	2132	1	msedge
460	27	77180	122620	3.63	3140	1	msedge
279	15	11776	30096	0.20	4860	1	msedge
194	11	8500	22368	0.11	6824	1	msedge
158	9	2112	10796	0.05	6936	1	msedge
340	17	12776	41352	1.13	8820	1	msedge
342	29	16756	52672	2.67	584	1	msedgewebview2
149	9	2184	7464	0.06	1044	1	msedgewebview2

Figure 3.1 Shows AV bypassed and all process listed

Step 2: Credential Harvesting via WMI

List Local Users

Get-WmiObject Win32_UserAccount -Filter "LocalAccount='True'"

```
PS C:\Windows\system32> Get-WmiObject Win32_UserAccount -Filter "LocalAccount='True'"
```

```
AccountType : 512
Caption      : DESKTOP-VT1A6VA\Administrator
Domain       : DESKTOP-VT1A6VA
SID          : S-1-5-21-158053766-1501495798-4244170523-500
FullName     : 
Name         : Administrator

AccountType : 512
Caption      : DESKTOP-VT1A6VA\DefaultAccount
Domain       : DESKTOP-VT1A6VA
SID          : S-1-5-21-158053766-1501495798-4244170523-503
FullName     : 
Name         : DefaultAccount

AccountType : 512
Caption      : DESKTOP-VT1A6VA\Guest
Domain       : DESKTOP-VT1A6VA
SID          : S-1-5-21-158053766-1501495798-4244170523-501
FullName     : 
Name         : Guest

AccountType : 512
Caption      : DESKTOP-VT1A6VA\sshd
Domain       : DESKTOP-VT1A6VA
SID          : S-1-5-21-158053766-1501495798-4244170523-1002
FullName     : sshd
Name         : sshd

AccountType : 512
Caption      : DESKTOP-VT1A6VA\WDAGUtilityAccount
Domain       : DESKTOP-VT1A6VA
SID          : S-1-5-21-158053766-1501495798-4244170523-504
FullName     : 
Name         : WDAGUtilityAccount

AccountType : 512
Caption      : DESKTOP-VT1A6VA\windows
Domain       : DESKTOP-VT1A6VA
SID          : S-1-5-21-158053766-1501495798-4244170523-1001
FullName     : 
Name         : windows
```

Figure 3.2 Shows all local users on system

Step 3: Show Current Logged-In User

Get-WmiObject -Class Win32_ComputerSystem | Select-Object UserName

```
PS C:\Windows\system32> Get-WmiObject -Class Win32_ComputerSystem | Select-Object UserName
UserName
-----
DESKTOP-VT1A6VA\windows
```

Figure 3.3 Shows current user

Step 4: Simulated Credential Extraction

Get-WmiObject Win32_UserAccount -Filter "LocalAccount='True'" |

Select-Object Name,SID | Export-Csv C:\Lab\UserList.csv

Observation: Local user accounts exported safely, simulating credential harvesting.

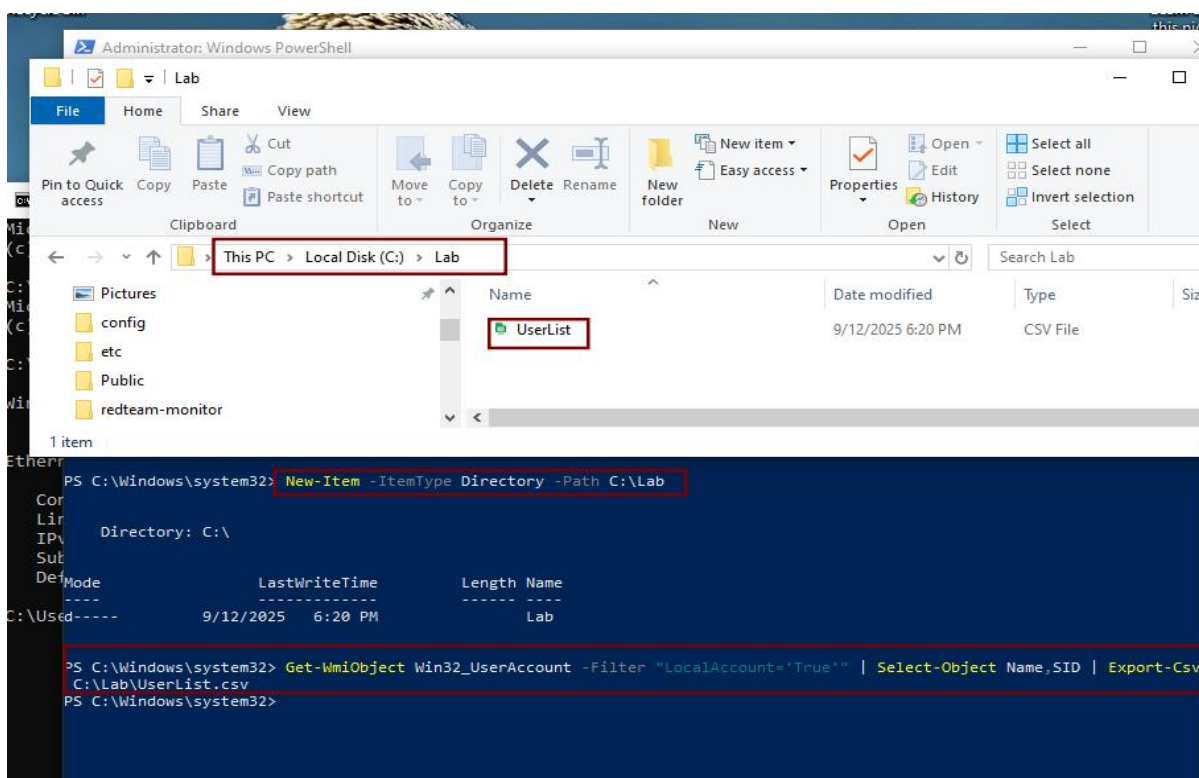


Figure 3.4 Harvesting credentials

Step 5: Cleanup

Remove-Item C:\Lab\UserList.csv

```
PS C:\Windows\system32>
PS C:\Windows\system32> Remove-Item C:\Lab\UserList.csv
PS C:\Windows\system32>
```

Figure 3.5 Shows cleaning up of exported files



4. Lab Logs

<i>Attack ID</i>	<i>Tool</i>	<i>Action</i>	<i>Notes</i>
LID001	PowerShell	Fileless execution	Bypassed AV
LID002	WMI	Credential harvest	Extracted user info (simulated)

Table 4.1 Shows Logs

5. Summary

In the Living-Off-the-Land lab, native Windows tools were leveraged for stealthy attacks. PowerShell enabled file less execution, bypassing antivirus detection, while WMI enumerated and harvested credentials from the system. This demonstrates how attackers can abuse legitimate system tools to perform malicious actions without leaving traces.