



WSL 2 and Security: Productivity Booster or Achilles Heel?





Authors



Max van der Horst

Incident
Responder

Master student at
UvA



Rareș Brătean

Threat Intel
Analyst

Master graduate
from UvA



Jeroen van Saane

Ethical
Hacker

Master graduate
from UvA



Bert-Jan Pals

Security
Analyst

Master student at
UvA



What is WSL?

“Windows Subsystem for Linux (WSL) is a compatibility layer for running Linux binary executables (in ELF format) natively on Windows 10, Windows 11, Windows Server 2019 and Windows Server 2022.” [1]

Linux system inside a Windows environment (Multiple Distributions)

Two versions:

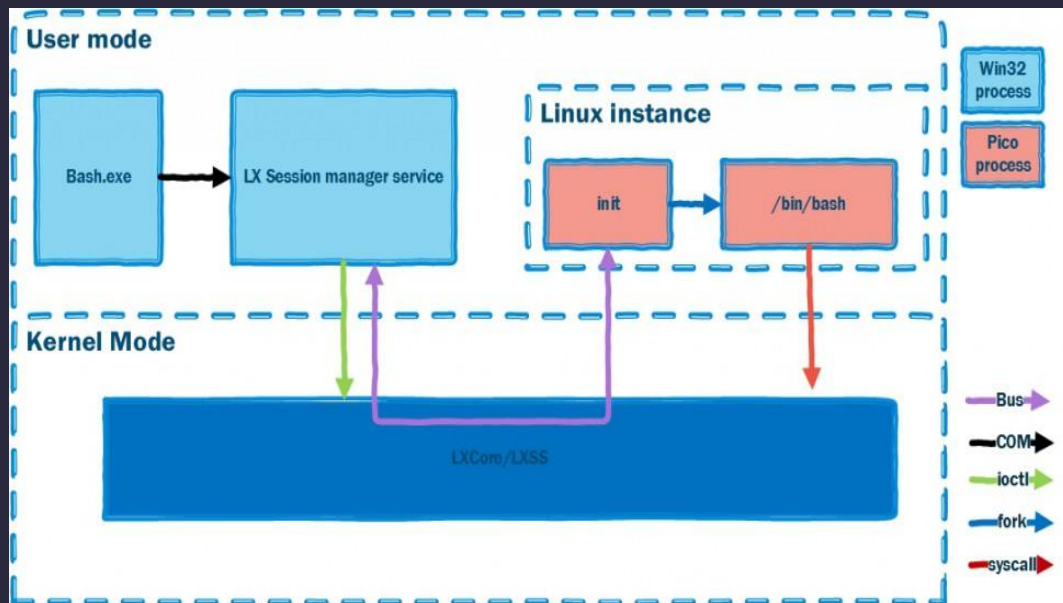
- WSL (2016 - Windows 10)
- WSL 2 (2019 - Windows 10)

[1] https://en.wikipedia.org/wiki/Windows_Subsystem_for_Linux



WSL 1 versus WSL 2

- Translation layer between Linux subsystem and Windows Kernel (syscalls)
- Wine-like fashion
- Used Pico Processes and providers
- Pico process (/bin/bash)
- Pico providers (lxss.sys and lxcore.sys drivers)



[1]

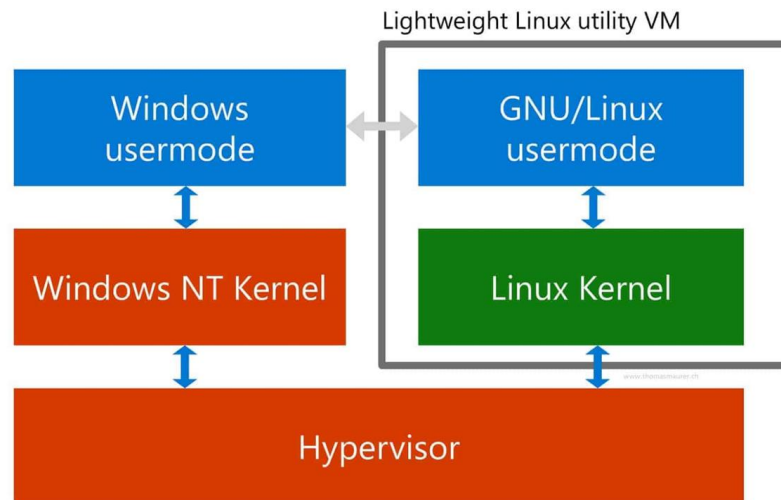
[1] <http://www.priyasaxena.co.uk/2018/04/wsl-in-nutshell.html>



WSL 1 versus WSL 2

- Virtualized approach using a hypervisor (Hyper-V)
- A lightweight VM that contains the Linux Kernel
- Distributed Kernels are open source and managed by Microsoft

WSL 2 architecture overview



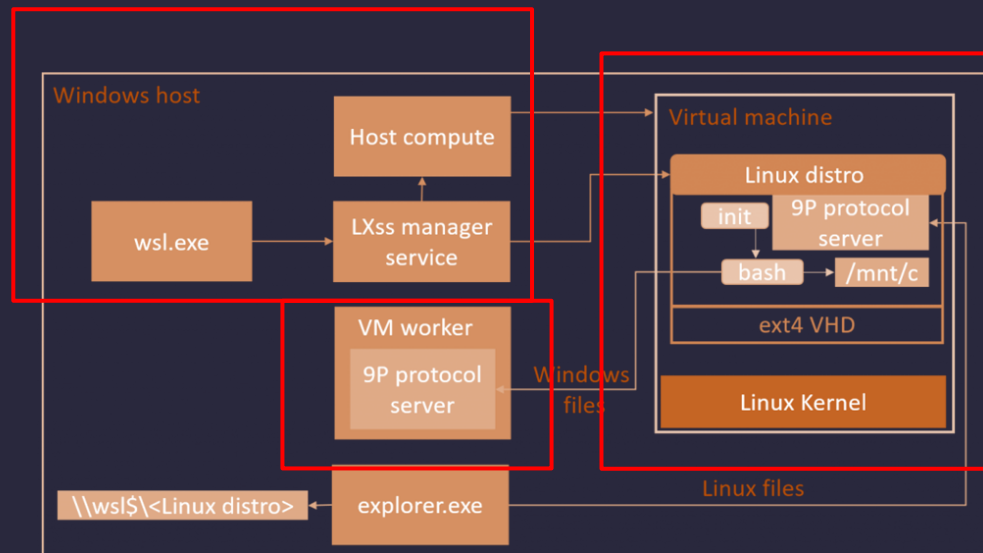
[2]

[2] https://fast.char.gd/public/char_gd/assets/images/_featured/chrome_2019-08-12_16-20-54_e942e65bcc436b594ece1c1a4fd47bda.jpg



WSL 1 versus WSL 2

- 20 times faster
- LXSS.exe session manager and mapper of distribution file system
- Uses vSocket to put stdin from wsl.exe in bash
- Using 9P protocol for filesystem communication (Server - Client)



[3]

[3] https://miro.medium.com/max/1400/0*!Qcj2anwrnXMYfxL.png



What is the addressed issue?

- Syscall translation in WSL 1 led to *Bashware* attacks bypassing Windows security [4]
- Bashware (among things) triggered a redesign of WSL 1, leading to WSL 2



- WSL 2 no longer includes Pico Processes but instead serves as a completely separate VM
- Enter F-Secure, *WSL 2: the other “other” attack surface* [5]

How secure is this redesign? How does it impact your Windows machine?

[4] <https://research.checkpoint.com/2017/beware-bashware-new-method-malware-bypass-security-solutions/>

[5] <https://blog.f-secure.com/wsl2-the-other-other-attack-surface/>

[6] <https://pandorafms.com/blog/wp-content/uploads/2018/11/malBASHware-1.png>



F-Secure Proof of Concept Capability

**Stealthy installation and
enablement WSL 2**

**Download and install Linux
distribution**

Install backdoor

**Expose backdoor
and call C2**



Test Environment

- Windows 10 21H1 build 19043
- WSL 2 build 21364
- Ubuntu 20.04 LTS Distribution



Threat Model

- Compromised Windows 10 machine
- Attacker has full access



[6] <https://assets.ubuntu.com/v1/29985a98-ubuntu-logo32.png>

[7] <https://www.freepnglogos.com/uploads/windows-logo-png/windows-logo-microsoft-exchange-pour-tous-microsoft-exchange-made-22.png>

[8] <https://cdn-icons-png.flaticon.com/512/1897/1897443.png>



How did we test it?

- Find points of interest (baseline) on which to base the attacker stories
- Three levels of Security
 - Default Security
 - Event Logging & Audit Policies
 - Premium Security Product (Sysinternals)





Chosen Attacker Stories

1. Windows Firewall Bypass
2. Detecting Reverse Shells
3. Detecting WSL 2 SSH connections
4. Resource exhaustion using WSL 2
5. Evaluating Possibility of Leakage through memory
6. Identifying WSL 2 Processes in Windows
7. Abusing Windows processes from WSL 2
8. Exploiting Environment Variables



Windows Firewall Bypass

- Block a domain in the windows firewall
 - **SUCCESS**

Default Security

- ☐ None

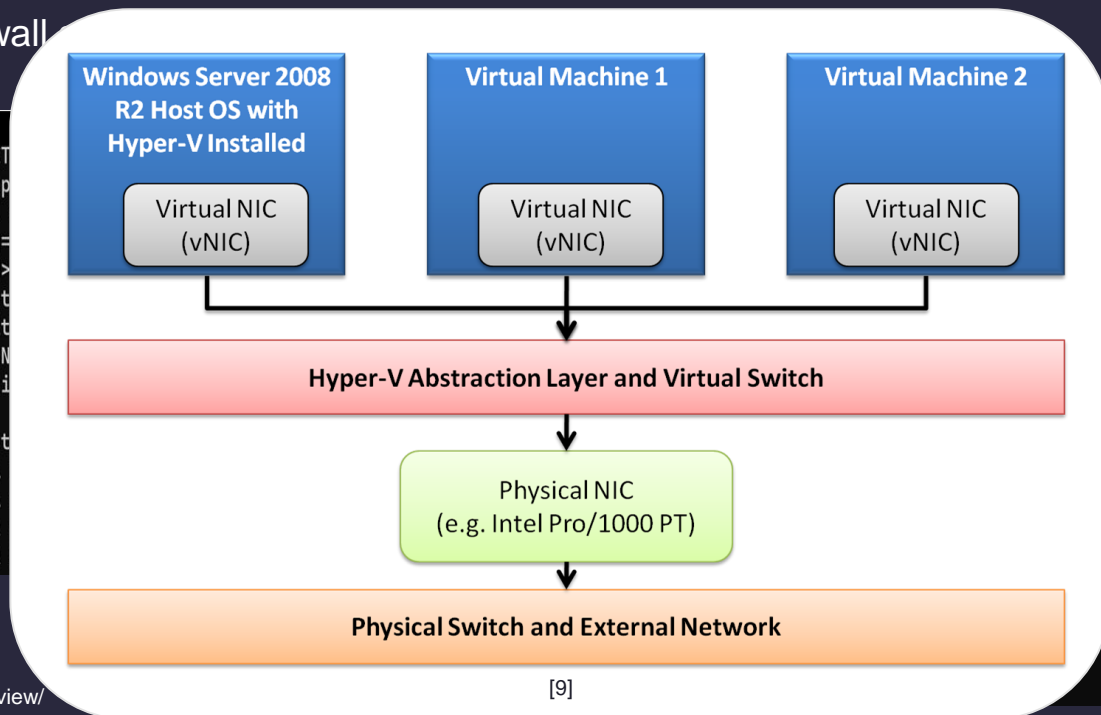
Policy changes

- ☐ Connection from WSL 2 private IP to the blocked IP

Premium security

- ☐ Connection to the Blocked IP from svchost.exe

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="utf-8">  
<title>SN</title>  
<meta name="description">  
<meta name="keywords">  
<link rel="stylesheet" href="css/main.css">
```





Detecting Reverse Shell

- Exfiltrate files through a reverse shell without being detected
 - SUCCESS

Default Security

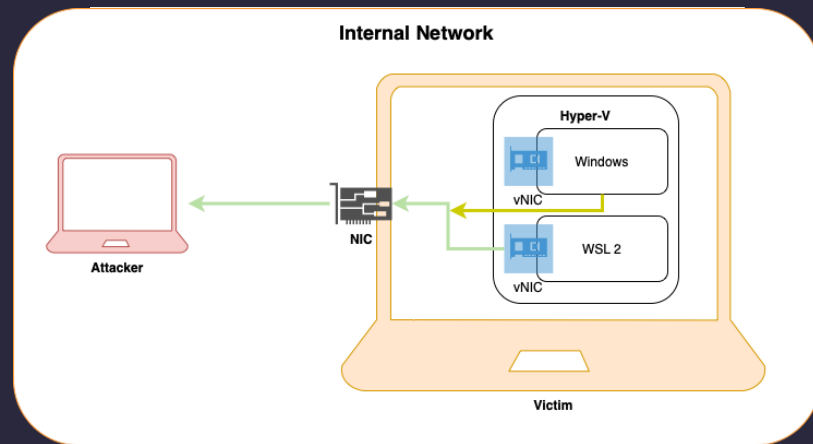
- ☐ None, but port forwarding and traffic was logged

Policy changes

- ☐ None

Premium security

- ☐ TCPView finds established connection including port under svchost.exe





Detecting WSL 2 SSH Connections

- Bypass identification of any SSH connection in the WSL 2 environment
 - **SUCCESS**

Default Security






- ☐ None, but port forwarding was logged

Policy changes

- ☐ SSH connection visible (external IP to port 2222)

Premium security

- ☐ SSH connection visible

Process Name	Process ID	Protocol	State	Local Address	Local Port	Remote Address	Remote Port
 ssh.exe	13204	TCP	Established	172.29.240.1	58757	172.29.240.181	2222
 svchost.exe	3404	TCPv6	Listen	::	7680	::	0
 svchost.exe	1804	TCP	Listen	0.0.0.0	49666	0.0.0.0	0
 svchost.exe	1608	TCP	Listen	0.0.0.0	49667	0.0.0.0	0
 svchost.exe	896	TCP	Listen	0.0.0.0	135	0.0.0.0	0



Resource Exhaustion using WSL2

- Attempt to exceed 80% of CPU/RAM usage by WSL 2
 - SUCCESS
 - LXssManager made unusable by fork bomb : () { : | : & } ; : - memory leak vulnerability

Default Security

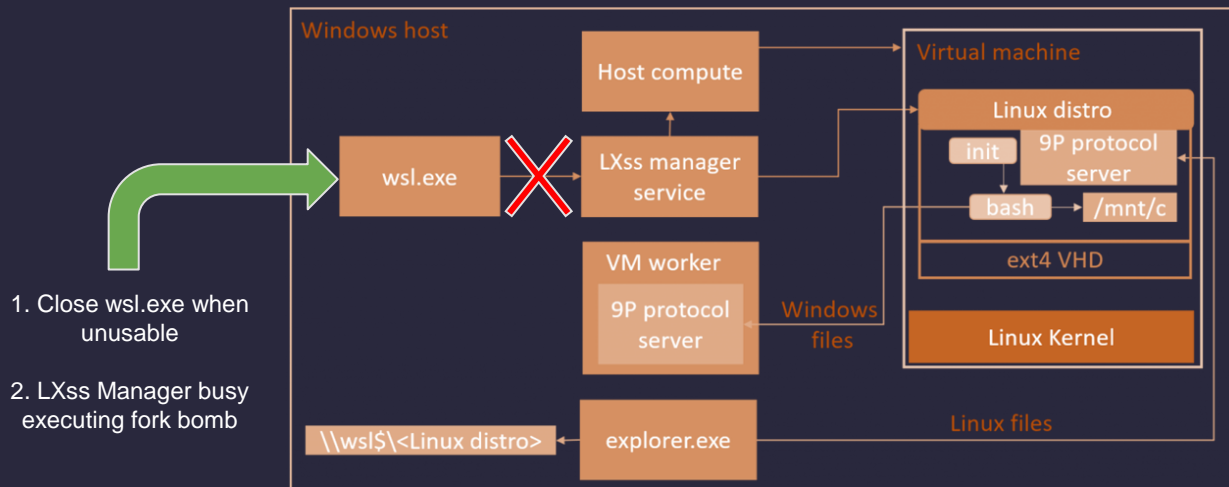
☐ None

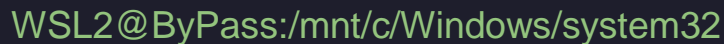
Policy changes

☐ None

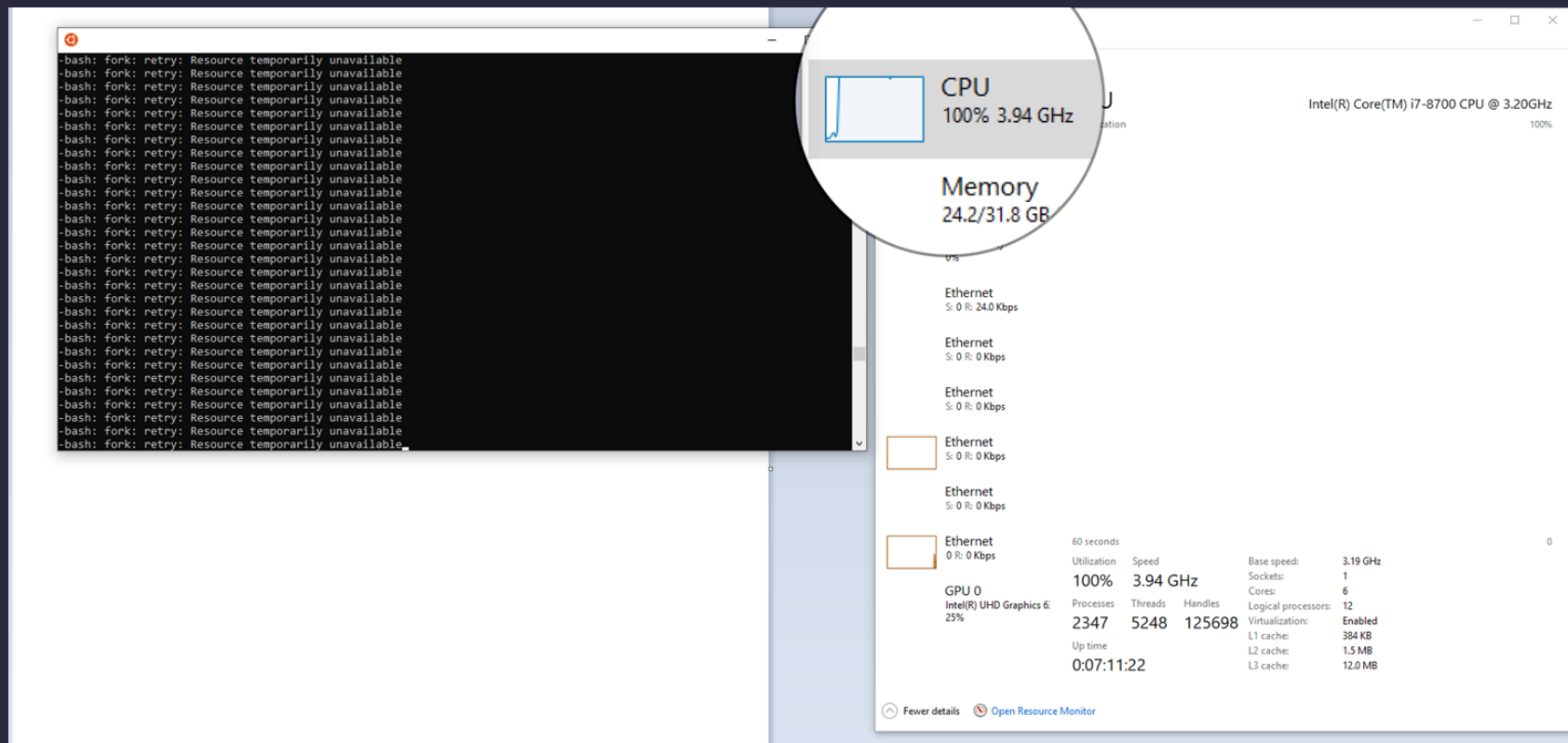
Premium security

☐ None





Resource Exhaustion using WSL2





Identifying Processes from WSL2

- Investigate if Windows fails to identify any processes running inside WSL 2
 - SUCCESS

Default Security

- ☐ Only VMEM and wsl.exe

Policy changes (Process creation)

- ☐ None

Premium security

- ☐ Only wslhost.exe and wsl.exe

Time o...	Process Name	PID	Operation	Path
5:56:59...	wsl.exe	12452	CreateFile	C:\Windows\System32\lxss\wslhost.exe
5:56:59...	wsl.exe	12452	CloseFile	C:\Windows\System32\lxss\wslhost.exe
5:56:59...	wsl.exe	12452	QueryNameInfo...	C:\Windows\System32\lxss\wslhost.exe
5:56:59...	wsl.exe	12452	CreateFile	C:\Windows\System32\lxss\wslhost.exe
5:56:59...	wsl.exe	12452	QueryAllInforma...	C:\Windows\System32\lxss\wslhost.exe
5:56:59...	wsl.exe	12452	CloseFile	C:\Windows\System32\lxss\wslhost.exe
5:56:59...	wsl.exe	12452	Process Create	C:\Windows\system32\lxss\wslhost.exe
5:56:59...	wslhost.exe	11416	Process Start	
5:56:59...	wslhost.exe	11416	Thread Create	
5:56:59...	wsl.exe	12452	QuerySecurityFile	C:\Windows\System32\lxss\wslhost.exe
5:56:59...	wsl.exe	12452	CreateFile	C:\Windows\apppatch\systemmain.sdb
5:56:59...	wsl.exe	12452	QueryBasicInfor...	C:\Windows\apppatch\systemmain.sdb
5:56:59...	wsl.exe	12452	CloseFile	C:\Windows\apppatch\systemmain.sdb
5:56:59...	wsl.exe	12452	QueryBasicInfor...	C:\Windows\System32\lxss\wslhost.exe
5:56:59...	wsl.exe	12452	QuerySecurityFile	C:\Windows\System32\lxss\wslhost.exe
5:56:59...	wsl.exe	12452	CreateFile	C:\Windows\apppatch\systemmain.sdb
5:56:59...	wsl.exe	12452	QueryBasicInfor...	C:\Windows\apppatch\systemmain.sdb
5:56:59...	wsl.exe	12452	CloseFile	C:\Windows\apppatch\systemmain.sdb
5:56:59...	wsl.exe	12452	QueryBasicInfor...	C:\Windows\System32\lxss\wslhost.exe



Abusing Windows Process from WSL2

- Start Windows processes from WSL 2 (Eicar and hacking tool using cmd.exe and powershell.exe)
 - **SUCCESS**

Default Security

- ❑ Detected inside Windows but not inside WSL 2 environment

Policy changes

- ❑ Windows process identified (Event Code 4688) with parent process wsl.exe
- ❑ Powershell command used for Eicar file monitored

Premium security

- ❑ Procmon, Sysmon, Process Explorer can detect the Windows process spawn from WSL 2

```
Context Information:
  DetailSequence= 1
  DetailTotal= 1

  SequenceNumber=23

  UserId=DESKTOP-GM8GA44\WSLTest
  HostName= ConsoleHost
  HostVersion= 5.1.19041.1237
  HostId=f6cb87ae-2f74-4c28-9a37-40fd3678e4d1
  HostApplication=powershell.exe
  EngineVersion= 5.1.19041.1237
  RunspaceId=ec30ccf3-a6ce-4431-987c-ec3577027d33
  PipelineId=6
  ScriptName=
  CommandLine=set-content "XSOIP%@AP[4\PZX54(P^)7CC)7]"$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H" -path "eicar.com"
```



Abusing Windows Process from WSL2 - MimiKatz

- Start mimikatz processes from WSL 2
 - PARTIAL SUCCESS

Wdigest authentication can be easily enabled (assumed that in this case it is)

Default Security

- Mimikatz was detected by the AV after execution (credentials dropped)

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl$\Ubuntu\mnt\wsl\test\mimikatz\x64> .\mimikatz.exe sekurlsa::logonpasswords

.#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## / \ ##   /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(commandline) # sekurlsa::logonpasswords

Authentication Id : 0 ; 67064591 (00000000:03ff530f)
Session           : Service from 0
User Name          : EC27BEDB-B6D7-4168-890C-445D3719F4F1
Domain            : NT VIRTUAL MACHINE
Logon Server       : (null)
Logon Time         : 14-10-2021 20:31:20
```



Unsuccessful Experiments

Evaluating Possibility of Leakage through memory

- **UNSUCCESSFUL**
- Page tables to other VMs inaccessible

Exploiting Environment Variables

- **UNSUCCESSFUL**
- The ENV variables can be shared just by processes spawn by WSL 2 and it cannot be persistent

```
#include <string.h>

int main(int argc, char *argv[]) {
    char* val = (char*)0x555555558040;
    printf("%s", val);
    return 0;
}
```

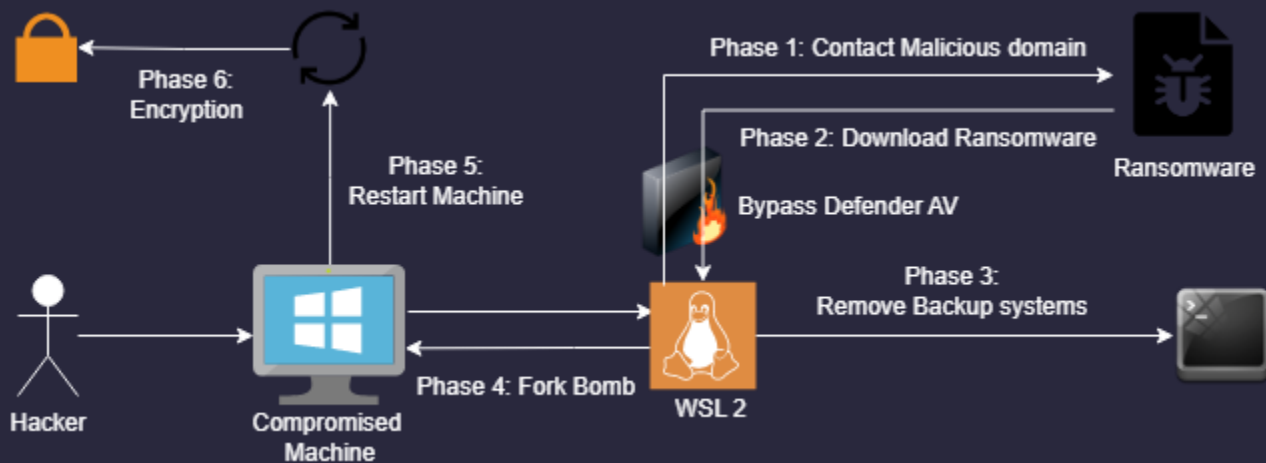
Example code of access attempt absolute addresses.



Putting it all together...

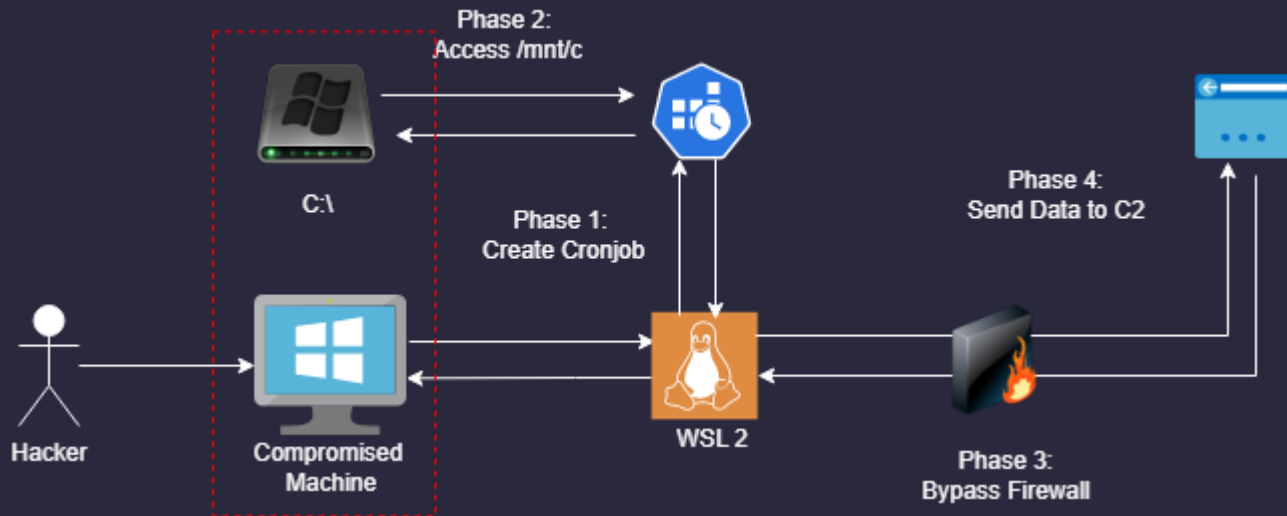


Scenario 1 - Ransomware





Scenario 2 - Data Exfiltration





WSL 2: Secure or liability?

Good

- Bashware-like attacks no longer viable
- Separation of memory
- Changes in Windows are picked up on
- Decent logging inside Windows
- Windows allows in-depth analysis with premium like tooling

Less Good

- Poor design choices like automatic mounts
- Bad documentation
- Network traffic bypasses firewall
- No fully integrated logging
- Blind spot for AV (possibly EDR)

Not insecure, however, needs reconsideration on design decisions and user support



Future Work

- Further research on the impact WSL 2 can have in Windows
 - Enhance the current experiments
 - Create new experiments (9P protocol)
- Expand the security measures to premium solutions and better configurations
- Perform the same experiments on the latest OS and WSL 2 Version
- Verify each scenario with an actual emulation

NOTE: The lack of documentation was extremely time consuming



Thank you!



[in/raresbratean/](#)



[/in/shaunster/](#)