BPF process for windows

<https://github.com/microsoft/ebpf-for-windows>

1. Installing VM -> Hyper-V, not with links

* Install ISO (disk image) windows 11
* Create VM (<https://www.youtube.com/watch?v=QvOmaf1jFbE>) (Development environment 11)

<https://github.com/microsoft/ebpf-for-windows/blob/main/docs/vm-setup.md>

NO DEBUGGING VM (**TODO**)

1. Building bpf for windows

<https://github.com/microsoft/ebpf-for-windows/blob/main/docs/GettingStarted.md>

* Prerequisites: 1-2-3-4-5-6-7-8-9
* Add to path the .exe
* Copy repository (**HERE**) -> CMake

git clone --recurse-submodules <https://github.com/microsoft/ebpf-for-windows.git>

C:\Windows\System32

1. Installing BPF

<https://github.com/microsoft/ebpf-for-windows/blob/main/docs/GettingStarted.md#installing-ebpf-for-windows>

* Test signing mode
* VM installation instructions (no debugging VM)
* Driver test signed **TODO**
  + Signtool needs to be added to the environment variables in Path

(windows kits / 10 / bin / build / x64)

* Test certificate installed **TODO**

1. Installing eBPF into a test VM (MSI)
2. eBPF tutorial

Followed process

1. Install VM -> DONE

<https://github.com/microsoft/ebpf-for-windows/blob/main/docs/vm-setup.md> -> DONE

* Setup VM -> DONE
  + Windows VM on Hyper-V
  + Enable test-signed binaries
* Install all prerequisites

<https://github.com/microsoft/ebpf-for-windows/blob/main/docs/GettingStarted.md>

* Debug VM -> DONE
  + Setting network debugging

<https://learn.microsoft.com/en-us/windows-hardware/drivers/debugger/setting-up-network-debugging-of-a-virtual-machine-host> -> DONE

* + - Hyper-V VM setup (WDK on VM for debug point 4)

<https://learn.microsoft.com/en-us/windows-hardware/drivers/download-the-wdk>

* + - Setting up network debugging of VM

External VM Switch -> IPv4: 192.168.1.7 -> CAMBIA

* + - Setting up VM target

Port -> 50005

Comando -> kdnet 192.168.1.7 50005

“”””””””””””””””

Enabling network debugging on Network debugging is supported by this Microsoft Hypervisor Virtual Machine.

To debug this vm, run the following command on your debugger host machine.

windbg -k net:port=50005,key=3q98fqaazvxgb.9af14h7lvxls.2cxecb7hhc3y1.1id4mb4hwgr6w

Then restart this VM by running shutdown -r -t 0 from this command prompt.

“”””””””””””””””

* + - Install WinDbg

Installed in C:\Program Files (x86)\Windows Kits\10\Debuggers\x64

* + - Execute the command in “””…””” in host

Connection for debugging VM

Immagine che contiene testo, elettronica, schermata, schermo

Descrizione generata automaticamenteDO NOT ABILITATE SECURE BOOT

IF NOT RUNNING -> shutdown -r -t 0 (reboot VM)

* + Enable driver verifier on eBPF drivers -> DONE

“””””””””””””””””””””””””””””””””””””””

Each time, before starting VM, in host command prompt as administrator:

* Secure boot off
* Ipconfig -> then kdnet in VM
* Check advanced session
* cd C:\Program Files (x86)\Windows Kits\10\Debuggers\x64
* windbg -k net:port=50005,key=3q98fqaazvxgb.9af14h7lvxls.2cxecb7hhc3y1.1id4mb4hwgr6w

secure boot still off (it says to turn it on).

“””””””””””””””””””””””””””””””””””””””

After restarting the VM

Immagine che contiene testo, schermata, Carattere, bianco e nero

Descrizione generata automaticamente

OS booted with a kernel debugger attached (?).

* Test signing mode
* Driver test signed
* Test certificate installed

1. Installing eBPF for windows -> DONE

<https://github.com/microsoft/ebpf-for-windows/blob/main/docs/InstallEbpf.md>

Method 1 is simplest.

Immagine che contiene testo, Carattere, schermata, bianco

Descrizione generata automaticamente

1. Tutorial basic

<https://github.com/microsoft/ebpf-for-windows/blob/main/docs/tutorial.md>

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

When creating a new file:

Immagine che contiene testo, schermata, bianco e nero, documento

Descrizione generata automaticamente

But everything works as in the tutorial.

TODO: check maps