Secure Programming

A.A. 2022/2023

Corso di Laurea in Ingegneria delle Telecomnicazioni

K. Environment: Kali Linux on WSL

Paolo Ottolino

Politecnico di Bari

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE

Secure Programming Lab: Course Program

A. Intro Secure Programming: «Who-What-Why-When-Where-How»
B. Building Security in: Buffer Overflow, UAF, Command Inection
C. SwA: Weaknesses, Vulnerabilities, Attacks
D. SwA (Software Assurance): Vulnerabilities and Weaknesses (CVE, OWASP, CWE)
E. Security & Protection: Objectives (CIA), Risks (Likelihood, Impact), Rating Methodologies
F. Security & Protection: Security Indicators, BIA, Protection Techniques (AAA, Listing, Duplication etc.)
G. Architecture and Processes: App Infrastructure, Three-Tiers, Cloud, Containers, Orchestration
H. Architecture and Processes 2: Ciclo di Vita del SW (SDLC), DevSecOps (OWASP DSOMM, NIST SSDF)
I. Free Security Tools: OWASP (ZAP, ESAPI, etc), NIST (SAMATE, SARD etc.)
J. Dynamic Security Test: VA, PT, DAST (cfr. VulnScanTools), WebApp Sec Scan Framework (Arachni, SCNR) :
K. Operating Environment: Kali Linux on WSL
L. Python: Powerful Language for easy creation of hacking tools
M. Exercises: SecureFlag

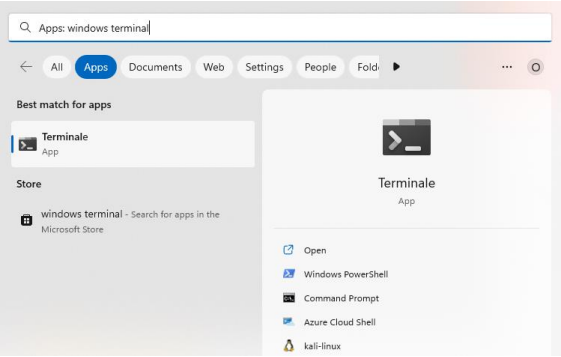
WSL & Kali Linux

1. **WSL**: Windows Subsystem for Linux
2. **Kali Linux on WSL**: Easy Install, Adding Package, Python Script, Network Hacking, Web Hacking
3. **References**: Secure Flag, VSC, SonarQube & Co.

1. WSL: DISM

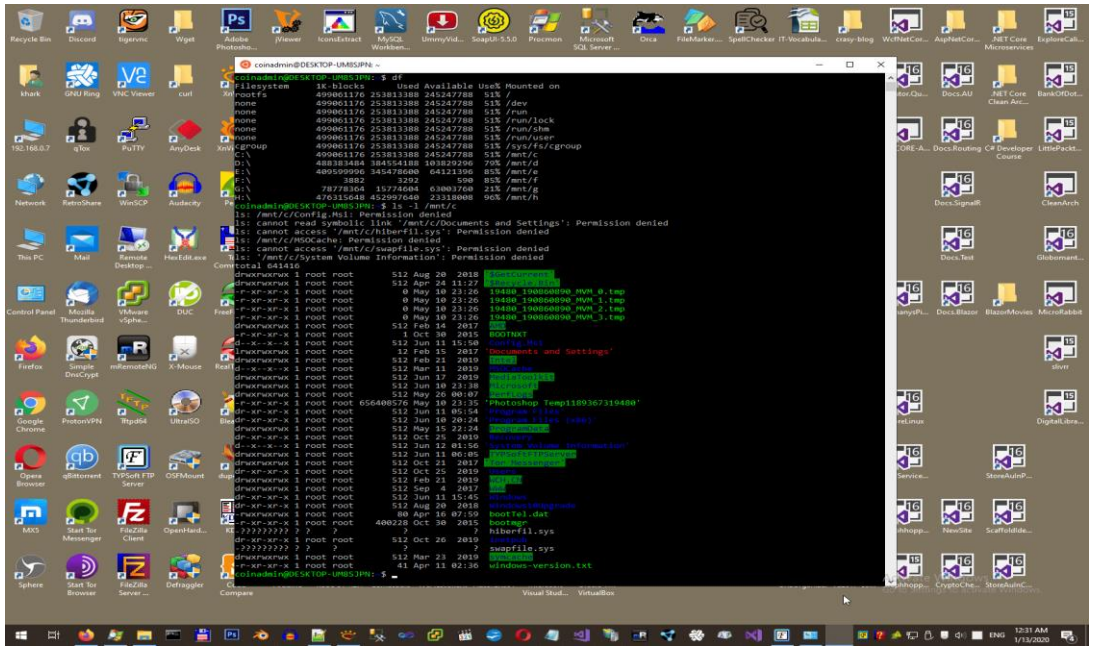
Introduction

Windows Terminal: deve essere attivato



WSL: To be activated with the command "DISM" (Deployment Image Servicing and Management tool).

Manual installation: https://learn.microsoft.com/en-us/windows/wsl/install-manual



1. open powershell, running it "as an Administrator".
2. enable the WSL feature with DISM command
3. issue dism with the following parameters:

dism /online /enable-feature /featurename:Microsoft-Windows-System-Linux /all /norestart

1. WSL: DISM

Explanation

DISM: command-line tool that can be used to service and prepare Windows images.

The images include:

a) current running Windows (/online)

-> WindowsPE (repair Windows Desktop ed, "Preinstallation Environment"): https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11

-> WindowsRE (recovery also for Server >= 2016, "Recovery Environment"): https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-recovery-environment--windows-re--technical-reference?view=windows-11

-> Windows Setup (bootable program that install the Windows O.S.): https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-setup-technical-reference?view=windows-11

b) offline Image (/image)

usage /image:<path-to-the-offline-image-file>

==> since we are configuring the current running windows --> /online

The features are functionality:

-> having a name, to be specified by the switch /featurename:

-> contained in a Package, that is "Windows Foundation Package" by default (otherwise the package should be specified with /PackageName switch).

-> enabling all parent features of the specified feature: /all

1. WSL: DISM

Execution

PowerShell: execute from powershell

```
PS C:\WINDOWS\system32> dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart

Deployment Image Servicing and Management tool
Version: 10.0.22621.1

Image Version: 10.0.22621.819

Enabling feature(s)
[==========================100.0%==========================]
The operation completed successfully.

WSL
-----------------------------------------------------------------------------
```

Now the WSL command should work.

1. WSL: Preparation

Update & Virtual Machine Platform in Windows

Preparing the system: https://www.kali.org/docs/wsl/wsl-preparations/

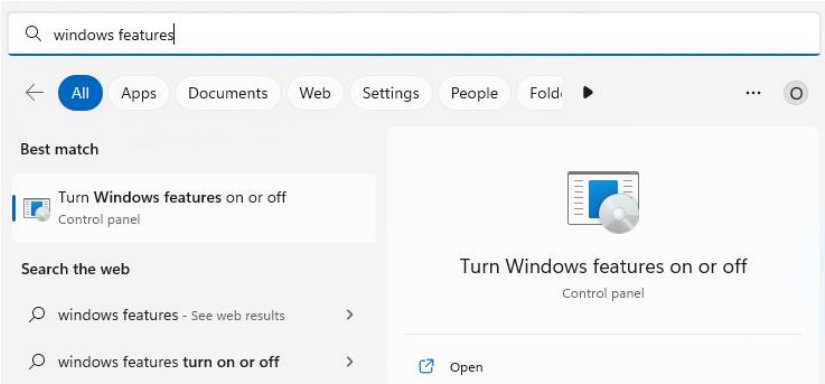
Kali Linux requires WSL2

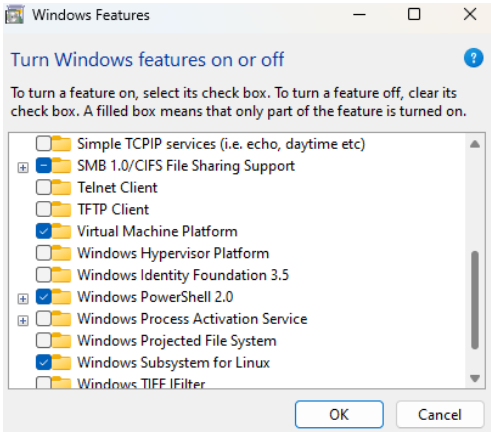
WSL update package: https://learn.microsoft.com/en-us/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package

- Step4: Download the Linux kernel update package: https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi
- Step5: Set WSL 2 as your default version

```
wsl --set-default-version 2
```

Turn On Virtual Machine Platform in Windows: https://support.microsoft.com/en-us/windows/enable-virtualization-on-windows-11-pcs-c5578302-6e43-4b4b-a449-8ced115f58e1





2. Kali Linux on WSL

Installation

Install: Kali Linux is a standard WSL installation package (available from Microsoft Store and installable from shell command)

```
wsl --install -d kali-linux
```

Add Packages: 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/

➔ https://www.kali.org/docs/general-use/metapackages/

1) $ sudo apt update

2) $ sudo apt install -y kali-linux-default

(About 2500 packages)

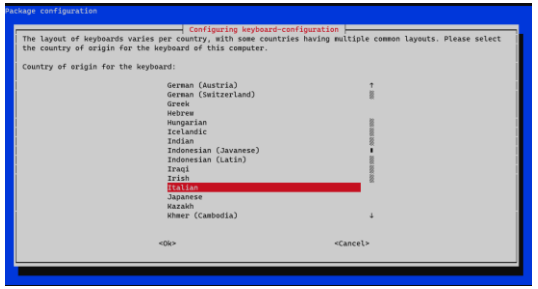
2. Kali Linux on WSL

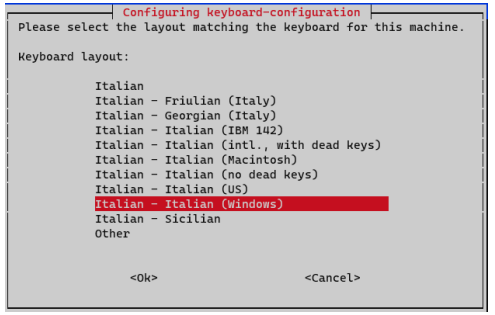
Kali-linux-default

Keyboard Configuration: Other

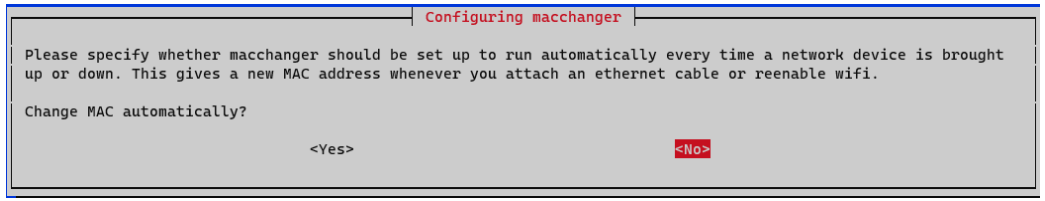
Italian

Italian: Italian (Windows)



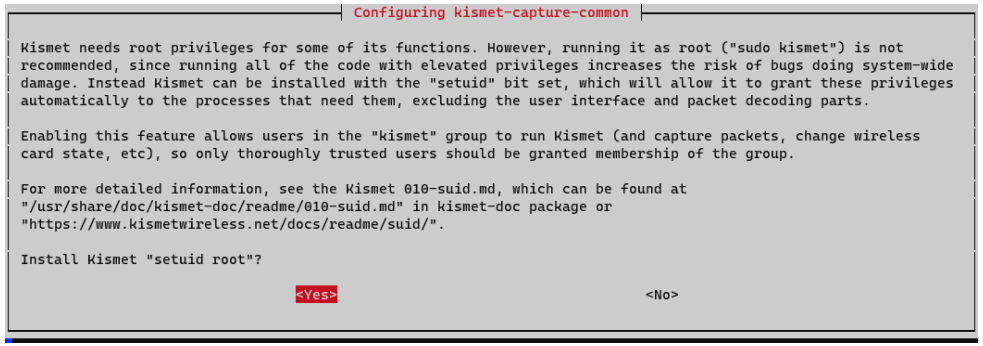


Macchanger: change MAC automatically (No)



Kismet-Capture-Common: setUID (Yes)

(users in the kismet group)



2. Kali Linux on WSL

Kali-linux-default

Wireshark-common: non-superuser able to capture packages

```
Configuring wireshark-common

Dumpcap can be installed in a way that allows members of the "wireshark" system group to capture packets. This
is recommended over the alternative of running Wireshark/Tshark directly as root, because less of the code will
run with elevated privileges.

For more detailed information please see /usr/share/doc/wireshark-common/README.Debian.gz once the package is
installed.

Enabling this feature may be a security risk, so it is disabled by default. If in doubt, it is suggested to
leave it disabled.

Should non-superusers be able to capture packets?

                                <Yes>                                <No>
```

sslh: standalone

```
sslh configuration

sslh can be run either as a service from inetd, or as a standalone server. Each choice has its own benefits.
With only a few connection per day, it is probably better to run sslh from inetd in order to save resources.

On the other hand, with many connections, sslh should run as a standalone server to avoid spawning a new process
for each incoming connection.

Run sslh:

                                from inetd
                                standalone

                                  <Ok>
```

Unpacking:

```
Unpacking libxcb-dri2-0:amd64 (1.15-1) ...
Selecting previously unselected package libxcb-dri3-0:amd64.
Preparing to unpack .../015-libxcb-dri3-0_1.15-1_amd64.deb ...
Unpacking libxcb-dri3-0:amd64 (1.15-1) ...
Selecting previously unselected package libxcb-present0:amd64.
Preparing to unpack .../016-libxcb-present0_1.15-1_amd64.deb ...
Unpacking libxcb-present0:amd64 (1.15-1) ...
Selecting previously unselected package libxcb-sync1:amd64.
Preparing to unpack .../017-libxcb-sync1_1.15-1_amd64.deb ...
Unpacking libxcb-sync1:amd64 (1.15-1) ...
Selecting previously unselected package libxcb-xfixes0:amd64.
Preparing to unpack .../018-libxcb-xfixes0_1.15-1_amd64.deb ...
Unpacking libxcb-xfixes0:amd64 (1.15-1) ...
Selecting previously unselected package libxshmfence1:amd64.
Preparing to unpack .../019-libxshmfence1_1.3-1_amd64.deb ...
Unpacking libxshmfence1:amd64 (1.3-1) ...
Selecting previously unselected package libegl-mesa0:amd64.
Preparing to unpack .../020-libegl-mesa0_22.2.0-1_amd64.deb ...
Unpacking libegl-mesa0:amd64 (22.2.0-1) ...
Selecting previously unselected package libegl1:amd64.
Preparing to unpack .../021-libegl1_1.5.0-1_amd64.deb ...
Unpacking libegl1:amd64 (1.5.0-1) ...
Selecting previously unselected package libxcb-glx0:amd64.
Preparing to unpack .../022-libxcb-glx0_1.15-1_amd64.deb ...
Unpacking libxcb-glx0:amd64 (1.15-1) ...
Selecting previously unselected package libxcb-shm0:amd64.
Preparing to unpack .../023-libxcb-shm0_1.15-1_amd64.deb ...
Unpacking libxcb-shm0:amd64 (1.15-1) ...

Progress: [  3%] [##.......................................................................]
```

2.1 Adding Packages

Win-KeX: Kali Desktop Experience for WSL

Win-Kek: fornisce Kali Desktop Experience in WSL2, e le seguenti funzionalità:

- Sound support
- Unprivileged and Root session support
- Shared clipboard for cut and paste support between Kali Linux and Windows apps
- Multi-session support: root window & non-priv window & seamless sessions concurrently
- Fully compatible with WSLg

Documentazione: sul sito Kali Linux: https://www.kali.org/docs/wsl/win-kex/

Install: dall'interno di Kali Linux WSL, eseguire l'installazione

$ sudo apt update

$ sudo apt install –y kali-win-kex

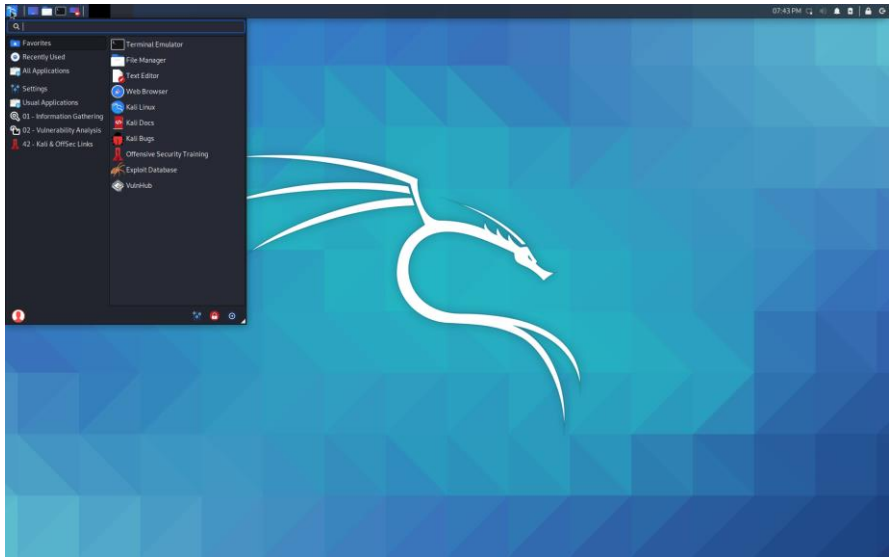
2.1 Adding Packages

Win-KeX: 3 Modi

Window mode: start a Kali Linux desktop in a dedicated window

```
$ kex -win -s
```

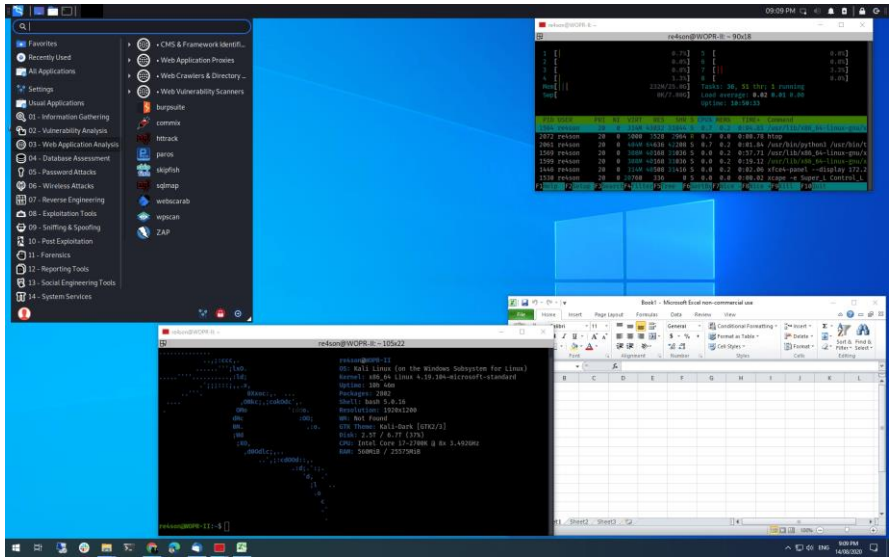
[Win-KeX Win usage documentation]



Enhanced Session Mode: with sound support and arm workaround

```
$ kex -esm -ip -s
```

[Win-KeX ESM usage documentation]



Seamless mode: share the Windows desktop between Windows and Kali apps and menus

```
$ kex -sl -s
```

[Win-KeX SL usage documentation]

2.2 Adding Packages

Python: easy_install e pip

python-pip: già installato con la «kali-linux-default

Test pip: `$ sudo pip install github3.py`

```
[sudo] password for pottol:
Collecting github3.py
  Downloading github3.py-3.2.0-py2.py3-none-any.whl (152 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 152.0/152.0 kB 3.9 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.6.0 in /usr/lib/python3/dist-packages (from github3.py) (2.8.2)
Requirement already satisfied: PyJWT[crypto]>=2.3.0 in /usr/lib/python3/dist-packages (from github3.py) (2.4.0)
Requirement already satisfied: requests>=2.18 in /usr/lib/python3/dist-packages (from github3.py) (2.28.1)
Requirement already satisfied: uritemplate>=3.0.0 in /usr/lib/python3/dist-packages (from github3.py) (4.1.1)
Requirement already satisfied: cryptography>=3.3.1 in /usr/lib/python3/dist-packages (from PyJWT[crypto]>=2.3.0->github3.py)
(3.4.8)
Installing collected packages: github3.py
Successfully installed github3.py-3.2.0
```

Verifica pip: `$ python`

```
Python 3.10.8 (main, Nov  4 2022, 09:21:25) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import github3
>>> exit()
```

2.3 Adding Packages

Python: WingIDE

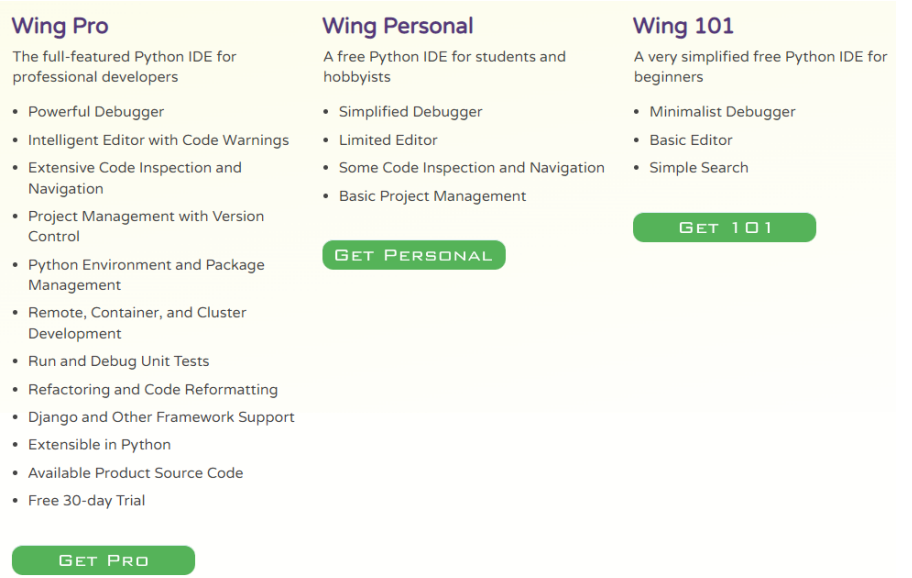
WingIDE: https://wingware.com/

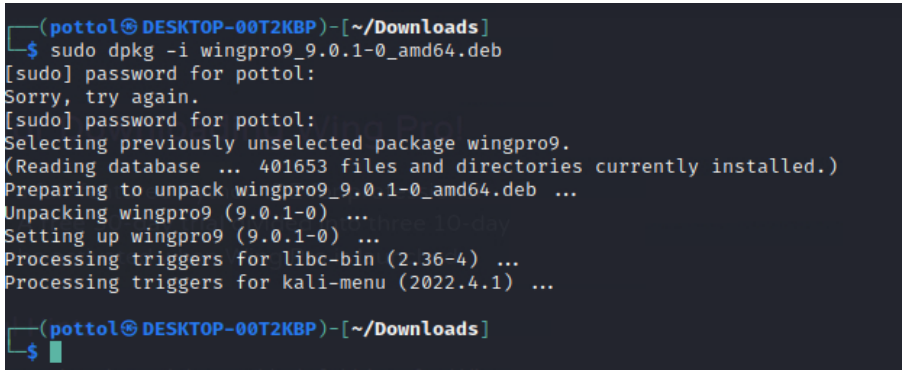
Download WingIDE: scaricare la versione di prova (con tutte le funzionalità)

Free License for WingPro: https://wingware.com/store/free

Installazione:

Sudo dpkg –i wingpro9_9.0.1-0_amd64.deb





```
┌──(pottol㉿DESKTOP-00T2KBP)-[~/Downloads]
└─$ sudo dpkg -i wingpro9_9.0.1-0_amd64.deb
[sudo] password for pottol:
Sorry, try again.
[sudo] password for pottol:
Selecting previously unselected package wingpro9.
(Reading database ... 401653 files and directories currently installed.)
Preparing to unpack wingpro9_9.0.1-0_amd64.deb ...
Unpacking wingpro9 (9.0.1-0) ...
Setting up wingpro9 (9.0.1-0) ...
Processing triggers for libc-bin (2.36-4) ...
Processing triggers for kali-menu (2022.4.1) ...

┌──(pottol㉿DESKTOP-00T2KBP)-[~/Downloads]
└─$
```

3.1 References: Secure Flag (OWASP)

OWASP: Secure Flag ()

SecureFlag Open Platform

The SecureFlag Platform is a training platform created for developers to learn and practice modern secure coding techniques through hands-on exercises. The platform helps develop secure coding skills through real-world challenges to ensure knowledge acquired during the course can be confidently and continuously applied in the real world.

The SecureFlag Open Platform is an OWASP Project and includes an SDK and developer tools to create Labs for the SecureFlag platform.

3.1a References: Secure Flag (OWASP)

Secure Flag: SDK Installation

The command-line tool sfsdk provides an SDK to develop new SecureFlag exercises. The tool allows the development and submission of new Labs for the SecureFlag platform.

Prerequisites

The following prerequisites are required to develop Labs for the SecureFlag platform.

- A computer running recent versions of Linux or MacOS. Kali Linux on WSL

```
Linux DESKTOP-00T2KBP 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021 x86_64 GNU/Linux
```

- Docker , make sure your user is part of the Docker group.

```
$ sudo usermod -a -G docker pottol
```

- Python version >= 3.6 and pip .

```
$ python -version Python 3.10.8
```

- An RDP client of your choice.

```
$ sudo apt install rdesktop
```

3.1b References: Secure Flag (OWASP)

Secure Flag: SDK Installation

Installation

- Install the SDK using pip3:

```
$ pip3 install --user sfsdk
```

- sfsdk command is not available in your shell, you'll need to add pip's bin path to your $PATH environment variable:

```
$ echo 'PATH="$PATH:$HOME/.local/bin"' >> ~/.profile
```

- ~/.profile must be reloaded to be used in the current shell:

```
$ source ~/.profile
```

Workspace

Developing new Labs with sfsdk will populate the ~/sf folder using the structure shown below:

```
~/sf/
  login.yml            # YAML with credentials
  images.yml           # YAML with images info
  img/                 # Directory containing images
    org-java-vapp/     # Build directory
      Dockerfile       # Dockerfile
      fs/              # Other files
```

3.1c References: Secure Flag (OWASP)

Secure Flag: SDK Installation

The command-line tool sfsdk provides an SDK to develop new SecureFlag exercises. The tool allows the development and submission of new Labs for the SecureFlag platform.

Installation

- Install the SDK using pip3:

```
$ pip3 install --user sfsdk
```

- sfsdk command is not available in your shell, you'll need to add pip's bin path to your $PATH environment variable:

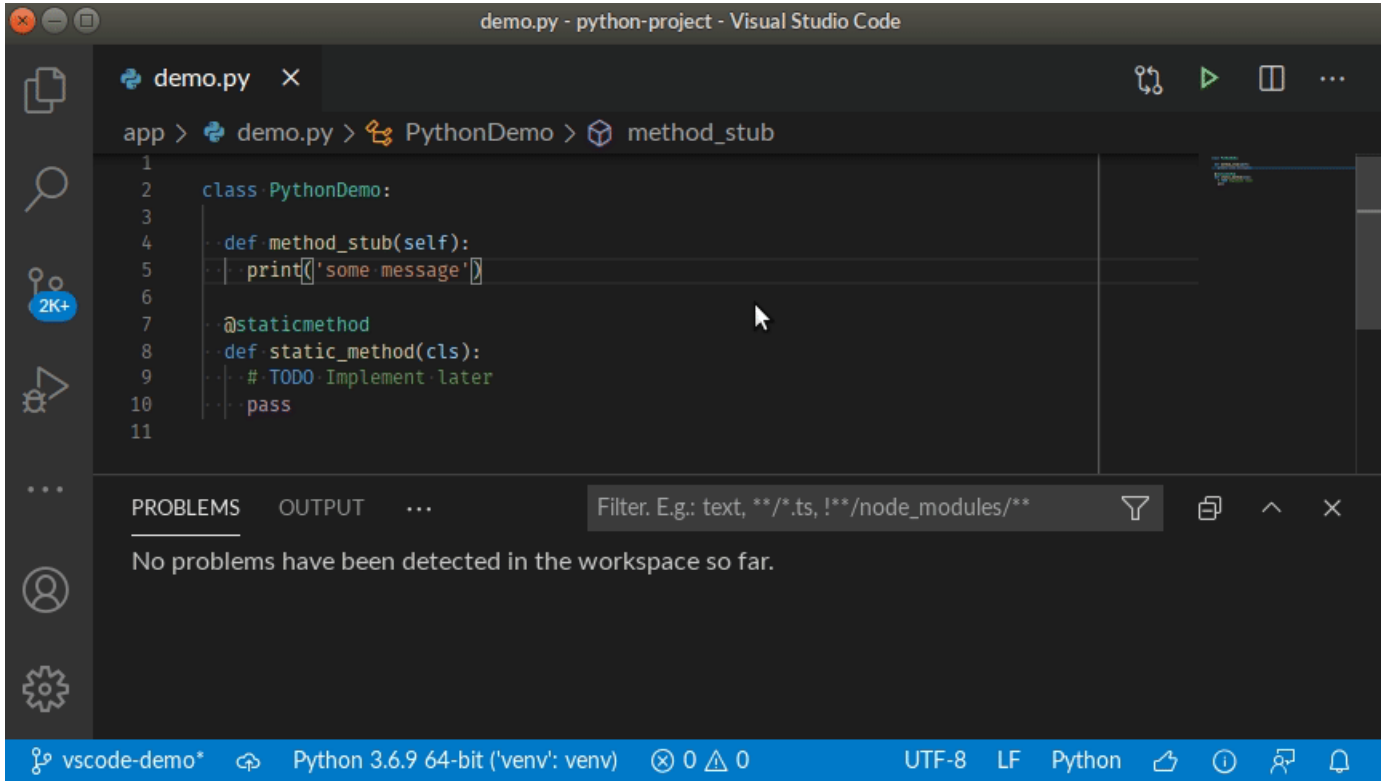
```
$ echo 'PATH="$PATH:$HOME/.local/bin"' >> ~/.profile
```

- ~/.profile must be reloaded to be used in the current shell:

```
$ source ~/.profile
```

3.2 References: VSC

Visual Studio Code



1. VSC: Visual Studio Code: https://code.visualstudio.com/download
2. Debug a C++ project in VS Code: https://www.youtube.com/watch?v=G9gnSGKYIg4
3. VS Code Debugging: https://www.youtube.com/watch?v=6cOsxaNC06c
4. Visual Studio Code Tutorial for Beginners – Introduction: https://www.youtube.com/watch?v=VqCgcpAypFQ

3.3 References: SonarQube & Co.

SonarLint, SonarQube, SonarCloud

1. **SonarQube**: https://docs.sonarqube.org/latest/ self-managed, automatic code review tool

2. **SonarLint**: https://www.sonarsource.com/products/sonarlint/ free IDE plugin available to install from your IDE marketplace (e.g. SonarLint for VSC: while coding the source file, seeing issues reported by SonarLint, highlighted in the code, and also listed in the 'Problems' panel.)

3. **SonarCloud**: https://docs.sonarcloud.io/ cloud-based code analysis service