

2 de junio de 2023

Desarrollo de Entorno Profesional para la Ciberseguridad

Realizado por:

Álvaro González Sayago en CFGS 2ASIR



HOJA RESUMEN – PROYECTO

Título del proyecto: Desarrollo de un Entorno Profesional para la Ciberseguridad	
Autor: Álvaro González Sayago	Fecha: 02/06/2023
Tutor: López Fernández, Olga María	
Titulación: CFGS 2 Administración de Sistemas Informáticos en Red	
Palabras clave: Ciberseguridad, entorno profesional, kitty, bspwm, zsh, parrot, arch, black arch, pentesting, picom, polybar, sxhdk.	
Resumen del proyecto: Crear un entorno profesional orientado a la ciberseguridad completamente desde cero con una distribución base, pudiendo personalizar por completo tanto el diseño gráfico de nuestro entorno como los atajos de teclado para una utilización lo más cómoda posible.	

ÍNDICE

HOJA RESUMEN – PROYECTO.....	2
Introducción	6
Introducción a la memoria.	6
Descripción.	6
Objetivos generales y beneficios.....	6
Motivaciones personales.....	7
Estructura de la memoria.	7
Estudio de viabilidad.....	8
Introducción.....	8
Requisitos del sistema.....	8
Requisitos.....	8
Priorización de los requisitos.....	8
Alternativas y selección de la solución.....	9
Alternativa 1. KITTY sobre Arch Linux.	9
Alternativa 2. BSPWM sobre Parrot.....	9
Conclusiones.....	9
Planificación del proyecto	10
Planificación temporal.....	10
Evaluación de riesgos.....	11
Posibles riesgos.....	11
Plan de contingencia.	11
Presupuesto.....	11
Estimación de coste material	11
Resumen y análisis coste-beneficio.	11
Conclusiones	11
Beneficios.....	11
Inconvenientes	11
Análisis de software.....	12
Introducción.....	12
Software utilizado.....	12
Pacstrap	12
AwesomeWM	12
PICOM	13
ZSH	13

LSD	14
BAT	14
Powerlevel10k	15
FZF	15
NeoVim	16
Mdcats	16
NeoFetch	17
NvChad	17
BurpSuite	18
Nmap	18
RustScan	19
Plocate	19
Impacket	19
Responder	20
WhatWeb	20
Wfuzz	21
Evil-WinRM	21
MetaSploit	22
ExploitDB	22
Personalización del entorno.....	23
Introducción.....	23
Personalización	23
Manual de atajos & comodidades	26
Introducción.....	26
Atajos y comodidades.....	26
Pruebas generales.....	27
Introducción.....	27
Pruebas de uso.....	27
Resultados obtenidos.....	29
Conclusiones	29
Webgrafía.....	30
Conclusión final.....	31
Fin del documento	31

Introducción

Introducción a la memoria.

En esta memoria recopilare toda la documentación necesaria para explicar lo mejor y más completo posible mi proyecto, además, estará dividida por las secciones que considero son más importantes para su posterior explicación.

La documentación constará de información tal como los objetivos del proyecto, costes, requisitos, ventajas, manuales de instalación/configuración, planificación, webgrafía, glosario de palabras etc.

Descripción.

Este proyecto se basa en la creación de un entorno profesional de ciberseguridad, configurado completamente desde cero, usará paquetes y software de licencia gratuita, además de una distribución base y limpia.

De los paquetes de los que estará compuesto el proyecto, en su mayoría a todos hay que instalarlos y configurarlos, siempre se aportará la fuente de instalación.

Además de la configuración de paquetes, con este proyecto conseguiremos un sistema fácil de utilizar y muy personalizable.

Recaltar que la inspiración de este proyecto es gracias a S4vitar:

<https://www.youtube.com/@s4vitar>

<https://www.twitch.tv/s4vitaar>

<https://s4vitar.github.io>

Objetivos generales y beneficios.

Los objetivos generales de este proyecto son:

- Aprender sobre los diferentes paquetes que se usen.
- Priorizar la comodidad del usuario.
- Obtener una interfaz completamente personalizable.
- Conseguir atajos de teclado modificables.
- Acabar con un sistema apto para la realización de técnicas de pentesting y ciberseguridad.
- Uso de paquetes de coste cero mediante licencias gratuitas o de código abierto.

Motivaciones personales.

Es un proyecto que me llama muchísimo la atención ya que es conseguir un entorno completamente diferente al que tenemos al inicio del proyecto además de tener muchas nuevas funcionalidades.

Me llamó bastante la atención desde el principio por estar basado en Linux y tener tantos paquetes distintos con muchas utilidades diferentes, cada uno con su propia configuración.

Me lo propongo como un reto a mí mismo ya que con tanto trabajo en una sola distribución, es muy complicado que no aparezcan varios errores por el camino, ya sean versiones, configuraciones, código etc.

Estructura de la memoria.

La estructura intentaré que sea lo más sencilla posible. Estará dividida en 12 bloques, que serán los siguientes:

- Introducción. (Este mismo bloque)
- Estudio de Viabilidad. (Bloque a continuación)
- Análisis.
- Personalización del entorno.
- Manual de atajos & comodidades.
- Pruebas generales
- Webgrafía
- Conclusión final
- Fin del documento

Excepción:

- Manual de Instalación & Configuración. (Documento a parte)

Cada bloque es posible que esté dividido en apartados, y estos mismos apartados en subapartados.

Cada bloque se separará del anterior mediante un salto de página (si son lo suficientemente grandes), para que el documento quede lo más limpio posible.

Entre apartado y apartado habrá un espacio mínimo para aumentar la claridad del documento.

Estudio de viabilidad

Introducción.

En este bloque documentaré de la mejor forma posible la viabilidad del proyecto, en aspectos generales, hablaré sobre requisitos, alternativas, planificación de recursos, riesgos etc. Será el apartado más grande y por lo tanto el que más organice el proyecto.

Requisitos del sistema

Requisitos.

Como ya dije anteriormente, los requisitos serán prácticamente los mismos que las distribuciones base a utilizar. En este caso, la distribución base será Arch Linux:

	MÍNIMO	RECOMENDADO
ARQUITECTURA	x86	x64
RAM	512MB	2GB
ALMACENAMIENTO	1GB	20GB

Como recomendación personal, ya que se usarán bastantes paquetes grandes y ficheros de configuración, además de una instalación de interfaz gráfica desde cero, utilizaría la configuración de requisitos recomendada.

Priorización de los requisitos.

Los requisitos principales serían el almacenamiento y la RAM, aunque lo recomendado sean 2GB yo intentaría conseguir 4GB o incluso 6. Además para un buen sistema operativo, intentaría también llegar a los 40GB de almacenamiento para asegurar que no acabemos con falta de espacio.

Alternativas y selección de la solución.

Este proyecto se puede abarcar de dos grandes formas (son las que yo he planteado/encontrado, pero realmente puede hacerse de infinitas formas distintas y con cualquier distribución).

Alternativa 1. KITTY sobre Arch Linux.

La primera alternativa es utilizar KITTY (emulador de terminal acelerado por GPU) y (Administrador de ventanas estilo Windows, pero para Linux) sobre Arch Linux. Utilizando los paquetes:

- Pacstrap
- Bootloader
- AUR
- VMware-tools
- AwesomeWM
- PICOM
- ZSH
- LSD
- BAT
- Powerlevel10k
- FZF
- Neovim
- Mdcats



Alternativa 2. BSPWM sobre Parrot.

La segunda y última alternativa, pero no menos importante, es utilizar BSPWM sobre Parrot. Utilizando los paquetes:

- SXHKD
- Polybar
- ROFI
- PICOM
- Slimlock
- Powerlevel10k
- BAT
- LSD
- FZF
- ZSH
- Ranger
- Oh My Tmux
- fastTCPscan



Conclusiones.

Las dos alternativas son aceptables y muy amplias, en mi caso utilizaré la primera ya que considero que puede dar más trabajo y es más interesante. Si el proyecto al final resultara no ser lo suficientemente grande, haría también la segunda alternativa (saldría en este documento final más adelante).

Planificación del proyecto

En este apartado explicaré cual será la estructura a seguir a la hora de crear el proyecto, intentaré dividir la planificación en pasos lo más pequeños posibles para que esté recopilado todo lo importante.

Planificación temporal.

La planificación del proyecto será la siguiente:

1. Descarga de VMware.
2. Descarga de la ISO de Arch.
3. Creación de la máquina virtual con Arch.
4. Definir las particiones con cfdisk.
5. Definir partición SWAP.
6. Formatear las particiones.
7. Instalación de paquetes necesarios con PACSTRAP.
8. Crear archivo fstab.
9. Instalación del GRUB.
10. Habilitamos AUR.
11. Instalación de los repositorios de Black Arch en Arch Linux.
12. Instalación de la interfaz gráfica.
13. Instalación de KITTY, VMware tools, Firefox, AwesomeWM y PICOM.
14. Instalación de plugins para ZSH, LSD y BAT.
15. Configuración KITTY y PICOM.
16. Instalación y configuración de Powerlevel10k, FZF, Neovim.
17. Instalación de herramientas de pentesting.
18. Configuración de teclas.
19. Definición de los nuevos atajos personalizados.
20. Instalación de MDCAT para visualizar archivos Markdown.

Remarcar que cada uno de los paquetes utilizados tendrá más adelante su apartado de explicación sobre su funcionamiento, configuración y enlaces/comandos de interés.

Evaluación de riesgos.

En esta sección nombraré los errores más grandes y comunes que nos pueden ocurrir, así como la forma de evitarlos.

Posibles riesgos

Como es un proceso de varias horas sobre un mismo sistema instalando y haciendo diversas configuraciones, y varias de ellas tienen que ver directamente con el sistema, es bastante probable que podamos tener algún fallo crítico en algún momento del proceso.

Plan de contingencia.

Como recomendación haría snapshots en la máquina virtual después de cada instalación/configuración que se realice correctamente, así, si cometemos algún fallo podremos volver al paso anterior sin problema.

Además, intentaría no tener más de 3 snapshots activas al mismo tiempo para evitar almacenamiento innecesario. Para ello podemos ir borrando snapshots antiguas a medida que vamos progresando en el proyecto.

Una vez tengamos el SO final, haremos una última snapshot con nuestro sistema funcionando a la perfección.

Presupuesto

En este apartado presentaré el coste material además del análisis coste-beneficio del proyecto.

Estimación de coste material

Este proyecto nos costará un total de 0€. Todos los paquetes, distribuciones, imágenes y configuraciones utilizadas son de código abierto o de licencia gratuita para que cualquiera pueda tenerlo.

Resumen y análisis coste-beneficio.

Al ser un entorno profesional de trabajo, con un coste de 0€ podríamos sacar una rentabilidad infinita, depende de nuestro trabajo.

Conclusiones

Beneficios

- Obtención de un entorno profesional de pentesting.
- Entorno 100% personalizable.
- Totalmente gratuito y en su mayoría de código abierto.
- Atajos configurables.

Inconvenientes

- Largo de hacer.
- Muchos paquetes distintos.
- Muchas configuraciones distintas.
- Muchos pasos a seguir.

Análisis de software

Introducción

En este bloque, nombraré todo el software utilizado para el proyecto. Obviaré lo más básico como VMware pero documentaré las utilidades que considere más raras. Además, insertaré a ser posible su página oficial o en su defecto su página de GitHub.

Software utilizado.

Sección donde documentaré de la forma más completa posible cada utilidad del proyecto.

Pacstrap

Pacstrap es un instalador de paquetes que viene incluido en la distribución de Arch Linux de base. Similar a apt-get o apt.

Links de interés:

Manual de Arch: <https://man.archlinux.org/man/pacstrap.8>

GitHub: <https://github.com/archlinux/arch-install-scripts/blob/master/pacstrap.in>

AwesomeWM

También llamado Awesome Window Manager, basado en C y en LUA. Diseñado para mejorar el entorno gráfico y pensado para desarrolladores.



Links de Interés:

Página Oficial: <https://awesomewm.org>

GitHub: <https://github.com/awesomeWM/awesome>

Vídeo de Intro: https://www.youtube.com/watch?v=qKtit_B7Keo

PICOM

Picom es un compositor para Xorg. Muy útil al usarlo con administradores de ventanas (Como AwesomeWM). Nos permite aplicar sombras, transparencias, efectos 3D etc.

Links de Interés:

Manual de Arch: <https://wiki.archlinux.org/title/picom>

GitHub: <https://github.com/yshui/picom>

Vídeo de intro: <https://www.youtube.com/watch?v=t6Klg7CvUxA>

ZSH

También llamado. Es un terminal muy potente para sistemas operativos basados en Unix. También es actualmente utilizado en Kali Linux. OhMyZsh es totalmente opcional pero mejora el uso y configuración de ZSH.



Links de Interés:

Manual de Arch: <https://wiki.archlinux.org/title/picom>

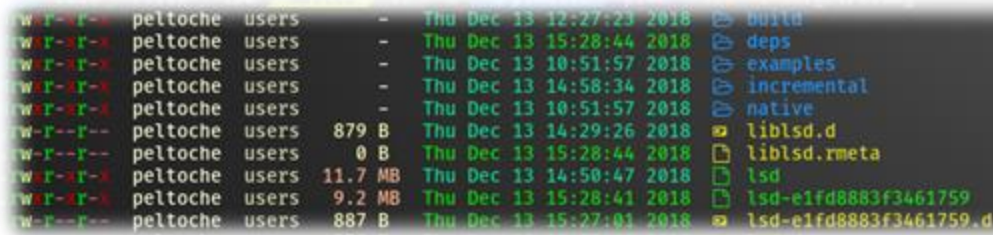
GitHub OhMyZSH: <https://github.com/ohmyzsh/ohmyzsh>

GitHub ZSH: <https://github.com/zsh-users/zsh>

Vídeo de intro: <https://www.youtube.com/watch?v=ADytC9a2g2Y>

LSD

También llamado LSDeluxe. Es una mejora del comando LS que nos permite colorear el output, añadir iconos y hacerlo mucho más visual.



```
rw-r--r-- peltoche users - Thu Dec 13 12:27:23 2018 d built
rw-r--r-- peltoche users - Thu Dec 13 15:28:44 2018 d deps
rw-r--r-- peltoche users - Thu Dec 13 10:51:57 2018 d examples
rw-r--r-- peltoche users - Thu Dec 13 14:58:34 2018 d incremental
rw-r--r-- peltoche users - Thu Dec 13 10:51:57 2018 d native
rw-r--r-- peltoche users 879 B Thu Dec 13 14:29:26 2018 d liblsd.d
rw-r--r-- peltoche users 0 B Thu Dec 13 15:28:44 2018 d liblsd.rmeta
rw-r--r-- peltoche users 11.7 MB Thu Dec 13 14:50:47 2018 d lsd
rw-r--r-- peltoche users 9.2 MB Thu Dec 13 15:28:41 2018 d lsd-e1fd8883f3461759
rw-r--r-- peltoche users 887 B Thu Dec 13 15:27:01 2018 d lsd-e1fd8883f3461759.d
```

Links de Interés:

GitHub: <https://github.com/lsd-rs/lsd>

Vídeo de intro: <https://www.youtube.com/watch?v=YQSGg9oWSTo>

BAT

Una mejora del comando CAT. Al igual que LSD con LS, BAT nos permite ver el output de un CAT pero con colores e identificación de código.



Links de Interés:

GitHub: <https://github.com/sharkdp/bat>

Vídeo de intro: <https://www.youtube.com/watch?v=myg9zTf5aC8>

Powerlevel10k

Powerlevel10k es simplemente un tema para la shell ZSH. Es muy configurable fácilmente y nos permite también añadir iconos en el prompt.



Links de Interés:

GitHub: <https://github.com/romkatv/powerlevel10k>

Vídeo de intro: <https://www.youtube.com/watch?v=KNLNyX9V1L8>

FZF

También llamado Fuzzy Finder. Es un buscador rápido y optimizado de archivos y carpetas mediante línea de comandos que nos muestra los archivos que va encontrando en tiempo real.



Links de Interés:

GitHub: <https://github.com/junegunn/fzf>

Vídeo de intro: <https://www.youtube.com/watch?v=J0QuhUThM5E>

NeoVim

Como su propio nombre indica, es VIM pero con interfaz y utilidades mejoradas. Utiliza un 30% menos de código que el VIM original.



Links de Interés:

- Página Oficial: <https://neovim.io>
- GitHub: <https://github.com/neovim/neovim>
- Vídeo de intro: <https://www.youtube.com/watch?v=x1TEl0Mxmkl>

Mdcat

Mdcat nos permite visualizar por consola archivos Markdown de forma sencilla.

Links de Interés:

- GitHub: <https://github.com/swsnr/mdcat>
- Manual de Arch: <https://man.archlinux.org/man/community/mdcat/mdcat.1.en>
- Vídeo de Intro: <https://www.youtube.com/watch?v=HvAhAytFRdY>

NeoFetch

Utilidad para buscar información del sistema de forma sencilla desde la línea de comandos.



Links de Interés:

- GitHub: <https://github.com/dylananaraps/neofetch>
- Manual de Arch: <https://archlinux.org/packages/community/any/neofetch/>
- Vídeo de Intro: <https://www.youtube.com/watch?v=btyrzunEwII>

NvChad

Utilidad que nos permite mejorar NeoVim e incluso hacerlo muy parecido a VSCode.



Links de Interés:

- GitHub: <https://github.com/NvChad/NvChad>
- Página oficial: <https://nvchad.com>
- Vídeo de Intro: <https://www.youtube.com/watch?v=WaSrieHjG6o>

BurpSuite

Plataforma gráfica para testear la seguridad de aplicaciones Web. Tiene versión Professional y Community siendo esta última gratuita.

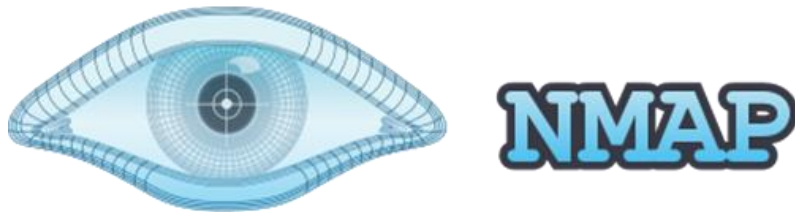


Links de Interés:

GitHub: <https://github.com/thehackingsage/burpsuite>
Página oficial: <https://portswigger.net/burp>
Vídeo de Intro: <https://www.youtube.com/watch?v=nXm324qSfRA>

Nmap

Escáner de puertos por defecto, el más usado.



Links de Interés:

GitHub: <https://github.com/nmap/nmap>
Página Oficial: <https://nmap.org>
Vídeo de Intro: <https://www.youtube.com/watch?v=blZXOqJwZt4>

RustScan

Un escáner de puertos modernos, equiparable a Nmap.



Links de Interés:

GitHub: <https://github.com/RustScan/RustScan>

Página Arch: https://archlinux.org/packages/community/x86_64/rustscan/

Vídeo de Intro: <https://www.youtube.com/watch?v=a6j8D6LhByM>

Plocate

Herramienta para localizar archivos en todo el sistema rápidamente.

Links de Interés:

GitHub: <https://github.com/Aetf/plocate>

Página Arch: <https://man.archlinux.org/man/plocate.1>

Vídeo de Intro: <https://www.youtube.com/watch?v=GnRvtgry1-M>

Impacket

Una serie de herramientas creadas en Python para trabajar con protocolos de red.

Links de Interés:

GitHub: <https://github.com/fortra/impacket>

Página Arch: <https://archlinux.org/packages/community/any/impacket/>

Vídeo de Intro: https://www.youtube.com/watch?v=Ule_iBhhawQ

Responder

Herramienta para conseguir rápidamente credenciales y posibles accesos remotos.



Links de Interés:

GitHub: <https://github.com/lgandx/Responder>
 Página Arch: <https://aur.archlinux.org/packages/responder>
 Vídeo de Intro: <https://www.youtube.com/watch?v=8VtzplbvX2U>

WhatWeb

Herramienta para extraer datos e identificar una Web en específico.



Links de Interés:

GitHub: <https://github.com/urbanadventurer/WhatWeb>
 Página Arch: <https://aur.archlinux.org/packages/whatweb>
 Vídeo de Intro: <https://www.youtube.com/watch?v=7Uo8bVL10xI>

Wfuzz

Utilidad para escanear estructuras de directorios mediante fuerza bruta. Sirve para lo mismo que GoBuster pero de diferente forma. También se puede usar ffuf, dirb, dirbuster y dirsearch. Todos sirven para lo mismo pero depende de la comodidad del usuario.



Links de Interés:

GitHub: <https://github.com/xmendez/wfuzz>
Página Arch: <https://aur.archlinux.org/packages/wfuzz>
Vídeo de Intro: <https://www.youtube.com/watch?v=Alj3pKck9Y>

Evil-WinRM

Usado en sistemas Windows con el servicio WinRM activo, normalmente funcionando por el puerto 5985. Creado en ruby.



Links de Interés:

GitHub: <https://github.com/Hackplayers/evil-winrm>
Página Arch: <https://aur.archlinux.org/packages/ruby-evil-winrm>
Vídeo de Intro: <https://www.youtube.com/watch?v=-6j3P0VB8ak>

MetaSploit

Framework que reúne vulnerabilidades, así como CVEs para su uso en pentesting. No es recomendado usarlo sin mirar directamente los scripts.



Links de Interés:

GitHub:	https://github.com/rapid7/metasploit-framework
Página Arch:	https://wiki.archlinux.org/title/Metasploit_Framework
Página Oficial:	https://www.metasploit.com
Vídeo de Intro:	https://www.youtube.com/watch?v=ZOFsQi1NVsM

ExploitDB

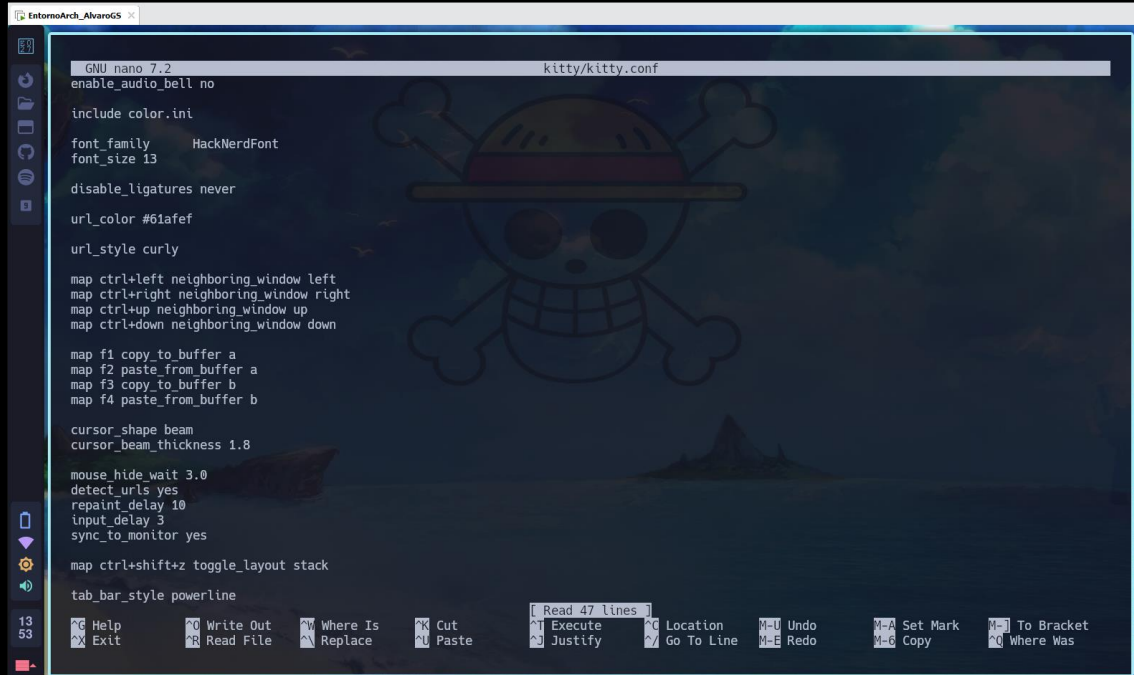
Base de datos con todos los CVEs registrados y como explotarlos.



Links de Interés:

GitLab:	https://gitlab.com/exploit-database/exploitdb
Página Oficial:	https://www.exploit-db.com
Vídeo de Intro:	https://www.youtube.com/watch?v=ZeBwhK2hQr4

Fichero “kitty.conf”. Sirve para modificar la apariencia de nuestro terminal, asignarle atajos de teclado y funcionalidad a ciertas teclas o combinaciones.



```
GNU nano 7.2 kitty/kitty.conf
enable_audio_bell no

include color.ini

font_family HackNerdFont
font_size 13

disable_ligatures never

url_color #61afef
url_style curly

map ctrl+left neighboring_window left
map ctrl+right neighboring_window right
map ctrl+up neighboring_window up
map ctrl+down neighboring_window down

map f1 copy_to_buffer a
map f2 paste_from_buffer a
map f3 copy_to_buffer b
map f4 paste_from_buffer b

cursor_shape beam
cursor_beam_thickness 1.8

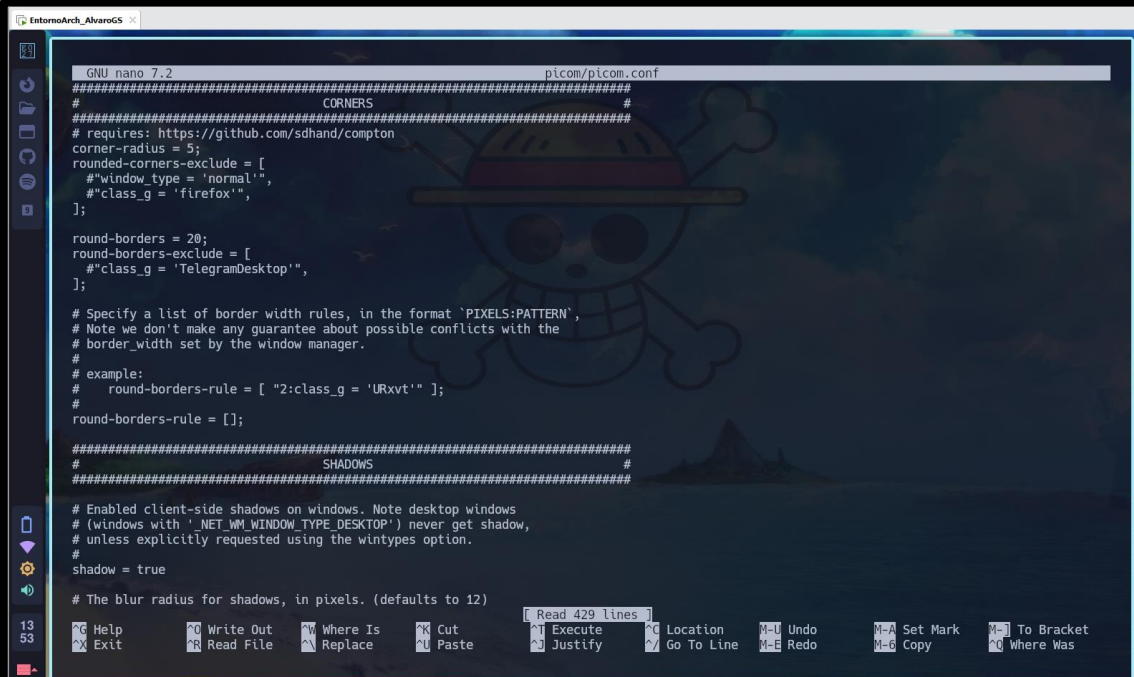
mouse_hide_wait 3.0
detect_urls yes
repaint_delay 10
input_delay 3
sync_to_monitor yes

map ctrl+shift+z toggle_layout stack

tab_bar_style powerline

[ Read 47 lines ]
[ Help ] [ Write Out ] [ Where Is ] [ Cut ] [ Execute ] [ Location ] [ M-U ] [ M-A ] [ M-J ]
[ Exit ] [ Read File ] [ Replace ] [ Paste ] [ Justify ] [ Go To Line ] [ M-E ] [ M-C ] [ M-Q ]
```

Fichero “picom.conf”. Utilizado para efectos de transparencia y opacidad, sombras de ventanas, animaciones y efectos de bordeado en las ventanas.



```
GNU nano 7.2 picom/picom.conf
#####
# CORNERS #
#####
# requires: https://github.com/sdhand/compton
corner-radius = 5;
rounded-corners-exclude = [
# "window_type = 'normal'",
# "class_g = 'firefox'",
];

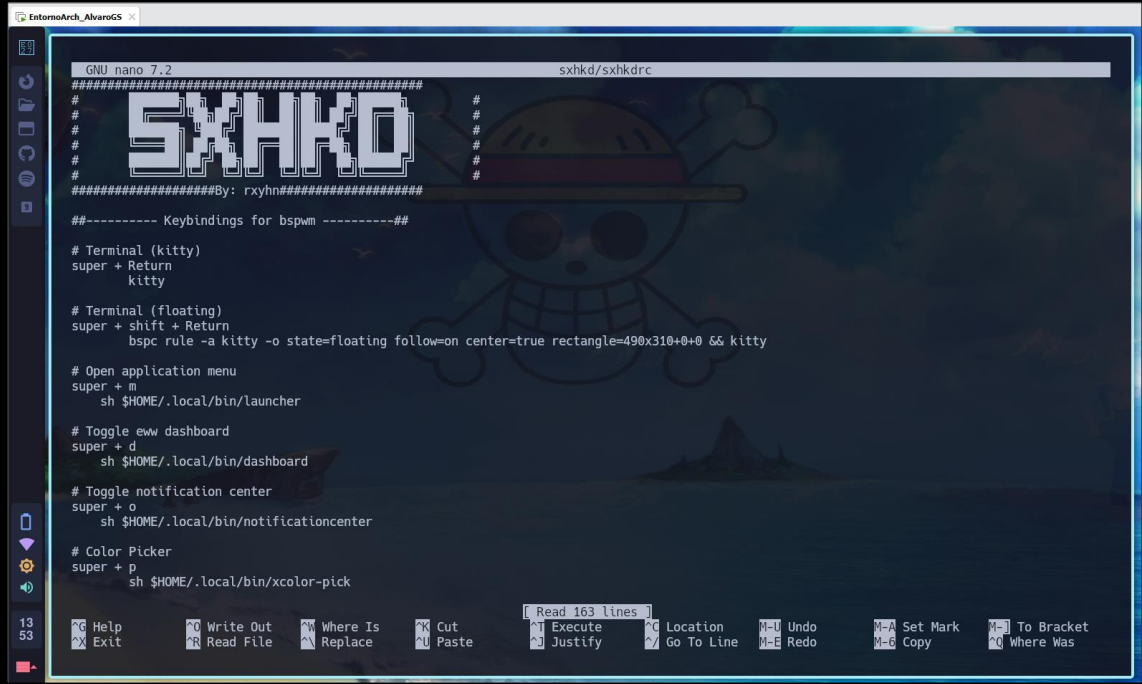
round-borders = 20;
round-borders-exclude = [
# "class_g = 'TelegramDesktop'",
];

# Specify a list of border width rules, in the format 'PIXELS:PATTERN'.
# Note we don't make any guarantee about possible conflicts with the
# border_width set by the window manager.
#
# example:
# round-borders-rule = [ "2:class_g = 'URxvt'" ];
round-borders-rule = [];

#####
# SHADOWS #
#####
# Enabled client-side shadows on windows. Note desktop windows
# (windows with 'NET_WM_WINDOW_TYPE_DESKTOP') never get shadow,
# unless explicitly requested using the wintypes option.
#
shadow = true

# The blur radius for shadows, in pixels. (defaults to 12)
```


Fichero “sxhkdr”. Utilizado para las combinaciones de teclas y atajos de teclado así como comandos personalizados.



The screenshot shows a terminal window titled "EntornoArch_AlvaroG5" with the GNU nano 7.2 editor open to the file "sxhk/sxhkdr". The file contains a large ASCII art skull with a skullcap, and a list of keybindings for the bspwm window manager. The keybindings include shortcuts for terminal (kitty), floating terminal, application menu, eww dashboard, notification center, and a color picker. The bottom of the window shows a menu bar with various editing and navigation options.

```
GNU nano 7.2 sxhk/sxhkdr
#####
#                               #
#                               #
#                               #
#                               #
#####By: rxyhn#####
#----- Keybindings for bspwm -----##
# Terminal (kitty)
super + Return
kitty

# Terminal (floating)
super + shift + Return
bspc rule -a kitty -o state=floating follow=on center=true rectangle=490x310+0+0 && kitty

# Open application menu
super + m
sh $HOME/.local/bin/launcher

# Toggle eww dashboard
super + d
sh $HOME/.local/bin/dashboard

# Toggle notification center
super + o
sh $HOME/.local/bin/notificationcenter

# Color Picker
super + p
sh $HOME/.local/bin/xcolor-pick

[ Read 163 lines ]
[?] Execute [?] Location [M-U] Undo [M-A] Set Mark [M-J] To Bracket
[?] Justify [?] Go To Line [M-E] Redo [M-C] Copy [M-Q] Where Was
[?] Help [?] Write Out [?] Where Is [?] Cut [?] Paste [?] Exit [?] Read File [?] Replace [?] Undo [?] Redo [?] Copy [?] Where Was
```

Manual de atajos & comodidades

Introducción

En esta sección, mostraré los atajos que están configurados por mí, de manera por defecto al descargar la máquina virtual. Todos estos atajos se pueden cambiar en el fichero SXHKD, yo solo documentaré algunos.

Adicionalmente, dejaré dentro del home de la propia MV, un fichero cheatsheet que tenga todos los comandos útiles.

Atajos y comodidades

Utilidad	Combinación
Buscar Comando	CTRL + R
Buscar interactivamente	CTRL + T
Abrir terminal	SUPER + ENTER
Cerrar terminal/pestaña	SHIFT + CTRL + W
Cambiar terminal	SUPER + FLECHA + IZQ/DER
Abrir pestaña	SHIFT + CTRL + T
Cambiar pestaña	SHIFT + CTRL + FLECHA IZQ/DER
Renombrar pestaña	SHIFT + CTRL + ALT + T
Bloquear sistema	CTRL + ALT + L
Borrar línea	CTRL + U
Tabla de Emojis	CTRL + SHIFT + U
Reiniciar GUI	CTRL + SHIFT + R
Abrir FireFox	CTRL + SHIFT + F
Abrir BurpSuite	CTRL + SHIFT + B

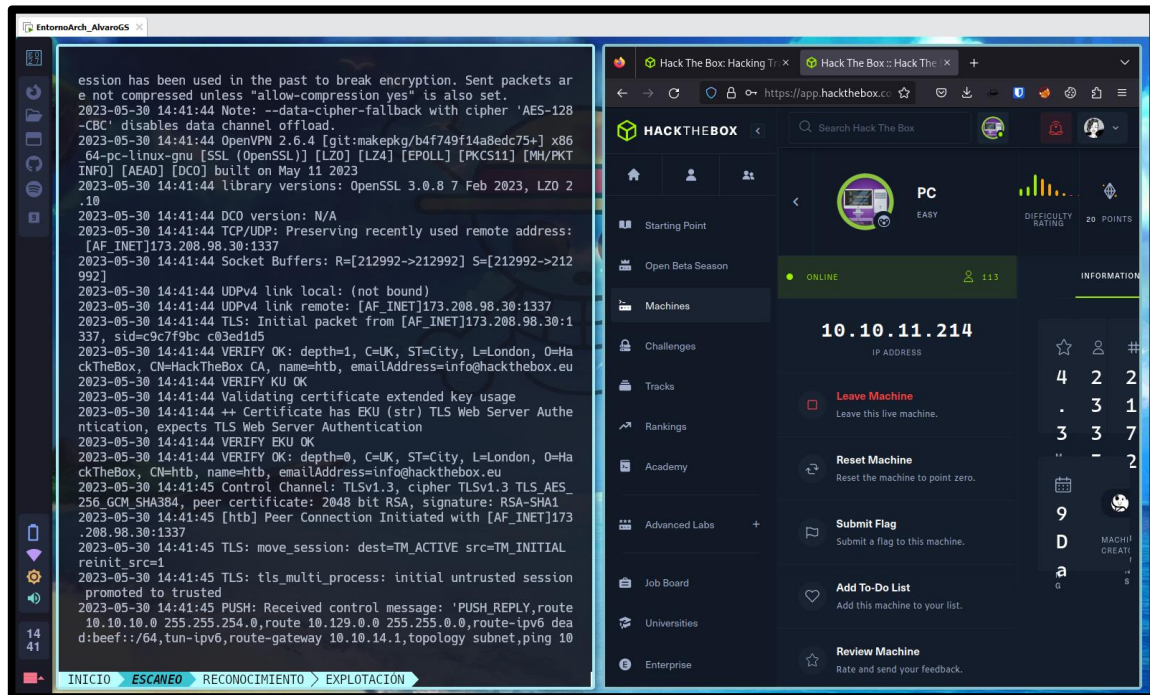
Pruebas generales

Introducción

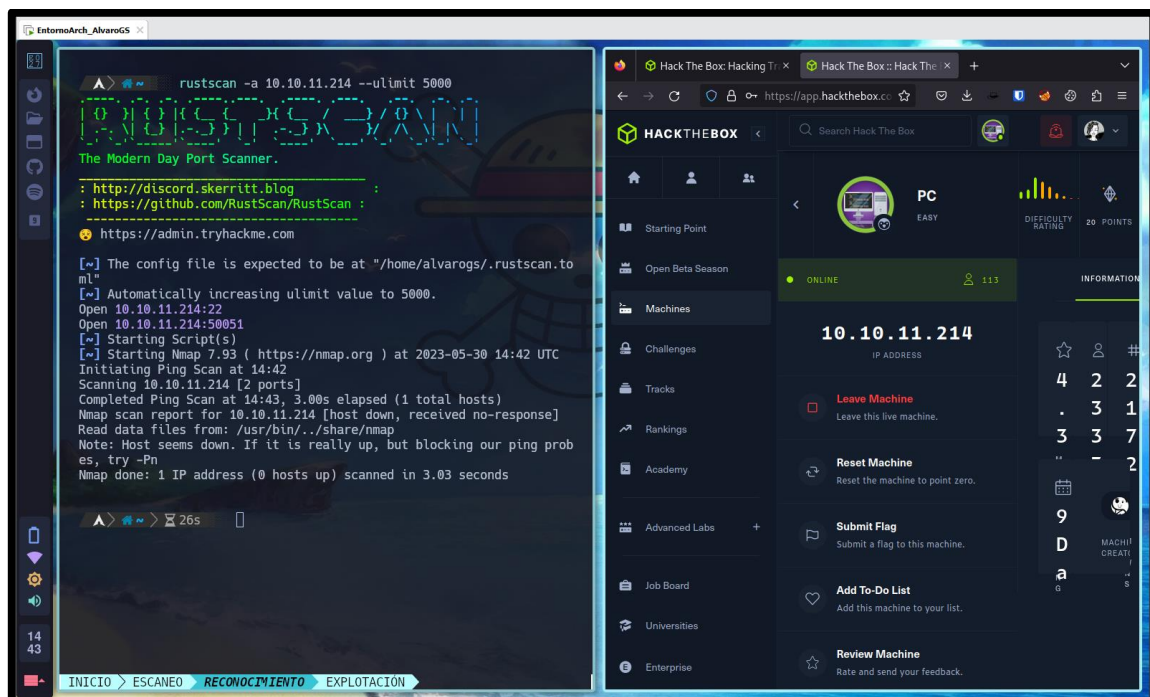
En este bloque, probaré las distintas utilidades para verificar su funcionamiento, así como el funcionamiento del sistema en general.

Pruebas de uso.

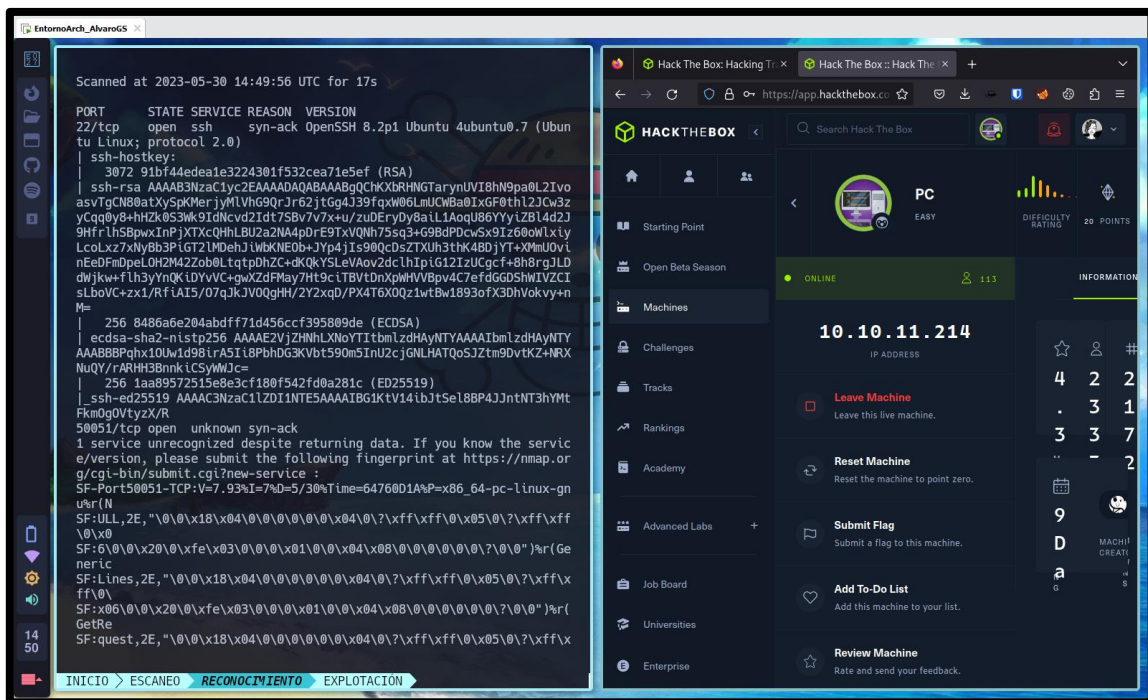
Uso de una VPN de HackTheBox para conectarme a una máquina virtual.



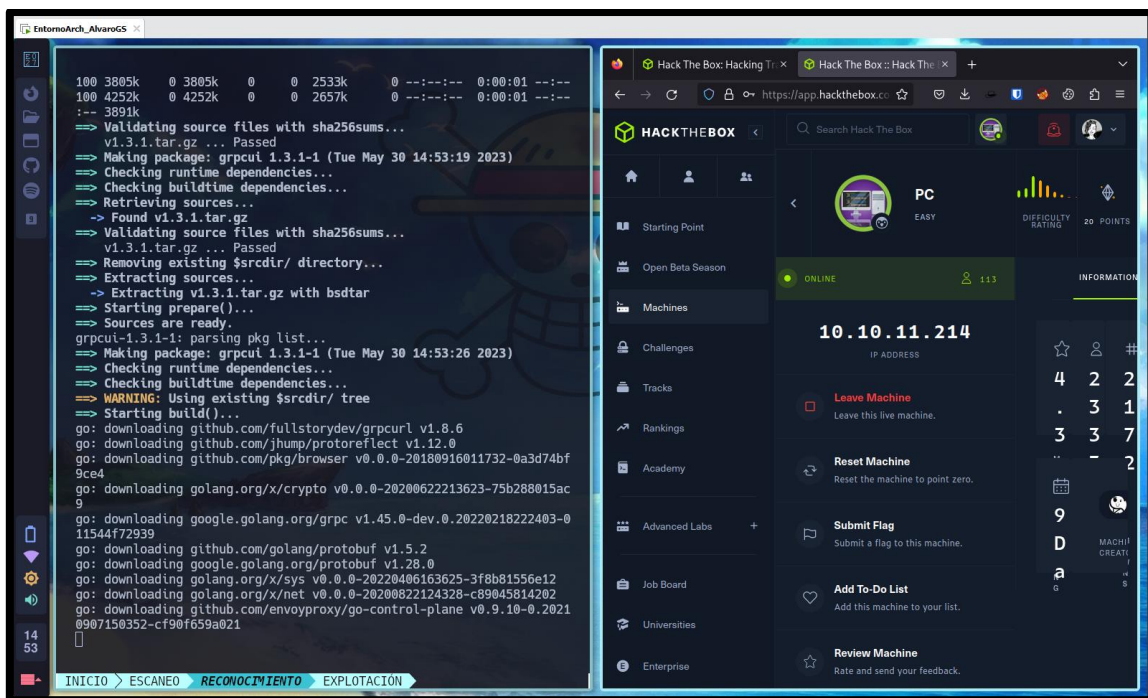
Escaneo de puertos inicial con Rustscan.



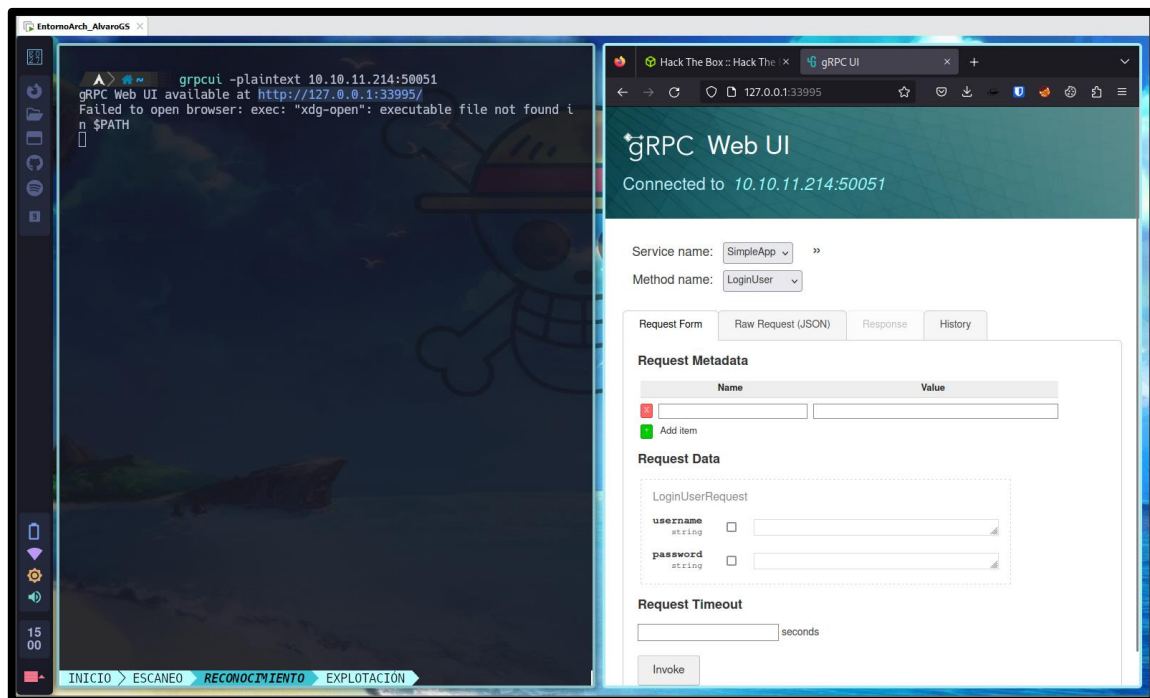
Escaneo intensivo a los puertos encontrados con RustScan (22 y 50051).



Instalación de utilidad grpcui.



Uso de la nueva utilidad contra la máquina virtual para obtener una página expuesta.



Resultados obtenidos

Tras usar el entorno un buen rato, haciendo uso de algunas utilidades y atajos de teclado ya mostrados anteriormente, puedo decir que sirve para trabajar de forma cómoda y rápida incluso sin estar todavía 100% acostumbrado al entorno.

Conclusiones

Considero que es un entorno útil y simple para hacer pentesting, y más todavía si estuviera acostumbrado al 100%. En cuanto a las expectativas que tenía, podría haber sido mejor pero igualmente cumple mucho con lo que me propuse al principio.

Webgrafía

<https://keepcoding.io/blog/que-es-metasploit-ciberseguridad/>

<https://www.ochobitshacenunbyte.com/2019/09/18/que-es-zsh-y-como-utilizar-sus-temas-y-complementos/>

<https://www.exploit-db.com/about-exploit-db>

<https://www.hackingarticles.in/a-detailed-guide-on-evil-winrm/>

<https://wfuzz.readthedocs.io/en/latest/>

<https://www.geeksforgeeks.org/rustscan-faster-nmap-scanning-with-rust/>

<https://nmap.org/book/>

<https://www.geeksforgeeks.org/what-is-burp-suite/>

<https://linuxhint.com/what-is-neofetch-for-linux/>

<https://neovim.io/charter/>

<https://www.freecodecamp.org/news/fzf-a-command-line-fuzzy-finder-missing-demo-a7de312403ff/>

<https://platzi.com/tutoriales/2383-prework-linux/12567-bat-y-lsd/>

<https://www.instructables.com/Bspwm-Installation-and-Configuration/>

<https://distro.tube/guest-articles/installing-and-using-sxhkd.html>

<https://dev.to/abdfnx/oh-my-zsh-powerlevel10k-cool-terminal-1no0>

[https://wiki.archlinux.org/title/Arch User Repository](https://wiki.archlinux.org/title/Arch_User_Repository)

<https://www.linuxfordevices.com/tutorials/linux/picom>

[https://www.reddit.com/r/i3wm/comments/vxfyk9/what is picom used for/](https://www.reddit.com/r/i3wm/comments/vxfyk9/what_is_picom_used_for/)

<https://itsfoss.com/pacman-command/>

Conclusión final

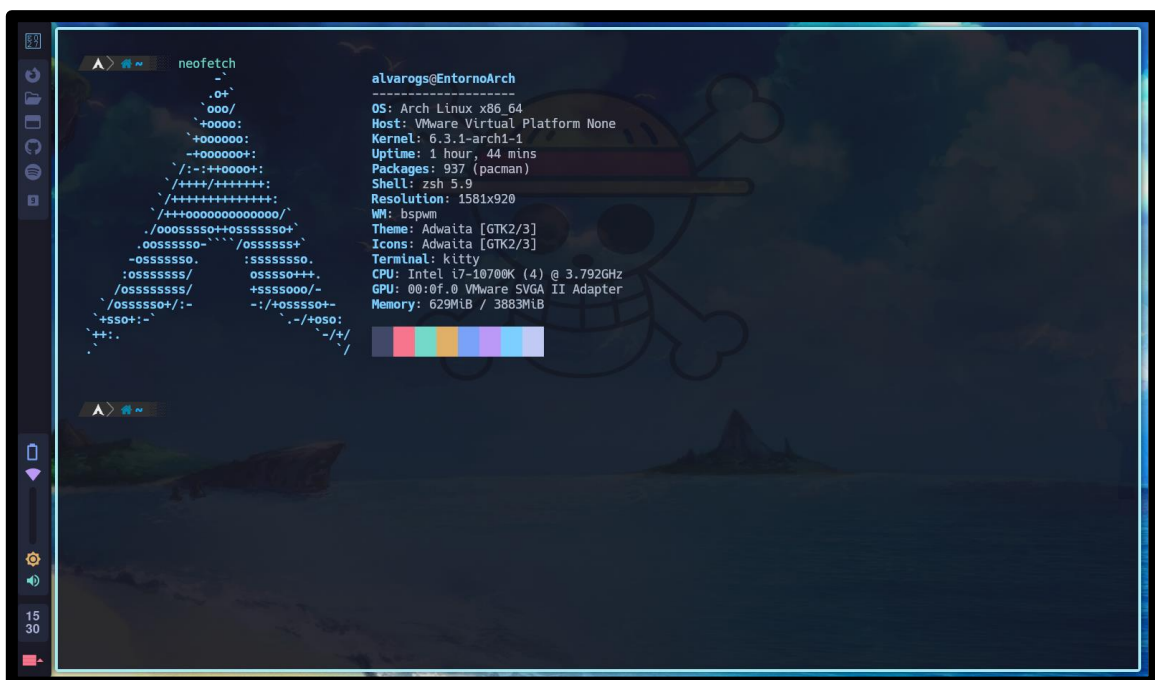
En resumen, me ha gustado mucho hacer este proyecto. Me he encontrado muchos errores e imprevistos por el camino pero al final gracias a las documentaciones y páginas de GitHub he conseguido ir encontrando la solución.

Construir el entorno poco a poco y desde cero también me ha ayudado a ver cómo funciona un entorno por dentro y cómo están formados. Así como la forma de implementar distintos entornos gráficos y la forma de editarlos.

También estoy bastante contento con el rendimiento final que tiene, aunque podría ser mejorable, sobre todo a la hora de abrir aplicaciones como FireFox, que tarda algo más de lo esperado.

Además, he conseguido que sea lo suficientemente útil como para poder usarse en trabajos de ciberseguridad de manera profesional, o al menos lo suficientemente cómodo como para resolver máquinas de HackTheBox desde el entorno de manera sencilla.

En general, estoy bastante satisfecho con mi proyecto.



Fin del documento