ODYSSEY LINUX · DOCUMENTACIÓN DE REFERENCIA

Odyssey Control Center — Guía Práctica

Cada panel, sección por sección. Qué hace, qué cambia, y qué pulsar — escrita para quien nunca lo ha usado antes.

Launch Edition

Don’t trust me. Verify me.

— nobody

Odyssey Linux — Odyssey Control Center: Guía Práctica
Launch Edition. Cubre el Control Center tal como se distribuye en la ISO de la Launch Edition — diez paneles completos, de Boot a About, más un capítulo marcador de posición para el módulo del controlador NVIDIA (visible solo en hardware NVIDIA, pendiente de capturas de pantalla desde una máquina que lo tenga).

Copyright © 2026 the Odyssey Linux project.
Publicada bajo la licencia Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).

Esta guía es el complemento práctico de la Security and Hardening Guide, que cubre los paneles Firewall y AppArmor con total profundidad técnica — aquí encontrarás la guía visual paso a paso y una referencia para el resto.

Cada pantalla, botón y comportamiento descrito aquí se leyó del código fuente real del Control Center y de capturas de pantalla reales, no de memoria.

Las etiquetas de los botones, los nombres de los paneles, las rutas de los archivos y los comandos permanecen en inglés porque son la interfaz real del programa: buscarlos traducidos en el Control Center no los encontrarías.

ÍNDICE

1 Qué es el Control Center 1
1.1 Patrones que verás en cada panel 2
2 Boot 3
2.1 Guía visual — encontrar otro sistema operativo y establecer el predeterminado 3
2.2 Guía visual — elegir qué arranca por defecto 5
2.3 Guía visual — la pestaña EFI 8
2.4 Parámetros del kernel 9
2.5 Advanced 9
3 Kernel 11
3.1 Guía visual — los preajustes 11
3.2 Guía visual — SYSCTL, grupo por grupo 12
3.3 Guía visual — CPU governor 14
3.4 Guía visual — hugepages 15
3.5 Guía visual — modules 16
3.6 Advanced 17
4 Memory & Swap 19
4.1 Guía visual 19
4.2 Advanced 20
5 Firewall 22
5.1 Guía visual 22
5.2 Advanced 23
6 AppArmor 24
6.1 Guía visual — STATUS 24
6.2 Guía visual — PROFILES 25
6.3 Guía visual — COMMUNITY 26
6.4 Guía visual — VIOLATIONS 28
6.5 Guía visual — EDITOR 29
6.6 Advanced 30
7 Privacy 32
7.1 Guía visual — SearXNG, paso a paso 32
7.2 Guía visual — Tor, paso a paso 34
7.3 Guía visual — combinándolos 37
7.4 Advanced 38
8 Containers 40
8.1 Guía visual — el motor 40
8.2 Guía visual — rootless vs. rootful, y el estado vacío 42
8.3 Guía visual — el catálogo curado, de principio a fin 42
8.4 Guía visual — la lista de contenedores y sus acciones 46
8.5 Guía visual — ejecutar un contenedor, y abrirlo 47
8.6 Uso de disco 48
8.7 Advanced 48
9 Gaming Mode 50
9.1 Guía visual — el interruptor principal 50
9.2 Guía visual — Packages y Steam 51
9.3 GameMode y MangoHud 53
9.4 Proton 54
9.5 Advanced 54
10 Telemetry 56
10.1 Guía visual 56
10.2 Advanced 58

11 About 60
11.1 Guía visual 60
11.2 Copy report 61
11.3 Advanced 61
12 NVIDIA driver 63

CHAPTER 1

Qué es el Control Center

El Odyssey Control Center son diez pequeños programas independientes detrás de un único lanzador — uno por cada área del sistema: Boot, Kernel, Memory & Swap, Firewall, AppArmor, Privacy, Containers, Gaming Mode, Telemetry, About, y (en hardware NVIDIA) el módulo del controlador NVIDIA.

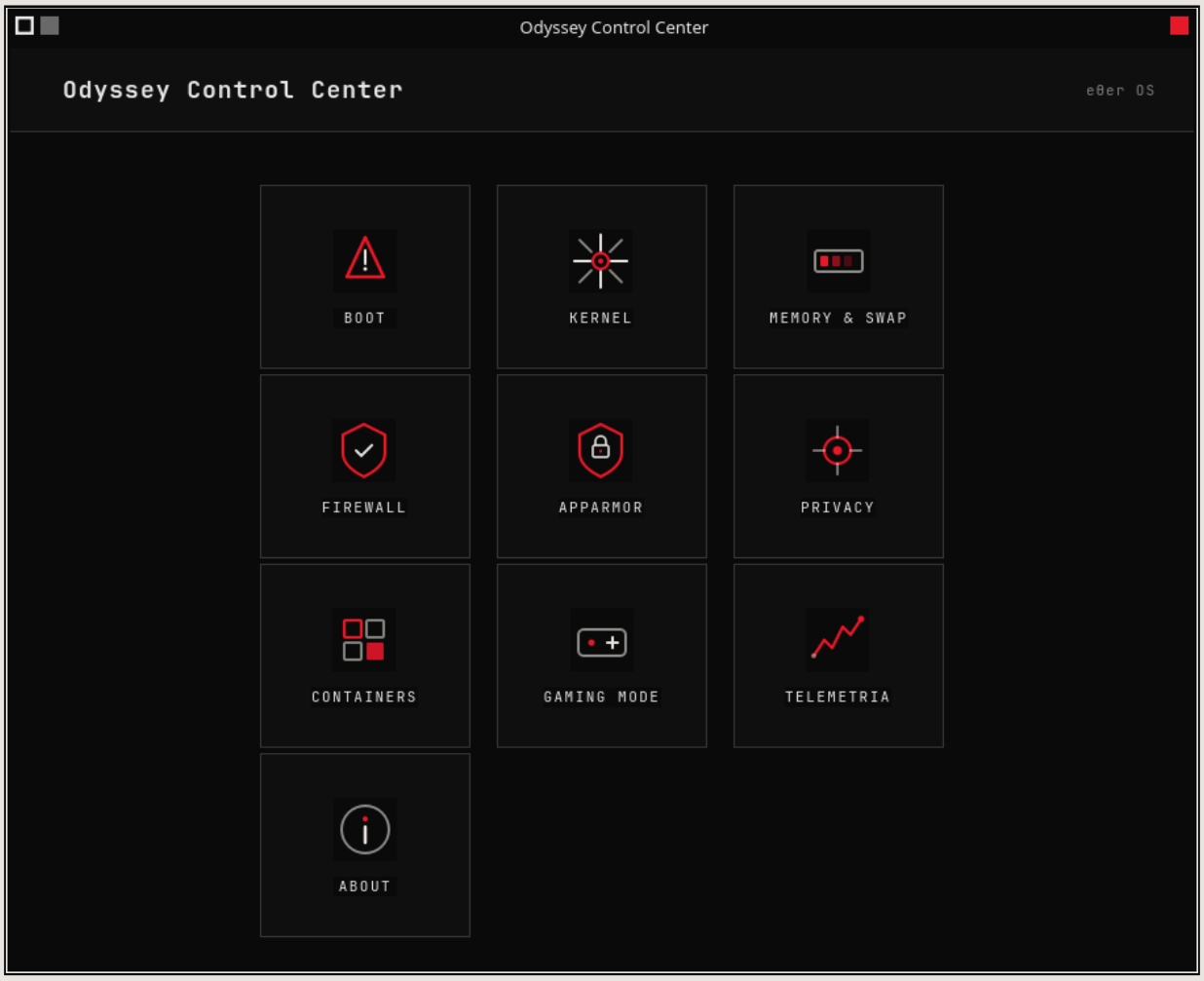

FIG · La pantalla de inicio del Control Center — un mosaico por cada panel.

Cada panel es una capa gráfica delgada sobre el mecanismo real de Linux subyacente: la configuración de GRUB, la propia interfaz sysctl del kernel, `nftables`, `podman`. Un panel nunca inventa su propio estado privado — lee los archivos y comandos reales, te los muestra, y escribe exactamente lo que pediste. Nada de lo que hace está oculto, y todo lo que hace también se puede hacer a mano, desde una terminal, con los comandos exactos que esta guía muestra junto a cada botón.

Esta es una guía práctica, no una referencia de cada opción — responde a "qué pulso, y qué pasa si lo hago." Cada capítulo tiene la misma estructura en dos partes: primero una **guía visual**, captura tras captura, botón tras botón, escrita para quien nunca ha abierto el Control Center antes — y luego una sección **ADVANCED**, marcada con un banner negro, que entra en el detalle: rutas exactas de archivos, los comandos reales que ejecuta cada botón por debajo, y qué hacer desde una terminal si prefieres

saltarte el panel por completo. Lee solo la guía visual, o salta directamente a Advanced si ya conoces Linux — ambas son completas por sí solas.

Para los dos paneles con implicaciones reales de seguridad — Firewall y AppArmor — la referencia profunda es la **Security and Hardening Guide**; aquí encontrarás la guía visual, lo esencial, y una referencia cruzada.

1.1 Patrones que verás en cada panel

Algunas convenciones se repiten en los diez paneles — aprenderlas una vez sirve para todos.

LA SOLICITUD DE CONTRASEÑA, UNA SOLA VEZ POR PANEL

La mayoría de los paneles necesitan leer o escribir archivos que pertenecen a root — la configuración de GRUB, sysctl, el conjunto de reglas del firewall. La primera vez que un panel lo necesita, recibes un único diálogo de autenticación. Los paneles están construidos para agrupar sus lecturas: abrir Boot, por ejemplo, lee la configuración de GRUB **y** el estado del firmware EFI en una sola solicitud, así que escribes tu contraseña una vez, no dos. Volver a abrir un panel en la misma sesión no la vuelve a pedir; las acciones posteriores (aplicar un cambio, recargar) muestran su propio progreso en el registro del panel mismo.

EL RASTREADOR DE PASOS

Los paneles que te guían a través de la instalación de algo antes de que se vuelva utilizable — os-prober en Boot, SearXNG y Tor en Privacy, el motor de contenedores en Containers — muestran un pequeño rastreador numerado en la parte superior: una fila de pasos, con el actual resaltado, y un único botón de acción que siempre hace **el siguiente paso**, sea cual sea. Nunca tienes que adivinar qué pulsar primero.

EL REGISTRO AL FINAL

Cada panel termina en un pequeño registro desplazable. No es decoración — es el comando exacto que se ejecutó, su salida, y un veredicto coloreado (verde para éxito, rojo para fallo). Si un panel hace algo que no entiendes, el registro es donde está la respuesta real.

TRY VS. MAKE PERSISTENT

Un par de botones recurrente, sobre todo en Kernel y Memory: **TRY NOW** (o **APPLY NOW**) cambia el sistema en ejecución de inmediato, y vuelve atrás en el próximo reinicio si no haces nada más. **MAKE PERSISTENT** escribe el mismo valor en un archivo de configuración bajo `/etc/`, para que sobreviva también a los reinicios. La separación existe a propósito — probar un valor no cuesta nada, y nada es permanente hasta que tú lo dices, dos veces.

CHAPTER 2

Boot

El panel Boot tiene dos pestañas en cualquier máquina que arranca en modo UEFI: **GRUB** (el propio menú de arranque) y **EFI** (el nivel de firmware debajo de él). En máquinas más antiguas, solo BIOS, únicamente existe GRUB, porque la capa EFI de abajo simplemente no está ahí.

> **DOS NIVELES, NO UNO**
>
> **GRUB** es el menú de arranque que ves en pantalla — decide qué sistema operativo arranca una vez que GRUB mismo ya está en ejecución. El **firmware EFI/UEFI** es un nivel por debajo de GRUB — decide a qué bootloader la máquina entrega el control en primer lugar, antes incluso de que GRUB se haya cargado. La mayoría de la gente nunca toca la pestaña EFI; existe para los casos que GRUB no puede alcanzar, como una segunda instalación de Windows que gestiona su propia entrada de arranque de forma independiente.

2.1 Guía visual — encontrar otro sistema operativo y establecer el predeterminado

La pestaña GRUB se abre en la sección **Other OS detection**. Odyssey no escanea otros sistemas por defecto — la mayoría de las instalaciones solo tienen Odyssey en el disco, y escanear cuesta tiempo en cada regeneración del menú de arranque para nada. Activarlo es una secuencia guiada de tres pasos, mostrada como un pequeño rastreador numerado.

1. **Installed**

   La herramienta de escaneo, os-prober, todavía no está en el sistema. Un solo botón: **INSTALL OS-PROBER**.

2. **Enable scan**

   La herramienta está instalada, pero GRUB todavía está configurado para ignorarla — el valor por defecto de Odyssey. Un solo botón: **ENABLE SCAN**.

3. **Scan**

   GRUB ahora la usará. Un solo botón, y el que volverás a usar cada vez que agregues o quites otro sistema operativo más adelante: **SCAN & REGENERATE**.

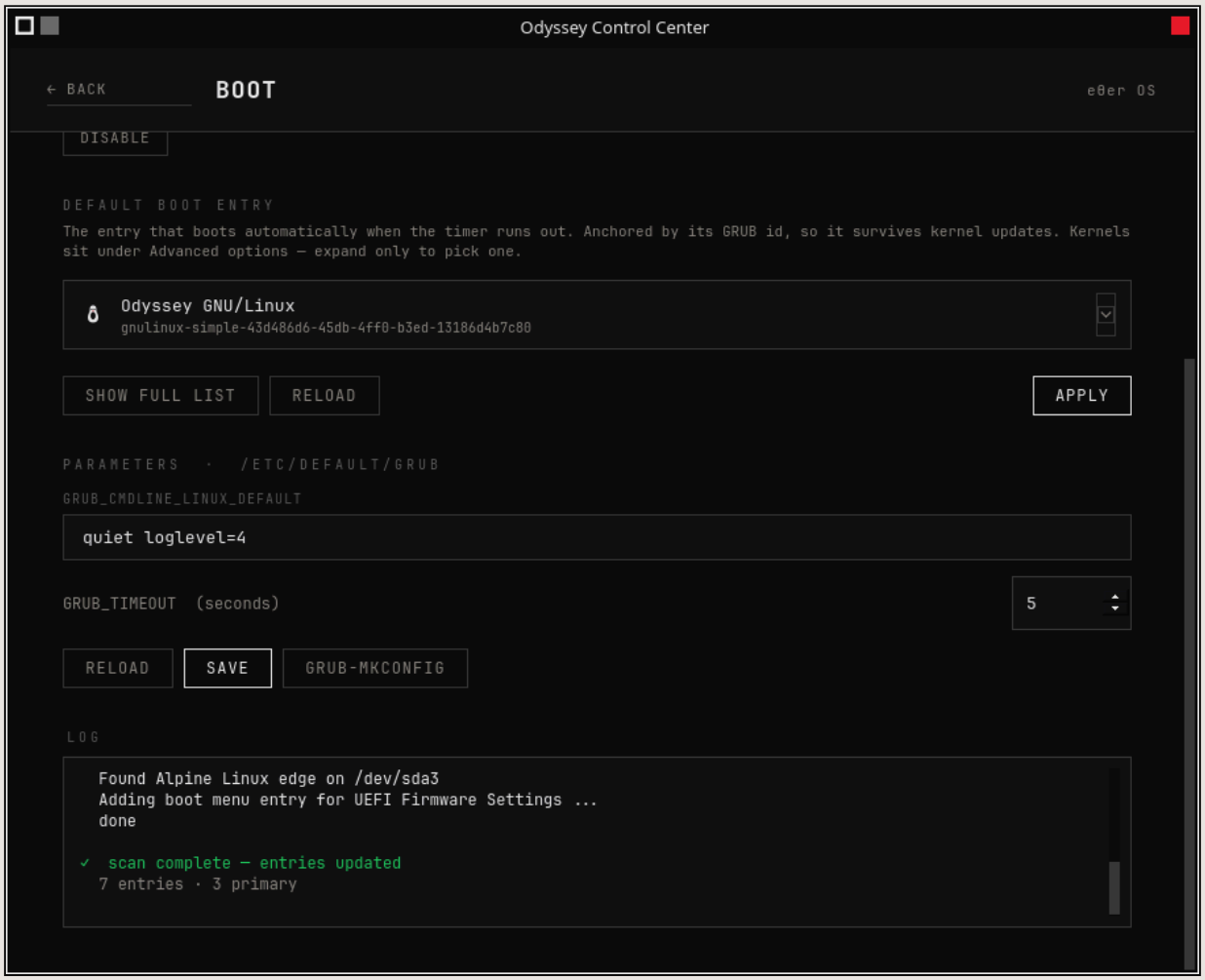

FIG · Los tres pasos en verde — os-prober instalado, escaneo habilitado, listo. El único botón de acción siempre corresponde al paso actual.

Haz clic en los pasos (cada uno vuelve a pedir autenticación solo si es necesario) y ejecuta el escaneo. El panel te muestra exactamente qué encontró antes de que dejes la sección:

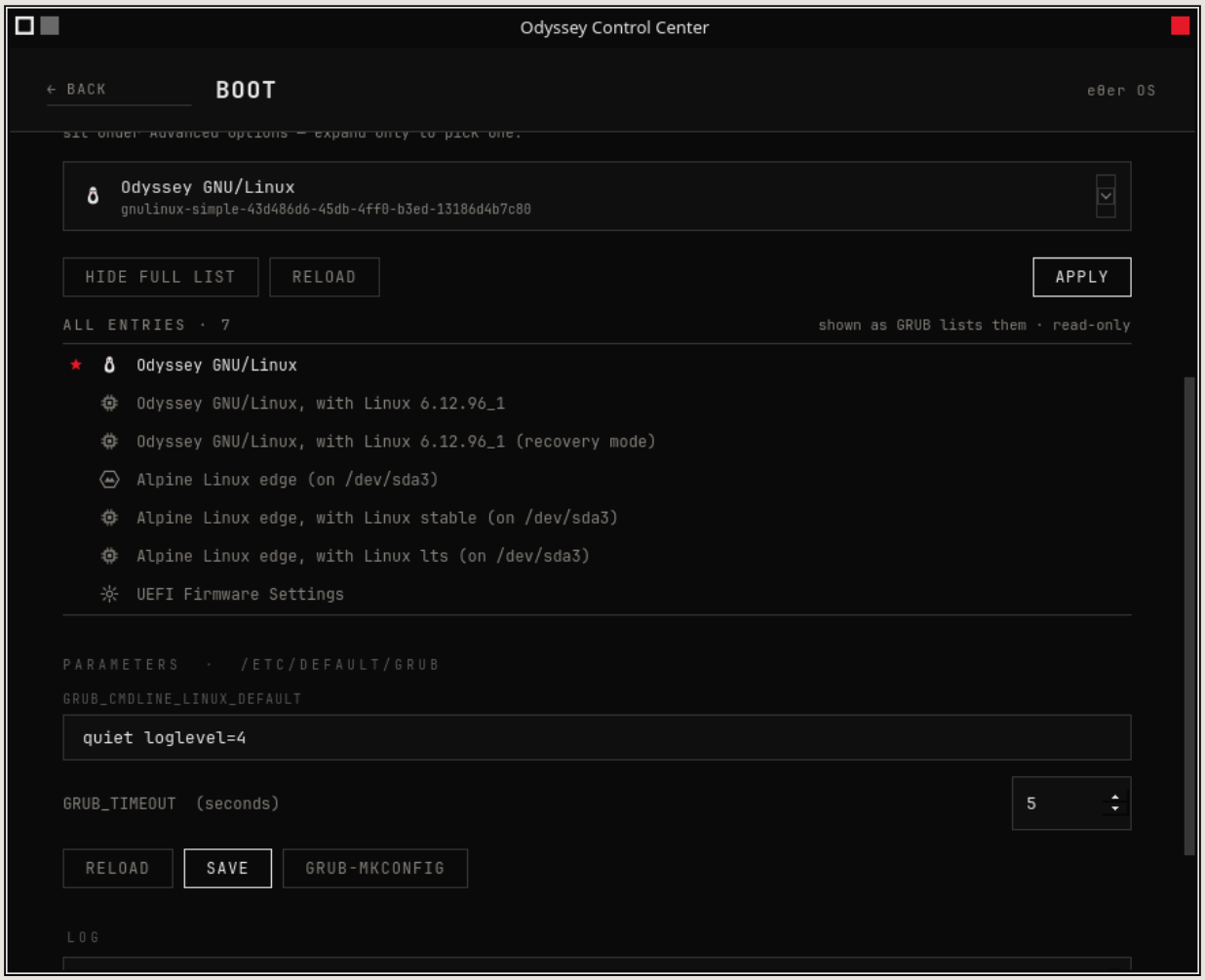

FIG · Escaneo completo: Alpine Linux encontrado en /dev/sda3, junto con las propias entradas de Odyssey y la entrada UEFI Firmware Settings — 7 entradas, 3 primarias.

El registro lo confirma en términos simples — **“scan complete — entries updated · 7 entries · 3 primary.”** Cualquier cosa que os-prober encuentre ahora aparece automáticamente en la sección de abajo, lista para ser elegida.

2.2 Guía visual — elegir qué arranca por defecto

Desplázate hasta **Default boot entry**. Muestra el valor predeterminado actual como una fila; al hacer clic se abre un menú desplegable con todo lo demás que puedes elegir en su lugar.

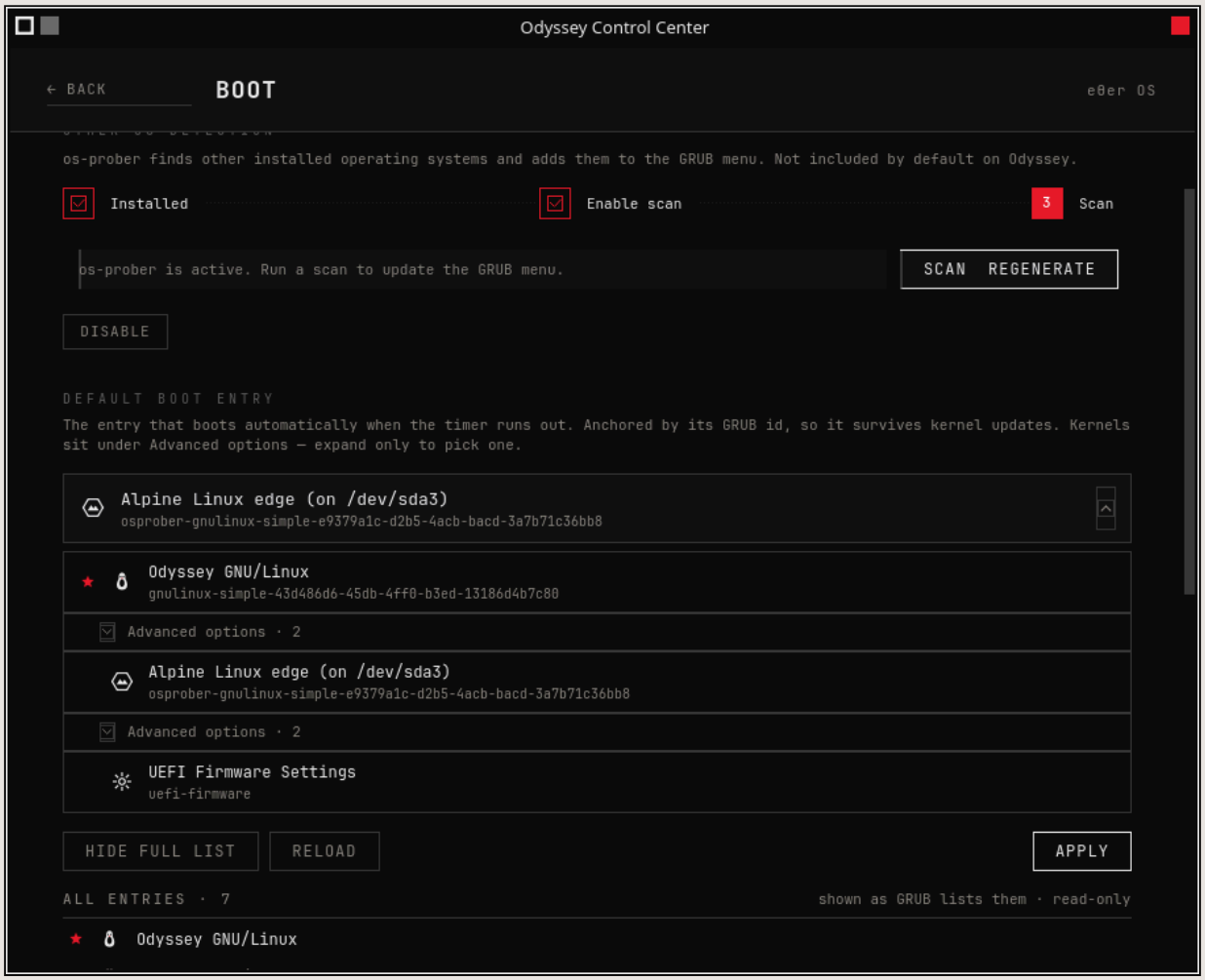

FIG · El menú desplegable abierto, con Alpine Linux seleccionado como elección pendiente — todavía no aplicada.

Elegir una entrada de la lista **pone en espera** la elección — todavía no tiene efecto. Se nota que solo está en espera porque el recuadro sigue mostrando la nueva selección pero nada se ha escrito. El botón que la confirma es **APPLY**, a la derecha:

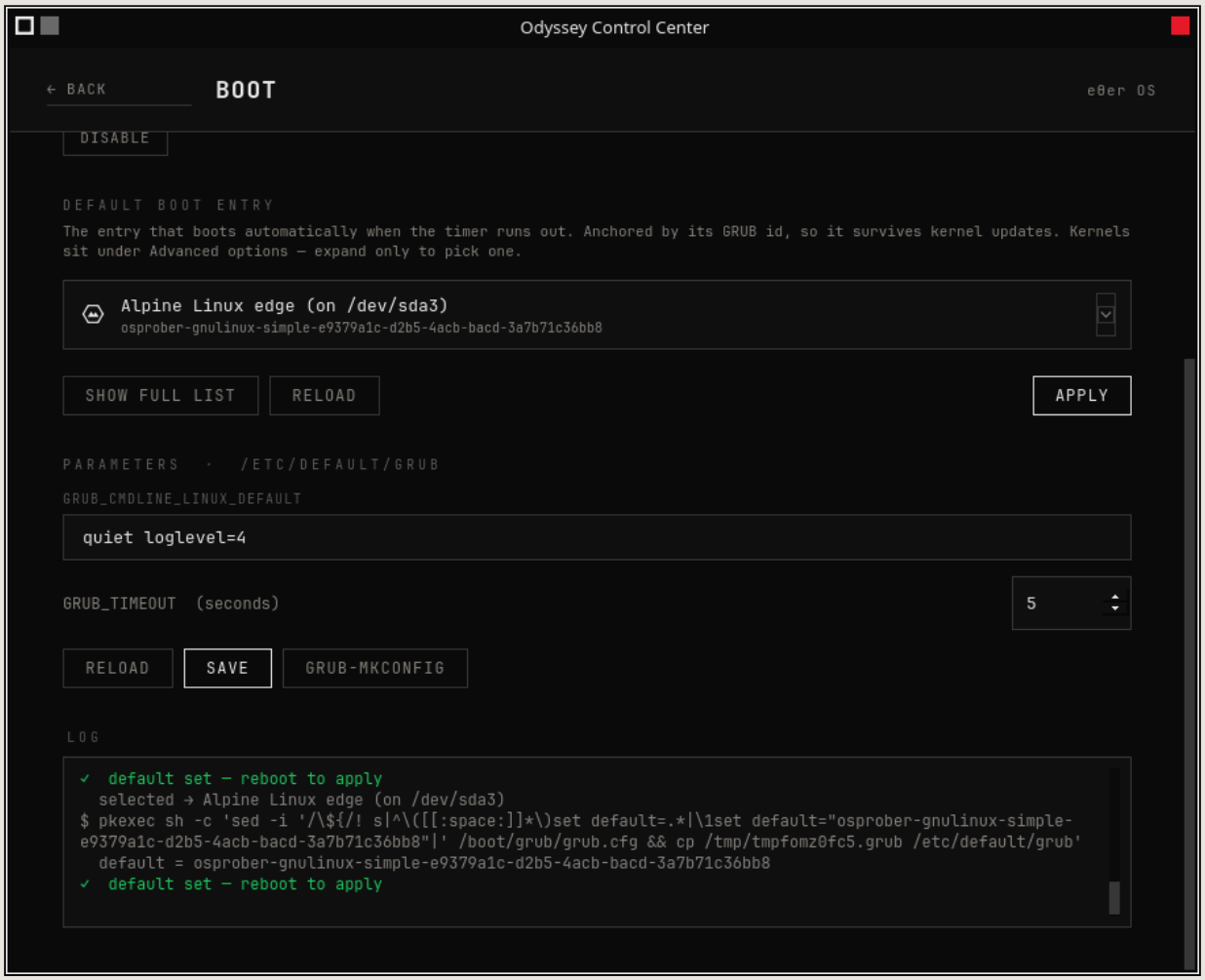

FIG · Después de APPLY: el registro confirma “default set — reboot to apply,” y el selector ahora muestra Alpine Linux como el predeterminado activo.

WHAT	Boot → pestaña GRUB → haz clic en la entrada predeterminada actual → elige una diferente del menú desplegable → haz clic en **APPLY**.
WHY	Tienes arranque dual y quieres que un sistema diferente arranque automáticamente, o una actualización del kernel dejó seleccionado un kernel antiguo y quieres volver al más reciente.
RESULT	La entrada elegida arranca automáticamente desde el próximo reinicio en adelante. El cambio está anclado al id estable de esa entrada, así que sobrevive a futuras actualizaciones del kernel.
RISK	Ninguno para tus datos — en el peor caso eliges la entrada equivocada y vuelves a cambiarla de nuevo.

SHOW FULL LIST muestra cada entrada exactamente como las lista GRUB mismo — solo lectura, agrupada de forma que las variantes por kernel individual queden bajo una fila expandible “Advanced options” en lugar de saturar la lista principal. **RELOAD** vuelve a leer la configuración de GRUB si crees que cambió fuera del panel.

2.3 Guía visual — la pestaña EFI

Haz clic en la pestaña **EFI** en la parte superior. Este es el nivel de firmware — lo que el propio menú de arranque de tu placa base muestra antes de que cualquier sistema operativo haya siquiera comenzado.

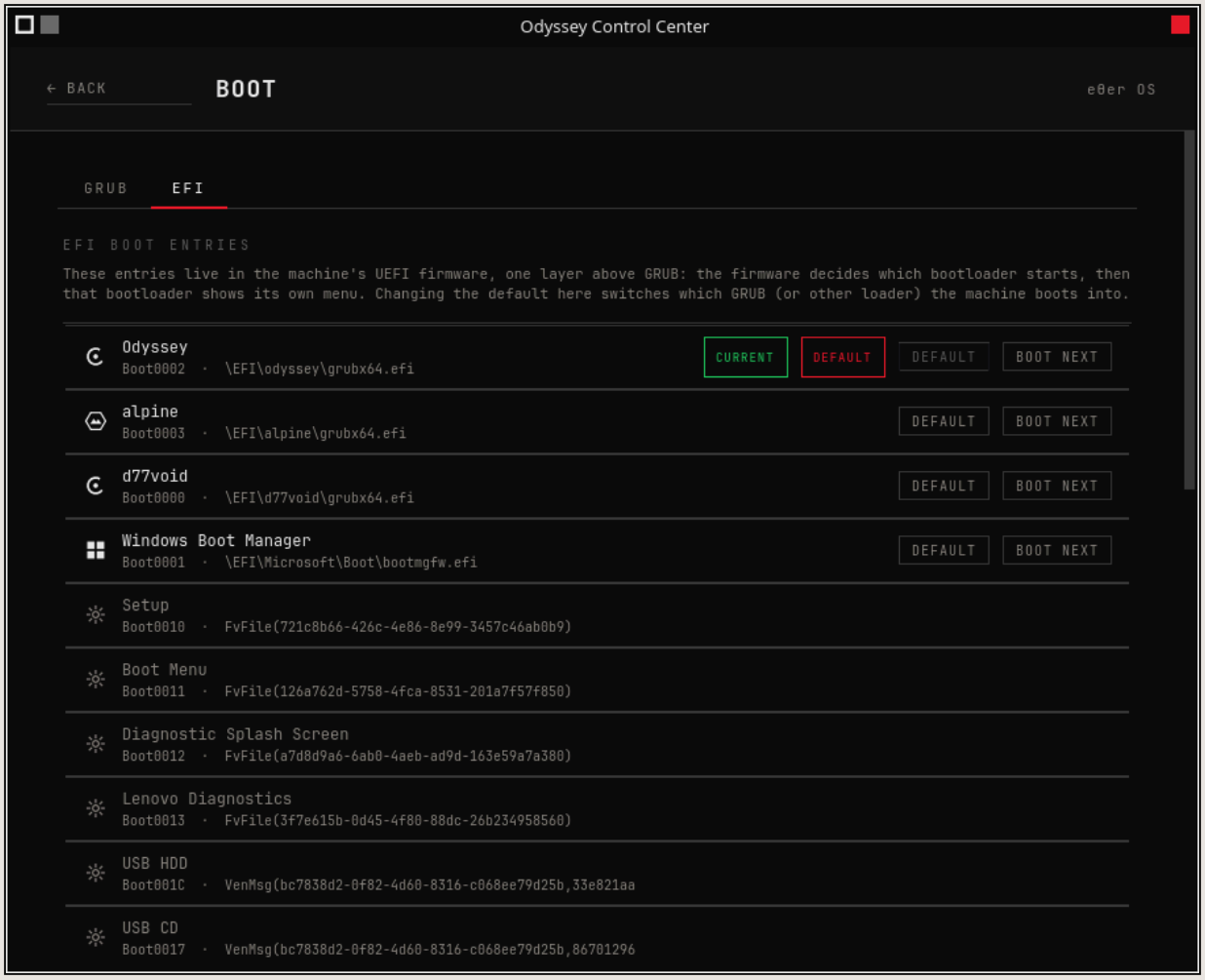

FIG · La pestaña EFI: Odyssey marcado como CURRENT y DEFAULT en verde/rojo, alpine, d77void y Windows Boot Manager cada uno con sus propios botones DEFAULT / BOOT NEXT, más las entradas de utilidad del firmware debajo.

Cada fila es una entrada de firmware — su nombre, su id de GRUB (`Boot0002` y así sucesivamente), y el archivo `.efi` exacto al que apunta. Dos botones por fila:

BOTÓN	QUÉ HACE
DEFAULT	Convierte esta entrada en el predeterminado permanente del firmware — arranca en cada reinicio a partir de ahora.
BOOT NEXT	Una excepción única — esta entrada arranca solo en el **próximo** reinicio, luego el orden normal se reanuda automáticamente. Aparece un banner en la parte superior mientras hay una pendiente, con un botón CANCEL.

WHAT	Boot → pestaña EFI → encuentra la entrada que quieres (p. ej. “Odyssey” o “Windows Boot Manager”) → haz clic en DEFAULT, o BOOT NEXT para una excepción única.
WHY	Si el propio menú de arranque de tu placa base está iniciando el bootloader equivocado por completo — no es un ajuste de GRUB, es del firmware — este es el único lugar que lo soluciona.
RESULT	DEFAULT: la entrada se mueve al frente del orden de arranque del firmware de forma permanente. BOOT NEXT: arranca una vez, luego revierte.
RISK	Bajo, y siempre recuperable desde esta misma pestaña o desde la propia pantalla de configuración UEFI de tu placa base (normalmente Supr o F2 al encender) si algo arranca de forma inesperada.

2.4 Parámetros del kernel

La última sección de la pestaña GRUB es la línea de comandos del kernel en texto plano — editable como un campo de texto normal, sin sintaxis de GRUB requerida — y el tiempo de espera del menú en segundos. **SAVE** los escribe en `/etc/default/grub`; **GRUB-MKCONFIG** regenera el menú de arranque real a partir de ese archivo, que es el paso que hace efectivo un parámetro guardado en el próximo arranque. El panel siempre vuelve a aplicar tu elección de predeterminado actual cuando regenera, así que esto nunca restablece qué sistema arranca por defecto.

```
▸ ADVANCED

Past this point: config file paths, raw commands, and what the panel is doing underneath.
```

2.5 Advanced

QUÉ ESTÁ PASANDO REALMENTE

La configuración de GRUB de Odyssey es **plana** — sin bloques `submenu`. Cada kernel es un `menuentry` de nivel superior, lo que significa que un id estable (no una posición numérica) siempre puede seleccionarlo correctamente, incluso entre actualizaciones del kernel. El agrupamiento “Advanced options” que ves en el selector es pura comodidad visual construida emparejando el UUID compartido entre la entrada simple de un sistema operativo y sus variantes por kernel — el propio archivo de GRUB no tiene ese agrupamiento.

Establecer un predeterminado escribe en dos lugares en la misma acción: el `GRUB_DEFAULT` persistente en `/etc/default/grub`, y — porque `grub.cfg` puede no regenerarse de inmediato — una edición directa y quirúrgica vía `sed` de la línea activa `set default="..."` en `/boot/grub/grub.cfg` mismo, saltándose la rama one-shot `${next_entry}` que usa `grub-reboot`. Por eso un predeterminado surte efecto en el próximo arranque sin forzar primero un `grub-mkconfig` completo.

```
$ cat /etc/default/grub | grep GRUB_DEFAULT
$ grep 'set default=' /boot/grub/grub.cfg
```

OS-PROBER A MANO

```
$ sudo xbps-install -y os-prober
# habilita el escaneo — edita GRUB_DISABLE_OS_PROBER en /etc/default/grub
$ sudo sed -i 's/GRUB_DISABLE_OS_PROBER=true/GRUB_DISABLE_OS_PROBER=false/' /etc/default/
grub
$ sudo os-prober
$ sudo grub-mkconfig -o /boot/grub/grub.cfg
```

EFI A MANO — EFIBOOTMGR

La pestaña EFI es una interfaz para `efibootmgr`. Todo lo que hace corresponde a un comando:

```
# lista todas las entradas de firmware, en el orden de arranque actual
$ sudo efibootmgr -v
# reordena: pon la entrada 0002 primero (= DEFAULT)
$ sudo efibootmgr -o 0002,0000,0001,0003
# arranque único en la entrada 0003 (= BOOT NEXT)
$ sudo efibootmgr -n 0003
# cancela un BootNext pendiente
$ sudo efibootmgr -N
```

CARGADORES HUÉRFANOS

Ocasionalmente un instalador escribe su propio bootloader en la EFI System Partition sin registrar una entrada de firmware para él — el cargador existe en el disco pero el firmware no sabe que está ahí. El panel de Odyssey escanea directamente la ESP y los lista por separado como **"ON DISK, NOT IN FIRMWARE,"** cada uno con un REGISTER de un clic. A mano, el equivalente es `efibootmgr -c -d /dev/sdX -p N -L "Nombre" -l '\EFI\nombre\bootx64.efi'` — ten en cuenta que las entradas recién creadas se colocan **primero** en el orden de arranque por el propio firmware, así que vuelve a aplicar DEFAULT en tu entrada habitual después, si no quieres que tome el control.

ARCHIVOS QUE TOCA ESTE PANEL

ARCHIVO	FUNCIÓN
`/etc/default/grub`	Línea de comandos del kernel, tiempo de espera, GRUB_DEFAULT, GRUB_DISABLE_OS_PROBER
`/boot/grub/grub.cfg`	El menú de arranque generado en sí — nunca lo edites a mano; se sobrescribe con grub-mkconfig
Variables EFI (NVRAM)	Entradas de arranque del firmware — no es un archivo en absoluto; se lee y escribe enteramente a través de efibootmgr

CHAPTER 3

Kernel

Cuatro pestañas — **SYSCTL**, **CPU GOVERNOR**, **HUGEPAGES**, **MODULES** — más una fila de preajustes de un clic encima de las cuatro, que se aplica a todas a la vez.

3.1 Guía visual — los preajustes

En la parte superior: la versión de tu kernel en ejecución, el planificador activo, y cuatro botones de preajuste — **DESKTOP**, **GAMING**, **SERVER**, **BALANCED**.

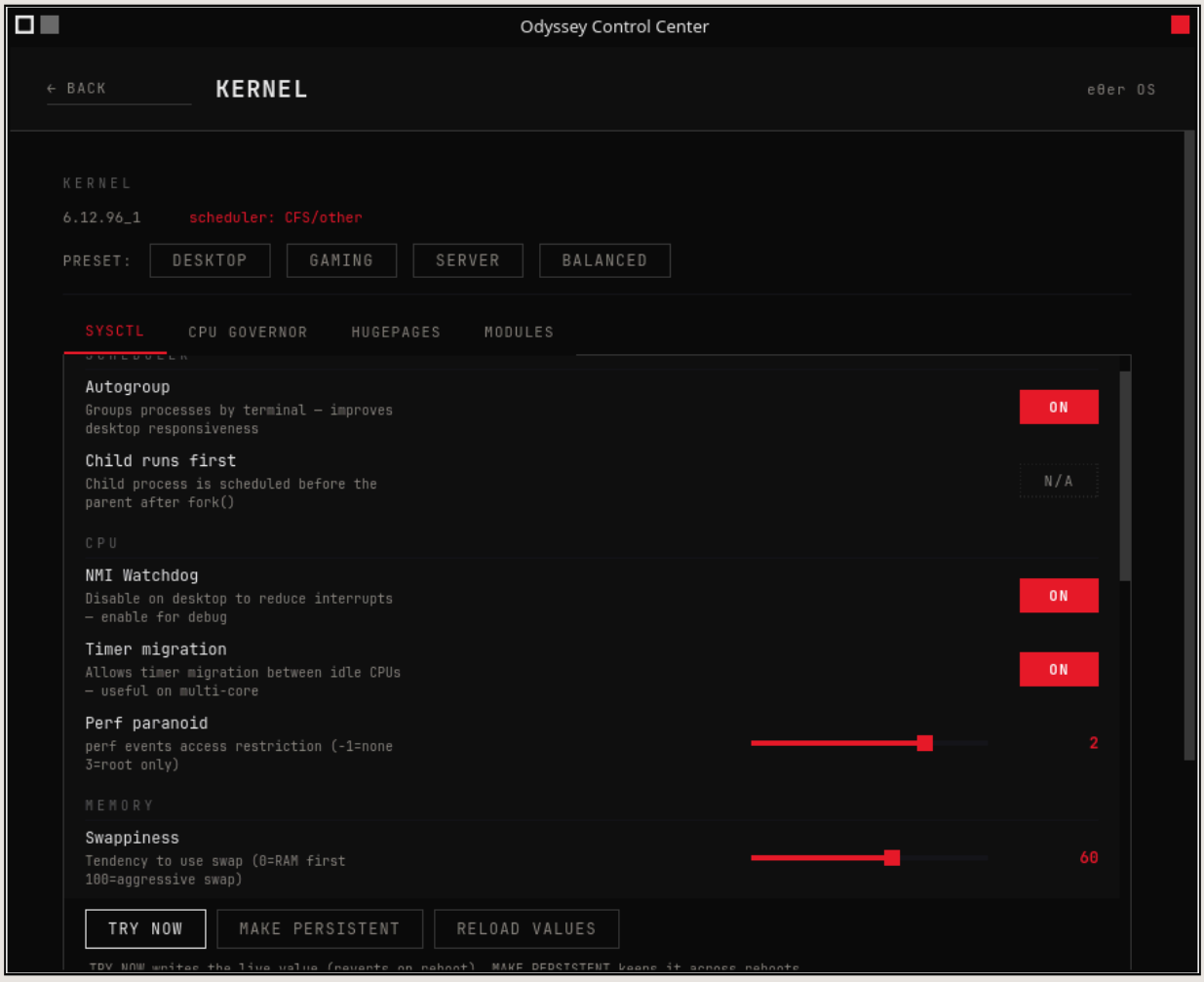

FIG · La pestaña SYSCTL recién abierta: kernel 6.12.96_1, planificador CFS/other, los cuatro preajustes, y los grupos Scheduler/CPU/Memory debajo.

PREAJUSTE	CUÁNDO VALE LA PENA USARLO
DESKTOP	Una máquina de uso diario — navegación, trabajo de oficina, multitarea ligera. Swappiness más baja, presión moderada sobre la caché, timer migration activa para un uso energético suave en reposo. El valor por defecto sensato para la mayoría de la gente.
GAMING	Swappiness empujada casi a cero (mantén todo en RAM), timer migration desactivada (los núcleos permanecen dedicados en lugar de migrar trabajo), restricciones de eventos perf relajadas para herramientas de perfilado como MangoHud. Úsalo cuando los frame times importan más que el consumo energético en reposo.
SERVER	Swappiness y tolerancia a la presión de caché más altas, NMI watchdog activo, restricciones más estrictas sobre punteros del kernel/registros. Úsalo en una máquina sin cabeza que prioriza estabilidad y seguridad sobre capacidad de respuesta interactiva.
BALANCED	Un punto intermedio entre Desktop y Gaming — para una máquina que a veces es una estación de trabajo, a veces una sesión de juego, y no quieres cambiar de preajuste constantemente.

WHAT	Kernel → haz clic en DESKTOP, GAMING, SERVER o BALANCED.
WHY	Ajustar a mano una docena de valores sysctl es pedir mucho a quien solo quiere “buenos ajustes para jugar” — un preajuste es exactamente eso, ya decidido.
RESULT	Cada valor en la pestaña SYSCTL salta a los números del preajuste, listos y visibles — de todos modos necesitas TRY NOW o MAKE PERSISTENT (abajo) para aplicarlos de verdad.
RISK	Ninguno — elegir un preajuste solo rellena el formulario.

3.2 Guía visual — SYSCTL, grupo por grupo

Desplázate hacia abajo y cada valor está agrupado bajo un encabezado, cada uno con una descripción en lenguaje sencillo directamente en el panel — nunca necesitas saber qué significa `vm.compaction_proactiveness` sin leerlo primero ahí.

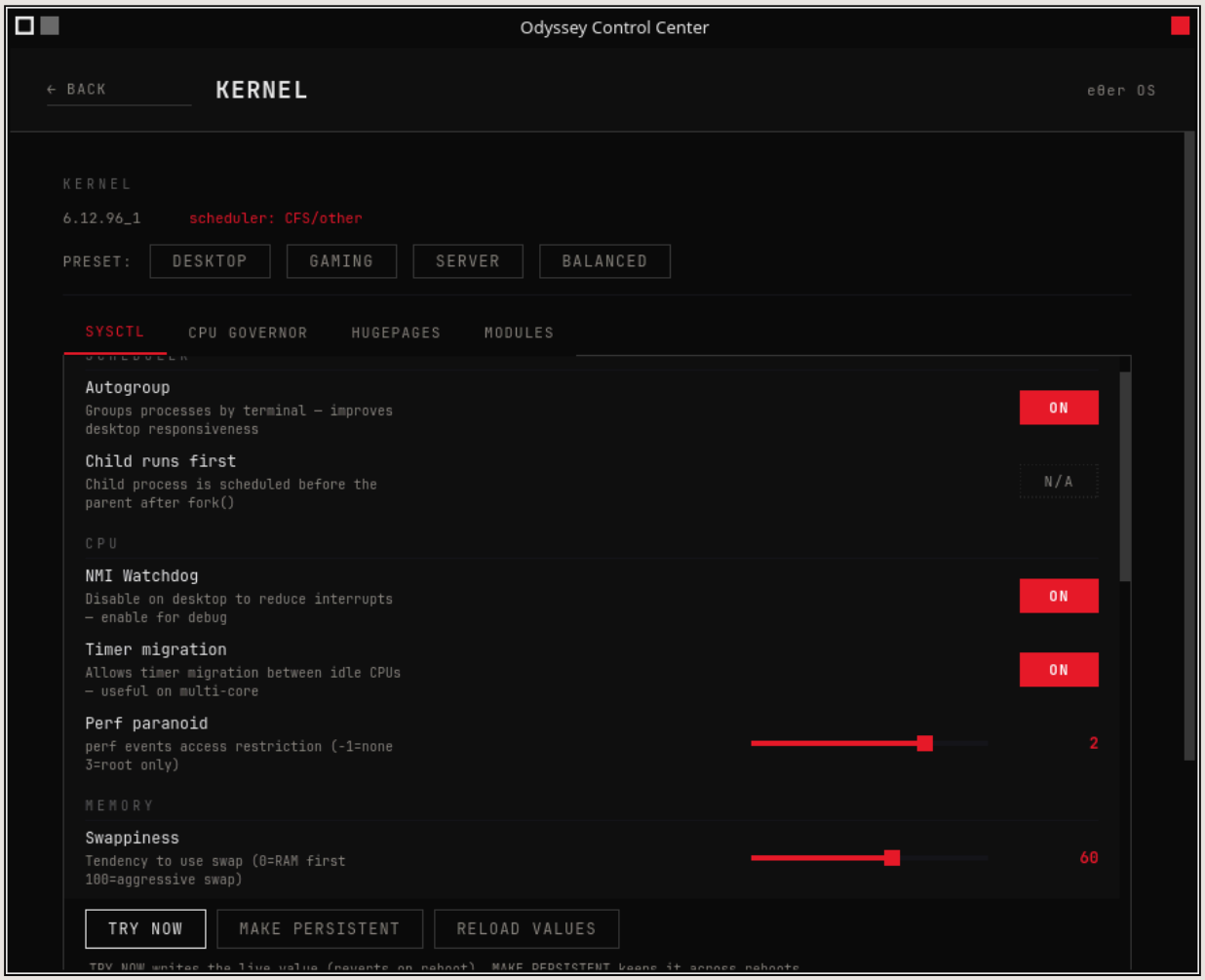

FIG · Grupos Scheduler y CPU: Autogroup, Child runs first, NMI Watchdog, Timer migration, Perf paranoid — cada uno con su propio interruptor o deslizador y una línea de explicación.

GRUPO	QUÉ CONTIENE
Scheduler	Cómo se agrupan y priorizan los procesos
CPU	Watchdog, timer migration, restricciones de monitoreo de rendimiento
Memory	Swappiness, presión de caché, umbrales de escritura de páginas sucias
Network	TCP Fast Open, tamaño de la cola de backlog, límites de los búferes de socket
Security	Si los punteros del kernel y los mensajes de registro se ocultan a usuarios sin privilegios

Dos botones al final aplican tus ediciones — y son deliberadamente diferentes:

WHAT	TRY NOW
WHY	Probar un valor con cero riesgo de que sobreviva a un mal reinicio.
RESULT	Escrito en vivo vía `sysctl` — activo de inmediato, solo para esta sesión.
RISK	Ninguno — revierte automáticamente al reiniciar.

WHAT	MAKE PERSISTENT
WHY	Has probado un valor (o un preajuste completo) y quieres que sobreviva a los reinicios.
RESULT	Escrito en `/etc/sysctl.d/99-odyssey.conf` **y** aplicado en vivo en el mismo clic.
RISK	Bajo — es un archivo de texto; borra la línea o vuelve a aplicar para deshacer.

RELOAD VALUES vuelve a leer lo que está actualmente activo, descartando cualquier cosa escrita pero no aplicada.

3.3 Guía visual — CPU governor

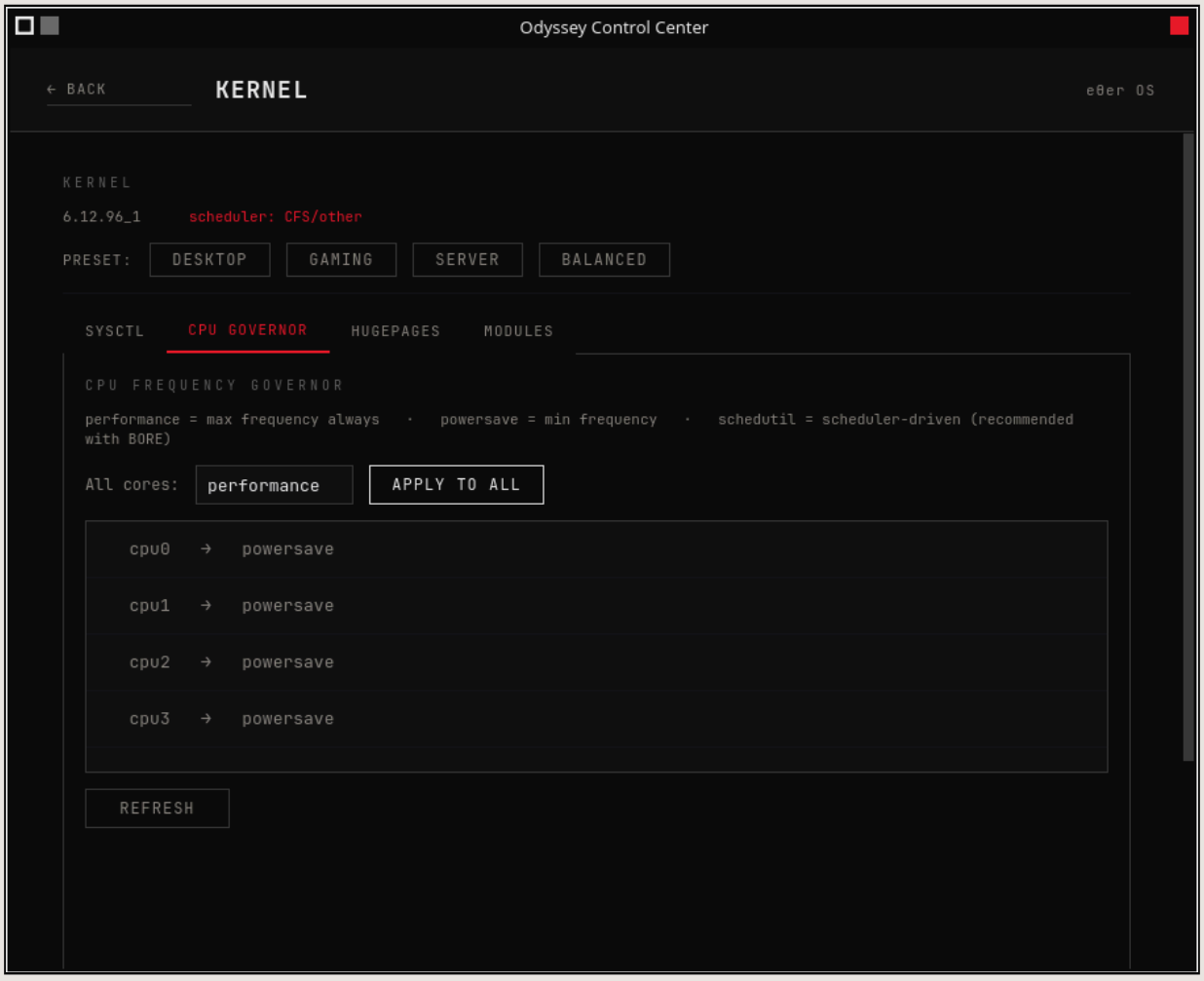

FIG · La pestaña CPU GOVERNOR: "All cores" fijado en performance con APPLY TO ALL, y una lista por núcleo debajo que todavía muestra todos en powersave — una discrepancia esperando ser corregida.

El governor decide con qué agresividad tu CPU cambia la frecuencia de reloj. Un menú desplegable aplica un governor a todos los núcleos a la vez; la lista debajo muestra el governor **en vivo** por núcleo individual — así es como detectas un núcleo atascado en el ajuste equivocado, exactamente como en la captura de arriba donde "All cores" dice performance pero cada núcleo debajo todavía marca powersave, lo que significa que APPLY TO ALL aún no se ha pulsado.

GOVERNOR	COMPORTAMIENTO
performance	Frecuencia máxima, siempre — más responsivo, más consumo
powersave	Frecuencia mínima, siempre — más silencioso, menos responsivo
schedutil	Frecuencia guiada por la propia visión de carga del planificador — recomendado con el planificador BORE

Solo los governors que tu CPU y kernel específicos realmente exponen aparecen en el menú desplegable.

3.4 Guía visual — hugepages

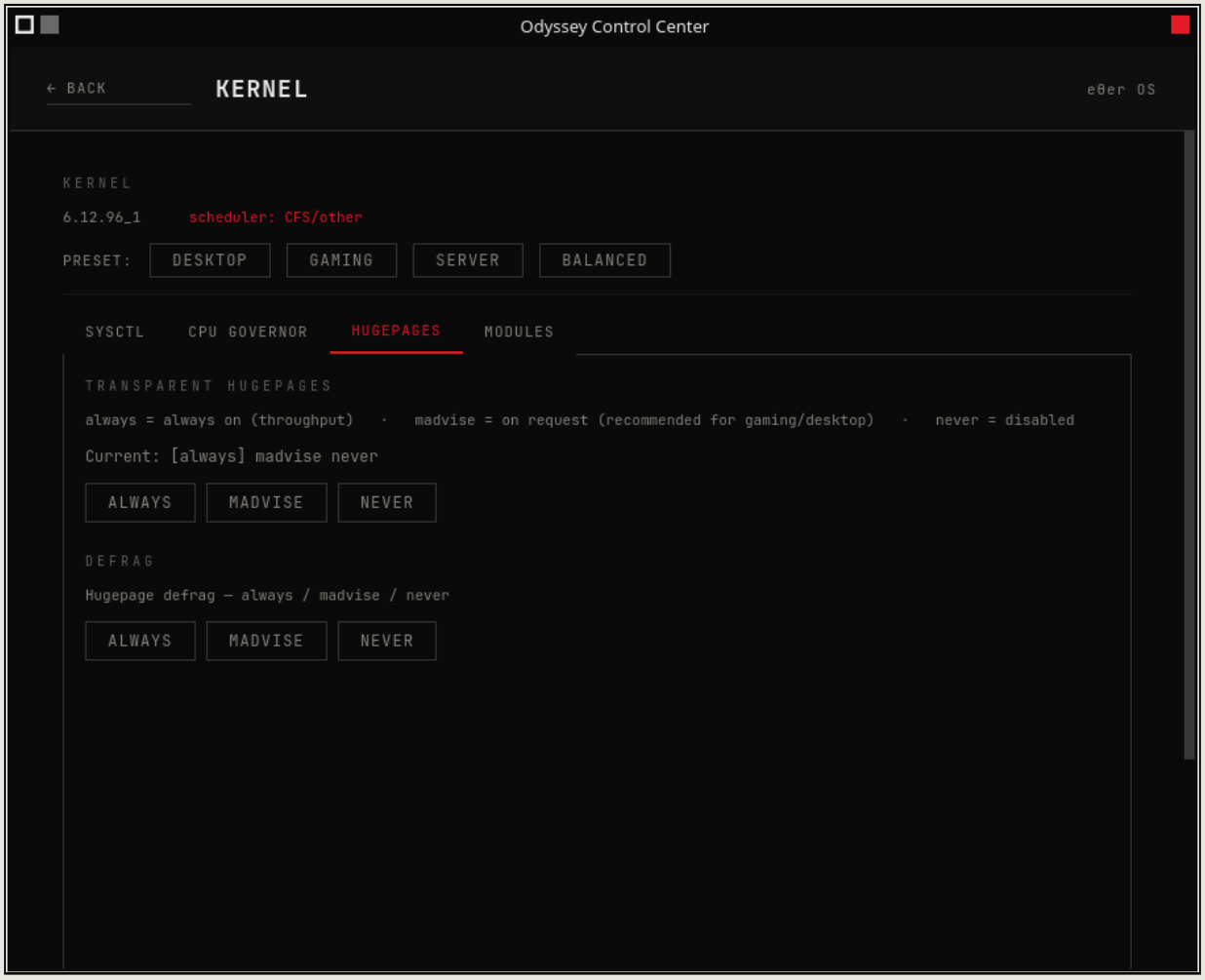

FIG · La pestaña HUGEPAGES: Transparent Hugepages mostrando "Current: [always] madvise never" con ALWAYS resaltado, y la misma fila de tres botones de nuevo para Defrag debajo.

Dos ajustes independientes, cada uno una fila de tres botones aplicados al instante del clic — el valor en vivo actual siempre se muestra entre corchetes sobre los botones, como `[always] madvise never` en la captura.

AJUSTE	QUÉ CONTROLA
Transparent hugepages	ALWAYS = siempre activo en todas partes (máximo throughput); MADVISE = solo cuando una app lo pide explícitamente (recomendado para gaming/escritorio); NEVER = completamente desactivado
Defrag	Con qué agresividad el kernel compacta la memoria para **formar** hugepages bajo petición — las mismas tres opciones, independiente del ajuste de arriba

3.5 Guía visual — modules

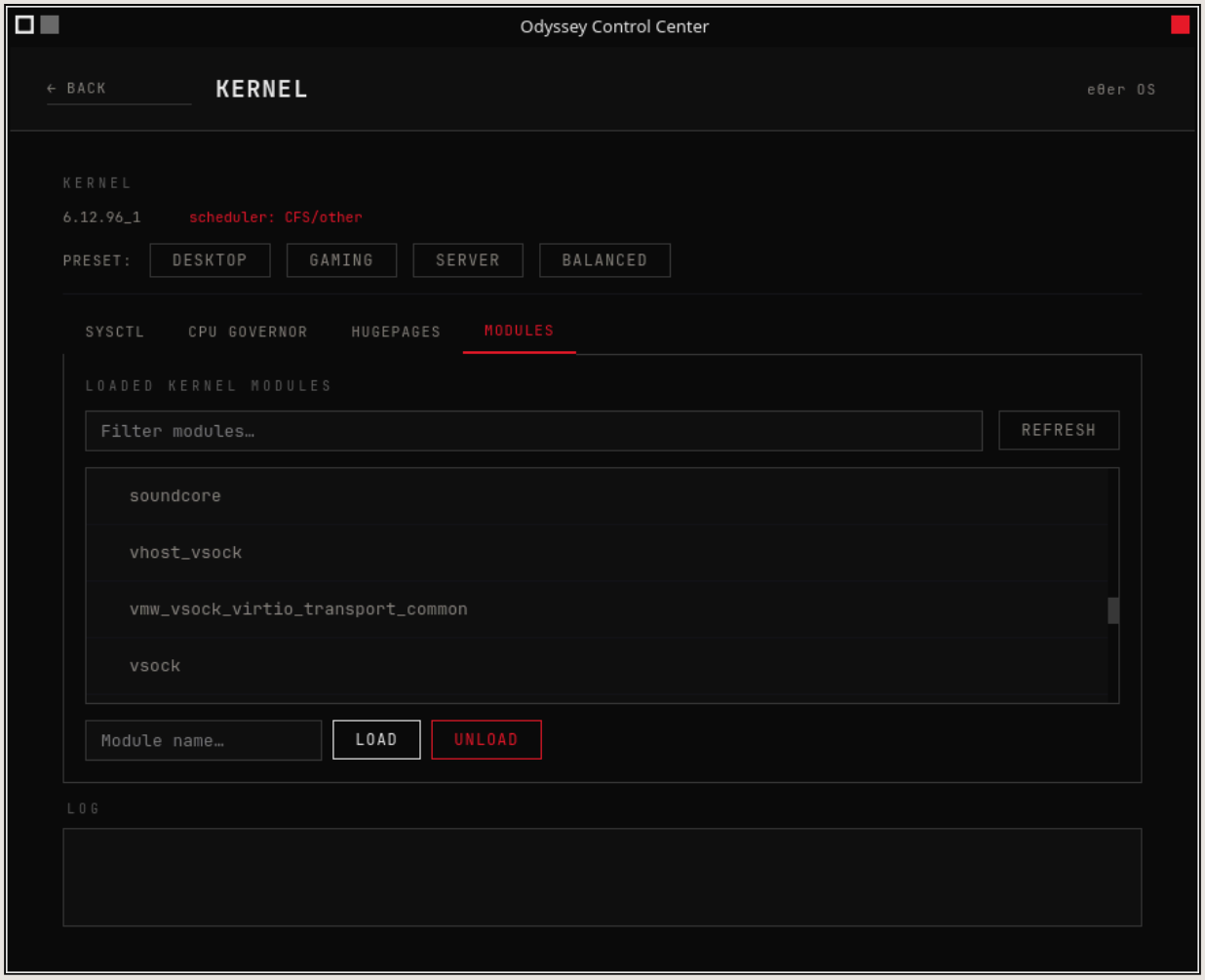

FIG · La pestaña MODULES: un campo de filtro, una lista en vivo (soundcore, vhost_vsock, vsock…), y una fila LOAD/UNLOAD al final para un módulo escrito por nombre.

Una lista en vivo y filtrable de cada módulo del kernel actualmente cargado — escribe en el filtro para acotarla, REFRESH para volver a leerla. A diferencia de las otras tres pestañas, esta te permite actuar directamente: escribe un nombre de módulo y haz clic en **LOAD** o **UNLOAD**.

WHAT	Kernel → pestaña MODULES → escribe un nombre de módulo → LOAD o UNLOAD.
WHY	Ocasionalmente necesitas un driver o módulo de función específico activo (o eliminado) sin reiniciar — probar hardware, liberar un dispositivo que otro módulo está reteniendo.
RESULT	El módulo se inserta o elimina del kernel en ejecución de inmediato.
RISK	Moderado específicamente para UNLOAD — eliminar un módulo del que depende otra cosa puede desestabilizar lo que lo estaba usando (un driver de pantalla, un sistema de archivos). LOAD es en general seguro; el kernel se niega directamente a cargar un módulo incompatible en lugar de aplicarlo a medias.

▸ **ADVANCED**

Past this point: config file paths, raw commands, and what the panel is doing underneath.

3.6 Advanced

LAS CLAVES SYSCTL EXACTAS

CLAVE	ETIQUETA EN EL PANEL
`kernel.sched_autogroup_enabled`	Autogroup
`kernel.sched_child_runs_first`	Child runs first
`kernel.nmi_watchdog`	NMI Watchdog
`kernel.timer_migration`	Timer migration
`kernel.perf_event_paranoid`	Perf paranoid
`vm.swappiness`	Swappiness
`vm.vfs_cache_pressure`	VFS cache pressure
`vm.dirty_ratio` / `vm.dirty_background_ratio`	Dirty ratio % / Dirty background ratio %
`vm.compaction_proactiveness`	Compaction proactiveness
`vm.watermark_boost_factor`	Watermark boost
`net.ipv4.tcp_fastopen`	TCP Fast Open
`net.core.netdev_max_backlog`	Netdev max backlog
`net.core.rmem_max`	Socket rmem max
`kernel.kptr_restrict` / `kernel.dmesg_restrict`	kptr restrict / dmesg restrict

VALORES DE LOS PREAJUSTES, DESKTOP VS GAMING

```
# DESKTOP
$ sysctl vm.swappiness=10 vm.vfs_cache_pressure=50 kernel.timer_migration=1
# GAMING
$ sysctl vm.swappiness=1 vm.vfs_cache_pressure=50 kernel.timer_migration=0
# las tablas completas de los preajustes se leen directamente de panel_kernel.py —
# abre el código fuente del Control Center si quieres cada campo
```

A MANO, SIN EL PANEL

```
# en vivo, temporal
$ sudo sysctl vm.swappiness=10
# persistente — el mismo archivo que escribe el panel
$ echo “vm.swappiness=10” | sudo tee -a /etc/sysctl.d/99-odyssey.conf
$ sudo sysctl -system
# CPU governor, por núcleo
$ echo schedutil | sudo tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
# hugepages
$ echo madvise | sudo tee /sys/kernel/mm/transparent_hugepage/enabled
$ echo madvise | sudo tee /sys/kernel/mm/transparent_hugepage/defrag
# módulos
$ sudo modprobe MODULE_NAME
$ sudo modprobe -r MODULE_NAME
```

POR QUÉ TRY NOW Y MAKE PERSISTENT SON LLAMADAS SEPARADAS

Las escrituras de `sysctl` son por naturaleza siempre solo en vivo — el kernel no tiene persistencia incorporada. MAKE PERSISTENT es Odyssey eligiendo un archivo canónico único, `/etc/sysctl.d/99-odyssey.conf`, y haciendo tanto la escritura en vivo como la escritura en archivo juntas para que las dos nunca se desalineen. Editar ese archivo directamente y ejecutar `sysctl --system` después es funcionalmente idéntico a pulsar el botón.

CHAPTER 4

Memory & Swap

Una sola pantalla, tres secciones: un resumen de solo lectura, swappiness, y ZRAM.

4.1 Guía visual

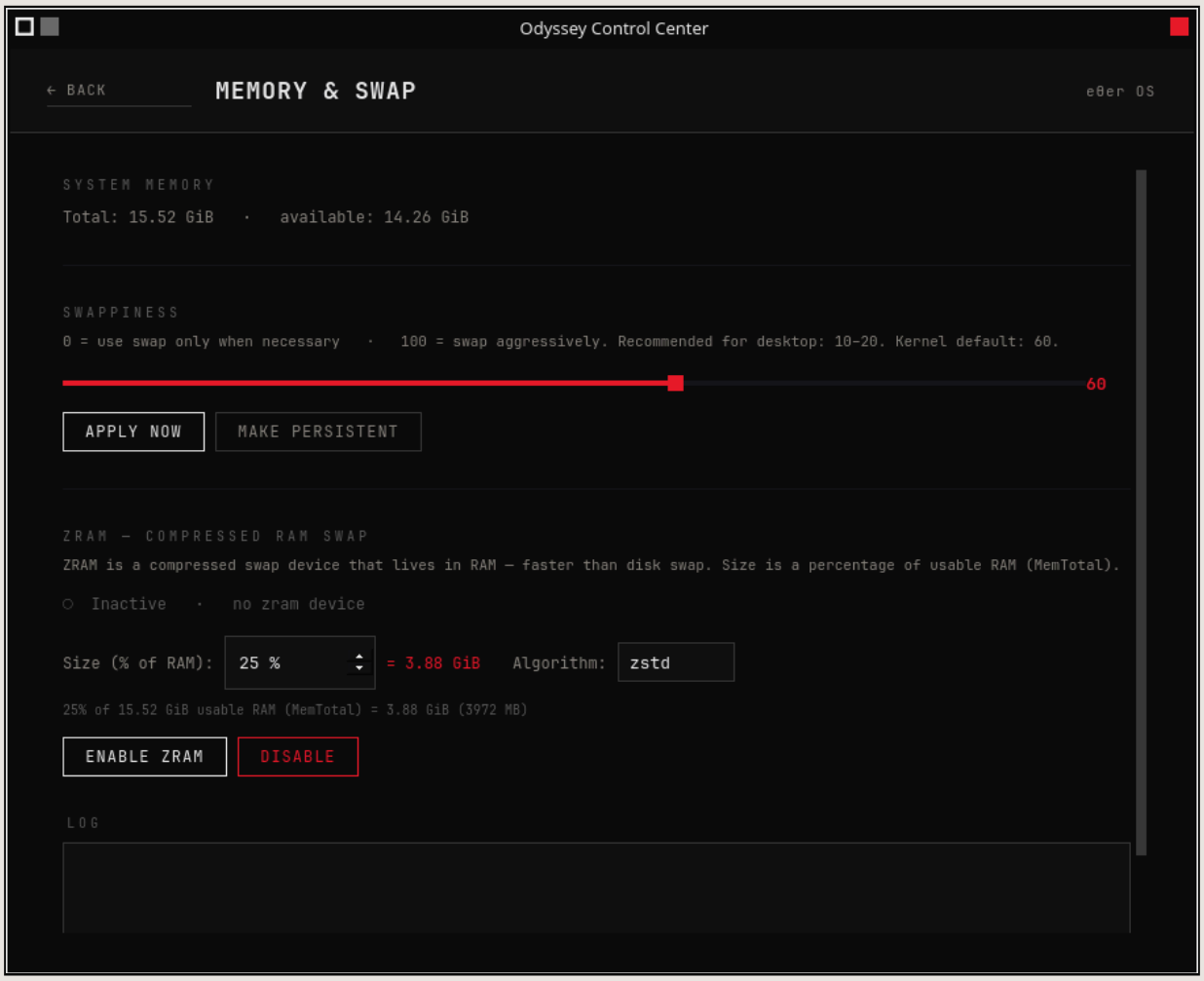

FIG · El panel completo: System memory (15,52 GiB totales, 14,26 GiB disponibles), el deslizador Swappiness en 60 con APPLY NOW / MAKE PERSISTENT, y ZRAM debajo mostrando "Inactive · no zram device" con tamaño, algoritmo y ENABLE/DISABLE.

SYSTEM MEMORY

Solo un resumen en la parte superior — RAM total y qué hay disponible actualmente. Sin controles; es el contexto para las dos secciones de abajo.

SWAPPINESS

Un único deslizador de 0 a 100. 0 significa “usa swap solo cuando realmente no hay otra opción”; 100 significa “usa swap agresivamente, mucho antes de que la RAM esté llena.” El valor por defecto del kernel es 60 (lo que muestra la captura); la propia sugerencia del panel indica 10–20 para un escritorio — el mismo número que fija el preajuste DESKTOP del panel Kernel.

WHAT	Arrastra el deslizador → APPLY NOW para probar, MAKE PERSISTENT una vez que estés satisfecho.
WHY	Un valor más bajo mantiene lo que usas activamente en la RAM rápida en lugar de moverlo al swap más lento — normalmente lo que quieres con suficiente memoria instalada.
RESULT	APPLY NOW lo cambia en vivo, de inmediato, solo para esta sesión. MAKE PERSISTENT lo escribe en un archivo de configuración para que también sobreviva al reinicio.
RISK	Muy bajo. Un valor extremadamente bajo en una máquina realmente escasa de RAM puede hacerla más propensa a quedarse sin memoria de forma abrupta bajo carga pesada en lugar de hacer swap de forma gradual — si eso pasa, súbelo de nuevo.

ZRAM — SWAP COMPRIMIDO EN RAM

ZRAM es un dispositivo de swap que vive **dentro** de la RAM misma, comprimido — notablemente más rápido que el swap a disco porque no hay ningún disco involucrado. El tamaño es un porcentaje de tu RAM utilizable total, no un número fijo, así que el ajuste se adapta con sentido a máquinas diferentes; una lectura en vivo junto al campo de tamaño convierte el porcentaje a GiB por ti (en la captura: 25% = 3,88 GiB en un sistema de 15,52 GiB).

WHAT	Fija el porcentaje de tamaño y elige un algoritmo (zstd, lz4, lzo, o deflate) → ENABLE ZRAM.
WHY	El swap comprimido en RAM absorbe la presión de memoria con un impacto en el rendimiento mucho menor que el swap a disco — útil en cualquier máquina, especialmente una con RAM limitada.
RESULT	Se crea y activa un dispositivo `zram0` con el tamaño y algoritmo elegidos; la línea de estado pasa de “Inactive” a activo.
RISK	Bajo. zstd (el predeterminado) equilibra bien la relación de compresión y el costo de CPU para la mayoría de las máquinas. DISABLE lo desactiva de forma limpia en cualquier momento.

▸ **ADVANCED**

Past this point: config file paths, raw commands, and what the panel is doing underneath.

4.2 Advanced

QUÉ SIGNIFICA REALMENTE “DEPENDE DEL HARDWARE” AQUÍ

Ambos ajustes interactúan con tu hardware específico, no solo con “más RAM = mejor”:

- **SSD NVMe rápido**: una swappiness en el rango 10–20 cuesta muy poco incluso si hace swap — el disco es lo bastante rápido como para que apenas se note. ZRAM sigue ganando en latencia y desgaste del SSD, pero la diferencia es menor que con un disco mecánico.

- **HDD mecánico o eMMC como swap**: mantén la swappiness baja (0–10) — el swap a disco en soportes lentos causa el tirón que se llama "thrashing." ZRAM es una victoria mucho mayor precisamente aquí.
- **Margen de CPU**: ZRAM cambia I/O de disco por ciclos de CPU gastados en comprimir y descomprimir. En una máquina limitada por CPU (hardware más antiguo, o uno ya al límite durante el gaming), `lz4` comprime más rápido que `zstd` con peor relación — un compromiso razonable si el tiempo de CPU es el recurso más ajustado. En cualquier cosa moderna, la mejor relación de `zstd` vale su costo ligeramente mayor.
- **Margen de RAM**: un porcentaje de ZRAM grande (como 50%) en una máquina con solo 8 GiB totales deja menos RAM real para las aplicaciones, ya que el propio ZRAM consume memoria real para almacenar las páginas comprimidas. Con 16 GiB o más, 25% es un valor por defecto cómodo con margen de sobra.

A MANO

```
# swappiness, en vivo
$ sudo sysctl vm.swappiness=15
# swappiness, persistente — el mismo archivo que escribe el panel
$ echo "vm.swappiness=15" | sudo tee /etc/sysctl.d/99-odyssey.conf
# zram, configuración manual
$ sudo modprobe zram
$ echo zstd | sudo tee /sys/block/zram0/comp_algorithm
$ echo 4G | sudo tee /sys/block/zram0/disksize
$ sudo mkswap /dev/zram0
$ sudo swapon /dev/zram0 -p 100
# verifica que esté activo
$ swapon --show
```

POR QUÉ EL PANEL REINICIA ZRAM0 EN LUGAR DE REDIMENSIONARLO EN VIVO

El tamaño de un dispositivo ZRAM no se puede cambiar mientras está activo y en uso como swap — el kernel requiere que primero se desactive y reinicie. El botón ENABLE ZRAM del panel ejecuta esta secuencia automáticamente (swapoff, reset, reconfiguración, swapon), por lo que cambiar el tamaño o el algoritmo en un dispositivo ya activo sigue apareciendo como una única acción limpia en lugar de un error.

CHAPTER 5

Firewall

El panel Firewall tiene su propio capítulo dedicado, en toda su profundidad, en la **Security and Hardening Guide** — el conjunto de reglas explicado línea por línea, los 21 preajustes con sus riesgos individuales, el editor de configuración en bruto, y los equivalentes de línea de comandos. Este capítulo es la versión guiada visualmente.

5.1 Guía visual

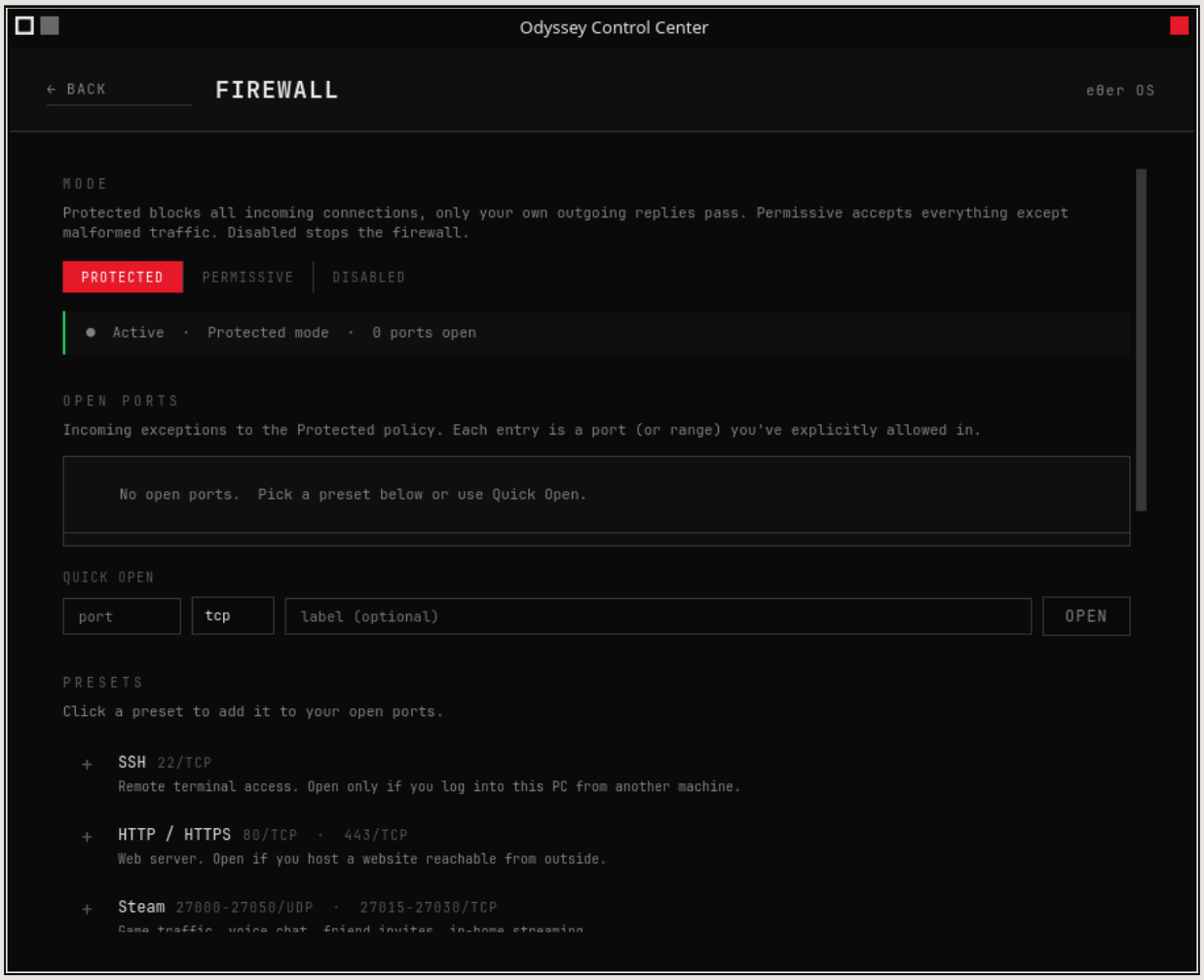

FIG · Modo PROTECTED activo, "0 ports open," los campos Quick Open, y los recuadros de preajuste que empiezan con SSH, HTTP/HTTPS y Steam.

MODE

Tres botones en la parte superior, y el panel explica cada uno en una sola línea justo encima de ellos:

MODO	QUÉ HACE
PROTECTED	El predeterminado. Bloquea todas las conexiones entrantes; solo pasan tus propias respuestas salientes.
PERMISSIVE	Acepta todo excepto tráfico malformado — un ajuste de diagnóstico para "¿es el firewall lo que está rompiendo esto?", no un lugar donde quedarse.
DISABLED	Detiene el firewall por completo.

La línea de estado debajo confirma qué está realmente activo — en la captura, **"Active · Protected mode · 0 ports open."**

OPEN PORTS Y PREAJUSTES

OPEN PORTS lista tus excepciones explícitas — vacío por defecto, como se muestra. **QUICK OPEN** toma puerto, protocolo y etiqueta opcional directamente, para cualquier cosa que no esté en la lista de preajustes. **PRESETS** debajo ofrece ocho servicios comunes a un clic cada uno, cada uno con una línea que explica cuándo realmente lo querrías (SSH: "acceso remoto por terminal, ábrelo solo si te conectas a esta PC desde otra máquina"), más otros trece bajo un "MORE PRESETS" expandible.

WHAT	Firewall → haz clic en un recuadro de preajuste, o rellena QUICK OPEN → SAVE & APPLY.
WHY	Estás ejecutando algo — un servidor de juego, un servidor de medios, un servidor de desarrollo — que otra máquina necesita alcanzar.
RESULT	Ese puerto se abre; todo lo demás permanece cerrado por la política PROTECTED.
RISK	Depende enteramente de qué preajuste — consulta las notas de riesgo por preajuste en la Security and Hardening Guide antes de abrir cualquier cosa más allá de SSH o HTTP/HTTPS en una máquina alcanzable desde internet.

Nada tiene efecto hasta que pulsas **SAVE & APPLY** — los cambios se ponen en espera primero, así puedes revisar exactamente qué estás a punto de abrir antes de confirmarlo.

```
▸ ADVANCED

Past this point: config file paths, raw commands, and what the panel is doing underneath.
```

5.2 Advanced

La referencia técnica completa — el conjunto de reglas por defecto de `nftables.conf` explicado regla por regla, cada uno de los 21 preajustes con su exposición específica y mitigación, recetas para limitar el acceso solo a la LAN, los equivalentes de línea de comandos `nft`, y la resolución de una sesión SSH bloqueada por fuera — vive en el **Capítulo 2 de la Security and Hardening Guide** (`/usr/share/doc/odyssey/` en tu sistema instalado). Este manual no lo duplica deliberadamente; considera ese capítulo la continuación de esta sección.

```
# el comando que vale la pena conocer ahora mismo:
$ sudo nft list ruleset
# muestra exactamente qué está cargado en el kernel, independientemente del panel
```

CHAPTER 6

AppArmor

Cinco pestañas: **STATUS**, **PROFILES**, **COMMUNITY**, **VIOLATIONS**, **EDITOR**. Misma referencia que Firewall: la referencia profunda — qué es realmente el control de acceso obligatorio, el flujo completo de confinamiento, un ejemplo práctico — vive en la Security and Hardening Guide. Esta es la guía visual.

6.1 Guía visual — STATUS

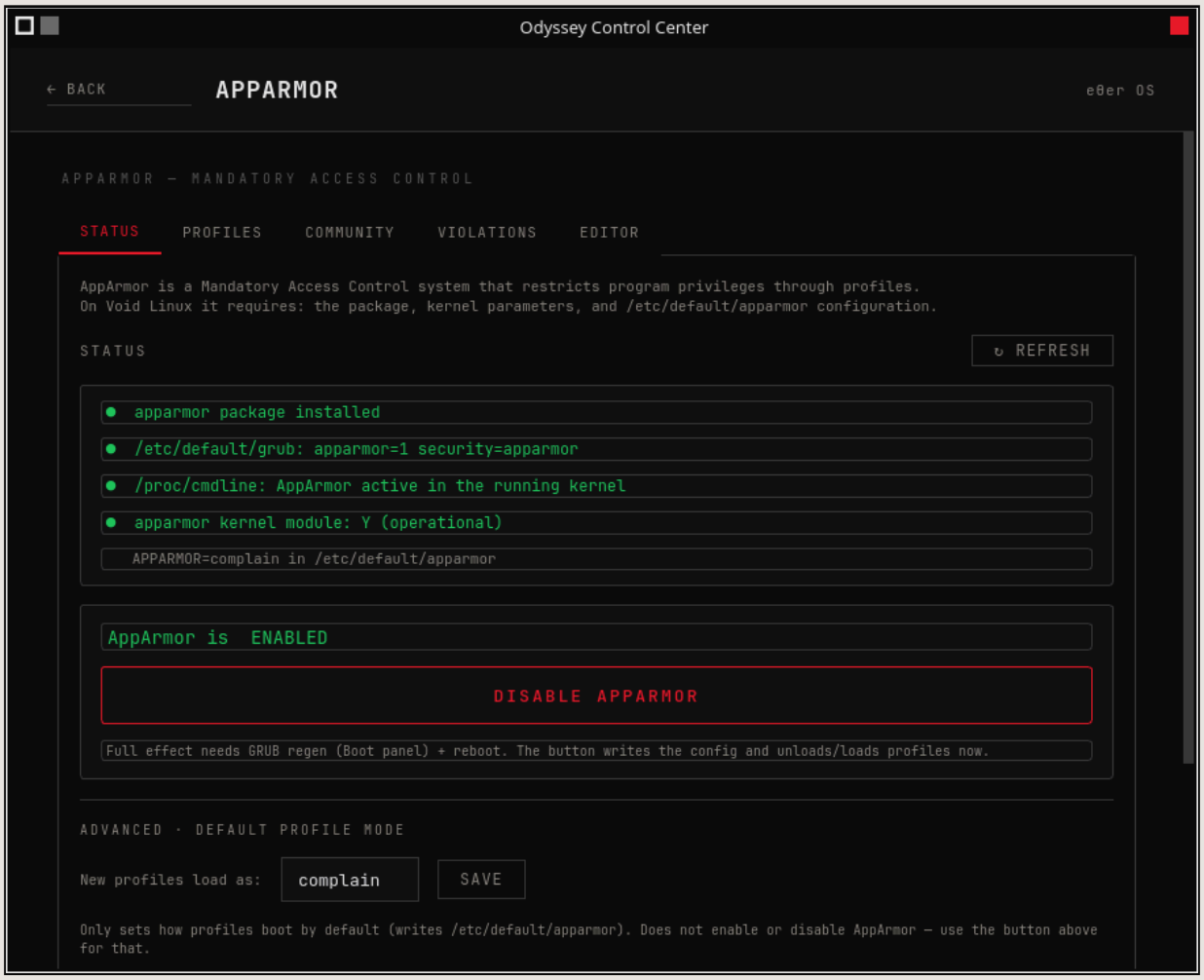

FIG · Pestaña STATUS: cuatro marcas verdes (paquete instalado, parámetros de GRUB, kernel en ejecución, módulo del kernel), "AppArmor is ENABLED" con un botón rojo DISABLE APPARMOR, y el modo de perfil predeterminado fijado en complain.

Cuatro comprobaciones verifican que el framework realmente funcione, de arriba a abajo: el paquete `apparmor` está instalado, los parámetros del kernel en GRUB están fijados, el kernel **en ejecución** realmente arrancó con ellos (esta es la que puede quedar atrás si no has reiniciado después de habil-

itarlos), y el propio módulo del kernel está operativo. Debajo: un botón grande de habilitar/deshabilitar, y **ADVANCED · DEFAULT PROFILE MODE** — si los perfiles recién cargados arrancan en complain o enforce.

WHAT	AppArmor → STATUS → confirma que las cuatro comprobaciones estén en verde.
WHY	Si alguna comprobación está en rojo o ámbar, nada más en este panel puede funcionar correctamente — es siempre lo primero que hay que mirar cuando algo con AppArmor parece ir mal.
RESULT	Certeza de que el framework está genuinamente activo, no solo instalado.
RISK	Ninguno — verificación de solo lectura.

6.2 Guía visual — PROFILES

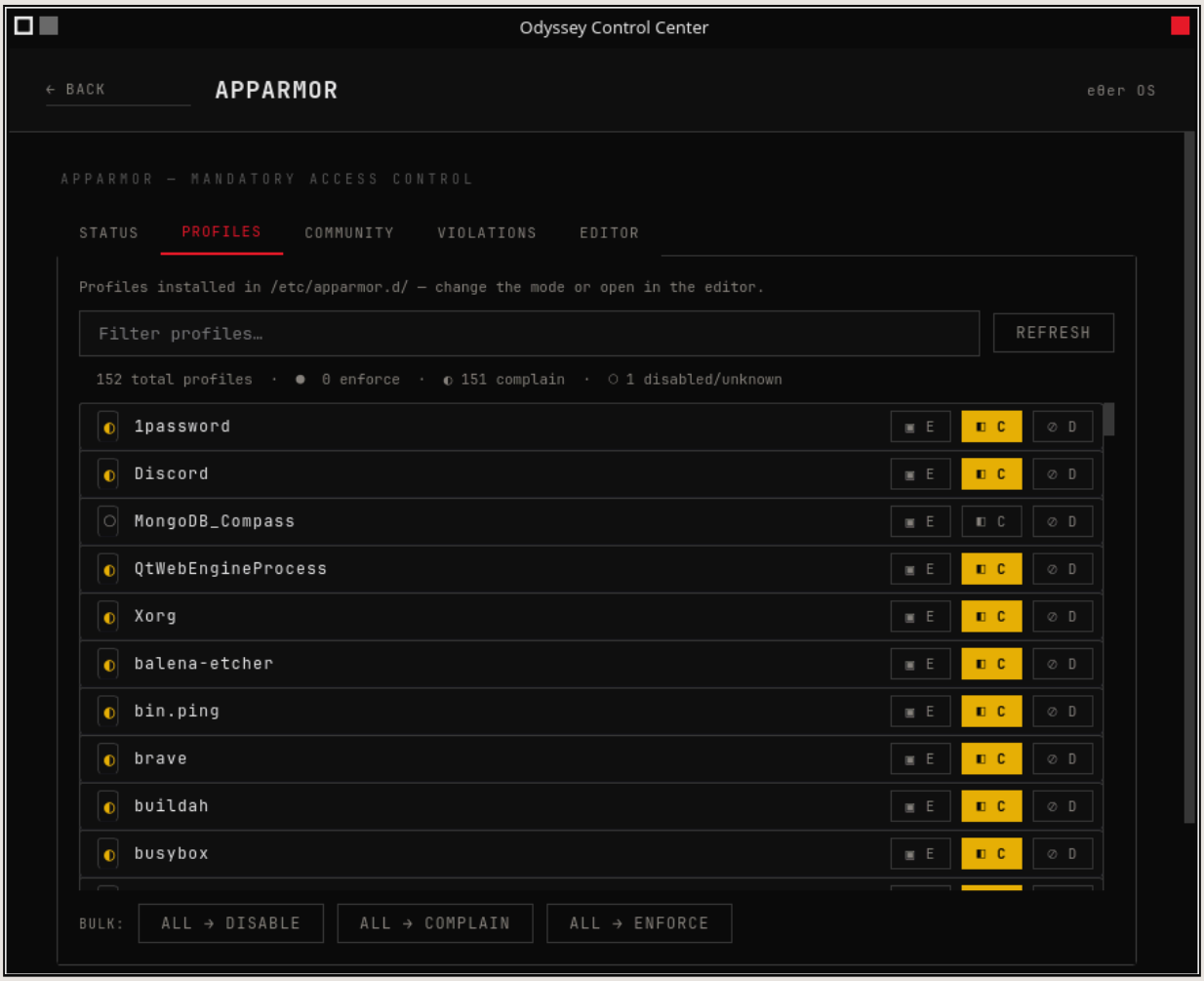

FIG · Pestaña PROFILES: 152 perfiles totales, 0 enforce, 151 complain, 1 disabled/unknown — una lista filtrable con 1password, Discord, MongoDB_Compass y más, cada uno con botones E / C / D, y en la parte inferior ALL→DISABLE / ALL→COMPLAIN / ALL→ENFORCE.

Cada perfil cargado, una fila cada uno, con un campo de filtro y un recuento en la parte superior (en la captura: 152 perfiles, todos menos uno en modo complain — el predeterminado de Odyssey). Tres botones por fila:

BOTÓN	EFECTO
E	Cambia este perfil a enforce — las violaciones se bloquean, no solo se registran
C	Cambia a complain — registra violaciones, no bloquea nada
D	Deshabilita el perfil por completo — el programa corre sin confinar

Los tres botones **BULK** al final aplican el mismo cambio a cada perfil de la lista a la vez — útil justo después de instalar un lote desde COMMUNITY, pero trata ALL→ENFORCE con especial cautela; ve el bloque de acción en la siguiente sección.

6.3 Guía visual — COMMUNITY

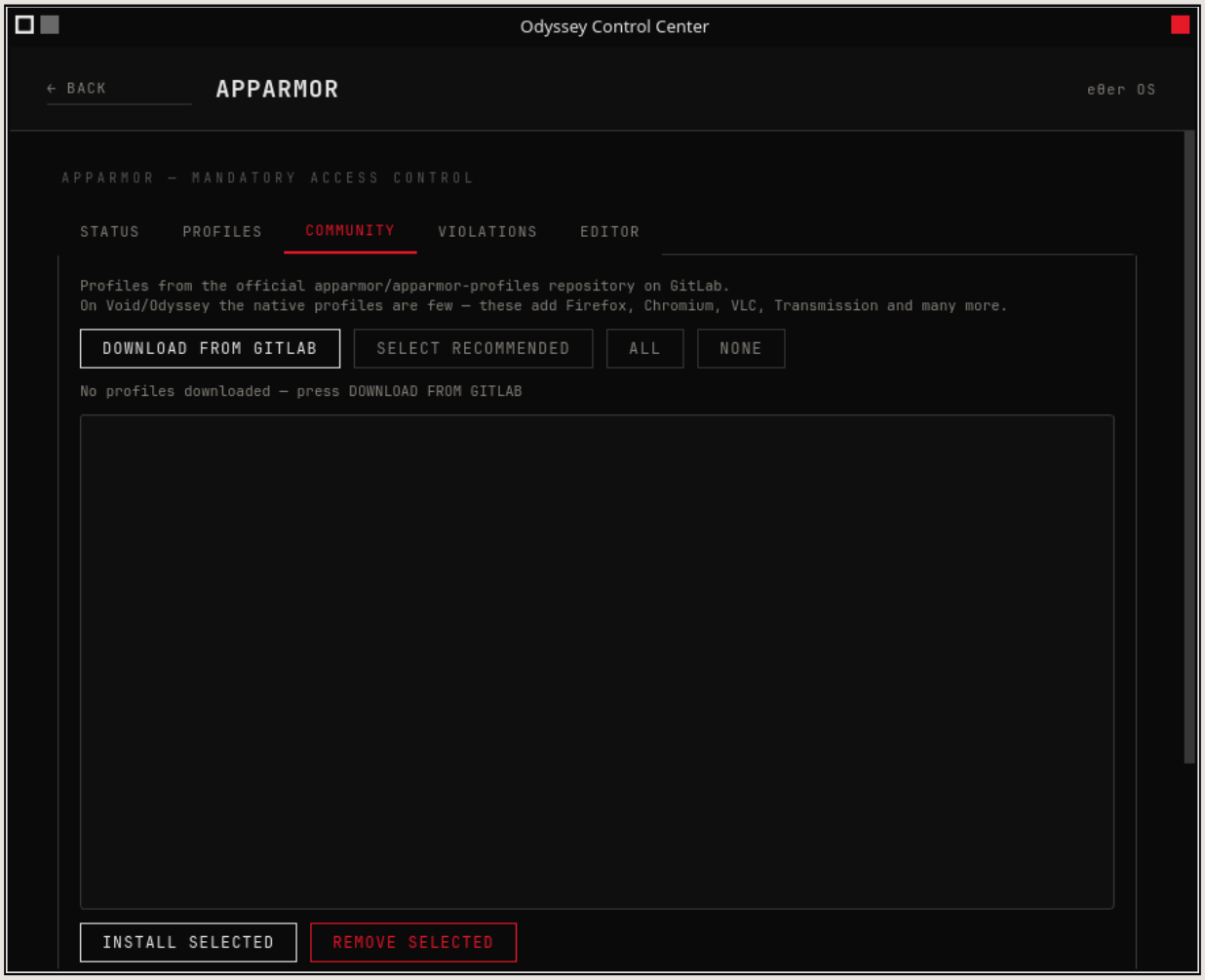

FIG · Pestaña COMMUNITY: "No profiles downloaded — press DOWNLOAD FROM GITLAB," con los botones DOWNLOAD FROM GITLAB, SELECT RECOMMENDED, ALL y NONE, e INSTALL SELECTED / REMOVE SELECTED al final.

AppArmor de fábrica confina sobre todo componentes del sistema — las aplicaciones de escritorio que realmente te preocupan (navegadores, apps de chat, reproductores multimedia) necesitan perfiles de la comunidad. Esta pestaña los obtiene directamente del repositorio oficial `apparmor/apparmor-profiles` en GitLab.

1 DOWNLOAD FROM GITLAB

Obtiene la lista actual de perfiles disponibles — todavía no se instala nada, esto solo puebla la lista.

2 SELECT RECOMMENDED (o ALL / NONE)

Marca un conjunto predeterminado sensato para escritorio — navegadores de la familia Firefox, Chromium, VLC, Transmission y similares.

3 INSTALL SELECTED

Instala los perfiles marcados — se cargan en modo complain, igual que todo lo demás, listos para el ciclo observar-luego-aplicar de abajo.

WHAT	AppArmor → COMMUNITY → DOWNLOAD FROM GITLAB → SELECT RECOMMENDED (o elige individualmente) → INSTALL SELECTED.
WHY	Las aplicaciones más expuestas a entradas no confiables — un navegador que renderiza una página maliciosa, un reproductor multimedia que abre un archivo manipulado — son exactamente las que un perfil protege mejor.
RESULT	Aparecen nuevos perfiles en PROFILES, en modo complain, registrando qué habrían bloqueado sin bloquear todavía nada.
RISK	Ninguno en la instalación — nada se aplica hasta que tú lo decides en PROFILES.

6.4 Guía visual — VIOLATIONS

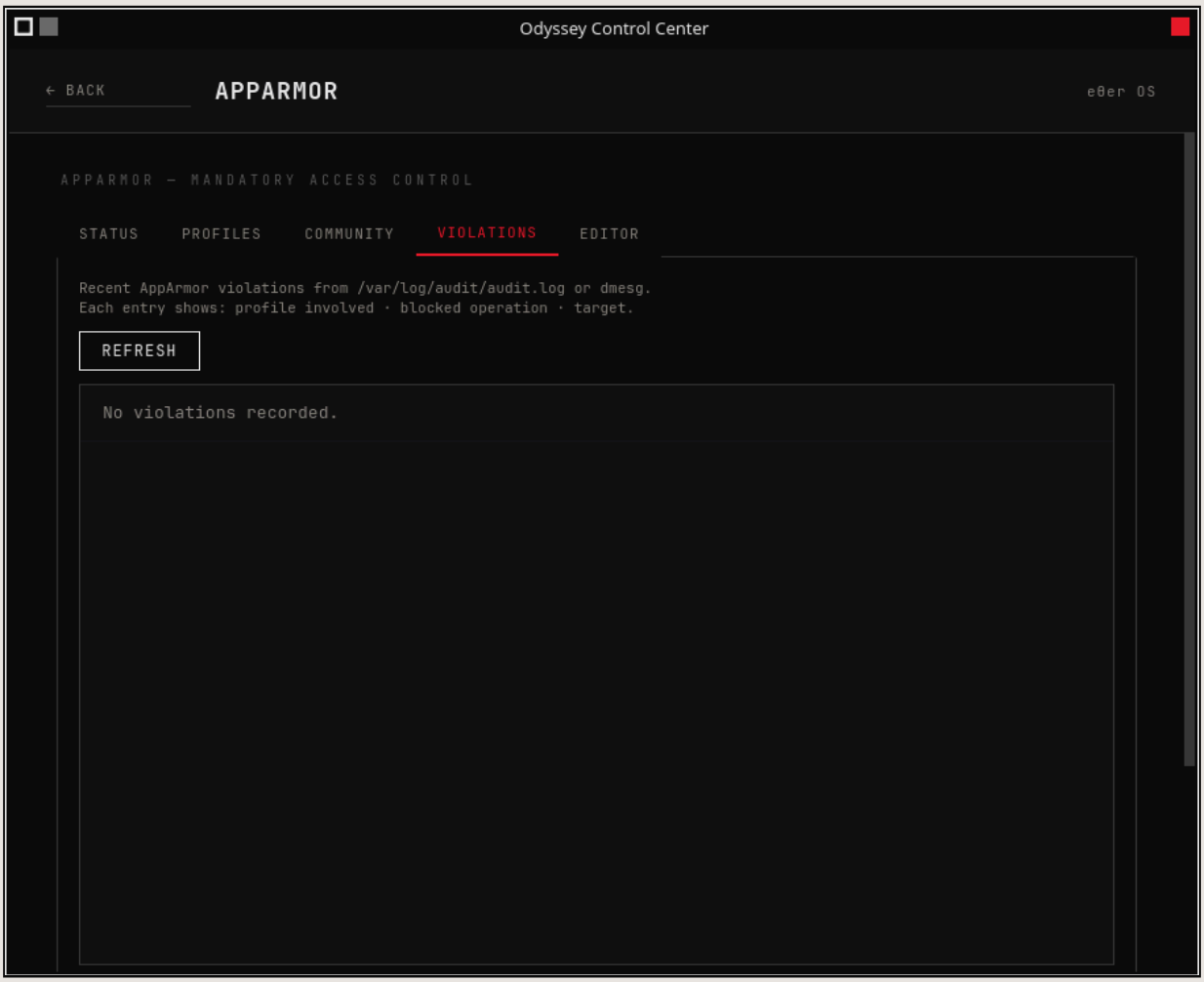

FIG · Pestaña VIOLATIONS: "No violations recorded," con un botón REFRESH y una nota de que las entradas provienen de /var/log/audit/audit.log o dmesg.

Esta pestaña lee el registro de auditoría de AppArmor del sistema y muestra qué un perfil **habría** bloqueado (en modo complain) o **bloqueó** (en modo enforce) — cada entrada indica el perfil, la operación bloqueada, y el archivo o recurso de destino.

> **UNA LISTA VACÍA ES LA LECTURA ESPERADA EN UN SISTEMA SANO**
>
> La captura muestra exactamente lo que verás típicamente: ninguna violación. Eso no es un error ni una señal de que el panel no funcione — significa que cada programa perfilado se ha mantenido dentro de los límites permitidos desde la última vez que se revisó el registro. Las violaciones aparecen aquí en el momento en que algo perfilado intenta hacer algo que su perfil no permite; ve a usar el programa en cuestión, vuelve, y haz REFRESH.

Una vez que aparecen entradas, hacer doble clic en una es el camino rápido hacia EDITOR con ese perfil exacto ya abierto — ve abajo.

6.5 Guía visual — EDITOR

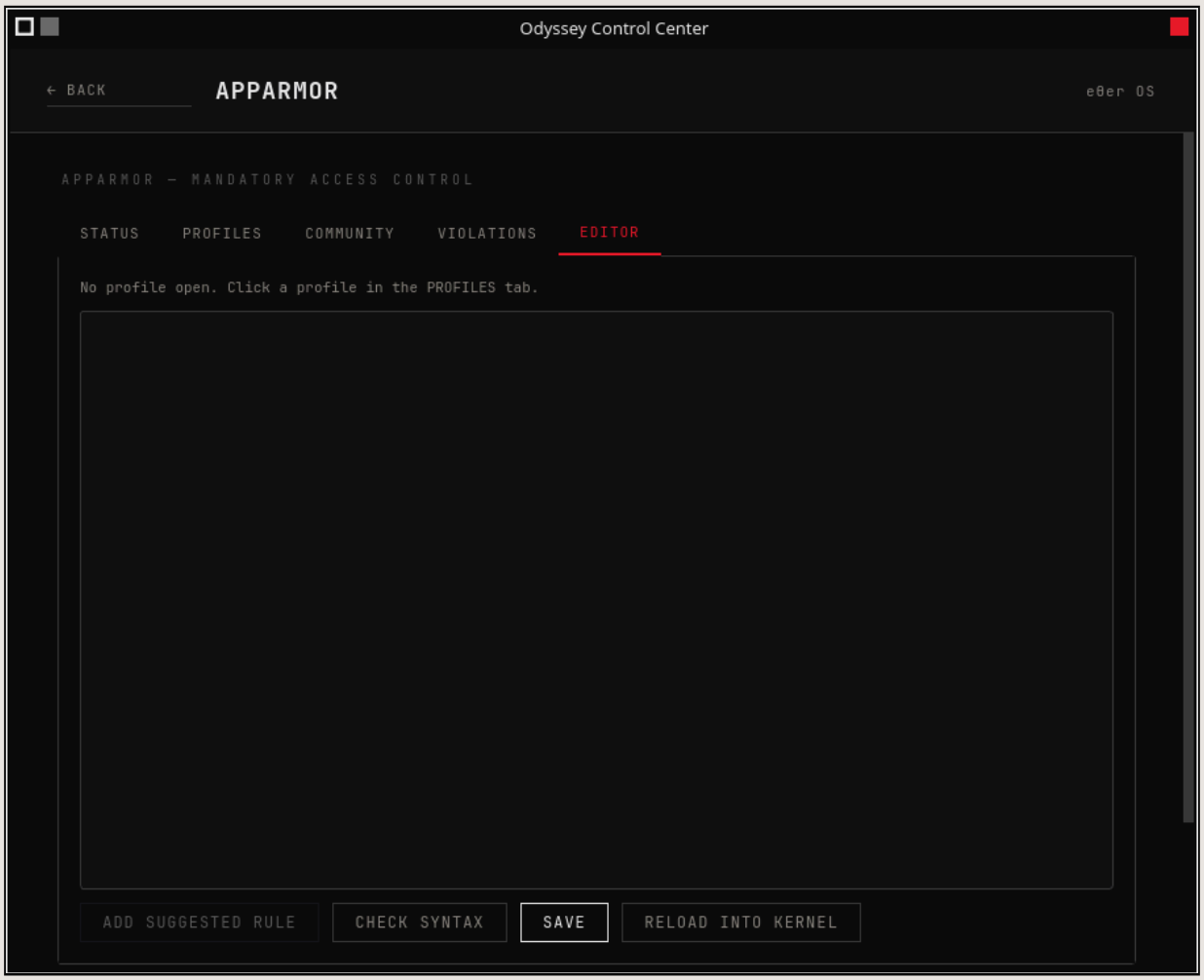

FIG · Pestaña EDITOR: "No profile open. Click a profile in the PROFILES tab," un área de texto vacía, y cuatro botones — ADD SUGGESTED RULE, CHECK SYNTAX, SAVE, RELOAD INTO KERNEL — todos menos SAVE desactivados.

El editor se abre deliberadamente **vacío**, como se muestra — solo edita un perfil a la vez, y eliges cuál desde PROFILES (haz clic en una fila) o desde VIOLATIONS (doble clic en una entrada). Esto evita el error fácil de editar el archivo equivocado por accidente.

BOTÓN	QUÉ HACE
ADD SUGGESTED RULE	Activo solo si llegaste desde una entrada de VIOLATIONS — inserta la línea de regla exacta que habría permitido la acción registrada
CHECK SYNTAX	Valida el perfil sin cargarlo — atrapa un error tipográfico antes de que importe
SAVE	Escribe tus ediciones en el archivo del perfil en disco
RELOAD INTO KERNEL	Hace efectiva una edición guardada de inmediato, sin esperar al próximo arranque o reinicio del servicio

WHAT	PROFILES → haz clic en el nombre de un perfil → EDITOR lo abre → haz una edición → CHECK SYNTAX → SAVE → RELOAD INTO KERNEL.
WHY	Este es el ciclo completo de confinamiento en una línea: observa en modo complain (VIOLATIONS), corrige lo que realmente falta (EDITOR), solo entonces pasa ese perfil a enforce (PROFILES).
RESULT	El perfil se comporta exactamente como se editó, en vivo, sin reiniciar.
RISK	Editar un perfil incorrectamente puede hacer que se bloquee algo legítimo más tarde bajo enforce — CHECK SYNTAX detecta reglas malformadas antes de SAVE, pero no los errores lógicos. Si un perfil que editaste empieza a causar problemas bajo enforce, vuélvelo a complain desde PROFILES y revisa VIOLATIONS de nuevo.

▸ ADVANCED

Past this point: config file paths, raw commands, and what the panel is doing underneath.

6.6 Advanced

EQUIVALENTES DE LÍNEA DE COMANDOS

```
$ sudo aa-status
# los mismos recuentos del encabezado de la pestaña PROFILES, desde la terminal
$ sudo aa-enforce /etc/apparmor.d/PROFILE_FILE
$ sudo aa-complain /etc/apparmor.d/PROFILE_FILE
$ sudo aa-disable /etc/apparmor.d/PROFILE_FILE
# equivalente a RELOAD INTO KERNEL después de una edición manual
$ sudo apparmor_parser -r /etc/apparmor.d/PROFILE_FILE
```

LEER EL REGISTRO DE AUDITORÍA EN BRUTO

```
$ sudo grep apparmor /var/log/audit/audit.log | tail -20
# o, si no hay un daemon de auditoría dedicado en ejecución:
$ sudo dmesg | grep -i apparmor
```

Campos útiles a conocer al leer una línea a mano: `apparmor="DENIED"` (enforce) o `apparmor="ALLOWED"` (complain, solo registrado), `operation=` (qué se intentó), `profile=` (qué perfil decidió — coincide con lo que abre EDITOR), `name=` (la ruta o el recurso), `comm=` (el proceso).

POR QUÉ ALL→ENFORCE EN BLOQUE ES ARRIESGADO

Escalar un perfil individualmente después de que su registro VIOLATIONS haya permanecido en silencio durante un tiempo es el patrón seguro — sabes, perfil por perfil, que el uso normal no lo activa. ALL→ENFORCE se salta ese período de observación para todos los perfiles a la vez, incluidos los que quizás acabas de instalar desde COMMUNITY y nunca has usado de verdad. Un perfil aplicado antes de haber sido vivido puede bloquear una función legítima en el mismo momento en que la usas por primera vez. Prefiere el ciclo de uno en uno; reserva las acciones en bloque para ALL→COMPLAIN (siempre seguro — solo afloja) o ALL→DISABLE al hacer troubleshooting.

ARCHIVOS QUE TOCA ESTE PANEL

RUTA	FUNCIÓN
`/etc/apparmor.d/`	Un archivo de perfil por cada programa confinado — lo que lista PROFILES y abre EDITOR
`/etc/default/apparmor`	Modo predeterminado al arrancar (`APPARMOR=complain`) y el estado enable/disable
`/etc/default/grub`	Los parámetros del kernel `apparmor=1 security=apparmor` que STATUS comprueba

CHAPTER 7

Privacy

Dos herramientas independientes, cada una con su propio flujo en tres pasos install → enable → running, y un puente entre ambas una vez que las dos están en ejecución.

7.1 Guía visual — SearXNG, paso a paso

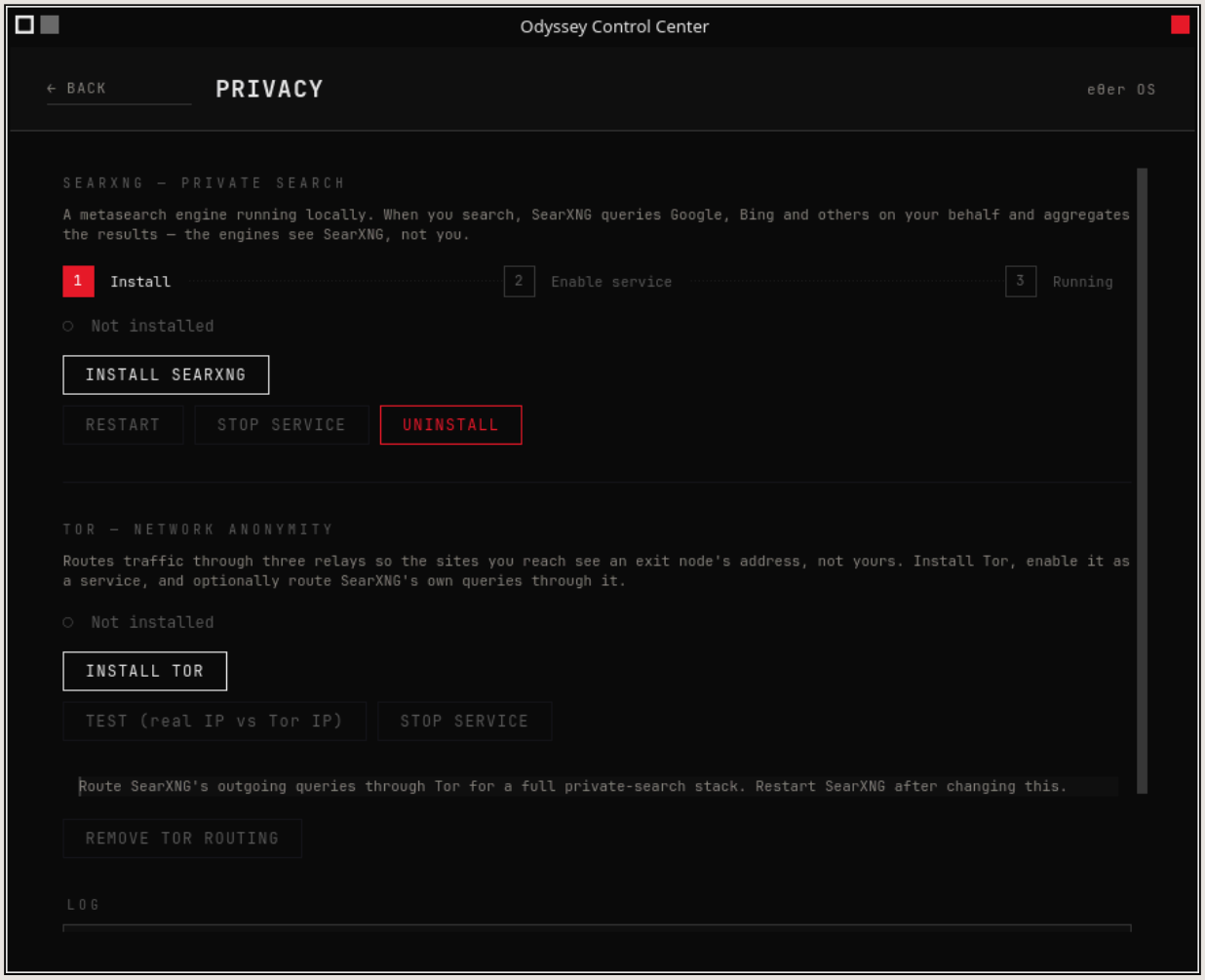

FIG · El panel en reposo: el rastreador de SearXNG en el paso 1 “Install,” estado “Not installed,” un solo botón INSTALL SEARXNG; el rastreador de Tor también muestra “Not installed” con INSTALL TOR debajo.

SearXNG es un motor de metabúsqueda que corre localmente en tu propia máquina — cuando buscas, es SearXNG quien consulta a Google, Bing y los demás en tu nombre, así esos motores ven a SearXNG hacer la solicitud, nunca a ti directamente.

1 **INSTALL SEARXNG**

Haz clic. El registro muestra la instalación real ocurriendo — clonación del código fuente, creación de un usuario de sistema, configuración de un entorno virtual de Python.

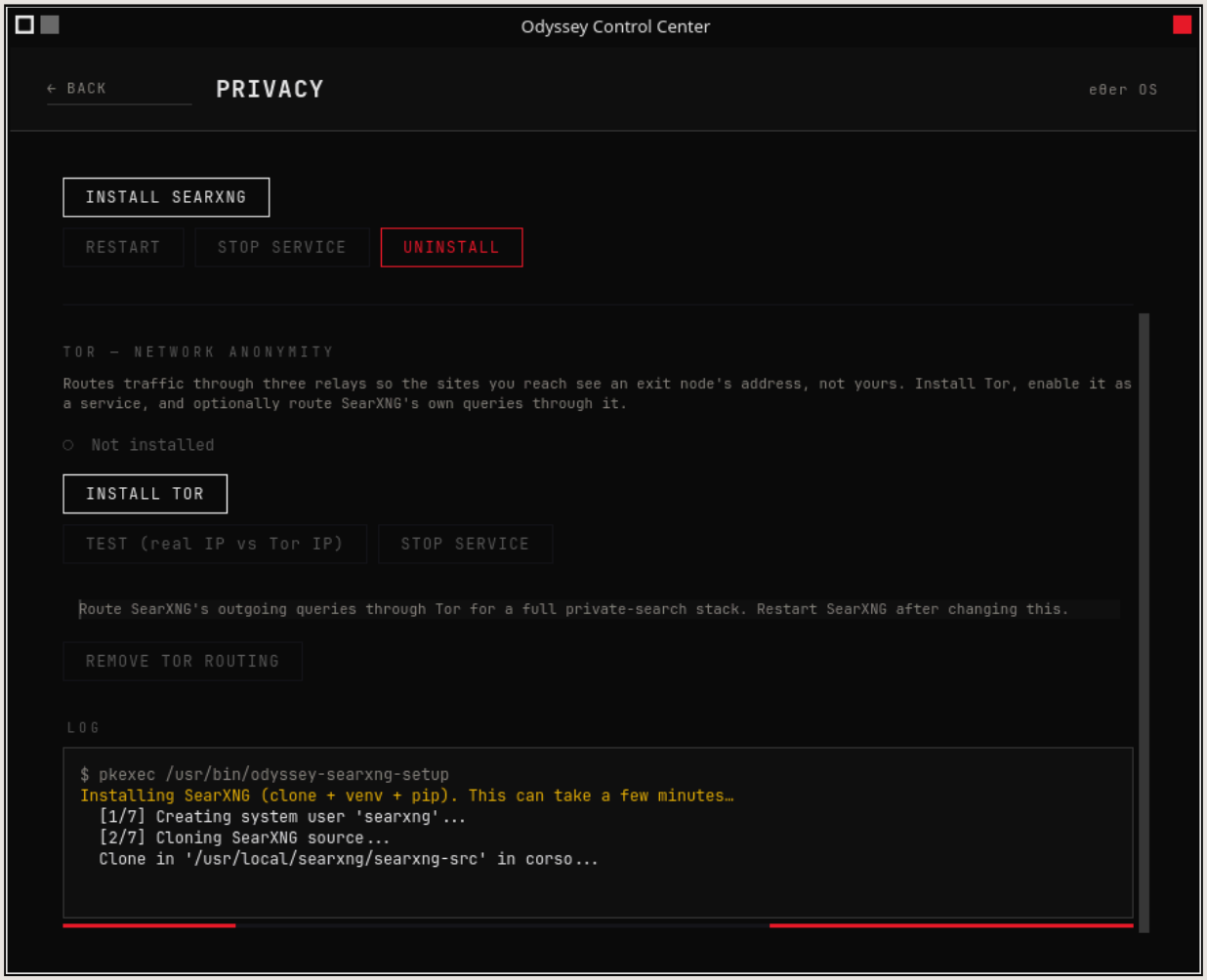

FIG · A mitad de instalación: el registro transmitiendo salida real — “Installing SearXNG (clone + venv + pip). This can take a few minutes…” con los pasos 1/7 y 2/7 visibles.

2 ENABLE SERVICE

Una vez instalado, el rastreador pasa al paso 2 y la línea de estado se vuelve amarilla: “Installed — service not enabled.” Un solo botón ahora: ENABLE SERVICE.

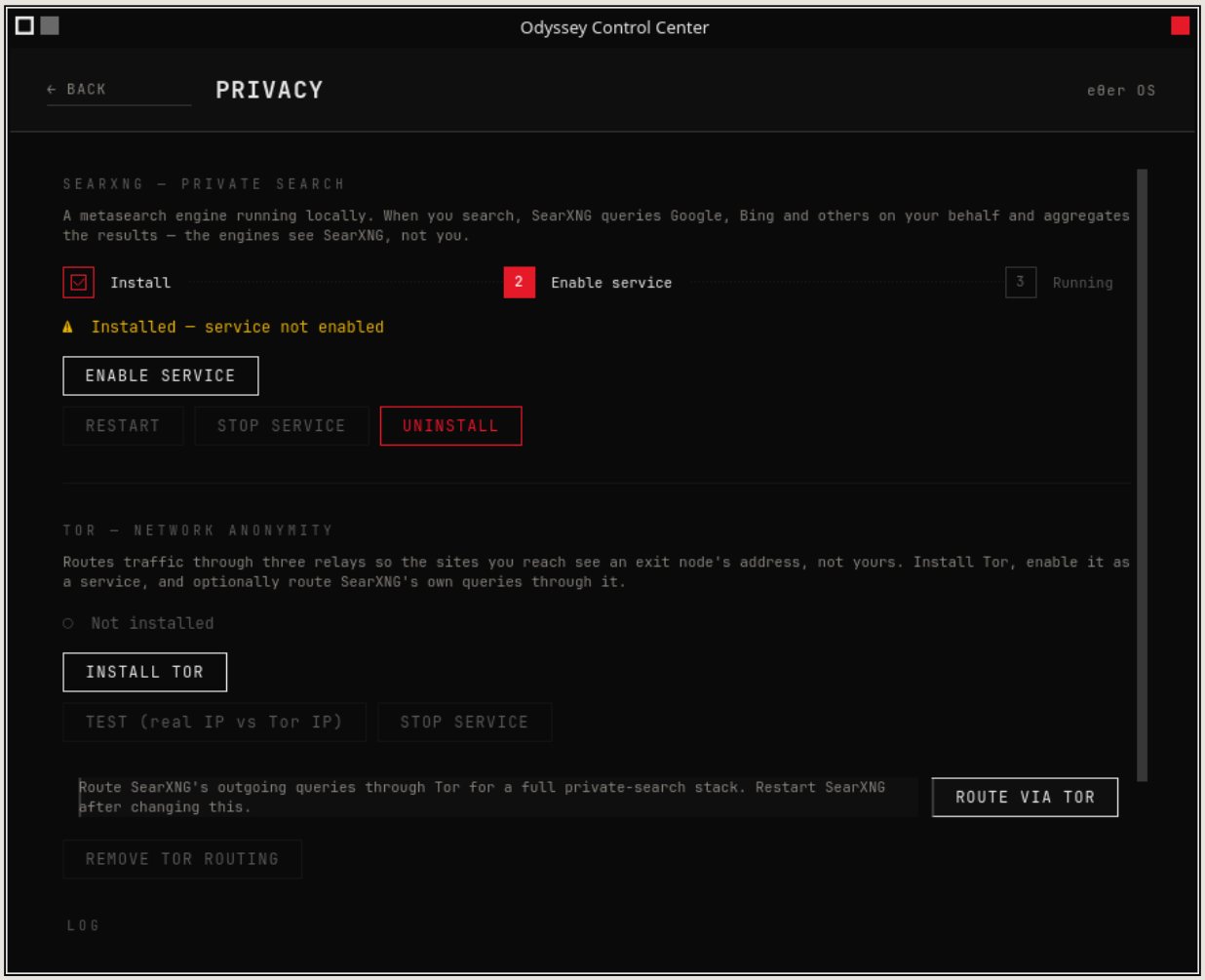

FIG · Paso 2 activo: “Installed — service not enabled,” ENABLE SERVICE como único botón activo, RESTART/STOP SERVICE/UNINSTALL todavía desactivados debajo.

3 Running

Después de habilitar, el rastreador llega al paso 3 y toda la fila de control se activa. Una vez en ejecución, **OPEN IN BROWSER** te lleva directamente ahí — SearXNG escucha en `http://localhost:8888` por defecto.

WHAT	Privacy → INSTALL SEARXNG → ENABLE SERVICE → OPEN IN BROWSER.
WHY	Cada motor de búsqueda comercial construye un perfil a partir de tus consultas a lo largo del tiempo; una instancia de metabúsqueda local significa que solo tú ves alguna vez la consulta sin agregar.
RESULT	Un motor de búsqueda privado corriendo en tu propia máquina, accesible en tu navegador, consultando a los grandes motores en tu nombre sin exponerte a ellos directamente.
RISK	Bajo — es un servicio web local como cualquier otro. RESTART si se comporta mal, STOP SERVICE para pausarlo sin eliminarlo, UNINSTALL para eliminarlo por completo.

7.2 Guía visual — Tor, paso a paso

La misma forma en tres pasos, rastreador independiente, justo debajo de la sección de SearXNG.

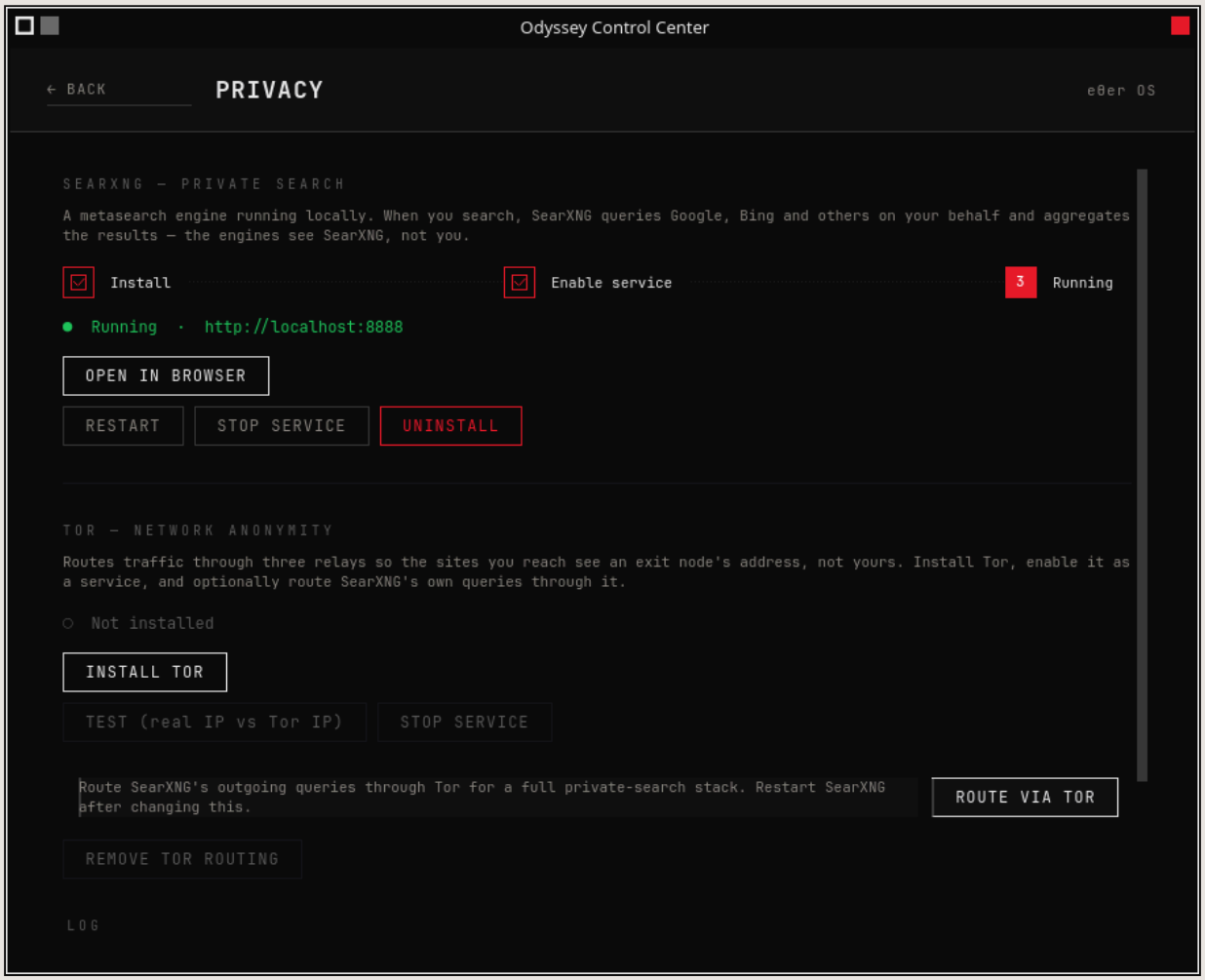

FIG · SearXNG completamente en ejecución arriba; la sección Tor debajo todavía en “Not installed” con INSTALL TOR como único botón activo.

Tor enruta tu tráfico a través de tres relés antes de que llegue a su destino, así el sitio que visitas ve la dirección del relé de salida, nunca la tuya.

1 **INSTALL TOR**

El registro confirma la instalación del paquete y muestra el servicio enlazado en runit.

2 **ENABLE SERVICE**

El estado pasa a “Installed — service not enabled” — mismo patrón que SearXNG.

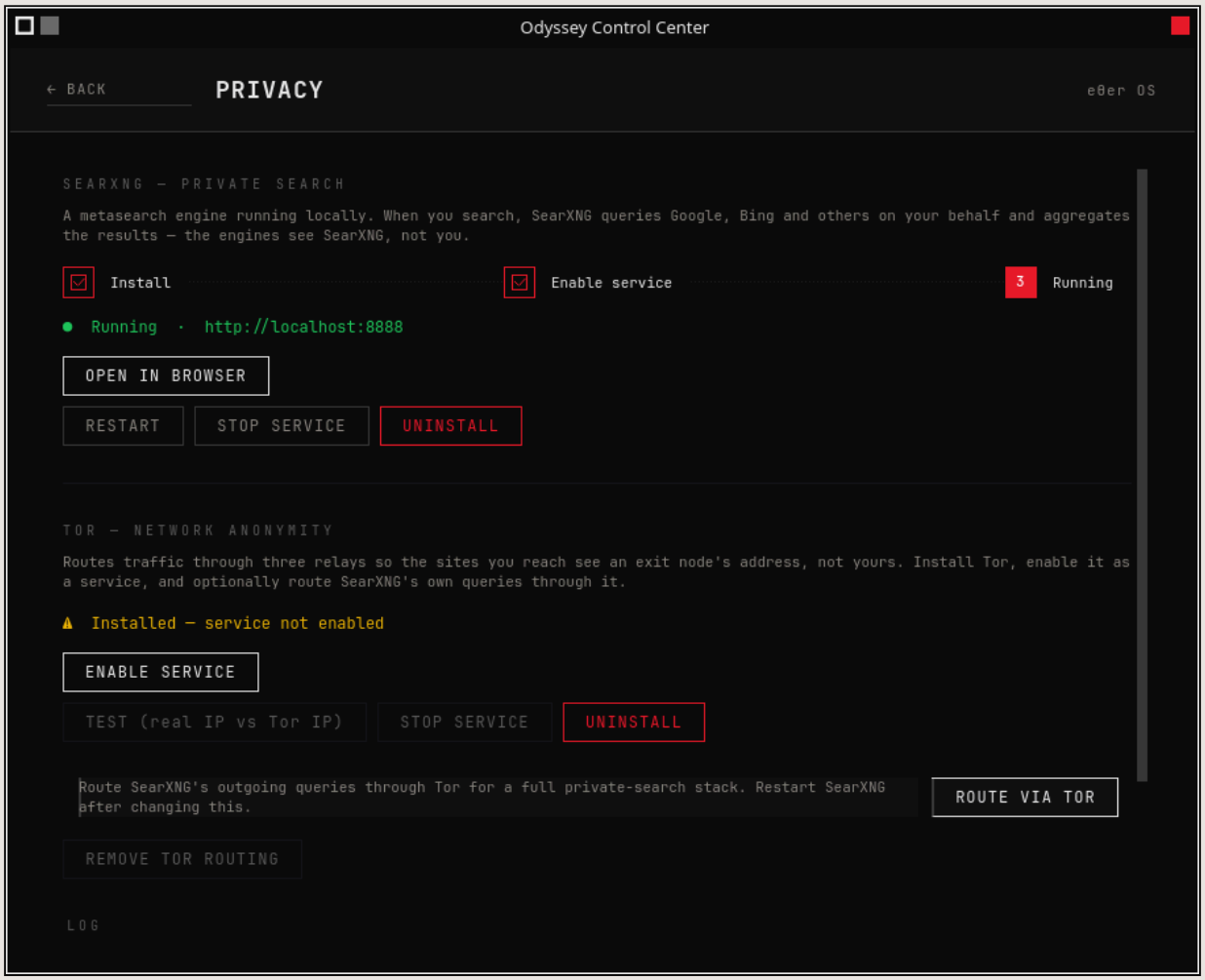

FIG · Tor en el paso 2: "Installed — service not enabled," ENABLE SERVICE activo, TEST y STOP SERVICE todavía desactivados.

3 Running, y verifícalo

Una vez habilitado, aparece un botón dedicado **TEST (real IP vs Tor IP)** — úsalo. Compara tu dirección real con la que Tor está presentando, lado a lado, así tienes una prueba real de que funcionó en lugar de un indicador de estado al que solo tienes que creer.

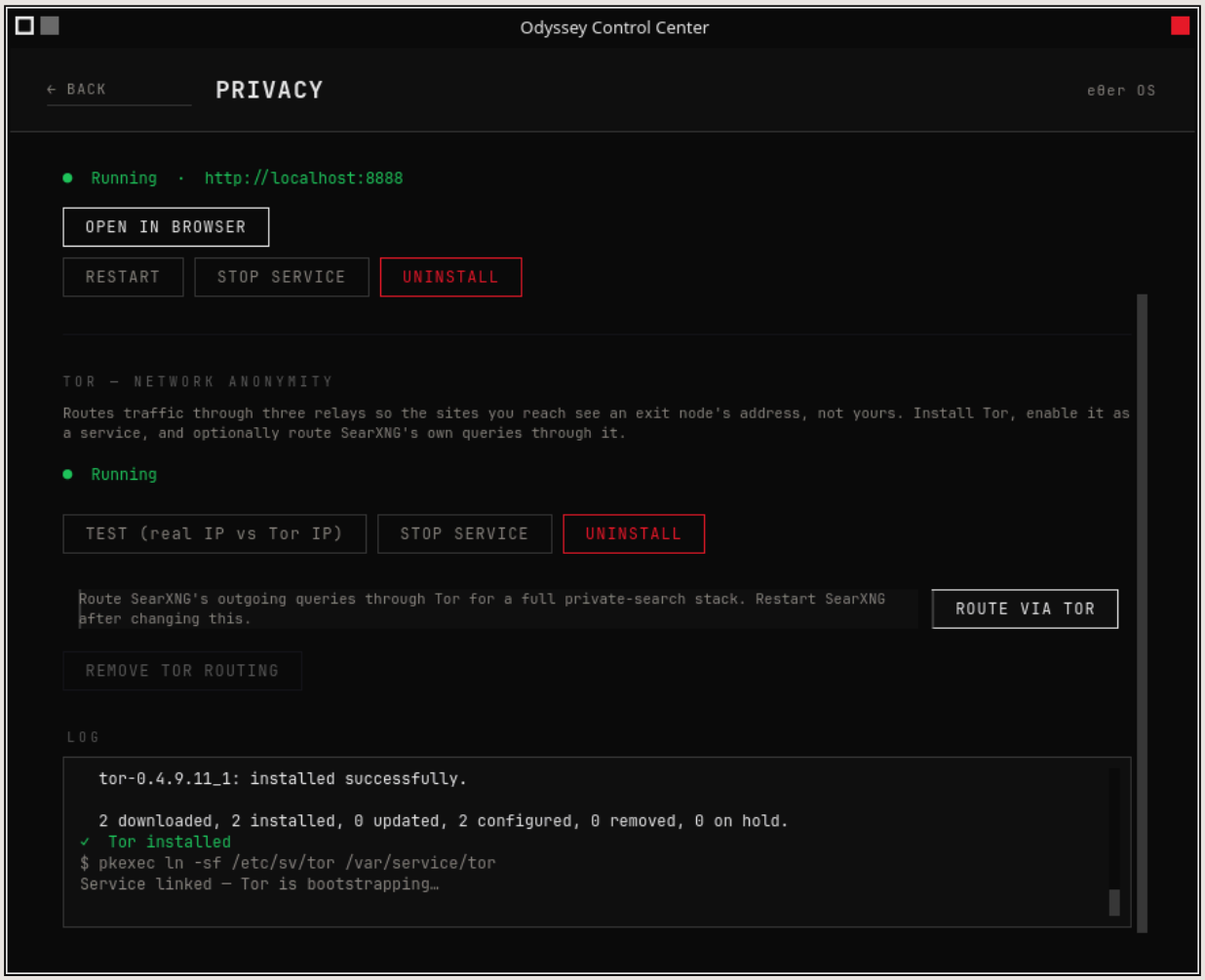

FIG · Ambas secciones en verde y en ejecución: SearXNG en localhost:8888 con OPEN IN BROWSER, Tor en ejecución con TEST / STOP SERVICE / UNINSTALL todos activos.

WHAT	Privacy → INSTALL TOR → ENABLE SERVICE → TEST (real IP vs Tor IP) para confirmar.
WHY	Enrutar la navegación sensible a través de Tor oculta tu dirección a los sitios que visitas — TEST te da una prueba, no solo una afirmación.
RESULT	Un servicio Tor en ejecución a través del cual otras aplicaciones pueden enrutarse; TEST muestra dos direcciones genuinamente diferentes si está funcionando.
RISK	El tráfico a través de Tor es notablemente más lento — es el precio del anonimato. STOP SERVICE lo pausa; UNINSTALL lo elimina.

7.3 Guía visual — combinándolos

Una vez que ambos están en ejecución, aparece un aviso en línea ofreciendo enrutar también las consultas **propias** de SearXNG a través de Tor — no solo tus solicitudes a SearXNG, sino las solicitudes de SearXNG a Google y Bing en tu nombre.

WHAT	Privacy → el aviso “Route SearXNG’s outgoing queries through Tor” → ROUTE VIA TOR → RESTART en la sección SearXNG para que tenga efecto.
WHY	Sin esto, SearXNG mismo sigue consultando a los grandes motores de búsqueda directamente — privado desde tu lado, visible desde el suyo. Enrutar el propio tráfico de SearXNG a través de Tor cierra ese hueco para una pila de búsqueda completamente privada.
RESULT	Las consultas salientes de SearXNG, no solo tus consultas hacia él, pasan por Tor.
RISK	Los resultados cargan notablemente más lento, por la misma razón que cualquier tráfico enrutado por Tor. REMOVE TOR ROUTING lo revierte, de nuevo seguido de un reinicio.

▸ ADVANCED

Past this point: config file paths, raw commands, and what the panel is doing underneath.

7.4 Advanced

QUÉ EJECUTA REALMENTE INSTALL SEARXNG

```
$ pkexec /usr/bin/odyssey-searxng-setup
# crea un usuario de sistema dedicado ‘searxng’,
# clona el código fuente en /usr/local/searxng/searxng-src,
# construye un venv de Python e instala las dependencias con pip
```

SERVICIOS Y PUERTOS, A MANO

```
# el servicio runit de SearXNG
$ sudo sv status searxng
$ sudo sv restart searxng
# el servicio runit de Tor — el paso de habilitación del panel es literalmente esto
$ sudo ln -sf /etc/sv/tor /var/service/tor
$ sudo sv status tor
# verifica que Tor realmente esté enrutando
$ curl -socks5 localhost:9050 https://check.torproject.org/api/ip
```

ENRUTAR SEARXNG A TRAVÉS DE TOR A MANO

El proxy SOCKS de Tor escucha en `127.0.0.1:9050` por defecto. Enrutar SearXNG a través de él significa apuntar su cliente HTTP saliente hacia ese proxy — el ajuste vive en el propio archivo `settings.yml` de SearXNG, bajo la configuración de solicitudes salientes, en lugar de en una variable de proxy a nivel de sistema. El botón ROUTE VIA TOR del panel edita ese archivo directamente y reinicia el servicio; hacerlo a mano significa localizar y editar el mismo bloque `outgoing:` en `/usr/local/searxng/searxng-src/searx/settings.yml`.

POR QUÉ SEARXNG NECESITA UN REINICIO DESPUÉS DE CAMBIOS DE ENRUTAMIENTO

SearXNG lee su configuración de proxy una sola vez, al arrancar — no es un ajuste recargable en caliente. Esto es cierto tanto si lo cambias a través del panel como a mano, por lo que ambos caminos terminan en el mismo paso RESTART.

ARCHIVOS QUE TOCA ESTE PANEL

RUTA	FUNCIÓN
`/usr/local/searxng/searxng-src/`	La propia instalación de SearXNG — código fuente, venv, configuración
`/etc/sv/searxng`, `/etc/sv/tor`	Definiciones de servicios runit
`/var/service/`	Donde habilitar un servicio lo enlaza en el árbol de supervisión en ejecución

CHAPTER 8

Containers

Gestión de contenedores de principio a fin: configuración del motor, una lista de contenedores en ejecución, un catálogo curado desde el cual empezar, comandos en bruto para cualquier cosa que no esté en él, y uso de disco — todo en un único panel desplazable.

8.1 Guía visual — el motor

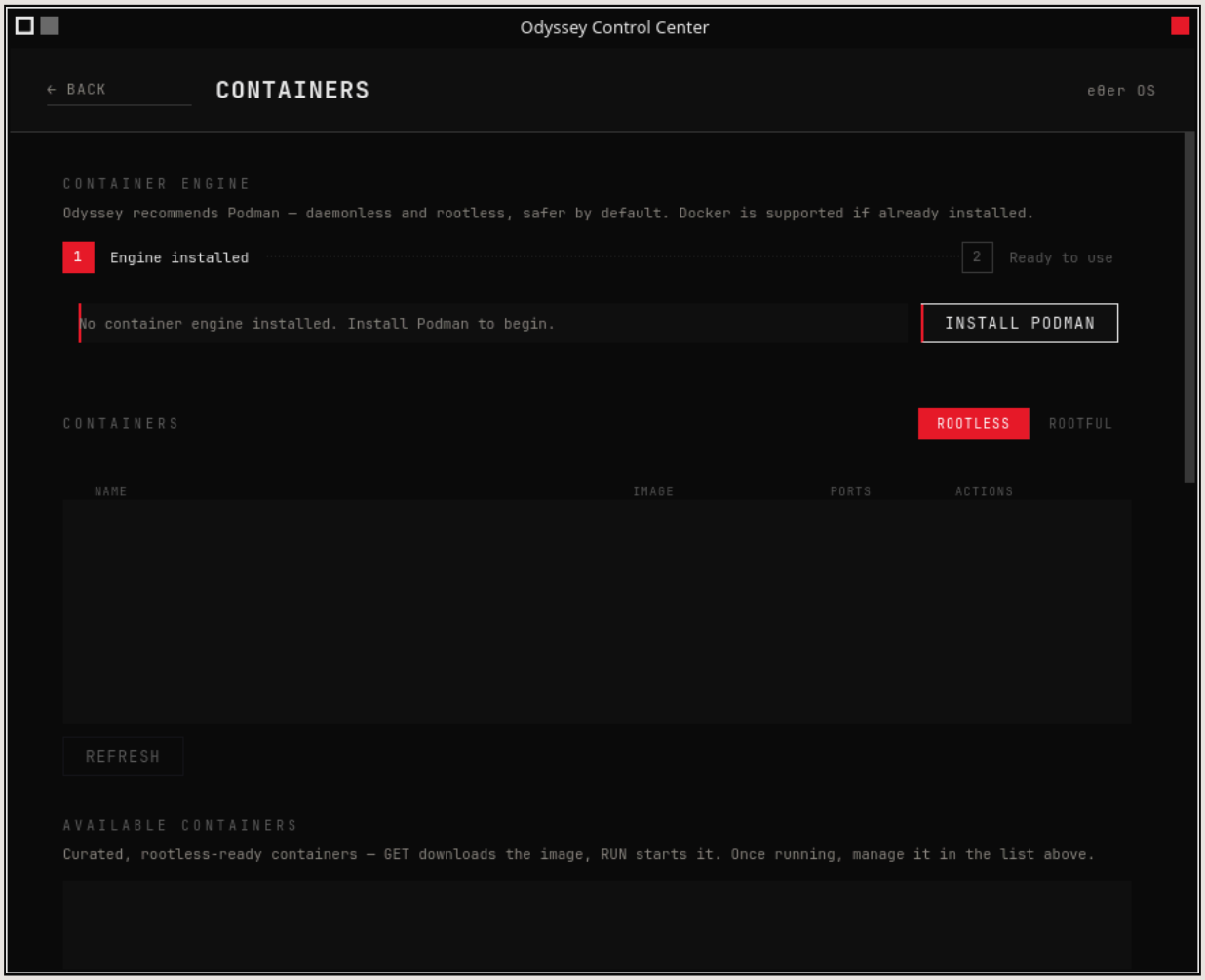

FIG · Paso 1 de 2: “No container engine installed. Install Podman to begin,” con INSTALL PODMAN como única acción activa. Las secciones Containers list y Available containers debajo son visibles pero en estado vacío.

Odyssey recomienda **Podman** — sin daemon y rootless, lo que significa que los contenedores corren sin un servicio en segundo plano permanente y, por defecto, sin privilegios de root. **Docker** también es compatible si ya es lo que usas.

1 **INSTALL PODMAN**

El registro muestra la instalación real del paquete.

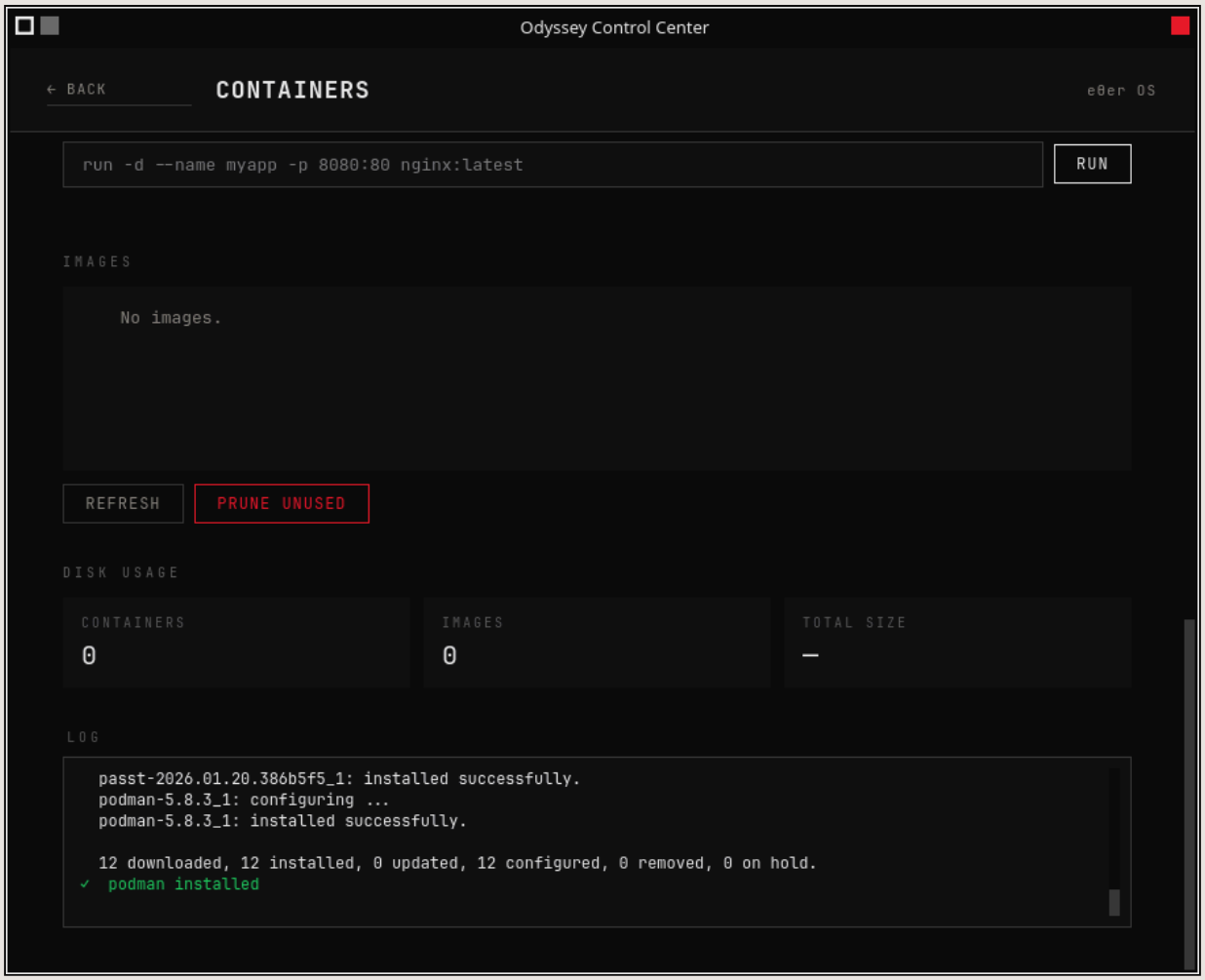

FIG · Después de instalar: “Engine installed” marcado en rojo, paso 2 “Ready to use,” la línea informativa que dice “podman version 5.8.3 · Podman installed · Docker not installed,” y un botón INSTALL DOCKER TOO para quien también quiera Docker en paralelo.

2 Ready to use

El motor está activo. **INSTALL DOCKER TOO** sigue disponible si quieres ambos motores lado a lado — coexisten sin conflicto.

8.2 Guía visual — rootless vs. rootful, y el estado vacío

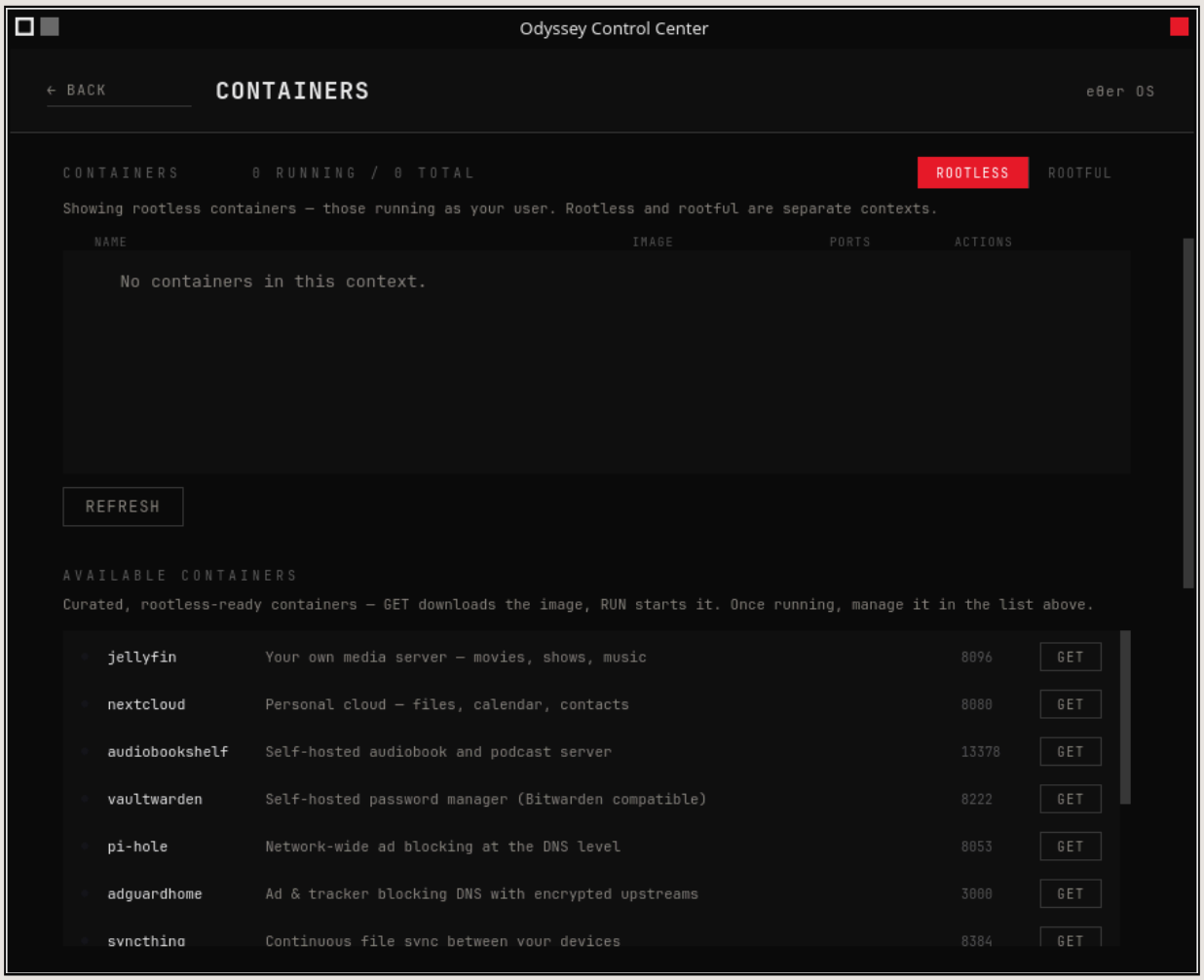

FIG · ROOTLESS seleccionado (resaltado en rojo) junto a ROOTFUL, "0 RUNNING / 0 TOTAL," y "No containers in this context." debajo de los encabezados de columna NAME / IMAGE / PORTS / ACTIONS.

El interruptor **ROOTLESS / ROOTFUL** en la parte superior cambia qué contexto estás viendo y gestionando — los dos son espacios de nombres de contenedores completamente separados, no un filtro sobre una lista combinada. Odyssey muestra ROOTLESS por defecto, y ese valor predeterminado es deliberado:

> **CUÁL DE LOS DOS DEBERÍA USAR**
>
> Rootless es la opción más segura y hacia la que el panel te orienta: un contenedor que escapa del confinamiento en modo rootless de todos modos no puede hacer nada que tu propia cuenta de usuario no pudiera hacer ya. Rootful existe para los casos específicos que lo necesitan — ciertos puertos de red de número bajo, algún acceso a hardware — y debería ser la excepción, no la costumbre.

8.3 Guía visual — el catálogo curado, de principio a fin

Desplázate hasta **Available containers** — una lista ya preparada para que el panel nunca sea una caja vacía esperando a que ya conozcas los nombres de las imágenes de memoria. Cada fila muestra el contenedor, una descripción en una línea, y el puerto que usa.

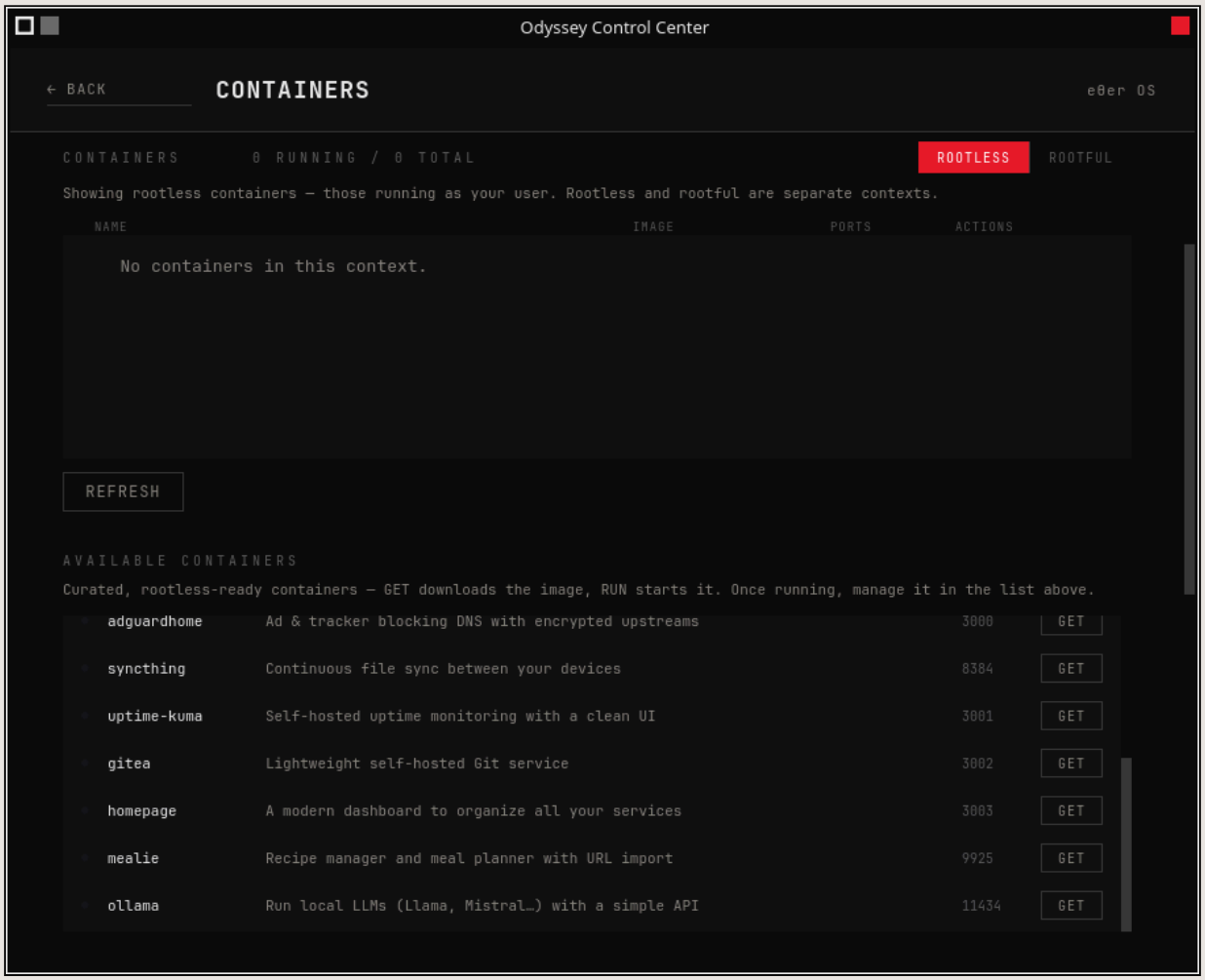

FIG · La lista del catálogo: adguardhome (puerto 3000), syncthing (8384), uptime-kuma (3001), gitea (3002), homepage (3003) — con un botón GET en cada fila.

Ese número de puerto es en el que alcanzarás el servicio una vez en ejecución — `homepage` con `3003`, por ejemplo, significa `http://localhost:3003` una vez activo.

1 **GET**

Haz clic en GET sobre `homepage`. Esto solo descarga la imagen del contenedor — nada está en ejecución todavía.

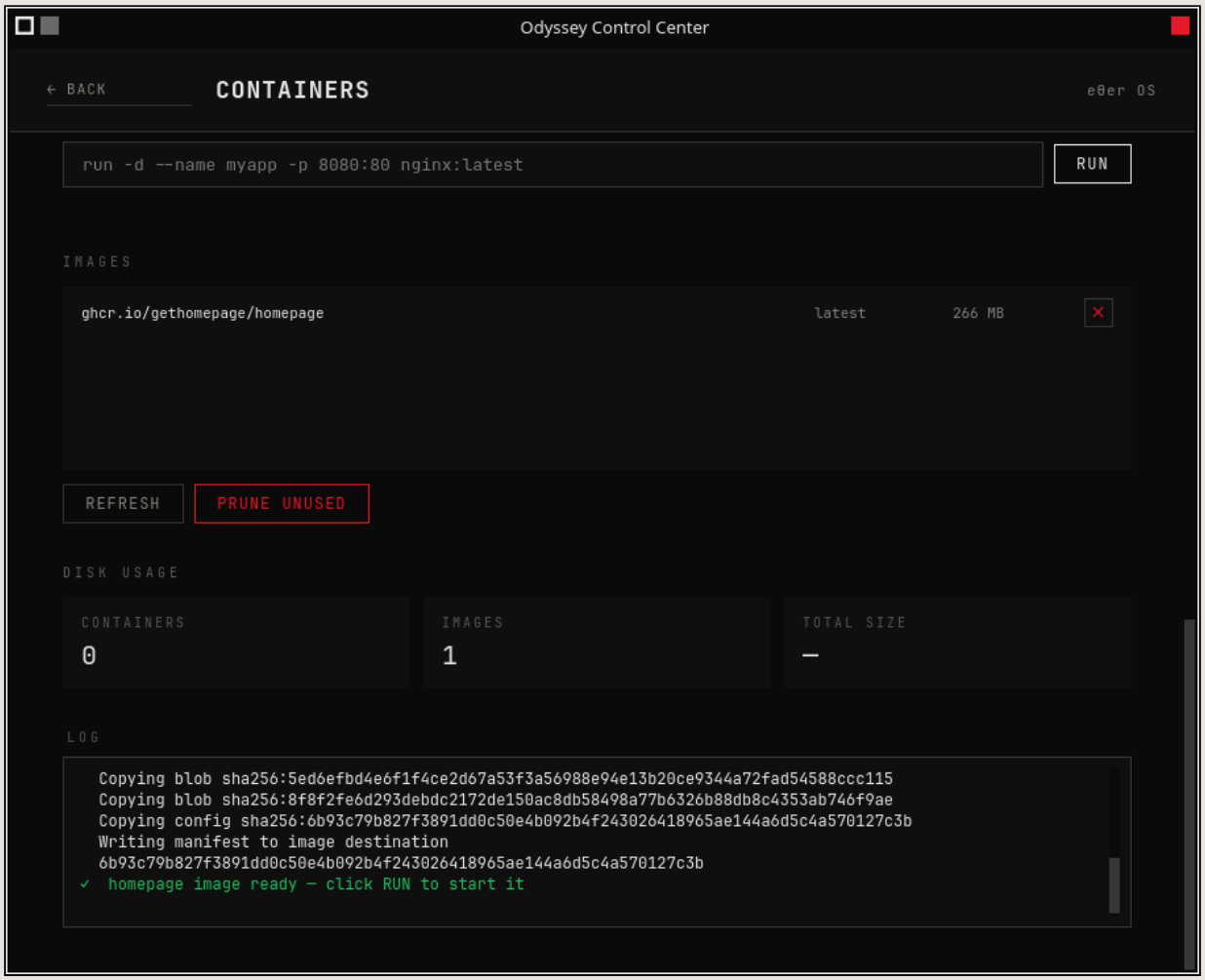

FIG · El registro transmitiendo una descarga real de imagen: “Copying blob sha256:5ed6…”, “Writing manifest to image destination,” terminando en verde — “homepage image ready — click RUN to start it.”

2 RUN

La fila del botón GET ahora muestra RUN en su lugar — haz clic para arrancar de verdad el contenedor.

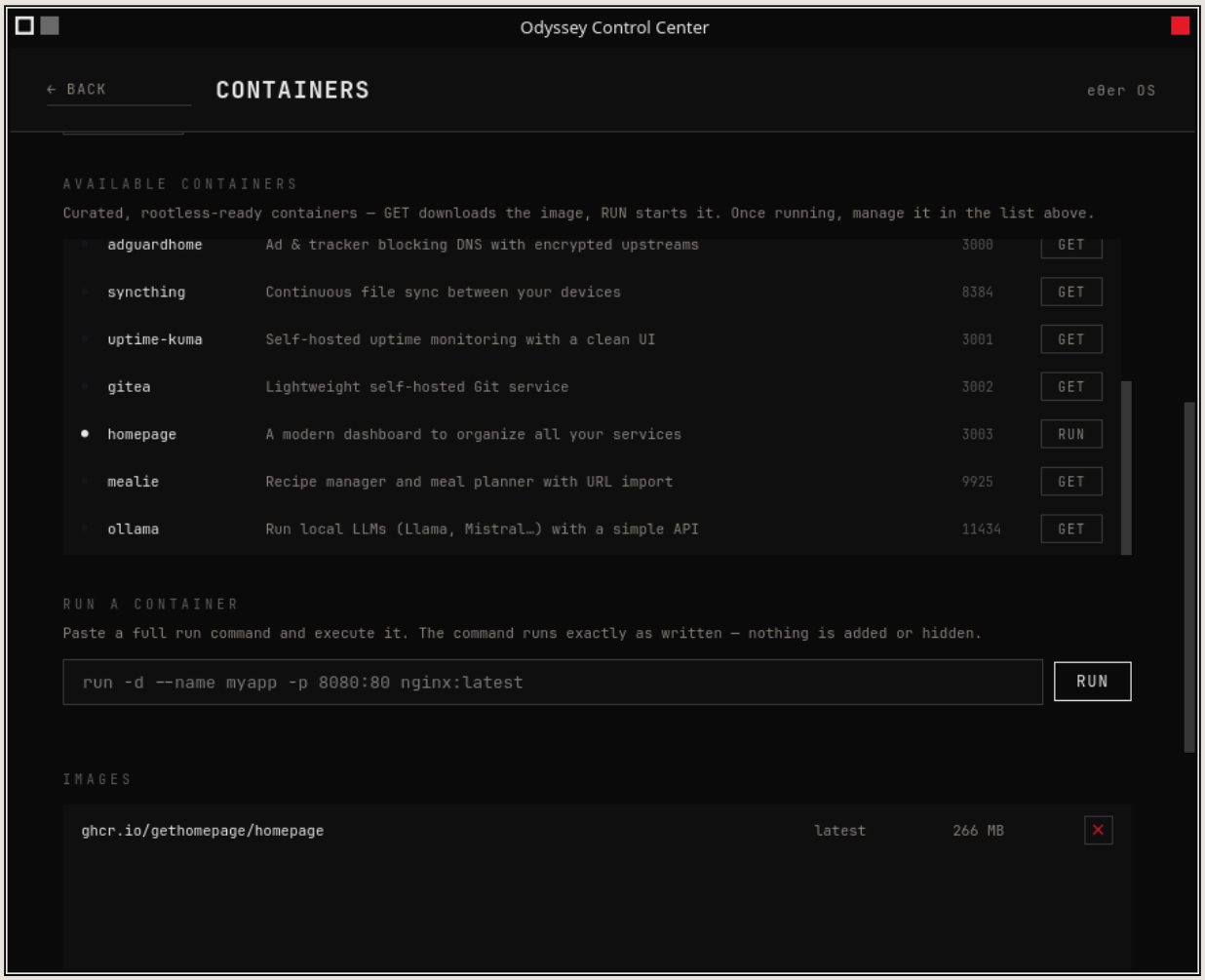

FIG · homepage ahora listado en la sección CONTAINERS arriba, punto verde, ruta de imagen ghcr.io/gethomepage/homepage, puertos "0.0.0.0:3003→…", con su propia fila de iconos de acción; todavía visible debajo en Available Containers con un nuevo botón RUN para arrancar otra instancia.

Una vez en ejecución, `homepage` aparece en dos lugares a la vez: como entrada activa en la lista **Containers** arriba (con su puerto e iconos de acción), y todavía en el catálogo debajo por si quieres arrancar una segunda instancia más tarde.

WHAT	Containers → Available containers → GET en algo del catálogo → RUN una vez descargado.
WHY	La división en dos pasos (descargar, luego arrancar) significa que puedes obtener una imagen por adelantado sin comprometerte a ejecutarla de inmediato — útil en una conexión lenta o cuando solo estás explorando qué hay disponible.
RESULT	Un contenedor en ejecución, visible en la lista principal con su mapeo de puerto, gestionable desde ahí en adelante.
RISK	Bajo — los servicios autoalojados del catálogo curado están elegidos para ser rootless-friendly y de bajo riesgo por defecto. La exposición es la misma que ejecutar cualquier servicio web: cualquier puerto en el que escuche es alcanzable según tus ajustes de Firewall.

8.4 Guía visual — la lista de contenedores y sus acciones

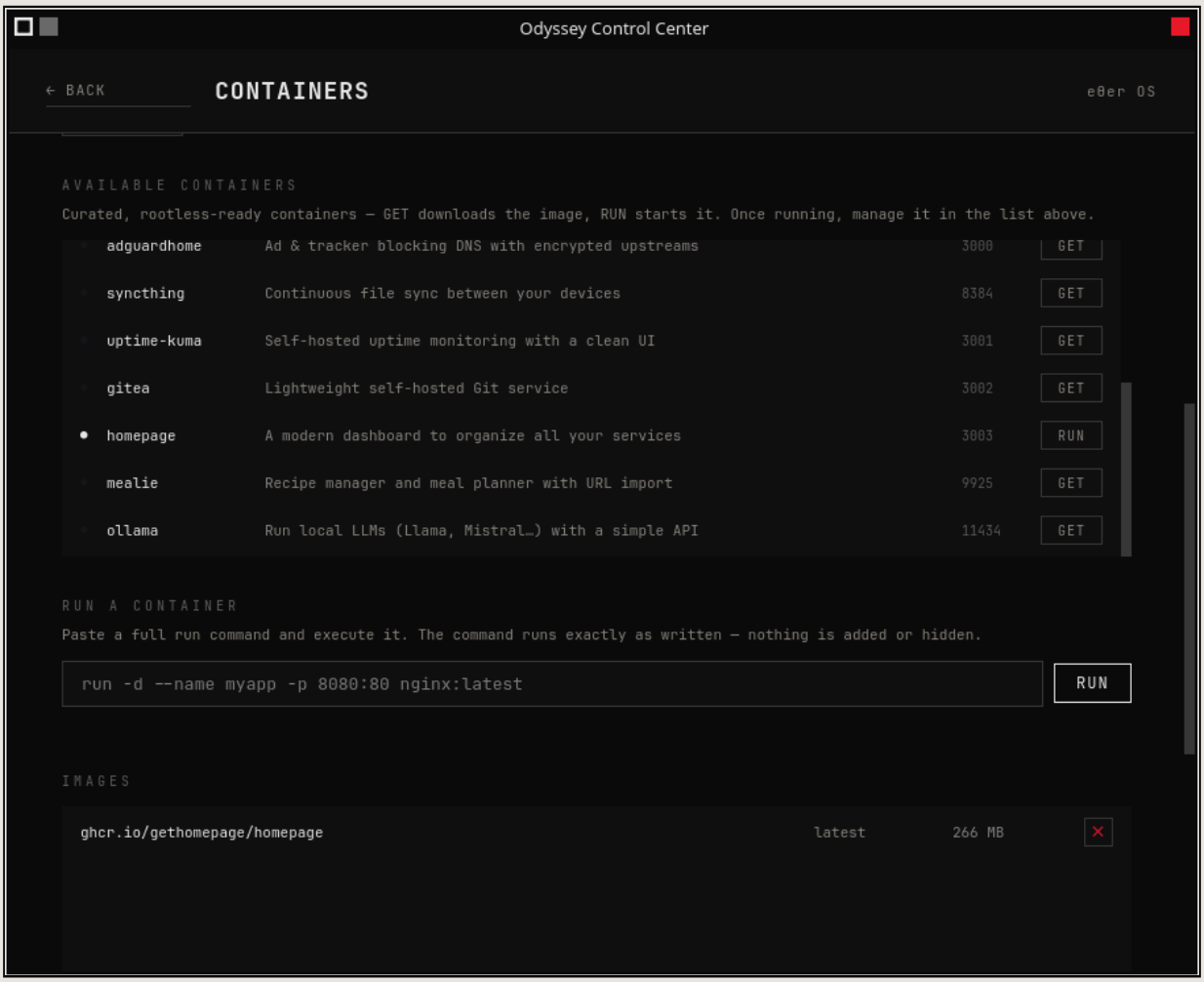

FIG · La fila de homepage en la lista Containers, con sus cinco iconos de acción visibles a la derecha — los que se explican debajo.

Cada fila de un contenedor en ejecución termina en un conjunto de iconos — pasa el ratón sobre cualquiera para ver su etiqueta, pero esto es lo que hace cada uno:

ICONO	ACCIÓN
abrir (flecha)	Abre el servicio directamente en tu navegador — el camino rápido una vez que algo está en ejecución
detener / iniciar (cuadrado / triángulo)	Detiene el contenedor sin eliminarlo, o lo vuelve a iniciar
reiniciar (flecha circular)	Lo detiene e inicia en una sola acción — la solución para "está en ejecución pero no responde"
registros (icono de líneas)	Abre la propia salida de registro del contenedor — el primer lugar donde mirar si algo no funciona
eliminar (×)	Elimina el contenedor por completo — la imagen en sí permanece descargada, listada por separado bajo Images

8.5 Guía visual — ejecutar un contenedor, y abrirlo

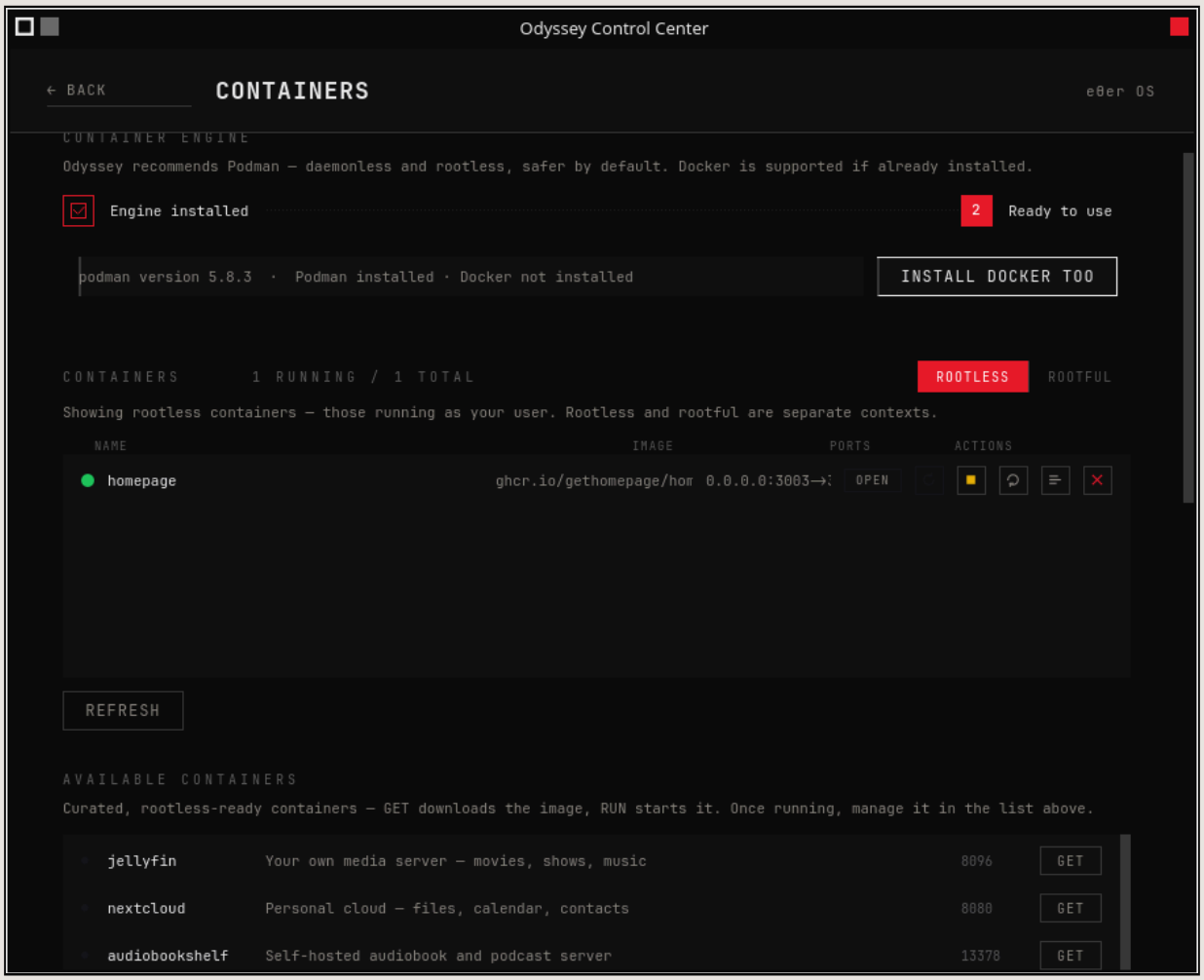

FIG · El campo "Run a container" con un comando de ejemplo como marcador de posición, el botón RUN, y la sección Images debajo listando ghcr.io/gethomepage/homepage · latest · 266 MB con un botón de eliminación.

Para cualquier cosa que no esté en el catálogo, **Run a container** toma un comando `podman run` (o `docker run`) completo y lo ejecuta exactamente como está escrito — nada se añade, nada se oculta.

WHAT	Containers → "Run a container" → pega un comando run completo (p. ej. `run -d --name myapp -p 8080:80 nginx:latest`) → RUN.
WHY	El catálogo curado cubre los casos comunes; esta es la vía de escape para una imagen específica, banderas específicas, puertos específicos que el catálogo no ofrece.
RESULT	El contenedor arranca exactamente como especifica el comando, y se une a la lista principal de Containers una vez en ejecución.
RISK	Lo que sea que haga el comando en sí — esto es un paso fiel, no una sandbox. Trátalo con el mismo cuidado con el que escribirías el mismo comando en una terminal.

Debajo, **Images** lista cada imagen descargada independientemente de si hay un contenedor en ejecución desde ella actualmente, con **REFRESH** y **PRUNE UNUSED** para recuperar espacio de las imágenes que ya nadie referencia.

ABRIR UN CONTENEDOR EN EJECUCIÓN

Hacer clic en OPEN en la fila de un contenedor en ejecución — la misma acción de abrir desde la fila de iconos de arriba — lanza tu navegador predeterminado directamente a su dirección y puerto, p. ej. `http://localhost:3003` para `homepage`. No hace falta que recuerdes tú mismo el número de puerto una vez que está en ejecución.

8.6 Uso de disco

Al final del panel, tres totales en tiempo real — contenedores, imágenes, tamaño total — así siempre sabes qué te cuestan los contenedores en disco sin abrir una terminal.

```
▸ ADVANCED

Past this point: config file paths, raw commands, and what the panel is doing underneath.
```

8.7 Advanced

EQUIVALENTES DE LÍNEA DE COMANDOS

```
# lo que el panel llama "GET"
$ podman pull ghcr.io/gethomepage/homepage
# lo que el panel llama "RUN" en una entrada del catálogo
$ podman run -d -name homepage -p 3003:3000 ghcr.io/gethomepage/homepage
# lista los contenedores en ejecución — contexto rootless
$ podman ps
# lo mismo, rootful
$ sudo podman ps
# registros, detener, reiniciar, eliminar — la fila de iconos, a mano
$ podman logs -f homepage
$ podman stop homepage
$ podman restart homepage
$ podman rm homepage
# imágenes y uso de disco
$ podman images
$ podman system df
$ podman image prune
```

ROOTLESS VS. ROOTFUL, POR DEBAJO

Podman rootless ejecuta los contenedores como tu propio usuario, dentro de un espacio de nombres de usuario — el "root" del contenedor se mapea a un rango de UID sin privilegios en el host, así que incluso un proceso de contenedor completamente comprometido sigue estando ligado a los permisos de tu propia cuenta. Podman rootful (`sudo podman ...`) funciona de la manera tradicional, con el root del contenedor mapeado al root real del host. El interruptor del panel en realidad solo está cambiando entre ejecutar `podman` simple o `sudo podman` — directorios de estado genuinamente separados, listas de contenedores separadas, nada compartido entre los dos contextos por defecto.

```
# dónde guarda cada contexto su propio estado
$ podman info -format '{{.Store.GraphRoot}}'
# rootless:  /.local/share/containers/storage
# rootful: /var/lib/containers/storage
```

POR QUÉ ALGUNAS ENTRADAS DEL CATÁLOGO NECESITAN PUERTOS ESPECÍFICOS ABIERTOS

El mapeo de puerto de un contenedor (`3003:3000` en el ejemplo de homepage) solo hace que el servicio sea alcanzable **desde esta máquina** por defecto — Podman lo enlaza a `localhost` a menos que se le indique lo contrario. Alcanzarlo desde otro dispositivo en tu red requiere además que ese puerto esté abierto en el panel Firewall (Capítulo 4) — los dos sistemas son capas independientes, y un contenedor en ejecución con un puerto mapeado no es automáticamente alcanzable desde internet o la LAN solo porque Podman esté escuchando ahí.

CHAPTER 9

Gaming Mode

Un interruptor principal en la parte superior, una fila de estado en tiempo real, y cuatro pestañas: **PACKAGES**, **GAMEMODE**, **MANGOHUD**, **PROTON**.

9.1 Guía visual — el interruptor principal

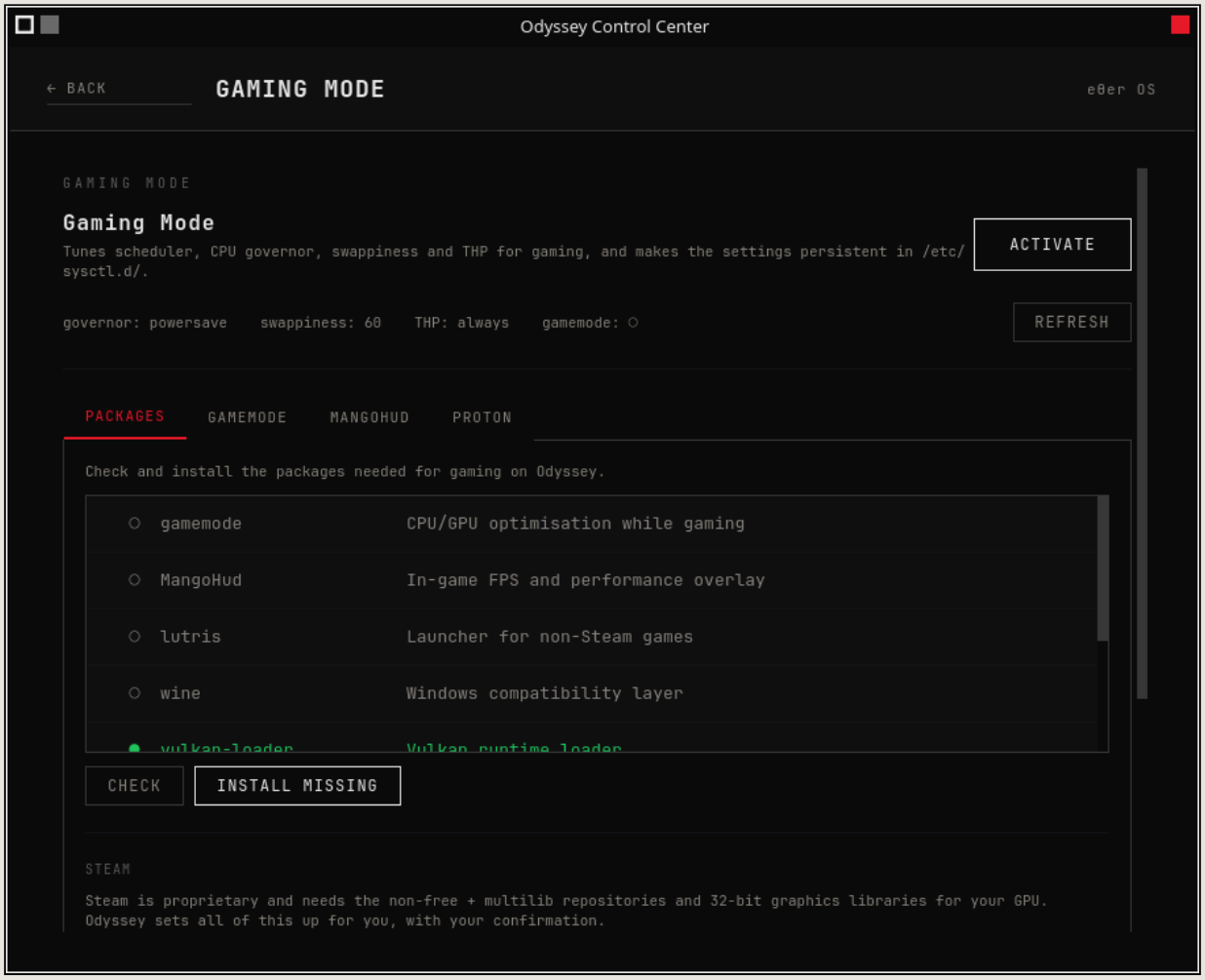

FIG · La sección superior del panel: "Gaming Mode — Tunes scheduler, CPU governor, swappiness and THP for gaming, and makes the settings persistent in /etc/sysctl.d/," un botón ACTIVATE, un botón REFRESH, y la fila de estado en vivo: governor powersave, swappiness 60, THP always, gamemode off.

La fila de estado muestra los mismos valores en vivo que enseñan los paneles Kernel y Memory — governor, swappiness, estado THP, y si el servicio `gamemode` está en ejecución — así puedes confirmar de un vistazo si la activación realmente surtió efecto, sin visitar por separado ninguno de los dos paneles.

WHAT	Gaming Mode → ACTIVATE.
WHY	Visitar manualmente Kernel y Memory para fijar a mano un governor, una swappiness y un valor THP adecuados para gaming es exactamente el tipo de tarea de varios pasos que una modalidad dedicada debería eliminar.
RESULT	Scheduler, governor, swappiness y THP pasan todos a valores adecuados para gaming, aplicados de inmediato y hechos persistentes — la fila de estado se actualiza para reflejar el cambio.
RISK	Bajo, y reversible — los mismos valores que fijarías de otro modo a mano en Kernel/Memory, solo agrupados en un clic.

9.2 Guía visual — Packages y Steam

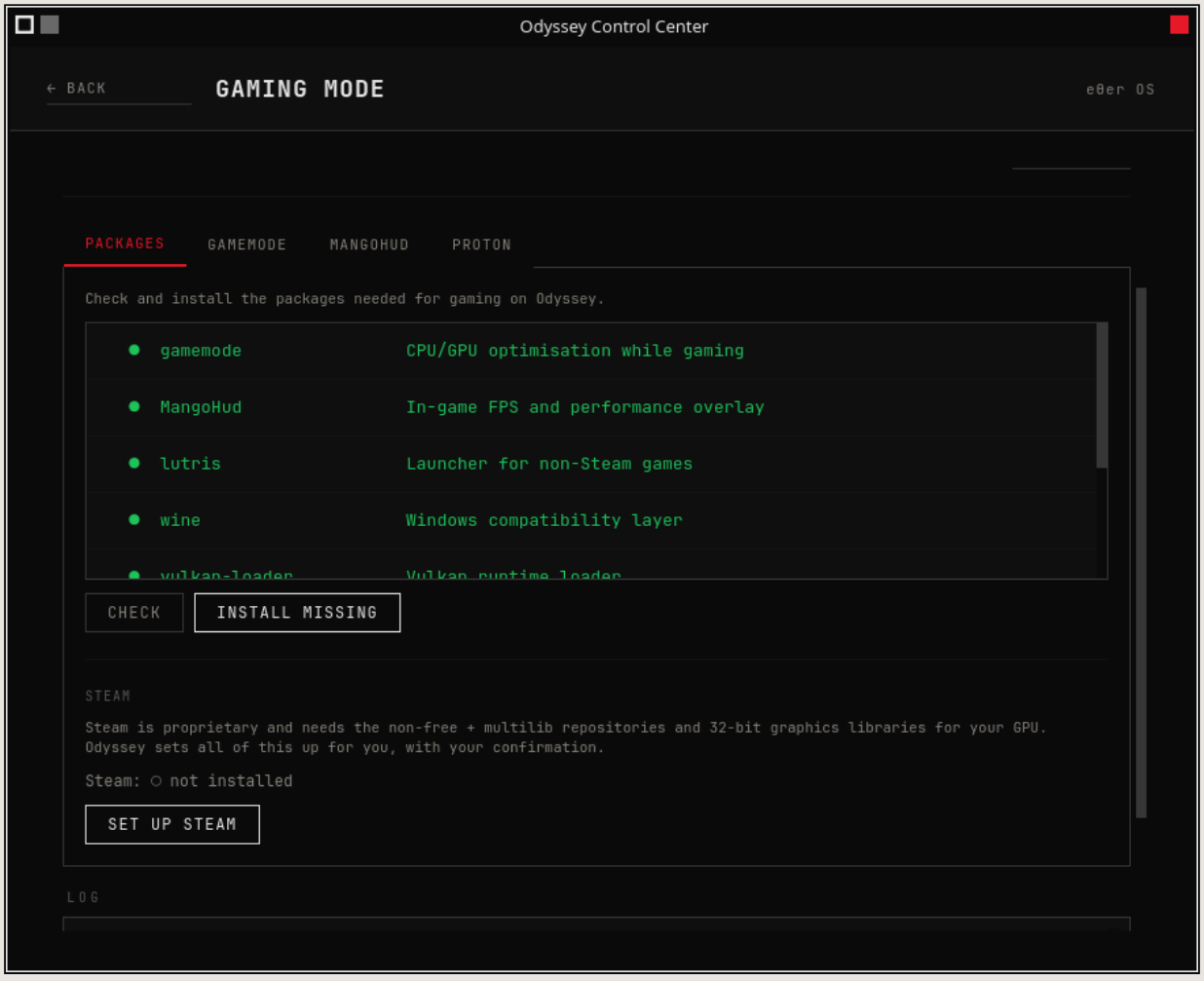

FIG · La pestaña PACKAGES: gamemode, MangoHud, lutris, wine, vulkan-loader todos con puntos verdes rellenos (instalados); CHECK e INSTALL MISSING arriba; la sección Steam debajo con "Steam: ● installed" y un botón REINSTALL STEAM.

Una lista de verificación de los paquetes generalmente necesarios para gaming — CHECK ve qué falta, INSTALL MISSING rellena los huecos en una sola acción. Una vez que todo muestra un punto verde relleno, como en la captura, estás completamente configurado.

Steam tiene su propio flujo dedicado debajo de la lista de verificación, porque necesita más que una simple instalación de paquete.

1 **SET UP STEAM**

Haz clic — no instala nada todavía, primero abre un diálogo de confirmación informativo.

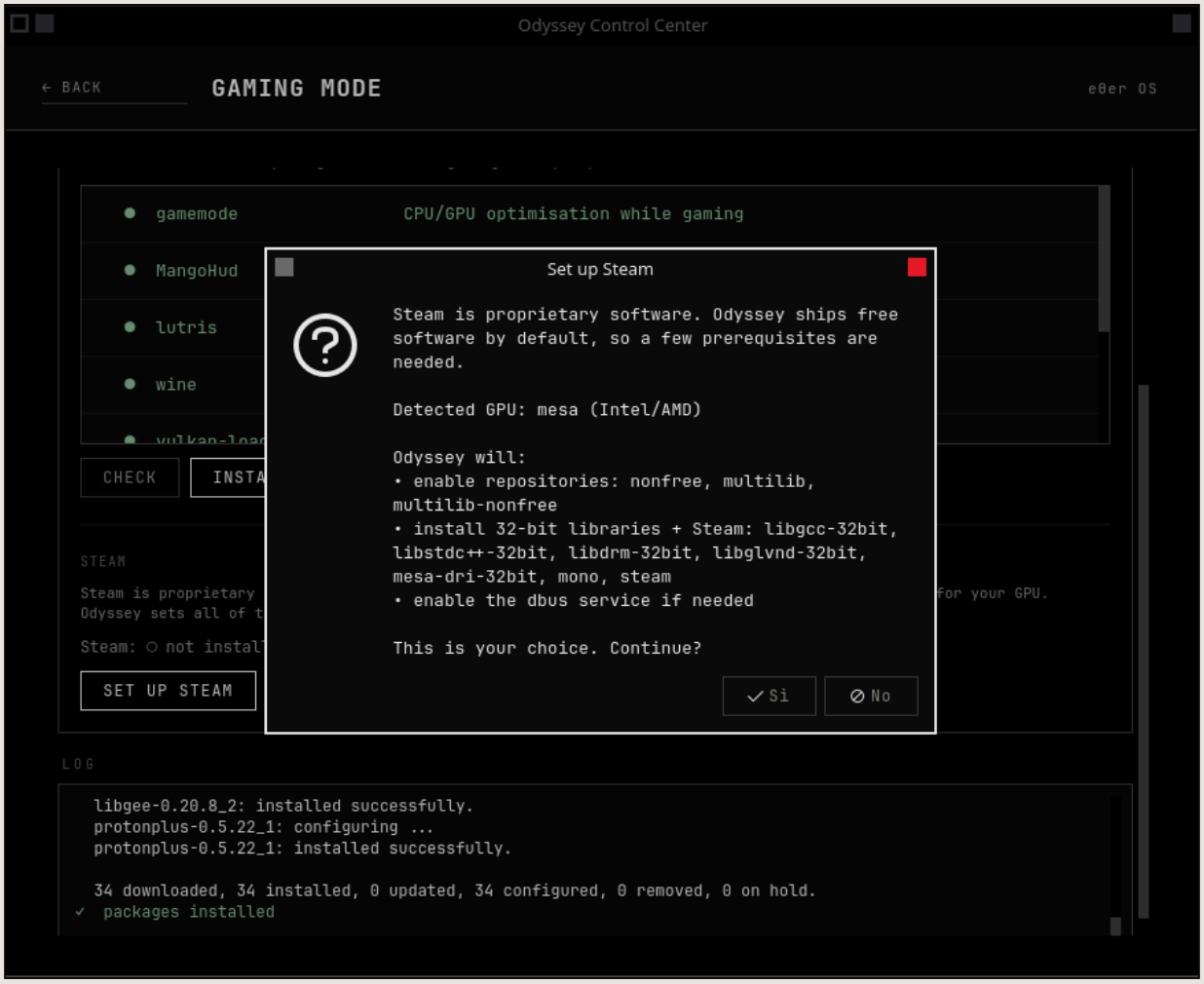

FIG · El diálogo de confirmación: “Steam is proprietary software. Odyssey ships free software by default, so a few prerequisites are needed. Detected GPU: mesa (Intel/AMD).” Una lista con viñetas de qué se habilitará e instalará exactamente, terminando en “This is your choice. Continue?” con los botones Sí / No.

El diálogo te dice exactamente qué está a punto de pasar antes de que lo aceptes: qué repositorios se habilitan (`nonfree`, `multilib`, `multilib-nonfree`), qué bibliotecas de 32 bits se instalan según tu GPU **detectada**, y que el servicio dbus se habilitará si aún no lo está. Nada se oculta detrás de la confirmación.

2 **Sí**

Lo confirma. El registro transmite la instalación real — descargas de paquetes, configuración, finalización.

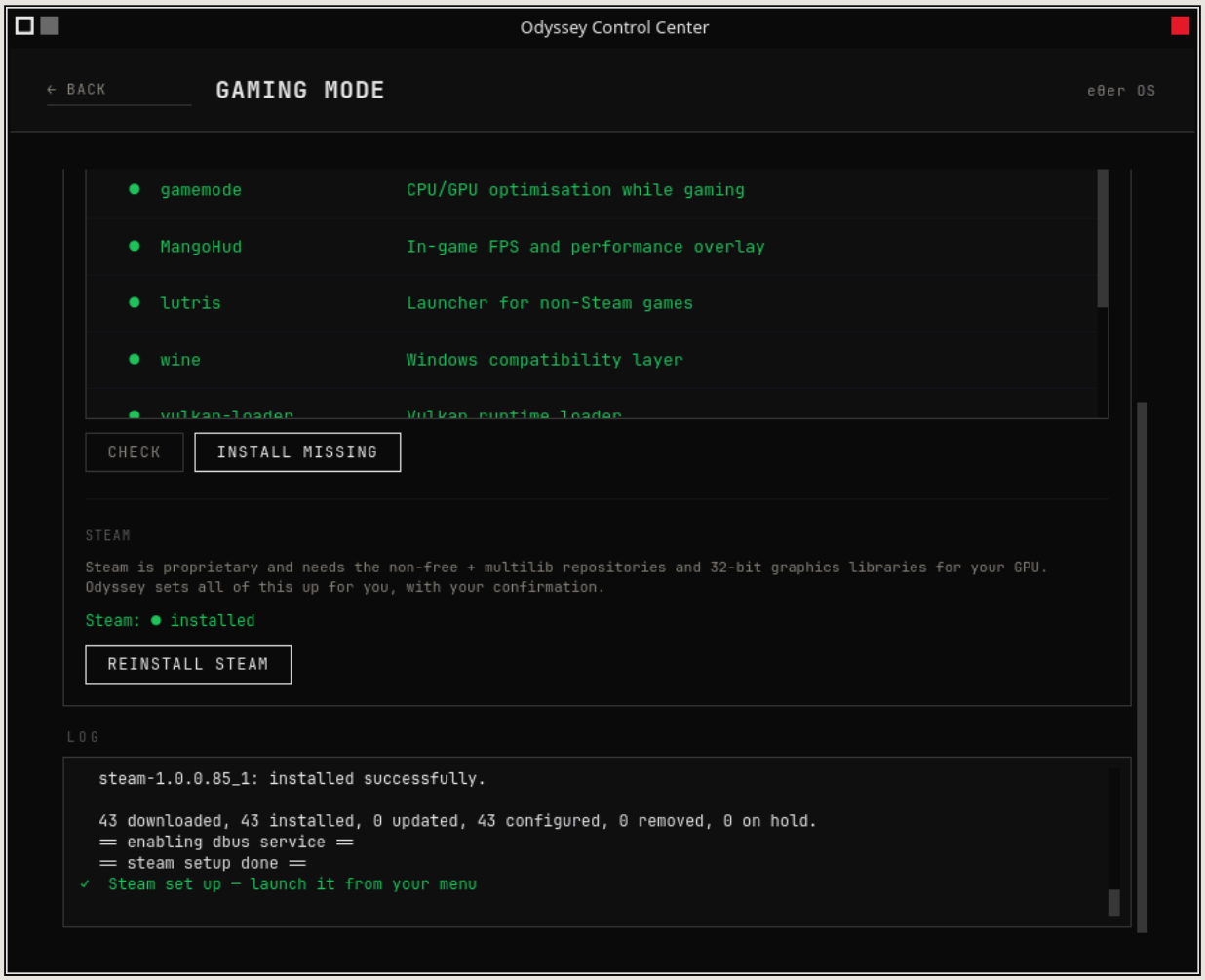

FIG · La pestaña PACKAGES después de la configuración: todos los paquetes en verde, Steam mostrando "● installed" con REINSTALL STEAM, y el registro terminando en "Steam set up — launch it from your menu."

WHAT	Gaming Mode → pestaña PACKAGES → SET UP STEAM → revisa el diálogo → Sí.
WHY	Steam es propietario y necesita los repositorios non-free/multilib más bibliotecas de 32 bits específicas para la GPU que Void no habilita por defecto — hacer esto bien a mano es un obstáculo común para quien empieza con un sistema basado en Void.
RESULT	Los repositorios necesarios, Steam mismo, y las bibliotecas gráficas de 32 bits correspondientes a tu GPU específica, todos instalados — arranca Steam desde tu menú de aplicaciones después.
RISK	Bajo — esto solo habilita repositorios y añade paquetes; nada existente se elimina ni reconfigura. El diálogo muestra la lista completa de qué pasará antes de que confirmes.

9.3 GameMode y MangoHud

GameMode (su propia pestaña) es el daemon de rendimiento por juego de Feral Interactive — un juego lanzado a través de él solicita un impulso temporal de rendimiento (governor de CPU, prioridad de I/O) solo para su propio proceso, devuelto automáticamente cuando el juego se cierra. **MangoHud** (su propia pestaña) es la superposición en pantalla que muestra FPS, frame time, carga de CPU/GPU y temperaturas durante el juego. Ambas pestañas ofrecen controles de instalación/verificación y, para MangoHud, configuración rápida de la apariencia de la superposición.

9.4 Proton

Gestión de la capa de compatibilidad para ejecutar juegos de Windows a través de Steam — comprobando qué está instalado y disponible, e instalando versiones adicionales de Proton, incluidas builds de la comunidad como Proton-GE, cuando un juego específico necesita una que Valve no distribuye por defecto.

```
▸ ADVANCED

Past this point: config file paths, raw commands, and what the panel is doing underneath.
```

9.5 Advanced

QUÉ ESCRIBE ACTIVATE

Los mismos archivos subyacentes que las acciones MAKE PERSISTENT de los paneles Kernel y Memory — Gaming Mode es un paquete, no un mecanismo separado:

```
# valores sysctl hechos persistentes
$ cat /etc/sysctl.d/99-odyssey.conf
# CPU governor — no es sysctl, se basa en sysfs, no se hace persistente
# por este mecanismo; vuelve a aplicarlo después de arrancar si dependes de él
$ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

QUÉ EJECUTA REALMENTE SET UP STEAM

El propio diálogo de confirmación es casi una divulgación técnica completa, pero en concreto, en un sistema Mesa (Intel/AMD):

```
$ sudo xbps-install -y void-repo-nonfree void-repo-multilib void-repo-multilib-nonfree
$ sudo xbps-install -Sy steam libgcc-32bit libstdc++-32bit libdrm-32bit libglvnd-32bit
mesa-dri-32bit mono
# los sistemas NVIDIA descargan además nvidia-libs-32bit en lugar de mesa-dri-32bit —
# el panel detecta el fabricante de tu GPU y ajusta la lista de paquetes en consecuencia
```

GAMEMODE Y MANGOHUD A MANO

```
# lanza cualquier juego con GameMode activo
$ gamemoderun ./game_binary
# o, para un juego de Steam, opciones de lanzamiento:
$ gamemoderun %command%
# superposición MangoHud, opciones de lanzamiento:
$ mangohud %command%
# el propio archivo de configuración de MangoHud, para ajustar qué muestra la superposición
$ $EDITOR  /.config/MangoHud/MangoHud.conf
```

INSTALAR UNA VERSIÓN ESPECÍFICA DE PROTON-GE

La pestaña PROTON envuelve `protonplus` o un gestor de versiones equivalente — el equivalente manual es descargar una release desde `GloriousEggroll/proton-ge-custom` en GitHub y extraerla en el directorio de herramientas de compatibilidad de Steam:

```
$ mkdir -p  /.steam/root/compatibilitytools.d
$ tar -xf GE-Proton-VERSION.tar.gz -C  /.steam/root/compatibilitytools.d
# luego selecciónala por juego en Steam: Properties → Compatibility
```

CHAPTER 10

Telemetry

A pesar del nombre, este panel no es un ajuste que configures — es un monitor en tiempo real. Odyssey no recopila ni envía nada sobre ti, así que no hay nada que desactivar aquí. Lo que este panel hace en cambio es mostrarte, en tiempo real, cada conexión saliente que tu **propio** sistema está haciendo — así "con qué está hablando mi máquina ahora mismo" es algo a lo que puedes responder mirando, no confiando en una afirmación. Se llama a sí mismo **Reverse Telemetry** precisamente por eso: da la vuelta a la observación, apuntándola hacia el propio sistema, en tu nombre.

10.1 Guía visual

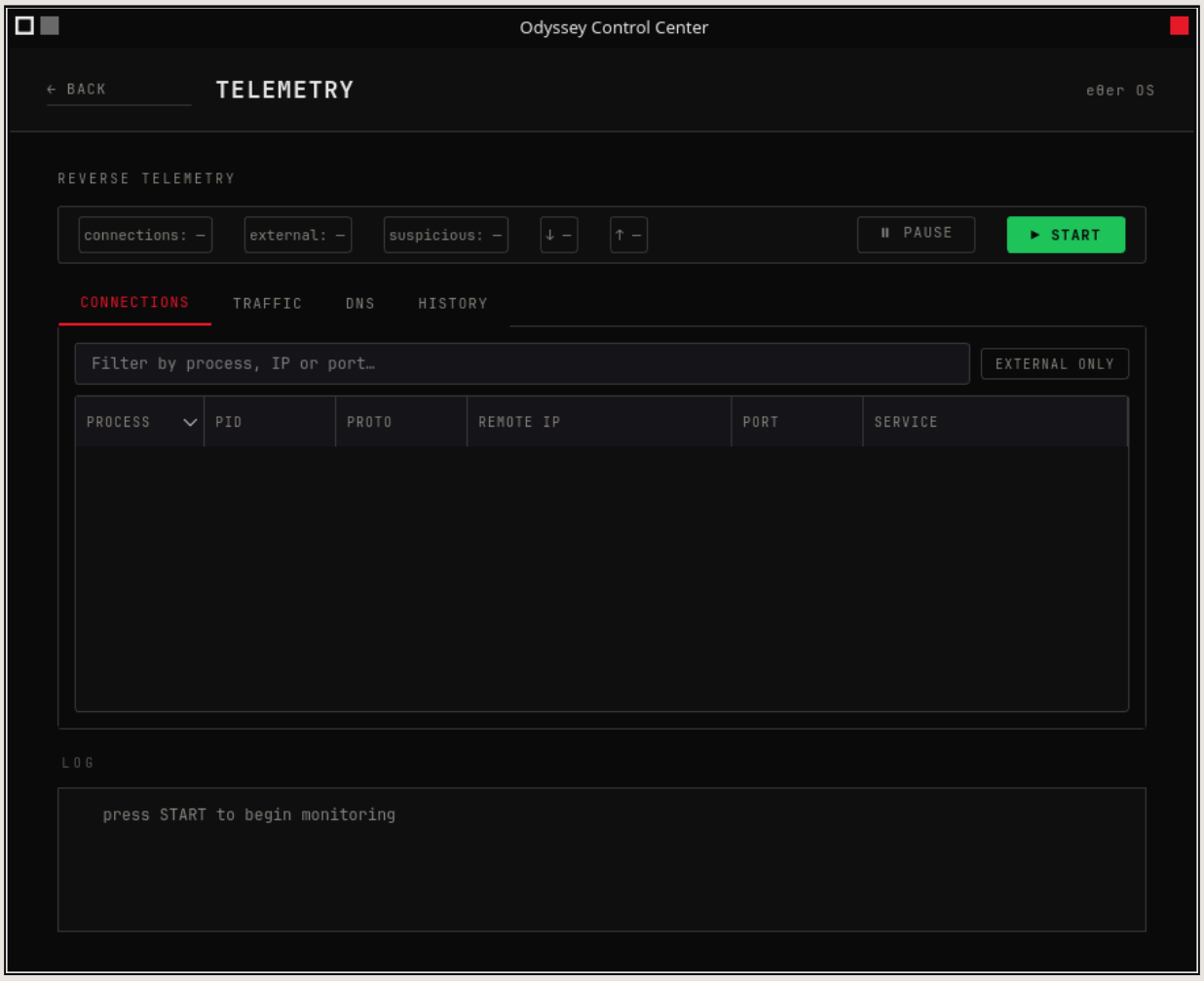

FIG · Antes de arrancar: connections/external/suspicious mostrando todos "–", un botón verde START, y el registro que dice "press START to begin monitoring." Las cuatro pestañas — CONNECTIONS, TRAFFIC, DNS, HISTORY — permanecen vacías sobre una tabla sin poblar.

1 START

Comienza el monitoreo en tiempo real — la barra de estado y la pestaña activa se pueblan de inmediato.

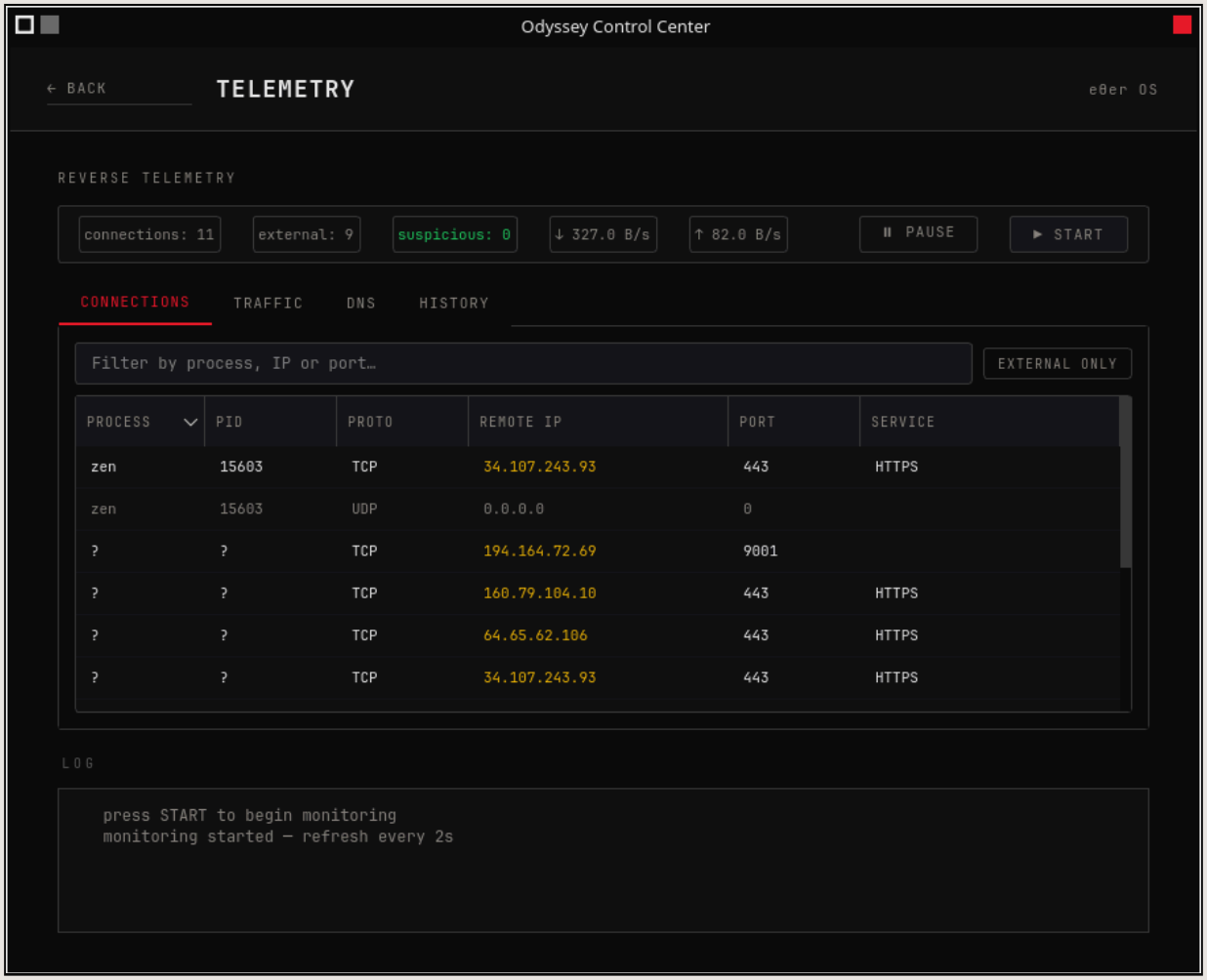

FIG · Después de arrancar: "connections: 11 · external: 9 · suspicious: 0 · ↓327.0 B/s · ↑82.0 B/s," PAUSE ahora disponible, y la tabla CONNECTIONS llena de filas reales: proceso "zen" en TCP hacia 34.107.243.93:443 (HTTPS), varias filas marcadas "?" para proceso/PID con conexiones hacia otras direcciones.

La barra de estado en la parte superior ofrece totales en tiempo real — conexiones, externas (vs. locales), sospechosas, velocidades actuales de descarga/subida — y **PAUSE** congela la vista sin perder lo capturado, **START** la reanuda.

LAS CUATRO PESTAÑAS

PESTAÑA	QUÉ MUESTRA
CONNECTIONS	Una fila por conexión activa: proceso, PID, protocolo, dirección remota, puerto, y el servicio conocido para ese puerto (443 → HTTPS, 22 → SSH). Ordenable por cualquier columna.
TRAFFIC	Throughput por interfaz — velocidad de descarga/subida en este momento, más totales acumulados desde que empezó el monitoreo.
DNS	Un flujo en tiempo real de las resoluciones de nombres de dominio a medida que ocurren — a menudo más claro que una IP desnuda, ya que un hostname te dice **qué**, no solo **dónde**.
HISTORY	Todo lo que la sesión ha capturado, como un registro desplazable — útil para atrapar algo breve que ya no es una conexión activa cuando lo miras.

Las filas donde el proceso muestra `?` (visibles en la captura) significan que la conexión se capturó pero el proceso de origen no pudo resolverse desde el PID en ese instante — común en conexiones muy breves. Un campo de **filtro** sobre la tabla acota por proceso, IP o puerto, y **EXTERNAL ONLY** oculta el tráfico puramente local.

WHAT	Telemetry → START → observa CONNECTIONS, o ordena por PROCESS o REMOTE IP para investigar algo específico.
WHY	Es la respuesta directa y concreta a "¿hay algo en esta máquina alcanzando la red que no debería" — leída directamente de las propias tablas de conexión del kernel, no de la afirmación de un proveedor sobre su propio software.
RESULT	Una imagen en vivo y precisa de cada conexión saliente, actualizada continuamente mientras está en ejecución y no en pausa.
RISK	Ninguno — enteramente de solo lectura.

Las conexiones de procesos sin una razón ordinaria para alcanzar la red por su cuenta — una shell desnuda, un intérprete de scripts en ejecución independiente — se marcan automáticamente bajo "suspicious" en la barra de estado, así una anomalía resalta en lugar de quedar enterrada entre el tráfico normal de tu navegador y los servicios del sistema.

▸ ADVANCED

Past this point: config file paths, raw commands, and what the panel is doing underneath.

10.2 Advanced

DE DÓNDE VIENEN LOS DATOS

El panel lee las mismas tablas de conexión a nivel de kernel que leería cualquier herramienta estándar, actualizadas a un intervalo corto mientras el monitoreo está activo — sin ningún driver o módulo del kernel especial propio. Los equivalentes de línea de comandos:

```
# cada socket abierto, con el proceso propietario — mismos campos que CONNECTIONS
$ sudo ss -tupn
# throughput por interfaz — mismos números que TRAFFIC
$ ip -s link
# una vista en vivo puntual, si la quieres fuera del panel
$ sudo nethogs
```

RESOLUCIONES DNS, A MANO

```
# captura en vivo de consultas DNS salientes (necesita una herramienta de captura de paquetes)
$ sudo tcpdump -i any -n port 53
```

POR QUÉ ALGUNAS FILAS MUESTRAN PROCESS COMO "?"

Emparejar una conexión en vivo con el proceso que la posee requiere leer `/proc/PID/fd` para cada proceso candidato en el momento de la captura — si la conexión se abre y cierra más rápido que

el intervalo de actualización del panel, o el proceso propietario ya ha terminado, el panel no tiene la ventana para hacer ese emparejamiento y muestra ? en lugar de adivinar. Es una limitación de tiempo, no una señal de que se esté ocultando algo.

CHAPTER 11

About

Una única ficha diagnóstica exhaustiva de la máquina, actualizada cada vez que abres la pestaña, así siempre refleja lo que ha cambiado en otros paneles sin necesidad de reabrir el Control Center.

11.1 Guía visual

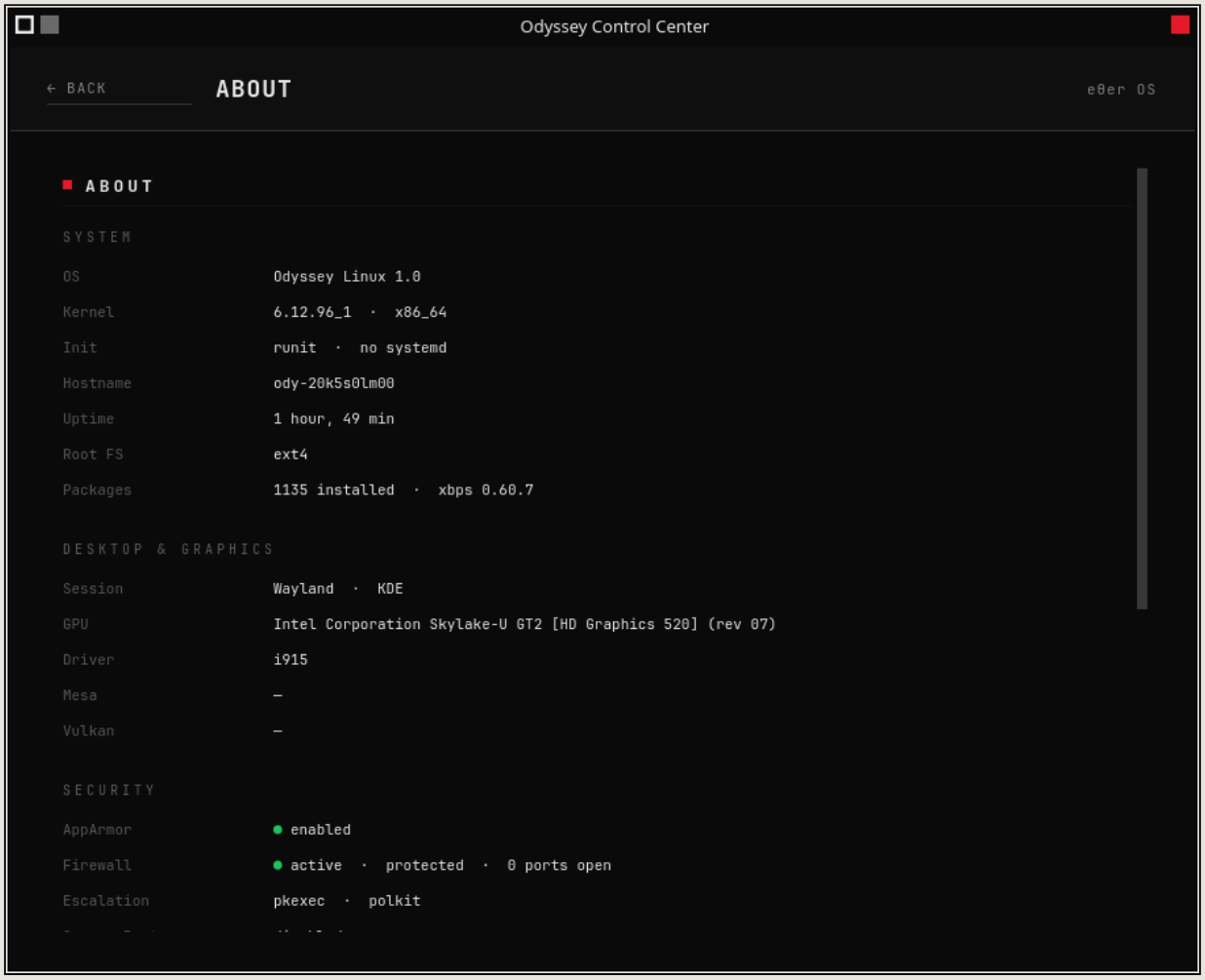

FIG · El panel desplazado hasta arriba: sección SYSTEM mostrando OS "Odyssey Linux 1.0," Kernel 6.12.96_1 · x86_64, Init runit · no systemd, Hostname, Uptime, Root FS ext4, Packages "1135 installed · xbps 0.60.7"; DESKTOP & GRAPHICS debajo con Session, GPU, Driver, Mesa, Vulkan; SECURITY comenzando a mostrar AppArmor "● enabled" y Firewall "● active · protected · 0 ports open."

Tres secciones, en orden: **SYSTEM** (kernel, init, hostname, uptime, sistema de archivos, recuento de paquetes), **DESKTOP & GRAPHICS** (tipo de sesión, GPU, driver, soporte Mesa/Vulkan), y más abajo **SECURITY** (estado de AppArmor y Firewall con indicadores coloreados, no solo texto — verde para habilitado/activo) y **HARDWARE** (placa base, audio).

WHAT	About → desplázate por las tres secciones para comprobar cualquier valor — versión del kernel, driver en uso, si AppArmor y el firewall realmente están activos.
WHY	Es la forma más rápida de confirmar el estado de todo el sistema en un solo lugar, sin abrir cinco paneles diferentes.
RESULT	Una instantánea completa y actual — refrescada en cada visita a la pestaña, así nunca muestra información obsoleta de una sesión anterior.
RISK	Ninguno — enteramente de solo lectura.

11.2 Copy report

Un solo botón en la parte superior del panel, **COPY REPORT**, pone toda la ficha en el portapapeles como texto plano — cada sección, cada valor — lista para pegar directamente en una publicación del foro o en un reporte del rastreador de errores.

WHAT	About → COPY REPORT → pega en tu reporte de error o publicación del foro.
WHY	Los reportes de errores son tan útiles como los detalles del sistema que los acompañan, y escribir a mano la versión del kernel, el driver, el estado de AppArmor y una docena de otros campos es exactamente el tipo de fricción que hace que mucha gente se salte el paso.
RESULT	Un reporte de sistema completo y correctamente formateado en el portapapeles, listo para pegar — la misma información que un maintainer de otro modo tendría que pedirte campo por campo.
RISK	Ninguno para tu sistema. El reporte es información diagnóstica (versiones, estados, modelo de hardware) — no incluye archivos personales ni credenciales — pero como con cualquier volcado diagnóstico, léelo antes de publicarlo en algún lugar público si no estás seguro de qué contiene.

▸ ADVANCED

Past this point: config file paths, raw commands, and what the panel is doing underneath.

11.3 Advanced

DE DÓNDE VIENE CADA CAMPO

Nada en este panel está inventado ni almacenado en caché desde el momento de la instalación — cada valor se consulta en vivo, cada vez que se abre la pestaña:

```
$ uname -r # Kernel
$ cat /proc/sys/kernel/osrelease
$ hostnamectl 2>/dev/null || hostname
$ uptime -p
$ xbps-query -l | wc -l # Packages installed
$ xbps-query -v # xbps version
$ lspci -k | grep -A3 VGA # GPU + Driver
$ glxinfo | grep “OpenGL version” # Mesa
$ vulkaninfo –summary # Vulkan
```

Sección Security en concreto:

```
$ sudo aa-status –enabled && echo enabled
$ sudo nft list ruleset | head -1 # firewall active/mode
```

POR QUÉ VALE LA PENA REVISAR ABOUT PRIMERO

Como vuelve a consultar el sistema en cada visita, About es la forma más rápida de confirmar que un cambio hecho en otro panel realmente surtió efecto — aplica un modo de firewall en Firewall, luego revisa la sección Security de About en lugar de reabrir Firewall mismo. Está construido para ser la comprobación cruzada de “¿eso realmente funcionó?” para el resto del Control Center.

CHAPTER 12

NVIDIA driver

PENDIENTE DE INTEGRACIÓN

Este capítulo es un marcador de posición. El módulo NVIDIA solo aparece en el Control Center en máquinas con hardware NVIDIA detectado, y las capturas de pantalla para él solo se pueden obtener en una máquina de ese tipo — se proporcionarán por separado y este capítulo se completará en una futura revisión de esta guía. Se mantiene al final, después de About, precisamente para que añadirlo más adelante no desplace la numeración de páginas ni las referencias de sección de ningún otro capítulo.

Lo que ya se sabe y anclará el capítulo terminado: el panel lleva una insignia **BETA** visible y una nota inicial de que el propio driver es propietario — empaquetado por Void a partir del instalador oficial de NVIDIA con un checksum fijado, pero no auditable de forma independiente, la misma honestidad que el resto de Odyssey se aplica a sí mismo. Detecta tu GPU específica comparándola con una base de datos integrada y elige automáticamente la rama de driver correcta (current/Turing-y-más-reciente, o una de tres ramas legacy para tarjetas más antiguas, recurriendo al driver de código abierto `nouveau` cuando nada propietario aplica). La instalación corre como una tubería rastreada en siete pasos con un indicador de progreso visible, y el panel imprime su propio comando de recuperación por adelantado — `odyssey-nvidia --revert` desde una consola de texto — antes incluso de que hagas clic en instalar. En portátiles híbridos, aparece automáticamente una sección PRIME render-offload, que permite que un juego o programa específico corra en la GPU NVIDIA mientras todo lo demás permanece en la GPU integrada para la autonomía de la batería.