HARDENING

PROHIBIDO EL PASO

SOLO PERSONAL AUTORIZADO

LAUTARO EMALHAO

Instalación del Sistema Operativo 3
Seguridad en la BIOS 3
Contraseña en la BIOS 3
Configuración Inicial 3
Particionado de Disco 5
Proceso post-instalación y verificaciones de red 6
Deshabilitar servicios innecesarios 6
Revisión de cuentas de usuario 7
Políticas de seguridad sobre sistemas de archivos 8
Política del sistema en general e implementación de SELinux 8
Configuración y rotación de logs 10
Auditorías 11
Backups 12
Servicios/Aplicaciones 12
SSH 12
Apache 14
Configuración general 14
MariaDB 16
WordPress 18
Uptime Kuma 21
WAF 26
Conclusión 27

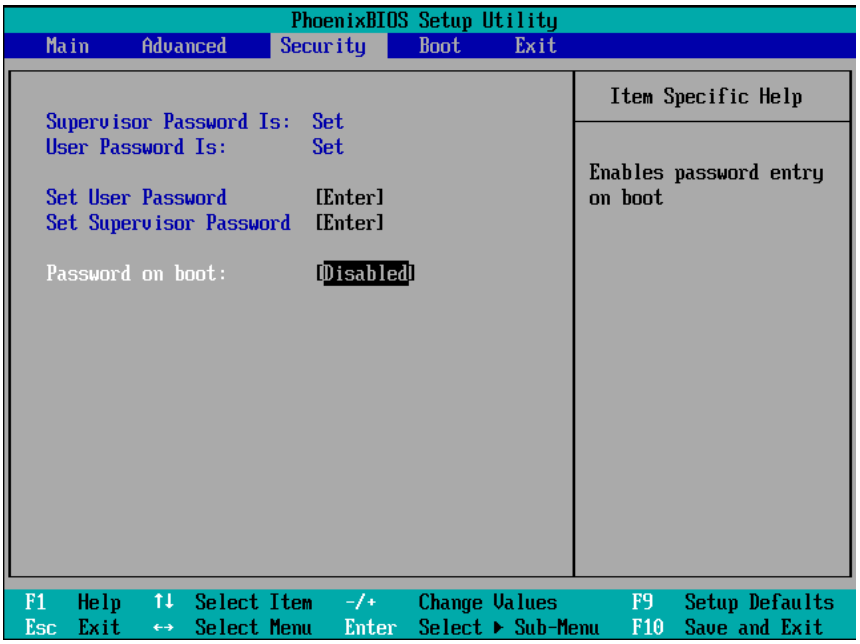
Instalación del Sistema Operativo

Seguridad en la BIOS

La protección con contraseña de la BIOS (o su equivalente) y del gestor de arranque puede evitar que usuarios no autorizados que tengan acceso físico a los sistemas inicien el equipo utilizando medios extraíbles u obtengan privilegios de root mediante el modo de usuario único.

Contraseña en la BIOS

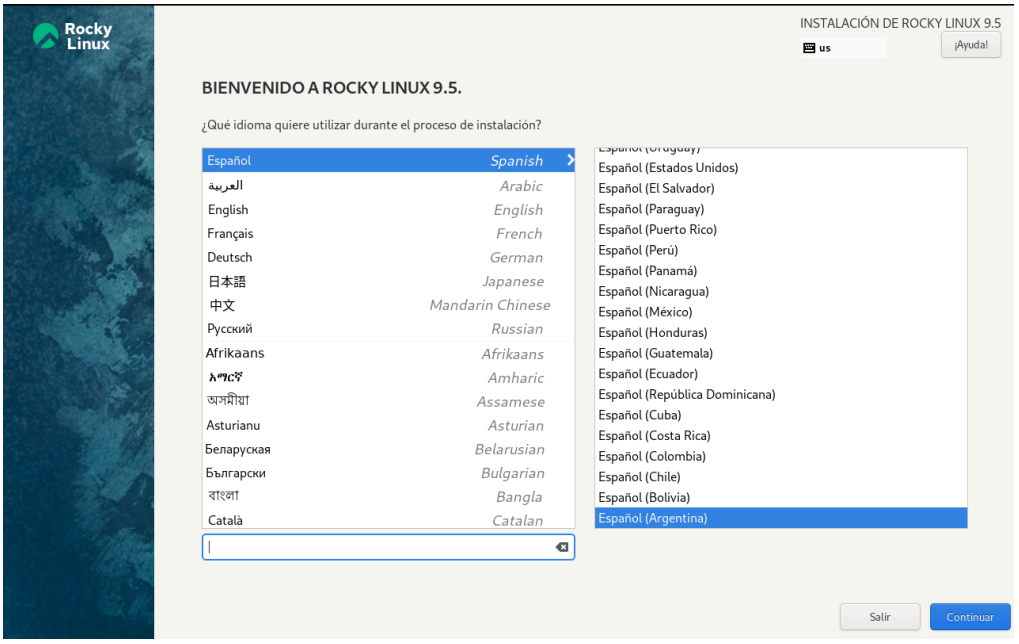
Los principales motivos para poner contraseña en la BIOS o gestor de arranque se basan en: no permitir cambios en la configuración, y agregar protección al proceso de booteo.



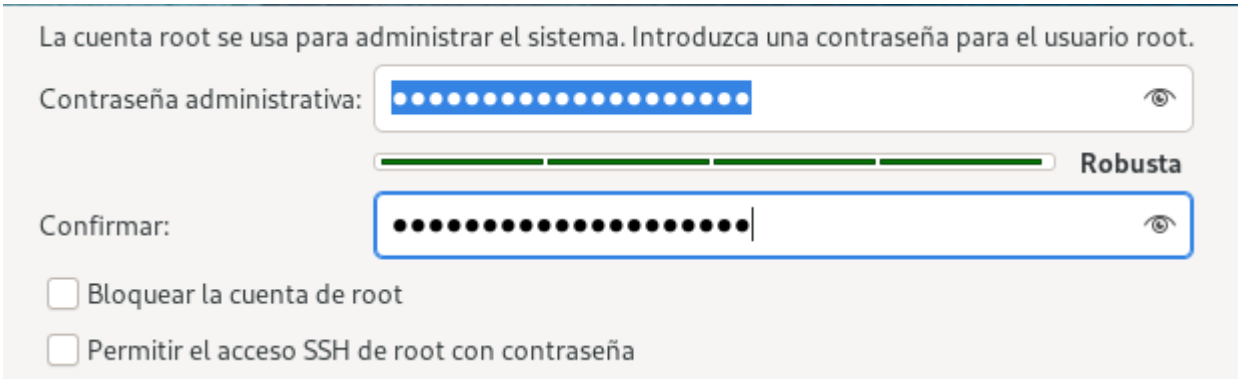
"Password on boot" permanece deshabilitado porque no es necesario poner la contraseña cada vez que se prende la máquina virtual. Sin embargo, sí se ingresa la contraseña cada vez que se intenta acceder a la BIOS.

Configuración Inicial

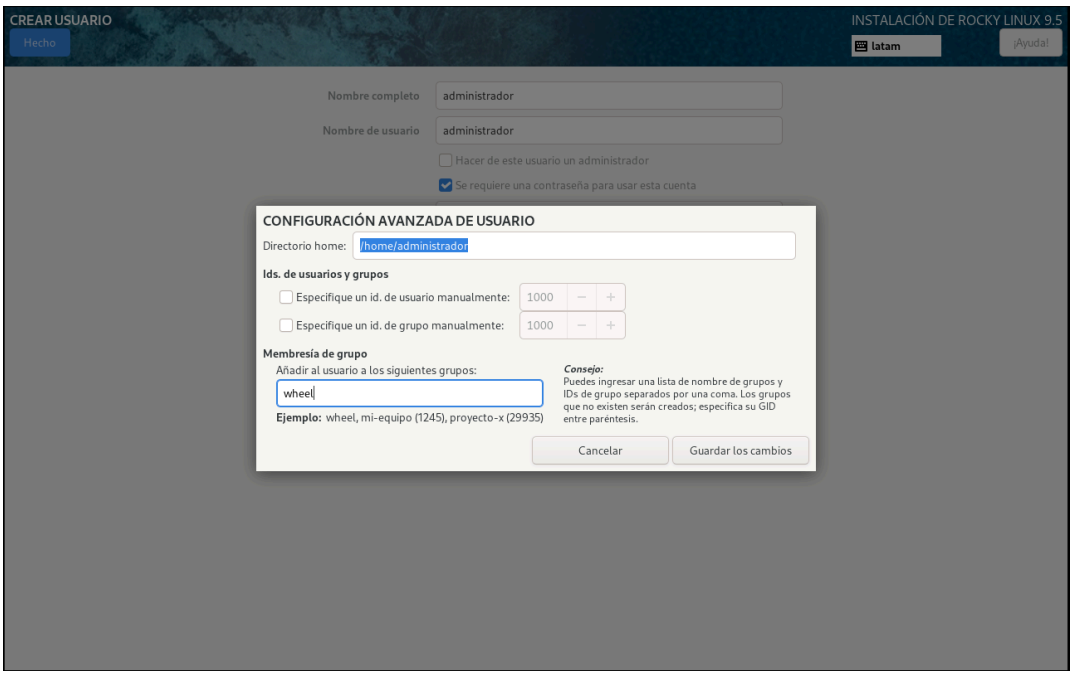
Primeramente, se selecciona el idioma y la configuración del teclado. Por cuestiones de comodidad, seleccionamos "Español".



Luego, pasamos a la configuración del sistema. Establecemos contraseña para el root y mantenemos deshabilitada la opción de acceder con SSH de root con contraseña:

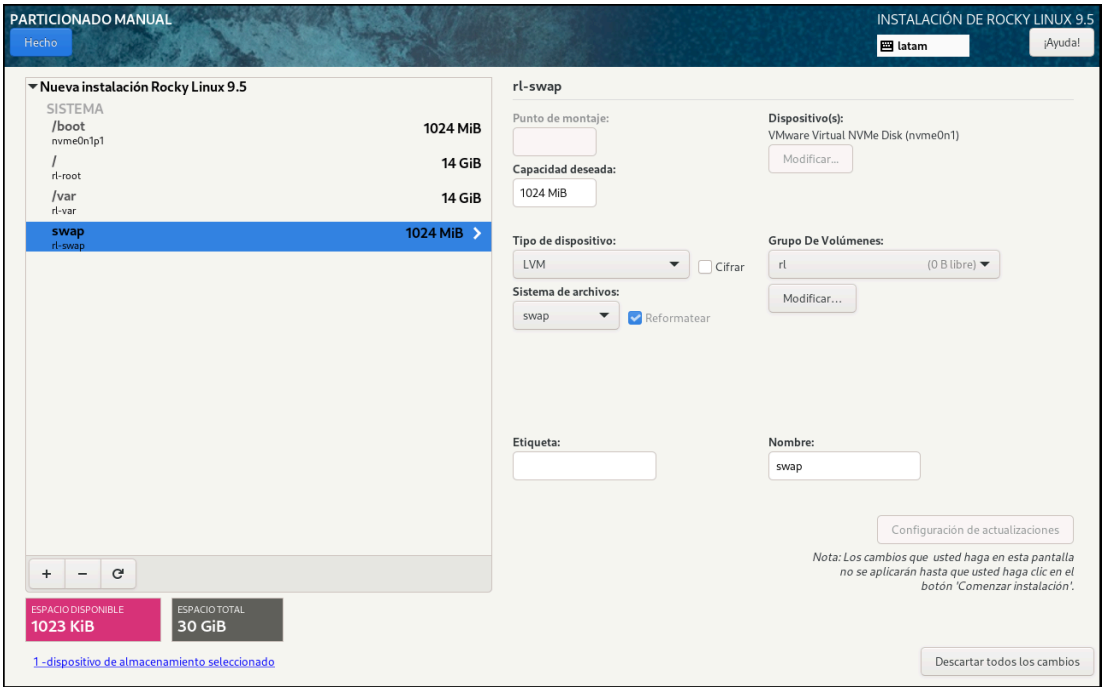


Creamos un usuario que sirva de administrador, que tenga privilegios sudo (grupo *wheel* en Rocky Linux), pero que no sea root. Esto permite tener mayor trazabilidad sobre los comandos ejecutados, puesto que se registran logs por el uso del comando sudo. En adición, respeta el principio de privilegio mínimo, puesto que solo el usuario con contraseña está autorizado a ejecutar comandos que requieran permisos elevados. Si un atacante vulnerase el sistema, necesitaría contraseña para poder hacer un daño.



Particionado de Disco

Para la configuración del disco, se crean cuatro particiones de disco. /boot almacena los archivos necesarios para arrancar (bootear) el sistema operativo; / alberga el sistema operativo, los binarios y la configuración principal; /var contiene datos variables: webs, bases de datos, logs, uploads, etc; y /swap es el área especial en disco que Linux usa como “memoria virtual” cuando la RAM física se llena.



Se encripta la partición /var con una “Frase de Paso” porque es la que va a tener los datos sensibles.

Proceso post-instalación y verificaciones de red

- Actualizar el sistema para tener siempre los últimos paquetes y tener las versiones más seguras.
    - Ingresar el comando: dnf update
- Verificar que el firewall está activado
    - systemctl status firewalld
    - systemctl start firewalld
    - systemctl enable firewalld
- Verificar los puertos activos

```
[root@localhost ~]# ss -tuln
Netid               State                Recv-Q               Send-Q                              Local Address:Port                               Peer Address:Port
udp                 UNCONN               0                    0                                   127.0.0.1:323                                   0.0.0.0:*
udp                 UNCONN               0                    0                                       [::1]:323                                      [::]:*
tcp                 LISTEN               0                    128                                   0.0.0.0:22                                    0.0.0.0:*
tcp                 LISTEN               0                    128                                      [::]:22                                       [::]:*
```

- Instalación de fail2ban
    - dnf install epel-release -y
    - dnf install fail2ban -y
    - systemctl enable fail2ban
    - systemctl start fail2ban
    - cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

```
[sshd]

# To use more aggressive sshd
# normal (default), ddos, ext
# See "tests/files/logs/sshd"
#mode   = normal
enabled = true
port = ssh
logpath = /var/log/secure
maxretry = 5
bantime = 600
findtime = 600
```

```
[apache-auth]

enabled = true
port = http,https
logpath = /var/log/httpd/error_log
maxretry = 5
bantime = 600
findtime = 600
```

Deshabilitar servicios innecesarios

Primero, verificamos qué servicios están activos y si son necesarios.

```
[root@localhost ~]# systemctl list-units --type=service --state=running
  UNIT                                LOAD   ACTIVE SUB     DESCRIPTION
  auditd.service                      loaded active running Security Auditing Service
  chronyd.service                     loaded active running NTP client/server
  crond.service                       loaded active running Command Scheduler
  dbus-broker.service                 loaded active running D-Bus System Message Bus
  firewalld.service                   loaded active running firewalld - dynamic firewall daemon
  getty@tty1.service                  loaded active running Getty on tty1
  irqbalance.service                  loaded active running irqbalance daemon
  NetworkManager.service              loaded active running Network Manager
  rsyslog.service                     loaded active running System Logging Service
  sshd.service                        loaded active running OpenSSH server daemon
  systemd-ask-password-wall.service loaded active running Forward Password Requests to Wall
  systemd-journald.service            loaded active running Journal Service
  systemd-logind.service              loaded active running User Login Management
  systemd-udevd.service               loaded active running Rule-based Manager for Device Events and Files
  user@1000.service                   loaded active running User Manager for UID 1000
```

Por caso, estos servicios son necesarios para mantener auditorías, tener el firewall y sincronización.

La lista de todos los servicios en Rocky Linux es:

```
[root@localhost ~]# systemctl list-units --type=service
  UNIT                                                                                LOAD   ACTIVE SUB     DESCRIPTION                                >
  auditd.service                                                                      loaded active running Security Auditing Service
  chronyd.service                                                                     loaded active running NTP client/server
  crond.service                                                                       loaded active running Command Scheduler
  dbus-broker.service                                                                 loaded active running D-Bus System Message Bus
  dracut-shutdown.service                                                             loaded active exited  Restore /run/initramfs on shutdown
  firewalld.service                                                                   loaded active running firewalld - dynamic firewall daemon
  getty@tty1.service                                                                  loaded active running Getty on tty1
  irqbalance.service                                                                  loaded active running irqbalance daemon
  kdump.service                                                                       loaded active exited  Crash recovery kernel arming
  kmod-static-nodes.service                                                           loaded active exited  Create List of Static Device Nodes
  lvm2-monitor.service                                                                loaded active exited  Monitoring of LVM2 mirrors, snapshots etc. using dm>
  NetworkManager-wait-online.service                                                  loaded active exited  Network Manager Wait Online
  NetworkManager.service                                                              loaded active running Network Manager
  nis-domainname.service                                                              loaded active exited  Read and set NIS domainname from /etc/sysconfig/net>
  rsyslog.service                                                                     loaded active running System Logging Service
  sshd.service                                                                        loaded active running OpenSSH server daemon
  systemd-ask-password-wall.service                                                   loaded active running Forward Password Requests to Wall
  systemd-boot-update.service                                                         loaded active exited  Automatic Boot Loader Update
  systemd-cryptsetup@luks\x2d87eb2252\x2d9fdb\x2d4dcf\x2d8cc1\x2d2c7ddd991d1f.service loaded active exited  Cryptography Setup for luks-87eb2252-9fdb-4dcf-8cc1>
  systemd-journal-flush.service                                                       loaded active exited  Flush Journal to Persistent Storage
  systemd-journald.service                                                            loaded active running Journal Service
  systemd-logind.service                                                              loaded active running User Login Management
  systemd-network-generator.service                                                   loaded active exited  Generate network units from Kernel command line
  systemd-random-seed.service                                                         loaded active exited  Load/Save OS Random Seed
  systemd-remount-fs.service                                                          loaded active exited  Remount Root and Kernel File Systems
  systemd-sysctl.service                                                              loaded active exited  Apply Kernel Variables
  systemd-tmpfiles-setup-dev.service                                                  loaded active exited  Create Static Device Nodes in /dev
  systemd-tmpfiles-setup.service                                                      loaded active exited  Create Volatile Files and Directories
  systemd-udev-trigger.service                                                        loaded active exited  Coldplug All udev Devices
  systemd-udevd.service                                                               loaded active running Rule-based Manager for Device Events and Files
  systemd-update-utmp.service                                                         loaded active exited  Record System Boot/Shutdown in UTMP
  systemd-user-sessions.service                                                       loaded active exited  Permit User Sessions
  user-runtime-dir@1000.service                                                       loaded active exited  User Runtime Directory /run/user/1000
  user@1000.service                                                                   loaded active running User Manager for UID 1000
```

Del listado, nis domain name service es un protocolo de servicio de directorio cliente-servidor para sistemas distribuidos, por lo que no es necesario. En adición, es un protocolo obsoleto y lleno de problemas de seguridad.

- systemctl disable nis-domainname.service
- systemctl stop nis-domainname.service

Revisión de cuentas de usuario

El siguiente paso es revisar las cuentas existentes en el sistema, desactivar permisos de inicio de sesión y eliminarlas si no son necesarias.

```
[root@localhost ~]# cut -d: -f1,3,7 /etc/passwd
root:0:/bin/bash
bin:1:/sbin/nologin
daemon:2:/sbin/nologin
adm:3:/sbin/nologin
lp:4:/sbin/nologin
sync:5:/bin/sync
shutdown:6:/sbin/shutdown
halt:7:/sbin/halt
mail:8:/sbin/nologin
operator:11:/sbin/nologin
games:12:/sbin/nologin
ftp:14:/sbin/nologin
nobody:65534:/sbin/nologin
tss:59:/usr/sbin/nologin
systemd-coredump:999:/sbin/nologin
dbus:81:/sbin/nologin
sssd:998:/sbin/nologin
chrony:997:/sbin/nologin
sshd:74:/usr/sbin/nologin
administrador:1000:/bin/bash
```

Por caso, no hay usuarios sospechosos. El usuario administrador fue creado al comienzo de la instalación con un propósito. Además, desactivamos acceder a root mediante ssh también al comienzo.

Políticas de seguridad sobre sistemas de archivos

```
[root@localhost ~]# ls -l /etc/passwd /etc/shadow /etc/group /etc/gshadow
-rw-r--r--. 1 root root 494 Jun  6 22:44 /etc/group
-rw-r-----. 1 root root 389 Jun  6 22:44 /etc/gshadow
-rw-r--r--. 1 root root 962 Jun  6 22:44 /etc/passwd
-rw-r-----. 1 root root 714 Jun  6 22:44 /etc/shadow
```

Se listan los permisos, propietarios, tamaño y fecha de modificación de los archivos críticos de seguridad del sistema. El detalle de los permisos es el siguiente:

- /etc/passwd: 644 (rw-r--r--)
- /etc/group: 644 (rw-r--r--)
- /etc/shadow: 640 (rw-r-----)
    - (lectura/escritura root, lectura grupo shadow)
- /etc/gshadow: 640 (rw-r-----)
    - (lectura/escritura root, lectura grupo shadow)

Política del sistema en general e implementación de SELinux

Este paso permite reforzar la seguridad general del sistema y asegurarse de que los controles de acceso obligatorios están activos.

SELinux (Security-Enhanced Linux) es un sistema de control de acceso obligatorio (MAC, Mandatory Access Control) que refuerza la seguridad del sistema operativo Linux, desarrollado originalmente por la NSA y mantenido por comunidades como Red Hat.

SELinux controla qué puede hacer cada proceso en el sistema, aunque el usuario sea root. Se basa en políticas de seguridad estrictas que definen:

- Qué archivos puede leer/escribir un programa
- Qué acciones puede ejecutar un proceso
- Qué recursos puede usar cada servicio

Verificar el estado de SELinux. Comprobar si SELinux está habilitado y en modo “enforcing”:

```
[root@localhost ~]# getenforce
Enforcing
```

Configurar SELinux para que siempre esté en "enforcing". Editar el archivo de configuración: nano /etc/selinux/config

```
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Revisar políticas de sudo. Solo usuarios autorizados deben pertenecer al grupo de administradores (wheel):

```
[root@localhost ~]# getent group wheel
wheel:x:10:administrador
```

Configurar políticas de contraseñas seguras. Editar o revisar /etc/security/pwquality.conf, /etc/login.defs y/o los módulos PAM. Por caso, se modifica /etc/security/pwquality.conf para poner una configuración en la cual la contraseña debe tener mínimo tamaño doce y tres tipos caracteres diferentes:

```
# Configuration for systemwide password quality limits
# Defaults:
#
# Number of characters in the new password that must not be present in the
# old password.
# difok = 1
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
 minlen = 12
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
# dcredit = 0
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
# ucredit = 0
#
# The maximum credit for having lowercase characters in the new password.
# If less than 0 it is the minimum number of lowercase characters in the new
# password.
# lcredit = 0
#
# The maximum credit for having other characters in the new password.
# If less than 0 it is the minimum number of other characters in the new
# password.
# ocredit = 0
#
# The minimum number of required classes of characters for the new
# password (digits, uppercase, lowercase, others).
 minclass = 3_
#
# The maximum number of allowed consecutive same characters in the new password.
# The check is disabled if the value is 0.
# maxrepeat = 0
#
# The maximum number of allowed consecutive characters of the same class in the
# new password.
# The check is disabled if the value is 0.
# maxclassrepeat = 0
```

Configuración y rotación de logs

Primero, verificamos que rsyslog está activo. Esto permite tener un registro de las actividades que ocurren en el sistema.

```
[root@localhost ~]# systemctl status rsyslog
• rsyslog.service - System Logging Service
     Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-06-08 17:29:33 -03; 14min ago
       Docs: man:rsyslogd(8)
             https://www.rsyslog.com/doc/
   Main PID: 911 (rsyslogd)
      Tasks: 3 (limit: 10856)
     Memory: 9.2M
        CPU: 186ms
     CGroup: /system.slice/rsyslog.service
             └─911 /usr/sbin/rsyslogd -n

Jun 08 17:29:32 localhost.localdomain systemd[1]: Starting System Logging Service...
Jun 08 17:29:33 localhost.localdomain systemd[1]: Started System Logging Service.
Jun 08 17:29:33 localhost.localdomain rsyslogd[911]: [origin software="rsyslogd" swVersion="8.2412.0-1.el9" x-pid="911" x-info="https://www.rsyslog.com"] st
Jun 08 17:29:33 localhost.localdomain rsyslogd[911]: imjournal: journal files changed, reloading...  [v8.2412.0-1.el9 try https://www.rsyslog.com/e/0 ]
```

Luego, verificamos que las principales carpetas que almacenan los logs funcionen:

- /var/log/secure (autenticaciones, sudo, ssh)
- /var/log/messages (eventos generales)

Por último, instalamos logrotate para que los archivos de log no llenen el disco.

```
[root@localhost ~]# cat /etc/logrotate.conf
# see "man logrotate" for details

# global options do not affect preceding include directives

# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# use date as a suffix of the rotated file
dateext

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# system-specific logs may be also be configured here.
```

Auditorías

auditd te permite auditar acciones en el sistema, especialmente cambios en archivos críticos y accesos sospechosos. Hay que instalarlo y verificar que esté corriendo en el sistema. Para verificar los logs de este programa, hay que poner: /var/log/audit/audit.log

```
[root@localhost ~]# systemctl status auditd
● auditd.service - Security Auditing Service
     Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-06-08 17:29:27 -03; 30min ago
       Docs: man:auditd(8)
             https://github.com/linux-audit/audit-documentation
    Process: 759 ExecStart=/sbin/auditd (code=exited, status=0/SUCCESS)
    Process: 763 ExecStartPost=/sbin/augenrules --load (code=exited, status=0/SUCCESS)
   Main PID: 760 (auditd)
      Tasks: 2 (limit: 10856)
     Memory: 6.8M
        CPU: 59ms
     CGroup: /system.slice/auditd.service
             └─760 /sbin/auditd

Jun 08 17:29:27 localhost augenrules[778]: enabled 1
Jun 08 17:29:27 localhost augenrules[778]: failure 1
Jun 08 17:29:27 localhost augenrules[778]: pid 760
Jun 08 17:29:27 localhost augenrules[778]: rate_limit 0
Jun 08 17:29:27 localhost augenrules[778]: backlog_limit 8192
Jun 08 17:29:27 localhost augenrules[778]: lost 0
Jun 08 17:29:27 localhost augenrules[778]: backlog 0
Jun 08 17:29:27 localhost augenrules[778]: backlog_wait_time 60000
Jun 08 17:29:27 localhost augenrules[778]: backlog_wait_time_actual 0
Jun 08 17:29:27 localhost systemd[1]: Started Security Auditing Service.
```

Es posible probar su funcionamiento mediante el comando ausearch -x su, esto audita el comando su.

Hay que crear reglas de forma permanente modificando el archivo /etc/audit/rules.d/audit.rules. Por caso, se agregan:

```
# Auditar cambios en archivos críticos
-w /etc/passwd -p wa -k passwd_changes
-w /etc/shadow -p wa -k shadow_changes
-w /etc/group -p wa -k group_changes

# Auditar uso de sudo
-w /usr/bin/sudo -p x -k sudo_usage
```

Estas reglas permiten auditar las contraseñas, los grupos y el uso del comando sudo. Para que estas reglas tengan efecto, hay que reiniciar el servicio. Es posible que al tener configurado SELinux, no se pueda reiniciar el servicio mediante el comando systemctl restart, por lo que hay que utilizar el comando service auditd reload para recargar las reglas de auditoría sin reiniciar el servicio.

```
[root@localhost ~]# auditctl -l
-w /etc/passwd -p wa -k passwd_changes
-w /etc/shadow -p wa -k shadow_changes
-w /etc/group -p wa -k group_changes
-w /usr/bin/sudo -p x -k sudo_ussage
```

Para ver eventos:

```
ausearch -k passwd_changes
ausearch -k sudo_usage
```

Backups

Es importante tener un respaldo en caso de un ataque o un cambio que afecte negativamente al sistema. Estos respaldos pueden ser copiar un archivo antes de modificarlo (pueden guardarse en otra carpeta, en otro disco o en la nube) o tomar snapshots si se está trabajando en un entorno de máquinas virtuales. Estas copias pueden hacerse antes de modificar un archivo y cuando se llega a un estado “óptimo” del sistema.

Servicios/Aplicaciones

SSH

El servicio de OpenSSH viene activado por defecto. Anteriormente configuramos de manera básica algunas restricciones de acceso.

```
[root@localhost ~]# systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-06-08 17:29:32 -03; 1h 20min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 824 (sshd)
      Tasks: 1 (limit: 10856)
     Memory: 2.7M
        CPU: 23ms
     CGroup: /system.slice/sshd.service
             └─824 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jun 08 17:29:31 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Jun 08 17:29:32 localhost.localdomain sshd[824]: Server listening on 0.0.0.0 port 22.
Jun 08 17:29:32 localhost.localdomain sshd[824]: Server listening on :: port 22.
Jun 08 17:29:32 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
```

Es posible agregar una configuración que limite el intento de accesos mediante ssh. Por defecto, el límite de intentos es 6, pero es más seguro reducir ese número. Por eso, primero copiamos el archivo original para tener un backup por si algo sale mal. Luego, se modifica el archivo y se reinicia el servicio.

```
[root@localhost ~]# cd /etc/ssh/
[root@localhost ssh]# ls
moduli      ssh_config.d        ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub  sshd_config.d
ssh_config  ssh_host_ecdsa_key  ssh_host_ed25519_key    ssh_host_rsa_key          sshd_config
[root@localhost ssh]# cp sshd_config sshd_config_original
[root@localhost ssh]# ls
moduli      ssh_config.d        ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub  sshd_config.d
ssh_config  ssh_host_ecdsa_key  ssh_host_ed25519_key    ssh_host_rsa_key          sshd_config           sshd_config_original
```

```
# Authentication:

LoginGraceTime 2m
PermitRootLogin no
StrictModes yes
MaxAuthTries 3
#MaxSessions 10
```

PermitRootLogin no deshabilita la opción de ingresar como root por ssh. StrictModes verifica que los permisos de .ssh/ sean seguros antes de aceptar una conexión.

Para ocultar la versión de Apache:

```
#Ocultar version
ServerSignature Off
ServerTokens Prod
```

```
[root@localhost ssh]# systemctl restart sshd
[root@localhost ssh]# systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-06-08 19:01:52 -03; 6s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 5627 (sshd)
      Tasks: 1 (limit: 10856)
     Memory: 1.4M
        CPU: 12ms
     CGroup: /system.slice/sshd.service
             └─5627 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jun 08 19:01:52 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Jun 08 19:01:52 localhost.localdomain sshd[5627]: Server listening on 0.0.0.0 port 22.
Jun 08 19:01:52 localhost.localdomain sshd[5627]: Server listening on :: port 22.
Jun 08 19:01:52 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
```

Apache

El primer paso para configurar apache es instalarlo y verificar que esté el servicio habilitado.

```
[root@localhost ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ 9078.057842] systemd-rc-local-generator[5982]: /etc/rc.d/rc.local is not marked executable, skipping.
[root@localhost ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@localhost ~]# systemctl start httpd
[root@localhost ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-06-08 19:59:17 -03; 4s ago
       Docs: man:httpd.service(8)
   Main PID: 5996 (httpd)
     Status: "Started, listening on: port 80"
      Tasks: 177 (limit: 10856)
     Memory: 26.2M
        CPU: 111ms
     CGroup: /system.slice/httpd.service
             ├─5996 /usr/sbin/httpd -DFOREGROUND
             ├─5997 /usr/sbin/httpd -DFOREGROUND
             ├─5998 /usr/sbin/httpd -DFOREGROUND
             ├─5999 /usr/sbin/httpd -DFOREGROUND
             └─6000 /usr/sbin/httpd -DFOREGROUND

Jun 08 19:59:17 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Jun 08 19:59:17 localhost.localdomain httpd[5996]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using localhost.local
Jun 08 19:59:17 localhost.localdomain httpd[5996]: Server configured, listening on: port 80
Jun 08 19:59:17 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
```

Configuración general

DocumentRoot (directorio de los sitios web):

Por defecto es /var/www/html. Acá se deben poner los archivos web.

Archivos principales de configuración:

/etc/httpd/conf/httpd.conf

Se modifica para permitir acceso únicamente a /var/www/html, permitir el uso de archivos .htaccess (necesario para WordPress y otras aplicaciones), evitar el listado de directorios si no hay archivo index y permitir el acceso web al contenido de esa carpeta.

```
<Directory />
    AllowOverride none
    Require all denied
</Directory>

#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#

#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html"

#
# Relax access to content within /var/www.
#
<Directory "/var/www">
    AllowOverride All
    # Allow open access:
    Options -Indexes
    Require all granted
</Directory>
```

¿Para qué se utilizan los archivos .htaccess?

Uso común	Ejemplo
Redirecciones	Redirigir de HTTP a HTTPS
Reescritura de URLs	URLs limpias: /noticia/42 en vez de noticia.php?id=42
Control de acceso	Bloquear IPs, proteger con contraseña (.htpasswd)
Configurar errores	Personalizar error 404, 403, etc.
Caché y compresión	Mejorar rendimiento del sitio

Archivos adicionales en /etc/httpd/conf.d/

Permisos y propietario:

El usuario bajo el que corre Apache en Rocky/CentOS es apache. Es fundamental que los archivos y carpetas sean accesibles, pero no demasiado permisivos.

```
[root@localhost ~]# chown -R apache:apache /var/www
[root@localhost ~]# chown -R 755 /var/www
[root@localhost ~]# ls -l /var/www
total 0
drwxr-xr-x. 2 755 apache 6 Apr 28 16:43 cgi-bin
drwxr-xr-x. 2 755 apache 6 Apr 28 16:43 html
```

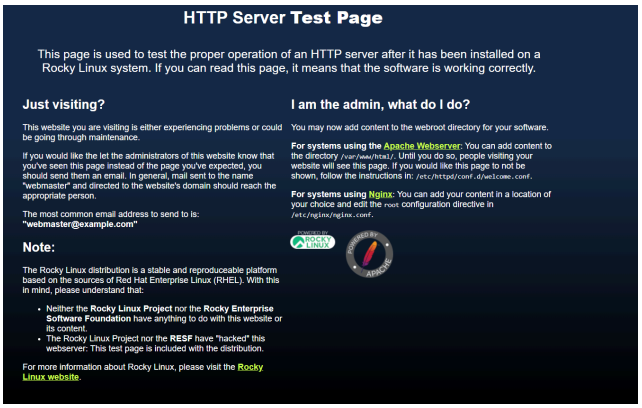
```
[root@localhost ~]# find /var/www/html -type d -exec chmod 755 {} \;
[root@localhost ~]# find /var/www/html -type f -exec chmod 644 {} \;
```

Para aplicar todos los cambios, reiniciar el servicio httpd.

Agregamos reglas en el firewall para permitir conexiones http y https.

```
[root@localhost ~]# firewall-cmd --permanent --add-service=http
success
[root@localhost ~]# firewall-cmd --permanent --add-service=https
success
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client http https ssh
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Si el procedimiento se hace correctamente, debería aparecer esta web al ingresar en http://IP/



MariaDB

Instalamos MariaDB con el comando dnf install mariadb-server, y luego iniciamos y habilitamos el servicio.

```
[root@localhost ~]# systemctl start mariadb
[root@localhost ~]# systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[ 1065.120505] systemd-rc-local-generator[3996]: /etc/rc.d/rc.local is not marked executable, skipping.
[root@localhost ~]# systemctl status mariadb
• mariadb.service - MariaDB 10.5 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
     Active: active (running) since Mon 2025-06-09 16:18:53 -03; 12s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 3939 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 14 (limit: 10856)
     Memory: 74.1M
        CPU: 459ms
     CGroup: /system.slice/mariadb.service
             └─3939 /usr/libexec/mariadbd --basedir=/usr

Jun 09 16:18:53 localhost.localdomain mariadb-prepare-db-dir[3894]: The second is mysql@localhost, it has no password either, but
Jun 09 16:18:53 localhost.localdomain mariadb-prepare-db-dir[3894]: you need to be the system 'mysql' user to connect.
Jun 09 16:18:53 localhost.localdomain mariadb-prepare-db-dir[3894]: After connecting you can set the password, if you would need to be
Jun 09 16:18:53 localhost.localdomain mariadb-prepare-db-dir[3894]: able to connect as any of these users with a password and without sudo
Jun 09 16:18:53 localhost.localdomain mariadb-prepare-db-dir[3894]: See the MariaDB Knowledgebase at https://mariadb.com/kb
Jun 09 16:18:53 localhost.localdomain mariadb-prepare-db-dir[3894]: Please report any problems at https://mariadb.org/jira
Jun 09 16:18:53 localhost.localdomain mariadb-prepare-db-dir[3894]: The latest information about MariaDB is available at https://mariadb.org/.
Jun 09 16:18:53 localhost.localdomain mariadb-prepare-db-dir[3894]: Consider joining MariaDB's strong and vibrant community:
Jun 09 16:18:53 localhost.localdomain mariadb-prepare-db-dir[3894]: https://mariadb.org/get-involved/
Jun 09 16:18:53 localhost.localdomain systemd[1]: Started MariaDB 10.5 database server.
```

Luego, se ejecuta el script de seguridad. Esto ayuda a poner una contraseña para root, eliminar usuarios anónimos, deshabilitar el acceso remoto de root y borrar la base de datos de prueba:

mysql_secure_installation

Hará varias preguntas:

```
[root@localhost ~]# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
 ... Success!


You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
 ... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.
```

```
Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Hay que acceder a MariaDB y crear la base de datos que vamos a utilizar, como también el usuario con el cual vamos a acceder a la base de datos.

```
[root@localhost ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.5.27-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER 'admin'@'localhost' IDENTIFIED BY 'superseguro';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'admin'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> EXIT;
Bye
```

WordPress

Dentro de /var/www/html instalamos WordPress. Primero instalamos tar para poder descomprimir archivos (dnf install tar) y después se ejecutan los siguientes comandos:

- curl -O https://wordpress.org/latest.tar.gz
- tar -xzvf latest.tar.gz

Debe quedar esto:

```
[root@localhost html]# ls
latest.tar.gz  wordpress
```

Instalamos las librerías de php necesarias para el correcto funcionamiento. Para una instalación básica y funcional de WordPress se necesita:

- php: El lenguaje base que WordPress necesita.
- php-mysqlnd: Para conectar WordPress con MariaDB/MySQL.

- php-gd: Para procesamiento de imágenes (subidas, redimensión, etc.).
- php-xml y php-mbstring: Para que WordPress procese textos y datos correctamente.
- php-curl: Para conexiones externas (actualizaciones, plugins, etc).
- php-json y php-zip: Para varias funciones modernas de WordPress y algunos plugins.

Se copia el contenido de la carpeta wordpress al directorio /var/www/html y se borra latest.tar.gz y el directorio wordpress. Debe quedar algo como esto:

```
[root@localhost html]# ls
index.php    wp-activate.php     wp-comments-post.php  wp-cron.php        wp-load.php   wp-settings.php   xmlrpc.php
license.txt  wp-admin            wp-config-sample.php  wp-includes        wp-login.php  wp-signup.php
readme.html  wp-blog-header.php  wp-content            wp-links-opml.php  wp-mail.php   wp-trackback.php
```

Los siguientes pasos son:
Copiar y editar el archivo de configuración: cp wp-config-sample.php wp-config.php
nano wp-config.php
Completa los datos de la base de datos:
define('DB_NAME', 'nombre_de_tu_base_de_datos');
define('DB_USER', 'usuario_de_tu_base_de_datos');
define('DB_PASSWORD', 'contraseña_de_tu_base_de_datos');
define('DB_HOST', 'localhost');

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'admin' );

/** Database password */
define( 'DB_PASSWORD', 'superseguro' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

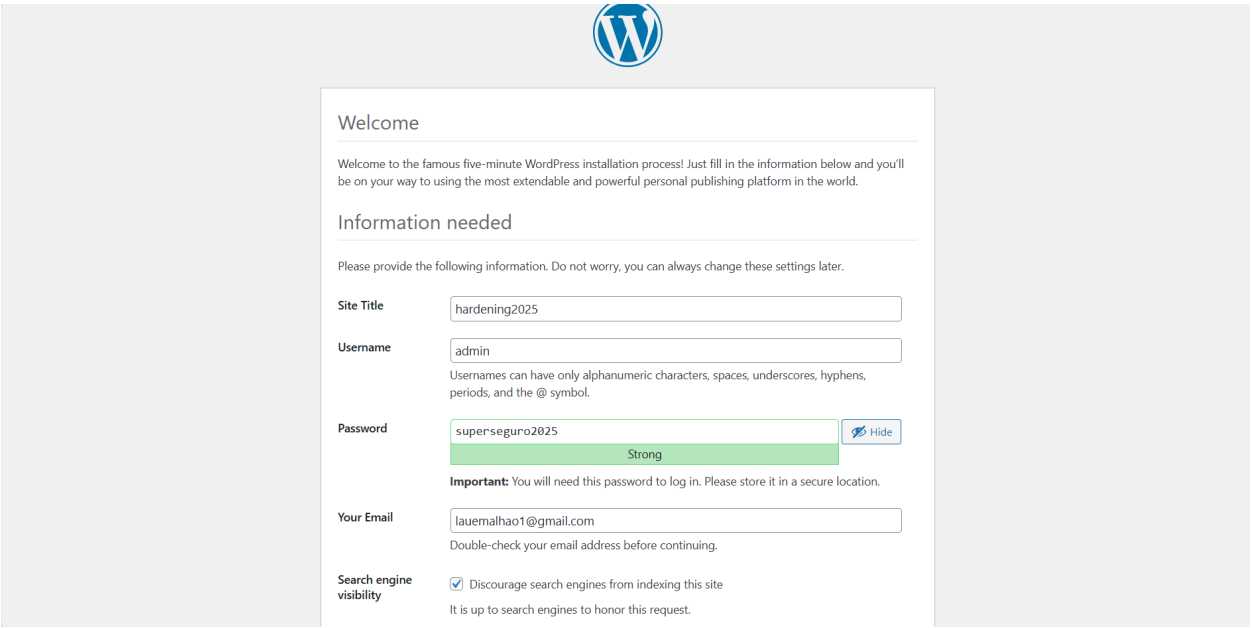
Para tener acceso a la web de WordPress, es necesario borrar la web por defecto de Apache.

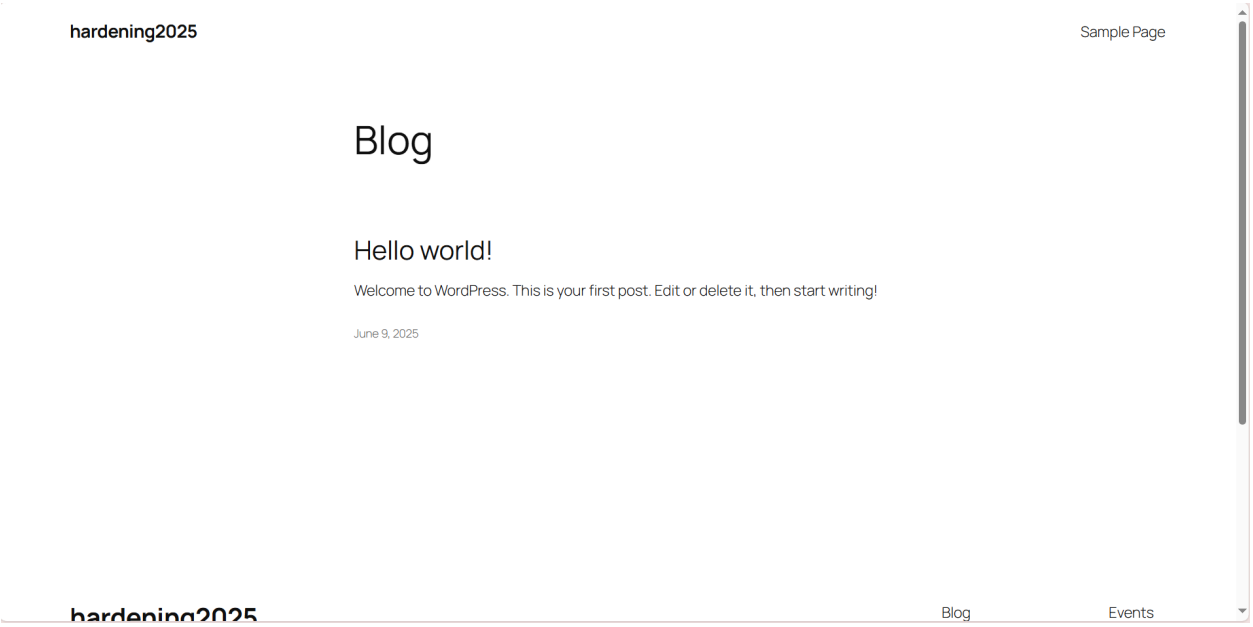
```
[root@localhost httpd]# ls
conf  conf.d  conf.modules.d  logs  modules  run  state
[root@localhost httpd]# cd conf.d
[root@localhost conf.d]# ls
README  autoindex.conf  php.conf  userdir.conf  welcome.conf
[root@localhost conf.d]# cp welcome.conf welcome.conf.bk
[root@localhost conf.d]# rm welcome.conf
rm: remove regular file 'welcome.conf'? y
```

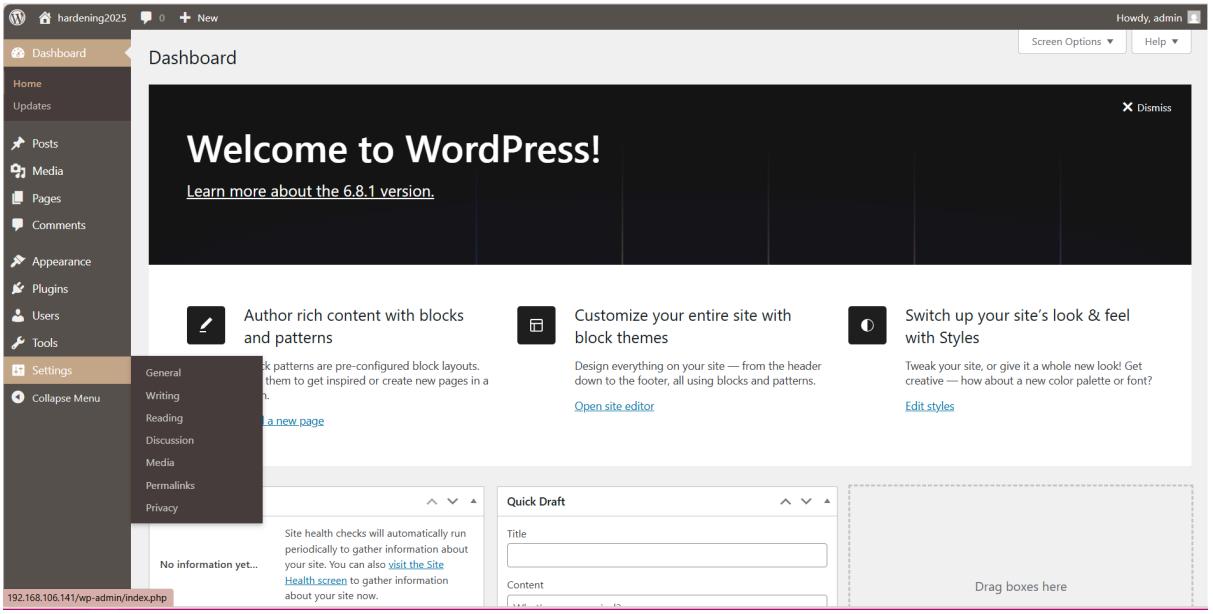
Al reiniciar el servicio de httpd, debe aparecer la página de WordPress al ingresar a http://IP/ y es posible configurarla.

La opción “Discourage search engines from indexing this site” (Disuadir a los motores de búsqueda de indexar este sitio) significa que, si se activa, WordPress añadirá una instrucción especial en el sitio web llamada robots.txt y/o meta-etiquetas que le dice a los motores de búsqueda (como Google, Bing, etc.) que no indexen ese sitio ni lo muestren en los resultados de búsqueda. Esto es útil en entornos de prueba o cuando la web todavía no está lista para ser accedida por el público.

Una vez creado el usuario e instalado WordPress, aparecerá la página principal con la información que se personalice.







Uptime Kuma

Uptime Kuma es una aplicación de monitoreo de estado (status monitoring) autohospedada, de código abierto, similar a servicios como Uptime Robot. Permite monitorear:

- Sitios web (HTTP/HTTPS)
- Pings ICMP
- Servicios TCP, UDP y DNS
- Certificados SSL (vencimiento)
- Integraciones con alertas (Telegram, Discord, correo, etc.)

Se utiliza para:

- Supervisar la disponibilidad de servicios en red (por ejemplo, servidores, APIs, sitios web, bases de datos, etc.)
- Obtener alertas instantáneas si un servicio falla o se recupera

- Ver métricas históricas de disponibilidad
- Monitorización ligera en entornos caseros o empresariales

Para instalarlo, primero se descargan paquetes necesarios:

- curl -fsSL https://rpm.nodesource.com/setup_18.x | bash -
- dnf install nodejs
- dnf install git
- npm install -g pm2

Luego, se instala el servicio que vamos a utilizar:

- cd /opt
- git clone https://github.com/louislam/uptime-kuma.git
- chown -R root:root uptime-kuma
- cd uptime-kuma
- npm run setup

Esto instala todas las dependencias y compila la aplicación.

Dentro del directorio /opt/uptime-kuma, se ejecuta:

```
[root@localhost uptime-kuma]# pm2 start server/server.js -- name uptime-kuma

                          -------------

__/\\\\\\\\\\\\\____/\\\\____________/\\\\____/\\\\\\\\\_____
 _\/\\\/////////\\\_\/\\\\\\________/\\\\\\__/\\\/////////\\\___
  _\/\\\_______\/\\\_\/\\\//\\\____/\\\//\\\_\///______\//\\\__
   _\/\\\\\\\\\\\\\/__\/\\\\///\\\/\\\/_\/\\\___________/\\\/___
    _\/\\\/////////____\/\\\__\///\\\/___\/\\\________/\\\//_____
     _\/\\\_____________\/\\\____\///_____\/\\\_____/\\\//________
      _\/\\\_____________\/\\\_____________\/\\\___/\\\/___________
       _\/\\\_____________\/\\\_____________\/\\\__/\\\\\\\\\\\\\\\_
        _\///______________\///______________\///__\///////////////__


                      Runtime Edition

       PM2 is a Production Process Manager for Node.js applications
                    with a built-in Load Balancer.

              Start and Daemonize any application:
              $ pm2 start app.js

              Load Balance 4 instances of api.js:
              $ pm2 start api.js -i 4

              Monitor in production:
              $ pm2 monitor

              Make pm2 auto-boot at server restart:
              $ pm2 startup

              To go further checkout:
              http://pm2.io/


                      -------------

[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /opt/uptime-kuma/server/server.js in fork_mode (1 instance)
[PM2] Done.
```

id	name	mode	↻	status	cpu	memory
0	server	fork	0	online	0%	43.1mb

Para que el servicio siga funcionando incluso después de un reinicio:

```
[root@localhost uptime-kuma]# pm2 startup
[PM2] Init System found: systemd
Platform systemd
Template
[Unit]
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target

[Service]
Type=forking
User=root
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Environment=PATH=/root/.local/bin:/root/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin:/usr/local/sbin:/usr/
local/bin:/usr/sbin:/usr/bin
Environment=PM2_HOME=/root/.pm2
PIDFile=/root/.pm2/pm2.pid
Restart=on-failure

ExecStart=/usr/lib/node_modules/pm2/bin/pm2 resurrect
ExecReload=/usr/lib/node_modules/pm2/bin/pm2 reload all
ExecStop=/usr/lib/node_modules/pm2/bin/pm2 kill

[Install]
WantedBy=multi-user.target

Target path
/etc/systemd/system/pm2-root.service
Command list
[ 'systemctl enable pm2-root' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-root.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-root...
[PM2] [v] Command successfully executed.
[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup systemd
```

```
[root@localhost uptime-kuma]# pm2 status
```

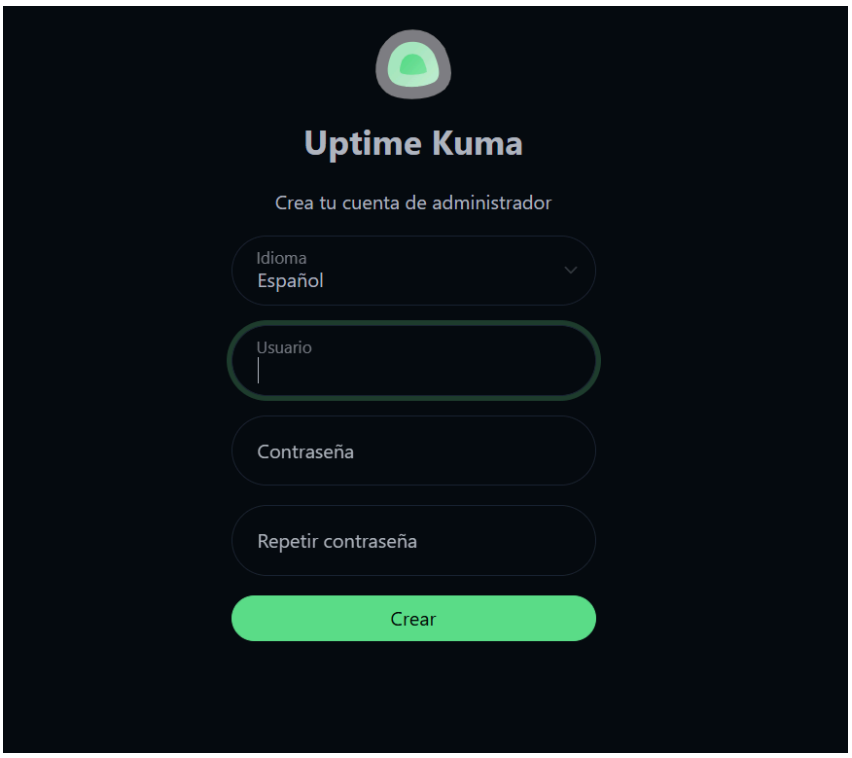
id	name	mode	↺	status	cpu	memory
0	server	fork	0	online	0%	140.2mb

```
[root@localhost uptime-kuma]# ss -tuln | grep 3001
tcp   LISTEN 0      511                *:3001             *:*
```

Se modifican las reglas del firewall para poder acceder al puerto donde escucha Uptime-Kuma.

```
[root@localhost uptime-kuma]# firewall-cmd --permanent --add-port=3001/tcp
success
[root@localhost uptime-kuma]# firewall-cmd --reload
success
```

La pantalla principal de Uptime-Kuma es:



Los pasos para configurar son:

- Iniciar sesión en la interfaz web
- Ir al navegador y entrar a:
    - http://<ip>:3001

- Iniciar sesión con un usuario administrador (o crear uno si no se ha hecho).
- Abrir la configuración de notificaciones
- En la barra lateral izquierda, hacer clic en **"Settings" (Configuración)**.
- Luego seleccionar la pestaña **"Notification"**.
- Añadir un nuevo canal de notificación
- Hacer clic en el botón ➕ Add New Notification.

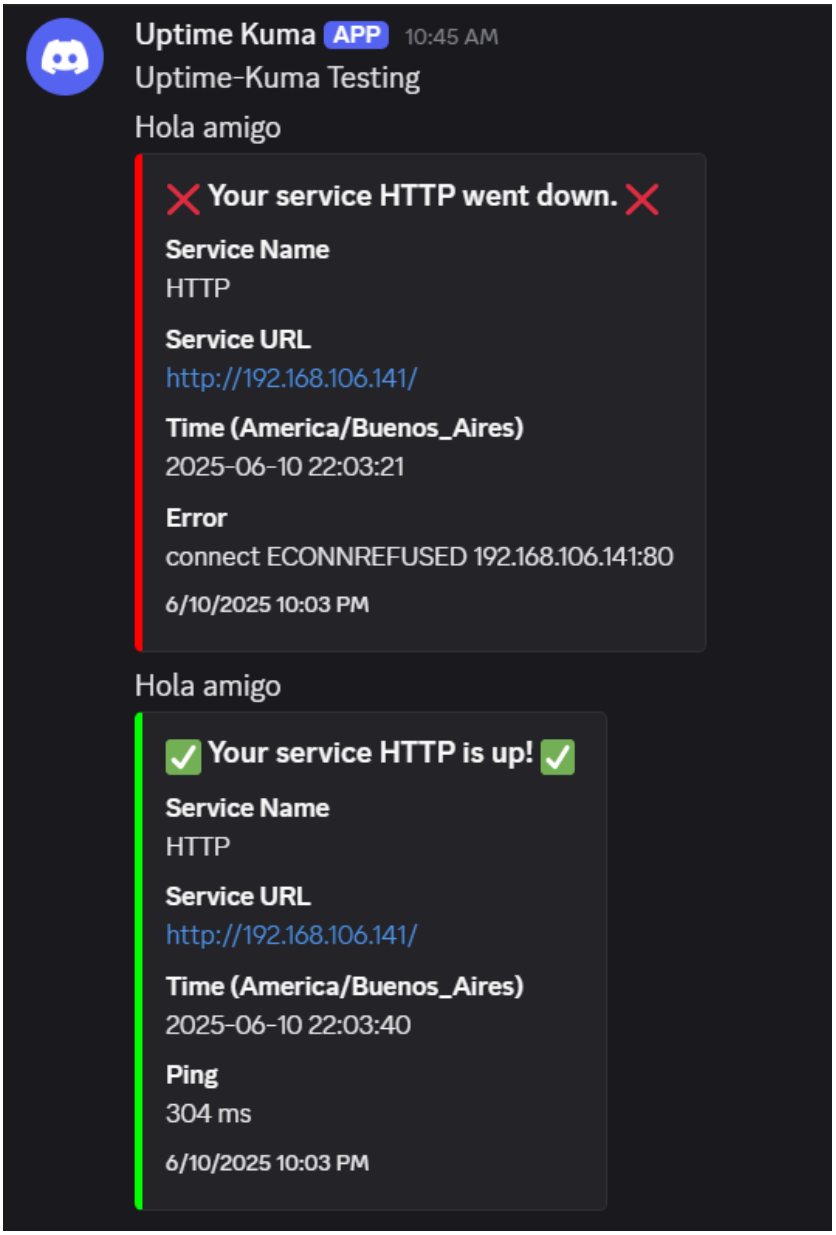
Se ve una lista larga de servicios compatibles, entre ellos:

- **Telegram**
- **Email (SMTP)**
- **Discord**
- **Slack**
- **Pushbullet**

- **Webhooks**
- **Gotify**
- **Pushover**
- **Signal** (requiere contenedor externo)
- **Matrix**, etc.

Luego de configurar qué se quiere monitorear y cómo notificar, se ve una pestaña como esta:

Ahora, como para ejemplificar se configuró que Uptime-Kuma envíe mensajes por discord, cada vez que el servidor se caiga mandará mensajes como estos:



WAF

Para mejorar la seguridad en capa de aplicación, hay que instalar mod security en Rocky Linux. Una vez instalado, se habilita en apache. Para verificarlo, ingresamos:

```
[root@localhost ~]# httpd -M | grep security
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using localhost.localdomain. Set the 'ServerName' directive globally to s
uppress this message
 security2_module (shared)
```

Que aparezca security2_module (shared) es una buena señal.

Hay que descargar un paquete de reglas, para eso, se clona un repositorio de github.

- cd /etc/httpd/
- sudo git clone https://github.com/coreruleset/coreruleset.git modsecurity-crs
- cd modsecurity-crs
- sudo cp crs-setup.conf.example crs-setup.conf

Luego, se edita mod_security.conf y se agregan los archivos relacionados al directorio con las nuevas reglas:

```
# ModSecurity Core Rules Set and Local configuration
    IncludeOptional modsecurity.d/*.conf
    IncludeOptional modsecurity-crs/crs-setup.conf
    IncludeOptional modsecurity-crs/rules/*.conf
    IncludeOptional modsecurity.d/activated_rules/*.conf
    IncludeOptional modsecurity.d/local_rules/*.conf
```

Dentro de rules se almacenan todas las reglas que se tienen que controlar en capa 7 que proporciona mod-security:

```
[root@localhost rules]# ls
REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example  REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION.conf  lfi-os-files.data
REQUEST-901-INITIALIZATION.conf                      REQUEST-944-APPLICATION-ATTACK-JAVA.conf             php-errors-pl2.data
REQUEST-905-COMMON-EXCEPTIONS.conf                   REQUEST-949-BLOCKING-EVALUATION.conf                 php-errors.data
REQUEST-911-METHOD-ENFORCEMENT.conf                  RESPONSE-950-DATA-LEAKAGES.conf                      php-function-names-933150.data
REQUEST-913-SCANNER-DETECTION.conf                   RESPONSE-951-DATA-LEAKAGES-SQL.conf                  php-variables.data
REQUEST-920-PROTOCOL-ENFORCEMENT.conf                RESPONSE-952-DATA-LEAKAGES-JAVA.conf                 restricted-files.data
REQUEST-921-PROTOCOL-ATTACK.conf                     RESPONSE-953-DATA-LEAKAGES-PHP.conf                  restricted-upload.data
REQUEST-922-MULTIPART-ATTACK.conf                    RESPONSE-954-DATA-LEAKAGES-IIS.conf                  scanners-user-agents.data
REQUEST-930-APPLICATION-ATTACK-LFI.conf              RESPONSE-955-WEB-SHELLS.conf                         sql-errors.data
REQUEST-931-APPLICATION-ATTACK-RFI.conf              RESPONSE-959-BLOCKING-EVALUATION.conf                ssrf.data
REQUEST-932-APPLICATION-ATTACK-RCE.conf              RESPONSE-980-CORRELATION.conf                        unix-shell.data
REQUEST-933-APPLICATION-ATTACK-PHP.conf              RESPONSE-999-EXCLUSION-RULES-AFTER-CRS.conf.example  web-shells-asp.data
REQUEST-934-APPLICATION-ATTACK-GENERIC.conf          iis-errors.data                                      web-shells-php.data
REQUEST-941-APPLICATION-ATTACK-XSS.conf              java-classes.data                                    windows-powershell-commands.data
REQUEST-942-APPLICATION-ATTACK-SQLI.conf             java-errors.data
```

Un dato para tener en cuenta es que si SecRuleEngine figura como On en el archivo mod_security.conf, va a prohibir el acceso a /wp-admin porque lo detecta como un ataque. Para solucionar esto, mientras tengan que hacerse cambios en la página se debe marcar como DetectionOnly. Si se desea, posterior a las configuraciones de la página, se puede volver a marcar como SecRuleEngine On.

Conclusión

El proceso de hardening realizado sobre Rocky Linux 9.5 permitió fortalecer significativamente la seguridad del sistema, reduciendo la superficie de ataque y alineando su configuración con buenas prácticas reconocidas, basadas en los manuales de *CIS Benchmarks* y *Red Hat Enterprise Linux*. A través de la eliminación de servicios innecesarios, la correcta configuración de permisos, la implementación de políticas de auditoría y de logs, la aplicación de actualizaciones, la configuración de dos firewalls (uno en las capas de red y transporte, y otro en la capa de aplicación), así como la adecuada configuración de los servicios y aplicaciones utilizados, se logró un entorno más robusto y confiable.

Este trabajo permite una aproximación al contexto de servidores reales, en los cuales deben aplicarse aún más medidas de seguridad. Además, deja en claro que el hardening no es una tarea única, sino un proceso continuo de monitoreo, revisión y actualización periódica, necesario para mantener la integridad y seguridad del sistema frente a amenazas en constante evolución.

Fuentes

Center for Internet Security. (s.f.). *CIS Benchmarks*. CIS. Recuperado el 13 de junio de 2025, de https://www.cisecurity.org/cis-benchmarks

Red Hat. (s.f.). *Security hardening - Red Hat Enterprise Linux 9*. Red Hat Documentation. Recuperado el 13 de junio de 2025, de https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html-single/security_hardening/index

OWASP ModSecurity Core Rule Set. (s.f.). *ModSecurity - Frequently Asked Questions (FAQ)*. GitHub. Recuperado el 13 de junio de 2025, de https://github.com/owasp-modsecurity/ModSecurity/wiki/ModSecurity-Frequently-Asked-Questions-%28FAQ%29#user-content-What_are_the_OWASP_ModSecurity_Core_Rules_CRS_and_should_I_use_them

Louis Lam. (s.f.). *Uptime Kuma*. GitHub. Recuperado el 13 de junio de 2025, de https://github.com/louislam/uptime-kuma

Arsys. (2023, 28 de agosto). *Cómo instalar y configurar Fail2ban para mejorar la seguridad de tu servidor*. Arsys Blog. https://www.arsys.es/blog/instalar-fail2ban

WordPress. (s.f.). *Hardening WordPress*. WordPress Developer Resources. Recuperado el 13 de junio de 2025, de https://developer.wordpress.org/advanced-administration/security/hardening/

MariaDB. (s.f.). *SELinux*. MariaDB Knowledge Base. Recuperado el 13 de junio de 2025, de https://mariadb.com/kb/en/selinux/

Red Hat. (s.f.). *What is SELinux?* Red Hat. Recuperado el 13 de junio de 2025, de https://www.redhat.com/en/topics/linux/what-is-selinux