

**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
LICENCIATURA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
SEGURIDAD EN TECNOLOGIA DE COMPUTACION
EXAMEN SEMESTRAL**

REALIZADO POR:
CHRISTIAN ESPINOZA

CONTRIBUIDORES:
ADRIÁN FRANCO (Aplicación Latch)
LUIS DOCARMO (Marco Teórico)

TEMA:
SERVIDOR WEB CON
SISTEMAS DE SEGURIDAD

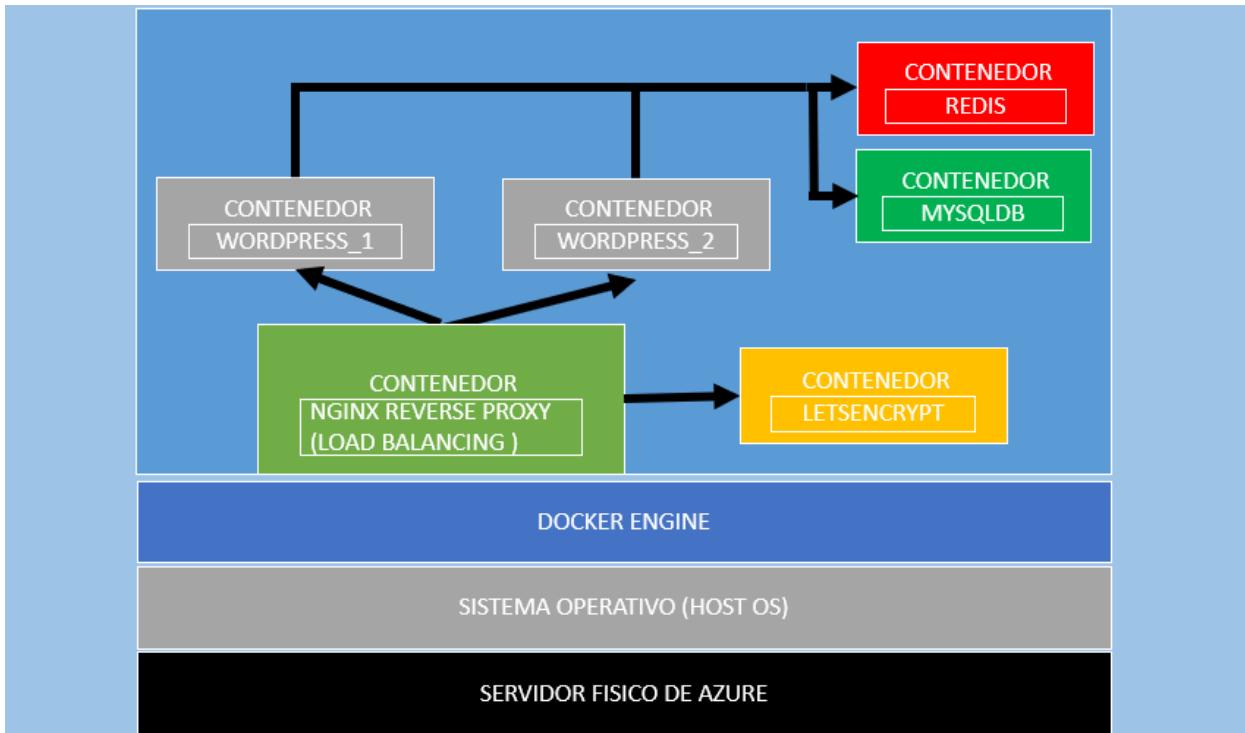
PROFESOR:
MGTR. JOSÉ MORENO

GRUPO:
IIL-153

AÑO LECTIVO:
2020

INTRODUCCIÓN

El objetivo principal es conseguir una infraestructura como se logra observar en la imagen.



Este trabajo contiene un marco teórico con información relevante a lo que se estará implementando. también una Guía de cómo se realizó el proyecto.

La tecnología de Docker usa contenedores que contienen aplicaciones que pueden trabajar como una sola. Nginx reverse proxy servirá para balancear las cargas de los contenedores. La base de datos que usaremos serán REDIS y MYSQL. Letsencrypt servirá para poder crear un certificado ssl este funciona como una extensión del nginx-proxy. La razón por la que se usara el balanceador de carga en la capa 7 es porque solo tenemos un servidor físico corriendo y lo que estaremos balanceando será las cargas en los contenedores que contiene la aplicación de wordpress.

se agregarán 2 plugins uno de redis y otro de wordfence cuando la aplicación de wordpress este corriendo. La aplicación de wordpress al final del trabajo sera escalable pero en la primera parte es colocada como dos servicios diferentes en el docker compose.

También se blindará herramientas seguridad de las cuales son importante para proteger nuestra aplicación en el servidor y los datos. el hardening de apache es una de las buenas prácticas a la hora de montar una aplicación web.

La administración de usuarios es muy importante a la hora de trabajar en equipo y la aplicación quotas y jailkit son efectiva para la limitación de tareas.

Para administrar de manera visual la implementación de webmin es maravillosa debido a que posee muchas herramientas para configurar el servidor.

Otra herramienta peculiar que vamos a implementar en nuestra aplicación web es latch. Latch es único en la perspectiva que trae a la seguridad de aplicaciones, el problema con las aplicaciones web es que para poder modificarlas uno tiene que acceder de alguna forma, y como usualmente se tratan con servidores remotos, la única forma de acceder es remotamente. Esto trae consigo problemas de seguridad, ya que, si tú puedes acceder remotamente, cualquier otra persona puede acceder remotamente. Como desarrollador podrías inhabilitar el acceso remoto, pero esto hace el desarrollo de la aplicación más complicado. Latch provee una solución, en la cual puedes inhabilitar el acceso a la máquina virtual sin tener un impacto en la productividad del desarrollador.

OBJETIVO GENERAL

Desarrollar un proyecto de seguridad en las aplicaciones, del mundo real, asegurando los tres pilares de la seguridad la confidencialidad, integridad y disponibilidad de la información protegida.

PERFIL DEL PROYECTO

Deberán implementar una metodología que salvaguarde un servidor web. Aplicar los conceptos de seguridad informática a redes de computadoras en las organizaciones con vistas a detectar fallos y mejorar la seguridad tanto en la red interna como en la posible red externa.

OBJETIVO ESPECIFICO

CONFIGURACIÓN WEB SERVER

PARTE 1

1. Implementar un balanceador de carga para alta disponibilidad
2. Instanciar dos servidores web como mínimo con dockers, debe responder por HTTPS y no por HTTP.
3. Instalar CMS Wordpress con todos los plugins de seguridad.
4. Wordpress debe usar variables de entorno para los datos de configuración.
5. La infraestructura de wordpress debe contar con REDIS.
6. Crear Certificado auto firmado de 4096 o 2048 bits TLS 1.2
7. Virtualhost

PARTE 2

1. WAF (Mod Security)
2. Instalar y configurar Mod_Security en sitio web ataques XSS
3. Instalar y configurar Mod_Evasive ataques DoS,DDoS, Fuerza bruta.
4. Instalar y configurar Mod_qos para ataques Slowloris.
5. Hardening de apache
6. Hardening de ssh: jailkit, fail2ban
7. Figlet y banner de conexión, quotas
8. Latch o pestillo digital en wordpress y ssh
9. Panel de administración Webmin

ÍNDICE

Contenido

INTRODUCCIÓN.....	2
OBJETIVO GENERAL.....	4
PERFIL DEL PROYECTO	4
OBJETIVO ESPECIFICO.....	4
CONFIGURACIÓN WEB SERVER.....	4
PARTE 1.....	4
PARTE 2.....	4
ÍNDICE.....	5
MARCO TEÓRICO	8
¿QUÉ ES UN SERVIDOR?.....	8
¿QUÉ ES PHP?	10
¿QUÉ ES APACHE?.....	11
¿QUÉ ES NGINX-PROXY RESERVE?	12
¿QUÉ ES UN CERTIFICADO AUTO FIRMADO?	13
¿QUÉ ES LET'S ENCRYPT?	14
¿QUÉ ES HTTPS?.....	15
¿QUÉ ES UN BALANCEADOR DE CARGA?.....	16
¿QUÉ ES DOCKER?.....	17
¿QUÉ ES DOCKER-COMPOSE?	17
¿QUÉ ES WORDPRESS?	18
¿QUÉ ES WORDFENCE?	18
¿QUÉ ES REDIS?	19
¿QUÉ ES REDIS OBJECT CACHE?.....	20
¿QUÉ ES MYSQL?	21
¿QUE ES WAF?	22
¿QUÉ ES INYECCIÓN SQL?.....	22
¿QUÉ ES DOS ATTACK?	23
¿QUÉ ES CLICKJACKING?	24

¿QUÉ ES XSS?	25
¿QUÉ ES MODSECURITY?	26
¿QUE ES MOD EVASIVE?	27
¿QUÉ ES MOD QOS?	27
¿QUÉ ES HARDENING DE APACHE?	28
¿QUÉ ES JAILKIT?.....	29
¿QUÉ ES FAIL2BAN?	29
¿QUÉ ES FIGLET?.....	30
¿QUÉ ES QUOTA?	30
¿LATCH O PESTILLO DIGITAL EN WORDPRESS Y SSH?.....	30
¿PANEL DE ADMINISTRACIÓN WEBMIN?	30
SERVIDOR	32
IP ESTÁTICO Y DNS, VIRTUALHOST	32
ENTRAR AL SERVIDOR CON PUTTY	33
INSTALACIÓN DE DOCKER.....	36
INSTALACIÓN DE LOS CONTENEDORES.....	40
INSTALACIÓN DE DOCKER-COMPOSE	41
Documentacion por parte del compose	47
DECLARACION.....	48
NGINX-PROXY	48
LETSENCRYPT.....	49
MYSQLDB.....	49
WORDPRESS_1 Y WORDPRESS_2.....	50
REDIS	52
CONFIGURACION DE REDIS	52
CONFIGURACION DEL WORDPRESS	53
PRUEBA DE PERCISTENCIA DE DATOS EN LOS CONTENEDORES.	62
NOTA.....	65
WAF (MOD SECURITY)	65
INSTALACION DE WAF CON CONFIGURACIÓN DE MOD_SECURITY	65

INSTALAR Y CONFIGURAR MOD_EVASIVE ATAQUES DOS, DDOS, FUERZA BRUTA..	69
INSTALAR Y CONFIGURAR MOD_QOS PARA ATAQUES SLOWLORIS.	76
HARDENING DE APACHE.....	78
ELIMINAR EL BANNER DE LA VERSIÓN DEL SERVIDOR.....	79
DESHABILITAMOS EL LISTADO DE DIRECTORIOS.....	80
TIMEOUT, LIMITREQUESTBODY, FILEETAG	81
CLICKJACKING	82
HARDENING DE SSH: JAILKIT, FAIL2BAN.....	87
FAIL2BAN	87
JAILKIT.....	93
FIGLET Y BANNER DE CONEXIÓN	104
QUOTAS	107
LATCH O PESTILLO DIGITAL EN WORDPRESS Y SSH.....	111
PANEL DE ADMINISTRACIÓN WEBMIN.....	120
GUARDAR CONTENEDOR EN REPOSITORIO	124
CONCLUSIONES	127
BIBLIOGRAFÍA.....	130

MARCO TEÓRICO

¿QUÉ ES UN SERVIDOR?

Un servidor es un equipo informático que forma parte de una red y provee servicios a otros equipos cliente.

Se denomina servidor dedicado, aquel que dedica todos sus recursos a atender solicitudes de los equipos cliente, sin embargo, un servidor compartido es aquel que no dedica todos sus recursos a servir las peticiones de los clientes, sino que también es utilizado por un usuario para trabajar de forma local.



Existen gran cantidad de tipos de servidores o roles que estos pueden desempeñar:

- Servidor de archivos: es aquel que almacena y sirve ficheros a equipos de una red.
- Servidor de Directorio Activo/Dominio: es el que mantiene la información sobre los usuarios, equipos y grupos de una red.
- Servidor de Impresión: se encarga de servir impresoras a los equipos cliente y poner en la cola los trabajos de impresión que estos generan.
- Servidor de Correo: se encarga de gestionar el flujo de correo electrónico de los usuarios, envía, recibe y almacena los correos de una organización.
- Servidor de Fax: gestiona el envío, recepción y almacenamiento de los faxes.
- Servidor Proxy: su principal función es guardar en memoria caché las páginas web a las que acceden los usuarios de la red durante un cierto tiempo, de esta forma las siguientes veces que estos acceden al mismo contenido, la respuesta es más rápida.
- Servidor Web: Almacena contenido web y lo pone al servicio de aquellos usuarios que lo solicitan.
- Servidor de Base de Datos: es aquel que provee servicios de base de datos a otros programas o equipos cliente.
- Servidor DNS: permite establecer la relación entre los nombres de dominio y las direcciones IP de los equipos de una red.

- Servidor DHCP: este dispone de un rango de direcciones con el cual, asigna automáticamente los parámetros de configuración de red IP a las máquinas cliente cuando estas realizan una solicitud.
- Servidor FTP: su función es permitir el intercambio de ficheros entre equipos, normalmente su aplicación va muy ligada a los servidores Web.

Como características necesarias de un servidor a nivel de software y hardware, podemos encontrar:

Software

Sistema Operativo: Se encarga de que el hardware funcione y logre interactuar con los servicios que corre el sistema. Algunos ejemplos son:

- Unix
- Linux
- Windows

Sistemas de archivos: Es una guía lógica que permite que el sistema pueda ubicar, ordenar y filtrar datos en el disco duro, con el fin de que podamos leerlos, modificarlos o eliminarlos.

Software servidor HTTP: Son los diferentes tipos de servidores web especializados en transmitir el contenido vía web (Apache, Nginx, IIS, Caddy, etc.).

Virtual Hosting: Permite que bajo el mismo web server e IP se alojen en varios sitios web distintos.

Despacho de ficheros estáticos y dinámicos:

- Los ficheros estáticos brindan soporte para alojar y despachar archivos como: JPG, GIF, PNG, BMP, CSS, TXT, HTML, Javascript, MP3 y MP4.
- Los ficheros dinámicos funcionan para información en PHP, ASP, Python, Ruby y GO.

Monitoreo de Red y Límites: Permite monitorear el tránsito de red, paquetes que entran y salen, así como servicios de sistema y uso de hardware como el uso del Almacenamiento, consumo de RAM, porcentaje de ocupación del CPU, velocidad de la red, rendimiento de escritura/lectura en disco.

Sistema de seguridad: El sistema de seguridad de un servidor debe:

- Imponer límites de acceso por dirección IP
- Denegar o permitirles acceso a ciertos archivos o URLs
- Solicitar usuario y contraseña para autenticación básica HTTP

- Realizar un filtrado de peticiones inseguras
- Dar soporte para despachar información cifrada con certificados de seguridad SSL vía HTTPS.

Hardware

Rack y gabinete: El rack se refiere al lugar donde se alojan los servidores físicamente y el gabinete es el armazón que sostiene los componentes de hardware de una computadora.

CPU: Es el centro de procesamiento de datos del servidor desde donde se realizan todos los cálculos lógicos y matemáticos para que el usuario pueda manipular y acceder a los datos como necesita.

Memoria RAM: Se utiliza para almacenar información y datos de forma temporal dependiendo de la demanda del usuario a través del sistema operativo.

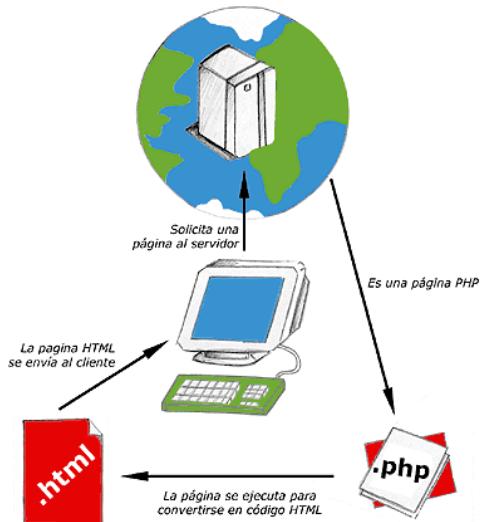
Unidades de almacenamiento: El almacenamiento de servidores web se hace en discos duros, los cuales permiten almacenar la información del sistema operativo, los servicios de sistema, y en última instancia los datos cargados por el usuario.

Puerto de red: El ancho de banda es el que te permite tener un volumen suficiente para transmitir información de ida y vuelta hacia y desde tu servidor web.

¿QUÉ ES PHP?

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.



Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores.

¿QUÉ ES APACHE?

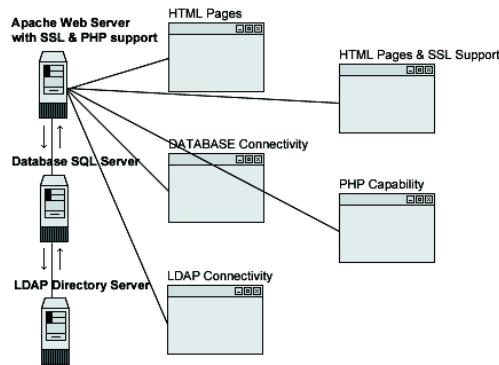
Apache es un software especializado en ofrecer servicios de servidor web. Es versátil, ligero y muy útil, además de ser completamente gratuito y de código abierto. Su popularidad es tal que, actualmente, cerca del 50% de las páginas web de todo el mundo se ejecutan en un servidor de este tipo.



Aunque se le conoce así, su nombre completo es Apache HTTP Server, y sus responsables tienen también un nombre similar: Apache Software Foundation. Esta es la firma responsable de todo el código que da forma a este software para servers que cualquiera puede utilizar sin necesidad de pagar, como también modificar a su total antojo al ser completamente abierto.

Lleva en activo desde el año 1995, tiempo más que suficiente para erigirse como el estándar que es en la actualidad. Fiable, robusto y muy flexible, permite al dueño de cualquier web publicar el contenido que desea en esta, como también gestionar todos sus ficheros de forma fácil y sencilla.

Actualmente se utiliza en plataformas Unix, Windows y Macintosh, de ahí que esté presente en la gran mayoría de páginas web de todo el mundo. Podemos hablar del software Apache como el nombre que aparece con más frecuencia en Internet, como también del responsable de que podamos entrar a la mayoría de webs en la que lo hacemos.



Apache sirve para mostrar toda la información en pantalla cuando un usuario realiza una búsqueda web. Es el software que se encarga de acceder a los ficheros alojados en el servidor para mostrar sus contenidos a petición del visitante y, así, permitir que este pueda navegar con total facilidad por la web sin toparse con problemas o contenido bloqueado.

¿QUÉ ES NGINX-PROXY RESERVE?

Es un servidor web de código abierto que, desde su éxito inicial como servidor web, ahora también es usado como proxy inverso, cache de HTTP, y balanceador de carga.

Nginx fue creado originalmente por Igor Sysoev, y tuvo su primer lanzamiento público en octubre de 2004. Igor concibió inicialmente el software como una respuesta al problema C10K, que se refiere al problema de rendimiento de manejar 10,000 conexiones concurrentes.

Debido a que sus raíces yacen en la optimización del rendimiento bajo escala, Nginx a menudo supera a otros populares servidores web en pruebas de rendimiento (Benchmarks), especialmente en situaciones con contenido estático y/o un elevado número de solicitudes concurrentes.



Nginx está diseñado para ofrecer un bajo uso de memoria y alta concurrencia. En lugar de crear nuevos procesos para cada solicitud web, Nginx usa un enfoque asincrónico basado en eventos donde las solicitudes se manejan en un solo hilo.

Con Nginx, un proceso maestro puede controlar múltiples procesos de trabajo. El proceso maestro mantiene los procesos de trabajo, y son estos lo que hacen el procesamiento real.

Algunas características comunes que se ven en Nginx son:

- Proxy inverso con caché
- IPv6
- Balanceo de carga
- Soporte FastCGI con almacenamiento en caché
- Websockets
- Manejo de archivos estáticos, archivos de índice y auto indexación
- TLS / SSL con SNI

¿QUÉ ES UN CERTIFICADO AUTO FIRMADO?

Un certificado auto firmado contiene una clave pública, información acerca del propietario del certificado y la firma del propietario. Tiene una clave privada asociada, aunque no verifica el origen del certificado a través de una tercera entidad emisora de certificados. Tras generar un certificado auto firmado en una aplicación de servidor SSL, debe:

- Extraerlo.
- Agregarlo al registro de certificados de la aplicación de cliente SSL.

Se permiten utilizar certificados auto firmados para probar una configuración SSL antes de crear e instalar un certificado firmado proporcionado por una entidad emisora de certificados.

El uso de certificados auto firmados depende de los requisitos de seguridad. Para obtener el nivel más alto de autenticación entre los componentes de software más importantes, no se debe utilizar los certificados auto firmados o hacerlo de forma selectiva. Se puede autenticar las aplicaciones que protegen los datos del servidor con certificados digitales

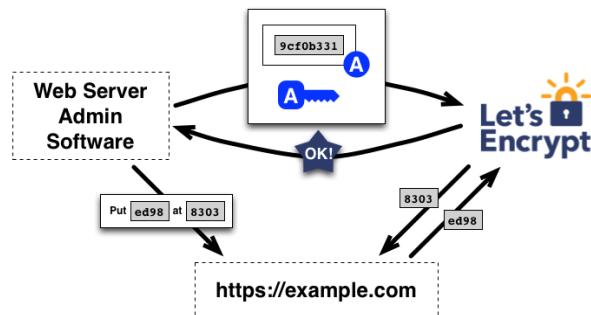
firmados y puede utilizar certificados auto firmados para autenticar navegadores web o adaptadores.

¿QUÉ ES LET'S ENCRYPT?

Es una autoridad de certificación (CA) gratuita, automatizada y abierta, que se ejecuta en beneficio del público por el Grupo de Investigación de Seguridad de Internet (ISRG).

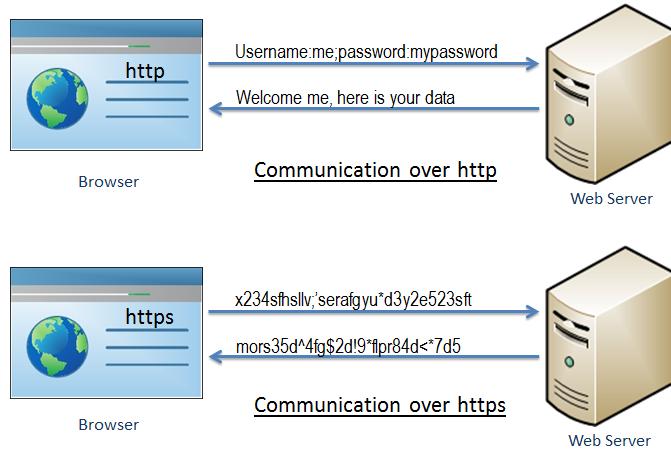
Let's Encrypt proporciona dos tipos de certificados. El SSL individual estándar y el SSL Wildcard, que no solamente cubre el dominio individual si no también todos sus subdominios. Ambos tipos de certificados Let's Encrypt se emiten por un período de 90 días, estos certificados son validados para el dominio y no requieren una dirección IP dedicada.

Tiene como objetivo hacer conexiones cifradas a servidores World Wide Web ubicua. Al eliminar el pago, la configuración del servidor web, gestión de correo electrónico de validación y las tareas de renovación del certificado, Let's Encrypt permite reducir significativamente la complejidad de la configuración y el mantenimiento de cifrado TLS. En un servidor web Linux, la ejecución de dos comandos es suficiente para configurar el cifrado HTTPS y adquirir e instalar certificados en el plazo de 20 a 30 segundos.



¿QUÉ ES HTTPS?

HTTPS se refiere a Hyper Text Transfer Protocol Secure. Es un protocolo para asegurar la comunicación entre dos sistemas, por ejemplo, el navegador y el servidor web.



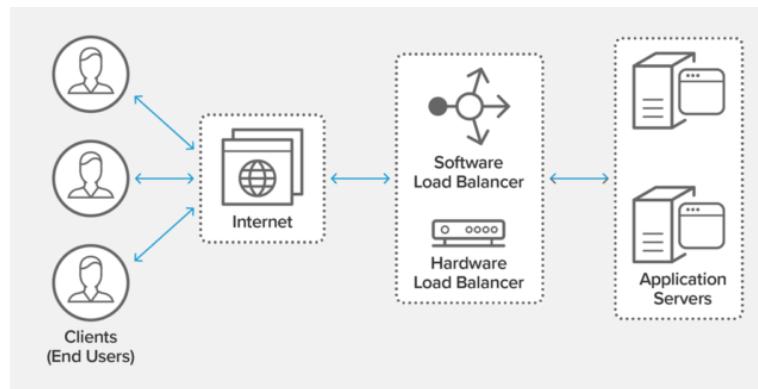
Ventaja que posee usar https:

- Comunicación segura: https establece una conexión segura al establecer un enlace cifrado entre el navegador y el servidor o cualquiera de los dos sistemas.
- Integridad de los datos: https proporciona integridad de los datos mediante el cifrado de los datos y, por lo tanto, incluso si los piratas informáticos logran atrapar los datos, no pueden leerlos ni modificarlos.
- Privacidad y seguridad: https protege la privacidad y seguridad de los usuarios del sitio web al evitar que los piratas informáticos escuchen pasivamente la comunicación entre el navegador y el servidor.
- Rendimiento más rápido: https aumenta la velocidad de transferencia de datos en comparación con http cifrando y reduciendo el tamaño de los datos.
- SEO: El uso de https aumenta la clasificación de SEO. En Google Chrome, Google muestra la etiqueta "No seguro" en el navegador si los datos de los usuarios se recopilan a través de http.
- Futuro: https representa el futuro de la web al hacer que Internet sea seguro para los usuarios y propietarios de sitios web.

¿QUÉ ES UN BALANCEADOR DE CARGA?

El balanceado de carga se refiere a la distribución eficiente del tráfico de red entrante en un grupo de servidores de fondo, también conocido como granja de servidores o grupo de servidores.

Los sitios web modernos de alto tráfico deben atender cientos de miles, sino millones, de solicitudes concurrentes de usuarios o clientes y devolver el texto, las imágenes, el video o los datos de la aplicación correctos, todo de manera rápida y confiable. Para escalar de manera rentable para cumplir con estos altos volúmenes, la mejor práctica informática moderna generalmente requiere agregar más servidores.



Un balanceador de carga actúa como el "policía de tráfico" frente a sus servidores y enruta las solicitudes de los clientes en todos los servidores capaces de satisfacer esas solicitudes de una manera que maximice la velocidad y la utilización de la capacidad y garantice que ningún servidor trabaje demasiado, lo que podría degradar el rendimiento. Si un solo servidor deja de funcionar, el balanceador de carga redirige el tráfico a los servidores en línea restantes. Cuando se agrega un nuevo servidor al grupo de servidores, el balanceador de carga comienza automáticamente a enviarle solicitudes.

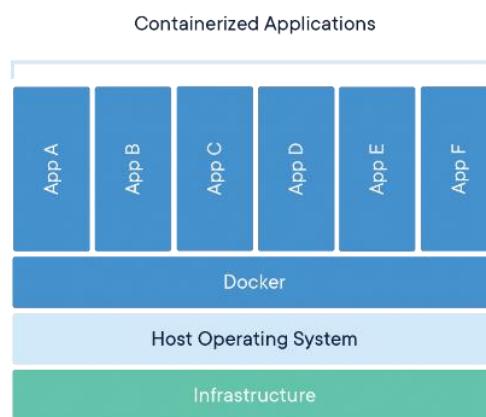
De esta manera, un balanceador de carga realiza las siguientes funciones:

- Distribuye las solicitudes de los clientes o la carga de la red de manera eficiente en varios servidores.
- Asegura alta disponibilidad y confiabilidad enviando solicitudes solo a servidores que están en línea
- Brinda la flexibilidad de agregar o restar servidores según lo exija la demanda

¿QUÉ ES DOCKER?

Docker es un programa de código abierto que permite que una aplicación Linux y sus dependencias se empaqueten como un contenedor.

La virtualización basada en contenedores aísla las aplicaciones entre sí en un sistema operativo (OS) compartido. Este enfoque estandariza la entrega del programa de la aplicación, permitiendo que las aplicaciones se ejecuten en cualquier entorno Linux, ya sea físico o virtual. Dado que comparten el mismo sistema operativo, los contenedores son portátiles entre diferentes distribuciones de Linux, y son significativamente más pequeños que las imágenes de máquinas virtuales (VM).



¿QUÉ ES DOCKER-COMPOSE?

Compose es una herramienta para definir y ejecutar aplicaciones Docker de contenedores múltiples. Con Compose, utiliza un archivo YAML para configurar los servicios de una aplicación. Luego, con un solo comando, se crea e inicia todos los servicios desde su configuración.

Compose funciona en todos los entornos:

- Producción
- puesta en escena
- Desarrollo
- Pruebas
- flujos de trabajo de CI

El procedimiento de Compose es básicamente un proceso de tres pasos:

1. Define el entorno de su aplicación con un Dockerfile para que pueda reproducirse en cualquier lugar.

2. Define los servicios que componen su aplicación docker-compose.yml para que puedan ejecutarse juntos en un entorno aislado.
3. Ejecuta docker-compose up y componer inicia y ejecuta toda su aplicación.

¿QUÉ ES WORDPRESS?

WordPress es un sistema de gestión de contenidos (CMS) que permite crear y mantener un blog u otro tipo de web.

Con casi 10 años de existencia y varios temas (plantillas) disponibles en su web oficial, no es solo un sistema sencillo e intuitivo para crear una web personal, sino que permite realizar toda clase de web más complejas.

WordPress es un sistema ideal para un sitio web que se actualice periódicamente. Si se escribe contenido con cierta frecuencia, cuando alguien accede al sitio web, puede encontrar todos esos contenidos ordenados cronológicamente (primero los más recientes y por último los más antiguos).

En muchas ocasiones se asocia WordPress con una herramienta que solo sirve para hacer blogs, utilizando WordPress podemos crear un blog, webs empresariales, tiendas online, periódico digital, central de reservas, entre muchas otras opciones.

¿QUÉ ES WORDFENCE?



Wordfence es un plugin de seguridad para WordPress que incluye muchas funcionalidades para proteger una página web, cuenta con más de 2 millones de instalaciones activas lo que hace a Wordfence el plugin de seguridad más utilizado.

Wordfence es un plugin freemium, es decir que es gratuito, pero tiene funcionalidades Premium, entre las características y funciones gratuitas de Wordfence podemos encontrar:

- Tablero de información (Wordfence Dashboard) es donde podemos ver avisos e información relevante sobre la seguridad de nuestro sitio.

- Escáner de Wordfence: Herramienta que nos permite detectar problemas en nuestra web como cambios en los archivos de WordPress, virus u otros elementos maliciosos.
- Wordfence Application Firewall: es una barrera que analiza y bloquea las peticiones a nuestra web que creen indicios o patrones de ataques y en definitiva sin ponerme muy técnico el Firewall es un filtro que bloquea Ip,s que estén intentando hacer algo malo en nuestra página web.
- Life Traffic: un visor que muestra una lista en tiempo real de los usuarios o bots que acceden a nuestra web y también muestra los usuarios y bots que bloquea el Firewall.
- Protección de ataques de fuerza bruta: Wordfence incluye configuraciones específicas para impedir ataques de fuerza bruta (Cracks con intentos masivos de contraseñas), como solo permitir contraseñas seguras, límite de intentos de inicio de sesión y bloquear nombres de usuarios entre otras.
- Bloqueo de IP: con Wordfence podemos bloquear Ip individuales, rangos de Ip, usuarios que vengan de un dominio en específico o que usen un navegador concreto.
- Limitar la velocidad de rastreo de bots y humanos: con Wordfence puedes configurar las veces que se intenta acceder o rastrear tu web y poder dificultar el rastreo o navegación por tu web a bots y usuarios sospechosos.
- Herramienta Whois Lookup: es un buscador que nos permite averiguar los datos relevantes a una ip o dominio como el nombre y email del dueño del dominio, el país de origen entre otros muchos datos.

¿QUÉ ES REDIS?

Redis es un motor de base de datos open source con licencia BSD. Basa su funcionamiento en el uso de tablas de hashes (clave – valor) y puede llegar a usarse como base de datos persistente.

Una de las particularidades de Redis es que, aparte del uso de strings, también permite datos abstractos como pueden ser:

- Sets de strings
- Listas de strings
- "Hashes" donde la clave y el valor son de tipo string



También según el tipo de valor que contenga se nos ofrecerá unas operaciones u otras. Redis nos ofrece también múltiples operaciones atómicas como inserciones, "joints", diferencias o listas ordenadas, entre otras.

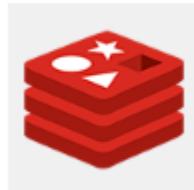
Los lenguajes soportados por Redis son los siguientes: Lua, PHP, Ruby, Python, ActionScript, C, C++, C#, Clojure, Common Lisp, Erlang, Go, Haskell, haXe, Io, Java, Node.js, Objective-C, Perl, Pure Data, Scala, Smalltalk y Tcl.

El funcionamiento general de Redis se basa en almacenar la información en memoria RAM, pero también incorpora 2 formas de hacer que esta información sea persistente. Aunque ambos procedimientos constituyen un gran costo en el rendimiento del servidor.

También permite la replicación de estos datos a un servidor esclavo y crear jerarquías en forma de árbol, dado que un servidor esclavo puede ser a su vez maestro de otros nodos.

El uso más común es para cachear archivos, dado que al almacenar la información en RAM ésta es muy rápidamente accesible.

¿QUÉ ES REDIS OBJECT CACHE?



Redis Object Cache

Por Hasta Krüss

Es un "backend" de caché de objetos persistente impulsado por Redis. Admite Predis , PhpRedis (PECL), HHVM, replicación, agrupación y WP-CLI.

Para ajustar los parámetros de conexión, prefijar las claves de caché o configurar la replicación / agrupación, consulte Otras notas.

Un backend de caché de objetos Redis de clase empresarial. Realmente confiable, altamente optimizado, totalmente personalizable y con un ingeniero dedicado cuando más lo necesita.

- Reescrito para un rendimiento bruto

- 100% compatible con la API de WordPress
- Serialización y compresión más rápidas
- Fácil depuración y registro
- Análisis de caché y precarga
- Totalmente probado en la unidad (100% de cobertura de código)
- Conexiones seguras con TLS
- Comprobaciones de estado a través de WordPress y WP CLI
- Optimizado para WooCommerce, Jetpack y Yoast SEO

¿QUÉ ES MYSQL?

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.



Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de doble licenciamiento anteriormente mencionado. La base de datos se distribuye en varias versiones, una Community, distribuida bajo la Licencia pública general de GNU, versión 2, y varias versiones Enterprise, para aquellas empresas que quieran incorporarlo en productos privativos. Las versiones Enterprise incluyen productos o servicios adicionales tales como herramienta de monitorización y asistencia técnica oficial.

Está desarrollado en su mayor parte en ANSI C y C++. tradicionalmente se considera uno de los cuatro componentes de la pila de desarrollo LAMP y WAMP.

¿QUE ES WAF?

ModSecurity es un firewall de aplicación web para el servidor web Apache. Además de proporcionar capacidades de registro, ModSecurity puede monitorear el tráfico HTTP en tiempo real para detectar ataques. ModSecurity también funciona como una herramienta de detección de intrusos, lo que le permite reaccionar ante eventos sospechosos que tienen lugar en sus sistemas web.

Aunque ModSecurity viene con una configuración predeterminada, esta guía utilizará OWASP ModSecurity Core Rule Set (CRS) versión 3.0.2. El objetivo del proyecto OWASP es "proporcionar un conjunto fácilmente 'conectable' de reglas genéricas de detección de ataques que brinden un nivel básico de protección para cualquier aplicación web", y el CRS está destinado a "proteger las aplicaciones web de una amplia gama de ataques con un mínimo de alertas falsas ". Esta versión del CRS requiere ModSecurity 2.8.0 o superior. La configuración se realiza a través de conjuntos de reglas para evitar ataques comunes como inyecciones de SQL, secuencias de comandos entre sitios y ejecución remota de código. Esta guía mostrará cómo configurar las reglas predeterminadas. Las configuraciones avanzadas se dejan como un desafío para el lector

¿QUÉ ES INYECCIÓN SQL?

Hace referencia a un ataque contra un sitio o aplicación web en el que se añade código de lenguaje de consulta estructurado (SQL) a un campo de entrada de un formulario web con el objetivo de acceder a una cuenta o modificar los datos.

Una consulta SQL es una petición de algún tipo de acción sobre una base de datos. La más habitual es la petición de un nombre de usuario y una contraseña en una página web. Dado que muchos sitios web solo supervisan la introducción de nombres de usuario y contraseñas, un hacker puede utilizar los cuadros de introducción de datos para enviar sus propias peticiones, es decir, inyectar SQL en la base de datos. De esta forma, los hackers pueden crear, leer, actualizar, modificar o eliminar los datos guardados en la base de datos back-end, normalmente para acceder a información confidencial, como los números de la seguridad social, los datos de las tarjetas de crédito u otra información financiera.

Un ataque de inyección SQL puede afectar a cualquier sitio o aplicación web que utilice una base de datos basada en SQL, es una de las formas de ciberataque más peligrosas y más antiguas, pero también más frecuentes. Lo que es todavía más preocupante: las inyecciones SQL están más vigentes que nunca, ya que ahora existen programas de

inyección SQL automatizada, lo que significa que los hackers pueden atacar y robar con más facilidad que nunca.

¿QUÉ ES DOS ATTACK?

Un ataque de denegación de servicio (DoS) es un ataque destinado a apagar una máquina o red, haciéndola inaccesible para sus usuarios previstos. Los ataques DoS logran esto inundando el objetivo con tráfico o enviándole información que desencadena un bloqueo. En ambos casos, el ataque DoS priva a los usuarios legítimos (es decir, empleados, miembros o titulares de cuentas) del servicio o recurso que esperaban.

Las víctimas de ataques DoS a menudo se dirigen a servidores web de organizaciones de alto perfil, tales como empresas bancarias, comerciales y de medios, u organizaciones gubernamentales y comerciales. Aunque los ataques DoS generalmente no resultan en el robo o la pérdida de información significativa u otros activos, pueden costarle a la víctima una gran cantidad de tiempo y dinero para manejar.

Hay dos métodos generales de ataques DoS: servicios de inundación o servicios de bloqueo. Los ataques de inundación se producen cuando el sistema recibe demasiado tráfico para que el servidor se almacene en el búfer, lo que hace que se desaceleren y finalmente se detengan. Los ataques de inundación populares incluyen:

- Ataques de desbordamiento de búfer: el ataque DoS más común. El concepto es enviar más tráfico a una dirección de red de lo que los programadores han creado para manejar el sistema. Incluye los ataques enumerados a continuación, además de otros que están diseñados para explotar errores específicos de ciertas aplicaciones o redes.
- Inundación ICMP: aprovecha los dispositivos de red mal configurados mediante el envío de paquetes falsificados que hacen ping a cada computadora en la red de destino, en lugar de solo una máquina específica. La red se activa para amplificar el tráfico. Este ataque también se conoce como el ataque de los pitufos o ping de la muerte.
- Inundación SYN: envía una solicitud para conectarse a un servidor, pero nunca completa el protocolo de enlace. Continúa hasta que todos los puertos abiertos estén saturados de solicitudes y ninguno esté disponible para que los usuarios legítimos se conecten.

Otros ataques DoS simplemente explotan vulnerabilidades que hacen que el sistema o servicio objetivo se bloquee. En estos ataques, se envían datos que aprovechan los errores en el objetivo que posteriormente bloquean o desestabilizan severamente el sistema, de modo que no se puede acceder ni usarlo.

Un tipo adicional de ataque DoS es el ataque de denegación de servicio distribuido (DDoS). Un ataque DDoS ocurre cuando varios sistemas organizan un ataque DoS sincronizado a un solo objetivo. La diferencia esencial es que, en lugar de ser atacado desde una ubicación, el objetivo es atacado desde muchas ubicaciones a la vez. La distribución de hosts que define un DDoS proporciona al atacante múltiples ventajas:

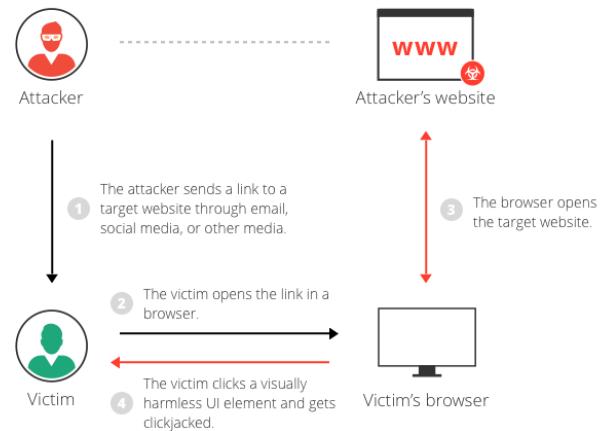
- Puede aprovechar el mayor volumen de máquina para ejecutar un ataque gravemente disruptivo
- La ubicación del ataque es difícil de detectar debido a la distribución aleatoria de los sistemas de ataque (a menudo en todo el mundo)
- Es más difícil apagar varias máquinas que una
- La verdadera parte atacante es muy difícil de identificar, ya que están disfrazados detrás de muchos sistemas (en su mayoría comprometidos)

Las tecnologías de seguridad modernas han desarrollado mecanismos para defenderse de la mayoría de los ataques DoS, pero debido a las características únicas de DDoS, todavía se considera una amenaza elevada y es de mayor preocupación para las organizaciones que temen ser atacadas por dicho ataque.

¿QUÉ ES CLICKJACKING?

Es un ataque que engaña a un usuario para que haga clic en un elemento de la página web que es invisible o disfrazado de otro elemento. Esto puede hacer que los usuarios descarguen malware involuntariamente, visiten páginas web maliciosas, proporcionen credenciales o información confidencial, transfieran dinero o compren productos en línea.

Por lo general, el clickjacking se realiza mostrando una página invisible o un elemento HTML, dentro de un iframe, en la parte superior de la página que ve el usuario. El usuario cree que está haciendo clic en la página visible, pero, de hecho, está haciendo clic en un elemento invisible en la página adicional transpuesta encima.



La página invisible podría ser una página maliciosa o una página legítima que el usuario no tenía la intención de visitar, por ejemplo, una página en el sitio bancario del usuario que autoriza la transferencia de dinero.

Ejemplo de ataques de clickjacking

- El atacante crea una página atractiva que promete ofrecer al usuario un viaje gratis a Tahití.
- En el fondo, el atacante verifica si el usuario ha iniciado sesión en su sitio bancario y, de ser así, carga la pantalla que permite la transferencia de fondos, utilizando parámetros de consulta para insertar los detalles bancarios del atacante en el formulario.
- La página de transferencia bancaria se muestra en un iframe invisible encima de la página de regalo gratuito, con el botón "Confirmar transferencia" alineado exactamente sobre el botón "Recibir regalo" visible para el usuario.
- El usuario visita la página y hace clic en el botón "Reservar mi viaje gratis".
- En realidad, el usuario hace clic en el iframe invisible y ha hecho clic en el botón "Confirmar transferencia". Los fondos se transfieren al atacante.
- El usuario es redirigido a una página con información sobre el obsequio (sin saber qué sucedió en el fondo).

¿QUÉ ES XSS?

XSS es el acrónimo usado para "Cross Site Scripting". XSS es una de las vulnerabilidades más comunes que tienen las aplicaciones web.

Estas vulnerabilidades ayudan a cualquier atacante a ejecutar cualquier código malicioso del lado del cliente en caso de que la aplicación web es vulnerable a XSS. Si el ataque

tiene éxito, permite a los atacantes robar contraseñas, redirigir a los usuarios a páginas falsas y hacer cosas muchos más atrevidas.

Daños de un ataque XSS puede causar:

- Redireccionar a sitios de phishing, o a páginas falsas.
- Robar las cookies e inicio de sesión de las cuentas de las víctimas para escalar privilegios y acceder a los sistemas.
- Inserción de enlaces en HTML para instalar software malicioso en el sistema.
- Los ataques de malware en el servidor.
- Derribar sitios Web.
- Denegación de Servicio (DoS).

¿Por qué XSS funciona y cómo se puede prevenir ataques?

XSS funciona debido a que no se validan los datos que se publican desde un campo de entrada, lo que provoca que, en cualquier secuencia de comandos en el campo de entrada, con ciertas características pueda aprovechar dichas vulnerabilidades.

El tratamiento de los datos impide casi todos los ataques XSS. Se recomienda también tener un Web Application Firewall (WAF) para una seguridad más efectiva.

Hay una cabecera HTTP que impide la ejecución de XSS en el navegador que se llama “X-XSS protection HTTP Header”.

Tipos de XSS

- Persistente: También se conoce como XSS almacenado. En este tipo de ataque XSS, el código malicioso presentado por el servidor se almacena en el servidor y se ejecuta siempre en la página.
- No persistente: También se le llama XSS reflejado, el cual es la más común de las vulnerabilidades más encontradas hoy en día. En este ataque el código presentado será enviado al servidor a través de una petición HTTP y el servidor de insertará dicho código en un archivo HTML, luego vuelve al cliente con la respuesta HTTP. Cada vez que el código se ejecuta desde el archivo HTML, se explota ese sistema.

¿QUÉ ES MODSECURITY?

ModSecurity es un grupo de herramientas para el monitoreo, registro y control de acceso de aplicaciones web en tiempo real. Su función es como un facilitador: no hay reglas estrictas que le digan qué hacer; en cambio, depende del usuario elegir su propio camino a través de las funciones disponibles.

La libertad de elegir qué hacer es una parte esencial de la identidad de ModSecurity y va muy bien con su naturaleza de código abierto. Con acceso completo al código fuente, extiende su capacidad de personalizar y extender la herramienta en sí misma para que se ajuste a sus necesidades.

Escenarios de uso:

- Monitoreo de seguridad de aplicaciones en tiempo real y control de acceso
- Registro completo de tráfico HTTP
- Evaluación continua de seguridad pasiva
- Endurecimiento de aplicaciones web

¿QUE ES MOD EVASIVE?

Los servidores web están sujetos a ataques DOS y DDOS. A veces, si los atacantes tienen suficiente ancho de banda, hay muy poco o nada que podamos hacer para detenerlos. La mayoría de las veces, somos capaces de implementar un conjunto de contramedidas que se utilizarán de manera proactiva para prevenir ataques DOS, ataques DDOS y raspado de la web. Para este propósito, instalamos y cargamos mod_evasive. El cual es capaz de determinar cuántas veces dentro de un intervalo de tiempo predefinido se puede acceder al sitio, cuántas veces se accede a un URI específico dentro de un intervalo de tiempo determinado y mucho más.

¿QUÉ ES MOD QOS?

Es un módulo de calidad de servicio (QoS) para el servidor Apache HTTP que implementa mecanismos de control que pueden proporcionar diferentes prioridades a diferentes solicitudes.

Los mecanismos de control están disponibles en los siguientes niveles:

- Control de nivel de solicitud: mod_qos controla el número de solicitudes simultáneas a un espacio de nombres (URL). Se utiliza para definir diferentes prioridades para diferentes páginas o aplicaciones dentro de un servidor web.
- Control de nivel de conexión: mod_qos controla el número de conexiones TCP al servidor web. Esto ayuda a limitar las conexiones que provienen de un único cliente o de redes desconocidas, a fin de reducir el número máximo de conexiones simultáneas a un servidor virtual o para implementar configuraciones dinámicas de mantenimiento de HTTP.
- Control de nivel de ancho de banda: acelera las solicitudes / respuestas a ciertas URL en el servidor web.

- La línea de solicitud genérica y el filtro de encabezado descartan las URL de solicitud sospechosas o los encabezados HTTP.

¿QUÉ ES HARDENING DE APACHE?

El servidor HTTP Apache es uno de los servidores web multiplataforma gratuitos y de código abierto más utilizados y populares.

Al ser la tecnología de servidor web más extendida, se convierte en uno de los servicios más vulnerables a los ataques. Al pasar por la base de datos CVE, el servidor HTTP Apache tiene alrededor de 205 vulnerabilidades a su nombre. A pesar de las vulnerabilidades, tiene un buen historial de seguridad y una gran comunidad de desarrolladores dedicada a garantizar la misma, aun así, es inevitable que se descubran algunos problemas (vulnerabilidades), críticos o bajos, en un software después de su lanzamiento. Por lo tanto, es de suma importancia que debe bloquear ciertos controles para asegurar y fortalecer la instalación de su servidor Apache HTTP para proteger sus aplicaciones, el servidor subyacente y la mayoría de todos los datos para que no entren en las manos equivocadas.

A continuación, presentamos los 7 controles más básicos para asegurar tu servidor apache de algunos de los ataques más comunes:

1. **Estar actualizado:** Siempre se recomienda tener la última versión de Apache ejecutándose para obtener la versión actualizada del software con todos los parches necesarios para cualquier vulnerabilidad existente.
2. **Ocultar el banner del servidor (versión y sistema operativo):** Se considera que este es el primer paso para fortalecer su instalación de Apache después de instalar la última versión. En la configuración predeterminada, el servidor apache muestra su versión actual y el sistema operativo subyacente. Aquí, podemos ver la versión de Apache y el sistema operativo en los Encabezados de respuesta ,esto le permite al atacante saber qué versión de Apache está ejecutando el servidor, mediante la cual puede encontrar vulnerabilidades específicas de la versión y sus vulnerabilidades.
3. **Desactive el listado de directorios:** De manera predeterminada, apache enumera los archivos en un directorio solicitando un directorio dentro de Document Root (a menos que contenga un archivo de índice). Esto permite que cualquier persona (atacante) vea todos los archivos dentro de ese directorio y también en directorios anidados.
4. **Deshabilite el método HTTP TRACE:** Por defecto, Apache permite el método TRACE HTTP. El método TRACE se utiliza para devolver la solicitud HTTP completa como respuesta, generalmente se utiliza para fines de depuración y debe desactivarse en un entorno de producción. Al habilitar el método TRACE, un atacante puede potencialmente robar información de cookies.

5. **Prevenir ClickJacking:** Clickjacking es una vulnerabilidad web bien conocida, en la que el atacante usa capas transparentes y opacas (usando CSS) para engañar a la víctima para que haga clic en un botón o enlace en otra página cuando tenía la intención de hacer clic en la página transparente superior. De esta manera, el atacante secuestra el clic del usuario para robar la información de los usuarios.
6. **Cookies con HttpOnly y conjunto de bandera segura:** La mayoría de los ataques XSS resultan en el robo de cookies. La prevención del ataque XSS se haría a nivel de aplicación, sin embargo, puede evitar el robo de cookies desde el nivel del servidor configurando el indicador HttpOnly y Secure establecido para todas las cookies en su servidor. Esto deshabilita la visualización de cookies a través de javascript y permite la manipulación de cookies utilizando solo HTTP.
7. **Prevención XSS utilizando el encabezado X-XSS-Protection:** El encabezado X-XSS-Protection se introdujo como una característica en muchos navegadores (IE 8 y superior, Chrome, Safari, que permite al navegador detener la carga de páginas cuando detectan un ataque XSS que se ejecuta en el navegador.

¿QUÉ ES JAILKIT?

Jailkit es una utilidad para enjaular usuarios dentro de un directorio y que no puedan acceder a otras partes del sistema

Es una herramienta especializada que se desarrolla con un enfoque en la seguridad. Abortará de manera segura si la configuración, la configuración del sistema o el entorno no es 100% seguro, y enviará mensajes de registro útiles que explican qué está mal en "syslo"

Jailkit es un conjunto de utilidades para limitar las cuentas de usuario a archivos específicos usando chroot () y / o comandos específicos. Configurar un shell chroot, un shell limitado a algún comando específico o un demonio dentro de una cárcel chroot es mucho más fácil y puede automatizarse usando estas utilidades.

¿QUÉ ES FAIL2BAN?

Fail2ban escanea los archivos de registro (por ejemplo, / var / log / apache / error_log) y prohíbe las IP que muestran signos maliciosos: demasiadas fallas de contraseña, búsqueda de vulnerabilidades, etc. En general, Fail2Ban se usa para actualizar las reglas del firewall para rechazar las direcciones IP durante un período de tiempo especificado, aunque también se puede configurar cualquier otra acción arbitraria (por ejemplo, enviar un correo electrónico). Fuera de la caja, Fail2Ban viene con filtros para varios servicios (apache, courier, ssh, etc.).

Fail2Ban es capaz de reducir la tasa de intentos de autenticación incorrectos, sin embargo, no puede eliminar el riesgo que presenta una autenticación débil. Configure los servicios para usar solo dos factores o mecanismos de autenticación públicos / privados si realmente desea proteger los servicios.

¿QUÉ ES FIGLET?

FIGlet es un programa de computadora que genera banners de texto, en una variedad de tipos de letra, compuestos por letras formadas por conglomerados de caracteres ASCII más pequeños. El nombre deriva de "Frank, Ian and Glenn's letters".

Al ser software libre, FIGlet se incluye comúnmente como parte de muchos sistemas operativos de distribución Unix (Linux, BSD, etc.), pero también ha sido portado a otras plataformas.

¿QUÉ ES QUOTA?

Es una característica integrada del kernel de Linux que se usa para establecer un límite de cuánto espacio en disco puede usar un usuario o un grupo. Quota también se usa para limitar la cantidad máxima de archivos que un usuario o grupo puede crear en Linux. Aunque Quota es una característica integrada del núcleo, el sistema de archivos donde desea utilizarla también debe ser compatible ella misma. Algunos de los sistemas de archivos que admiten Quota en Linux son ext2, ext3, ext4, xfs, etc. Quota debe definirse para cada sistema de archivos y para cada usuario o grupo por separado.

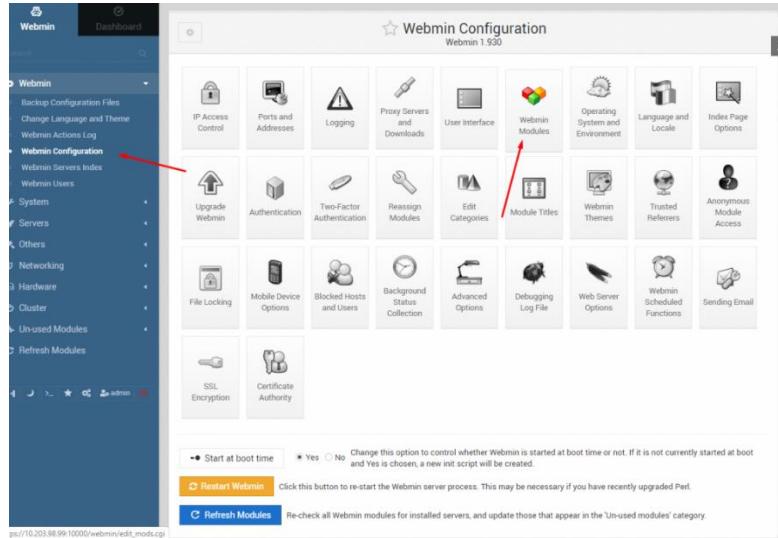
¿LATCH O PESTILLO DIGITAL EN WORDPRESS Y SSH?

Es una herramienta creada por desarrolladores españoles especialistas en seguridad. Sirve para muchas más aplicaciones Web aparte de WordPress. Su funcionalidad es crear un nivel de seguridad adicional en nuestro panel de control. De esta forma, al activar Latch, no podremos acceder a nuestro sitio, ni siquiera nosotros mismos, y para poder acceder, tendremos que desbloquear la seguridad de Latch, lo que nos permitirá acceder solo a nosotros, durante el tiempo que necesitemos trabajar, y después poder volver a bloquearlo y que nadie pueda acceder.

¿PANEL DE ADMINISTRACIÓN WEBMIN?

Webmin es un panel de control con interfaz web que se puede utilizar para administrar servidores. Con él no se necesita tener ningún conocimiento de consola, scripts o archivos de configuración, puesto que el propio panel se encargará de presentar opciones gráficas fáciles de usar y entender.

Evita tener que editar archivos de configuración, ejecutar comandos para crear usuarios, configurar servidores web o añadir a mano redireccionamientos de correo electrónico. Webmin permite hacer todo esto a través de una interfaz web muy fácil de usar y, de forma automática, actualiza todos los archivos de configuración necesarios.



De esta forma, la dificultad de estas tareas queda relegada a un segundo plano y Webmin se encarga de toda la parte técnica, dejando al usuario la toma de decisiones. Así no se necesita investigar los detalles de cómo implementar las opciones que se desean. Evita tambien tener que arreglar los posibles errores causados por fallos tipográficos o de sintaxis que pudiesen ser causados al administrar el servidor por línea de comandos.

Webmin soporta la mayor parte de sistemas operativos basados en Unix, tales como Linux, BSD, Solaris o HP/UX, entre otros. Webmin también puede instalarse en servidores Windows, pero con funcionalidad limitada.

SERVIDOR

Instance details

- Virtual machine name *
- Region *
- Availability options
- Image *
- Azure Spot instance
- Size *
- Administrator account
 - Authentication type
 - Username *
 - SSH public key *
- Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

 - Public inbound ports *
 - Select inbound ports *
 - ⚠️ This will allow all IP addresses to access your virtual machine.** This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Para la creación de la máquina virtual utilizamos los servicios de Microsoft Azure. La imagen utilizada es Ubuntu 18.04. Se utilizó llave SSH para ingresar al servidor. Los puertos de entrada utilizados fueron 23, 80 y 443. Todas las demás opciones fueron el estándar de Azure.

IP ESTÁTICO Y DNS, VIRTUALHOST

Guardar Descartar

Asignación

Dirección IP

Tiempo de espera de inactividad (minutos)

Etiqueta de nombre DNS (opcional)

Conjuntos de registros de alias

¿Quiere realizar un seguimiento exhaustivo de esta dirección IP pública? Cree un registro de alias en Azure DNS. [Más información.](#)

+ Crear un registro de alias

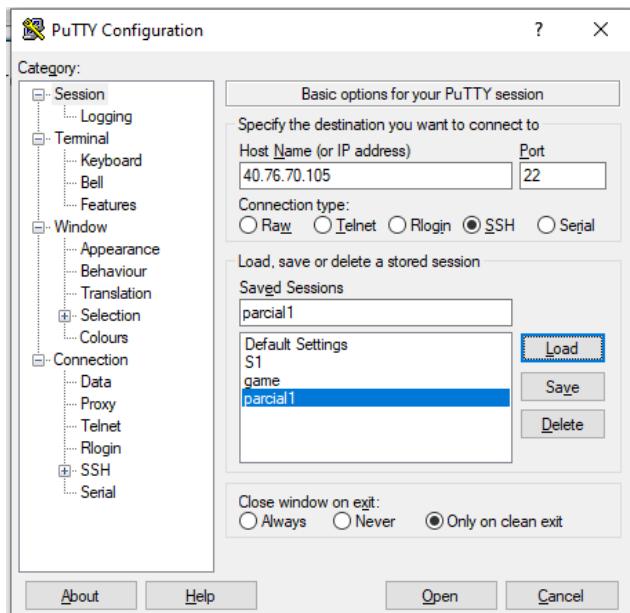
Suscripción	Zona DNS	Nombre	Tipo	TTL
Sin resultados.				

ENTRAR AL SERVIDOR CON PUTTY

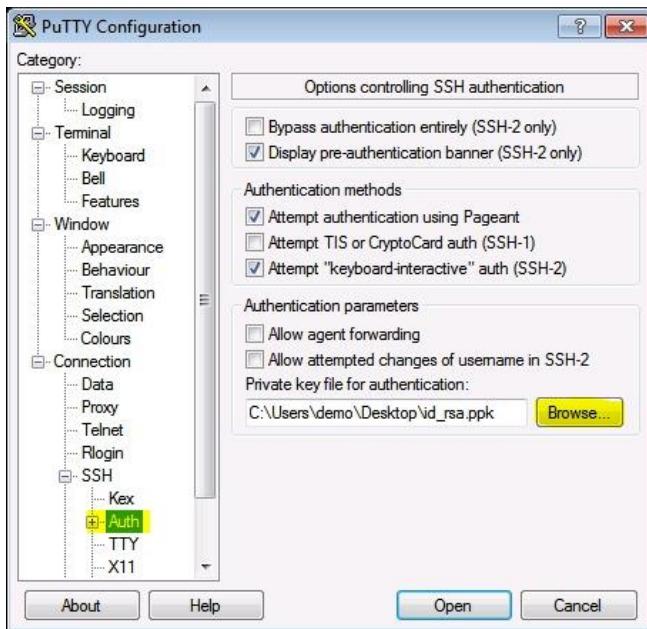
1. En sesión se coloca el ip address dado en azure.

El ip se puede obtener en la página de azure.

2. Se coloca el puerto 22



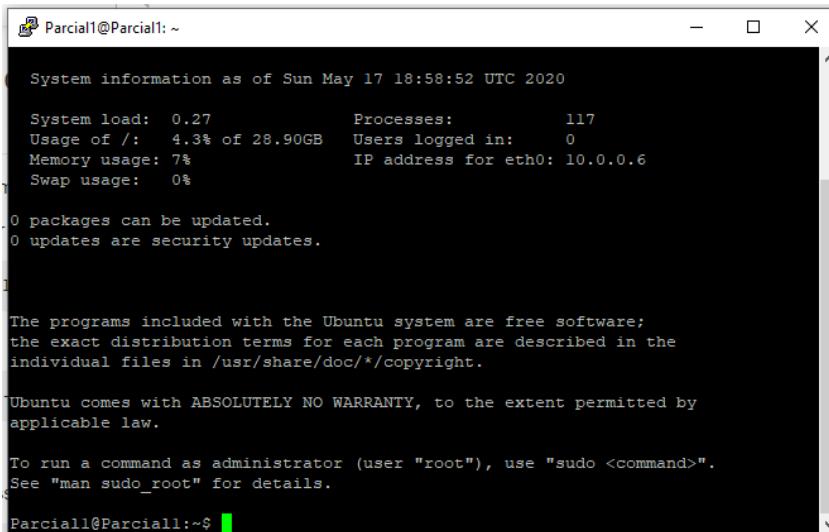
3. en Auth se selección el archivo con el ssh.



4. guardas y abres.

5. haces login

- ?
- Login as: Parcial1B
Phrase for key "parcial1".



Parcial1@Parcial1: ~

```
System information as of Sun May 17 18:58:52 UTC 2020
System load: 0.27      Processes:          117
Usage of /: 4.3% of 28.90GB  Users logged in:    0
Memory usage: 7%
Swap usage:  0%
0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

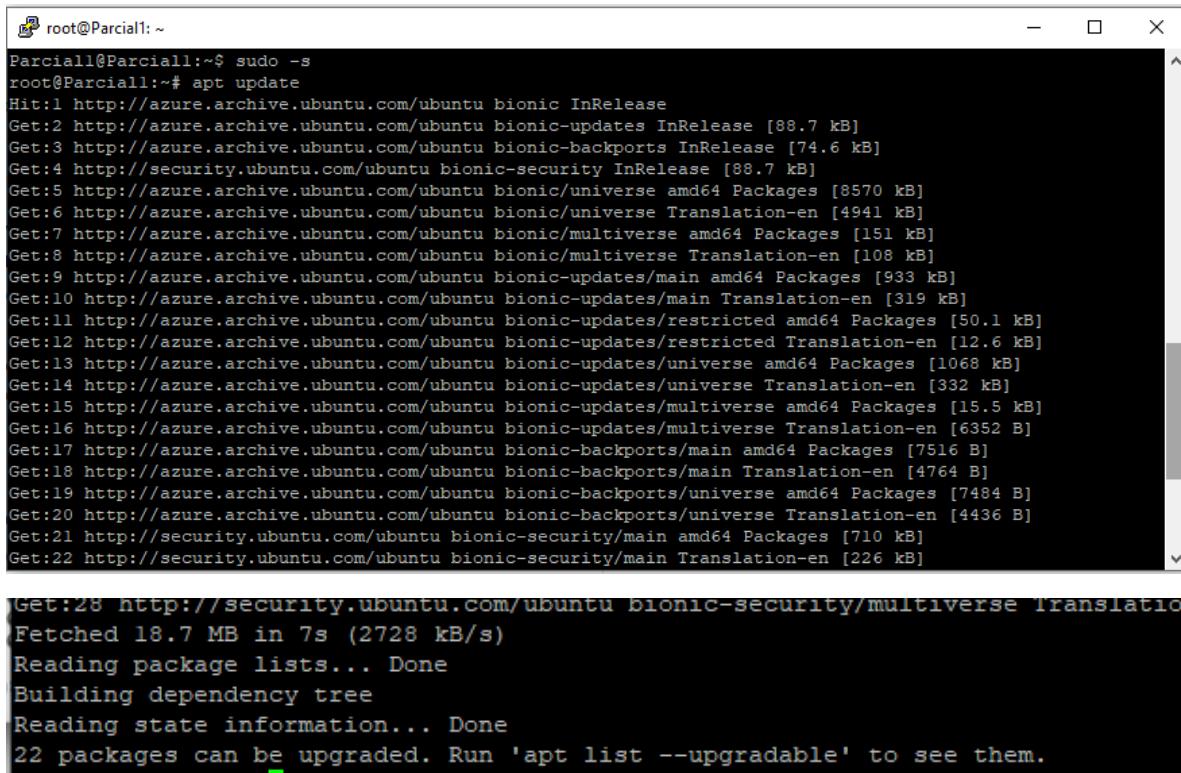
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Parcial1@Parcial1:~$
```

Actualizar servidor

Sudo -s

Apt update.



```
root@Parcial1:~$ sudo -s
root@Parcial1:~# apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [933 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [319 kB]
Get:11 http://azure.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [50.1 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [12.6 kB]
Get:13 http://azure.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1068 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [332 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [15.5 kB]
Get:16 http://azure.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6352 B]
Get:17 http://azure.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [7516 B]
Get:18 http://azure.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [4764 B]
Get:19 http://azure.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [7484 B]
Get:20 http://azure.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [4436 B]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [710 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [226 kB]

Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [18.7 MB]
Fetched 18.7 MB in 7s (2728 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
22 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

INSTALACIÓN DE DOCKER

Actualice el apt índice del paquete e instale paquetes para permitir apt utilizar un repositorio sobre HTTPS:

Ejecutar los siguientes comandos. <https://docs.docker.com/engine/install/ubuntu/>

```
$ sudo apt-get update

$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common
```

```
root@Parcial1:~# sudo apt-get install \
>   apt-transport-https \
>   ca-certificates \
>   curl \
>   gnupg-agent \
>   software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
curl is already the newest version (7.58.0-2ubuntu3.8).
curl set to manually installed.
software-properties-common is already the newest version (0.96.24.32.12).
software-properties-common set to manually installed.
apt-transport-https is already the newest version (1.6.12ubuntu0.1).
The following packages were automatically installed and are no longer required:
```

Agregue la clave GPG oficial de Docker

```
root@Parcial1:~#
root@Parcial1:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
root@Parcial1:~#
```

Verifique que ahora tiene la clave con la huella digital 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88, buscando los últimos 8 caracteres de la huella digital.

```
root@Parcial1:~# sudo apt-key fingerprint 0EBFCD88
pub    rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
uid            [ unknown] Docker Release (CE deb) <docker@docker.com>
sub    rsa4096 2017-02-22 [S]

root@Parcial1:~# z
```

Use el siguiente comando para configurar el repositorio estable.

Para saber la versión el kernel : \$ uname -a

```
sudo add-apt-repository \
```

```
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
```

```
root@Parcial1:~# sudo add-apt-repository \
->     "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
->     $(lsb_release -cs) \
->     stable"
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [11.2 kB]
Fetched 328 kB in 1s (540 kB/s)
Reading package lists... Done
root@Parcial1:~#
```

INSTALAR DOCKER ENGINE

Actualice el aptíndice del paquete e instale la última versión de Docker Engine y del contenedor, o vaya al siguiente paso para instalar una versión específica:

```
root@Parcial1:~# sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 252 kB in 1s (397 kB/s)
Reading package lists... Done
root@Parcial1:~#
```

```
root@Parcial1:~# sudo apt-get install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  grub-pc-bin linux-headers-4.15.0-99
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  aufs-tools cgroupfs-mount libltdl17 pigz
The following NEW packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce docker-ce-cli libltdl17 pigz
0 upgraded, 7 newly installed, 0 to remove and 22 not upgraded.
Need to get 87.0 MB of archives.
After this operation, 393 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://azure.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
```

Verificamos la versión

```
root@Parciall:~# sudo docker -v
Docker version 19.03.8, build afacb8b7f0
```

Corremos el hello world para verificar que la instalacion fue hecha correctamente y todo funciona de manera normal.

```
$ sudo docker run hello-world
```

```
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Ver imágenes del Docker

```
Parciall@Parciall:~$ sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED
SIZE
hello-world      latest   bf756fb1ae65   4 months ago
13.3kB
Parciall@Parciall:~$
```

```
Parciall@Parciall:~$ sudo docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED
STATUS           PORTS      NAMES
16d9d622c4fb    hello-world   "/hello"     4 minutes ago
Exited (0) 4 minutes ago          quizzical_davinci
Parciall@Parciall:~$
```

Eliminar imágenes

\$ Sudo docker rmi image

Image es la imagen a eliminar.

\$sudo docker rmi hellow world

```
Parcial1B@Parcial1B:~$ sudo docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
mysql           8.0          30f937e841c8    2 days ago   541MB
httpd           2.4          d4e60c8eb27a    7 days ago   166MB
ubuntu          latest        1d622ef86b13    4 weeks ago  73.9MB
hello-world     latest        bf756fb1ae65    4 months ago 13.3kB
Parcial1B@Parcial1B:~$ sudo docker rmi hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:6a65f928fb91fcfbc963f7aa6d57c8eeb426ad9a20c7ee045538ef34847f44f1
Deleted: sha256:bf756fb1ae65adf866bd8c456593cd24beb6a0a061def42b26a993176745f6b
Deleted: sha256:9c27e219663c25e0f28493790cc0b88bc973ba3b1686355f221c38a36978ac63
Parcial1B@Parcial1B:~$ sudo docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
mysql           8.0          30f937e841c8    2 days ago   541MB
httpd           2.4          d4e60c8eb27a    7 days ago   166MB
ubuntu          latest        1d622ef86b13    4 weeks ago  73.9MB
```

INSTALACIÓN DE LOS CONTENEDORES

Ahora procedemos a descargar las imágenes que serán nuestros contenedores de la página docker hub luego instalamos las imágenes para que sean nuestras aplicaciones o contenedores. Las imágenes descargadas son como iso los cuales serán instaladas en la maquina física.

También vamos a crear un volumen para que cada contenedor aloje información publica que se pueda acceder a otros contenedores algo así como una carpeta compartida.

El nombre que le pondremos a los contenedores será: mysql, redis, wordpress_1y wordpress_2.

Para que toda la instalación se haga más fácil usaremos docker-compose lo cual es un script **yml** que nos permitiría preparar la ruta del volumen, los puertos, descarga de la imagen, correr los contenedores entre otro... para mas informacion puede visitar este enlace:

<https://devhints.io/docker-compose>

<https://docs.docker.com/compose/>

Usaremos un balanceador de carga llamado nginx proxy reverse

Y para obtener un certificado ssl usaremos un complemento para nginx proxy reverse llamado letsencrypt

Comandos para la descarga

- ❖ \$ sudo docker pull wordpress
- ❖ \$ sudo docker pull mysql
- ❖ \$ sudo docker pull httpd
- ❖ \$ sudo docker pull redis

Para más información aquí se encuentra los enlaces:

https://hub.docker.com/_/wordpress

https://hub.docker.com/_/mysql

https://hub.docker.com/_/httpd

https://hub.docker.com/_/redis

INSTALACIÓN DE DOCKER-COMPOSE

Para instalar docker compose podemos seguir la documentación en el siguiente enlace
<https://docs.docker.com/compose/install/>

```
$sudo apt install docker-compose
```

Comprobamos la versión

```
$ docker-compose --version
```

```
Parcial1B@Parcial1B:~/Wordress$ cd
Parcial1B@Parcial1B:~$ sudo docker-compose -v
docker-compose version 1.17.1, build unknown
Parcial1B@Parcial1B:~$
```

Creamos un directorio

```
Parcial1B@Parcial1B:~$ ls
WordPress
Parcial1B@Parcial1B:~$
```

Creamos nuestro compose

```
Parcial1B@Parcial1B:~$ cd WordPress/
Parcial1B@Parcial1B:~/WordPress$ vi docker-compose.yml
```

El compose que utilizamos fue el siguiente :

version: '3.3'

volumes:

wp-data:

html:

vhostd:

networks:

wp-back:

services:

db:

```
container_name: mysqlDb

image: mysql:5.7

volumes:
  - wp-data:/var/lib/mysql

environment:
  MYSQL_ROOT_PASSWORD: 8L82G6?Vd3jLZ
  MYSQL_DATABASE: wordpress
  MYSQL_USER: wp-user
  MYSQL_PASSWORD: 8L82G6?Vd3jLZ

ports:
  - 8889:3306

networks:
  - wp-back
```

```
nginx-proxy:
  container_name: nginx_proxy

  depends_on:
    - app1
    - app2

  restart: always

  image: jwilder/nginx-proxy
```

```
  ports:
    - 80:80
    - 443:443
```

```
  volumes:
```

```
- /var/run/docker.sock:/tmp/docker.sock:ro  
- ./certificates:/etc/nginx/certs:rw  
- vhostd:/etc/nginx/vhost.d  
- html:/usr/share/nginx/html
```

labels:

```
- com.github.jrcs.letsencrypt_nginx_proxy_companion.nginx_proxy
```

networks:

```
- wp-back
```

letsencrypt:

```
image: jrcs/letsencrypt-nginx-proxy-companion
```

```
restart: always
```

environment:

```
- NGINX_PROXY_CONTAINER=nginx-proxy
```

volumes:

```
- ./certificates:/etc/nginx/certs:rw  
- vhostd:/etc/nginx/vhost.d  
- html:/usr/share/nginx/html  
- /var/run/docker.sock:/var/run/docker.sock:ro
```

redis:

```
image: redis:latest
```

```
container_name: redis
```

```
restart: always
```

ports:

- "6379:6379"

networks:

- wp-back

app1:

container_name: wordpress_1

depends_on:

- db

- redis

image: wordpress:latest

expose:

- 80

- 443

environment:

WORDPRESS_DB_HOST: db

WORDPRESS_DB_USER: wp-user

WORDPRESS_DB_PASSWORD: 8L82G6?Vd3jLZ

VIRTUAL_HOST: parcial1.eastus.cloudapp.azure.com

LETSENCRYPT_HOST: parcial1.eastus.cloudapp.azure.com

LETSENCRYPT_EMAIL: christianespinoza@gmail.com

volumes:

- ./wordpress-files:/var/www/html

networks:

- wp-back

app2:

container_name: wordpress_2

depends_on:

- db

- redis

image: wordpress:latest

expose:

- 80

- 443

environment:

WORDPRESS_DB_HOST: db

WORDPRESS_DB_USER: wp-user

WORDPRESS_DB_PASSWORD: 8L82G6?Vd3jLZ

VIRTUAL_HOST: parcial1.eastus.cloudapp.azure.com

VIRTUAL_HOST: parcial1.eastus.cloudapp.azure.com

LETSENCRYPT_HOST: parcial1.eastus.cloudapp.azure.com

LETSENCRYPT_EMAIL: christianespinoza@gmail.com

volumes:

- ./wordpress-files:/var/www/html

networks:

- wp-back

Copiando esto y luego presionando "esc" + ":wq!" Se guardara.

Luego levantamos el compose con

```
$ Sudo docker-compose -d up
```

Algunos otros comandos de docker compose:

- ② docker-compose start
- ② docker-compose stop
- ② docker-compose pause
- ② docker-compose unpause
- ② docker-compose ps
- ② docker-compose up -d
- docker-compose down

Documentacion por parte del compose

El compose que se muestra a continuacion para verlo de una manera mas sencilla esta representado en la siguiente imagen como ejemplo. En vez de usar links usamos una red.

```
version: '3.2'
services:
  backend1:
    build: ./backend
    tty: true
    volumes:
      - './backend/src:/backend-dir-inside-container'

  backend2:
    build: ./backend
    tty: true
    volumes:
      - './backend/src:/backend-dir-inside-container'

  backend3:
    build: ./backend
    tty: true
    volumes:
      - './backend/src:/backend-dir-inside-container'

  loadbalancer:
    build: ./load-balancer
    tty: true
    links:
      - backend1
      - backend2
      - backend3
    ports:
      - '8080:8080'

volumes:
  backend:
```

Ilustración 1 Imagen de ejemplo

DECLARACION

version: '3.3' #versión del compose a utilizar

volumes: #nombre del volumen

wp-data:

html:

vhostd:

networks: #nombre de la red

wp-back:

NGINX-PROXY

services:

nginx-proxy: #nombre del servicio

container_name: nginx_proxy #Nombre del contendor

depends_on: #dependencias de otros servicios

- app1

- app2

restart: always

image: jwilder/nginx-proxy # imagen que usara para la instalacion del contenedor

ports: #puerto a utilizar por ngínx

- 80:80

- 443:443

volumes: #volumenes donde se guardará la información

- /var/run/docker.sock:/tmp/docker.sock:ro

- ./certificates:/etc/nginx/certs:rw

- vhostd:/etc/nginx/vhost.d

- html:/usr/share/nginx/html

```
labels: #label de requisito para instalar el ssl con letsencrypt
```

```
- com.github.jrcs.letsencrypt_nginx_proxy_companion.nginx_proxy
```

```
networks: #red para la comunicacion entre contenedores
```

```
- wp-back
```

Para mas informacion puede visitar la documentacion:

LETSENCRYPT

services:

```
letsencrypt: #nombre del servicio
```

```
image: jrcs/letsencrypt-nginx-proxy-companion # imagen que usara para la instalacion del contenedor
```

```
restart: always
```

```
environment: #variable de entorno
```

```
- NGINX_PROXY_CONTAINER=nginx-proxy
```

```
volumes: #volumenes donde se guardara la informacion
```

```
- ./certificates:/etc/nginx/certs:rw
```

```
- vhostd:/etc/nginx/vhost.d
```

```
- html:/usr/share/nginx/html
```

```
- /var/run/docker.sock:/var/run/docker.sock:ro
```

Para mas informacion puede visitar la documentacion:

MYSQLDB

services:

```
db: #nombre del servicio
```

```
container_name: mysql #nombre del contenedor
```

```
image: mysql:5.7 # imagen que usara para la instalacion del contenedor
```

volumes: #volumenes donde se guardará la información

- wp-data:/var/lib/mysql

environment: #variables de entorno

MYSQL_ROOT_PASSWORD: 8L82G6?Vd3jLZ

MYSQL_DATABASE: wordpress

MYSQL_USER: wp-user

MYSQL_PASSWORD: 8L82G6?Vd3jLZ

ports: #puerto a utilizar por mysql

- 8889:3306

networks: #red para la comunicación entre contenedores

- wp-back

WORDPRESS_1 Y WORDPRESS_2

esta configuración es aplicada para la segunda aplicación.

services:

app1: #nombre del servicio

container_name: wordpress_1 #nombre del contenedor

depends_on: #dependencias de otros servicios

- db

- redis

image: wordpress:latest # imagen que usara para la instalación del contenedor

expose: #informa a Docker que el contenedor escucha en los puertos de red especificados en tiempo de ejecución.

- 80

- 443

environment: #variable de entorno

WORDPRESS_DB_HOST: db

WORDPRESS_DB_USER: wp-user

WORDPRESS_DB_PASSWORD: 8L82G6?Vd3jLZ

VIRTUAL_HOST: parcial1.eastus.cloudapp.azure.com

LETSENCRYPT_HOST:

LETSENCRYPT_EMAIL: christianespinoza@gmail.com

volumes: #volumenes donde se guardará la informacion

- ./wordpress-files:/var/www/html

networks: #red para la comunicación entre contenedores

- wp-back

REDIS

services:

```
redis: #nombre del servicio
```

```
image: redis:latest # imagen que usara para la instalacion del contenedor
```

```
container_name: redis # nombre del contenedor
```

```
restart: always
```

```
ports: #puerto a utilizar por redis
```

```
- "6379:6379"
```

```
networks: #red para la comunicacion entre contenedores
```

```
- wp-back
```

CONFIGURACION DE REDIS

Ahora procedemos a realizar una serie de paso para que redis trabaje junto con wordpress. Los siguientes paso deben hacer en la ruta del volumen del wordpress.

Primero debemos configurar el archivo wp-config.php ubicado en la siguiente ruta

```
Parcial1B@Parcial1B:~/WordPress$ cd wordpress-files/
Parcial1B@Parcial1B:~/WordPress/wordpress-files$ ls
index.php      readme.html      wp-admin          wp-comments-post.php  wp-config.php
license.txt    wp-activate.php  wp-blog-header.php  wp-config-sample.php  wp-content
Parcial1B@Parcial1B:~/WordPress/wordpress-files$
```

Luego abrimos

```
# sudo nano wp-config.php
```

```
Parcial1B@Parcial1B:~/WordPress/wordpress-files$ nano wp-config.php
Parcial1B@Parcial1B:~/WordPress/wordpress-files$
```

pegamos lo siguiente

```
define('WP_REDIS_HOST', 'redis');
define('WP_CACHE_KEY_SALT', 'wp-docker-5DknyYepdjjyJMo8gDqrLhrpAJUQ');

define('WP_REDIS_HOST', 'redis');
define('WP_CACHE_KEY_SALT', 'wp-docker-5DknyYepdjjyJMo8gDqrLhrpAJUQ');
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress');
```

presionamos ^x + y +enter para guardar y salir.

CONFIGURACION DEL WORDPRESS

Primero entramos al sitio con el dns que colocamos o podemos hacerlo con la dirección ip.

Conectar Iniciar Reiniciar Detener Captura Eliminar Actualizar

Advisor (1 de 2): Habilitación de la replicación de la máquina virtual para proteger las aplicaciones frente a una interrupción regional →

Grupo de recur... (cambiar) : Parcial

Estado : En ejecución

Ubicación : Este de EE. UU.

Suscripción (cambiar) : Azuré for Students

Id. de suscripción : 2a794b04-6c24-495d-b593-6b0f02919eb5

Nombre del equipo : Parcial1B

Sistema operativo : Linux (ubuntu 18.04)

Tamaño : B2s estándar (2 vcpu, 4 GiB de memoria)

Etiquetas (cambiar) : Haga clic aquí para agregar etiquetas.

Azure Spot : N/D

Dirección IP pública : 40.114.1.219

Dirección IP privada : 10.0.0.7

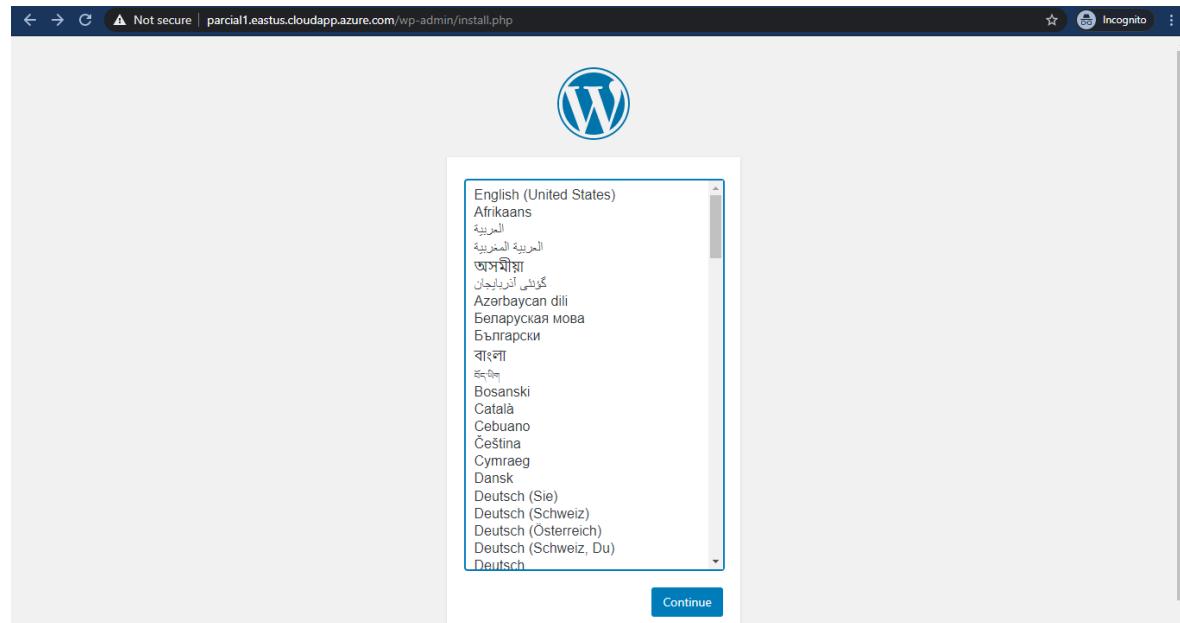
Dirección IP pública (IPv6) : -

Dirección IP privada (IPv6) : -

Red virtual/subred : Parcial-vnet/default

Nombre DNS : parcial1.eastus.cloudapp.azure.com

Nos saldra una pagina



Seleccionamos el idioma e introducimos la informacion para el registro.

The screenshot shows a two-step configuration process for a website.

Step 1: Language Selection

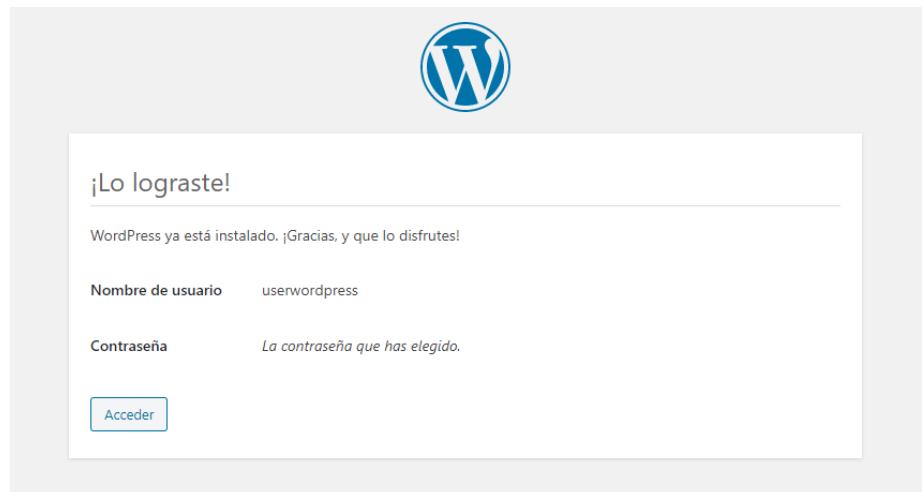
A dropdown menu lists various languages:

- Deutsch (Sie)
- Deutsch (Schweiz)
- Deutsch (Österreich)
- Deutsch (Schweiz, Du)
- Deutsch
- Ελληνικά
- English (New Zealand)
- English (UK)
- English (South Africa)
- English (Australia)
- English (Canada)
- Esperanto
- Español de Venezuela
- Español** (highlighted in blue)
- Español de Colombia
- Español de Perú
- Español de México
- Español de Costa Rica
- Español de Uruguay
- Español de Guatemala
- Español de Puerto Rico

Step 2: Site Information

The right panel contains fields for entering site details:

- Título del sitio:**
- Nombre de usuario:**
A note below states: "Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @."
- Contraseña:**
A note below states: "Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro."
- Tu correo electrónico:**
A note below states: "Comprueba bien tu dirección de correo electrónico antes de continuar."
- Visibilidad en los motores de búsqueda:** "Disuadir a los motores de búsqueda de indexar este sitio"
A note below states: "Depende de los motores de búsqueda atender esta petición o no."



Accedemos y iniciamos sesión

1. Username: Userwordpress
2. Password: 8L82G6?Vd3jLZ

A screenshot of the WordPress login page. At the top is the blue 'W' logo. Below it is a form with two input fields: 'Nombre de usuario o correo electrónico' containing 'userwordpress' and 'Contraseña' containing a series of dots. To the right of the password field is an 'eye' icon for password visibility. Below the inputs are two buttons: a checkbox labeled 'Recuérdame' and a blue 'Acceder' button. At the bottom of the form area, there is a link '¿Has olvidado tu contraseña?' and a link '← Volver a Parcial1'.

Clicamos en plugins

Luego en el botón de añadir nuevo

Ahora buscamos el plugins de redis object cache

The screenshot shows the WordPress plugin search interface. The search term 'redis' is entered in the search bar. Below the search bar, there are tabs for 'Resultados de la búsqueda', 'Destacados', 'Populares', 'Recomendados', and 'Favoritos'. The search results page shows two plugins: 'Redis Object Cache' and 'LiteSpeed Cache'. Both plugins have a rating of 5 stars. 'Redis Object Cache' has over 60,000 installations and was last updated a week ago. 'LiteSpeed Cache' has over 1 million installations and was also last updated a week ago. Both are compatible with the current version of WordPress.

Activamos

The screenshot shows the 'Redis Object Cache' settings page. A checkbox labeled 'Redis Object Cache' is checked, indicating it is active. Below the checkbox, there are links for 'Activar' (Activate), 'Borrar' (Delete), 'Versión 1.6.3', and 'Por Till Krüss | Ver detalles' (Version 1.6.3 | By Till Krüss | View details).

configuramos

The screenshot shows the 'Redis Object Cache' settings page. A section titled 'Redis Object Cache' is expanded, showing a 'Settings' link and a 'Desactivar' (Deactivate) button.

Luego clicamos en activar la cache de objetos

The screenshot shows the 'Redis Object Cache' settings page. The status is 'Caché de objetos desactivada.' (Object cache deactivated). The 'Estado' (Status) is 'Desactivado' (Deactivated). The 'Prefijo de clave:' (Prefix key:) field contains 'wp-docker-5DknvYepdjhjMo8gDqrLhrpAJUQ'. A blue button 'Activar la caché de objetos' (Activate object cache) is visible. To the right, a sidebar for 'Redis Cache Pro' is shown, describing it as a business class object cache backend. It lists features like raw performance, API compliance, and WooCommerce optimization. A 'Learn more' button is at the bottom of the sidebar. At the bottom of the main page, there is a 'Servidores' (Servers) table with two rows:

Alias	Protocol	Host	Port	Database	Password
Master	TCP	redis	6379	0	No
Alias	Protocol	Host	Port	Database	Password

Luego saldra el estado en conectado.

The screenshot shows a Redis configuration interface with the following details:

- Estado:** Conectado
- Cliente:** Predis (v1.1.1)
- Prefijo de clave:** wp-docker-5Dk...AJUQ

At the bottom, there are two buttons: "Vaciar Caché" (Empty Cache) and "Desactivar la caché de objetos" (Disable object cache).

Para mayor informacion:

<https://labs.bilimedtech.com/cloud-computing/3/3.4.html>

<https://labs.bilimedtech.com/cloud-computing/3/3.5.html>

Tambien instalamos el wordfence

The screenshot shows the WordPress plugin search results for "wordfence". The search bar contains "wordfence". The results page displays the following items:

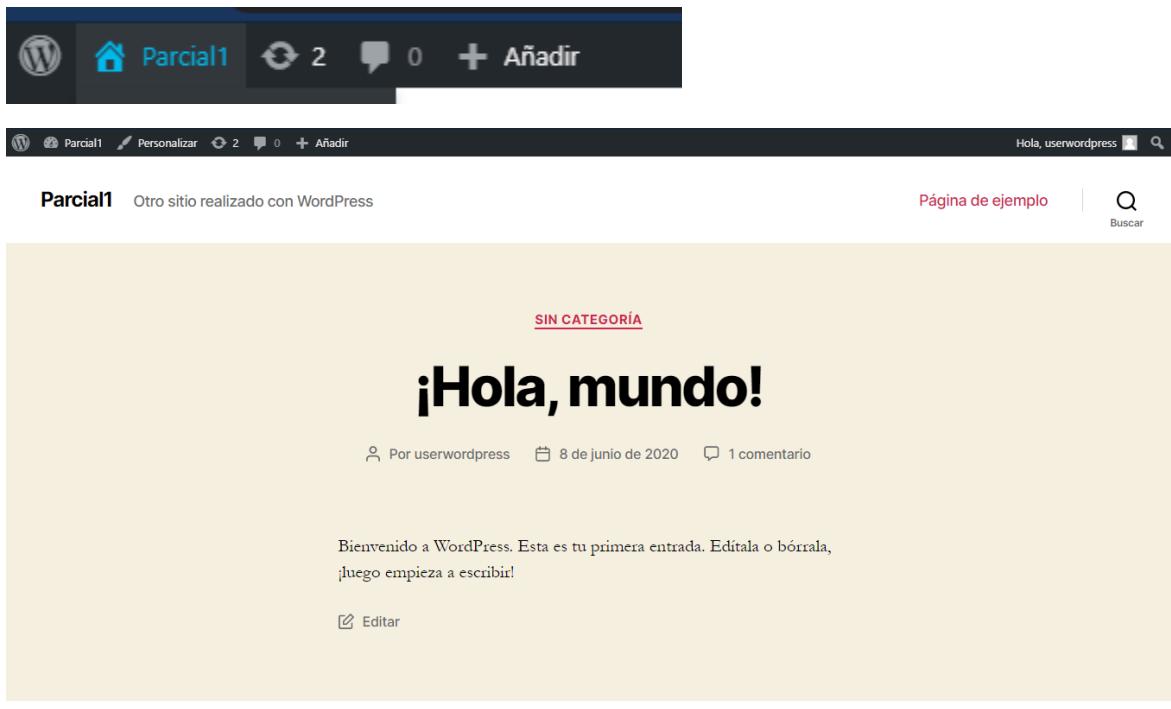
- Wordfence Security – Firewall & Malware Scan** by Wordfence
 - Thumbnail: Wordfence logo
 - Description: Asegura tu web con el plugin de seguridad WordPress más completo. Cortafuegos, exploración de malware, bloques, tráfico en directo, seguridad de acceso y más.
 - Rating: ★★★★★ (3.550)
 - Instalar ahora (Install now) button
 - Más detalles (More details) link
 - Última actualización: hace 2 meses
 - Compatible con tu versión de WordPress
 - 3+ millones instalaciones activas
- Wordfence Login Security** by Wordfence
 - Thumbnail: Lock icon
 - Description: Secure your website with Wordfence Login Security, providing two-factor authentication, login and registration CAPTCHA, and XML-RPC protection.
 - Rating: ★★★★★ (5)
 - Instalar ahora (Install now) button
 - Más detalles (More details) link
 - Última actualización: hace 2 meses
 - Compatible con tu versión de WordPress
- Wordfence Assistant** by Wordfence
 - Thumbnail: Gear icon
 - Description: Wordfence Assistant provides data management utilities for Wordfence users.
 - Rating: ★★★★★ (3.550)
 - Instalar ahora (Install now) button
 - Más detalles (More details) link
 - Última actualización: hace 2 meses
 - Compatible con tu versión de WordPress
- OMGF | Host Google Fonts Locally**
 - Thumbnail: OMGF logo
 - Description: Con solo 2 clics de un botón, OMGF descarga automáticamente en la carpeta de contenido de WordPress las fuentes de Google que quieras, genera una hoja de estilos para ...
 - Instalar ahora (Install now) button
 - Más detalles (More details) link

Activamos

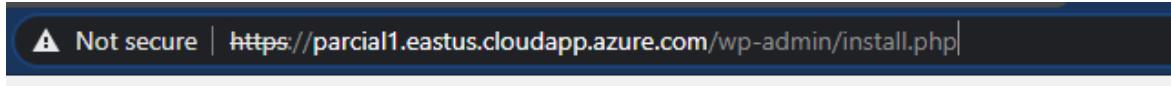
The screenshot shows the activation screen for the Wordfence Security plugin. It includes the following elements:

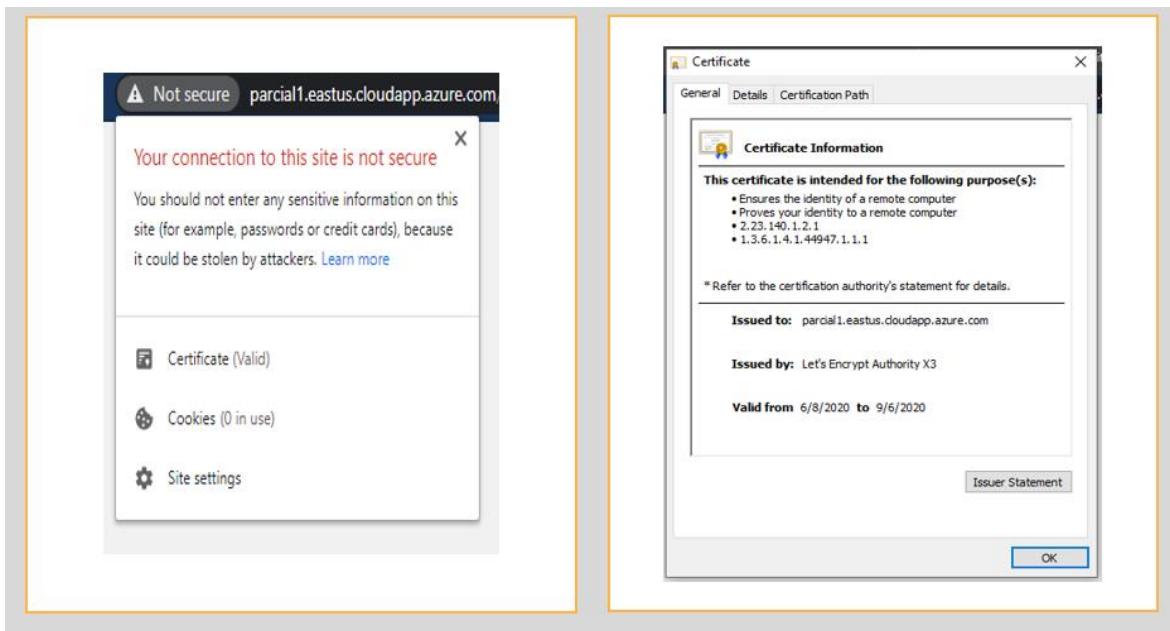
- A checkbox labeled "Wordfence Security".
- A "Activar" (Activate) button in blue.
- A "Borrar" (Delete) button in red.
- A "Ver" (View) button in green.

Procedemos a diseñar la pagina y para eso clicamos en parcial 1



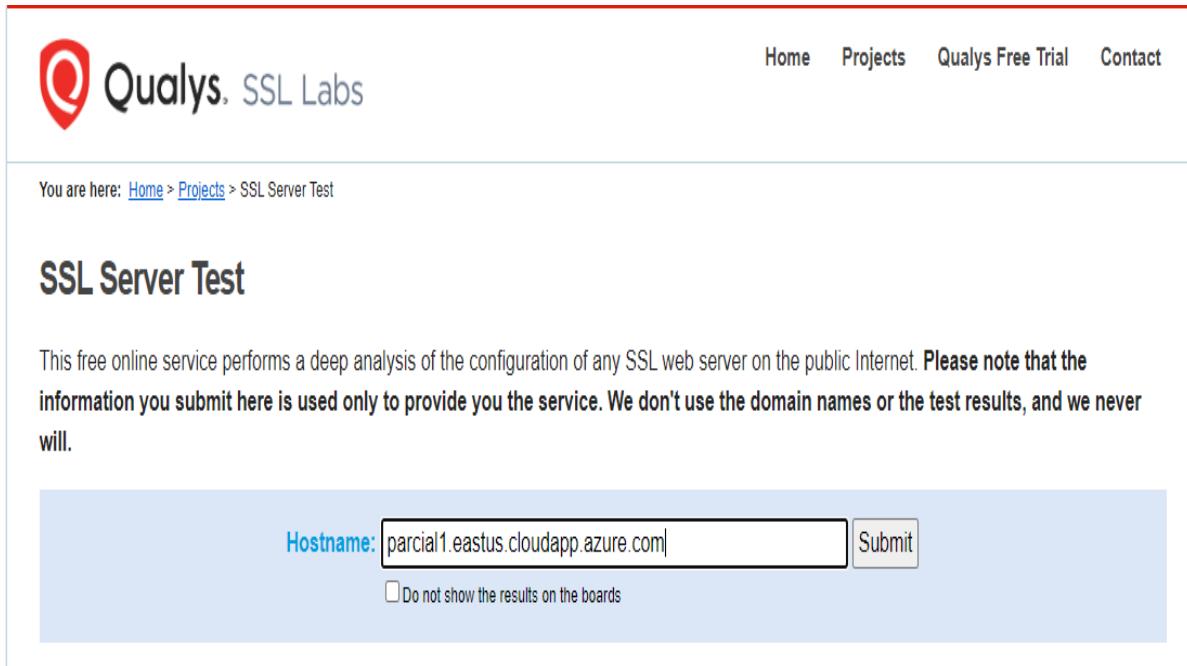
Verificamos si esta respondiendo por https y tiene instalado el certificado ssl





En esta pagina podemos verificar el certificado con una prueba.

<https://www.ssllabs.com/index.html>



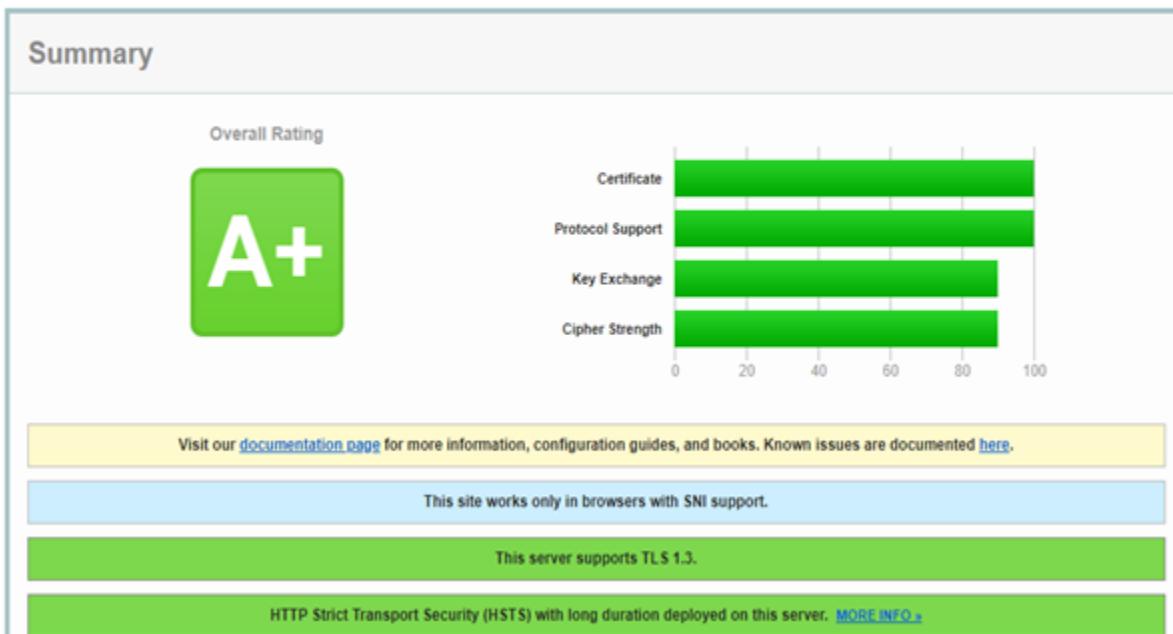
You are here: [Home](#) > [Projects](#) > SSL Server Test

SSL Server Test

This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet. Please note that the information you submit here is used only to provide you the service. We don't use the domain names or the test results, and we never will.

Hostname:

Do not show the results on the boards

SL Report: parcial1.eastus.cloudapp.azure.com (40.114.1.219)Scanned on: Mon, 08 Jun 2020 17:36:59 UTC | [Clear cache](#)[Scan Another](#)

PRUEBA DE PERCISTENCIA DE DATOS EN LOS CONTENEDORES.

Nos dirigimos al terminal y verificamos los contenedores.

```
Parcial1B@Parcial1B:~/WordPress$ sudo docker-compose ps
          Name        Command     State            Ports
-----  -----
mysql_db      docker-entrypoint.sh mysqld    Up      0.0.0.0:8889->3306/tcp, 33060/tcp
nginx_proxy   /app/docker-entrypoint.sh ...  Up      0.0.0.0:443->443/tcp, 0.0.0.0:80->80/tcp
redis         docker-entrypoint.sh redis ...  Up      0.0.0.0:6379->6379/tcp
wordpress_1   docker-entrypoint.sh apach ...  Up      443/tcp, 80/tcp
wordpress_2   docker-entrypoint.sh apach ...  Up      443/tcp, 80/tcp
wordpress_letsencrypt_1 /bin/bash /app/entrypoint. ...  Up
Parcial1B@Parcial1B:~/WordPress$
```

```
Parcial1B@Parcial1B:~/WordPress$ sudo docker container stop wordpress_1
wordpress_1
Parcial1B@Parcial1B:~/WordPress$
```

```
Parcial1B@Parcial1B:~/WordPress$ sudo docker-compose ps
          Name        Command     State            Ports
-----  -----
mysql_db      docker-entrypoint.sh mysqld    Up      0.0.0.0:8889->3306/tcp, 33060/tcp
nginx_proxy   /app/docker-entrypoint.sh ...  Up      0.0.0.0:443->443/tcp, 0.0.0.0:80->80/tcp
redis         docker-entrypoint.sh redis ...  Up      0.0.0.0:6379->6379/tcp
wordpress_1   docker-entrypoint.sh apach ...  Exit 0
wordpress_2   docker-entrypoint.sh apach ...  Up      443/tcp, 80/tcp
wordpress_letsencrypt_1 /bin/bash /app/entrypoint. ...  Up
Parcial1B@Parcial1B:~/WordPress$
```

Dejamos un mensaje en la aplicación de wordpress_2 para verificar si se ve en el wordpress_1

The screenshot shows the WordPress dashboard with the following content:

- Primeros pasos** (First steps):
 - Personaliza tu sitio** (Customize your site) - This button is highlighted in blue.
 - [o cambia tu tema por completo](#) (Change your theme completely)
- Siguientes pasos** (Next steps):
 - Escribe tu primera entrada en el blog** (Write your first blog post)
 - Añade una página «Acerca de»** (Add a «About» page)
 - Establece tu página de inicio** (Set up your homepage)
 - Ver tu sitio** (View your site)
- Más acciones** (More actions):
 - Gestionar widgets**
 - Gestionar menús**
 - Activa o desactiva los comentarios**
 - Aprende más sobre cómo empezar**

The screenshot shows the WordPress editor interface with the following content:

Añadir el título (Add the title)

Empieza a escribir o escribe « / » para elegir un bloque ⊕



Detenemos los 2

```
Parcial1B@Parcial1B:~/WordPress$ sudo docker-compose ps
          Name           Command   State        Ports
----- 
mysql_db      docker-entrypoint.sh mysqld    Up      0.0.0.0:8889->3306/tcp, 33060/tcp
nginx_proxy   /app/docker-entrypoint.sh ...   Up      0.0.0.0:443->443/tcp, 0.0.0.0:80->80/tcp
redis         docker-entrypoint.sh redis ... Up      0.0.0.0:6379->6379/tcp
wordpress_1   docker-entrypoint.sh apach ... Exit 0
wordpress_2   docker-entrypoint.sh apach ... Exit 0
wordpress_letsencrypt_1 /bin/bash /app/entrypoint. ... Up
Parcial1B@Parcial1B:~/WordPress$
```

Refrescamos la pagina



Ahora empezamos cualquier contenedor

```
Parcial1B@Parcial1B:~/WordPress$ sudo docker container start wordpress_1
wordpress_1
Parcial1B@Parcial1B:~/WordPress$ sudo docker-compose ps
          Name           Command   State        Ports
----- 
mysql_db      docker-entrypoint.sh mysqld    Up      0.0.0.0:8889->3306/tcp, 33060/tcp
nginx_proxy   /app/docker-entrypoint.sh ...   Up      0.0.0.0:443->443/tcp, 0.0.0.0:80->80/tcp
redis         docker-entrypoint.sh redis ... Up      0.0.0.0:6379->6379/tcp
wordpress_1   docker-entrypoint.sh apach ... Up      443/tcp, 80/tcp
wordpress_2   docker-entrypoint.sh apach ... Exit 0
wordpress_letsencrypt_1 /bin/bash /app/entrypoint. ... Up
Parcial1B@Parcial1B:~/WordPress$
```

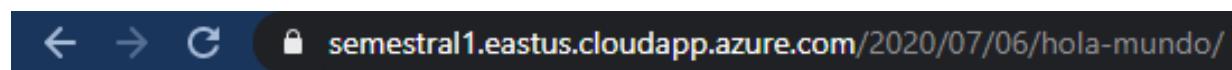


Luego comenzamos el wordpress_1 y detenemos el wordpress_2

```
Parcial1B@Parcial1B:~/WordPress$ sudo docker container start wordpress_1
wordpress_1
Parcial1B@Parcial1B:~/WordPress$ sudo docker container stop wordpress_2
wordpress_2
Parcial1B@Parcial1B:~/WordPress$
```

```
Parcial1B@Parcial1B:~/WordPress$ sudo docker-compose ps
          Name           Command     State            Ports
----- 
mysql_db      docker-entrypoint.sh mysqld   Up    0.0.0.0:8889->3306/tcp, 33060/tcp
nginx_proxy    /app/docker-entrypoint.sh ...  Up    0.0.0.0:443->443/tcp, 0.0.0.0:80->80/tcp
redis          docker-entrypoint.sh redis ... Up    0.0.0.0:6379->6379/tcp
wordpress_1    docker-entrypoint.sh apache ... Up    443/tcp, 80/tcp
wordpress_2    docker-entrypoint.sh apache ... Exit 0
wordpress_letsencrypt_1 /bin/bash /app/entrypoint. ... Up
Parcial1B@Parcial1B:~/WordPress$
```

Veremos que los datos prevalecen.



El candado del certificado aparecerá unos minutos después.

NOTA

La siguiente configuracion se hará con un dns diferente debido a que el dns lo hemos cambiado por semestral1.eastus.cloudapp.azure.com

WAF (MOD SECURITY)

Instalamos el waf web application Firewall

Hay 2 tipos de WAF: Los que se residen en la red (es decir son un elemento más de la red) y los que se basan en el servidor de aplicaciones (residen en el servidor).

Los WAF son elementos complementarios a las medidas de seguridad que soportan los Firewall clásicos.

Estará formado por tres mod (mod security, mod evasive, mod qos).

INSTALACION DE WAF CON CONFIGURACIÓN DE MOD_SECURITY

Actualizamos

```
# apt-get update
```

```
root@71c7031e2c6d:/var/www/html# apt-get update
Get:1 http://security.debian.org/debian-security buster/updates InRelease [65.4
kB]
Get:2 http://deb.debian.org/debian buster InRelease [121 kB]
Get:3 http://deb.debian.org/debian buster-updates InRelease [51.9 kB]
Get:4 http://security.debian.org/debian-security buster/updates/main amd64 Packa
ges [208 kB]
Get:5 http://deb.debian.org/debian buster/main amd64 Packages [7905 kB]
Get:6 http://deb.debian.org/debian buster-updates/main amd64 Packages [7868 B]
Fetched 8360 kB in 2s (3898 kB/s)
Reading package lists... Done
root@71c7031e2c6d:/var/www/html#
```

Instalamos la librería de libapache2-mod-security2

```
# apt-get install libapache2-mod-security2
```

Luego aceptamos con y

```
root@71c7031e2c6d:/var/www/html# apt-get install libapache2-mod-security2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcurl3-gnutls liblua5.1-0 libyajl2 modsecurity-crs
Suggested packages:
  lua geoip-database-contrib ruby python
The following NEW packages will be installed:
  libapache2-mod-security2 libcurl3-gnutls liblua5.1-0 libyajl2
  modsecurity-crs
0 upgraded, 5 newly installed, 0 to remove and 4 not upgraded.
Need to get 920 kB of archives.
After this operation, 4865 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

apt-cache show libapache2-mod-security2

```
root@c29ba9ebced8:/var/www/html# apt-cache show libapache2-mod-security2
Package: libapache2-mod-security2
Source: modsecurity-apache
Version: 2.9.3-1
Installed-Size: 1071
Maintainer: Alberto Gonzalez Iniesta <agi@inittab.org>
Architecture: amd64
Depends: libxml2 (>= 2.9.0), libapr1 (>= 1.2.7), libaprutil1 (>= 1.4.0), libc6 (
```

Verificamos # apachectl -M | grep security

```
root@951843f88c3b:/var/www/html# apachectl -M | grep security
AH00558: apache2: Could not reliably determine the server's fully qualified
           security_module (shared)
root@951843f88c3b:/var/www/html#
```

Movemos y cambiamos el nombre de la carpeta por defecto de ModSecurity

Reiniciamos

```
# /etc/init.d/apache2 reload
```

```
# rm -rf /usr/share/modsecurity-crs
```

```
root@c29ba9ebced8:/var/www/html# rm -rf /usr/share/modsecurity-crs
root@c29ba9ebced8:/var/www/html#
```

Instalamos git

```
# apt install git
```

```
root@c29ba9ebced8:/var/www/html# apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man less liberror-perl libpcre2-8-0 libxml2 openssh-client xauth
Suggested packages:
  gettext-base git-daemon-run | git-daemon-sysvinit git-doc git-el git-email
  git-gui gitk gitweb git-cvs git-mediawiki git-svn keychain libpam-ssh
  monkeysphere ssh-askpass
The following NEW packages will be installed:
  git git-man less liberror-perl libpcre2-8-0 libxml2 openssh-client xauth
0 upgraded, 8 newly installed, 0 to remove and 4 not upgraded.
Need to get 8472 kB of archives.
After this operation, 42.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian buster/main amd64 less amd64 487-0.1+bd1 [129 kB]
Get:2 http://deb.debian.org/debian buster/main amd64 openssh-client amd64 1:7.9p1-10+deb10u2 [782 kB]
Get:3 http://deb.debian.org/debian buster/main amd64 libpcre2-8-0 amd64 10.32-5
```

descargamos la última versión de mod_security CRS con el siguiente comando

```
# git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git
/usr/share/modsecurity-crs
```

```
root@c29ba9ebced8:/var/www/html# git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git /usr/share/modsecurity-crs
Cloning into '/usr/share/modsecurity-crs'...
remote: Enumerating objects: 10486, done.
remote: Total 10486 (delta 0), reused 0 (delta 0), pack-reused 10486
Receiving objects: 100% (10486/10486), 3.33 MiB | 24.52 MiB/s, done.
Resolving deltas: 100% (7687/7687), done.
```

Ejecutamos los siguientes comandos

```
# cd /usr/share/modsecurity-crs
# mv crs-setup.conf.example crs-setup.conf
# mv /etc/modsecurity/modsecurity.conf-recommended
/etc/modsecurity/modsecurity.conf
```

Editamos modsecurity.conf con el siguiente comando

```
# nano /etc/modsecurity/modsecurity.conf
```

```
root@c29ba9ebced8:/var/www/html# nano /etc/modsecurity/modsecurity.conf
root@c29ba9ebced8:/var/www/html#
```

Luego documentamos

```
#SecRuleEngine DetectionOnly
```

Agregamos

SecRuleEngine On

```
GNU nano 3.2          /etc/modsecurity/modsecurity.conf

# -- Rule engine initialization ----

# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
#SecRuleEngine DetectionOnly
SecRuleEngine On

# -- Request body handling ----

# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
#SecRequestBodyAccess On
```

Luego editamo security2.conf

```
# nano /etc/apache2/mods-enabled/security2.conf
```

Agregamos:

```
IncludeOptional "/usr/share/modsecurity-crs/*.conf"
```

```
IncludeOptional "/usr/share/modsecurity-crs/rules/*.conf"
```

```
semestral@semestral: ~/semestral
GNU nano 3.2          /etc/apache2/mods-enabled/security2.conf

<IfModule security2_module>
    # Default Debian dir for modsecurity's persistent data
    SecDataDir /var/cache/modsecurity

    # Include all the *.conf files in /etc/modsecurity.
    # Keeping your local configuration in that directory
    # will allow for an easy upgrade of THIS file and
    # make your life easier
    IncludeOptional /etc/modsecurity/*.conf

    # Include OWASP ModSecurity CRS rules if installed
    IncludeOptional /usr/share/modsecurity-crs/*.load
    IncludeOptional "/usr/share/modsecurity-crs/*.conf"
    IncludeOptional "/usr/share/modsecurity-crs/rules/*.conf

</IfModule>
```

Reiniciamo el apache

```
# /etc/init.d/apache2 reload
```

Hacemos la prueba con el siguiente comando

```
# curl localhost/index.html?exec=/bin/bash
```

```
root@c29ba9ebced8:/var/www/html# curl localhost/index.html?exec=/bin/bash
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.38 (Debian) Server at localhost Port 80</address>
</body></html>
root@c29ba9ebced8:/var/www/html#
```

INSTALAR Y CONFIGURAR MOD_EVASIVE ATAQUES DOS, DDOS, FUERZA BRUTA.

Para instalar el módulo mod_evasive en Debian / Ubuntu, ingrese lo siguiente

```
# apt-get install libapache2-mod-evasive
```

```
root@cb3dbf75f6a5:/var/www/html# sudo apt-get install libapache2-mod-evasive
bash: sudo: command not found
root@cb3dbf75f6a5:/var/www/html# apt-get install libapache2-mod-evasive
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bsd-mailx cron exim4-base exim4-config exim4-daemon-light libevent-2.1-6
  libgnutls-dane0 libblockfile-bin libblockfile1 libunbound8 netbase psmisc
Suggested packages:
  anacron logrotate checksecurity exim4-doc-html | exim4-doc-info eximon4
  spf-tools-perl swaks dns-root-data
The following NEW packages will be installed:
  bsd-mailx cron exim4-base exim4-config exim4-daemon-light libapache2-mod-evasive
  libevent-2.1-6 libgnutls-dane0 libblockfile-bin libblockfile1 libunbound8 netbase
  psmisc
0 upgraded, 13 newly installed, 0 to remove and 4 not upgraded.
Need to get 3380 kB of archives.
After this operation, 7250 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian buster/main amd64 cron amd64 3.0p11-134+deb10u1 [99.
0 kB]
Get:2 http://deb.debian.org/debian buster/main amd64 netbase all 5.6 [19.4 kB]
Get:3 http://deb.debian.org/debian buster/main amd64 libblockfile-bin amd64 1.14-1.1 [19
```

Una vez que mod_evasive está instalado, puede verificarlo con el siguiente comando:

```
# apachectl -M | grep evasive
```

```
root@c29ba9ebced8:/var/www/html# apachectl -M | grep evasive
AH00558: apache2: Could not reliably determine the server's fully qualified domain
name, using 172.24.0.4. Set the 'ServerName' directive globally to suppress this
message
  evasive20_module (shared)
root@c29ba9ebced8:/var/www/html#
```

El archivo de configuración predeterminado Mod_evasive se encuentra en /etc/apache2/mods-enabled/evasive.conf. Por defecto, las opciones de configuración mod_evasive están deshabilitadas. Puede habilitarlo editando el evasive.confarchivo y luego personalizarlo según sus requisitos.

Editar la siguiente ruta:

```
# nano /etc/apache2/mods-enabled/evasive.conf
```

Se habilita lo siguiente :

```
<IfModule mod_evasive20.c>
```

```
    DOSHashTableSize 3097
```

```
    DOSPageCount 2
```

```
    DOSSiteCount 50
```

```
    DOSPageInterval 1
```

```
    DOSSiteInterval 1
```

```
    DOSBlockingPeriod 10
```

```
    DOSEmailNotify email@yourdomain.com
```

```
    DOSSystemCommand "su - someuser -c '/sbin/... %s ...'"
```

```
    DOSLogDir "/var/log/mod_evasive"
```

```
</IfModule>
```

```
semestral@semestral: ~
GNU nano 3.2          /etc/apache2/mods-enabled/evasive.conf

IfModule mod_evasive20.c>
  #DOSHashTableSize    3097
  #DOSPageCount        2
  #DOSSiteCount        50
  #DOSPageInterval     1
  #DOSSiteInterval     1
  #DOSBlockingPeriod   10

  #DOSEmailNotify      you@yourdomain.com
  #DOSSystemCommand    "su - someuser -c '/sbin/... %s ...'"
  #DOSLogDir           "/var/log/mod_evasive"
  DOSHashTableSize 3097
  DOSPageCount 2
  DOSSiteCount 50
  DOSPageInterval 1
  DOSSiteInterval 1
  DOSBlockingPeriod 10
  DOSEmailNotify christian.espinoza1008@gmail.com
  DOSSystemCommand "su - someuser -c '/sbin/... %s ...'"
  DOSLogDir "/var/log/mod_evasive"
  DOSWhitelist 127.0.0.1

;/IfModule>
```

Guarde y cierre el archivo, luego cree un directorio de registro para mod_evasive.

```
mkdir /var/log/mod_evasive
```

```
chown -R www-data:www-data /var/log/mod_evasive
```

```
root@c29ba9ebced8:/var/www/html# mkdir /var/log/mod_evasive
root@c29ba9ebced8:/var/www/html# chown -R www-data:www-data /var/log/mod_evasive
root@c29ba9ebced8:/var/www/html#
```

Restart el apache para aplicar cambios

```
# /etc/init.d/apache2 reload
```

```
root@cb3dbf75f6a5:/var/www/html# /etc/init.d/apache2 restart
[...] Restarting Apache httpd web server: apache2Terminated
root@cb3dbf75f6a5:/var/www/html# semestral@semestral:~$
```

La configuración anterior es totalmente personalizable y debe configurarse según las capacidades de su servidor y los flujos de tráfico esperados.

La explicación de cada parameter es la siguiente:

- DOSHashTableSize: especifica cómo mod_evasive realiza un seguimiento de quién accede a qué. Aumentar el número mejora el rendimiento, pero también consume más memoria.
- DOSPageCount: especifica el umbral para el número de solicitudes para la misma página por intervalo de página.
- DOSSiteCount: especifica el umbral para el número total de solicitudes de cualquier objeto por el mismo cliente en el mismo oyente por intervalo de sitio.
- DOSPageInterval: el intervalo utilizado en el umbral de recuento de páginas.
- DOSSiteInterval: el intervalo utilizado en el umbral de recuento de sitios.
- DOSBlockingPeriod: especifica la cantidad de tiempo (en segundos) que una IP está bloqueada.
- DOSEmailNotify: especifica la dirección de correo electrónico de notificación si la dirección IP aparece en la lista negra.
- DOSLogDir: especifica el directorio de registro.

Una vez que todo esté configurado correctamente, hagamos una prueba para ver si el módulo funciona correctamente.

Aquí, utilizaremos un test.plscript escrito por desarrolladores mod_evasive para probar mod_evasive.

Este es un script perl ubicado en /usr/share/doc/libapache2-mod-evasive/examples/test.pl.

Ejecute el script con el siguiente comando:

```
# perl /usr/share/doc/libapache2-mod-evasive/examples/test.pl
```

```
root@cb3dbf75f6a5:/usr/share/doc/libapache2-mod-evasive/examples# perl test.pl
HTTP/1.1 400 Bad Request
```

Si tira un error por no estar el test.pl en la carpeta examples entonces tenemos que crear el fichero.

Navegamos hasta la carpeta examples

```
# cd /usr/share/doc/libapache2-mod-evasive/examples/
```

```
root@c29ba9ebced8:/var/www/html# perl /usr/share/doc/libapache2-mod-evasive/examples/test.pl
Can't open perl script "/usr/share/doc/libapache2-mod-evasive/examples/test.pl":
No such file or directory
root@c29ba9ebced8:/var/www/html# cd /usr/share/doc/libapache2-mod-evasive/examples/
root@c29ba9ebced8:/usr/share/doc/libapache2-mod-evasive/examples#
```

Creamos un fichero llamado test.pl

```
# touch test.pl
```

```
root@c29ba9ebced8:/usr/share/doc/libapache2-mod-evasive/examples# touch test.pl
root@c29ba9ebced8:/usr/share/doc/libapache2-mod-evasive/examples# ls
test.pl
root@c29ba9ebced8:/usr/share/doc/libapache2-mod-evasive/examples#
```

Abrimos para editar.

```
# nano test.pl
```

```
root@c29ba9ebced8:/usr/share/doc/libapache2-mod-evasive/examples# nano test.pl
```

Pegar lo que sale en este enlace y guardar.

https://github.com/KoHead/mod_evasive/blob/master/test.pl

Código:

```
#!/usr/bin/perl
```

```
# test.pl: small script to test mod_dosevasive's effectiveness
```

```
use IO::Socket;
```

```
use strict;
```

```
for(0..100) {
```

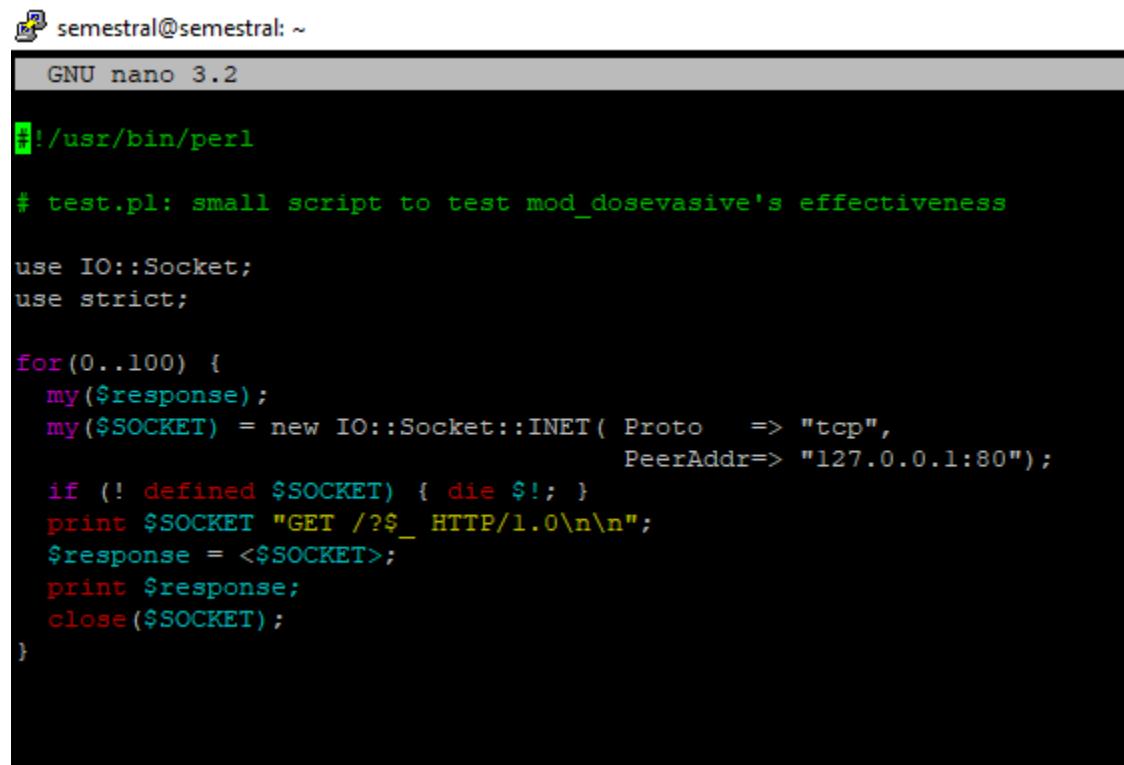
```
    my($response);
```

```
my($SOCKET) = new IO::Socket::INET( Proto  => "tcp",
                                    PeerAddr=> "127.0.0.1:80");

if (! defined $SOCKET) { die $!; }

print $SOCKET "GET /?$_ HTTP/1.0\n\n";
$response = <$SOCKET>;
print $response;
close($SOCKET);

}
```



The screenshot shows a terminal window titled 'semestral@semestral: ~'. The title bar indicates 'GNU nano 3.2'. The terminal content displays a Perl script named 'test.pl'. The script uses the IO::Socket module to create a TCP socket and send an HTTP GET request to '127.0.0.1:80'. It then reads the response from the socket and prints it. The script is run 100 times in a loop.

```
#!/usr/bin/perl

# test.pl: small script to test mod_dosevasive's effectiveness

use IO::Socket;
use strict;

for(0..100) {
    my($response);
    my($SOCKET) = new IO::Socket::INET( Proto    => "tcp",
                                        PeerAddr=> "127.0.0.1:80");
    if (! defined $SOCKET) { die $!; }
    print $SOCKET "GET /?$_ HTTP/1.0\n\n";
    $response = <$SOCKET>;
    print $response;
    close($SOCKET);
}
```

Ejecute el script con el siguiente comando:

```
# perl /usr/share/doc/libapache2-mod-evasive/examples/test.pl
```

```
root@cb3dbf75f6a5:/usr/share/doc/libapache2-mod-evasive/examples# perl test.pl
HTTP/1.1 400 Bad Request
```

INSTALAR Y CONFIGURAR MOD_QOS PARA ATAQUES SLOWLORIS.

```
# apt-get install libapache2-mod-qos
```

```
root@cb3dbf75f6a5:/usr/share/doc/libapache2-mod-evasive/examples# apt install libapache2modos
test.pl
root@cb3dbf75f6a5:/usr/share/doc/libapache2-mod-evasive/examples# apt-get install libapache2-mod-qos
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libapache2-mod-qos
```

Entramos a la ruta de mods-available

```
# cd /etc/apache2/mods-available/
```

```
root@c29ba9ebced8:/usr/share/modsecurity-crs# cd /etc/apache2/mods-available/
root@c29ba9ebced8:/etc/apache2/mods-available#
```

Verificamos la ruta del fichero qos.load

```
#nano qos.load
```

```
root@c29ba9ebced8:/etc/apache2/mods-available# nano qos.load
root@c29ba9ebced8:/etc/apache2/mods-available#
```

```
semestral@semestral: ~/semestral
GNU nano 3.2                               qos.load

LoadModule qos_module /usr/lib/apache2/modules/mod_qos.so
```

Editamos el fichero qos.conf

```
# nano qos.conf
```

```
semestral@semestral: ~
GNU nano 3.2                               qos.conf                         Modified
# allows max 50 connections from a single ip address:
QS_SrvMaxConnPerIP      50

# handles connections from up to 100000 different IPs
QS_ClientEntries 100000
# will allow only 50 connections per IP
QS_SrvMaxConnPerIP 50
# maximum number of active TCP connections is limited to 256
MaxClients      256
# disables keep-alive when 70% of the TCP connections are occupied:
QS_SrvMaxConnClose    180
# minimum request/response speed (deny slow clients blocking the server, ie$)
QS_SrvMinDataRate     150 1200
# and limit request header and body (carefull, that limits uploads and post$)
# LimitRequestFields    30
# QS_LimitRequestBody   102400
</IfModule>

Save modified buffer? (Answering "No" will DISCARD changes.)
Y Yes
N No          ^C Cancel
```

Lineas:

```
QS_SrvMaxConnPerIP 50
```

```
QS_ClientEntries 100000
```

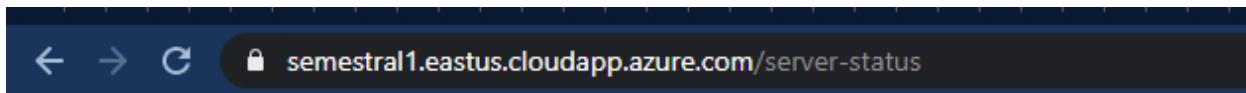
```
MaxClients 256
```

```
QS_SrvMaxConnClose 180
```

```
root@cb3dbf75f6a5:/etc/apache2/mods-available# vi qos.conf
root@cb3dbf75f6a5:/etc/apache2/mods-available# a2enmod qos
Module qos already enabled
root@cb3dbf75f6a5:/etc/apache2/mods-available# /etc/init.d/apache2 restart
[....] Restarting Apache httpd web server: apache2Terminated
root@cb3dbf75f6a5:/etc/apache2/mods-available# semestral@semestral:~$
```

Reiniciamos

```
/etc/init.d/apache2 reload
```



Forbidden

You don't have permission to access this resource.

Apache/2.4.38 (Debian) Server at semestral1.eastus.cloudapp.azure.com Port 80

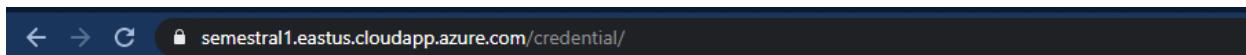
HARDENING DE APACHE

Antes de implementar el hardening de apache2 primero creamos una carpeta para validar que funciona la configuración.

Creamos una carpeta llamada credential

```
# mkdir credential
```

```
root@c29ba9ebced8:/var/www/html# mkdir credential
root@c29ba9ebced8:/var/www/html# ls
credential          wp-blog-header.php    wp-load.php
index.php           wp-comments-post.php  wp-login.php
license.txt         wp-config-sample.php  wp-mail.php
modsecurity.conf    wp-config.php       wp-settings.php
owasp-modsecurity-crs wp-content        wp-signup.php
readme.html         wp-cron.php       wp-trackback.php
wp-activate.php     wp-includes       xmlrpc.php
wp-admin            wp-links-opml.php
root@c29ba9ebced8:/var/www/html#
```



Forbidden

You don't have permission to access this resource.

Apache/2.4.38 (Debian) Server at semestral1.eastus.cloudapp.azure.com Port 80

ELIMINAR EL BANNER DE LA VERSIÓN DEL SERVIDOR

Lo primero que notamos es que no tenemos a los recursos y eso esta bien debido a la implementación de los mod sin embargo muestra la versión de apache Apache/2.4.38 el nombre del sistema operativo (Debian) y el puerto donde esta corriendo Port 80.

Escribimos:

```
# cd /etc/apache2/conf-enabled/
```

```
root@c29ba9ebced8:/var/www/html# cd /etc/apache2/conf-enabled/
root@c29ba9ebced8:/etc/apache2/conf-enabled#
```

Configuramos el archivo security.conf

```
# nano security.conf
```

```
root@c29ba9ebced8:/etc/apache2/conf-enabled# ls
charset.conf          other-vhosts-access-log.conf  serve-cgi-bin.conf
docker-php.conf        remoteip.conf
localized-error-pages.conf  security.conf
root@c29ba9ebced8:/etc/apache2/conf-enabled# nano security.conf
```

Cambiamos a ServerTokens Prod, ServerSignature off

```
semestral@semestral: ~/semestral
GNU nano 3.2           security.conf           Modified

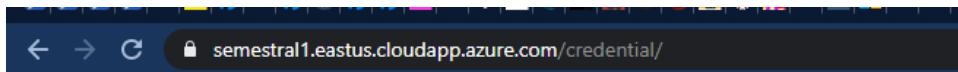
# and compiled in modules.
# Set to one of: Full | OS | Minimal | Minor | Major | Prod
# where Full conveys the most information, and Prod the least.
#ServerTokens Minimal
#ServerTokens OS
ServerTokens Prod
#ServerTokens Full

#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of: On | Off | EMail
#ServerSignature Off
#ServerSignature On
ServerSignature Off
```

reiniciamos

```
/etc/init.d/apache2 reload
```

Ahora Podemos ver que la información esta oculta.



Forbidden

You don't have permission to access this resource.

DESHABILITAMOS EL LISTADO DE DIRECTORIOS

```
# cd/etc/apache2/
```

```
# nano apache2.conf
```

Borramos la palabra Indexes

```
semestral@semestral: ~/semestral
GNU nano 3.2                               apache2.conf

        AllowOverride None
        Require all denied
</Directory>

<Directory /usr/share>
        AllowOverride None
        Require all granted
</Directory>

<Directory /var/www/>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

Reiniciamos

```
/etc/init.d/apache2 reload
```

TIMEOUT, LIMITREQUESTBODY, FILEETAG

Para poder hacer esto debemos ir a la ruta

```
# cd /etc/apache2/
```

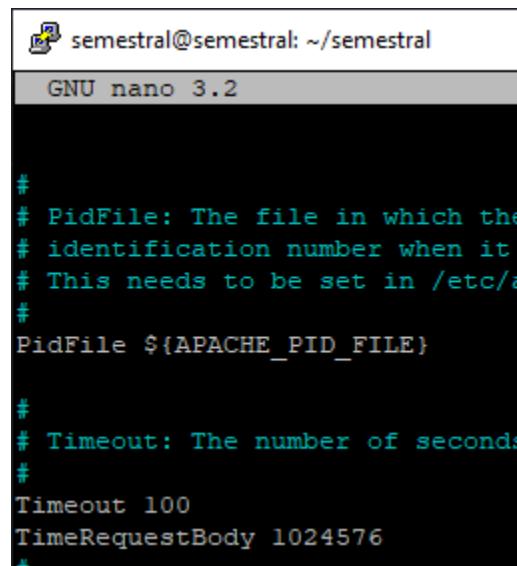
Ejecutamos el comando

```
# nano apache2.conf
```

Cambiamos el Timeout 300 que viene por defecto a 100 y agregamos:

```
LimitRequestBody 1024576
```

```
FileETag none
```



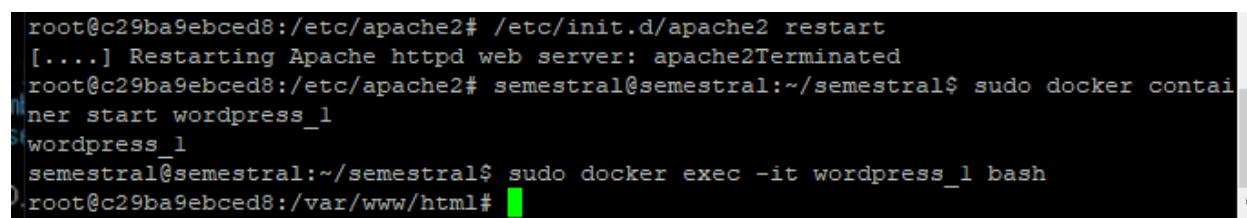
```
semestral@semestral: ~/semestral
GNU nano 3.2

#
# PidFile: The file in which the
# identification number when it
# This needs to be set in /etc/a
#
PidFile ${APACHE_PID_FILE}

#
# Timeout: The number of seconds
#
Timeout 100
LimitRequestBody 1024576
#
```

Reiniciamos

```
/etc/init.d/apache2 reload
```

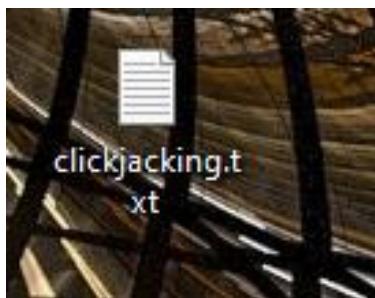


```
root@c29ba9ebced8:/etc/apache2# /etc/init.d/apache2 restart
[....] Restarting Apache httpd web server: apache2Terminated
root@c29ba9ebced8:/etc/apache2# semestral@semestral:~/semestral$ sudo docker container start wordpress_1
wordpress_1
semestral@semestral:~/semestral$ sudo docker exec -it wordpress_1 bash
root@c29ba9ebced8:/var/www/html#
```

CLICKJACKING

Antes de implementarlo debemos crear un archivo para hacer el clickjacking y probar si funciona

Creamos un fichero .txt



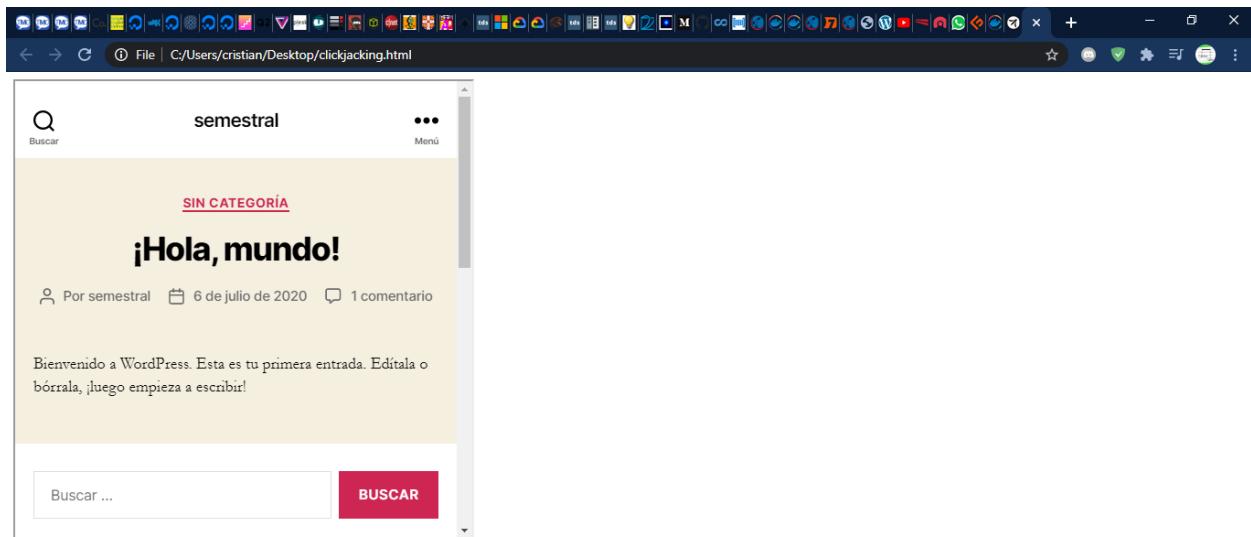
Luego dentro pegamos el siguiente script de html siendo `http://www.target.site` la dirección de la pagina:

```
<html>
  <head>
    <title>Clickjack test page</title>
  </head>
  <body>
    <iframe src="http://www.target.site/" width="500" height="500"></iframe>
  </body>
</html>
```

Luego renombramos como .hmtl



Si la <http://www.target.site> página se carga correctamente en el marco, entonces el sitio es vulnerable y no tiene ningún tipo de protección contra los ataques de clickjacking



Ahora procederemos a crear la protección para ello vamos a la ruta del apache y editamos el fichero .http

```
# cd /var/www/html
```

Luego creamos un fichero llamado headers.php

```
# touch headers.php
```

```
root@af0311a8abe7:/var/www/html# touch headers.php
root@af0311a8abe7:/var/www/html# ls
credential          wp-admin           wp-links-opml.php
headers.php         wp-blog-header.php  wp-load.php
index.php          wp-comments-post.php wp-login.php
license.txt        wp-config-sample.php wp-mail.php
modsecurity.conf   wp-config.php      wp-settings.php
owasp-modsecurity-crs wp-content       wp-signup.php
readme.html         wp-cron.php       wp-trackback.php
wp-activate.php    wp-includes      xmlrpc.php
root@af0311a8abe7:/var/www/html#
```

Ahora abrimos el fichero y colocamos lo siguiente

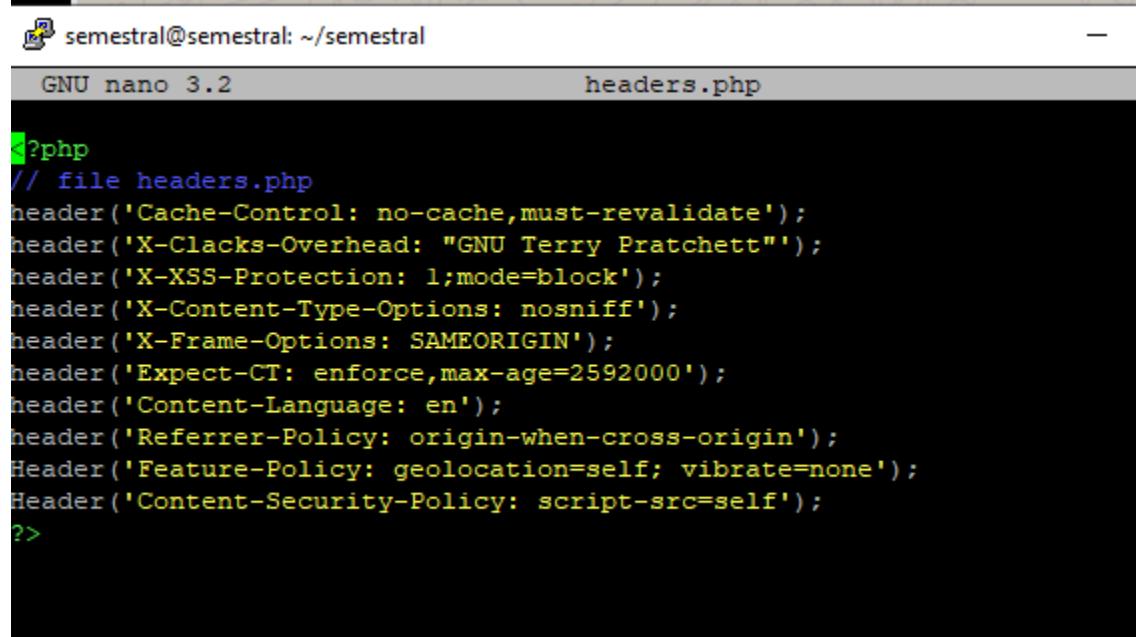
```
# nano headers.php
```

Lineas:

```
<?php
```

```
// file headers.php
```

```
header('Cache-Control: no-cache,must-revalidate');
header('X-Clacks-Overhead: "GNU Terry Pratchett"');
header('X-XSS-Protection: 1;mode=block');
header('X-Content-Type-Options: nosniff');
header('X-Frame-Options: SAMEORIGIN');
header('Expect-CT: enforce,max-age=2592000');
header('Content-Language: en');
header('Referrer-Policy: origin-when-cross-origin');
Header('Feature-Policy: geolocation=self; vibrate=none');
Header('Content-Security-Policy: script-src=self');
?>
```



```
semestral@semestral: ~/semestral
GNU nano 3.2          headers.php

<?php
// file headers.php
header('Cache-Control: no-cache,must-revalidate');
header('X-Clacks-Overhead: "GNU Terry Pratchett"');
header('X-XSS-Protection: 1;mode=block');
header('X-Content-Type-Options: nosniff');
header('X-Frame-Options: SAMEORIGIN');
header('Expect-CT: enforce,max-age=2592000');
header('Content-Language: en');
header('Referrer-Policy: origin-when-cross-origin');
Header('Feature-Policy: geolocation=self; vibrate=none');
Header('Content-Security-Policy: script-src=self');
?>
```

Abrimos el fichero .htaccess

```
# nano .htaccess
php_value auto_prepend_file /var/www/html/headers.php
```

```
semestral@semestral: ~/semestral
GNU nano 3.2 .htaccess

# BEGIN WordPress
# Las directivas (lneas) entre `BEGIN WordPress` y `END WordPress`
# , y solo se deberan modificar mediante filtros de WordPress.
# Cualquier cambio en las directivas que hay entre esos marcadores ser
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>
# END WordPress
php_value auto_prepend_file /var/www/html/headers.php
```

Reiniciamos

/etc/init.d/apache2 reload

```
root@af0311a8abe7:/var/www/html# nano headers.php
root@af0311a8abe7:/var/www/html# nano .htaccess
root@af0311a8abe7:/var/www/html# /etc/init.d/apache2 reload
[ ok ] Reloading Apache httpd web server: apache2.
root@af0311a8abe7:/var/www/html#
```

Nos dirigimos a la siguiente pagina para probar su funcionamiento

<https://securityheaders.com/>

ponemos nuestra dirección dns y le damos en scan.

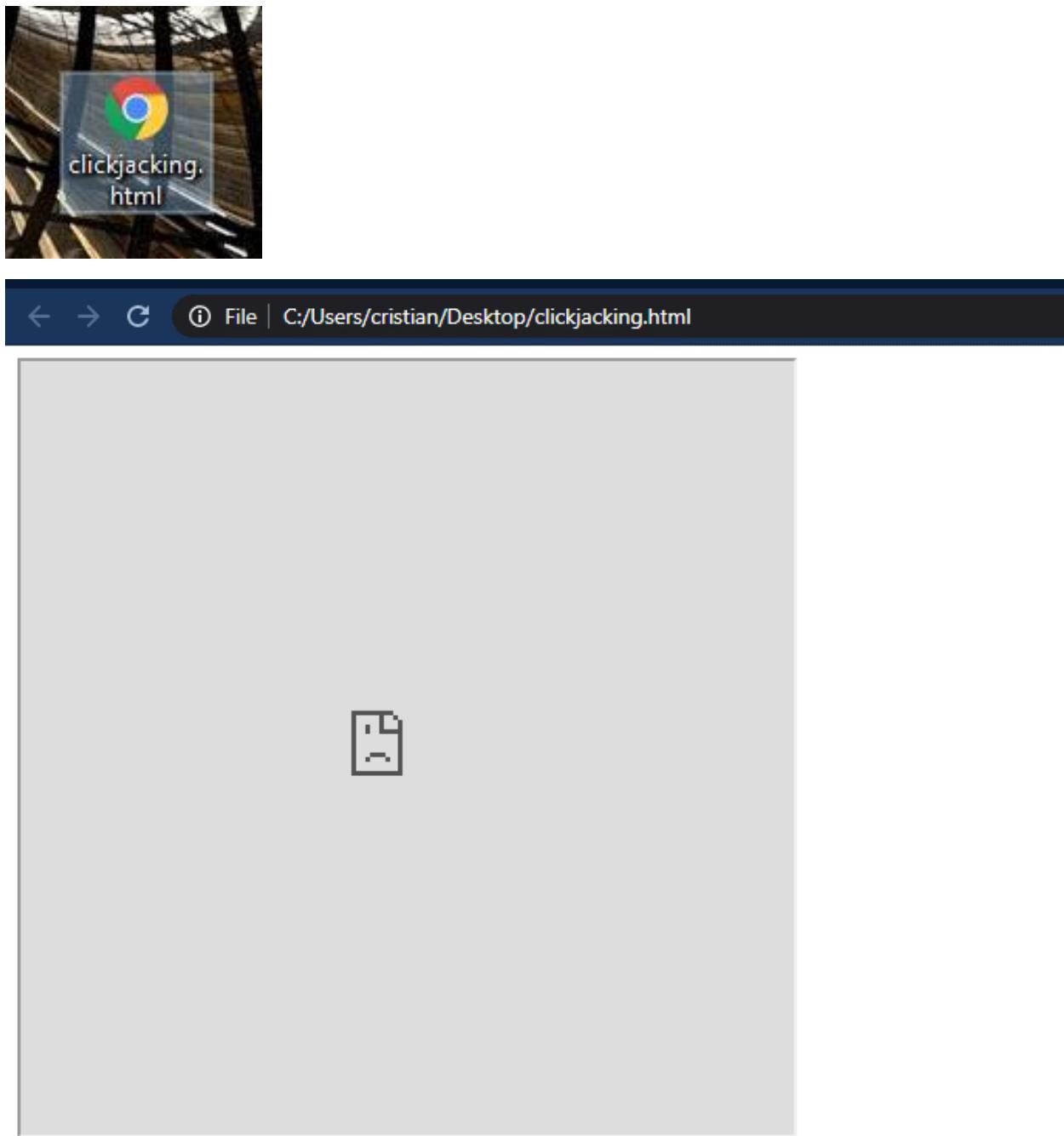
The screenshot shows the Security Headers website interface. At the top, it says "Security Headers" and "Sponsored by Report URI". Below that is a large button labeled "Scan your site now". Underneath is a text input field containing "https://semestral1.eastus.cloudapp.azure.com" and a "Scan" button. Below the input field are two checkboxes: "Hide results" (unchecked) and "Follow redirects" (checked). The main result area is titled "Security Report Summary" and shows a large green "A" icon. It lists the following details:

- Site: <https://semestral1.eastus.cloudapp.azure.com/>
- IP Address: 13.90.43.38
- Report Time: 15 Jul 2020 04:28:54 UTC
- Headers:
 - ✓ X-Content-Type-Options
 - ✓ X-Frame-Options
 - ✓ Referrer-Policy
 - ✓ Feature-Policy
 - ✓ Content-Security-Policy
 - ✓ Strict-Transport-Security
- Warning: Grade capped at A, please see warnings below.

This is a detailed view of the "Security Report Summary" page from the Security Headers website. It provides a comprehensive breakdown of the security headers for the specified site. The key findings are:

- Site:** <https://semestral1.eastus.cloudapp.azure.com/>
- IP Address:** 13.90.43.38
- Report Time:** 15 Jul 2020 04:28:54 UTC
- Headers:** The page lists several header types as being present and correctly configured, including X-Content-Type-Options, X-Frame-Options, Referrer-Policy, Feature-Policy, Content-Security-Policy, and Strict-Transport-Security.
- Warning:** A note states "Grade capped at A, please see warnings below."

Y ejecutamos el fichero html nuevamente para comprobar si funciona.



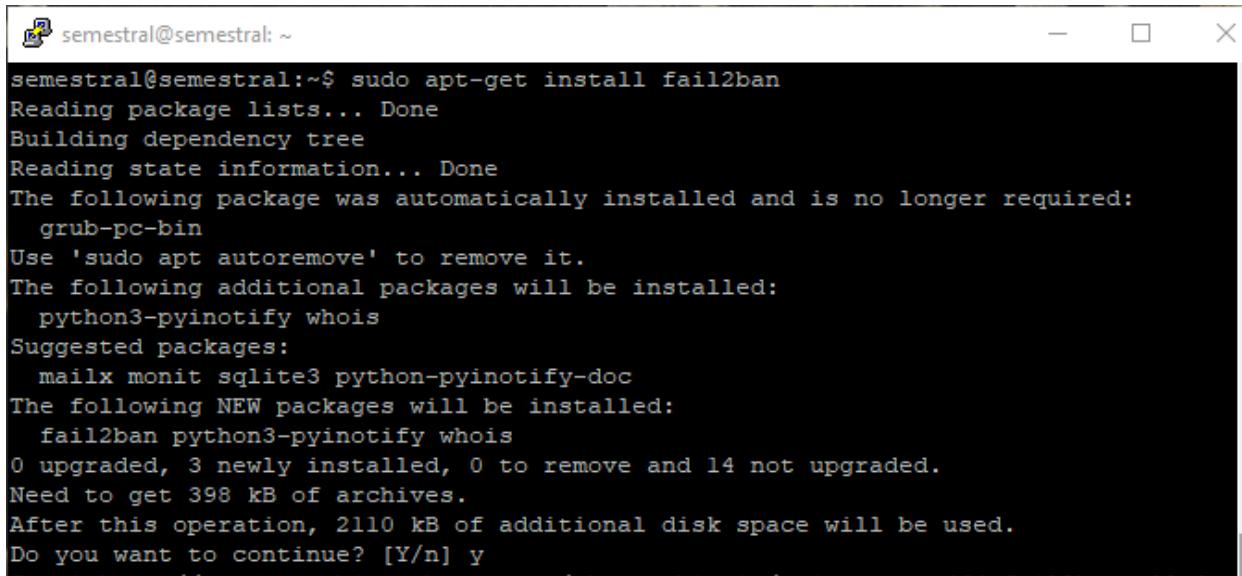
Podemos observar que ha funcionado perfectamente

HARDENING DE SSH: JAILKIT, FAIL2BAN

FAIL2BAN

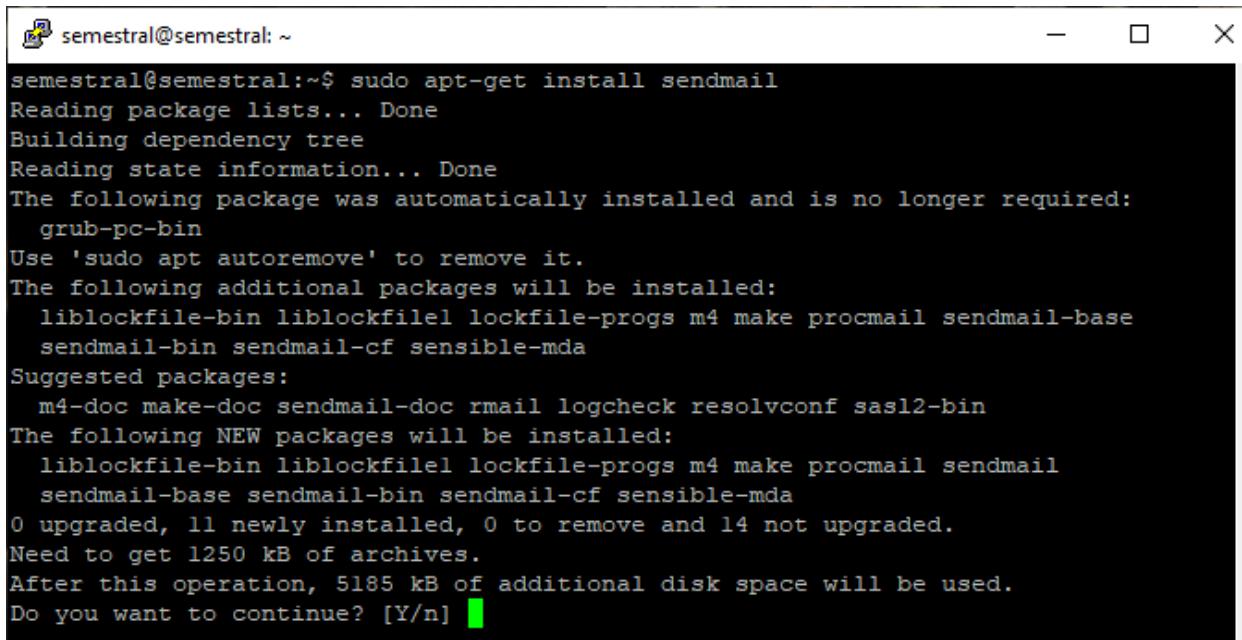
Para instalar fail2ban, escribimos lo siguiente en la terminal:

```
$ sudo apt-get install fail2ban
```



```
semestral@semestral: ~$ sudo apt-get install fail2ban
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  python3-pyinotify whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 14 not upgraded.
Need to get 398 kB of archives.
After this operation, 2110 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
$ sudo apt-get install sendmail
```



```
semestral@semestral: ~$ sudo apt-get install sendmail
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libblockfile-bin libblockfilel lockfile-progs m4 make procmail sendmail-base
  sendmail-bin sendmail-cf sensible-mda
Suggested packages:
  m4-doc make-doc sendmail-doc rmail logcheck resolvconf sasl2-bin
The following NEW packages will be installed:
  libblockfile-bin libblockfilel lockfile-progs m4 make procmail sendmail
  sendmail-base sendmail-bin sendmail-cf sensible-mda
0 upgraded, 11 newly installed, 0 to remove and 14 not upgraded.
Need to get 1250 kB of archives.
After this operation, 5185 kB of additional disk space will be used.
Do you want to continue? [Y/n] █
```

Ahora procedemos a configurar el fail2ban primero vamos al directorio fail2ban y creamos un fichero llamado jail.local

```
$ cd /etc/fail2ban
```

```
$sudo nano jail.local
```

```
semestral@semestral: /etc/fail2ban  
semestral@semestral:~$ cd /etc/fail2ban  
semestral@semestral:/etc/fail2ban$ sudo nano jail.local
```

Pegamos lo siguiente

[DEFAULT]

```
#ignoreip = nuestraip  
bantime = 1m  
findtime = 1m  
maxretry = 3  
destemail = correo@dominio.com  
sender = correo@dominio.com  
sendername = Fail2ban  
mta = sendmail  
action = %(action_mwl)s
```

[sshd]

```
enabled = true  
port = 22  
filter = sshd  
logpath = /var/log/auth.log
```

```
semestral@semestral: /etc/fail2ban
GNU nano 2.9.3                               jail.local

[DEFAULT]
bantime = 1m
findtime = 1m
maxretry = 3
destemail = christian.espinoza1008@gmail.com
sender = christian.espinoza1008@gmail.com
sendername = Fail2ban
mta = sendmail
action = %(action_mw)s

[sshd]
enabled = true
port = 22
filter = sshd
logpath = /var/log/auth.log
```

reiniciamos

```
semestral@semestral:/etc/fail2ban$ ls
action.d      filter.d    jail.local          paths-debian.conf
fail2ban.conf  jail.conf   paths-arch.conf    paths-opensuse.conf
fail2ban.d     jail.d     paths-common.conf
```

\$ sudo /etc/init.d/fail2ban restart

```
semestral@semestral:/etc/fail2ban$ sudo /etc/init.d/fail2ban restart
[ ok ] Restarting fail2ban (via systemctl): fail2ban.service.
semestral@semestral:/etc/fail2ban$
```

Verificamos

Primero nos aseguramos que se haya iniciado correctamente con los logs.

\$ sudo tail -f /var/log/fail2ban.log (verificar los logs del fail2ban)

```
semestral@semestral:/etc/fail2ban$ sudo tail -f /var/log/fail2ban.log
2020-07-25 19:49:38,232 fail2ban.jail      [7169]: INFO  Jail 'sshd' uses pyinotify {}
2020-07-25 19:49:38,235 fail2ban.jail      [7169]: INFO  Initiated 'pyinotify' backend
2020-07-25 19:49:38,236 fail2ban.filter   [7169]: INFO  maxLines: 1
2020-07-25 19:49:38,254 fail2ban.server   [7169]: INFO  Jail sshd is not a JournalFilter instance
2020-07-25 19:49:38,254 fail2ban.filter   [7169]: INFO  Added logfile: '/var/log/auth.log' (pos = 317664, hash =
1593fb9413bb37aca9b04889cac40797led0036e)
2020-07-25 19:49:38,257 fail2ban.filter   [7169]: INFO  encoding: UTF-8
2020-07-25 19:49:38,257 fail2ban.filter   [7169]: INFO  maxRetry: 3
2020-07-25 19:49:38,257 fail2ban.filter   [7169]: INFO  findtime: 60
2020-07-25 19:49:38,258 fail2ban.actions [7169]: INFO  banTime: 60
2020-07-25 19:49:38,262 fail2ban.jail      [7169]: INFO  Jail 'sshd' started
```

\$ sudo fail2ban-client status (estado numero de jail)

```
semestral@semestral:/etc/fail2ban$ sudo fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:    sshd
semestral@semestral:/etc/fail2ban$
```

\$ sudo fail2ban-client status sshd (verificar los intentos fallidos de iniciar session)

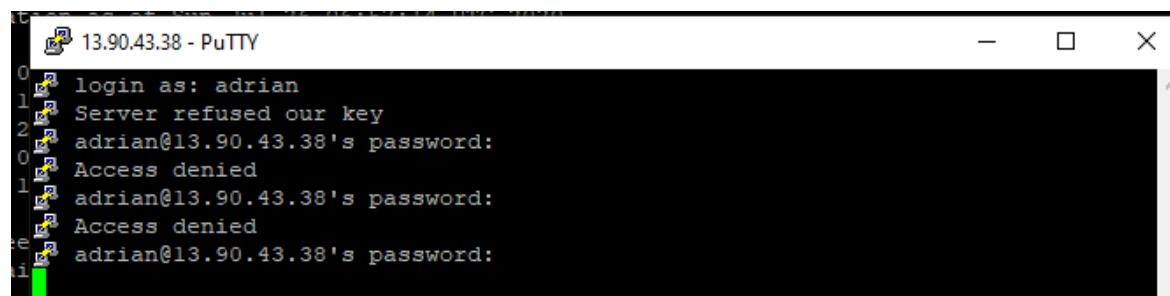
```
semestral@semestral:/etc/fail2ban$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed:      0
| ` File list:          /var/log/auth.log
`- Actions
  |- Currently banned: 0
  |- Total banned:     0
  ` Banned IP list:
semestral@semestral:/etc/fail2ban$
```

Ahora probamos el ban

```
semestral@semestral:/etc/fail2ban$ exit
```

Entramos nuevamente atravez del puerto 22 con el ssh

Luego intentamos entramos a cualquier usuario y entramos 3 veces erroneamente la contraseña para comprobar el baneo.



En el intento 4 nos dejara esperando debido a que hemos sido baneado por el tiempo que definimos en el fail2ban



Esperamos a que pase el tiempo de baneo para volver a entrar

```

13.90.43.38 - PuTTY
login as: semestral

semestral@semestral: ~
Memory usage: 10%          IP address for eth0:      10.0.2.4
Swap usage:   0%          IP address for docker0: 172.17.0.1

* "If you've been waiting for the perfect Kubernetes dev solution for
macOS, the wait is over. Learn how to install Microk8s on macOS."
https://www.techrepublic.com/article/how-to-install-microk8s-on-macos/

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

14 packages can be updated.
0 updates are security updates.

Last login: Sat Jul 25 19:24:42 2020 from 190.219.214.35
+-----+
|S|E|M|E|S|T|R|A|L| |D|E| |S|E|G|U|R|I|D|A|D| |E|N| |T|E|C|N|O|L|O|G|I|A| |D|E|
+-----+
|C|O|M|P|U|T|A|C|I|O|N|
+-----+
semestral@semestral:~$ 
```

Verificamos nuevamente

\$ sudo tail -f /var/log/fail2ban.log (verificar los logs del fail2ban)

```

semestral@semestral:/etc/fail2ban$ sudo tail -f /var/log/fail2ban.log
2020-07-25 19:49:38,257 fail2ban.filter      [7169]: INFO      encoding: UTF-8
2020-07-25 19:49:38,257 fail2ban.filter      [7169]: INFO      maxRetry: 3
2020-07-25 19:49:38,257 fail2ban.filter      [7169]: INFO      findtime: 60
2020-07-25 19:49:38,258 fail2ban.actions    [7169]: INFO      banTime: 60
2020-07-25 19:49:38,262 fail2ban.jail       [7169]: INFO      Jail 'sshd' started
2020-07-25 19:56:37,842 fail2ban.filter      [7169]: INFO      [sshd] Found 190.219.214.35 - 2020-07-25 19:56:37
2020-07-25 19:57:14,191 fail2ban.filter      [7169]: INFO      [sshd] Found 190.219.214.35 - 2020-07-25 19:57:14
2020-07-25 19:57:27,261 fail2ban.filter      [7169]: INFO      [sshd] Found 190.219.214.35 - 2020-07-25 19:57:27
2020-07-25 19:57:27,657 fail2ban.actions    [7169]: NOTICE    [sshd] Ban 190.219.214.35
2020-07-25 19:58:28,264 fail2ban.actions    [7169]: NOTICE    [sshd] Unban 190.219.214.35 
```

\$ sudo fail2ban-client status sshd (verificar los intentos fallidos de iniciar session)

```
semestral@semestral:/etc/fail2ban$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed:      3
| `-' File list:        /var/log/auth.log
`- Actions
  |- Currently banned: 0
  |- Total banned:     1
  `-' Banned IP list:
semestral@semestral:/etc/fail2ban$
```

JAILKIT

Antes de empezar configuramos lo siguiente

sudo nano /etc/ssh/sshd_config

```
semestral@semestral: ~$ sudo nano /etc/ssh/sshd_config
```

cambiamos lo siguiente

PermitRootLogin prohibit-password

a

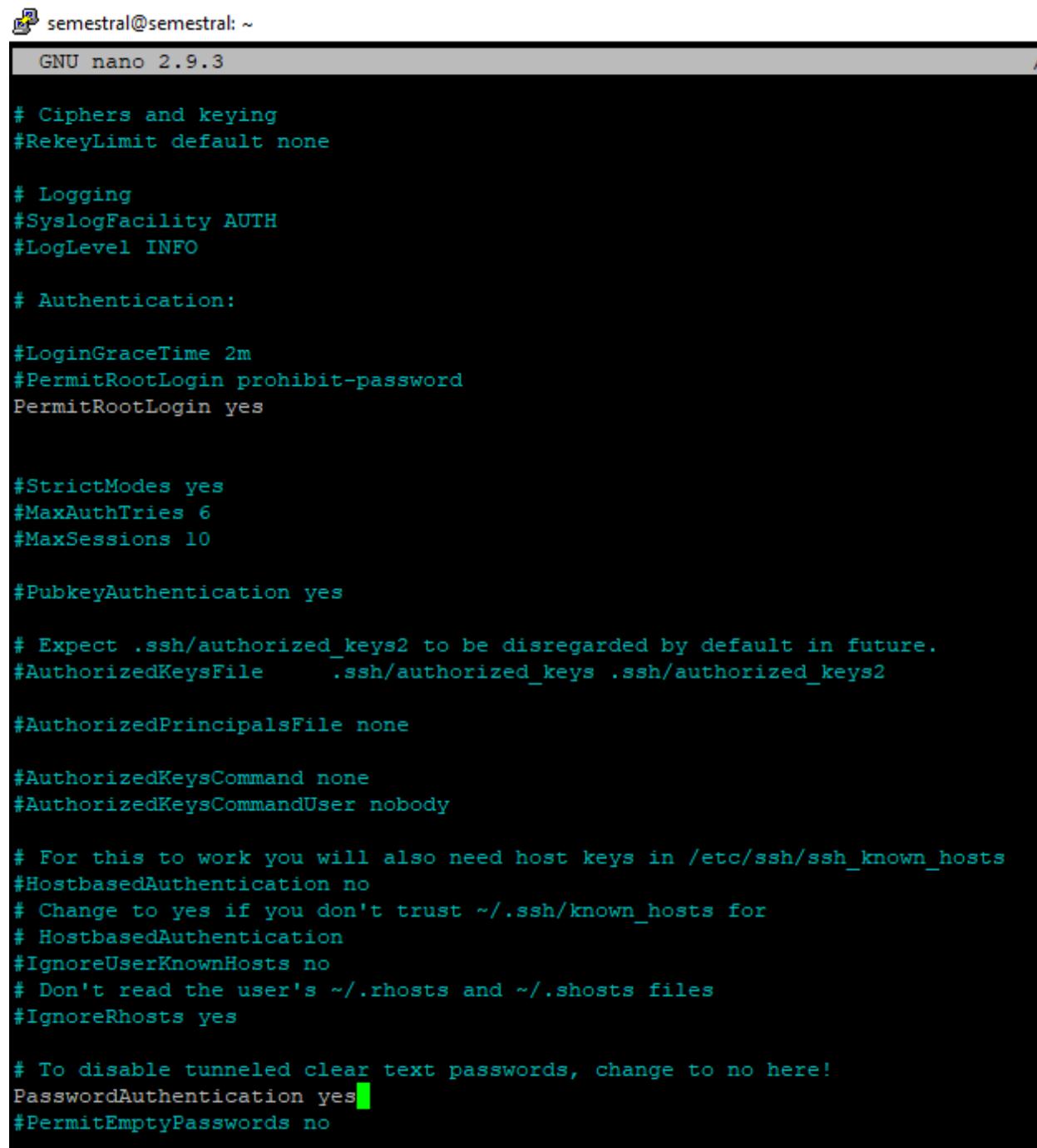
PermitRootLogin yes

Tambien

PasswordAuthentication no

a

PasswordAuthentication yes



The screenshot shows a terminal window titled "semestral@semestral: ~" running the "GNU nano 2.9.3" editor. The file being edited is the SSH daemon's configuration file, likely /etc/ssh/sshd_config. The configuration includes various parameters such as ciphers, logging levels, authentication methods (including password, public key, and host-based), and host keys. A specific line of interest is highlighted in green: "PasswordAuthentication yes". This line controls whether password authentication is enabled for the SSH service.

```
# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
PermitRootLogin yes

#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

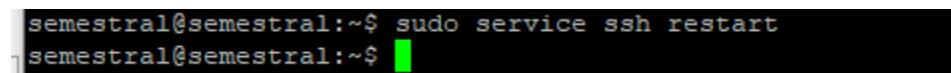
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
```

Reiniciamos

sudo service ssh restart



The screenshot shows a terminal window with the command "sudo service ssh restart" being typed. The command is intended to restart the SSH service to apply the changes made in the configuration file. The terminal prompt is "semestral@semestral:~\$".

```
semestral@semestral:~$ sudo service ssh restart
semestral@semestral:~$
```

Instalar jailkit en Ubuntu / Debian

Instale los siguientes paquetes

```
$ sudo apt-get install build-essential autoconf automake libtool flex bison debhelper binutils-gold
```

```
semestral@semestral:~$ sudo apt-get install build-essential autoconf automake libtool flex bison debhelper binutils-gold
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'binutils' instead of 'binutils-gold'
The following package was automatically installed and is no longer required:
  grub-pc-bin
```

Descargue Jailkit de la siguiente url, o visite el sitio web para obtener la última url si ha cambiado.

```
$ cd /tmp
```

<http://olivier.sessink.nl/jailkit/jailkit-2.16.tar.gz>

```
$ wget http://olivier.sessink.nl/jailkit/jailkit-2.16.tar.gz
```

```
semestral@semestral:/tmp$ wget http://olivier.sessink.nl/jailkit/jailkit-2.16.tar.gz
$: command not found
semestral@semestral:/tmp$ wget http://olivier.sessink.nl/jailkit/jailkit-2.16.tar.gz
--2020-07-25 20:39:00--  http://olivier.sessink.nl/jailkit/jailkit-2.16.tar.gz
Resolving olivier.sessink.nl (olivier.sessink.nl)... 95.97.76.243, 2001:470:1f14:4ab::2
Connecting to olivier.sessink.nl (olivier.sessink.nl)|95.97.76.243|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://olivier.sessink.nl/jailkit/jailkit-2.16.tar.gz [following]
--2020-07-25 20:39:00--  https://olivier.sessink.nl/jailkit/jailkit-2.16.tar.gz
Connecting to olivier.sessink.nl (olivier.sessink.nl)|95.97.76.243|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 139407 (136K) [application/x-gzip]
Saving to: 'jailkit-2.16.tar.gz'

jailkit-2.16.tar.gz          100%[=====] 2020-07-25 20:39:01 (479 KB/s) - 'jailkit-2.16.tar.gz' saved [139407/139407]

semestral@semestral:/tmp$
```

Extraer el archivo

```
$ tar -vxzf jailkit-2.16.tar.gz
```

```
semestral@semestral:/tmp$ ls
jailkit-2.16      systemd-private-e999fe6327bd4576a0eda096c89343f3-systemd-resolved.service-Nzd0Ie
jailkit-2.16.tar.gz  systemd-private-e999fe6327bd4576a0eda096c89343f3-systemd-timesyncd.service-fcno50
semestral@semestral:/tmp$
```

Compile jailkit y cree un archivo deb

Jailkit ya viene con el código y las configuraciones necesarias para compilarse en un archivo deb que se puede instalar perfectamente en sistemas basados en Debian. Simplemente ejecute el siguiente comando y debería hacerlo.

```
$ cd jailkit-2.16/
```

```
$ echo 5 > debian/compat
```

```
semestral@semestral:/tmp$ cd jailkit-2.16/  
semestral@semestral:/tmp/jailkit-2.16$ echo 5 > debian/compat  
semestral@semestral:/tmp/jailkit-2.16$
```

```
$ sudo ./debian/rules binary
```

```
semestral@semestral:/tmp/jailkit-2.16$ sudo ./debian/rules binary
```

```
dpkg-deb: building package 'jailkit' in '../jailkit_2.16-1_amd64.deb'.  
dpkg-deb: building package 'jailkit-dbgsym' in 'debian/.debscratch-space/  
build-jailkit/jailkit-dbgsym_2.16-1_amd64.deb'.  
      Renaming jailkit-dbgsym_2.16-1_amd64.deb to jailkit-dbgsym_2.16-1_amd64.  
ddeb  
make[1]: Leaving directory '/tmp/jailkit-2.16'  
semestral@semestral:/tmp/jailkit-2.16$
```

Instale el archivo deb

El comando anterior crearía un archivo deb llamado jailkit_2.16-1_amd64.deb .

```
$ cd ..
```

```
$ sudo dpkg -i jailkit_2.16-1_*amd64.deb
```

```
semestral@semestral:/tmp/jailkit-2.16$ cd ..  
semestral@semestral:/tmp$ sudo dpkg -i jailkit_2.16-1_*amd64.deb  
Selecting previously unselected package jailkit.  
(Reading database ... 93844 files and directories currently installed.)  
Preparing to unpack jailkit_2.16-1_amd64.deb ...  
Unpacking jailkit (2.16-1) ...  
Setting up jailkit (2.16-1) ...  
Processing triggers for ureadahead (0.100.0-21) ...  
Processing triggers for systemd (237-3ubuntu0.41) ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
semestral@semestral:/tmp$
```

Nos volvemos usuario root

```
$sudo su
```

```
semestral@semestral:/tmp$ sudo su
+-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+
|S|E|M|E|S|T|R|A|L| |D|E| |S|E|G|U|R|I|D|A|D| |E|N| |T|E|C|N|O|L|O|G|I|A| |D|E|
+-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+
|C|O|M|P|U|T|A|C|I|O|N|
+-----+
root@semestral:/tmp# 
```

Configurar el entorno de la cárcel

Es necesario que haya un directorio donde se configure todo el entorno de la cárcel. Vamos a hacerlo en /opt/jail. Esto puede ser lo que sea.

```
$ cd /opt
```

```
# mkdir jail
```

```
root@semestral:/tmp# cd /opt
root@semestral:/opt# mkdir jail
root@semestral:/opt# ls
containerd  jail
root@semestral:/opt# 
```

Root debería ser el propietario de este directorio. Así que sácalo.

```
# sudo chown root:root /opt/jail
```

```
root@semestral:/opt# sudo chown root:root /opt/jail
root@semestral:/opt# 
```

Configure los programas para que estén disponibles dentro de la cárcel

Primero vamos a

```
# cd /tmp/jailkit-2.16
```

Todos los programas que deben estar disponibles en la cárcel deben copiarse dentro de ella utilizando el comando jk_init.

METODO 1

```
# sudo jk_init -v /jail basicshell
```

```
# sudo jk_init -v /jail editors
```

```
# sudo jk_init -v /jail extendedshell
```

```
# sudo jk_init -v /jail netutils
```

```
# sudo jk_init -v /jail ssh
```

```
# sudo jk_init -v /jail sftp
```

```
# sudo jk_init -v /jail/jk_lsh
```

O de una vez

METODO 2

```
# sudo jk_init -v /opt/jail netutils basicshell jk_lsh openvpn ssh sftp
```

Cree el usuario que será encarcelado

Necesita un usuario para poner dentro de la cárcel. Vamos a crear uno pero debemos tomar en cuenta que los nombres son en minúscula.

```
# sudo adduser adrian
```

```
Password: semestral
```

Luego nos pedira algunos datos

Enter the new value, or press ENTER for the default

```
Full Name []: Adrian Franco
```

```
Room Number []:
```

```
Work Phone []:
```

```
Home Phone []:
```

```
Other []:
```

```
Is the information correct? [Y/n] y
```

```
root@semestral:/tmp/jailkit-2.16# sudo adduser adrian
Adding user `adrian' ...
Adding new group `adrian' (1001) ...
Adding new user `adrian' (1001) with group `adrian' ...
Creating home directory `/home/adrian' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for adrian
Enter the new value, or press ENTER for the default
    Full Name []: Adrian Franco
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

Tenga en cuenta que este es un usuario normal que se crea en el sistema de archivos real y no dentro de la cárcel.

En el siguiente paso, este usuario será encarcelado dentro de la cárcel.

En este punto, si echa un vistazo a / etc / passwd, verá una entrada al final que se ve así

Usamos este comando

```
# egrep adrian /etc/passwd
```

```
root@semestral:/tmp/jailkit-2.16# egrep adrian /etc/passwd
adrian:x:1001:1001:Adrian Franco,,,,:/home/adrian:/bin/bash
root@semestral:/tmp/jailkit-2.16#
```

O podemos ir directo

```
# nano /etc/passwd
```

Debería salir esto

adrian:x:1001:1001:Adrian Franco,,,,:/home/adrian:/bin/bash Encarcelar al usuario

Ahora es el momento de poner al usuario dentro de la cárcel.

```
# sudo jk_jailuser -m -j /opt/jail/ adrian
```

Al hacer esto, el usuario ladrón ahora ha sido encarcelado.

Ahora, si echa un vistazo a / etc / passwd, la última entrada se vería así

```
#egrep adrian /etc/passwd
```

```
root@semestral:/tmp/jailkit-2.16# egrep adrian /etc/passwd
adrian:x:1001:1001:Adrian Franco,,,,:/opt/jail/.:home/adrian:/usr/sbin/jk_chroots
h
root@semestral:/tmp/jailkit-2.16#
```

Tenga en cuenta que las últimas 2 partes que indican el usuario doméstico y el tipo de shell han cambiado. El directorio de inicio del usuario ahora está dentro del entorno de la cárcel en / opt / jail. El shell del usuario ahora es un programa especial llamado jk_chrootsh que proporcionará el shell encarcelado.

Es este shell particular llamado jk_chrootsh que lleva al usuario dentro de la cárcel, cada vez que inicia sesión en el sistema.

La configuración de la cárcel por ahora está casi terminada. Pero si intenta conectarse a la identificación desde ssh, fallará de esta manera:

```
# ssh adrian@localhost
```

```
root@semestral:/tmp/jailkit-2.16# ssh adrian@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:hbbv4/U3GnvQcBWAff6vK5TOMS2Zf2K33Z4BSFyTnhA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
adrian@localhost's password:
```

La conexión se cerrará. Esto sucede porque el usuario realmente tiene un shell limitado.

Dar bash shell al usuario dentro de la cárcel

```
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-1032-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Sat Jul 25 20:53:41 UTC 2020

 System load:  0.0          Processes:           132
 Usage of /:   13.8% of 28.90GB  Users logged in:    1
 Memory usage: 11%
 Swap usage:   0%          IP address for eth0:  10.0.2.4
                           IP address for docker0: 172.17.0.1

 * "If you've been waiting for the perfect Kubernetes dev solution for
   macOS, the wait is over. Learn how to install Microk8s on macOS."
   https://www.techrepublic.com/article/how-to-install-microk8s-on-macos/

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

14 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

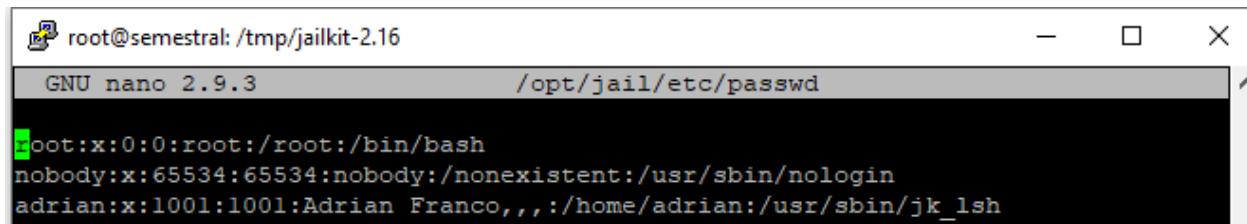
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Connection to localhost closed.
root@semestral:/tmp/jailkit-2.16#
```

Lo siguiente que debe hacer es darle al usuario un shell bash adecuado, pero dentro de la cárcel.

Abre el siguiente archivo

```
# nano /opt/jail/etc/passwd
```



The screenshot shows a terminal window titled "root@semestral: /tmp/jailkit-2.16". The title bar also displays "GNU nano 2.9.3" and the file path "/opt/jail/etc/passwd". The main area of the terminal shows the contents of the passwd file:

```
root:x:0:0:root:/root:/bin/bash
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
adrian:x:1001:1001:Adrian Franco,,,,:/home/adrian:/usr/sbin/jk_lsh
```

Es el archivo de contraseña dentro de la cárcel. Se vería algo así

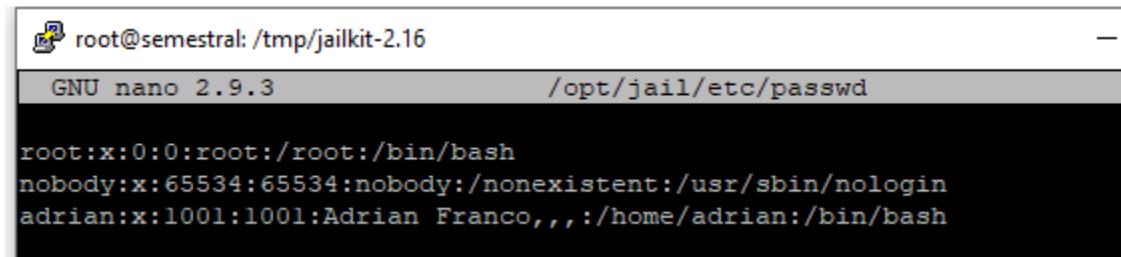
```
root:x:0:0:root:/root:/bin/bash
```

```
adrian:x:1006:1005:,,,:/home/adrian:/usr/sbin/jk_lsh
```

Cambie / usr / sbin / jk_lsh a / bin / bash

```
root:x:0:0:root:/root:/bin/bash
```

```
adrian:x:1006:1005:,,,:/home/adrian:/bin/bash
```



The screenshot shows a terminal window titled "root@semestral: /tmp/jailkit-2.16". The title bar also displays "GNU nano 2.9.3" and the file path "/opt/jail/etc/passwd". The main area of the terminal shows the contents of the passwd file, which has been modified:

```
root:x:0:0:root:/root:/bin/bash
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
adrian:x:1001:1001:Adrian Franco,,,,:/home/adrian:/bin/bash
```

volvemos a entrar

```
# ssh adrian@localhost
```

```
root@semestral:/tmp/jailkit-2.16# ssh adrian@localhost
adrian@localhost's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-1032-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Sat Jul 25 20:56:48 UTC 2020

 System load:  0.01          Processes:           134
 Usage of /:   13.8% of 28.90GB  Users logged in:    1
 Memory usage: 11%          IP address for eth0:  10.0.2.4
 Swap usage:   0%          IP address for docker0: 172.17.0.1

 * "If you've been waiting for the perfect Kubernetes dev solution for
macOS, the wait is over. Learn how to install Microk8s on macOS."
https://www.techrepublic.com/article/how-to-install-microk8s-on-macos/

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

14 packages can be updated.
0 updates are security updates.

Last login: Sat Jul 25 20:53:41 2020 from 127.0.0.1
bash: groups: command not found
bash: figlet: command not found
adrian@semestral:~$
```

O usamos este comando

```
# sudo su adrian
```

Hacemos pruebas dentro del usuario encarcelado

```
touch hola
```

```
ls
```

```
nano hola
```

```
apt-get install nano
```

```
rm hola
```

```
ls
```

```
adrian@semestral:~$ sudo su adrian
bash: sudo: command not found
adrian@semestral:~$ touch hola
adrian@semestral:~$ ls
hola
adrian@semestral:~$ nano hola
bash: nano: command not found
adrian@semestral:~$ apt-get install nano
bash: apt-get: command not found
adrian@semestral:~$ rm hola
adrian@semestral:~$ ls
adrian@semestral:~$
```

Para salir

```
$ exit
```

Borrar un usuario

```
$ sudo deluser nombredelusuario
```

Lista de todos los usuarios

```
$ cut -d: -f1 /etc/passwd
```

```
root@semestral:/tmp/jailkit-2.16# cut -d: -f1 /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
syslog
messagebus
_apt
lxd
uuid
dnsmasq
landscape
sshd
pollinate
semestral
smmta
smmsp
adrian
root@semestral:/tmp/jailkit-2.16#
```

FIGLET Y BANNER DE CONEXIÓN

Primero descargamos la librería con el siguiente comando

```
$ sudo apt-get install figlet
```

```
semestral@semestral:~$ sudo apt-get install figlet
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  figlet
0 upgraded, 1 newly installed, 0 to remove and 12 not upgraded.
Need to get 133 kB of archives.
After this operation, 752 kB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu bionic/universe amd64 figlet amd64 2.2.5-3 [133 kB]
Fetched 133 kB in 0s (5358 kB/s)
Selecting previously unselected package figlet.
(Reading database ... 85875 files and directories currently installed.)
Preparing to unpack .../figlet_2.2.5-3_amd64.deb ...
```

Para ver las fuentes escribimos:

```
$ showfigfonts
```

```
semestral@semestral:~$ showfigfonts
```

```
semestral@semestral: ~

small :
[REDACTED]

smscript :
[REDACTED]
```

```
$ showfigfonts semestral
```

```
semestral@semestral:~$ showfigfonts semestral
banner :

#####
# ###### #   # #####  ##### ##### #####
#   #   ## #   #   #   #   #   #   #
##### ###### # ## # #####  ##### ##### #####
#   #   #   #   #   #   #   #   #   #
#   #   #   #   #   #   #   #   #   #
##### ###### #   # #####  ##### ##### #####
#   #   #   #   #   #   #   #   #   #

big :

#####
# ###### #   # #####  ##### ##### #####
#   #   ## #   #   #   #   #   #
##### ###### # ## # #####  ##### ##### #####
#   #   #   #   #   #   #   #   #
#   #   #   #   #   #   #   #   #
##### ###### #   # #####  ##### ##### #####
#   #   #   #   #   #   #   #   #   #
```

Nos dirigimos al directorio etc y editamos el archivo bash.bashrc

```
$ cd /etc/
```

```
$ sudo nano bash.bashrc
```

```
semestral@semestral:~$ cd /etc/
semestral@semestral:/etc$ sudo nano bash.bashrc
```

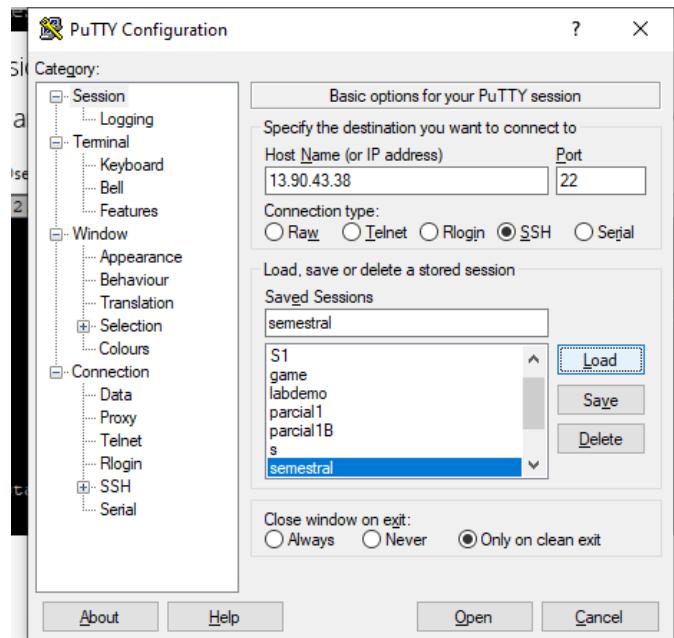
Agregamos lo siguiente:

```
$ figlet -f digital SEMESTRAL DE SEGURIDAD EN TECNOLOGIA DE COMPUTACION
```

```
semestral@semestral:/etc$ nano bash.bashrc
GNU nano 2.9.3
bash.bashrc

/usr/lib/command-not-found -- "$1"
return $?
elif [ -x /usr/share/command-not-found/command-not-found ]; then
/usr/share/command-not-found/command-not-found -- "$1"
return $?
else
printf "%s: command not found\n" "$1" >&2
return 127
fi
}
fi
figlet -f digital SEMESTRAL DE SEGURIDAD EN TECNOLOGIA DE COMPUTACION
```

Abrimos otra terminal en el putty para probar su funcionamiento



```

semestral@semestral: ~
└── login as: semestral
    └── Authenticating with public key "rsa-key-20200516"
        └── Passphrase for key "rsa-key-20200516":
    └── Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-1032-azure x86_64)

    * Documentation: https://help.ubuntu.com
    * Management: https://landscape.canonical.com
    * Support: https://ubuntu.com/advantage

    System information as of Wed Jul 15 18:51:20 UTC 2020

    System load: 0.01      Users logged in:          1
    Usage of /: 12.6% of 28.90GB  IP address for eth0:      10.0.2.4
    Memory usage: 12%      IP address for docker0:    172.17.0.1
    Swap usage: 0%        IP address for br-8da4f2a0960c: 172.25.0.1
    Processes: 141        IP address for br-c5dbc94d9be7: 172.24.0.1

    * "If you've been waiting for the perfect Kubernetes dev solution for
      macOS, the wait is over. Learn how to install Microk8s on macOS."
      https://www.techrepublic.com/article/how-to-install-microk8s-on-macos/

    * Canonical Livepatch is available for installation.
      - Reduce system reboots and improve kernel security. Activate at:
        https://ubuntu.com/livepatch

    11 packages can be updated.
    1 update is a security update.

    Last login: Wed Jul 15 18:46:38 2020 from 190.141.232.86
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |S|E|M|E|S|T|R|A|L| |D|E| |S|E|G|U|R|I|D|A|D| |E|N| |T|E|C|N|O|L|O|G|I|A| |D|E|
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |C|O|M|P|U|T|A|C|I|O|N|
    +---+---+---+---+---+---+
semestral@semestral:~$ 

```

QUOTAS

Primero agregamos un usuario

```
semestral@semestral:~$ sudo adduser christian
[sudo] password for semestral:
Adding user `christian' ...
Adding new group `christian' (1002) ...
Adding new user `christian' (1002) with group `christian' ...
Creating home directory `/home/christian' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for christian
Enter the new value, or press ENTER for the default
      Full Name []:
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []:
Is the information correct? [Y/n] y
```

Instalación de las herramientas de cuota

Para establecer y verificar las cuotas, primero necesitamos instalar las herramientas de línea de comando de cuotas usando apt. Actualicemos nuestra lista de paquetes, luego instale el paquete:

```
$ sudo apt update
```

```
$ sudo apt install quota
```

```
semestral@semestral:~$ sudo apt install quota
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libtirpc1
Suggested packages:
  libnet-ldap-perl rpcbind
The following NEW packages will be installed:
  libtirpc1 quota
0 upgraded, 2 newly installed, 0 to remove and 14 not upgraded.
Need to get 336 kB of archives.
After this operation, 1744 kB of additional disk space will be used.
```

Puede verificar que las herramientas estén instaladas ejecutando el `quotacomando` y solicitando su información de versión:

\$ quota -version

```
semestral@semestral:~$ quota --version
Quota utilities version 4.04.
Compiled with: USE_LDAP_MAIL_LOOKUP EXT2_DIRECT HOSTS_ACCESS RPC RPC_SETQUOTA BSD_BEHAVIOUR
Bugs to jack@suse.cz
semestral@semestral:~$
```

instalación del módulo de kernel de cuota

Si está en un servidor virtual basado en la nube, su instalación predeterminada de Ubuntu Linux puede no tener los módulos del núcleo necesarios para admitir la gestión de cuotas. Para verificar, usaremos find para buscar los módulos quota_v1 y quota_v2 en el /lib/modules/... directorio:

\$ find /lib/modules/`uname -r` -type f -name '*quota_v*.ko'

```
semestral@semestral:~$ find /lib/modules/`uname -r` -type f -name '*quota_v*.ko'
/lib/modules/5.3.0-1032-azure/kernel/fs/quota/quota_v1.ko
/lib/modules/5.3.0-1032-azure/kernel/fs/quota/quota_v2.ko
semestral@semestral:~$
```

Actualizar las opciones de montaje del sistema de archivos

Para activar las cuotas en un sistema de archivos en particular, necesitamos montarlo con algunas opciones relacionadas con las cuotas especificadas. Hacemos esto actualizando la entrada del sistema de archivos en el /etc/fstab archivo de configuración. Abra ese archivo en su editor de texto favorito ahora:

\$ sudo nano /etc/fstab

Agregamos lo siguiente

usrquota,grpquota

```
semestral@semestral:~$ nano /etc/fstab
# CLOUD_IMG: This file was created/modified by the Cloud Image build process
UUID=b0dd9d06-536e-4144-ac5f-6db8e20295b3      /       ext4    defaults,discard      0 0
UUID=25DA-4525 /boot/efi     vfat    defaults,discard      0 0
/dev/disk/cloud/azure_resource-part1   /mnt    auto    defaults,nofail,x-systemd.requires=cloud-init.service,comment=cloudconfig      0      2
```

Este cambio nos permitirá habilitar cuotas de usuario (usrquota) y de grupo (grpquota) en el sistema de archivos. Si solo necesita uno u otro, puede omitir la opción no utilizada. Si su fstab línea ya tenía algunas opciones enumeradas en lugar de defaults, debería agregar las nuevas opciones al final de lo que ya esté allí, asegurándose de separar todas las opciones con una coma y sin espacios.

Vuelva a montar el sistema de archivos para que las nuevas opciones surtan efecto:

\$ sudo mount -o remount /

```
semestral@semestral:~$ sudo mount -o remount /
semestral@semestral:~$
```

Podemos verificar que las nuevas opciones se utilizaron para montar el sistema de archivos mirando el /proc/mountsarchivo. Aquí, usamos grep para mostrar solo la entrada del sistema de archivos raíz en ese archivo:

```
$ cat /proc/mounts | grep '^'
```

```
semestral@semestral:~$ cat /proc/mounts | grep '^'
/dev/sdbl / ext4 rw,relatime,discard,quota,usrquota,grpquota 0 0
```

habilitación de cuotas

Antes de finalmente activar el sistema de cuotas, necesitamos ejecutar manualmente el quotacheck comando una vez:

```
$ sudo quotacheck -ugm /
```

```
semestral@semestral:~$ sudo quotacheck -ugm /
semestral@semestral:~$
```

Podemos verificar que se crearon los archivos apropiados enumerando el directorio raíz:

```
$ ls /
```

```
semestral@semestral:~$ ls /
aquota.group  bin  dev  home  initrd.img.old  lib64  media  opt  root  sbin  srv  tmp  var  vmlinuz.old
aquota.user   boot  etc  initrd.img  lib      lost+found  mnt  proc  run  snap  sys  usr  vmlinuz  webmin-setup.out
semestral@semestral:~$
```

```
sudo quotaon -v /
```

```
semestral@semestral:~$ sudo quotaon -v /
/dev/sdbl [/]: group quotas turned on
/dev/sdbl [/]: user quotas turned on
semestral@semestral:~$
```

Nuestro servidor ahora está monitoreando y aplicando cuotas, ¡pero aún no hemos establecido ninguna! A continuación, estableceremos una cuota de disco para un solo usuario.

configuración de cuotas para un usuario

Hay algunas formas en que podemos establecer cuotas para usuarios o grupos. Aquí, veremos cómo establecer cuotas con los comandos edquotay setquota.

Usar edquotapara establecer una cuota de usuario

```
$ sudo edquota -u christian
```

```
semestral@semestral: ~
GNU nano 2.9.3                                         /tmp//EdP.alEynNV

Disk quotas for user christian (uid 1002):
Filesystem      blocks    soft    hard   inodes    soft    hard
/dev/sdbl          16        0        0       4        0        0
```

Cambiamos

```
semestral@semestral: ~
GNU nano 2.9.3                                         /tmp//EdP.alEynNV

Disk quotas for user christian (uid 1002):
Filesystem      blocks    soft    hard   inodes    soft    hard
/dev/sdbl          16     100M    100M       4        0        0
```

generación de informes de cuotas

Para generar un informe sobre el uso actual de la cuota para todos los usuarios en un sistema de archivos en particular, use el repquotacomando:

\$ sudo repquota -s /

```
semestral@semestral:~$ sudo repquota -s /
*** Report for user quotas on device /dev/sdbl
Block grace time: 7days; Inode grace time: 7days
                                                 Space limits                               File limits
User        used    soft    hard grace    used    soft    hard grace
-----
root      4052M    OK    OK      197k    0    0
daemon    64K     OK    OK       4    0    0
man      1620K    OK    OK      83    0    0
www-data  121M    OK    OK      5430    0    0
systemd-network 84K     OK      OK      21    0    0
systemd-resolve 32K     OK      OK      11    0    0
syslog    4992K    OK    OK       6    0    0
_apt      24K     OK    OK       4    0    0
lxd       4K     OK    OK       1    0    0
landscape 8K     OK    OK       3    0    0
pollinate 4K     OK    OK       2    0    0
semestral 5232K    OK    OK      375    0    0
smmta    48K     OK    OK       8    0    0
smmsp    4K     OK    OK       1    0    0
adrian   32K     OK    OK       9    0    0
christian 16K    100M  100M      4    0    0
#62583   4K     OK    OK       2    0    0
#501    10068K    OK    OK       1    0    0
#999    222M    OK    OK      393    0    0

semestral@semestral:~$
```

LATCH O PESTILLO DIGITAL EN WORDPRESS Y SSH

Try Community for free ×

Register as a Latch Developer

[Already a Latch user?](#) [Create a new developer account](#)

Enter the credentials of the account that you want to activate as a developer account.

 Email *

 Password *

You can join the Developer Program using this account by entering in the following information.

 Address

 Country ▼

 NIF

 Company

By checking this box, I confirm that I have read and agree to the [Latch Service Agreement for Developers](#) and the [Privacy Policy](#).

I would like to receive emails about Offers and Marketing communications.

[Join the Developer Program](#) [Download contract in PDF](#)

Primero hay que registrarnos en la página de latch.

The screenshot shows the Latch application management interface. The left sidebar includes links for 'Latch Website', 'How it works', 'My applications' (which is selected and highlighted in blue), 'Documentation & SDKs', 'Help & Support', and 'My subscription / Buy'. The main content area is titled 'My applications' and shows the message: 'There aren't any applications yet. Please, click on the "Add a new application" button to create a new one.' A teal button labeled '+ Add a new application' is located at the bottom right of this section. Below this, there is a section titled 'User API Access keys' with a sub-instruction: 'Create access keys for authentication with the Latch user API.' A teal button labeled '+ Generate User API keys' is located at the bottom right of this section.

The screenshot shows a modal or form titled 'My applications' with the sub-section 'Adding a new application'. The top navigation bar shows 'Home > My applications'. The form has a 'Name' field containing 'Semestral' and a teal 'Add application' button at the bottom.

Añadir una nueva aplicación.

My applications

Home > My applications

Edit

Name: Semestral | Application ID: QX6tEXTcmDqNRQzMu8F
Secret: merYCN2bkFPzf6FVj2AHgVPGsNxwvCb3HkrGza2
2-Factor OTP: Disabled | Lock latches after request: Disabled | Save changes

Contact email: semestraleseguridad@gmail.com
Contact phone: 67452317

Developer contact information to show in end user alerts

Operations: Add | Limited secrets: Add | WebHooks: Add

URL: [Input field] Add

	WordPress: The Latch plugin for WordPress allows you to integrate Latch in the WordPress authentication process.	<ul style="list-style-type: none"> • WordPress version 1.5 or later, • Curl extensions active in PHP (uncomment <code>extension=php_curl.dll</code> or <code>extension=curl.so</code> in Windows or Linux <code>php.ini</code> respectively). 		DOWNLOAD
--	---	---	--	--------------------------

Branch: master ▾ Go to file ▾ Code ▾

Bocanegra committed 55f39e0 on May 24, 2019 ...

- latch ADD Latch headers
- .gitignore ADD Latch headers
- README.md ADD Latch headers

README.md

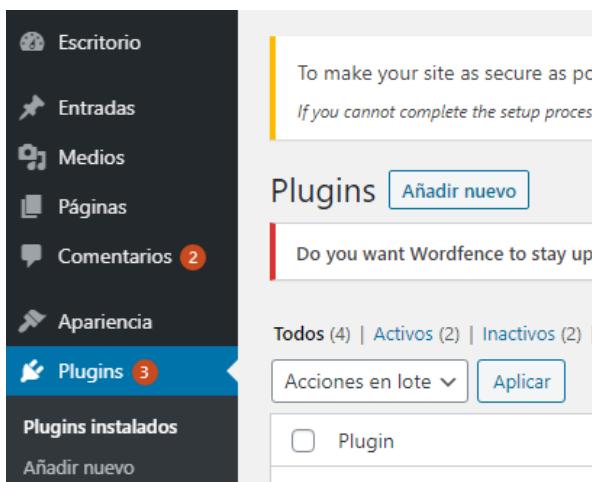
Clone with HTTPS ⓘ
Use Git or checkout with SVN using the web URL.
<https://github.com/ElevenPaths/latch-p>

Open with GitHub Desktop

Download ZIP

LATCH INSTALLATION GUIDE FOR WORDPRESS

Llenar los espacios y descargar el zip del plugin Latch para WordPress.



Entramos a la sección de plugins de WordPress y seleccionamos la opción de añadir nuevo plugin. Elegimos el archivo que descargamos previamente e instalamos. Finalmente, iniciamos el plugin.

Añadir plugins [Subir plugin](#)

Do you want Wordfence to stay up-to-date automatically? [Yes, enable auto-update.](#) | [No thanks.](#)

Si tienes un plugin en un archivo .zip, puedes subirlo e instalarlo desde aquí.

latch.zip

Instalando plugin desde el archivo: latch.zip

Do you want Wordfence to stay up-to-date automatically? [Yes, enable auto-update.](#) | [No thanks.](#)

Descomprimiendo...

Instalando el plugin...

Plugin instalado correctamente.

[Activar plugin](#)

[Volver al instalador de plugins](#)

Algunas de tus traducciones necesitan actualizarse. Espera unos segundos más mientras las actualizamos también.

Actualizando las traducciones de Latch (es_ES)...

Traducción actualizada correctamente.

Do you want Wordfence to stay up-to-date automatically? [Yes, enable auto-update.](#) | [No thanks.](#)

To make your site as secure as possible, take a moment to optimize the Wordfence Web Application Firewall: [CLICK HERE TO CONFIGURE](#) [DISMISS](#)

Ajustes Latch

Ajustes globales

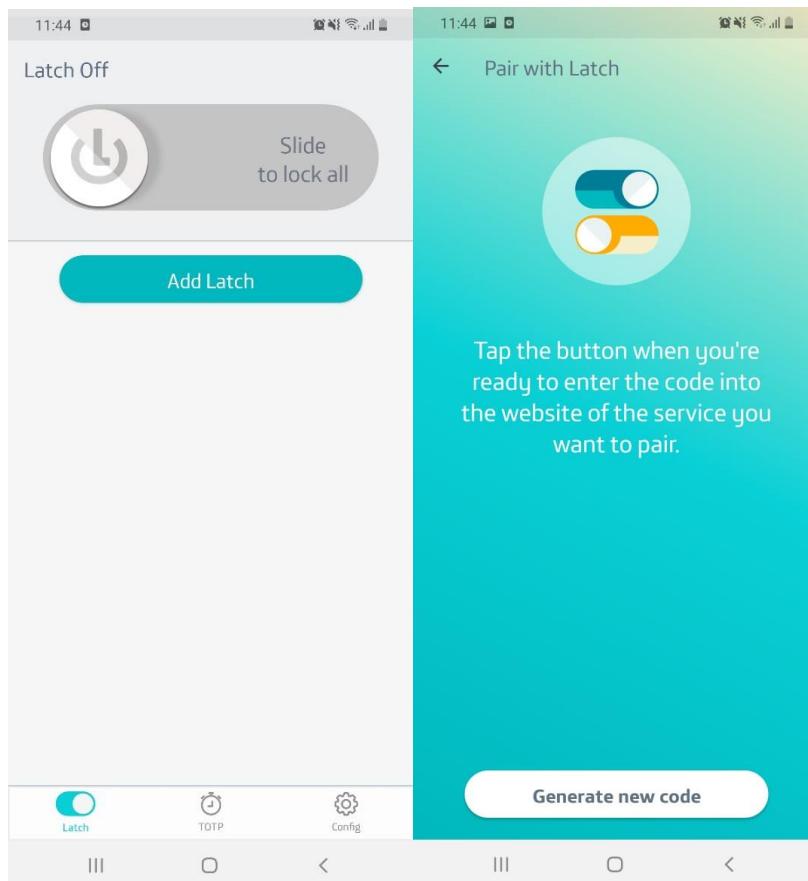
Introduzca los datos generados al registrar la aplicación en Latch:

ID de aplicación: QX6tEXTCcmDqNRQzMu8F

Secreto: merYCN2bkFPzf6FVj2AHgVPXGsNXwvCb3HkrGza2

API URL:

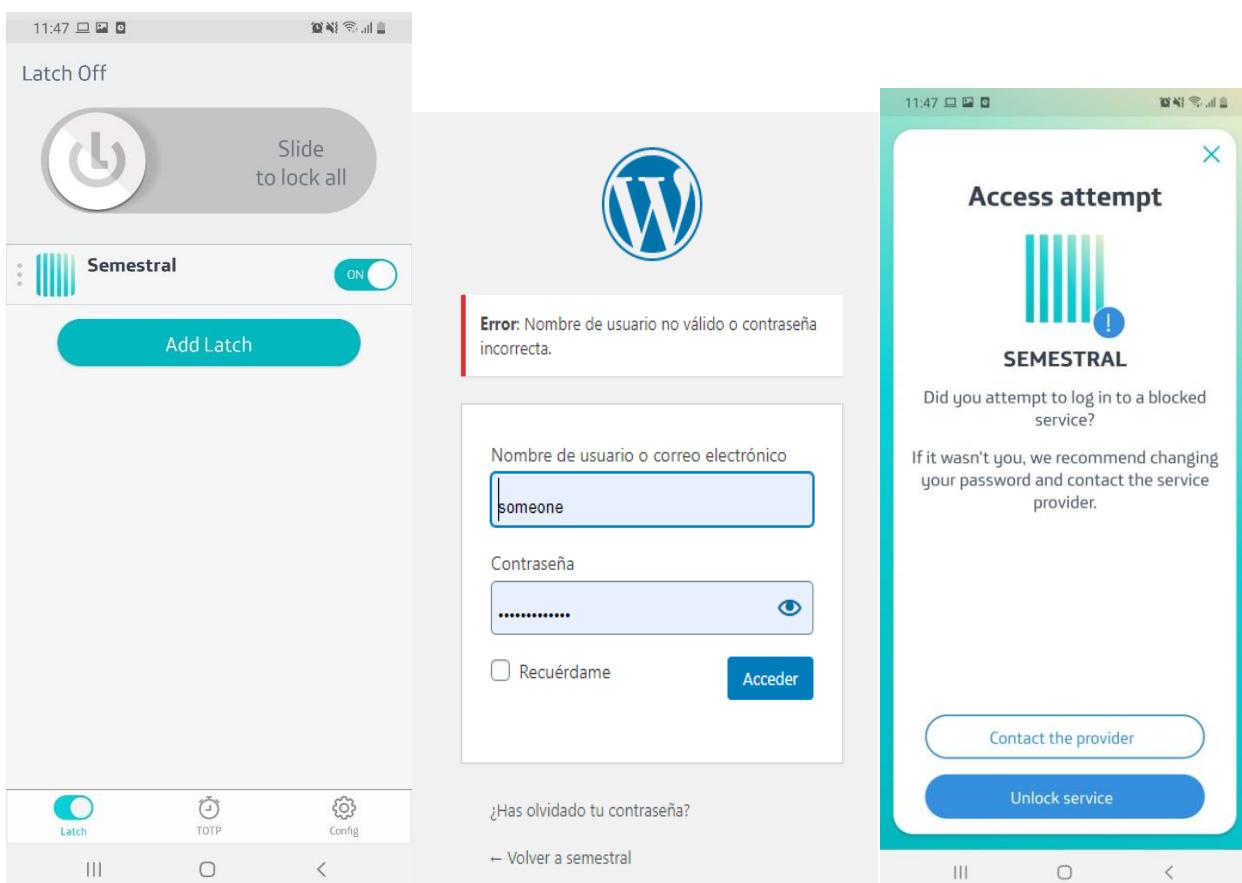
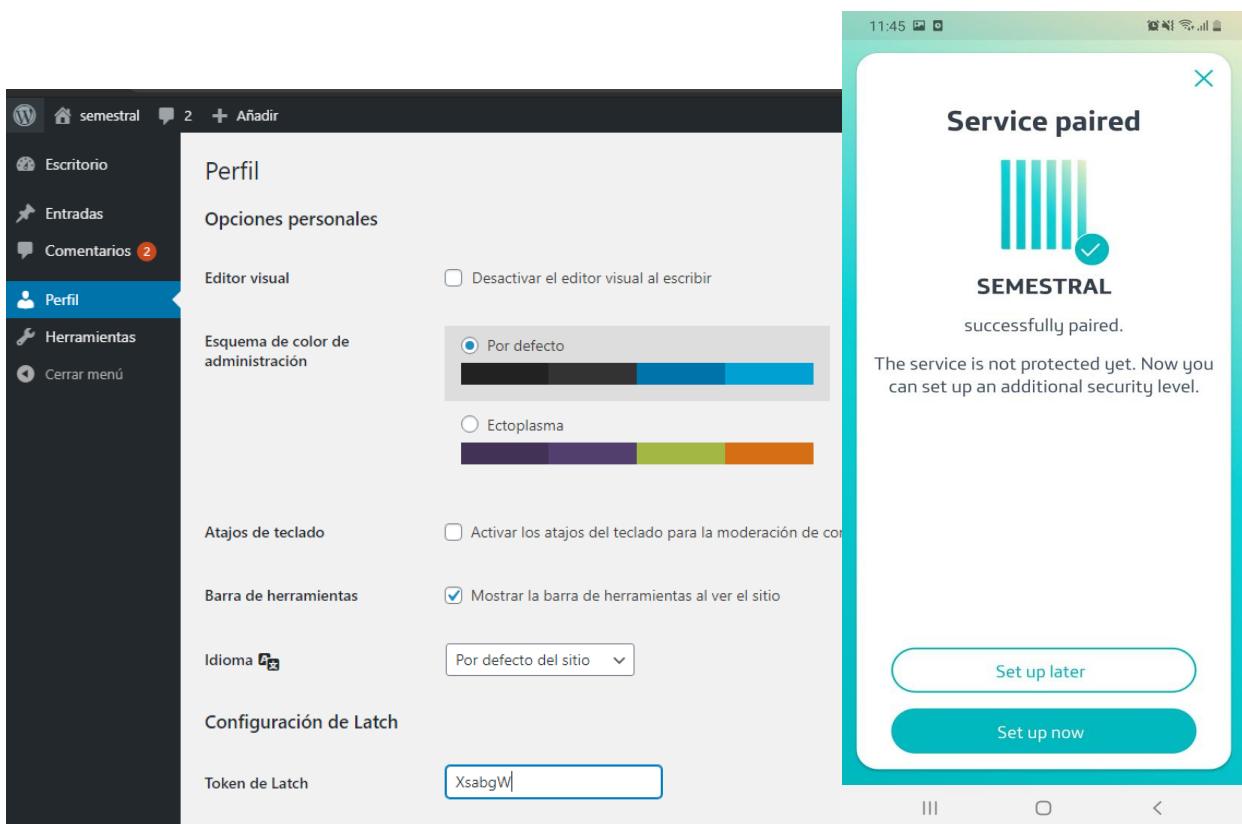
Guardar cambios



Entramos a la sección de ajustes de Latch, e insertamos el ID y Secreto que conseguimos al añadir la aplicación a nuestra cuenta de Latch.

Para probar el plugin hay que descargar la aplicación móvil.

Accedemos, añadimos un nuevo Latch y generamos un código. El código lo introducimos en la sección de perfil de un usuario, bajo "token de Latch". Si activamos Latch en la aplicación, evita que alguien se



Configuración de latch en SSH

Primero hay que descargar las librerías necesarias:

```
semestral@semestral:~$ sudo apt-get install gcc make
semestral@semestral:~$ sudo apt-get install libpam0g-dev libcurl4-openssl-dev libssl-dev
```

Descargamos el plugin desde el github

```
semestral@semestral:~$ git clone https://github.com/ElevenPaths/latch-plugin-unix.git
Cloning into 'latch-plugin-unix'...
remote: Enumerating objects: 2, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 555 (delta 0), reused 0 (delta 0), pack-reused 553
Receiving objects: 100% (555/555), 508.10 KiB | 1.55 MiB/s, done.
Resolving deltas: 100% (237/237), done.
semestral@semestral:~$
```

Instalamos el plugin

```
semestral@semestral:~$ cd latch-plugin-unix
semestral@semestral:~/latch-plugin-unix$ ./configure prefix=/usr sysconfdir=/etc && make && sudo make install
```

Editamos el archivo de config de latch y ponemos nuestro ID y llave secreta como en la configuración de latch de wordpress

```
GNU nano 2.9.3                               /etc/latch/latch.conf                         Modified

#
# Configuration file for the latch UNIX plugin
#
# Identify your Application
# Application ID value
#
app_id = 8aKkuN4XzEqK3NnM7vEL

# Secret key value
#
secret_key = Kyv8xvxyeEVkDtesQYkFPNgXurZRErKFWNDzDjp

# Latch host value
#
latch_host = https://latch.elevenpaths.com

#
^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos
^X Exit         ^R Read File     ^\ Replace       ^U Uncut Text    ^T To Spell     ^_ Go To Line
```

Movemos el archivo pamLatch.so al directorio /lib*/security

```
semestral@semestral:~/latch-plugin-unix$ sudo mv /usr/lib/pamLatch.so /lib*/security
```

Hacemos la configuración del plugin editando el archivo /etc/pam.d/sshd e introducimos la línea a continuación

```
GNU nano 2.9.3                               /etc/pam.d/sshd                         Modified

# PAM configuration for the Secure Shell service

# Standard Unix authentication.
@include common-auth

# Disallow non-root logins when /etc/nologin exists.
account    required    pam_nologin.so

# Uncomment and edit /etc/security/access.conf if you need to set complex
# access limits that are hard to express in sshd_config.
# account    required    pam_access.so

# Standard Unix authorization.
@include common-account
auth required pamLatch.so config=/etc/latch/latch.conf accounts=/etc/latch/latch.accounts
# SELinux needs to be the first session rule. This ensures that any
# lingering context has been cleared. Without this it is possible that a
# module could execute code in the wrong domain.
session [success=ok ignore=ignore module_unknown=ignore default=bad]          pam_sselinux.so close

^G Get Help      ^O Write Out      ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos
^X Exit         ^R Read File      ^\ Replace       ^U Uncut Text    ^T To Spell     ^ Go To Line
```

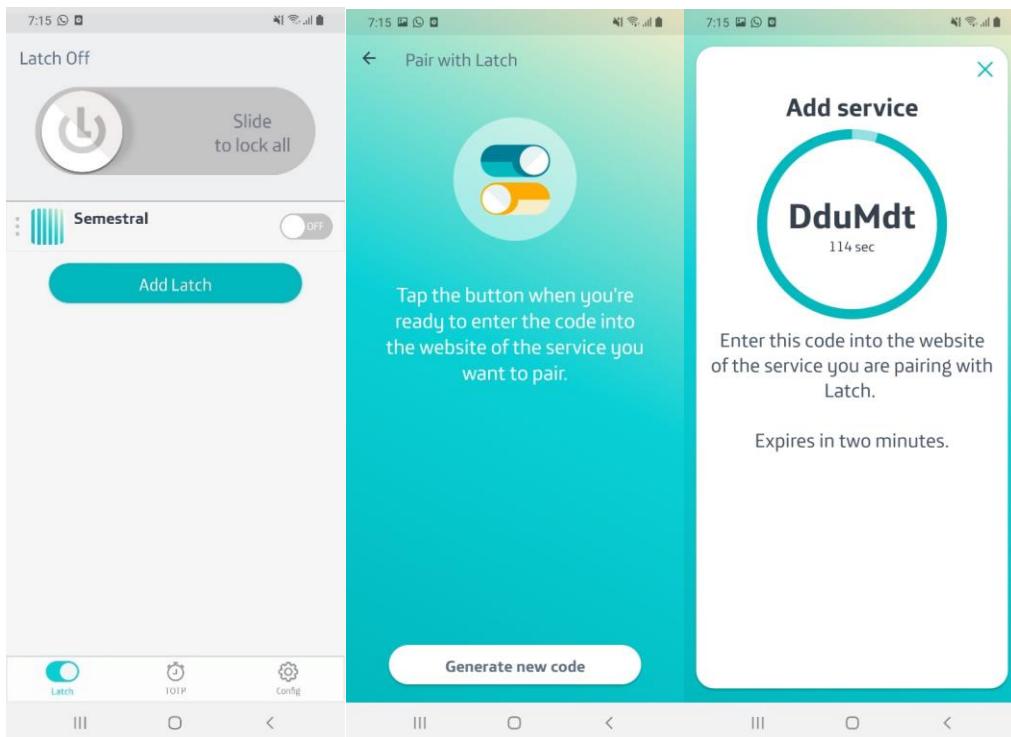
Editamos el archivo /etc/ssh/sshd_config

Nos aseguramos de que las próximas líneas están de la próxima forma:

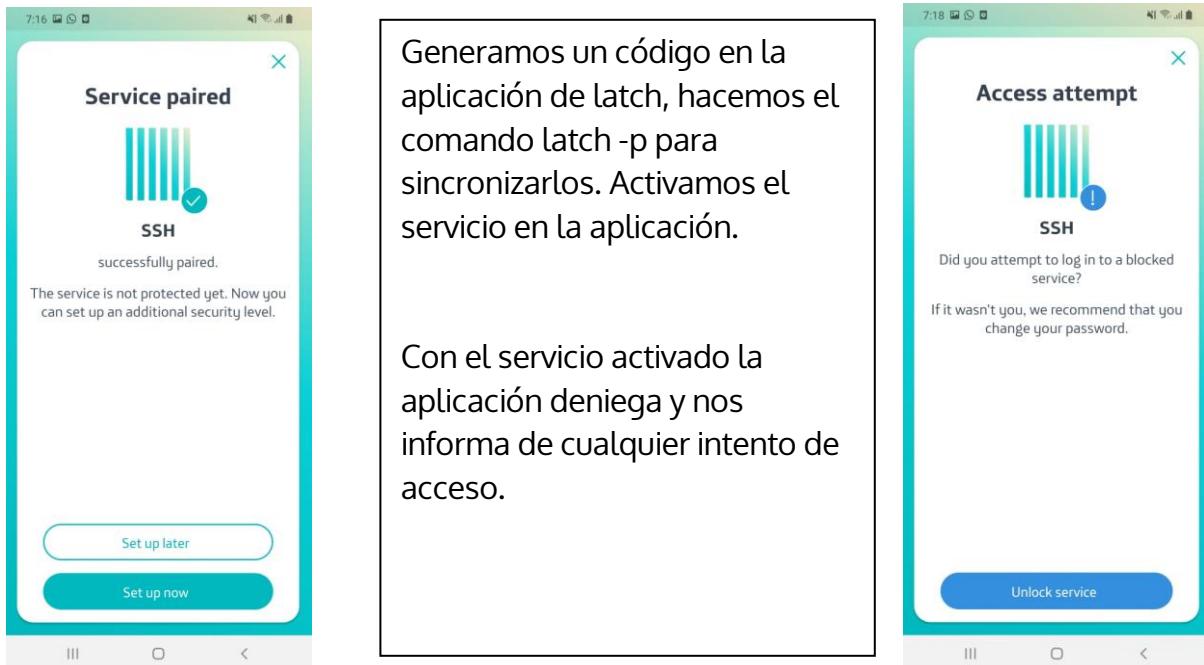
- ChallengeResponseAuthentication yes
- PasswordAuthentication no
- UsePAM yes

Reiniciamos el servicio ssh con: sudo service ssh restart

```
semestral@semestral:~/latch-plugin-unix$ latch -p DduMdt
Account successfully paired to the user semestral
```



```
semestral@semestral:~$ latch -p DduMdt
```



```
[+] login as: semestral
[+] semestral@13.90.43.38's password:
[+] Access denied
[+] semestral@13.90.43.38's password:
```

PANEL DE ADMINISTRACIÓN WEBMIN

Comience actualizando la lista de paquetes e instalando las dependencias:

```
$ sudo apt update
```

```
$ sudo apt install software-properties-common apt-transport-https wget
```

```
semestral@semestral:~$ sudo apt install software-properties-common apt-transport-https wget
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.96.24.32.13).
wget is already the newest version (1.19.4-lubuntu2.2).
wget set to manually installed.
apt-transport-https is already the newest version (1.6.12ubuntu0.1).
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
```

A continuación, importe la clave GPG de Webmin con el siguiente comando wget :

```
$ wget -q http://www.webmin.com/jcameron-key.asc -O- | sudo apt-key add -
```

```
semestral@semestral:~$ wget -q http://www.webmin.com/jcameron-key.asc -O- | sudo
  apt-key add -
OK
```

Y habilite el repositorio de Webmin escribiendo:

```
$ sudo add-apt-repository "deb [arch=amd64] http://download.webmin.com/down
```

```
semestral@semestral:~$ sudo add-apt-repository "deb [arch=amd64] http://download
.webmin.com/download/repository sarge contrib"
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:5 https://security.ubuntu.com/ubuntu bionic-security InRelease
Ign:6 http://download.webmin.com/download/repository sarge InRelease
Get:7 http://download.webmin.com/download/repository sarge Release [16.9 kB]
Get:8 http://download.webmin.com/download/repository sarge Release.gpg [173 B]
Get:9 http://download.webmin.com/download/repository sarge/contrib amd64 Packages [1390 B]
Fetched 18.4 kB in 1s (22.6 kB/s)
Reading package lists... Done
```

Instale la Última versión de Webmin escribiendo:

```
$sudo apt install webmin
```

```
semestral@semestral:~$ sudo apt install webmin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  apt-show-versions libapt-pkg-perl libauthen-pam-perl libio-pty-perl
  libnet-ssleay-perl perl-openssl-defaults unzip
Suggested packages:
  zip
The following NEW packages will be installed:
  apt-show-versions libapt-pkg-perl libauthen-pam-perl libio-pty-perl
  libnet-ssleay-perl perl-openssl-defaults unzip webmin
0 upgraded, 8 newly installed, 0 to remove and 14 not upgraded.
Need to get 29.9 MB of archives.
After this operation, 311 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 perl-openssl-defa
ults amd64 3build1 [7012 B]
```

Ajuste el cortafuegos

Por defecto, Webmin escucha las conexiones en el puerto 10000 en todas las interfaces de red.

Si su servidor ejecuta un firewall UFW, deberá abrir el puerto Webmin.

Para permitir el tráfico en el puerto, 10000ejecute el siguiente comando:

```
sudo ufw allow 10000/tcp
```

```
semestral@semestral:~$ sudo ufw allow 10000/tcp
Rules updated
Rules updated (v6)
```

Ahora actualizamos la contraseña

```
sudo /usr/share/webmin/changepass.pl /etc/webmin root Sem
estr@l2020
```

```
semestral@semestral:~$ sudo /usr/share/webmin/changepass.pl /etc/webmin root Sem
estr@l2020
Updated password of Webmin user root
semestral@semestral:~$
```

Por último, agregamos en azure redes el puerto 10000

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción	
300	⚠ SSH	22	TCP	Cualquiera	Cualquiera	✓ Permitir	...
310	80	80	Cualquiera	Cualquiera	Cualquiera	✓ Permitir	...
320	443	443	Cualquiera	Cualquiera	Cualquiera	✓ Permitir	...
330	Port_10000	10000	Cualquiera	Cualquiera	Cualquiera	✓ Permitir	...
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	✓ Permitir	...
65001	AllowAzureLoadBalancer...	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	✓ Permitir	...
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	✗ Denegar	...

Acceso a la interfaz web de Webmin

Ahora que Webmin está instalado en su sistema Ubuntu, abra su navegador favorito y escriba el nombre de host de su servidor o la dirección IP pública seguida del puerto de Webmin 10000:

https://your_server_ip_or_hostname:10000/

<https://semestral1.eastus.cloudapp.azure.com:10000/>

abrimos internet explorer o Edge

hacemos clic en **details** y luego en **Go on to the webpage**

This site is not secure

This might mean that someone's trying to fool you or steal any info you send to the server. You should close this site immediately.

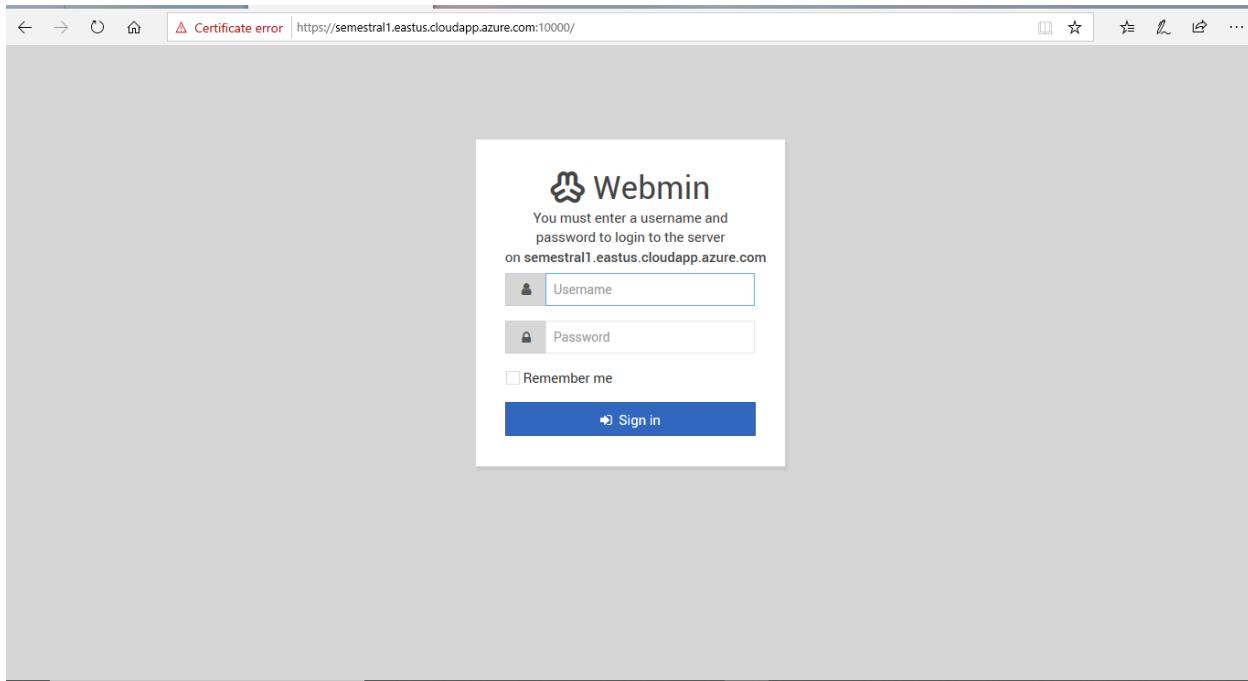
Go to your Start page

Details

Your PC doesn't trust this website's security certificate.
The hostname in the website's security certificate differs from the website you are trying to visit.

Error Code: DLG_FLAGS_INVALID_CA
DLG_FLAGS_SEC_CERT_CN_INVALID

[Go on to the webpage \(Not recommended\)](#)



Luego de introducir nuestras credenciales podemos observar el dashboard de webmin

System hostname	semestral.sfprtgiojtevb15gakwqj1pbh.bx.inter nal.cloudapp.net (10.0.2.4)	Operating system	Ubuntu Linux 18.04.4
Webmin version	1.953	Authentic theme version	19.52
Time on system	Sunday, July 26, 2020 3:01 AM	Kernel and CPU	Linux 5.3.0-1032-azure on x86_64
Processor information	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, 2 cores	System uptime	7 hours, 37 minutes
Running processes	126	CPU load averages	0.00 (1 min) 0.01 (5 mins) 0.04 (15 mins)
Real memory	598.33 MiB used / 1.37 GiB cached / 3.81 GiB total	Local disk space	4.79 GiB used / 31.92 GiB free / 36.71 GiB total
Package updates	14 package updates are available, of which 5 are security updates		

Actualización de Webmin

Para actualizar su instalación de Webmin cuando se publiquen nuevas versiones, puede usar el apt procedimiento de actualización normal del administrador de paquetes:

```
$ sudo apt update
$ sudo apt upgrade
```

GUARDAR CONTENEDOR EN REPOSITORIO

Primero nos logueamos

```
# sudo docker login
```

Entramos nuestras credenciales y nos enviara un mensaje como este

```
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/seimestral/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
seimestral@seimestral:~/seimestral$
```

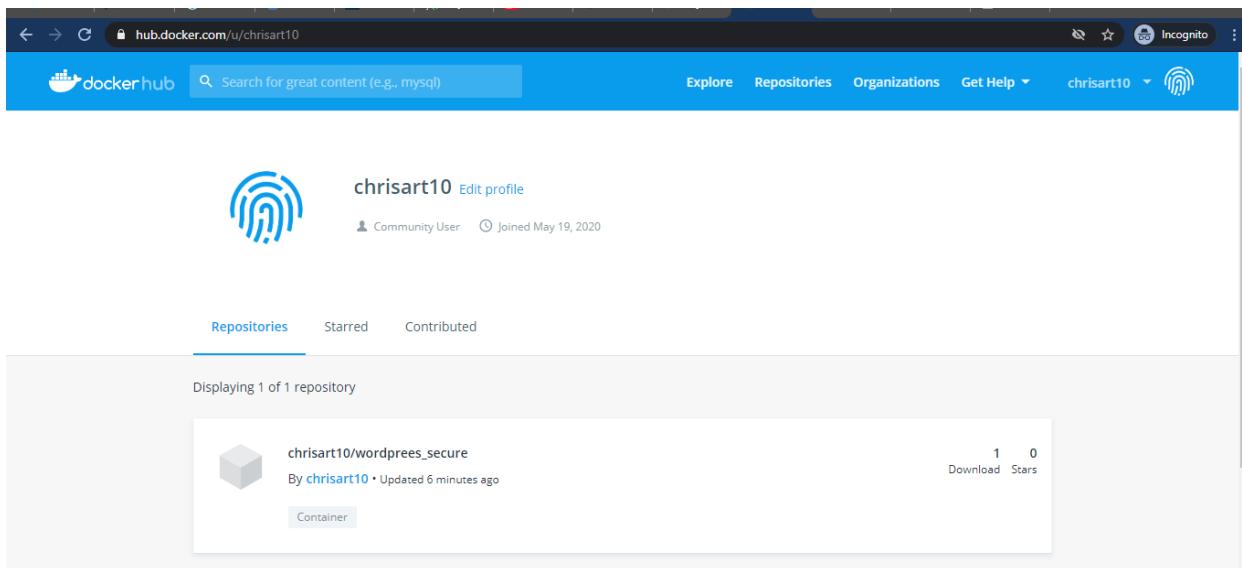
Luego hacemos un commit

Formato: sudo docker commit contenedor_a_guardar
nombredelusuario/nombre_del_contenedor

```
# sudo docker commit wordpress_1 chrisart10/wordpress_secure:1
```

```
seimestral@seimestral:~/seimestral$ sudo docker commit wordpress_1 chrisart10/wordpress_secure:1
sha256:d7e57f5766103f5f69b2060051ffc0a44412523701627calca2a9e2e8620901
seimestral@seimestral:~/seimestral$ sudo docker push chrisart10/wordpress_secure:1
The push refers to repository [docker.io/chrisart10/wordpress_secure]
laac436a2967: Pushed
c1220311208a: Pushed
d271a9bd322b: Pushed
49d405c756a8: Pushed
167656f68153: Pushed
e0b7e4096718: Pushed
e588250d07e8: Pushed
e311d0b20abb: Pushed
1d4c499d6d58: Pushed
82d7beecf3b0: Pushed
3b72b2c76867: Pushed
da080ee202c8: Pushed
8046a7edd0e7: Pushed
7d68f3060008: Pushed
32e2a84acc26: Pushed
5be098521317: Pushed
fe2160934099: Pushed
db497de51efd: Pushed
a5df928da0a7: Pushed
4e53c951cb3b: Pushed
13cbl4c2acd3: Pushed
1: digest: sha256:59e93c2141dled67fcc2ca484bee5cf067e5c93ded72cebacf5375cfb3ef00a5 size: 4713
seimestral@seimestral:~/seimestral$
```

Revisamos el repositorio en docker_hub



Listo ahora podemos usar la imagen en el compose y escalar.

Nota

1 - lo único que no se guardo fue la configuración del clickjacking. Solo repetimos el mismo paso de crear el archivo headers.php o hacemos un pull desde el repositorio.

Ajustamos el compose para poder escalarlo ya que un servicio no puede tener un nombre personalizado porque esto impide escalarlo lo eliminamos.

Github:

The screenshot shows the GitHub repository page for 'chrisart10/compose-wordpress-secure'. The repository is private. It has 1 branch ('master') and 0 tags. There are 13 commits in total, with the latest commit being 'Ubuntu v3' made 10 hours ago. The repository files listed are 'wordpress-files' (version v3), 'README.md' (updated 10 hours ago), and 'docker-compose.yml' (second commit, 10 hours ago). The 'README.md' file content is visible, listing four items under the heading 'compose-wordpress-secure': 1. Wordpress (scalable): apache hardening, mod security,mod evasive & mod quo. 2. Redis 3. Mysql 4. nginx-proxy

<https://github.com/chrisart10/compose-wordpress-secure>

2 - En reddit podemos usar el fichero (wp-config.php) del repositorio haciendo un pull.

3 - Abrir los puertos 80 y 443 en caso de no salir la página.

Ejecutamos el comando

```
$ sudo docker-compose up -d -- scale wordpress:1
```

Name	Command	State	Ports
compose_letsencrypt_1	/bin/bash /app/entrypoint. ...	Up	
compose_wordpress_1	docker-entrypoint.sh apach ...	Up	443/tcp, 80/tcp
compose_wordpress_2	docker-entrypoint.sh apach ...	Up	443/tcp, 80/tcp
compose_wordpress_3	docker-entrypoint.sh apach ...	Up	443/tcp, 80/tcp
mysql_db	docker-entrypoint.sh mysql	Up	0.0.0.0:8889->3306/tcp, 33060/tcp
nginx_proxy	/app/docker-entrypoint.sh ...	Up	0.0.0.0:443->443/tcp, 0.0.0.0:80->80/tcp
redis	docker-entrypoint.sh redis ...	Up	0.0.0.0:6379->6379/tcp

CONCLUSIONES

Christian Espinoza

La base utilizada en este proyecto es muy buena para crear una comunidad, crear una página de comercio electrónico, o para empezar a entrar en materia de cloud computing.

Con docker compose montar una aplicación es mucho más sencillo que escribir comando en la consola y también te permite visualizar mejor lo que vas a construir. la reutilización del archivo yml es sumamente genial porque puedes compartir esa estructura con otro tipo de aplicaciones o reutilizarlo.

Letsencrypt es una herramienta para certificarte de manera gratuita y esto es bueno para las personas que no tienen recurso o quieran reducir costos al montar una página web.

La parte que más me llama la atención del proyecto fue la del proxy inverso debido a los beneficios que ofrece comparado con un balanceador de carga en la capa 3 aunque esto puede ser debatible.

Me sorprendió las diversas maneras de crear una infraestructura y la documentación que contaba, pero tuvimos que elegir una base para facilitar más su construcción, ya que lo que se busca es ahorrar tiempo y comprender con una base sencilla, pero útil.

Con respecto a la implementación de herramienta que protegen contra ataques como dos, inyección sql tuvimos problemas con que se almacenara la configuración. Pero pudimos resolverlo guardando una imagen del contenedor en docker hub y algunas configuraciones en github y ahora se puede instalar más rápido y escalar.

Hardening de apache es una de las buenas prácticas que podemos encontrar para proteger nuestra aplicación web de atacantes como ddos, xss , clickjacking entre otros.

Jailkit es una herramienta útil al momento de manejarse entre varios usuarios y combinado con quotas sirve para limitar el uso o acceso al contenido en cada usuario en el servidor Ubuntu.

Fail2ban es una sorprendente herramienta que nos protege de los ataques de fuerza bruta en el Shell donde nos conectarnos de manera remota y combinado con la técnica de llave publica rsa aumenta la seguridad. Sin embargo, las herramientas latch nos permiten bloquear el acceso inclusive a nosotros mismo mediante una aplicación móvil lo cual crea una fuerte protección contra atacantes.

La implementación del webmin generó problemas debido a que faltaba agregar el puerto 10000 y para poder visualizar tuvimos que utilizar el navegador Edge. También hubo problemas con la mensajería en el fail2ban.

La aplicación latch resulta muy útil para bloquear los intentos de acceder a ya sea nuestra aplicación de wordpress como al servidor, esto aumenta la seguridad de nuestro sitio web como servidor.

Por último, pero no menos importante el banner con la herramienta figlet nos permite crear un mensaje de bienvenida para los usuarios que están involucrados en el desarrollo de trabajo.

Este proyecto me servirá mucho para mi portafolio y me gusto obtener este aprendizaje.

Adrián Franco

Verdaderamente considero que esta actividad es una de las más prácticas y útiles que he hecho en mi vida. Hoy en día Docker se ha establecido como el estándar de la industria, y tener el conocimiento de cómo utilizarlo va a ser útil en mi vida laboral y en mis ambiciones personales; sobre todo entendiendo el aspecto de seguridad.

Tener certificados de SSL en tu página web ya es esperado. Pero cuando uno es un desarrollador joven y sin experiencia es difícil encontrar los recursos necesarios para aprender a desarrollar como los profesionales. Ahora siento que tengo un entendimiento completo de desarrollo de aplicaciones web y de seguridad.

El concepto de平衡adores de carga es sumamente útil. Las nuevas plataformas te permiten hacerlo solo con un clic, pero todo tiene sus ventajas y desventajas. Saber cómo implementar tu propio balanceador de carga te da una ventaja sobre todas las demás personas y te abre opciones a la hora de desarrollar una aplicación.

Personalmente, yo me preguntaba cómo podía una aplicación resguardarse de ataques DDOS, ahora he visto que con MOD_Evasive y MOD_QOS se puede limitar la regularidad de interacción que tienen los usuarios con tu aplicación y el movimiento que hacen a través de las páginas.

Tener redundancias de seguridad en tu aplicación no es una decisión errónea. Implementarlas puede tomar tiempo extra, pero esto solo afecta al comienzo del desarrollo, a medida que el proyecto avanza, ya no hay que volver a tocar estas herramientas. Lo cual hace que la inversión de tiempo inicial valga la pena. Como existen múltiples tipos de ataque, existen múltiples tipos de herramientas que defienden.

Finalmente, estoy sumamente contento de haber aprendido estas herramientas y arquitectura web.

Luis Do Carmo

Resaltando principalmente el uso de las diferentes aplicaciones para armar nuestra web y protegerla, puedo considerar que el sistema de Wordpress nos facilita la parte de interfaz visual de nuestra página y la información que esta va a contener, es decir para tener un formato principal antes de tener que invertir en diseños y UI personalizados.

Docker compose fue la herramienta estrella de nuestro proyecto, ya que con este pudimos mantener fácilmente cada contenedor y la documentación oficial del mismo fue la que más nos indicó exactamente qué hacer y cuando hacerlo, luego de este pudimos ubicar las bases de datos es sus particiones sin ningún problema.

Destaco que esta actividad ha sido una de las más complejas para mi persona, ya que realmente no tenía ninguna experiencia de haber trabajado con contenedores anteriormente, agradezco la oportunidad de aprender acerca de esta aplicación sencilla y de gran utilidad, y como el uso de contenedores proporciona una organización y seguridad en nuestro sistema, donde a su vez pudimos emplear tanto balanceadores de carga como una segunda "imagen" de nuestra página en caso de que la primera fallara, todos estos aspectos que deben ser utilizados y tomados en cuenta en el ámbito profesional real.

En cuanto a la parte de seguridad, podemos demostrar que no solo el hardening de Apache es importante, sino que también es necesario implementar otros tipos de medidas como Fail2ban y Latch para asegurar nuestro servicio debido a lo común que es el funcionamiento de Apache.

Una de mis herramientas preferidas de la colección fue el Jailkit debido a que, aunque el atacante haya penetrado nuestras defensas este puede agarrarlo el medio de nuestro sistema y salvar de que el siga hacia los puntos más importantes de nuestro servicio (Por ejemplo, una base de datos de clientes, direcciones y links específicos)

Como aprendizaje este proyecto nos demostró que es importante la seguridad de nuestros servicios, y que realmente la aplicación de cada una de estas herramientas de seguridad no es tan difícil (y mucho menos costosas) de aplicar en nuestros servicios con tal de establecer un portal seguro donde podamos resguardar la información.

BIBLIOGRAFÍA

Información, Tutoriales y Software:

- Tatham, S. (2020). Download PuTTY - a free SSH and telnet client for Windows. Retrieved 17 May 2020, from <https://www.putty.org/>
- Docker Inc. (2020). Install Docker Engine on Ubuntu. Retrieved 17 May 2020, from <https://docs.docker.com/engine/install/ubuntu/>
- Tutorials Teachers. (2020). What is https. Retrieved 17 May 2020, from <https://www.tutorialsteacher.com/https/what-is-https>
- NGINX. (2020). What Is Load Balancing? How Load Balancers Work. Retrieved 17 May 2020, from <https://www.nginx.com/resources/glossary/load-balancing/>
- Rouse, M. (2020). ¿Qué es Docker? - Definición en WhatIs.com. Retrieved 17 May 2020, from <https://searchdatacenter.techtarget.com/es/definicion/Docker>
- Webempresa. (2020). WordPress y sus características principales. Retrieved 17 May 2020, from <https://www.webempresa.com/wordpress/que-es-wordpress.html>
- Auladell, G. (2017). ¿Qué es Redis?. Retrieved 17 May 2020, from <https://www.drauta.com/que-es-redis>
- IBM. (2020). IBM Knowledge Center, Certificados autofirmados. Retrieved 17 May 2020, from https://www.ibm.com/support/knowledgecenter/es/SSIGMP_1.0.0/igi/sql/install_config/c_adk_cert_self_signed_ins.htm
- Boucheron, B. (2018). How To Install WordPress with LAMP on Ubuntu 18.04 | DigitalOcean. Retrieved 17 May 2020, from <https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-lamp-on-ubuntu-18-04>
- Drake, M. (2018). How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 18.04 | DigitalOcean. Retrieved 17 May 2020, from <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04>
- Docker Inc. (2013). Use volumes. Retrieved 21 May 2020, from <https://docs.docker.com/storage/volumes/>
- Linuxize. (2019). How To Remove Docker Containers, Images, Volumes, and Networks. Retrieved 21 May 2020, from <https://linuxize.com/post/how-to-remove-docker-images-containers-volumes-and-networks/>
- Dietrich, E. (2019). How to Create a Docker Image From a Container | Scalyr. Retrieved 28 May 2020, from <https://www.scalyr.com/blog/create-docker-image/>

- Read the Docs. (2020). Step 4: Add additional Services –BilimEdtech Labs documentation. Retrieved 9 June 2020, from <https://labs.bilimedtech.com/cloud-computing/3/3.4.html>
- Qualys Inc. (2009). Qualys SSL Labs. Retrieved 9 June 2020, from <https://www.ssllabs.com/index.html>
- IMPERVA. What is Clickjacking | Attack Example | X-Frame-Options Pros & Cons | Imperva. Retrieved 10 July 2020, from <https://www.imperva.com/learn/application-security/clickjacking/>
- AVAST TEAM. (2016). Inyección SQL. Retrieved 9 July 2020, from <https://www.avast.com/es-es/c-sql-injection>
- PALO ALTO NETWORK. What is a denial of service attack (DoS) ?. Retrieved 10 July 2020, from <https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos>
- Soto, M. (2016). ¿Qué es XSS? Retrieved 12 July 2020, from <https://medium.com/@marvin.soto/qu%C3%A9-es-xss-b9330eedbc07>
- ModSecurity. (2004). ModSecurity Overview. Retrieved 10 July 2020, from <https://www.modsecurity.org/about.html>
- Linux Academy. (2020). Hands-On Lab: Configure Mod Evasive | Linux Academy. Retrieved 10 July 2020, from <https://linuxacademy.com/hands-on-lab/cb00db0a-5b53-4f3d-9e9b-a6b780bd3325/>
- Tyagi, R. (2018). Securing and Hardening an Apache Web Server | Lucideus Research. Retrieved 12 July 2020, from <https://blog.lucideus.com/2018/03/securing-and-hardening-apache-web.html>
- Sessink, O. (2013). Jailkit - chroot jail utilities. Retrieved 10 July 2020, from <https://olivier.sessink.nl/jailkit/>
- Fail2ban. (2016). Retrieved 10 July 2020, from https://www.fail2ban.org/wiki/index.php/Main_Page
- Cowan, J. (2012). FIGlet - hosted by PLiG. Retrieved 10 July 2020, from <http://www.figlet.org/>
- Shovon, S. (2019). How to use Quota on Ubuntu –Linux Hint. Retrieved 12 July 2020, from https://linuxhint.com/disk_quota_ubuntu/
- Sanso, R. (2016). Configurar Latch en WordPress. Retrieved 12 July 2020, from <http://www.rafelsanso.com/configurar-latch-en-wordpress/#:~:text=Qu%C3%A9%20es%20Latch,en%20nuestro%20panel%20de%20control.>
- Raiola Team. (2018). Manual completo de Webmin, Usermin y Virtualmin. Retrieved 15 July 2020, from <https://raiolanetworks.es/blog/webmin-usermin-virtualmin/#:~:text=Webmin%20es%20un%20panel%20de,f%63%A1ciles%20de%20usar%20y%20entender.>

Videos y Multimedia

- Canal: MyOnlineEdu.com, (2019) Duracion: 10 Min. 52 Seg. How to Install Docker on Ubuntu 18.04 LTS, from https://www.youtube.com/watch?v=W7ByS942UZA&list=LLA3hrN7tv4VGo3H7D_o-toUA&index=2&t=530s
- Strong password Generator (2006), utilizado para creación de contraseñas seguras from <https://strongpasswordgenerator.com/>
- Clase y explicación del profesor José Moreno (2020) <https://web.microsoftstream.com/video/2364369c-3437-422c-8f50-44ebbf659ee7?App=mteamsBot&refId=f.947829455068815927>
- Canal: Techworld with nana, (2019) duracion: 6Min. 02 Seg. Docker Volumes explained in 6 minute, from https://www.youtube.com/watch?v=p2PH_YPCsis
- Canal: programador novato (2019), Lista de Reproducción con 10 videos, Docker de principiante a experto, from https://www.youtube.com/playlist?list=PLCTD_CpMeFKTj_n9XY0vz9n6Asi-g0kRg
- Gite, V. (2018). Start / Stop and Restart Apache 2 Web Server Command - nixCraft. Retrieved 21 May 2020, from <https://www.cyberciti.biz/faq/star-stop-restart-apache2-webserver/>
- Canal: Pelado Nerd (2018), duración 12 min. 54 seg. VOLUMENES y PUERTOS en DOCKER! TUTORIAL! from https://www.youtube.com/watch?v=Gfl_ltu-eyw
- Canal: Talk2 Amareswaran (2019), duracion 7 Min. 47 Seg. Data Volume with MySQL Docker container, from https://www.youtube.com/watch?v=r-ggFM_Y_9U
- Canal: ProgrammingKnowledge (2015), duracion 13 min. 25 seg. How to Install and Configure Git and Repositories on GitHub on Ubuntu Linux from <https://www.youtube.com/watch?v=SwK2dPFXhpU&t=519s>
- Canal: JCGamingLab (2019), duracion 2 Min. 27 Seg. Linux / Ubuntu Basics - Custom Terminal Welcome Using Figlet, from <https://www.youtube.com/watch?v=VlaAXppIBok>

Ayuda

- How to determine Linux kernel architecture?. (2011). Retrieved 17 May 2020, from <https://unix.stackexchange.com/questions/12453/how-to-determine-linux-kernel-architecture>

- How do I see what packages are installed on Ubuntu Linux? - nixCraft. (2018). Retrieved 21 May 2020, from <https://www.cyberciti.biz/faq/apt-get-list-packages-are-installed-on-ubuntu-linux/>
- How to install tzdata on a ubuntu docker image?. (2019). Retrieved 21 May 2020, from <https://serverfault.com/questions/949991/how-to-install-tzdata-on-a-ubuntu-docker-image>
- How do I edit a file after I shell to a Docker container?. (2015). Retrieved 21 May 2020, from <https://stackoverflow.com/questions/30853247/how-do-i-edit-a-file-after-i-shell-to-a-docker-container>
- How to grant all privileges to root user in MySQL 8.0. (2018). Retrieved 28 May 2020, from <https://stackoverflow.com/questions/50177216/how-to-grant-all-privileges-to-root-user-in-mysql-8-0/50197630>
- Support HTTPS · Issue #46 · docker-library/wordpress. (2015). Retrieved 9 June 2020, from <https://github.com/docker-library/wordpress/issues/46>
- Docker-compose cheatsheet. (2018). Retrieved 9 June 2020, from <https://devhints.io/docker-compose>
- Van Reems, R. (2013). Fail2ban - Community Help Wiki. Retrieved 16 July 2020, from <https://help.ubuntu.com/community/Fail2ban>
- Jevtic, G. (2019). mod_evasive on Apache: Install & Configure to Defend DDoS Attacks. Retrieved 15 July 2020, from <https://phoenixnap.com/kb/apache-mod-evasive>
- Rapid7. (2017). How to Configure ModEvasive with Apache on Ubuntu Linux. Retrieved 13 July 2020, from <https://blog.rapid7.com/2017/04/09/how-to-configure-modevasive-with-apache-on-ubuntu-linux/>
- Kohead. (2012). KoHead/mod_evasive. Retrieved 15 July 2020, from https://github.com/KoHead/mod_evasive/blob/master/test.pl
- Buchbinder, P. (2020). mod_qos - Denial of Service Defense. Retrieved 15 July 2020, from <http://mod-qos.sourceforge.net/dos.html>
- 1&1IONOS INC. (2020). Using the Apache Module mod_evasive - 1&1 Hosting (US). Retrieved 14 July 2020, from <https://www.ionos.com/community/server-cloud-infrastructure/apache/using-the-apache-module-mod-evasive/>
- Monzilla media. (2020). Increase Security with X-Security Headers | .htaccess made easy. Retrieved 16 July 2020, from <https://htaccessbook.com/increase-security-x-security-headers/>
- Elevenpaths. (2015). Retrieved 14 July 2020, from https://latch.elevenpaths.com/www/public/documents/resources/plugin_manuals/en/Wordpress.pdf

- Various Users. (2019). .htaccess headers being ignored by Apache. Retrieved 16 July 2020, from <https://stackoverflow.com/questions/53259981/htaccess-headers-being-ignored-by-apache>
- Jackson, B. (2019). X-Frame-Options - How to Combat Clickjacking - KeyCDN. Retrieved 17 July 2020, from <https://www.keycdn.com/blog/x-frame-options>
- Various Users. (2017). jailkit-dev - prolem with raspberrypi 3. Retrieved 17 July 2020, from <http://nongnu.13855.n7.nabble.com/prolem-with-raspberrypi-3-td220074.html>
- Various users. (2017). Error Permission denied (publickey) when I try to ssh | DigitalOcean. Retrieved 17 July 2020, from <https://www.digitalocean.com/community/questions/error-permission-denied-publickey-when-i-try-to-ssh>
- Kishore, S. (2014). How to create a jailed ssh user with Jailkit on Debian Wheezy. Retrieved 18 July 2020, from <https://www.howtoforge.com/how-to-create-a-jailed-ssh-user-with-jailkit-on-debian-wheezy>
- Moon, S. (2013). Install Jailkit on Ubuntu/Debian. Retrieved 18 July 2020, from <https://www.binarytides.com/install-jailkit-ubuntu-debian/>
- Moon, S. (2013). Setup a jailed shell with jailkit on ubuntu. Retrieved 18 July 2020, from <https://www.binarytides.com/setup-jailed-shell-jailkit-ubuntu/>
- Boucheron, B. (2019). How To Set Filesystem Quotas on Ubuntu 18.04 | DigitalOcean. Retrieved 19 July 2020, from <https://www.digitalocean.com/community/tutorials/how-to-set-filesystem-quotas-on-ubuntu-18-04>
- Linuxize. (2019). How to Install Webmin on Ubuntu 18.04. Retrieved 19 July 2020, from <https://linuxize.com/post/how-to-install-webmin-on-ubuntu-18-04/#:~:text=Webmin%20is%20an%20open%2Dsource,Ubuntu%2018.04%20server>.
- Obtención de certificados con let's encrypt. (2019, May 16). Adictos al trabajo. <https://www.adictosaltrabajo.com/2016/07/21/obtencion-de-certificados-con-lets-encrypt/>