

Seguridad en Aplicaciones Webs

Trabajo Práctico Integrador

“Escenarios Vulnerables”

Grupo N° 6

APELLIDO Y NOMBRE	LEGAJO N°	EMAIL CONTACTO
Biondi Bonomini, Marco Fernando	153.052-5	marco.fbb@gmail.com
Tolaba, Emiliano	152.413-6	emi.tolaba95@gmail.com
Kaynar, Martin	146.475-9	tinchokaynar@gmail.com

Docente: Ing. Gonzalo Vilanova

Introducción	4
Preparando el laboratorio	5
Servidor Backend	5
Servidor proxy - CloudFlare	5
Vincular dominio a la IP	7
Explotando vulnerabilidades	8
Local File Inclusion	8
Bypass Cloudflare	9
Acceso a la BD	10
Bypass Restrict Login IP	12
Session Hacking	14
Bypass restrict upload only image mime type	16
Infectar servidor	19
Anexo I	21
Crear maquina virtual backend	21
Instalar sistema operativo	21
Montar Imagen	21
Instalar panel de control hosting	25
Configurando VestaCP	29
Crear Dominio	29
Crear base de datos	30
Instalar Wordpress	31
Instalar Plugins Wordpress	34
Crear maquina virtual proxy - Cloudflare	37
Instalar sistema operativo	37
Instalar Varnish	37
Anexo II	38
Datos servidor backend	38
Datos servidor proxy inverso	39
Anexo III	40
Plugins	40
Restrict login to IP	40
Thumbnail img	40
Upload Media IMG	41
Disable WPadmin Menu	41

Introducción

El sitio web **elcronista-reportero.com** utiliza como CMS la plataforma WordPress, la cual fue desarrollada en PHP y que utiliza MySQL como motor de base de datos. El portal para ahorrar ancho de banda y esconder la IP del servidor utiliza **CloudFlare** como proxy inverso. El propietario del sitio web instaló diversos plugins, entre ellos:

- **Thumbnail generator**, que dada una url o ruta de imagen vía parámetro GET, la obtiene y re-dimensiona.
- **WP login restrict**, que restringe la acción iniciar sesión a una única IP.
- **Upload Media IMG** para usuarios logueados con permisos de administrador, deja subir archivos únicamente de imagen y devuelve su URL directa.

Mediante una vulnerabilidad del plugin **thumbnail generator**, vamos a obtener el archivo de configuración de WordPress, llamado *wp-config.php*, en donde se encuentran las credenciales de acceso a la base de datos.

Debido a una mala configuración de **Cloudflare**, obtendremos la IP original del servidor. Dado que tenemos las credenciales de la base de datos, la IP del servidor y que el mismo no tiene cerrado el **puerto 3306**, que permite conexiones remotas a la base de datos, podremos acceder a la misma. La tabla **wp_options** está en modo READ-ONLY impidiendo editar sus valores.

En la base de datos reemplazamos la contraseña del usuario administrador y buscamos cual es la IP permitida para login del plugin **WP login restrict**.

Sabiendo las credenciales de login del usuario administrador y la IP autorizada para hacer login, utilizamos una vulnerabilidad del plugin **WP login restrict** que simplemente compara que la IP obtenida sea igual a la autorizada para dejar loguear. En donde la IP obtenida comprueba si existe la cabecera creada por cloudflare de `HTTP_CF_CONNECTING_IP` en caso de existir utiliza dicha IP para comparar, debido a la falta de seguridad del plugin creamos un pequeño software que envía dicha cabecera con la IP autorizada y obtenemos la **cookie con la sesión ID** de WordPress.

Dicha cookie la pondremos en nuestro navegador para estar logueados como administrador. Utilizando una vulnerabilidad del plugin **Upload Media IMG** el cual no deja subir archivos sin formato de imagen, podemos realizar un **bypass** a dicha medida de seguridad cambiando el TYPE MIME del archivo. Realizaremos este bypass para subir una **shell** y obtener el control total del hosting.

Preparando el laboratorio

Para realizar este laboratorio necesitamos las máquinas virtuales “VMwp” y “VMcloudflare”. Dichas máquinas vienen preconfiguradas para que el enfoque del laboratorio sea acotado al desarrollo del enunciado y no a la instalación y configuración de los requerimientos previos.

En el anexo I se dejan asentados los pasos realizados para la creación de las máquinas virtuales. En el anexo II se encuentra la información de credenciales de acceso y datos propios de la configuración. En el anexo III, por último, se encuentra una breve descripción de cada uno de los plugins utilizados en la plataforma.

Servidor Backend

El servidor backend será el encargado de ejecutar el servidor HTTP apache, el motor de base de datos MySQL y la ejecución del lenguaje de programación PHP del CMS WordPress.

La máquina virtual tiene el sistema operativo **Centos 7 64bits minimal**, y utiliza VestaCP como panel de control para hosting web.

Se deberá montar la máquina virtual y obtener la IP mediante la ejecución del siguiente comando:

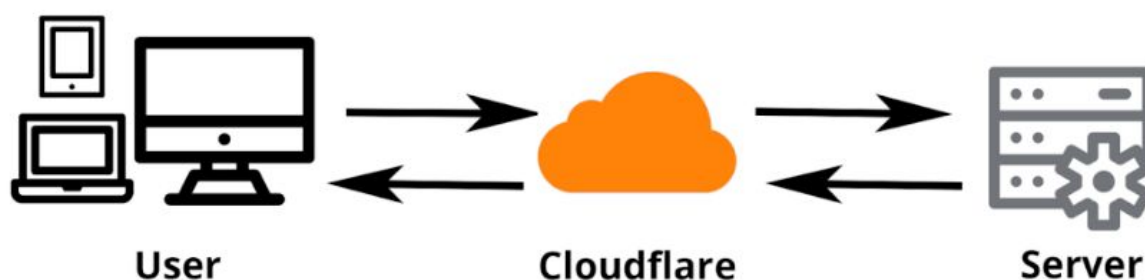
```
# Comando para saber la IP  
ip addr
```

Anotar la dirección IP porque la usaremos más adelante, en el documento utilizaremos la nomenclatura “IP-VMwp” para hacer referencia a ella.

Servidor proxy - CloudFlare

Cloudflare brinda servicios de red de entrega de contenido, mitigación de DDoS, seguridad de Internet y servicios de servidores de nombres de dominio distribuidos.

En nuestro ejemplo, el propietario del sitio web utiliza Cloudflare para ahorrar ancho de banda en la entrega de contenido estático y esconder la dirección IP del servidor backend a los visitantes.



El visitante se conectara a Cloudflare y éste, con su IP, se conecta a nuestro servidor enviando por campos de la cabecera HTTP la dirección real del visitante.

La configuración de CloudFlare está hecha de la siguiente manera:

Type	Name	Content	TTL	Proxy status
A	www	168.192.1.100	Auto	Proxied
A	ftp	168.192.1.100	Auto	DNS only
A	elcronista-report...	168.192.1.100	Auto	Proxied

Debido a que el laboratorio está pensado para hacerse sin conexión a internet, debemos simular el servidor CloudFlare. Utilizaremos el software Varnish Cache como proxy inverso para esconder nuestra IP y configuraremos Varnish Cache para incluir en la cabecera HTTP el campo `CF-CONNECTING-IP` con la IP real del visitante.

Se deberá montar la máquina virtual y obtener la IP mediante la ejecución del siguiente comando:

```
# Comando para saber la IP
ip addr
```

Anotar la dirección IP porque la usaremos más adelante, en el documento utilizaremos la nomenclatura "IP-VMcloudflare" para hacer referencia a ella.

Vincular dominio a la IP

Debido a que no tenemos los derechos de propiedad del dominio “elcronista-reportero.com”, y que el laboratorio está pensado para ser realizado de forma offline, hay que simular la resolución DNS de forma local.

Lo que realizaremos es apuntar el dominio “elcronista-reportero.com” y algunos subdominios a una determinada IP mediante la edición del archivo “hosts”. Dicho archivo se utiliza para mapear nombres de dominios a direcciones IP de forma estática y con máxima prioridad, es decir, si tenemos acceso a internet y existe una resolución DNS para el dominio, el sistema operativo primero se fija en nuestro archivo hosts y utiliza dicha regla reemplazando a las del servidor DNS.




Para editar el archivo “host”, realizamos los siguientes pasos

1. Abrir el notepad con permisos administrador
2. En el tab “Archivo” click en “Abrir”
3. En el cuadro “Abrir” en nombre de archivo ponemos la ruta “C:\Windows\System32\Drivers\etc\hosts”
4. Click botón abrir
5. Agregamos en la parte inferior del archivo

```
IP-VMcloudflare elcronista-reportero.com
IP-VMcloudflare www.elcronista-reportero.com
IP-VMwp ftp.elcronista-reportero.com
```

6. Guardamos el archivo

Este paso simula tener el dominio “elcronista-reportero.com” con los NS Server de Cloudflare, y en Cloudflare tener configurado los registros “www.elcronista-reportero.com” y “elcronista-reportero.com” con Proxy inverso activado y “ftp.elcronista-reportero.com” con Proxy inverso desactivado.

Type	Name	Content	TTL	Proxy status
A	www	168.192.1.100	Auto	 Proxied ✕
A	ftp	168.192.1.100	Auto	 DNS only ✕
A	elcronista-report...	168.192.1.100	Auto	 Proxied ✕

CloudFlare - Configuración

En la imagen anterior la IP 168.192.1.100 es la IP-VMwp, se utiliza en los tres registros, pero cuando activamos el Proxy inverso (nubecita naranja), Cloudflare hace el cambio internamente para que la IP sea la de su proxy inverso y dicho proxy inverso sepa que la IP destino es 168.192.1.100

Explotando vulnerabilidades

A continuación, explicaremos de manera concisa la explotación de las vulnerabilidades.

Local File Inclusion

Mediante una vulnerabilidad del plugin “**Thumbnail img**” logramos descargar el archivo **wp-config.php** del sitio web. En dicho archivo se alojan las credenciales de la base de datos con la cual el sitio web funciona.

```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define( 'DB_NAME', 'admin_wp' );  
  
/** MySQL database username */  
define( 'DB_USER', 'admin_wp' );  
  
/** MySQL database password */  
define( 'DB_PASSWORD', '123456' );  
  
/** MySQL hostname */  
define( 'DB_HOST', 'localhost' );
```

Dicha vulnerabilidad se da porque el plugin no sintetiza los datos de entrada suministrados por el usuario, no limita las carpetas donde el plugin puede localizar imágenes, ni corrobora que el archivo destino sea una imagen.

Para descargar el archivo wp-config.php, y explotar la vulnerabilidad, ingresamos en el navegador a la siguiente URL:

<http://elcronista-reportero.com/wp-content/plugins/thumbnail-img/img.php?data=../../wp-config.php>

Dicha petición hace ejecutar la función **file_get_contents** nativa de PHP que, dado una ruta de archivo / URL, devuelve el contenido.

```
file_get_contents("../../wp-config.php");
```


El plugin muestra por pantalla dicho contenido, permitiéndonos obtener el contenido del archivo deseado.

Para arreglar esta vulnerabilidad algunas medidas a tener en cuenta son

- Formato de entrada preestablecido, limpiando rutas relativas
- Limitar las carpetas donde el plugin puede recolectar imágenes
- Corroborar que el archivo solicitado para comprimir es realmente una imagen

Como resultado de esta vulnerabilidad, obtuvimos las credenciales de la base de datos.

Bypass Cloudflare

En el paso anterior, obtuvimos las credenciales de la base de datos, pero para conectarnos a la misma, necesitamos la IP del servidor destino y el puerto donde corre el motor de la base de datos.

Debido a que el sitio web utiliza Cloudflare para, entre otras cosas, esconder la dirección IP real del servidor donde se aloja el sitio web y que sólo permite conexiones mediante protocolos HTTP y HTTPS, se nos hace imposible establecer conexión con el motor de la base de datos de forma remota.

Para poder conectarnos con el motor de la base de datos necesitamos conocer la IP real del servidor y el puerto donde opera el motor de la base de datos para así poder saltar los servidores de Cloudflare.

El salto de los servidores de Cloudflare se logra por un error muy común de los usuarios del mismo, que dejan la configuración predeterminada.

Cuando se introduce un nuevo dominio para ser utilizado en una plataforma, Cloudflare realiza un escaneo por defecto de los Registros DNS actuales para incorporarlos de forma automática en la plataforma, y de este modo evitar que el usuario tenga que configurarlos uno por uno. Habilitando por defecto el sistema de proxy inverso únicamente para los registros `www.web.com` y `web.com` (es decir, con y sin `www`), todos los demás registros quedan con Cloudflare apagado por defecto, dejando visible la ip del servidor en registros frecuentes como *ftp.web.com*, *mail.web.com*, *cpanel.web.com*, entre otros.

Para explotar dicha vulnerabilidad[1], realizamos el siguiente PING:

```
# realizamos el ping
ping ftp.elcronista-reportero.com

# el resultado obtenido fue el siguiente:
PING ftp.elcronista-reportero.com (192.168.78.128) 56(84) bytes of data.
64 bytes from ftp.elcronista-reportero.com (192.168.78.128): icmp_seq=1 ttl=64 time=0.397 ms
64 bytes from ftp.elcronista-reportero.com (192.168.78.128): icmp_seq=2 ttl=64 time=0.473 ms
64 bytes from ftp.elcronista-reportero.com (192.168.78.128): icmp_seq=3 ttl=64 time=0.406 ms
64 bytes from ftp.elcronista-reportero.com (192.168.78.128): icmp_seq=4 ttl=64 time=0.447 ms
64 bytes from ftp.elcronista-reportero.com (192.168.78.128): icmp_seq=5 ttl=64 time=0.543 ms
^C
--- ftp.elcronista-reportero.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4095ms
rtt min/avg/max/mdev = 0.397/0.453/0.543/0.054 ms
```

[1] Considerado como vulnerabilidad porque el usuario quería ocultar la IP real del servidor, sin haber considerado los demás registros que no pasan por Cloudflare de forma predeterminada al inicializar la configuración.

Para solucionar dicha vulnerabilidad hay que borrar todo registro DNS que no se utiliza, y habilitar Cloudflare en los que correspondan.

Finalizado este paso, hemos obtenido la IP real del servidor VMwp

Acceso a la BD

Para acceder a la base de datos, podemos utilizar cualquier Software de gestión de base de datos que cuente con la posibilidad de conexión remota (Por ejemplo: HeidiSQL, Adminer).

Ya disponemos de las credenciales de la base de datos y la IP real del servidor, por lo que resta obtener el puerto escucha de MySQL. Por defecto, el puerto es 3306, valor con el cual realizamos la prueba (utilizando, en nuestro caso, Adminer [1]).

Login

Motor de base de datos	MySQL ▼
Servidor	IP-REAL-SERVIDOR
Usuario	usuario
Contraseña	*****
Base de datos	base_de_datos

☒ Guardar contraseña

Logramos ingresar exitosamente, por lo que concluimos que el dueño del sitio web no modificó el puerto por defecto y tampoco denegó el acceso al mismo desde el exterior del servidor.

Notamos que todas las tablas tienen permisos de lectura y escritura, a excepción de la tabla “wp_options”, que está en modo solo lectura.

Utilizando el acceso a la base de datos, vamos a modificar la contraseña del usuario administrador para poder ingresar a el tablero de control de WordPress. Para realizar esto, seguimos los siguientes pasos

1- Ir a los registros de la tabla “wp_users”

[Comando SQL](#) [Importar](#)
[Exportar](#) [Crear tabla](#)

[registros wp_commentmeta](#)
[registros wp_comments](#)
[registros wp_links](#)
[registros wp_options](#)
[registros wp_postmeta](#)
[registros wp_posts](#)
[registros wp_termmeta](#)
[registros wp_terms](#)
[registros wp_term_relationships](#)
[registros wp_term_taxonomy](#)
[registros wp_usermeta](#)
[registros wp_users](#)

2- Click en modificar, en la fila del usuario administrador

Mostrar: wp_users

[Visualizar contenido](#) [Mostrar estructura](#) [Modificar tabla](#) [Nuevo Registro](#)

Mostrar

Condición

Ordenar

Limite
50

Longitud de texto
100

Acción
Mostrar

SELECT * FROM `wp_users` LIMIT 50 (0.000 s) [Modificar](#)

<input type="checkbox"/> Modify	ID	user_login	user_pass	user_nicename	user_email
<input type="checkbox"/> modificar	1	admin	\$P\$B/9D9pV.3yhD9JIiZCHih26/gR43Te1	admin	

Resultado completo
☐ 1 registro

Modify
Guardar

Selected (0)
Modificar Clonar Eliminar

Exportar (1)

[Importar](#)

3- En el campo “user_pass” seleccionamos codificación MD5, introducimos la nueva contraseña, y guardamos los cambios.

Modificar: wp_users

ID	<input type="text"/>	1
user_login	<input type="text"/>	admin
user_pass	<input type="text" value="md5"/>	hackeado123
user_nicename	<input type="text"/>	admin
user_email	<input type="text"/>	SEGDEVAPP@gmail.com
user_url	<input type="text"/>	
user_registered	<input type="text"/>	2019-09-26 17:42:18
user_activation_key	<input type="text"/>	
user_status	<input type="text"/>	0
display_name	<input type="text"/>	admin

Finalizada esta etapa disponemos del nombre de usuario “admin” y le cambiamos la contraseña a “hackeado123”.

Algunas medidas a implementar para evitar lo descrito anteriormente sería cambiar el puerto por defecto del motor MySQL y bloquear las conexiones entrantes desde el exterior a dicho puerto.

[1] Pueden descargar Adminer desde <https://www.adminer.org/>

Bypass Restrict Login IP

En el paso anterior, obtuvimos el usuario y contraseña del administrador del sitio web. El paso siguiente es intentar loguearnos desde la url default de WordPress, ingresando a

`http://elcronista-reportero.com/wp-login.php`

Al intentar loguearse con el usuario “admin” y la contraseña “hackeado123” nos muestra un error que dice que nuestra IP no está habilitada para iniciar sesión.

ERROR: IP not allowed.

Identificamos que el plugin que utilizan para dicha medida de seguridad es “**Restrict login to IP**” y, analizando su código fuente, descubrimos que tiene un importante fallo de seguridad.

El mismo se origina cuando se desea agregar compatibilidad con Cloudflare. Al estar activado, todas las peticiones realizadas por cualquier visitante son hechas con la IP de Cloudflare, y éste notifica la IP del visitante mediante la cabecera HTTP en el campo **CF-CONNECTING-IP**.

El plugin obtiene la dirección IP real de la conexión y, en caso de que exista una IP en el campo CF-CONNECTING-IP de la cabecera HTTP, la utiliza como IP del visitante.

```
function get_my_ip_restrict_login(){
    $ip = $_SERVER["REMOTE_ADDR"];
    if(isset($_SERVER["HTTP_CF_CONNECTING_IP"]) and !empty($_SERVER["HTTP_CF_CONNECTING_IP"])) $ip =
    $_SERVER["HTTP_CF_CONNECTING_IP"];
    return $ip;
}
```

En la variable **\$_SERVER["REMOTE_ADDR"]** se encuentra la IP real de la conexión TCP, que en nuestro caso sería la IP del servidor Cloudflare, y en la variable **\$_SERVER["HTTP_CF_CONNECTING_IP"]** está el valor del campo CF-CONNECTING-IP de la cabecera HTTP.

Como el plugin no corrobora que la IP real sea la de Cloudflare para posteriormente leer la cabecera CF-CONNECTING-IP, podemos hacer uso de esto con el objetivo de saltar la seguridad del bloqueo del login a una IP determinada.

Para poder enviar un valor personalizado en el campo CF-CONNECTING-IP de la cabecera HTTP es necesario hacer un Bypass a Cloudflare, es decir, saltarnos a Cloudflare para evitar que nos edite la cabecera con la IP real nuestra, en vez de enviar el valor personalizado que nosotros querramos.

Para realizar lo descrito anteriormente, desarrollamos una aplicación en PHP en donde introducimos los siguientes datos. Dicha aplicación se llama **bypass_restrict_login.php**

1. **IP Real Server:** Es la IP obtenida en el paso Bypass Cloudflare, la cual coincide con IP-VMwp
2. **URL Login WordPress:** Es la URL donde hacemos la petición para loguearnos. Es la siguiente <http://elcronista-reportero.com/wp-login.php>
3. **IP Habilitada:** Dicha IP es la que está permitida para iniciar sesión, la obtenemos en la base de datos > tabla “wp_options” > option_name llamado “restric_login_ip_ip”

Mostrar: wp_options

Visualizar contenido
Mostrar estructura
Modificar tabla
Nuevo

Mostrar
Condición

option_name ▼ = ▼ restric_login_ip_ip

(donde sea) ▼ = ▼

SELECT * FROM `wp_options` WHERE `option_name` = 'restric_login_ip_ip' LIMIT 50

<input type="checkbox"/> Modify	option_id	option_name	option_value	autoload
<input type="checkbox"/> modificar	187	restric_login_ip_ip	181.28.190.82	yes

Resultado completo
☐ 1 registro

Modify
Guardar

Selected (0)

Modificar
Clonar
Eliminar

Importar

4. Usuario y Contraseña: Ponemos los datos de las credenciales de WordPress obtenidos en pasos anteriores.

IP Real Server:

IP-VMwp

URL login WordPress

http://elcronista-reportero.com/wp-login.php

IP Habilitada:

181.28.190.82

Usuario:

admin

Contraseña:

hackeado123

Enviar

Una vez completados los datos, hacemos clic en **enviar** y obtendremos como respuesta el nombre y valor de la Cookie generada por WordPress para ser utilizada en la siguiente fase del experimento.

Cookie name: wordpress_46d03120c78d5bfd3171ee77555b5a1c
Cookie Value: admin|1572177360|5DArPfmM1DWITwFVwgU44IW4iOf5l

Session Hacking

Con los datos anteriores (Cookie name y Cookie value) utilizamos cualquier extensión del navegador que nos permita incluir una Cookie personalizada hardcoded a nuestra navegación. Nosotros utilizamos “EditThisCookie”.



EditThisCookie

Ofrecido por: editthiscookie.com

★★★★★ 11,170

[Herramientas del programador](#)

👤 2,698,389 usuarios

Link de EditThisCookie:

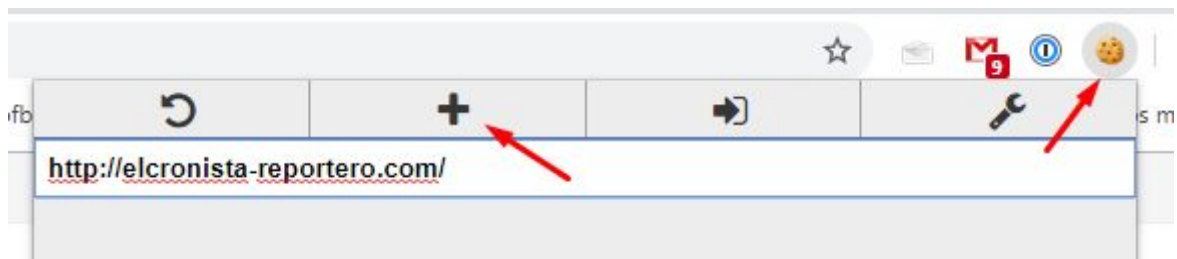
<https://chrome.google.com/webstore/detail/editthiscookie/fngmhnnpilhplaeedifhccceomclgfbg?hl=es-419>

Los pasos a seguir para incorporar la Cookie de forma hardcodeada es la siguiente:

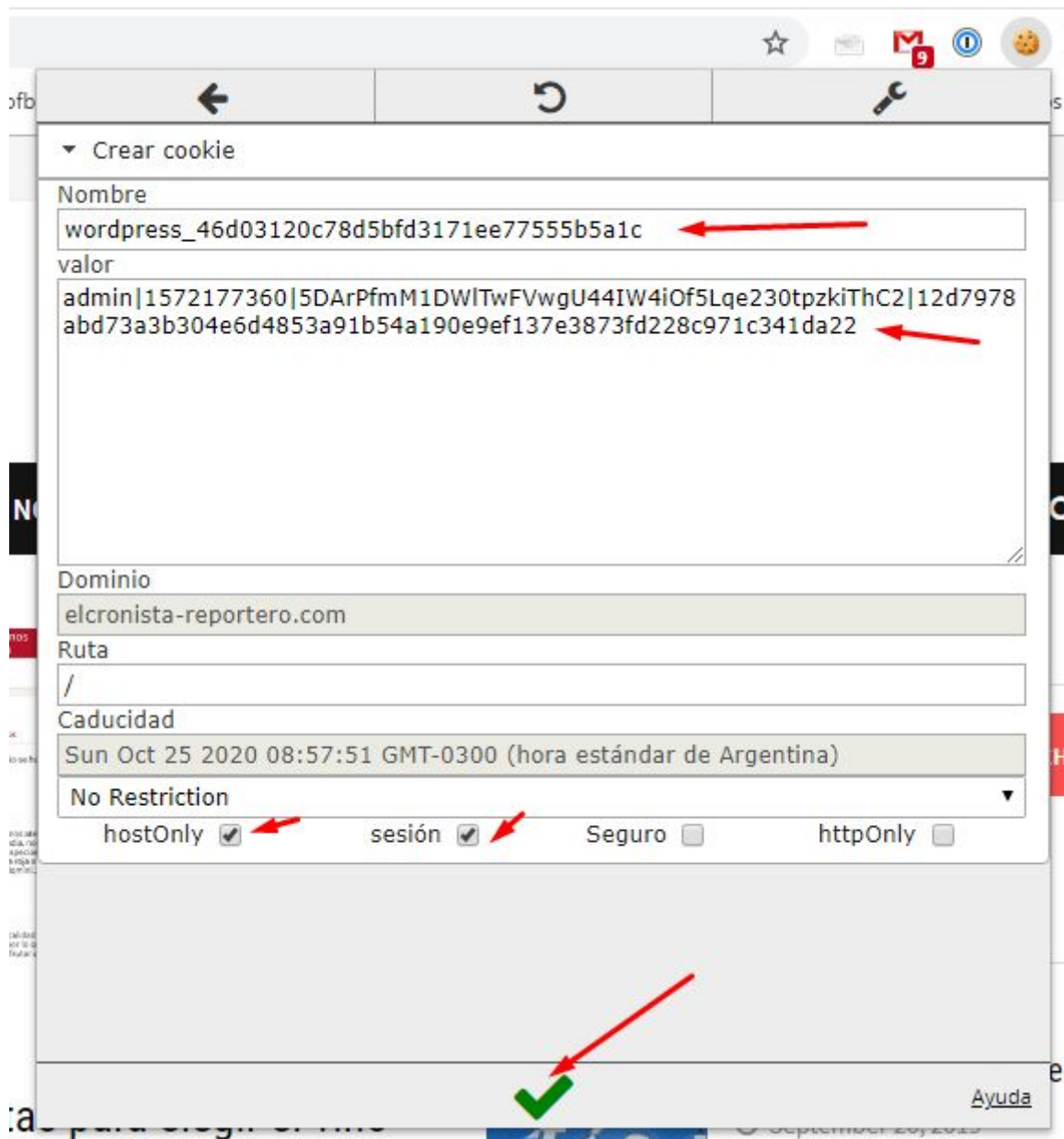
1. Ingresar al sitio web

<http://elcronista-reportero.com/>

2. Click en el icono generado por la extensión “EditThisCookie” para abrir su panel de control y Click en agregar Cookie



3. Agregamos los datos de la Cookie y tildamos los checkbox “hostOnly” y “sesion”



4. Damos click en guardar e ingresamos a la URL del panel de control de WordPress

`http://elcronista-reportero.com/wp-admin/`

Ya estamos logueados como administrador.

Bypass restrict upload only image mime type

Navegando por el panel de administración, encontramos un plugin que incorpora la posibilidad de subir imágenes a nuestro sitio web. Dicho plugin es público en el repositorio de plugins de WordPress. Analizamos su código y encontramos la siguiente vulnerabilidad:

La verificación que hace el plugin para corroborar que el archivo subido sea una imagen tiene el problema de que sólo valida el MIME TYPE, pero no la extensión y contenido del archivo. Asimismo, guarda el archivo con la extensión original sin modificación alguna.

Para engañar al upload de que nuestro archivo es una imagen, debemos agregar el encabezado correspondiente de una imagen a nuestro archivo.

El encabezado en hexadecimal para un archivo formato PNG es el siguiente:

89 50 4E 47 0D 0A 1A 0A	
-------------------------	--

Byte(s)	Propósito
89	Tiene el bit más alto puesto a 1 para detectar sistemas de transmisión que no soportan datos de 8 bits y para reducir el riesgo de que un fichero de texto sea erróneamente interpretado como PNG.
50 4E 47	En ASCII , las letras "PNG" permitiendo que una persona identifique el formato en caso de verlo en un editor de texto.
0D 0A	Una nueva línea con estilo DOS (CRLF) para detectar las conversiones de final de línea entre DOS y UNIX.
1A	Un byte que detiene el despliegue del fichero bajo DOS cuando se ha usado el comando TYPE.
0A	Una nueva línea en UNIX (LF) para detectar la conversión de final de línea entre DOS y UNIX.

El archivo original de prueba es el siguiente:

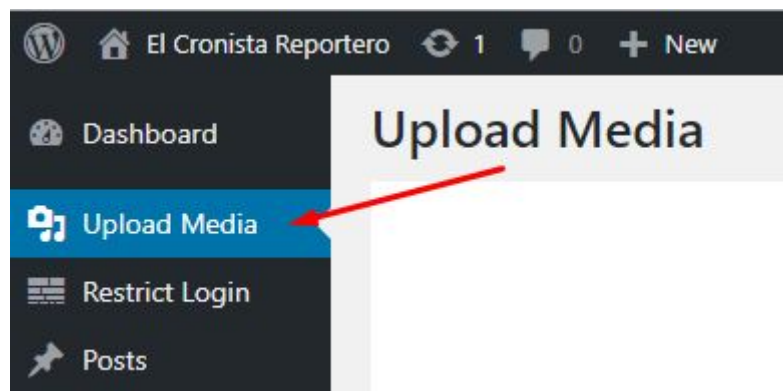


Para agregar el encabezado personalizado y hacerlo pasar por una imagen, necesitamos utilizar un editor hexadecimal. En nuestro caso, utilizamos el software **HxD** que lo pueden descargar de la web oficial <https://mh-nexus.de/en/hxd/>.

Abrimos el archivo con el programa HxD y agregamos al inicio el encabezado los datos hexadecimales "89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52" correspondiente al encabezado de una imagen PNG y un poco de información sobre la imagen.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Texto decodificado
00000000	89	50	4E	47	0A	1A	0A	00	00	00	0D	49	48	44	52		%PNG.....IHDR
00000010	3C	3F	70	68	70	20	65	63	68	6F	20	27	48	6F	6C	61	<?php echo 'Hola
00000020	20	4D	75	6E	64	6F	27	3B	20	3F	3E						Mundo'; ?>

Guardamos el archivo como “hola_mundo_simula_ser_imagen.php”, y lo subimos por medio del plugin de subida de imágenes localizado en el tablero de control de WordPress.



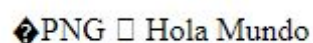
Seleccionamos el archivo, y le damos click en “Upload Image”



La subida fue exitosa y nos devuelve el enlace directo del archivo



Ingresamos al enlace directo, y vemos cómo se ejecuta correctamente nuestro archivo PHP.



Cuando disponemos de un upload de imágenes, es recomendable corroborar:

- El MIME TYPE del archivo subido sea una imagen
- Que la extensión del archivo subido sea una imagen
- Corroborar que el archivo subido se pueda interpretar como código de imagen
- Denegar ejecución de código en el directorio donde se suben las imagenes.

En este paso, logramos engañar al upload para hacerle creer que un archivo PHP es una imagen, dando así la posibilidad de subir cualquier índole de archivo y ejecutarlo.

Infectar servidor

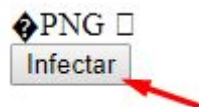
En el paso anterior, logramos engañar al upload para poder subir archivos PHP y ejecutarlos en el servidor. En este paso, vamos a intentar subir una Shell al servidor y tener acceso a los archivos del mismo.

A fines educativos, vamos a utilizar un gestor de archivos programado en PHP, que lo pueden descargar desde aquí <https://github.com/alexantr/filemanager>.

Necesitamos infectar el servidor con un archivo PHP que no disponga de la cabecera de archivo en formato PNG, ya que la misma puede ocasionar problemas en la correcta ejecución del código. En consecuencia, vamos a crear un PHP que crea un archivo sin la cabecera PNG con el código que nosotros deseamos, en nuestro caso, el pseudo Shell.

Repetimos el paso anterior con el archivo "infectar.php". Una vez agregada la cabecera PNG, lo guardamos como "infectar_simula_ser_imagen.php" y lo subimos vía upload web, ingresando al enlace directo.

Una vez ingresado al enlace directo, damos click en el botón infectar



Al dar click en el botón infectar, se crea la pseudo shell en el mismo directorio

```

===== SHELL SUBIDA =====
nombre del archivo: shell.php
  
```

Enlace directo de la pseudo shell

`http://elcronista-reportero.com/wp-content/plugins/upload-media-plugin/public/img/shell.php`

Mediante esta pseudo shell podemos tener control de los archivos del servidor.



<input type="checkbox"/>	Name	Size	Modified	Perms
<input type="checkbox"/>	.well-known	Folder	26.09.19 20:40	0755
<input type="checkbox"/>	abc	Folder	25.10.19 15:34	0755
<input type="checkbox"/>	cgi-bin	Folder	26.09.19 20:37	0755
<input type="checkbox"/>	public	Folder	26.09.19 21:23	0755
<input type="checkbox"/>	wp-admin	Folder	26.09.19 21:38	0755
<input type="checkbox"/>	wp-content	Folder	24.10.19 06:27	0755
<input type="checkbox"/>	wp-includes	Folder	21.05.19 21:24	0755
<input type="checkbox"/>	.htaccess	517 B	24.10.19 12:53	0644
<input type="checkbox"/>	adminer-4.7.4-mysql.php	350.86 KiB	24.10.19 06:17	0644
<input type="checkbox"/>	index.php	420 B	01.12.17 02:11	0644
<input type="checkbox"/>	license.txt	19.47 KiB	01.01.19 23:37	0644
<input type="checkbox"/>	readme.html	7.27 KiB	15.10.19 01:50	0644
<input type="checkbox"/>	wp-activate.php	6.76 KiB	12.01.19 09:41	0644
<input type="checkbox"/>	wp-blog-header.php	369 B	01.12.17 02:11	0644
<input type="checkbox"/>	wp-comments-post.php	2.23 KiB	21.01.19 04:34	0644
<input type="checkbox"/>	wp-config.php	2.66 KiB	26.09.19 20:42	0600
<input type="checkbox"/>	wp-cron.php	3.76 KiB	09.01.19 11:37	0644
<input type="checkbox"/>	wp-links-opml.php	2.44 KiB	16.01.19 08:29	0644
<input type="checkbox"/>	wp-load.php	3.23 KiB	01.12.17 02:11	0644
<input type="checkbox"/>	wp-login.php	38.62 KiB	26.09.19 20:42	0644
<input type="checkbox"/>	wp-mail.php	8.21 KiB	01.12.17 02:11	0644
<input type="checkbox"/>	wp-settings.php	18.53 KiB	26.09.19 20:42	0644

Anexo I

Crear maquina virtual backend

Instalar sistema operativo

Utilizaremos el sistema operativo **Centos 7 64 bits minimal**, se puede descargar desde la web oficial.

Link de descarga Centos 7 64 bits minimal:

http://espejito.fder.edu.uy/centos/7.7.1908/isos/x86_64/CentOS-7-x86_64-Minimal-1908.iso

Link corto: <https://bit.ly/2VpuaPv>

Montar Imagen

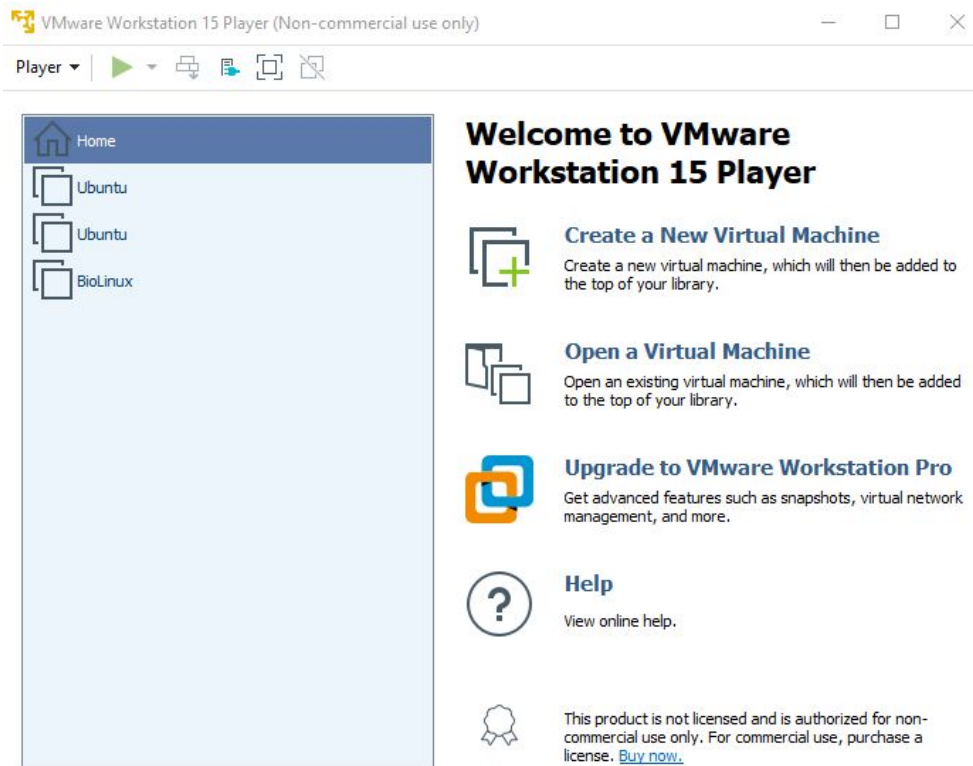
Una vez descargada la imagen de Centos (.iso), vamos a montarla. Para esto vamos utilizar VMWare (pueden utilizar VirtualBox).

Link de descarga VMWare:

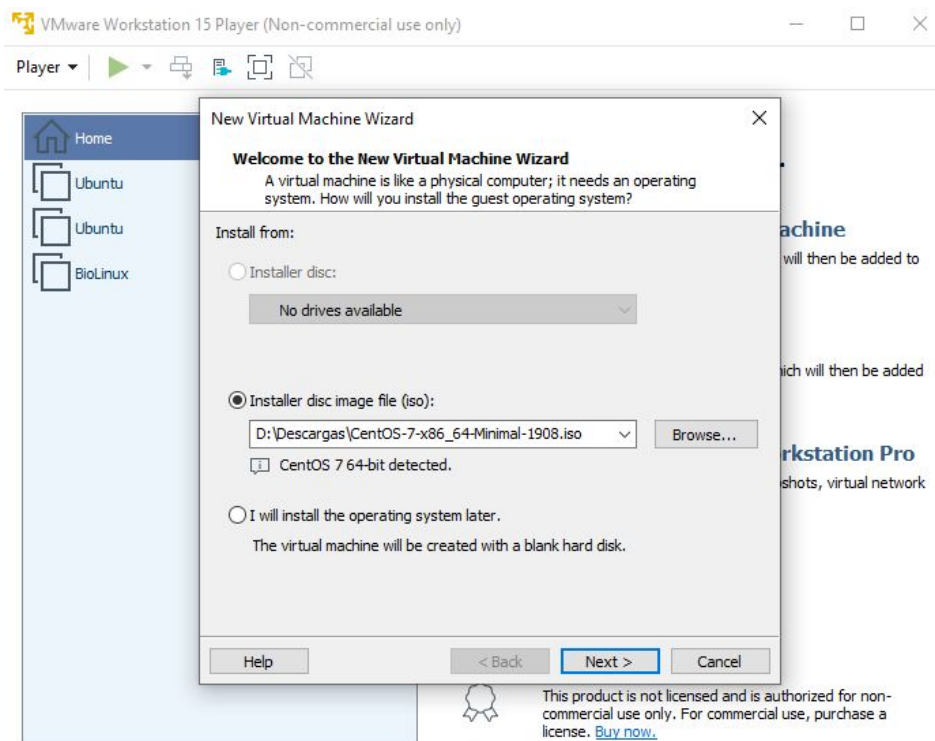
https://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/15_0

Link corto: <https://bit.ly/2pHwaDL>

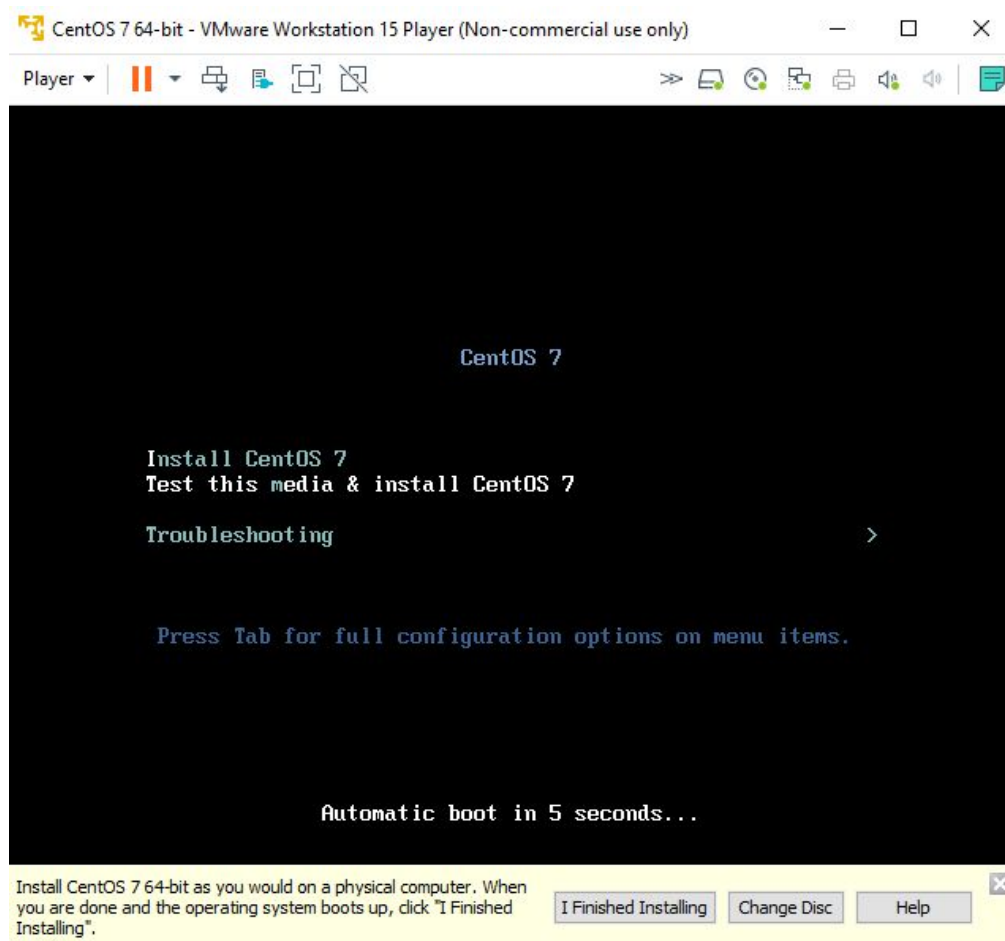
Una vez descargado, lo instalan siguiendo la instrucciones y lo abren. La pantalla que van a ver es la siguiente:



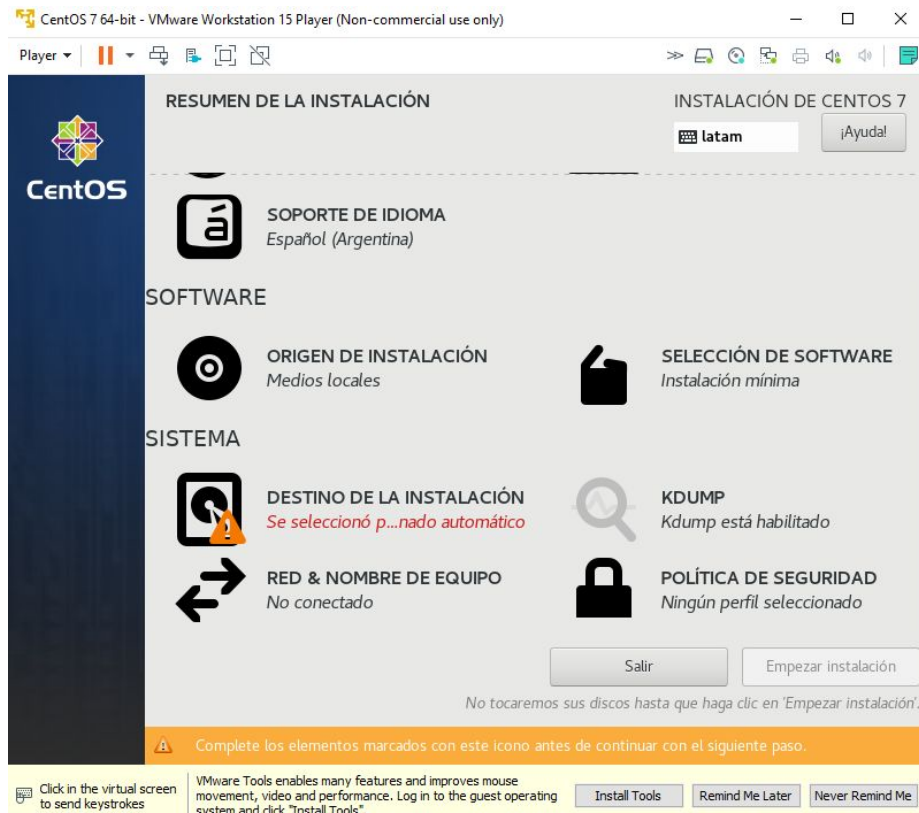
Paso siguiente hacemos click en “Create a New Virtual Machine”



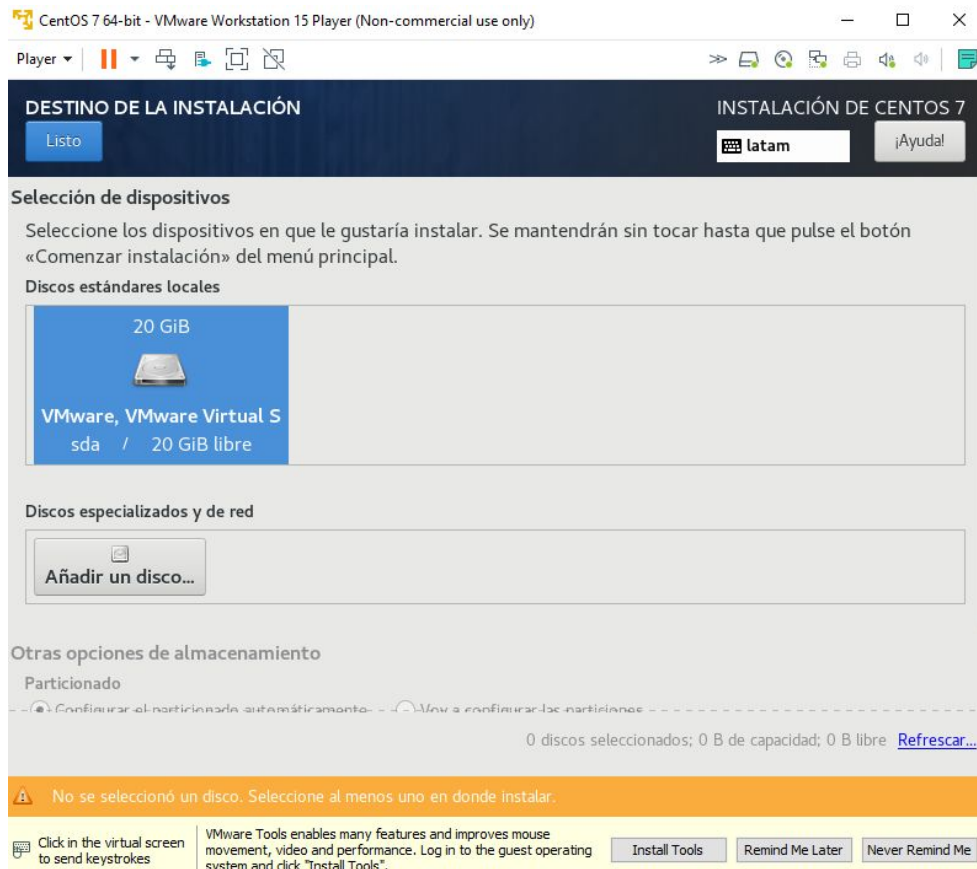
En la pantalla que nos abrió buscamos la imagen de Centos que descargamos anteriormente (como se ve en la imagen) y le damos next hasta finalizar. La VM comenzará a instalarse. En la siguiente pantalla lo que van a tener que hacer es seleccionar “Install CentOS 7”.



Continuando, van a tener que seleccionar el idioma, y luego esto el disco donde se va a instalar el OS.



Hacen click sobre “Destino de la Instalación”, seleccionan el disco de VMWare, le dan listo y “Empezar Instalación”.



Paso siguiente comenzará la instalación propiamente dicha y mientras esperamos que finalice, necesitamos poner una contraseña al usuario root del OS (para el caso del laboratorio la contraseña para el root será “123456”, pero ustedes pueden elegir la que quieran). No es necesario crear un usuario.

Para finalizar le van a tener que dar “Reiniciar”.

Recuerden que el usuario es “root” y la contraseña es “123456”.

Instalar panel de control hosting

Utilizaremos el panel de control hosting VestaCP, por ser gratuito y contar de forma predeterminada con lo que necesitamos para llevar adelante el laboratorio.

La instalación se lleva a cabo mediante SSH ejecutando los siguientes comandos

```
# Download installation script
wget http://vestacp.com/pub/vst-install.sh

# Run it
bash vst-install.sh --nginx no --apache yes --phpfpm no --named yes --remi yes --vsftpd
yes --proftpd no --iptables yes --fail2ban yes --quota no --exim no --dovecot no
--spamassassin no --clamav no --softaculous no --mysql yes --postgresql no --force
```

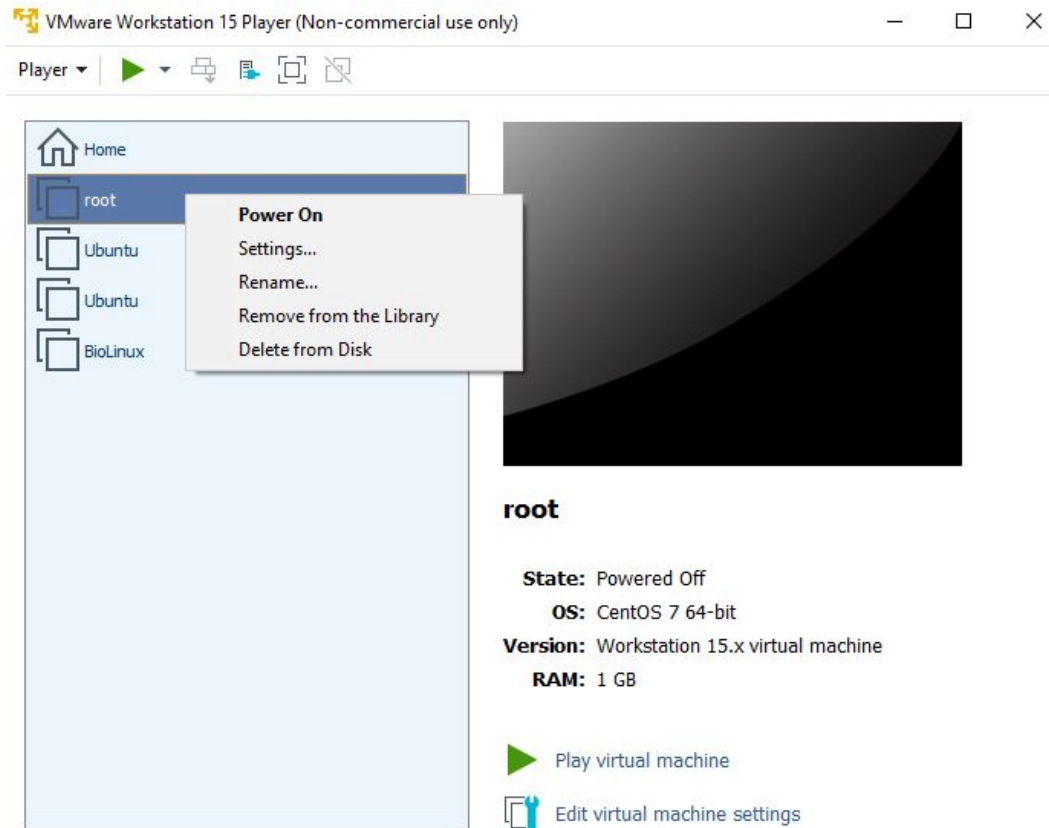
En el caso de que el OS no les reconozca el comando “*wget*”, lo pueden instalar mediante la siguiente línea “*yum install wget*”

El instalador lo guiará para completarlo con éxito, una vez finalizado le mostrará por pantalla los datos de acceso al panel de control. Deberá completar la tabla 1 del anexo II con dichos datos (La ip todavía no, ya que es necesario poner como Host Only a nuestro servidor).

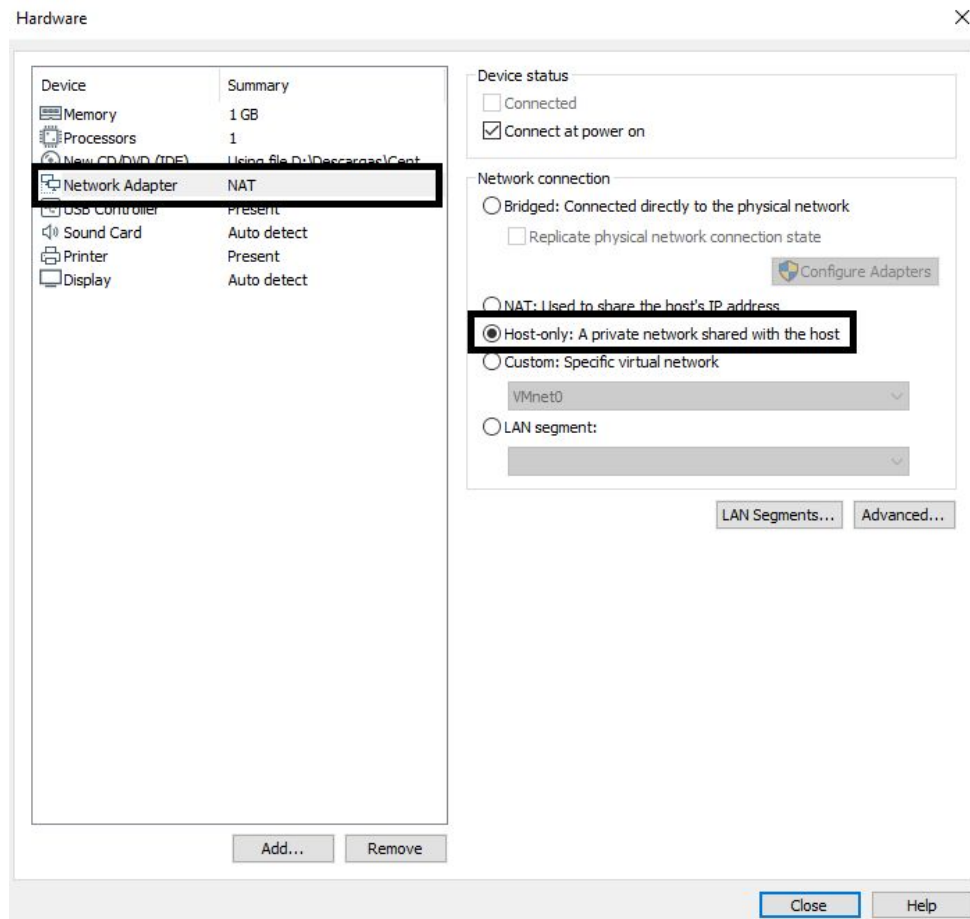
Ejemplo:

[illegible]

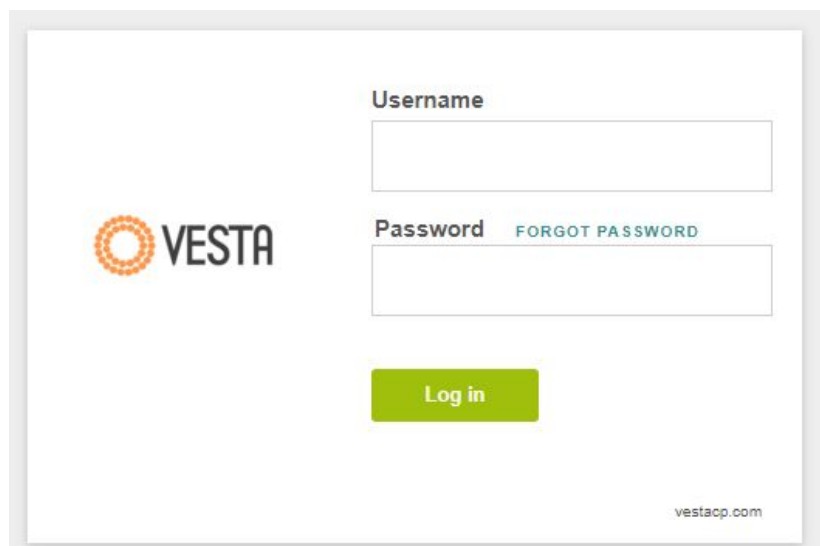
Antes de continuar vamos tenemos que apagar la VM, y luego en la pantalla del VMWare hacemos click derecho para ir a Settings.



Dentro de Settings, configuramos nuestra VM como “Host-Only de la siguiente manera”, y la volvemos a encender.



Una vez que vuelve a iniciar la VM, obtenemos la ip mediante el comando “ip a” y la completamos en la tabla 1. Desde cualquier navegador podemos acceder al panel de control (<https://IP-VMwp:8083/>) que se verá de la siguiente forma:

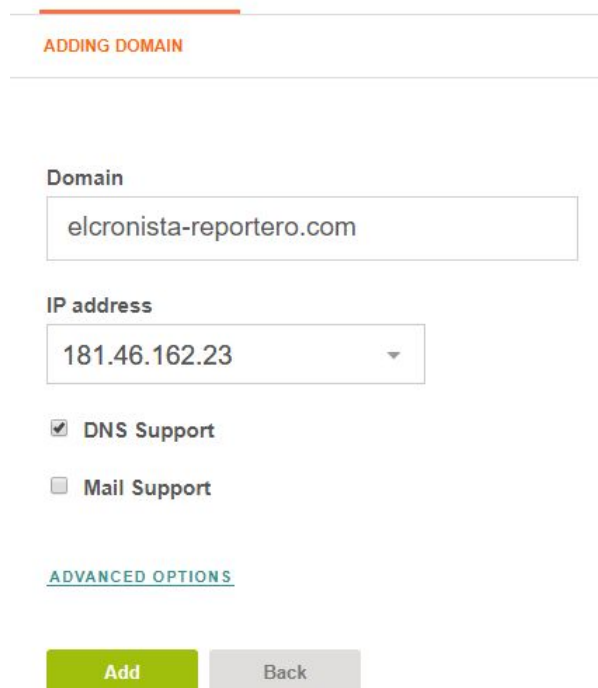


En caso de querer cambiar la contraseña al usuario admin pueden utilizar el comando “v-change-user-password admin **newpassword**” o “passwd admin”. Nosotros cambiamos la contraseña a “123456”

Configurando VestaCP

Crear Dominio

Cuando se encuentren logeados en el panel de control, se dirigen a la pestaña “WEB”, donde van a ver un botón con un símbolo “+”. Lo clickean y verán lo siguiente, donde deberán crear el dominio elcronista-reportero.com:



ADDING DOMAIN

Domain

elcronista-reportero.com

IP address

181.46.162.23

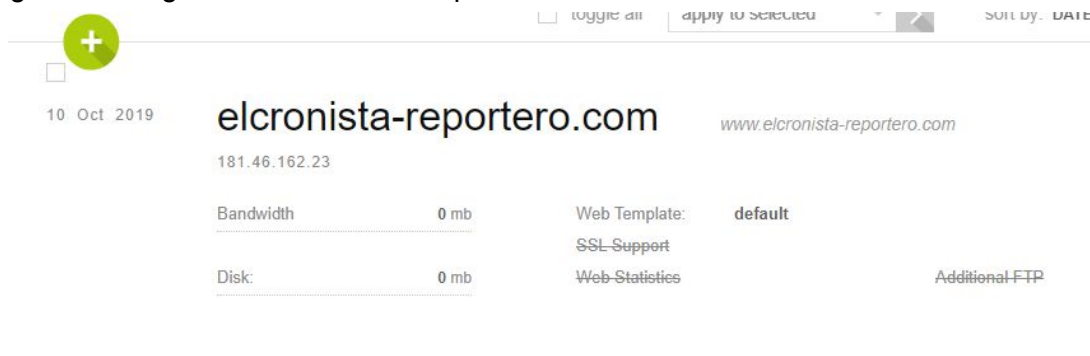
☒ DNS Support

☐ Mail Support

[ADVANCED OPTIONS](#)

Add Back

La siguiente imagen es como debería quedarles creado:



<input type="checkbox"/>	10 Oct 2019	elcronista-reportero.com	www.elcronista-reportero.com		
		181.46.162.23			
Bandwidth	0 mb	Web Template:	default		
		SSL Support			
Disk:	0 mb	Web Statistics		Additional FTP	

Una vez finalizado este paso es muy importante realizar lo descrito en “Vincular dominio a la IP” para poder asociar el dominio a la IP de la VMwp. [\[Click aqui para ir a “Vincular dominio a la IP”\]](#) (Importante: en vez de IP-VMcloudflare utilizar IP-VMwp dado que todavía no está configurada la VMcloudflare)

El archivo host debería quedar de la siguiente manera

IP-VMwp elcronista-reportero.com IP-VMwp www.elcronista-reportero.com IP-VMwp ftp.elcronista-reportero.com
--

Crear base de datos

Ahora se dirigen la pestaña “DB”, nuevamente hacen click sobre el símbolo “+” y con los datos que se encuentra en la tabla 1 del anexo 2 completan el formulario de la siguiente forma:

ADDING DATABASE

*Prefix **admin_** will be automatically added to database name and database user.*

Database

wp

admin_wp

User (maximum 16 characters length, including prefix)

wp

admin_wp

Password / generate

123456

👁

Type

mysql

▼

Host

localhost

▼

Charset

utf8

▼


Send login credentials to email address

Add

Back

Una vez creado, así deberían verlo:

USER	WEB	DNS	MAIL	DB	CRON	BACKUP
users: 1 suspended: 0	domains: 1 aliases: 1 suspended: 0	domains: 2 records: 27 suspended: 0	domains: 1 accounts: 0 suspended: 0	databases: 2 suspended: 0	jobs: 8 suspended: 0	backups: 0

 PHPMYADMIN <input type="checkbox"/> toggle all		apply to selected	sort by: DATE ↓
09 Oct 2019		EDIT	SUSPEND
admin_wp		DELETE	
Disk	0 mb	User: admin_wp	Host: localhost
Charset:	UTF8	Type:	mysql

Instalar Wordpress

El siguiente paso es descargar el tar.gz de WordPress, y transferirlo vía ftp a la VMwp creada anteriormente.

Link de descarga: <https://wordpress.org/download/>

Dicho paso se puede realizar con la ayuda del programa “Filezilla” o utilizando la consola de la siguiente manera:

```
# Desde su computadora, pararse en el path donde se descargó el archivo tar.gz
cd <path-de-wordpres.tar.gz>
```

```
# Iniciar sesion ftp
ftp elcronista-reportero.com
```

```
# Se pedirá el user y contraseña antes de confirmar la operación
user: admin
password: <password obtenida al instalar vestacp>
```

```
# Luego, se visualizará una consola ftp con el siguiente formato:
ftp >
```

```
# Cambiar a directorio (dentro de la vm) donde se desea transferir el archivo
ftp> cd tmp
```

```
# Setear lo siguiente
ftp > binary
```

```
# Transferir archivo tar.gz descargado
ftp> put wordpress-5.2.3.tar.gz
```

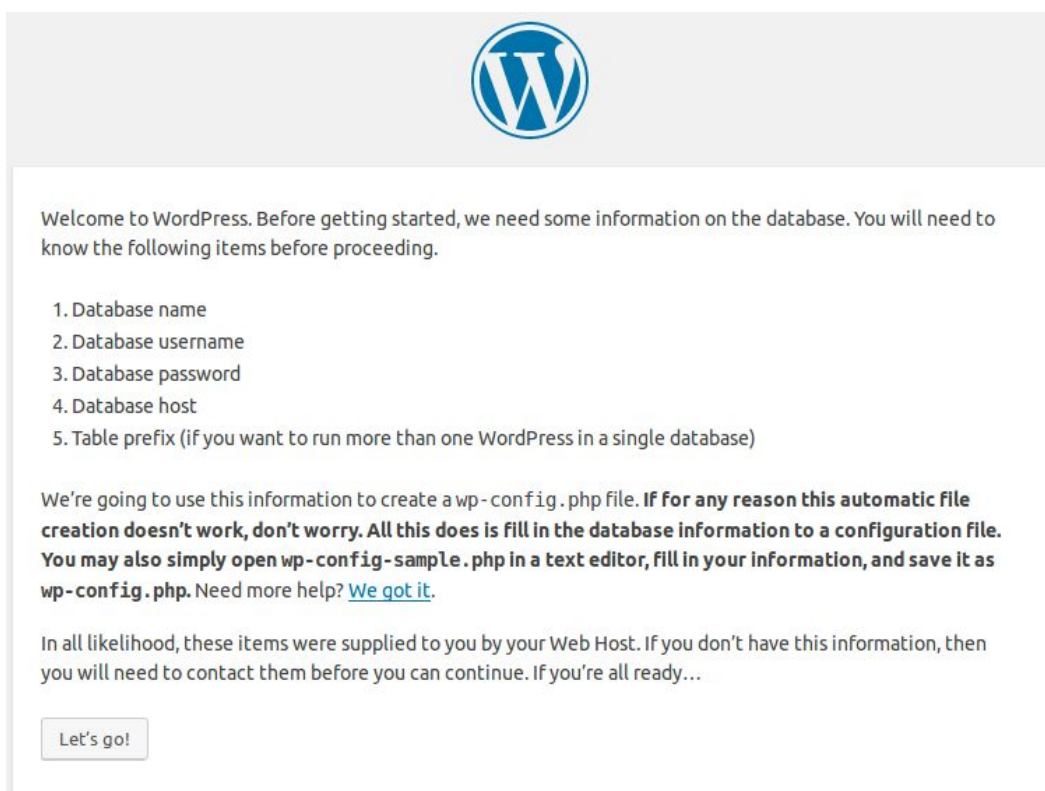
```
# Salir de ftp
ftp> bye
```

```
# En la consola de VMwp, mover archivo transferido
cd /home/admin/tmp
mv wordpress-5.2.3.tar.gz /home/admin/web/elcronista-reportero.com

# descomprimir
cd /home/admin/web/elcronista-reportero.com
tar -xf wordpress-5.2.3.tar.gz

# mover archivos descomprimidos a:
mv wordpress/* ./public_html
```

Si todo sale bien, al intentar acceder a **elcronista-reportero.com** desde el browser, aparecerá la siguiente pantalla:



Al apretar en **Let's go!**, llevará un formulario donde se deben completar los datos de la base de datos (Ver anexo II - Tabla con datos importantes y credenciales).

Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="admin_wp"/>	The name of the database you want to use with WordPress.
Username	<input type="text" value="admin_wp"/>	Your database username.
Password	<input type="text" value="123456"/>	Your database password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

Una vez ingresados los datos, se preguntará la confirmación para instalar wordpress

All right, sparky! You've made it through this part of the installation. WordPress can now communicate with your database. If you are ready, time now to...

Antes de comenzar a instalar, se preguntarán los datos iniciales, los cuales deben ser completados con los valores de la tabla.

Site Title	<input type="text" value="El Cronista"/>
Username	<input type="text" value="cronista"/> <small>Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.</small>
Password	<div><input type="password" value="123456"/><input type="button" value="Hide"/></div> <div>Very weak</div> <p>Important: You will need this password to log in. Please store it in a secure location.</p>
Confirm Password	<input checked="" type="checkbox"/> Confirm use of weak password
Your Email	<input type="text" value="usuario@elcronista.com"/> <small>Double-check your email address before continuing.</small>
Search Engine Visibility	<input type="checkbox"/> Discourage search engines from indexing this site <small>It is up to search engines to honor this request.</small>

Una vez finalizado, se mostrará el siguiente cartel de éxito:

Success!

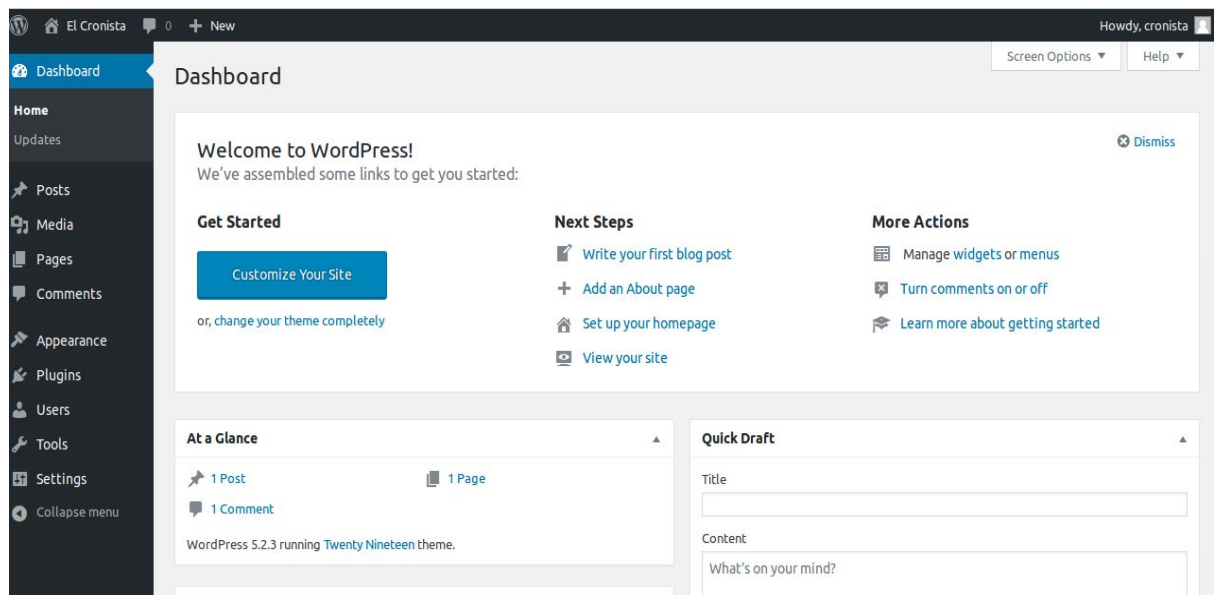
WordPress has been installed. Thank you, and enjoy!

Username cronista

Password *Your chosen password.*

Log In

Una vez terminado, loguearse con el usuario. Esto llevará a la pantalla de inicio del admin



Instalar Plugins Wordpress

Hay dos forma para realizar la instalación de los plugins, utilizando el panel de control de WordPress > Plugin > Add New o subiendolo de forma manual en la carpeta destinada para alojar los plugins en WordPress

Recomendamos utilizar la primera opción por su simpleza y rapidez, a modo teórico dejamos descrito las instrucciones de como hacerlo manualmente.

De forma manual se pueden subir los archivos hacer mediante programa “Filezilla” o vía línea de comando.

Para instalar un plugin, deben ubicarse en la carpeta:

`/home/admin/web/elcronista-reportero.com/public_html/wp-content/plugins`

```
# Desde su computadora, pararse en el path donde se descargó el archivo tar.gz
cd <path-de-plugin.zip>
```

```
# Iniciar sesion ftp e ingresar usuario y contraseña
ftp elcronista-reportero.com

# Cambiar a directorio (dentro de la vm) donde se desea transferir el archivo
ftp> cd tmp

# Setear lo siguiente
ftp > binary

# Transferir archivos zip descargado
ftp> put upload-media-plugin.zip

# Salir de ftp
ftp> bye

# dentro de la vm, mover archivo transferido
cd /home/admin/tmp
mv upload-media-plugin.zip
/home/admin/web/elcronista-reportero.com/public_html/wp-content/plugins

# cambiar a directorio de plugins
cd /home/admin/web/elcronista-reportero.com/public_html/wp-content/plugins

# descomprimir plugins para instalarlos
unzip upload-media-plugin.zip
```

Una vez hecho esto para cada plugin, ir al dashboard del admin y dirigirse a la sección **Plugins -> Installed Plugins**. Allí, se mostrarán los plugins que acaban de instalarse.

Plugins Add New	
All (6) Inactive (6)	
Bulk Actions ▼ Apply	
<input type="checkbox"/> Plugin	Description
<input type="checkbox"/> Akismet Anti-Spam Activate Delete	Used by millions, Akismet is quite possibly the best way in the world to protect your blog from spam. Then go to your Akismet Settings page to set up your API key. Version 4.1.2 By Automattic Visit plugin site
<input type="checkbox"/> Hello Dolly Activate Delete	This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation growing up with the comfort of technology. This is the last of WordPress's "default" plugins, and it's the one that's been around the longest. See a lyric from Hello, Dolly in the upper right of your admin screen on every page. Version 1.7.2 By Matt Mullenweg Visit plugin site
<input type="checkbox"/> Restrict login to IP Activate Delete	Restringir login a WPADMIN a una unica IP (Compatible con Cloudflare) Version 1.0.0 By SEGAPPWEB
<input type="checkbox"/> Simplificar wordpress admin Activate Delete	Plugin para simplificar el WPADMIN desactivando algunas funciones Version 1.0.0 By SEGAPPWEB
<input type="checkbox"/> Thumbnail img Activate Delete	Comprime sus imagenes (de ser posible) para reducir su tamaño sin perder calidad. Version 1.0.0 By SegAPPweb
<input type="checkbox"/> Upload Media IMG Activate Delete	Sistema para subir imagenes, comprueba mime type para mayor seguridad. Version 1.0.0 By SEGAPPWEB
<input type="checkbox"/> Plugin	Description

El último paso es seleccionar **Activate** por cada plugin que desee utilizarse. Es necesario que activen como **último plugin** en Simplificar wordpress admin.

Para finalizar tiene que configurar el restrict login to ip, para esto hacen click en "Restrict Login" en la columna de izquierda e ingresan la siguiente ip "181.28.190.82".

← → ↻ 🏠 ⚠ No es seguro | elcronista-reportero.com/wp-admin/admin.php?page=restrict-login-ip-page

El Cronista 0 + New

[Dashboard](#)
[Upload Media](#)
[Restrict Login](#)
[Posts](#)
[Pages](#)
[Comments](#)
[Plugins](#)
[Users](#)
[Collapse menu](#)

Restrict Login IP

Settings

Your IP:

In blank for disabled.

[Save Changes](#)

Crear maquina virtual proxy - Cloudflare

Instalar sistema operativo

Mismo procedimiento que en la sección “Anexo I > Crear maquina virtual backend > Instalar sistema operativo” [\[ir\]](#)

Instalar Varnish

Ejecutar en consola

```
# Instalar repositorio EPEL
yum -y install epel-release

# Instalar varnish
yum -y install varnish

# Iniciar Varnish
systemctl restart varnish

# Iniciar en el boot
systemctl enable varnish
```

El Software Varnish Cache por defecto viene configurado para ejecutarse en el puerto “6081”, tenemos que cambiarlo a que escuche en el puerto “80”

```
# Entrar en modo editor de texto
vi /etc/varnish/varnish.params

# Buscar “6081” y reemplazarlo por “80”
# La linea quedaria asi: VARNISH_LISTEN_PORT=80
# Guardar archivo

# Reinicio Varnish
systemctl restart varnish
```

El último paso para configurar Varnish Cache es decir el archivo en donde se pone las reglas para avisarle que hacer al proxy inverso con cada petición.

```
# Abrimos para editar el archivo de reglas de varnish
```

```
vi /etc/varnish/default.vcl
```

```
# Cambiar IP y Puerto destino
```

```
# Buscar "127.0.0.1" y reemplazarlo por "IP-VMwp"
```

```
# Buscar "8080" y reemplazarlo por "80"
```

```
# Agregamos dentro del procedimiento sub vcl_recv {
```

```
# Agregamos debajo de "sub vcl_recv {" el código del recuadro siguiente
```

```
set req.http.CF-Connecting-IP = client.ip;
```

```
return(pipe);
```

```
# Guardamos el archivo
```

```
# Reiniciar varnish
```

```
systemctl restart varnish
```

Puede ser que al terminar estos paso no se pueda ingresar a la web

"elcronista-reportero.com", esto es por que la vm de cloudflare tiene el puerto 80 cerrado,

para abrirlo deben correr el comando **"*sudo firewall-cmd --zone=public --add-port=80/tcp --permanent*"**

Anexo II

Datos servidor backend

Dato	Valor
Server Settings	
Dirección IP	IP-VMwp
Puerto SSH	22
Root Password	123456
VestaCP Settings	
VestaCP Login URL	https://IP-VMwp:8083/

Username	admin
Password	123456
DB Settings	
DB User	admin_wp
DB Base	admin_wp
DB Password	123456
Wordpress Settings	
WP User	cronista
WP Password	123456

Tabla 1 - Datos servidor backend

Datos servidor proxy inverso

Dato	Valor
Server Settings	
Dirección IP	IP-VMcloudflare
Puerto SSH	22
Root Password	123456

Tabla 2 - Datos servidor proxy inverso

Anexo III

Plugins

Restrict login to IP

Modelo simplificado de un plugin que restringe el iniciar sesión únicamente a una IP determinada, no se contempla restricción por usuario, grupo de usuario, se aplica la restricción a todos los usuarios sin discriminación alguna.

En términos generales agregamos una rutina de código que es ejecutado de forma automática por el sistema de filtros de WordPress, dotada una alta prioridad para que se ejecute por encima de cualquier rutina por defecto.

La parte del código que realiza lo descrito anteriormente es la siguiente

```
42 add_filter( 'authenticate', 'restrict_login_ip_auth_signon', 8, 3 );
43 function restrict_login_ip_auth_signon( $user, $username, $password ) {
44     // Corroboro que el usuario y la contraseña no esten vacias
45     if (empty($username) or empty($password)){
46         return $user;
47     }
48     // Corroboro que el usuario no este instanciado
49     if ( $user instanceof WP_User ) {
50         return $user;
51     }
52
53     // Corroboro errores en filtros anteriores
54     if ( is_wp_error( $user ) ) {
55         return $user;
56     }
57     // Obtengo la IP del visitante que se intenta loguear
58     $ip_login = get_my_ip_restrict_login();
59
60     // Obtengo la IP autorizada
61     $ip_allowed = get_option( 'restric_login_ip_ip' );
62
63     // Si es vacia, todas las IPs estan autorizadas
64     if(empty($ip_allowed)) return $user;
65
66     // Si la IP autorizada es igual a la del visitante que se intenta loguear entonces OK
67     if($ip_allowed == $ip_login) return $user;
68
69     // Caso contrario, doy error
70     $user = new WP_Error( 'authentication_failed_ip_not_allowed', __( '<strong>ERROR</strong>: IP not allowed.' ) );
71
72     // Apago los demas filtros de WordPress, debe haber una mejor forma de hacerlo pero no afecta el objetivo del TP.
73     remove_action( 'authenticate', 'wp_authenticate_username_password', 20);
74     remove_action( 'authenticate', 'wp_authenticate_email_password', 20);
75     remove_action( 'authenticate', 'wp_authenticate_spam_check', 20);
76
77     // Devuelvo $user, pero notar que anteriormente se creo un ERROR, filtros posteriores notaran dicho error.
78     return $user;
79 }
```

Filtro utilizado “Authenticate”, documentacion en

https://codex.wordpress.org/Plugin_API/Filter_Reference/authenticate

Thumbnail img

Modelo simplificado de plugin que comprime las imágenes puestas como destacadas en WordPress en caso de no conocer cómo comprimir el contenido muestra el original.

Dicho modelo está inspirado en el plugin TimThumb, siendo el mismo vulnerable a File Inclusion y Arbitrary Code Execution afectando a miles de sitios webs.

Fuente:

<https://blogvault.net/common-attacks-on-wordpress-sites-101-file-inclusion-arbitrary-code-execution/>

<https://blog.sucuri.net/2019/08/timthumb-attacks-the-scale-of-legacy-malware-infections.html>

El funcionamiento del modelo es muy básico, reemplaza las URL de las imagenes miniaturas de WordPress por la URL del archivo PHP encargado de comprimir las imágenes y ponerlas en caché, pasando por parámetro GET la URL original del archivo.

Upload Media IMG

Si bien WordPress viene con un sistema de multimedia muy robusto, con el fin de mostrar una vulnerabilidad muy frecuente en los sistemas de subida de archivos, decidimos incluir este plugin, el cual usa el que publica entradas en el sitio web para subir las imágenes, dicho sistema corrobora que el archivo sea una imagen mediante el TYPE MIME.

Disable WPadmin Menu

También mencionado como “Simplificar wordpress admin”, dicho plugin oculta menús y botones del panel de administración para evitar desviar la atención a pantallas que no tienen relevancias para el objetivo del proyecto.