

Año 0 - Número 11
30.09.2013

Hackers & DEVELOPERS

Change

RESPONSABLE EDITORIAL
Eugenia Bahit

HACKER TEAM:

Eugenia Bahit
María José Montes Díaz
Milagros Infante Montero
Pablo Bernardo Baragaño
y colaboradores



Hacker

es alguien que disfruta
jugando con la inteligencia

HACKERS & DEVELOPERS MAGAZINE
MAGAZINE DIGITAL DE EDICIÓN MENSUAL SOBRE SOFTWARE LIBRE, HACKING Y PROGRAMACIÓN

CHANGE

S U P L E M E N T O E S P E C I A L
H A C K E R S & D E V E L O P E R S M A G A Z I N E

NUESTRO COMPROMISO CON QUIENES APUESTAN AL VERDADERO CAMBIO

PAPERS DE • EUGENIA BAHIT • MARÍA JOSÉ MONTES DÍAZ • MILAGROS INFANTE MONTERO • PABLO BERNARDO • Y COLABORADORES

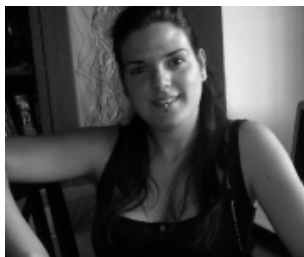
EN ESTA EDICIÓN:

Protegiendo GRUB.....	3
Creación de paquetes .deb.....	6
Ataque ARP Spoofing: Detectándolo mediante Bash Scripting.....	11
PseInt: Una herramienta de programación indispensable	14
Entendiendo el código fuente de los archivos SVG creados con Sozi e Inkscape..	17
La importancia de estar documentado – Mozilla Developer Network.....	23
El software y el Derecho.....	27

Protegiendo GRUB

GRUB, por defecto, permite que todos los usuarios puedan editar las entradas del menú en el arranque. Esto es para poder restablecer contraseñas y otras tareas. Sin embargo, si utilizamos un equipo compartido, es posible que necesitemos proteger el acceso a la edición de entradas. Aquí veremos cómo hacerlo.

Escrito por: **María José Montes Díaz** (Docente & Programadora)



Estudiante de Grado Ingeniería en Tecnología de la información. Técnico en informática de gestión. Monitora FPO. Docente de programación Python y Scratch para niños de 6-12 años. Activista del software libre y cultura libre.

Webs:

Blog: <http://archninf.blogspot.com.es/>

Redes sociales:

Twitter: [@MMontesDiaz](https://twitter.com/MMontesDiaz)

Una característica de **GRUB** (y otros cargadores de arranque), es que nos proporcionan una línea de comandos o un editor para poder ajustar los parámetros que le pasamos al **kernel**. Esto es muy útil en ciertos casos, como por ejemplo, tener la necesidad de arrancar en un modo diferente (en lugar de gráfico, en modo multiusuario), desactivar alguna opción de algún controlador, etc.

Una de las cosas que nos permite hacer es arrancar un intérprete de órdenes (**shell**), sin necesidad de realizar un **login**, opción que utilizamos en el artículo anterior para restablecer la contraseña del usuario **root**, pero que también podemos utilizar para recuperar un sistema dañado.

Esto está muy bien y es muy útil, hasta que compartimos la máquina con otras personas, nos dejamos el portátil en casa de alguien, nos levantamos del puesto de trabajo un momento, etc. En ese caso, podríamos tener problemas, pues **el arranque que se realiza se hace con permisos de usuario root**.

Entonces, ¿por qué no viene protegido el gestor de arranque? En la mayor parte de los casos, las personas que tienen acceso físico al equipo, tienen la posibilidad de acceder al mismo con múltiples mecanismos, con lo que añadir una protección aquí, de entrada, solo complica más la recuperación del sistema. En realidad, donde se hace realmente necesario proteger el arranque es en equipos compartidos, normalmente en lugares de trabajo, donde debemos establecer muy bien los permisos y roles de las diferentes personas que utilizarán el mismo.

Cómo no podía ser de otra manera, nuestro GRUB nos proporciona mecanismos para proteger esta edición. El mecanismo por defecto es muy restrictivo: No podemos ejecutar la entrada y tampoco editarla (aunque se puede suavizar esta restricción). Pero... ¿cómo podemos restringir tanto la ejecución como la edición de entradas en el GRUB? Disponemos de varias opciones. Empezaré por la más simple de todas.

Lo primero que debemos hacer es editar el archivo de configuración de GRUB, `/boot/grub/grub.cfg`. Al principio del archivo, debemos indicar qué usuarios dispondrán de permisos para ejecutar y editar las entradas de GRUB. Esto se hace estableciendo una lista de usuarios, separadas por comas, espacios o puntos y comas, de la siguiente manera:

```
set supersusers="root,mary"
```

Ya tenemos los usuarios que podrán ejecutar y editar las entradas del menú. Ahora hay que indicar cuál será la clave para estos usuarios. Hay que tener en cuenta que no es la clave de sistema: El sistema no ha arrancado. Es una clave específica para GRUB. Además, aparte del usuario **root**, podemos establecer los que queramos, sin necesidad de que coincidan con los usuarios reales del sistema. Añadimos también una entrada por cada usuario, con su clave, así como usuarios que no queremos que ejecuten todas las entradas, pero sí las que especifiquemos:

```
password root clave_root  
password mary clave_mary  
password usuario clave_usuario
```

A partir de aquí, sólo los usuarios **root** y **mary** podrán ejecutar todas las opciones del menú de GRUB. Sus claves serán **clave_root** y **clave_mary**, respectivamente. La última línea es para establecer una clave a **usuario**, que utilizaremos después.

Pero... ¿No hay algo "raro"? Pues sí, **las claves están especificadas en texto plano**. Esto, en un entorno de trabajo, no es recomendable, así que vamos a generar un **hash** para ocultar nuestra clave. Para ello debemos utilizar un comando que nos proporciona **GRUB**. Ejecutamos, en la línea de órdenes:

```
grub-mkpasswd-pbkdf2
```

Nos solicitará que introduzcamos una contraseña y, después, que la confirmemos, devolviendo algo similar a esto:

```
$ grub-mkpasswd-pbkdf2  
  
Introduzca la contraseña:  
  
Reintroduzca la contraseña:  
  
El hash PBKDF2 de su contraseña es  
grub.pbkdf2.sha512.10000.B2B745D9862E47F95CA64C5DEB31CE0443E038122D2775F4294A7C5F15F79D5E1CC44  
97A9BBBEA080F4847433291385AFADD253564037E79F999877CCDBEDB58.1951C740709F4EB38228E2049F180607F6  
E36E6F5E0DA3F2552419C0B934DE6E573FFCC0B43D1C2B3D0D69644224259F451178D7CFE99DFF38FFF5356E7978CA
```

Ahora, en lugar de poner la clave en texto plano en el archivo `/boot/grub/grub.cfg`, debemos introducir el hash devuelto e indicar que la clave no está en texto plano:

```
password_pbkdf2 root grub.pbkdf2.sha512.10000.B2B745D986...  
...
```

Todas las entradas del menú de GRUB están protegidas. Aunque, en ciertas ocasiones, puede que nos interese habilitar la ejecución a algún usuario en particular. Para ello, en la entrada a la que queramos habilitar la ejecución a algún usuario que **no** esté en la lista **supersusers**, añadimos lo siguiente:

```
menuentry 'Arch Linux, con Linux core repo kernel' --class arch --class gnu-linux --class gnu  
--class os $menuentry_id_option 'gnulinux-core repo kernel-true-2288d60f-b933-4c45-98f7-  
536130a19caf' --users usuario { ... }
```

Si queremos que la entrada pueda ser ejecutada por cualquier usuario:

```
menuentry 'Arch Linux, con Linux core repo kernel' --class arch --class gnu-linux --class gnu
--class os $menuentry_id_option 'gnulinux-core repo kernel-true-2288d60f-b933-4c45-98f7-
536130a19caf' --unrestricted { ... }
```

En cualquier caso, aunque todos los usuarios puedan ejecutar esta entrada, es decir, no se pide clave para ejecutar, solo los usuarios establecidos en la lista **superusers** podrán editarla. Esto es igual para el caso anterior, solo los usuarios indicados en **--users** pueden ejecutar la entrada, pero solo los **superusers** pueden, además, editarla. Con estas opciones, parece que ya sí es práctico.

Desgraciadamente, el comando **grub-mkconfig** no soporta añadir automáticamente la restricción de usuarios por entrada. Así que podemos hacer lo siguiente:

Editamos el archivo `/etc/grub.d/00_header` y le añadimos la lista de usuarios y claves:

```
cat << EOF

set superusers="root"
password_pbkdf2 root <clave>
password_pbkdf2 usuario1 <clave2>
password_pbkdf2 usuario2 <clave2>
...

EOF
```

Con esto, tendríamos todas las entradas generadas por **GRUB** protegidas. Las entradas en las que deseemos especificar a un grupo de usuarios o bien, que no estén restringidas, debemos añadirlas manualmente. Para ello utilizaremos el archivo `/etc/grub.d/40_custom`, donde debemos añadir la entrada correspondiente, por ejemplo:

```
menuentry 'Arch Linux, with Linux core repo kernel unrestricted' --class arch --class gnu-
linux --class gnu --class os $menuentry_id_option 'gnulinux-core repo kernel-true-2288d60f-
b933-4c45-98f7-536130a19caf' --unrestricted {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1
--hint-baremetal=ahci0,msdos1 2288d60f-b933-4c45-98f7-536130a19caf
    else
        search --no-floppy --fs-uuid --set=root 2288d60f-b933-4c45-98f7-536130a19caf
    fi
    echo 'Cargando Linux core repo kernel...'
    linux /boot/vmlinuz-linux root=UUID=2288d60f-b933-4c45-98f7-536130a19caf rw
    clocksource=tsc quiet splash
    echo 'Cargando imagen de memoria inicial...'
    initrd /boot/initramfs-linux.img
}
```

Al ejecutar **grub-mkconfig**, nos generará por defecto las entradas que ya teníamos y estarán protegidas (solo los usuarios especificados en la lista **superusers** podrán editarlas y ejecutarlas). Además, una entrada que podrá ser ejecutada sin necesidad de introducir claves.

Fuentes:

<https://www.gnu.org/software/grub/manual/grub.html>
<https://wiki.archlinux.org/index.php/GRUB>

Creación de paquetes .deb

Un simple *script* que no requiere más que un simple "ejecutar", se distribuye muy fácilmente. Pero ¿qué sucede con las aplicaciones con gran cantidad de archivos? Sobre todo, con aquellas aplicaciones donde un *tarball* no simplifica la tarea de repartir los ficheros que deban almacenarse en rutas diversas. Sin dudas, saber crear un paquete Debian, es la forma más segura de crear un instalador para nuestra aplicación.

Escrito por: **Eugenia Bahit** (GLAMP Hacker & eXtreme Programmer)



Eugenia es **Arquitecta de Software**, docente e instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y **eXtreme Programming**. Miembro de la **Free Software Foundation**, **The Linux Foundation** y **Debian Hackers**. Creadora de **python-printr**, **Europio Engine** y colaboradora de **Vim**. Fundadora y Responsable Editorial de **Hackers & Developers Magazine**.

Webs:

Cursos de programación: www.cursosdeprogramacionadistancia.com

Web personal: www.eugeniabahit.com

Redes sociales:

Twitter / Identi.ca: [@eugeniabahit](https://twitter.com/eugeniabahit)

En ediciones anteriores de Hackers & Developers Magazine, comentábamos acerca de cómo distribuir aplicaciones y/o módulos Python a través del Python Package Index para luego instalarlos mediante un simple `pip install`. Incluso, más adelante, estuvimos viendo cómo se crean las páginas man del Manual de GNU/Linux. Para continuar con esta serie de procesos básicos que todo programador experto debe conocer, hoy, en unos pasos muy sencillos, tomaremos nota de cómo se crean los archivos `.deb`

Preparando la estructura de directorios

Lo primero que se debe tener bien en claro, es cuál será la ruta o rutas de destino de la aplicación al momento de desempaquetar el `.deb` resultante. Teniendo en claro esto, se debe crear la estructura de directorios correspondiente. ¿Cómo? Primero necesitamos una carpeta donde trabajar:

```
mkdir myproject
```

Dentro de esa carpeta, es donde debemos emular (o clonar) las rutas de destino de la aplicación. Por ejemplo, imaginemos una aplicación Web que debe colocarse en el directorio `/var/www/myapp` pero que además, requiere que en el directorio `/var/log` se cree un directorio `myapp`:

```
mkdir -p myproject/var/www/myapp
mkdir -p myproject/var/log/myapp
```

De esta forma, la estructura de directorios se vería como la siguiente:

```
myproject/
├── var
│   ├── log
│   │   └── myapp
│   └── www
│       └── myapp
```

El siguiente paso, será copiar dentro de las carpetas creadas, todos los archivos que sean necesarios. Es decir que, si nuestra aplicación se encuentra en una carpeta llamada MyApp1_2_0, todo el contenido de esa carpeta, debería copiarse dentro myproject/var/www/myapp.

Generando un archivo de control del paquete Debian

Todo paquete Debian (archivo .deb) debe contener un directorio **DEBIAN** con sus archivos correspondientes. Lo primero entonces, será crear la carpeta DEBIAN dentro de nuestro directorio de trabajo:

```
mkdir myproject/DEBIAN
```

Existen **19 tipos de archivos** entre los que se “deben” colocar y los que se “pueden”. Esto significa que mientras algunos archivos serán obligatorios, otros, serán opcionales a tal punto que en muchos casos, si no son estrictamente necesarios, se “debe” evitar colocar dichos archivos en el paquete.

La lista de archivos obligatorios puede obtenerse junto a su detalle correspondiente, ingresando en <http://www.debian.org/doc/manuals/maint-guide/dreq.es.html> mientras que la de los opcionales, se encuentra en <http://www.debian.org/doc/manuals/maint-guide/dother.es.html>.

Entre los archivos que se “deben” colocar, se encuentra el archivo **control** quien contiene ciertas directivas que serán utilizadas por gestores de paquetes como apt, aptitude y dpkg -entre otros-, al momento de instalar (o mejor dicho, “desempaquetar”) el .deb.

Las directivas completas del archivo control, se encuentran descritas en el **Capítulo 5 del Manual de Políticas de Debian**, el cual puede ser consultado desde <http://www.debian.org/doc/debian-policy/ch-controlfields.html>.

El archivo control se encuentra **dividido en dos bloques**: el primero, contiene información pura y exclusivamente sobre los fuentes del paquete (aquellos archivos requeridos para armar y construir la aplicación final que utilizará el usuario), mientras que el segundo, la contiene sobre el binario de la aplicación en sí misma (es decir, la aplicación que finalmente será utilizada).

De las **directivas del archivo DEBIAN/control**, las más interesantes (y casi ineludibles), son las siguientes:

Package

Nombre del binario. Sigue las mismas reglas de nombre que la directiva source del bloque 1.

Priority

Indica la importancia que la instalación del paquete implica para el usuario. Los diferentes niveles de prioridades de un paquete Debian, pueden ser consultado en <http://www.debian.org/doc/debian-policy/ch-archive.html#s-priorities>. No obstante, los mismos no necesitan ser demasiado explicados ya que poseen nombres sumamente descriptivos:

required, important, standard, optional, extra.

Section

Área de aplicación en la cual se ha clasificado el paquete. Las secciones pueden ser:

admin, cli-mono, comm, database, debug, devel, doc, editors, education, electronics, embedded, fonts, games, gnome, gnu-r, gnustep, graphics, hamradio, haskell, httpd, interpreters, introspection, java, kde, kernel, libdevel, libs, lisp, localization, mail, math, metapackages, misc, net, news, ocaml, oldlibs, otherosfs, perl, php, python, ruby, science, shells, sound, tasks, tex, text, utils, vcs, video, web, x11, xfce, zope

Maintainer

Nombre y correo electrónico del desarrollador original. Debe guardar el formato:

Nombre Apellido <usuario@dominio.foo>

Version

Versión del paquete binario (para entender mejor como versionar de manera estándar una aplicación, te recomiendo leer el Anexo I al final de este artículo).

Architecture

Generalmente suele utilizarse como valor **any**, cuando el binario se ha escrito en lenguaje compilado y **all**, si se tratase de un lenguaje interpretado. No obstante, una referencia completa puede obtenerse en <http://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Architecture>.

Description

Descripción de la aplicación, dividida en dos sectores: descripción corta (primera línea) y descripción larga (líneas siguientes). Los dos sectores de descripción, deben contener el siguiente formato:

descripción corta de hasta 60 caracteres
descripción larga, tabulada con espacios en blanco
que además, puede ser de varias líneas.

Depends, Recommends, Suggests

Muy descriptivos los nombres de estas directivas, indicarán de cuáles paquetes depende la aplicación y cuáles son recomendados y sugeridos para el mejor funcionamiento de ésta.

Homepage

La URL del paquete y preferentemente se recomienda colocar aquella de la cuál el paquete pueda ser descargado.

Un ejemplo completo del archivo **DEBIAN/control**, podría verse de la siguiente manera:

Package: europiocli


```
Priority: extra
Section: php
Maintainer: Eugenia Bahit <ebahit@member.fsf.org>
Version: 3.2.17
Architecture: all
Description: Command Line Interface para Europio Engine
  Interfaz de línea de comandos que permite agilizar tareas de comunes
  en Europio Engine, como crear o eliminar módulos; agregar modelos,
  vistas y controladores, etc.
Depends: php5 (>=5.3.10), php5-cli (>=5.3.10), mysql-server (>=5.5.0)
Homepage: http://europio.org/some/dir/to/download/current-package
```

Por favor, nótese que las dependencias son separadas por una coma seguida de un espacio en blanco.

Entonces, creamos el archivo control con el contenido anterior, reemplazando el valor de cada directiva según corresponda:

```
touch myproject/DEBIAN/control
```

Construyendo el .deb

Para construir el paquete .deb se utilizará dpkg. Para evitar el uso de fakeroot al momento de construir el paquete con dpkg es necesario modificar el propietario de los archivos de forma recursiva:

```
sudo chown -R root:root myproject/
```

Si **dpkg-dev** no se encuentra instalado, debe instalarse ya que será necesario para la construcción:

```
sudo apt-get install dpkg-dev
```

Para construir el paquete, solo habrá que situarse en el nivel superior del directorio de trabajo y ejecutar:

```
dpkg -b myproject destino.deb
```

Se puede probar el paquete utilizando la opción -i de dpkg:

```
dpkg -i mipaquete.deb
```

Anexo I: Estándar para asignar números de versiones

El número de versión de un Software se encuentra conformado por cuatro componentes:

- Mayor y menor número de versión
- Mayor y menor número de revisión

Mayor número de versión

El mayor número de versión, debe ser modificado cuando el cambio hecho a la aplicación, requiere cambiar todo el paquete.

Menor número de versión

Éste, debe ser modificado cuando se realizan cambios significativos en la aplicación que requieran modificar una parte importante del paquete pero no todo.

Mayor número de revisión

Ante cualquier cambio de sentido y/o significado (por más mínimo que sea) de la aplicación, el número de revisión mayor es modificado.

Menor número de revisión

Se modifica ante cambios menores que generalmente tienen que ver con pequeñas cuestiones estéticas o correcciones de errores de mecanografiado, pero que no implican un cambio de sentido en la aplicación.

Solo los 3 primeros componentes del número de versión son los que importan y tienen un verdadero significado. Frente a números de versión como 2.1.0.0 se prefiere utilizar 2.1.0 directamente.

En entregas posteriores, intentaremos abarcar el estudio de otros archivos del directorio DEBIAN, procurando ahondar además, en la ejecución de *Scripts* post-instalación.

¿nos echarías
una mano?





Pagos seguros

con tu donación
nos ayudas a mantener vivo
este proyecto

SI EL ENLACE DE LA IMAGEN NO FUNCIONA, POR FAVOR, INGRESA LA SIGUIENTE URL: <http://www.hdmagazine.org/donar>

Ataque ARP Spoofing: Detectándolo mediante Bash Scripting

Cualquiera puede ser víctima de un ataque de ARP Spoofing, detecta este ataque mediante Bash Scripting. Es importante entender como funciona este tipo de ataque para poder protegerse de él.

Escrito por: **Pablo González Pérez** (Project Manager)



Ingeniero Técnico en Informática de Sistemas e Ingeniero en Informática por URJC. Project Manager en 11Paths, empresa perteneciente a Telefónica Digital. Premio Extraordinario Fin de Carrera por la URJC en Ingeniería Técnica en Informática de Sistemas en 2009. Premio al mejor expediente de su promoción por la Escuela Técnica Superior de Ingeniería Informática de la URJC en 2009. Fundador de Flu-Project. Autor de: Metasploit para Pentesters y Pentesting con Kali.

Webs:

www.flu-project.com

Redes sociales:

Twitter: [@pablogonzalezpe](https://twitter.com/pablogonzalezpe)

Un ataque ARP Spoofing es una de las técnicas más utilizadas para interceptar una comunicación en redes LAN, *Local Area Network*. Todos los dispositivos de una red de ordenadores que utilizan direccionamiento IPv4 se comunican mediante el nivel de enlace, ya que se encuentran en la misma red LAN. Los dispositivos necesitan conocer la dirección MAC o dirección física de los otros equipos de la red para poder enviar la información entre dichas máquinas. El protocolo ARP, *Address Resolution Protocol*, permite a partir de una dirección IP conocida descubrir cuál es la dirección física asociada. Los dispositivos de una red disponen de una tabla ARP, donde se asocian las direcciones IP a direcciones físicas. El protocolo dispone de dos tipos acciones: *request* y *reply*. En la acción *request* un dispositivo no conoce la dirección física asociada a una dirección IP por lo que envía un paquete *request* de ARP a todos los dispositivos de la red, es decir realiza *broadcast*, preguntando por una dirección IP. Será la máquina con dicha dirección IP la única que contestará al dispositivo que generó el *request*. Dicha máquina contestará con un paquete de *reply* indicando su dirección física.

Ejemplo de envenenamiento de ARP

Un ataque de ARP Spoofing es denominado *Man In The Middle* o MITM y permite a un usuario malicioso situarse en medio de una comunicación en una red de área local. El atacante, al colocarse en medio de la comunicación puede leer, inyectar o modificar información de la comunicación entre dos máquinas, sin que ninguna de éstas conozca dicha situación.

En otras palabras, el atacante se colocará entre el equipo A y el equipo B. Cuando el equipo A envíe tráfico al equipo B, dicho tráfico será realmente enviado al equipo del atacante. El atacante reenviará el tráfico al equipo B, para que nadie pierda información en la comunicación, por lo que nadie se dará cuenta de lo que sucede.

Se propone un ejemplo teórico para entender cómo funciona el envenenamiento:

- Dirección IP Router: 192.168.1.1 MAC: CA:FE:CA:FE:CA:FE
- Dirección IP Víctima: 192.168.1.33 MAC: FA:BA:DA:FA:BA:DA
- Dirección IP atacante 192.168.1.35 MAC: AA:BB:AA:BB:AA:BB

Supongamos que la tabla ARP del router tiene el siguiente aspecto:

IP Address	MAC Address
192.168.1.33	FA:BA:DA:FA:BA:DA
192.168.1.35	AA:BB:AA:BB:AA:BB

La tabla ARP de la víctima tiene, por ejemplo, el siguiente aspecto:

IP Address	MAC Address
192.168.1.1	CA:FE:CA:FE:CA:FE

El atacante enviará dos peticiones de tipo *reply*, una para el router y otra para la víctima. El contenido del paquete es el siguiente:

- La petición *reply* que el atacante envía al router indica que la dirección MAC de la víctima ha cambiado, indicándole que la nueva dirección MAC es la del dispositivo del atacante. De esta manera el router "piensa" que la dirección MAC pertenece al equipo de la víctima. Se puede observar como ahora hay dos entradas en la tabla con la misma dirección física. Este hecho es un indicador de que algo extraño sucede en la tabla ARP. La tabla ARP del router, tras el envenenamiento queda de la siguiente manera:

IP Address	MAC Address
192.168.1.33	AA:BB:AA:BB:AA:BB
192.168.1.35	AA:BB:AA:BB:AA:BB

- La petición *reply* que el atacante envía a la víctima le indica que la dirección MAC del router ha cambiado, indicándole que la nueva dirección MAC es la del dispositivo del atacante. La tabla ARP del router, tras el envenenamiento queda de la siguiente manera:

IP Address	MAC Address
192.168.1.1	AA:BB:AA:BB:AA:BB

De esta manera todos los envíos que realice la máquina de la víctima a Internet, lógicamente tendrán que circular a través del router o *gateway*, circularán por la máquina del atacante y viceversa.

El atacante podrá visualizar las credenciales que circulan mediante protocolos no seguros como HTTP o FTP, podrá reconstruir imágenes o archivos de una comunicación entre la víctima e Internet, podrá realizar robos de sesiones mediante la interceptación de *cookies* y suplantación de éstas, entre otras muchas cosas. La técnica de ARP Spoofing es la base a otros muchos ataques de redes que se pueden realizar.

Creación de Script en Bash para detectarlo

El código que se expone a continuación permite al usuario indicar cuál es la dirección física de su *gateway*, la dirección IP de éste y la interfaz de red por la que se navega en Internet. El script va a "vigilar" que en la tabla ARP el valor de la dirección física de dicha IP no cambie. En caso de que haya una petición *reply* que indique dicho cambio notificará dicha situación al usuario y "tumbará" la interfaz para que la información no pueda ser interceptada por el atacante.

```
#!/bin/bash  
  
if [ $# -ne 3 ]
```

```
then
    echo "Usage ./mitm.sh <mac address> <ip router address> <network interface>"
    exit
fi

mac=$1
router=$2
interface=$3

while true
do
    entradaRouter=$( arp -a | grep $router | cut -d' ' -f4 )

    if [ $entradaRouter != "" ]
    then
        if [ $mac != $entradaRouter ]
        then
            echo "[+] Warning: Atento la Mac ha cambiado"
            echo "[+] Desconectado interface $interface"
            ifconfig $interface down
            break
        fi
    fi
    sleep 5
done
```

Una vez se codifica el script se debe configurar como ejecutable, para ello se hace uso del comando *chmod*.

```
pablo@pablo-VirtualBox:~$ chmod u+x detectMitm.sh
pablo@pablo-VirtualBox:~$ ls -l detectMitm.sh
-rwxrw-r-- 1 pablo pablo 358 2013-08-27 21:52 detectMitm.sh
```

En caso de que los parámetros de entrada solicitados no sean proporcionados, el script mostrará información sobre su uso y los parámetros requeridos: dirección MAC que no debe cambiar, dirección IP del dispositivo que no debe cambiar y la interfaz por la que se tiene la comunicación con la red.

```
pablo@pablo-VirtualBox:~$ sudo ./detectMitm.sh
Usage ./mitm.sh <mac address> <ip router address> <network interface>
pablo@pablo-VirtualBox:~$ sudo ./detectMitm.sh d0:ae:ec:d5:85:ac 192.168.1.1 eth1
[+] Warning: Atento la Mac ha cambiado
[+] Desconectado interface eth1
pablo@pablo-VirtualBox:~$
```

Cada cinco segundos el script comprueba la tabla ARP con los parámetros indicados anteriormente, en caso de encontrarse que la dirección MAC ha cambiado se notificará por pantalla, se realizará un "down" sobre la interfaz de red y se finalizará la ejecución del script.

Se recomienda utilizar aplicaciones que monitoricen la tabla ARP para visualizar anomalías en ésta o en el comportamiento de la red, por ejemplo demasiados *reply* en un período de tiempo pequeño. Algunos profesionales prefieren sus pequeños trucos y scripts para llevar a cabo estas acciones.

PseInt: Una herramienta de programación indispensable

En la edición No. 1 de H&D, mi compañera Milagros Infante hizo una excelente introducción a esta estupenda herramienta de programación en su artículo: «PseInt: Una Invitación para entrar en el maravilloso mundo de la programación»¹. En esta ocasión la exploraremos a profundidad y veremos que su utilidad no solamente se centra exclusivamente en el campo didáctico, sino que también nos puede resultar muy provechosa para la realización de pruebas de concepto y documentación de proyectos entre otros propósitos.

Escrito por: **Mario Iván Ruvalcaba** (Profesional Técnico en Informática)



Mario Iván Ruvalcaba (a.k.a. **Perseo**) es estudiante de Ingeniería en Computación, Profesional Técnico en Informática y Cofundador de **<° DesdeLinux**. Usuario y amante de GNU/Linux, idealista, entusiasta y defensor del Software y Cultura Libre; activista y blogger. Linux user: 547468.

Webs:
www.perseosblog.com

Redes sociales:
Twitter / Identi.ca: [@icaroperseo](https://twitter.com/icaroperseo)

Sin lugar a dudas, el proceso de formación por el que tiene que pasar cualquier estudiante o entusiasta de la ingeniería de software es largo y paulatino. Durante dicho proceso se busca desarrollar las habilidades básicas necesarias para comenzar a crear o diseñar pequeños algoritmos que resuelvan de manera satisfactoria ciertos problemas específicos, siendo los **diagramas de flujo** y el **pseudocódigo** las herramientas más utilizadas dentro del paradigma de la programación estructurada o modular. Pero afrontémoslo, su naturaleza empírica las vuelve poco populares o simplemente “tediosas” entre los expertos y/o futuros desarrolladores llegando a ser infravaloradas o ignoradas por completo.

Debido a lo anteriormente expuesto, el objetivo principal de PseInt es proporcionar las herramientas necesarias para que el usuario centre su atención exclusivamente en los conceptos fundamentales del diseño de algoritmos, prescindiendo del conocimiento previo de algún lenguaje de programación así como también, el poder observar su funcionamiento de forma práctica. Otra ventaja es que una vez que ha sido implementado el algoritmo como una aplicación este pueda formar parte de su documentación.

¿Qué es PseInt?

PseInt² (PSeudo Intérprete) es una aplicación multiplataforma (GNU/Linux y sistemas privativos) y distribuida como software libre bajo licencia GPL, que ofrece un **entorno integrado de desarrollo**³ (IDE), el cual está formado por: un editor de textos, editor de diagramas de flujo, intérprete y depurador (*debugger*). Aunado a lo anterior, PseInt también permite el uso de perfiles e incluye una extensa documentación.

¹ <http://www.hdmagazine.org/?magazine=HackersAndDevelopers&num=1>

² <http://pseint.sourceforge.net/slide/pseint.html>

³ https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado

La aplicación es bastante rápida y ligera ya que consume una mínima cantidad de memoria RAM (144 kB aproximadamente) y la aplicación completa no sobrepasa los 5 MB en espacio en disco duro. Estas características lo hacen ideal para equipos poco potentes o antiguos.

Otro gran acierto por parte de PseInt es el proporcionar una interfaz sencilla, lo cual le brinda una imagen de extrema simplicidad de uso (que de hecho lo es) en contrapartida a lo que podríamos esperar de un clásico IDE (una ventana sobrecargada de opciones, menús y barras de herramientas, sin adentrarnos en el consumo de recursos). Por otra parte, las diversas funcionalidades se encuentran muy bien distribuidas por toda la ventana de la aplicación, sin que lleguen a resultar intrusivas.

Editor de texto

Además de las típicas características propias de un editor de textos (rehacer, deshacer, buscar, reemplazar, imprimir archivo, archivos por pestañas, mostrar el número de línea, entre otras) también posee las siguientes funcionalidades:

- Verificación, resaltado, autocompletado de sintaxis e indentación⁴ automática de pseudocódigo.
- También cuenta con la posibilidad de exportar directamente el pseudocódigo a código HTML con coloreado de sintaxis o a código C++ (actualmente esta funcionalidad aún se encuentra en fase experimental).

Los archivos creados en PseInt se guardan como archivos de texto plano utilizando la extensión `.psc`, por lo que pueden abrirse con cualquier aplicación que admita este formato.

Documentación

PseInt pone a nuestra disposición una vasta documentación misma que podemos consultar en todo momento, ya sea que deseemos obtener ayuda sobre el funcionamiento del programa o acerca de las principales características del pseudocódigo. También es posible obtener el listado de **Operadores y Funciones** aritméticos/lógicos (menú lateral izquierdo) o consultar la **Ayuda Rápida** (Shift + F1) para obtener más información acerca de alguna expresión o estructura de control determinada.

Dentro de la documentación también es posible encontrar algunos ejemplos de algoritmos como: convertir números enteros a notación romana, el juego de TaTeTi y las torres de Hanoi.

Editor de diagramas de flujo

Contrariamente a lo que se podría llegar a pensar, ambos editores no son independientes el uno del otro ya que trabajan de forma conjunta en el diseño del algoritmo, es decir que puede diseñarse el algoritmo a través de pseudocódigo y obtener su representación gráfica de forma automática como diagrama de flujo o como diagrama de Nassi-Shneiderman⁵ (también conocido como diagramas de Chapin) y viceversa. Una vez obtenido el diagrama de flujo puede escalarse y/o exportarse como imagen PNG o BMP.

Intérprete

PseInt incorpora un intérprete de pseudocódigo (en español) el cual se procesa como si se tratara de algún lenguaje de programación de alto nivel (C, C++, Pascal, etc.).

Depurador (debugger)

La principal finalidad de este componente del IDE es la de examinar el pseudocódigo en busca de errores de sintaxis o lógicos, una vez validado dará paso a la ejecución del algoritmo.

Un aspecto importante es que el depurador cuenta con la posibilidad de ejecutar el algoritmo paso a paso, algo bastante útil para poder darle seguimiento al control de flujo dentro de nuestro algoritmo así como el poder efectuar **pruebas de escritorio**. La información arrojada por el depurador es de lo más variada ya que podremos ver en tiempo real el contenido de variables y otros tipos de datos como arreglos o vectores. Por si esto fuera poco, PseInt cuenta con un "modo especial" donde el intérprete explica con lujo de detalles que acciones se están realizando en el paso actual.

⁴ <https://es.wikipedia.org/wiki/Indentaci%C3%B3n>

⁵ https://es.wikipedia.org/wiki/Diagrama_Nassi-Shneiderman

El uso tanto de un intérprete como de un depurador es bastante significativo no solo por lo que ello representa, sino que también resulta ser una introducción al uso de este tipo de herramientas que se emplean en otros IDE's específicos para lenguajes de programación de alto nivel.

Perfiles

PseInt cuenta con la funcionalidad de permitir la personalización o creación de varias reglas para la utilización del pseudocódigo (perfiles). Algunas de estas modificaciones son:

- Establecer la declaración de variables como obligatorio.
- No permitir el uso de variables u otras estructuras de datos sin inicializar.
- Requerir el uso del ";" como delimitador de expresiones.

De forma predeterminada, la aplicación incluye varios perfiles proporcionados por escuelas y universidades que lo han adoptado como herramienta didáctica.

Instalación

Actualmente no existen paquetes oficiales para las diversas distribuciones GNU/Linux (RPM, DEB, etc.) por lo que es necesario descargar el archivo *.tgz* directamente desde su página oficial⁶. Una vez descargado dicho archivo, basta con descomprimirlo y ejecutar el archivo *wxPseInt*, esto lanzará la aplicación y creará un acceso directo en nuestro menú de aplicaciones de nuestra distribución, posteriormente puede ejecutarse directamente desde ahí.

PseInt se encuentra en constante desarrollo por lo que siempre es recomendable estar al tanto de las actualizaciones mismas que aportan correcciones de errores y nuevas características o funcionalidades. En su página de contacto⁷ se puede encontrar material sobre pseudocódigo y programación además de contar con la posibilidad de realizar reportes de errores, sugerencias u opiniones.

⁶ <http://pseint.sourceforge.net/index.php?page=descargas.php>

⁷ <http://pseint.sourceforge.net/index.php?page=contacto.php>

Entendiendo el código fuente de los archivos SVG creados con Sozi e Inkscape

Actualmente, los usuarios de Software Libre cuentan con herramientas gráficas como Inkscape y Sozi para la realización de presentaciones en modo visual, cuyo resultado final se basa en código 100% declarativo. En esta edición, Milagros nos introduce en el código fuente de estos archivos SVG y nos abre el camino a lograr una precisión exacta tanto en la dimensión como en la animación de los elementos que componen la presentación.

Escrito por: Milagros Alessandra Infante Montero (Est. Ing. Informática)



Estudiante de Ingeniería Informática. Miembro de la comunidad de software libre **Lumenhack**. Miembro del equipo de traducción al español de **GNOME**. Apasionada por el desarrollo de software, tecnología y gadgets. Defensora de tecnologías basadas en software libre y de código abierto.

Webs:
Blog: www.milale.net

Redes sociales:
Twitter: [@milale](https://twitter.com/milale)

Existen muchas herramientas que permiten a los usuarios de Software Libre realizar presentaciones, pero muchas de ellas, les terminan resultando un tanto aburridas y les quitan la libertad de plasmar sus ideas y creatividad a un cien por ciento. Sin embargo, los usuarios han logrado hallar una solución para ello, mediante una combinación que les permite crear divertidas presentaciones añadiéndole un toque personal y además causando mucho impacto en quien las vaya a ver. Se trata de Inkscape y el complemento Sozi.

Inkscape

Este poderoso editor de gráficos vectoriales es una de las mejores herramientas que existen. Se caracteriza por ser una gran alternativa para lograr resultados increíbles; una de las principales características, es el uso del estándar de la W3C, el formato de archivo de gráficos vectoriales redimensionables (SVG). Este formato es un **lenguaje de marcado para gráficos en dos dimensiones** que nos permitirá manejar imágenes como elementos geométricos vectoriales además de mapa de bits y texto.

SVG permite la redimensión de una imagen al tamaño sin pérdida de calidad.

No está demás indicar que al ser un proyecto open source, si estás interesado en involucrarte y colaborar, puedes hacerlo mediante el reporte de bugs, registrando un blueprint y más⁸.

Sozi

Sozi⁹ es una extensión de Inkscape la cual, en combinación con éste, dan al usuario la oportunidad de hacer magia con las presentaciones. Esta herramienta está basada de estándares abiertos y distribuida bajo licencia GPL 3.0. **Las presentaciones resultantes son archivos SVG**, en los cuales una secuencia de marcos representadas por rectángulos, definirán las regiones que contendrán todos los elementos que sean colocados en el espacio de trabajo.

Se puede decir que Sozi es un póster donde se colocan elementos libremente y luego de ello, mediante acercamientos, rotaciones y demás, se puede enfocar todo lo que se desee. Y adivinen, ¿qué característica le da otro plus a esta extensión?, pues, que desde cualquier navegador compatible¹⁰ puedes visualizar la presentación (sin necesidad de instalar ningún complemento más). Nuevamente, al ser un proyecto libre y si te sientes interesado por contribuir, no dudes en involucrarte, reportando un problema o proponiendo una nueva característica¹¹.

Instalación

Lo primero por hacer es instalar el editor Inkscape y la extensión sozi por separado y en dicho orden, para evitar conflictos de dependencias:

```
sudo apt-get install inkscape
sudo apt-get install sozi
```

Si se desea probar la realización de una presentación sencilla a modo de ejemplo, una vez abierto Inkscape, se accede al menú Archivo → Nuevo → desktop_800x600 para comenzar la creación.

A pesar de que el SVG final tiene como resultado una sucesión de diapositivas, el trabajo se realiza en una sola hoja. Es importante que contemos como mínimo con dos capas, ya que una de ellas contendrá todos los elementos (gráficos, textos, etc) y la otra capa será la de control o animación. Lo que esto permite, es separar el plano donde se tiene los objetos y por otro lado, los rectángulos que van a representar cada una de las pantallas.

- **Capa de contenido:** Después de colocar todos nuestros elementos se debe ir al menú Trayecto → Objeto a trayecto, lo que significa que todo se ha convertido a vector (path). Ya que si no se realiza este cambio no se podrá visualizar, por ejemplo, si el texto no se convierte a vector, se verá la pantalla de animación pero no lo que dice. Debo recalcar algo muy importante, cuando inkscape convierte a vector a los elementos, no importa el grado al que nos acerquemos, lo realmente importante es que ya es un vector y ya no podemos editar el texto (si fuera el caso), si luego se desea editar esos elementos, nos llevamos la sorpresa de que ahora se editan como imágenes y podemos darle algunas formas que simulen un cambio o efecto.
Colocar todos los elementos que se quieran mostrar y finalizado ello, empezar a añadirle animación.
- **Capa de control o animación:** Se empieza a marcar el orden de las diapositivas con recuadros (rectángulos de preferencia), se debe colocar tantos rectángulos como número de vistas se desee. Se guardará cada cuadro como si fuera una

⁸ <https://launchpad.net/inkscape>

⁹ <http://sozi.baierouge.fr/wiki/en:install>

¹⁰ <http://sozi.baierouge.fr/wiki/en:browsers>

¹¹ <https://github.com/senshu/Sozi/issues>

diapositiva nueva y es aquí donde Sozi empieza en acción. Iremos al menú Extensiones → Sozi, aparecerá un diálogo que diga: "'Sozi' en proceso, espere, por favor..." y una ventana en la que se debe rellenar los datos necesarios para la animación (juguemos con cada dato para probar en que se diferencia cada perfil y que "efecto" obtenemos).

El diálogo: "'Sozi' en proceso, espere, por favor...", aparece ya que el sistema de extensiones de Inkscape está diseñado por extensiones "Efecto". Un efecto se implementa como un script que transforma el documento y devuelve el resultado a Inkscape. Es decir, que desde que el proceso de transformación empieza, el diálogo se mostrará hasta que el script finalice.

Sozi nos permitirá también incrustar videos o audio. Esto se encuentra en el menú Extensiones → Sozi extras → Agregar video o audio. Debemos tener cuidado que al momento de elegir esta opción no se encuentre ningún elemento seleccionado, ya que se creará uno nuevo al marcarla.

Edición de archivos SVG

Al abrir el archivo .svg con un editor de texto, podemos darnos cuenta que todo el código que se ha generado es completamente lenguaje declarativo, donde están todas las instrucciones de Sozi y todas las instrucciones de la capa de control y por lo tanto obtenemos un código legible. Es por ello, que Sozi, al estar basado en XML y Javascript, nos da la posibilidad de editar la presentación en un editor de texto.

La hoja en la que se basa la presentación, tiene un aspecto similar al siguiente:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg
  <!--
    Los espacios de nombre XML que se declaran a continuación
    son recomendaciones de la W3C que proporcionan elementos y
    atributos (con nombre único)
  -->
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:cc="http://creativecommons.org/ns#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:svg="http://www.w3.org/2000/svg"
  xmlns="http://www.w3.org/2000/svg"
  xmlns:sodipodi="http://sodipodi.sourceforge.net/DTD/sodipodi-0.dtd"
  xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape"

  <!-- Se indica el ancho y altura de la página -->
  width="800px"
  height="600px"
  id="svg2985"
  version="1.1"
  inkscape:version="0.48.4 r9939"
  sodipodi:docname="hdmagazine_hoja">
<defs
  id="defs2987" />
<sodipodi:namedview
  id="base"
  <!-- se indican los colores y detalles del border -->
  pagecolor="#ffffff"
  bordercolor="#666666"
  borderopacity="1.0"
  <!-- se empieza a declarar la opacidad, sombra, zoom, ejes -->
  inkscape:pageopacity="0.0"
  inkscape:pageshadow="2"
  inkscape:zoom="0.6"
  inkscape:cx="400"
  inkscape:cy="300"
  inkscape:current-layer="capa001"
  inkscape:document-units="px"
  showgrid="false"
  inkscape:window-width="1280"
  inkscape:window-height="741" />
```

```
<metadata
  id="metadata2990">
  <rdf:RDF>
    <cc:Work
      rdf:about="">
      <dc:format>image/svg+xml</dc:format>
      <dc:type
        rdf:resource="http://purl.org/dc/dcmitype/StillImage" />
      <dc:title></dc:title>
    </cc:Work>
  </rdf:RDF>
</metadata>
<g
  id="capa001"
  inkscape:label="capa001"
  inkscape:groupmode="layer" />
</svg>
```

Cada elemento con todas sus características, estará definido de la siguiente forma:

```
<g
  <!--id, nombre y grupo de la capa -->
  id="layer1"
  inkscape:label="capa001"
  inkscape:groupmode="layer">
  <!-- ruta del elemento -->
  <path
    <!-- Ejemplo de una espiral-->
    sodipodi:type="spiral"
    style="fill:none;stroke:#000000; stroke-width:1px; stroke-linecap:butt;
      stroke-linejoin:miter;stroke-opacity:1"
    id="path3012"
    sodipodi:cx="545"
    sodipodi:cy="370"
    sodipodi:expansion="1"
    sodipodi:revolution="3"
    sodipodi:radius="108"
    sodipodi:argument="-18"
    sodipodi:t0="0"
    d="m 545,370 c 4.1782,3.50969 -2.66961,7.36369 -5.83333,6.94444 -8.57349,-1.13612
-10.86246,-11.7921 -8.05556,-18.61111 5.02088,-12.19759 20.57591,-14.928 31.38889,-9.16666
15.86849,8.455 19.10138,29.47428 10.27778,44.16666 -11.76049,19.58264 -38.41273,23.31994
-56.94445,11.38889 C 492.51307,389.7082 488.27162,357.35209 503.33334,335 c 18.24176,-27.07137
56.33818,-31.81691 82.49999,-13.61111 30.83079,21.45491 36.08052,65.31286 14.72222,95.27777
-24.659,34.59568 -74.29197,40.34974 -108.05555,15.83333 -38.36432,-27.8571 -44.62289,-83.27418
-16.94444,-120.83332 31.05104,-42.13568 92.25868,-48.8989 133.6111,-18.05555 45.90903,34.24196
53.17705,101.24487 19.16667,146.38888"
    transform="translate(61.666667,61.666667)" />
  </g>
```

Dicho elemento es animado usando sozi y para ello, el script generado por sozi es bastante largo y debemos recordar que se mantiene activo mientras se hagan las modificaciones de estilos.

```
<style
  ns1:version="12.06-13110110"
  id="sozi-style">#sozi-toc text{fill:#eff;font-family:Verdana,sans-serif;font-
size:12pt}#sozi-toc text:hover{fill:#0cf;cursor:pointer}#sozi-toc text.sozi-toc-
current{fill:#fa4}#sozi-toc-background{stroke:#222;stroke-opacity:.1;stroke-
width:10;fill:#222;fill-opacity:.9}.sozi-toc-arrow{fill:#eff;fill-
opacity:.75;stroke:none}.sozi-toc-arrow:hover{fill:#0cf}#sozi-framenumber
circle{stroke:#222;stroke-opacity:.1;stroke-width:4;fill:#222;fill-opacity:.9}#sozi-
```

```
framenumber text{fill:#eff;font-family:Verdana,sans-serif;font-size:12pt}</style>

<ns1:frame
  <!-- se indica el perfil de animación, el tiempo de espera, el zoom,
        la duración de la transición, etc. -->
  ns1:transition-profile="strong-accelerate-decelerate"
  ns1:timeout-enable="false"
  ns1:timeout-ms="2000.0"
  ns1:transition-zoom-percent="30.0"
  ns1:title="hdsozi001"
  ns1:clip="true"
  ns1:hide="true"
  ns1:transition-duration-ms="3000.0"
  id="frame1"
  ns1:sequence="1"
  ns1:refid="path3012" />
```

De esta manera, se añaden los objetos y animaciones efectuadas con sozi. Quizás para los script de sozi, pueden ayudarse del que aparece al correr la extensión en Inkscape para empezar a estudiarlo un poco y posiblemente, también hacerle modificaciones para mejores y más avanzados resultados.

Si se llega a realizar todo el ejemplo desde el código, la ventaja principal con la que se contará, es exactitud. Por ejemplo, al momento de haber definido valores y no simplemente calculado a pulso en la interfaz, la precisión puede ser un factor de diferenciación, sobretodo cuando el diseño que se realizará es mucho más profesional. En Google Groups, pueden encontrar a sozi-users¹², donde los montones de usuarios llegan a realizar sus comentarios, dudas, mejoras y demás.

Pueden abrir su navegador un ejemplo sencillo que realicé con Inkscape y Sozi, accediendo desde el siguiente enlace de mi blog: <http://milale.net/wp-content/uploads/2013/10/milagros.svg>

Para trasladarnos por las vistas, pueden pulsar la tecla [enter], el botón derecho del ratón, desplazarlos con las flechas adelante y atrás, o escribir el número del *slide* en la dirección URL del navegador (después del símbolo #).

¹² <https://groups.google.com/forum/#!forum/sozi-users>

HACKERS & DEVELOPERS MAGAZINE

PRESENTA



mozilla
zone

con el apoyo de:



mozilla
FOUNDATION

La importancia de estar documentado – Mozilla Developer Network

La consulta de una buena documentación es fundamental para el correcto aprendizaje de cualquier tecnología, algo que muchas veces se pasa por alto cuando se trabaja para la web.

Escrito por: **Pablo Bernardo** (Miembro Colaborador de la Fundación Mozilla)



Desarrollador Web especializado en GNU/Linux, PHP, HTML, CSS y JavaScript. Director y presentador del programa **#linuxIO** en **DesarrolloWeb.com**. Miembro Colaborador de la Fundación Mozilla y traductor en la **Mozilla Developer Network**. Instructor de PHP en **escuelaIT**. Usuario y promotor de GNU/Linux. VIM lover.

Webs:

Web personal: www.elkarmadelteclado.com

Redes sociales:

Twitter: [@voylinux](https://twitter.com/voylinux)

Como desarrollador o diseñador web, te ves en la necesidad de trabajar casi a diario con tecnologías o características de los lenguajes con las cuáles nunca lo habías hecho antes. Sales a buscar información por internet pero quizá, no le estés prestando la importancia necesaria a esta parte del proceso. Estudiar correctamente en cada fase del aprendizaje, marcará en el futuro la diferencia entre ser un auténtico profesional o solo alguien que pasa el día “sobreviviendo” entre líneas de código.

Ser conscientes de la importancia que tiene para un desarrollador el proceso de estudio y documentación, es una cuestión de responsabilidad.

La verdadera independencia de un desarrollador pasa por un proceso imprescindible: un aprendizaje correcto.

Aprender correctamente, leer documentación

Tiene que acabarse la costumbre de copiar y pegar código y solo cambiar algunos datos. Debe ser sustituida por un esfuerzo consciente de entender cada cosa que estamos haciendo. Internet está lleno de malos ejemplos, pero la buena documentación también existe en abundancia y está ahí disponible para ayudarte a trabajar mejor. Solo hay que saber dónde buscar y todo puede ser

más fácil.

Cuando se trata de tecnologías web, yo soy un gran partidario de los tutoriales y el auto aprendizaje, pero suelo recomendar que primero se consulte documentación para entender realmente de qué hablamos y qué podemos hacer con ello. Después podemos estudiar todos los ejemplos necesarios y hacer todas las pruebas.

Es fundamental que aquellas webs que se conviertan en una referencia para ti, sean rigurosas, amplias, accesibles, estén vivas y en continua ampliación y actualización.

¿Qué es Mozilla Developer Network?

Siempre recomiendo a mis alumnos que sean cuidadosos con el lugar del que extraen la documentación y en concreto suelo remitirles a documentarse en [Mozilla Developer Network](https://developer.mozilla.org/es/)¹³ (MDN). La mayoría de las veces, la respuesta que me suelen dar al cabo de solo unos días de consultarla es: "Es increíble, hay de todo".

Mozilla Developer Network es la propuesta de Mozilla para ofrecer un lugar donde todo el mundo pueda acceder a documentación sobre tecnologías web y aprender a crear. Al contrario de lo que mucha gente piensa al oír hablar de esta web, Mozilla Developer Network no es solo una lugar en el que aprender sobre productos y tecnologías de Mozilla. Se trata de una extensísima Wiki sobre tecnologías web y estándares web. Como parte del esfuerzo de la Fundación Mozilla por mantener la web libre y accesible y el conocimiento como algo abierto, en MDN te ofrecemos la posibilidad de aprender sobre [HTML](https://developer.mozilla.org/es/docs/Web/HTML)¹⁴, [CSS](https://developer.mozilla.org/es/docs/Web/CSS)¹⁵, [JavaScript](https://developer.mozilla.org/es/docs/Web/JavaScript)¹⁶ y además, por supuesto, sobre muchas otras tecnologías y características abiertas propuestas por Mozilla.

Todo lo que en MDN tenemos para ti

La documentación incluida en MDN propone un aprendizaje e implementación basados en las buenas prácticas. No se trata de una web donde aprender el último *hack* para hacer funcionar algo en un navegador concreto, sino acerca de las bases para trabajar con cada lenguaje de manera correcta. Tal como ha sido creado y como debe ser utilizado para ser accesible al mayor número de usuarios posible.

MDN se construye en forma de Wiki, por lo que cualquier persona que lo desee puede contribuir. Y esto no es una utopía, sino algo totalmente real que lleva años funcionando y se mantiene por una comunidad enorme de voluntarios que llevamos a cabo diferentes tareas para hacerlo realidad. Todo el trabajo de este equipo, unido por la filosofía y valores de Mozilla son una garantía de que MDN es una documentación viva, en constante crecimiento, actualización, mejora y traducción. Por ejemplo, puedes consultar documentos en MDN en más de treinta lenguas, incluido el español, por supuesto.

En Mozilla tenemos claro que lo que queremos es que todo el mundo pueda aprender a crear para la web. Por eso la documentación de MDN puede llevarte a través de todos los pasos del aprendizaje. Desde la documentación más técnica, hasta los tutoriales paso a paso y las demostraciones en línea.

Algo que te gustará mucho de MDN es que, a pesar de tratarse de documentación rigurosa, su lectura no resulta árida. Por el contrario es directa y clara. Encontrarás un resumen explicativo del apartado, sintaxis, ejemplos de código y notas aclaratorias para evitar ambigüedad o errores de implementación. Así mismo verás también que habitualmente se enlaza a la especificación oficial, así como a otros documentos tanto dentro como fuera de MDN que complementen la información ofrecida.

Además MDN, como todo en Mozilla, es abierto. Puedes ver qué autores crearon el artículo o cuando y por quién fue actualizado o incluso aportar acerca de errores o posibles mejoras.

¹³ <https://developer.mozilla.org/es/>

¹⁴ <https://developer.mozilla.org/es/docs/Web/HTML>

¹⁵ <https://developer.mozilla.org/es/docs/Web/CSS/CSS3>

¹⁶ <https://developer.mozilla.org/es/docs/JavaScript>

Conclusión

Haz un esfuerzo por ir a buscar documentación allí donde encuentres información de valor y no solo en los primeros resultados que encuentres en tu buscador. Un truco que yo utilizo es dejar una pestaña fija en Firefox con MDN de manera que tengo allí un manual preparado a solo dos teclas de distancia.

Te invito como desarrollador a que pases unos días utilizando MDN como material de consulta. Te resultará fácil familiarizarte con la manera intuitiva en que se organiza el contenido y seguramente se convierta en un recurso básico para ti. Verás que pronto empezarás a disfrutar del aprendizaje que se deriva de una base bien construida.

Si te gusta MDN puedes ayudar recomendándole su contenido a otras personas y ayudando a promocionar el uso de buena documentación. Y si aun así quieres más, hay [muchas maneras de contribuir con MDN](#)¹⁷, no solo programando. Te animo a que [te unas a la comunidad de Mozilla Hispano](#)¹⁸ y participes de la forma que consideres más adecuada. Colaboradores encargados, se pondrán en contacto contigo par guiarte en el proceso.

*Ayúdanos a crear una web libre, accesible y segura para todos en la
poder trabajar, aprender, disfrutar y comunicarnos. Utiliza y
promociona la documentación de MDN.*

Tu saldo de **PayPal**

cóbralo **desde cualquier parte del mundo**

- ✓ Tarjeta de débito prepaga **MasterCard**
- ✓ **Compras** con tu tarjeta alrededor del mundo
- ✓ Extracción de **dinero en efectivo** desde Cajeros Automáticos
- ✓ **Cuenta bancaria virtual en USA**
(para transferir el dinero desde PayPal)

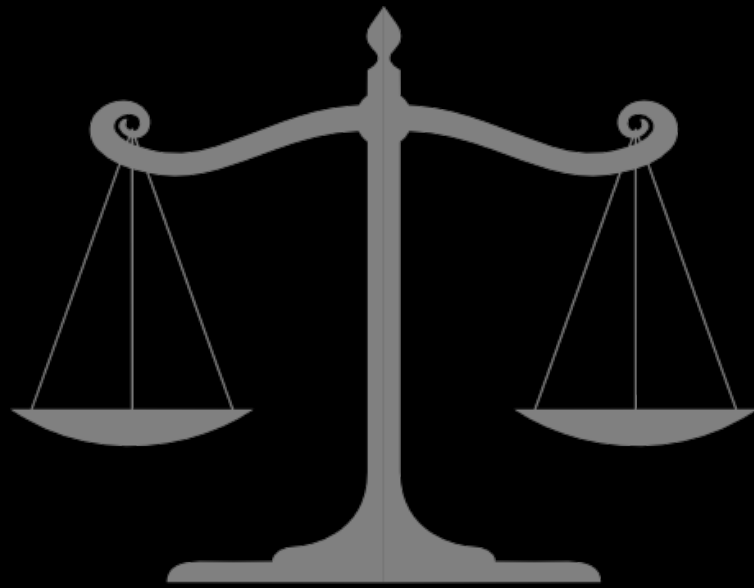
**Regístrate ahora y recibe USD 25.- de regalo
con tu primera carga de USD 100.-**

Clic aquí

Payoneer®
MasterCard

¹⁷ https://developer.mozilla.org/es/docs/Project:MDN/Contribuyendo/Como_ayudar

¹⁸ <https://www.mozilla-hispano.org/documentacion/Colabora>



INFORMÁTICA
& *Derecho*

El software y el Derecho

Se cumplen 30 años desde que el proyecto GNU vio la luz en un mundo de oscuras restricciones a la naciente cultura informática. Mucho debemos a los movimientos liberadores del software; potenciaron la creatividad, generaron innovaciones a ritmos acelerados y, entre otros logros, han creado la necesidad de replantear la concepción inmaterial sobre todo lo que el ser humano ha creado desde su imaginación, regido hasta el día de hoy por un sistema de normas de espíritu arcaico y restrictivo: el Derecho de Autor.

Escrito por: **Silvio Messina** (Desarrollador Web - Est. de Derecho, Universidad Nac. de Buenos Aires)



Silvio es **Desarrollador, Profesor en Tecnologías Web, Wordpresero y estudiante de Derecho** orientado al estudio del Copyright, Propiedad Intelectual, Software y Cultura Hacker. Defensor de la Educación abierta, el auto desarrollo intelectual y el libre intercambio.

Webs:

Web personal: www.silviomessina.com.ar

Redes sociales:

Twitter: [@smessina_](https://twitter.com/smessina_)

Homenajeando los 30 años del proyecto GNU, presentamos esta serie de artículos en donde se explicarán las características, elección, uso y aplicaciones de las licencias de Software Libre compatibles con la General Public Licence (GPLv3) de GNU. En esta primer entrega, antes de aventurarnos en el estudio de las diversas licencias aplicables al desarrollo de software y afines, nos parece prudente iniciar esta serie de artículos haciendo una introducción a los conceptos jurídicos que se aplican y discuten en la temática sobre derechos de autor.

Aclarando el panorama

La industria del Software es relativamente la más nueva de las corrientes creadoras de los denominados "productos intelectuales", es decir aquellas cosas que surgen de la combinación de imaginación, conocimiento, azar y lógica constructiva. Así como la música y las obras literarias son productos ideales (es decir que nacen de las ideas), para que estos circulen en el mundo real, son plasmados en formatos físicos que luego son distribuidos al público.

El derecho de autor aparece en la historia de los seres humanos como una forma de reconocer la propiedad del autor (es decir, ser dueño) del valor intelectual que la obra conlleva en su formato de distribución.

Aquí aparecen algunos términos que es conveniente definirlos:

- **Derecho:**
normas que regulan la conducta de los seres humanos, reconociendo los límites y potestades que éste posee sobre sí mismo, los terceros y los objetos.
- **Obra:**
es el producto resultante del trabajo artístico, lógico, científico y/o intelectual.
- **Autor:**
es toda persona que crea una obra.
- **Propiedad:**
es el derecho de ejercer el poder sobre un objeto o un bien, otorgando la capacidad de disponer del mismo, con los límites que la ley imponga.

El derecho de autor aparece formalmente en el siglo XVIII, concretamente con el Estatuto de la Reina Ana de 1710 del Reino Unido; aunque si bien existen otros antecedentes, fue en este punto donde formalmente se le da el reconocimiento a las obras, de ser protegidas bajo el derecho de propiedad. Previo a esto, las obras eran libres de ser reproducidas; pero tomando en cuenta que el acceso a la cultura y la educación durante casi toda la historia de la humanidad no fue de carácter universal, las obras constituían un rol poco activo en las sociedades, siendo incluso un producto casi folclórico de las altas clases sociales.

La aparición de la imprenta generó lo que muchos autores reconocen como la "Era de la Razón", constituyendo un medio de reproducción masivo de las obras literarias y permitiendo que estas llegaran a más gente sin distinción de su clase social. Aquí es donde también hay un quiebre en la mentalidad de los autores, denunciando que la imprenta permitía que sus obras fuesen reproducidas y comercializadas por terceros, sin que ellos pudieran participar ni percibir ganancias de dichas transacciones. Se empezó a exigir que las creaciones intelectuales fuesen reconocidas como productos de igual envergadura que los bienes materiales, demandando que los mismos derechos de ejercer la propiedad fuesen aplicables a las obras.

Copyright: el candado de la cultura

El derecho anglosajón lo reconoció como Copyright, cuya traducción literal es "Derecho de copia" y que si bien hoy es admitido como sinónimo de Derecho de Autor, el Copyright es solo un derecho inherente dentro de todo el marco del Derecho de Autor.

No obstante, el correr del tiempo ha llevado a asociar al derecho de autor como el derecho de lucro por sobre la obra creada, privilegiando el valor económico por sobre el valor cultural y académico que las obras llevan en sí misma; el cual se fue moldeando por la aparición de nuevas formas de plasmar las obras en formatos que la sociedad empieza a "consumir", formatos impulsados por industrias que, sin desmerecer las innovaciones tecnológicas que aportaron al mundo, han mantenido en un círculo cerrado los medios para producir nuevas obras.

El software aparece en el plano jurídico al mismo nivel que los inventos y las obras científicas, pero que dada sus cualidades, fue dotando la concepción del producto ideal con nuevos puntos de vista.

Los dos puntos más controversiales son:

- **Software como producto:**
esta concepción reconoce al software como una obra completa de la que se puede tener disposición completa sobre su funcionamiento, sus prestaciones y su composición.
- **Software como medio:**
en esta el software es un servicio que se presta bajo los términos de uso establecidos, determinando los límites y alcances de su utilización.

Entre ambos extremos hay un sin fin de combinaciones, como una balanza que oscila entre las libertades otorgadas al usuario y los límites determinados por el autor.

Software: el abstracto motor del progreso

La ciencia informática es caracterizada por su naturaleza de crecimiento dinámico, acelerado y meramente intelectual y la implementación del Copyright sobre los logros que ésta produce, conlleva a una limitación a la disponibilidad y eventual evolución. No es casualidad que casi todo el software existente se haya gestado con prácticas que están por fuera del Derecho de Autor, prácticas que se convirtieron en iconos de la cultura *Hacker*.

Esta dicotomía fue la que inspiró a Richard Stallman para crear una alternativa al sistema operativo Unix, dando nacimiento al proyecto GNU. Pero más allá del "producto", GNU es la corriente reivindicatoria de que se puede desarrollar, distribuir y evolucionar por vías alternativas a las leyes del Copyright. Gracias al proyecto GNU y a otras corrientes liberadoras del software, los Hackers, hoy contamos con herramientas legales que nos permiten continuar con las prácticas de libertad y, al mismo tiempo, dotar a las obras de una protección jurídica acorde a su valor intelectual.

En algún punto, el proyecto GNU y los demás proyectos afines han vuelto a poner en discusión el concepto de "Derecho de Autor" y, es por ello que celebramos este clima de cambio explicando los aportes jurídicos que el proyecto GNU ha generado en el Derecho.

En las siguientes entregas, se publicarán una serie de artículos dedicados a las Licencias del Software que se sustentan en las prácticas de libertad y colaboración. Como adelanto de lo que se viene, la primera revisión, será sobre la licencia "Madre" que configura el estándar de libertad en materia de software: la licencia GNU GPL v3.

¡Nos vemos en la próxima entrega!



HECHO POR HACKERS DE TODO EL MUNDO PARA PROFESIONALES DE TODO EL PLANETA
PRODUCIDO EN LA REPÚBLICA ARGENTINA

