MAGAZINE GNU/LINUX cotej

Año 02 - N° 05

HONEYPOT
a la caza de los Piratas Informáticos

PYTHON
Curso Parte II

Curso Parte II
php

Prefiero GNU/Linux

ENTREVISTA
Juan Carlos Karroum
Charla con nosotros

ANDROID EL FUTURO DEL MOVIL.

TIPS UBUNTU

OPENVPN

PROYECTO INFOMOVIL

safecreative
PROTEGE TU OBRA

YoSiUso Software Libre

Revista digital bimensual

Staff:
Juan C. Karroum (JCK)
Héctor A. Mantellini (Xombra)

Colaboradores:
Naudy Villaroel
John Vera
AWVEN
Carlos D. Ortiz
Eduard Lucena
Eduardo Echeverria

Herramientas usadas:
Inkscape 0.48.1
Scribus 1.3.9

Editorial

En lo que va de año ha sido de bastante ajetreo, han sido eventos tras eventos, charlas tras charlas y nos da beneplácito que todas las comunidades de software libre hablan de nosotros como la comunidad en Venezuela que más ha trabajado en la difusión del software libre.

Hace unos días celebramos el Debian Day (18 años de Debian), específicamente el día 13-08-2011 donde asistieron alrededor de 130 personas durante todo el evento.

EL día 17/09/2011 estaremos en Puerto La Cruz compartiendo con el Grupo CUSLAnz de esa región del país el Software Freedom Day, estan cordialmente invitados.

El compañero @Roliverio configuró a Wadameka (nombre de nuestro servidor) con IPv6 y zona de resolución de nombres inversos, excelente trabajo!

Hemos recibido ayuda de nuevos e importantes patrocinadores: INCES, Valencia del Rey, Rcp Suministros, VIPMaxx, además Venehosting continua con nosotros.

Gracias especiales a todos y cada uno de las personas e instituciones que nos han apoyado.

Quieres aprender y apoyar el movimiento del software libre, solo contáctanos.

Participa con nosotros, unete a la Fuerza!

Publicidad

CURSOS Y TALLERES

XHTML1.0 & CSS

GNU/Linux Básico

PHP y MySQL

CANAIMA Educativo

OpenOffice / LibreOffice

Interesados dirigirse a:
Unidad de Computación.
Preguntar por: Lic. Luisa Ochoa

Dirección: Av. Universidad - Naguanagua

Entrevista realizada a uno de los fundadores del Grupo VaSlibre. En esta oportunidad Juan Carlos Karroum, conocido en la comunidad como jck77.

1.- Como conociste el Software Libre?

En la universidad un compañero de clases me hablo sobre el software libre, luego de unos días estaba comprándome un libro de Red Hat 6 el cual venia con sus discos de la distro. De ahí en adelante se puede decir que fue el principio de una gran aventura que ha sido el Software Libre. Le tendría que dar las gracias a Shay Ohayon quien pudiera decir fue quien en el IRC me terminó de guiar por el camino del Software Libre.

2.- Como fue tu primera experiencia en el uso del Software Libre?

Diría que fue genial y a su vez frustrante, horas en la computadora solo para instalar el OS, hoy día no me arrepiento de todas las horas que pasé tratando de entender el sistema, ya que en ese entonces los métodos de instalación no eran tan amigables como ahora. El simple hecho de que podía instalar cualquier aplicación sin necesidad de pagar o estar buscando un serial o crack fue lo mejor que me pudo haber pasado.

3.- Cual ha sido tu distribución favorita y porque?

De verdad no he sido nunca un usuario o administrador que se va por una distribución favorita. Creo que de las muchas que hay la que más he usado es Debian, lo vengo usando desde la versión Potato. En la Actualidad uso Debian para mis servidores y Ubuntu para mi desktop. Pero hay distribuciones que uso para cosas especificas como las de recovery y las de test de penetración como BlackTrack.

4.- Desde cuando perteneces a VaSlibre, cual es tu opinión del grupo?

Desde el día en que nos sentamos a discutir el nombre del grupo. Para mi VaSLibre siempre ha sido uno de los grupos que siempre ha luchado por crecer como grupo y dar lo mejor en todos los eventos donde se participa. Espero que siga así por siempre y para siempre.

5.- Que opinas del Software Libre en Venezuela?

A pesar de no vivir en Venezuela en la actualidad me parece que el movimiento ha crecido bastante. Espero que se siga luchando por el uso del software libre.

6.- Estas en algún proyecto actualmente en la comunidad del Software Libre?

Si, aparte de poder ayudar a VaSlibre en lo que pueda, en la actualidad participo en el grupo de Ubuntu del estado de Florida USA.

7.- Eres de los puristas y radicales en cuanto al SL (software libre) o eres de los light & Cool?

Sinceramente no creo ser ninguno de los dos. Quizás me atrevería a decir que un poco de las dos cosas dependiendo de la situación.

8.- Que opinas de los eventos que se están haciendo Flisol - CNSL - Encuentro de Comunidades - Cayapas?

Me parece bien que se sigan haciendo eventos grandes a nivel Nacional, pero como siempre lo dije, es mejor preocuparse por áreas pequeñas donde el grupo como tal llegue con mejor facilidad, sin tanto esfuerzo y que el objetivo del grupo se cumpla, esto no quiere decir que el grupo no participe en dichos eventos, pero si que se enfoque más en lo regional. Pero en general me parece excelente la organización en los eventos.

9.- Tienes algún blog, escribes con frecuencia?

En la actualidad no tengo blog, lo que significa que no escribo con mucha frecuencia. Pero cuando tengo algun documento o algún howto lo comparto mediante el medio que sea.

10.- Eres Fans de Richard Stallman?

No, y me imagino que no tengo que explicar el porque.

Gracias a todos y Saludos a toda la comunidad del Software Libre

Juan Carlos Karroum (jck77)

Gracias por tu tiempo jck77, exitos en tus proyectos.

Software Freedom Day
Puerto La Cruz

CUSLAnz, ha invitado cordialmente a VaSlibre a participar en el Software Freedom Day en Puerto La Cruz, el día 17-09-2011.

Lugar
Escuela Técnica Industrial Eugenio Mendoza, Avenida José Antonio Anzoátegui

Horario
8:30am a 12:00pm y 2:00pm a 6:00pm

Comunidades o grupos involucrados
- CUSLAnz
- AuLA Libre
- Ubuntu-ve
- Fedora Venezuela
- Tuxinfo
- Vaslibre
- Activistasxsl
- Grulica

http://wiki.softwarefreedomday.org/2011/

Tips: Ubuntu

En esta nueva sección se presentará diferentes trucos que aumentan el rendimiento de nuestros Ubuntu o derivados por ejemplo Linux Mint.

1.- Optimización de Discos: Optimizaremos el performance de nuestro disco duro con hdparm:

Desde la consola o terminal

sudo aptitude install hdparm

Una vez instalado

hdparm -q -W1 -M254 -A1 /dev/sda

Luego, debemos modificar la carga cuando iniciamos el sistema operativo, modificamos el siguiente archivo:

sudo gedit /etc/rc.local y añadimos

hdparm -q -W1 -M254 -A1 /dev/sda

Guardamos y listo!

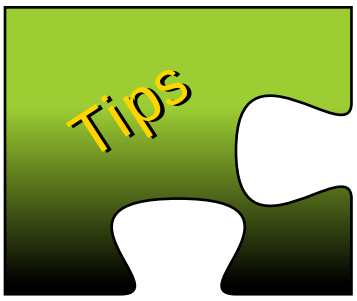
2.- Modem 3G: Usualmente cuando usamos algún modem 3G este no es soportado por nuestro sistema operativo, para minimizar esto realicemos esto:

Se deberán instalar los siguientes paquetes:

sudo aptitude install network-manager network-manager-gnome modem-manager mobile-broadband-provider-info usb-modeswitc

Con esto ya no debería presentar problemas

3.- Cacheando DNS: Esto aumentará considerablemente el rendimiento mientras navegamos por la Internet:



Instalamos dnsmasq

sudo apt-get install dnsmasq

Una vez instalado:

Editamos esolv.conf

sudo gedit /etc/resolv.conf

Y cambiamos el contenido por el siguiente

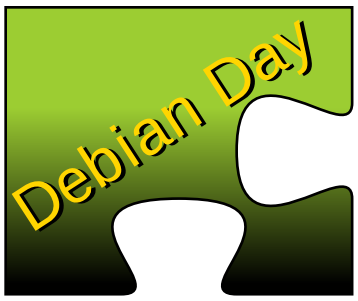
#local
127.0.0.1
#DNS de Comodo
156.154.70.22
#DNS de OpenDNS
208.67.220.220

Guardamos y ejecutamos

sudo /etc/init.d/networking restart

En próximas ediciones iremos colocando sencillos tips de herramientas y/o configuraciones para nuestra distribución.

Artículo de:
VaSlibre
http://www.vaslibre.org.ve
Twitter: @vaslibre
Identi.ca: vaslibre



Debian Day en Valencia

El Debian Day (Día Debian) es celebrado cada 16 de agosto en diferentes países y ciudades. VaSlibre lo celebró el día 13 de agosto de 2011 en las instalaciones del INCES de los Colorados.

Este evento conmemora la Fundación del Proyecto Debian (*Debian Project*) en el año 1993 por Ian Murdock, el proyecto ha crecido de un grupo reducido de hackers y se ha convertido en una comunidad mundial dedicada al mantenimiento y desarrollo de este poderoso y versatil sistema operativo.

Entre los colaboradores para este importante evento estuvieron presentes: awven.com, Ubuntu VE, rcpsuministros.com, venehosting.com, valenciadelrey.com.

Las charlas que se hicieron ese día fueron:

- Debian y derivados por *Angel Cruz (Abr4xas)*
- Implementaciones de Debian por *Jesús Palencia (SinFallas)*
- Licencias en Debian por *Héctor A. Mantellini (Xombra)*
- Hacking Kernell por *Julio Ortega (Roliverio)*
- Yo sí uso Ubuntu a Diario por *Nelson Delgado (nejode)*

En esta oportunidad VaSlibre y Ubuntu Carabobo unieron fuerzas y lograron el exito que se esperaba.

Asistieron aproximadamente 130 personas que disfrutaron de las ponencias y rifas que se efectuaron.

En Venezuela se celebró en Valencia - Barquisimeto - Merida y Caracas.

En Valencia eL diario El Carabobeño nos abrió sus puertas para promocionar el evento, al igual que Unión Radio, en varias de sus cadenas de emisoras.

Patrocinantes

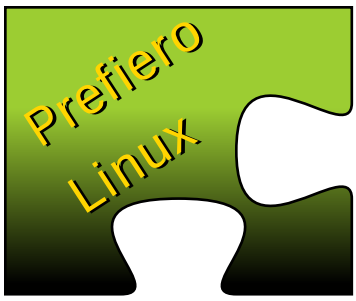
Prefiero GNU/Linux

Pequeñas cosas que te llevan a probar algo diferente, más que diferente... algo libre.

Microsoft Windows como todos conocemos desde pequeños algunos y desde ya mayores otros, siendo o no el caso, es que lamentándolo mucho es el sistema más comercial y usado por una gran mayoría de usuarios en el mundo, pero de ¿verdad es bueno? ¿es confiable? ¿Tiene todo lo que necesito? Para algunos la respuesta es si, tal vez para otros no, en el circulo de la tecnología no debemos ser conformistas, claro teniendo en cuenta que hoy adquieres un hardware y ya mañana es completamente obsoleto; sin embargo, debemos ver más allá de lo que en verdad son las cosas y probar nuevas alternativas.

Vale probrar algo que siendo tecnológico y estable se haga familiar y divertido, hacer que tu computador donde trabajas o estudias y pasas tu tiempo, sea en realidad algo libre. Eso te ayudaría a sentirte libre a eso es lo que me refiero con usar Software libre.

En lo personal a mi me llevo a realizar el cambio o la alternativa "*Windows -> Linux*" más que curiosidad me llamó la atención la poca probabilidad de virus, olvidar la idea de un "*crack*" o serial para un software determinado, esos molestosos anuncios advirtiendo que tal licencia venció, de no saber que ocurría en mi máquina, de pagar por esto o por lo otro. Ojo no quiero decir con esto que Linux sea gratis, es sencillamente libre más no gratis, puedes pagar por tu licencia en aplicaciones y distribuirlas y seguirá siendo original cosa que en Windows es otro tema más grande.



Las gran variedad de aplicaciones y/o paquetes que ofrece Linux en todas sus diferentes variantes conocidas como distribuciones no tienen nada que envidiarle a las de Windows, por ejemplo Ubuntu posee 29 mil aplicaciones en sus repositorios, es más para el punto de vista de muchos son mejores como por ejemplo en caso de libreoffice puedes agregar memoria RAM para que esta funcione más óptimo no tanto las aplicaciones el sistema como tal es mejor porque trabaja mejor los archivos, tienes un solo usuario administrador puedes tener ejemplo.txt , eJemplo.txt , ejEmplo.txt en un mismo directorio y ser diferentes archivos. Me gusta también como maneja los procesos y los recursos de memoria.

Linux no depende tanto de hardware como en Windows aqui no veras designado para x versión porque Linux no trabaja de esa manera lo puedes instalar en una calculadora, en un smartphone sin irnos muy lejos y va a funcionar ..¡sin exagerar!, te olvidarás de ese repetido Alt,Ctrl,Supr que era necesario para liberar procesos en Windows en tal caso Linux lo maneja con el comando top en el cual puedes ver/manejar los procesos que están siendo llevados por el sistema.

Yo como analista lo recomiendo y ya estoy haciendo cosas más avanzadas, ya no es mi primera vez, ya estoy conociendo a Linux... estoy usando Ubuntu.

Artículo de:
Carlos David Ortiz
Twitter: @Ccorleon3

Phenobarbital con Soda
Blog de Jesús Lara importante pieza del Software Libre en Venezuela. Tips de Debian, Canaima y otras distribuciones
http://phenobarbital.wordpress.com

Mozilla Hispano
Comunidad en español de Mozilla
http://www.mozilla-hispano.org/

El Atareado
Blog dedicado a Ubuntu y al software libre relacionado con la ingenieria. Aplicaciones informaticas
http://www.atareao.es/

Proyecto Infomóvil

La Fundación Infocentro cuenta con varios proyectos que activa la participación de los diferentes colectivos organizados, uno de estos es el Infomóvil. tiene como objetivo principal llegar a las comunidades ubicadas en aquellos lugares de difícil acceso, sea por razones geográficas, como las comunidades indígenas de la Selva Amazónica y del Delta, los caseríos de la Sierra de Falcón, los pueblos profundo de los Llanos Venezolanos, las pequeñas comunidades ubicadas en la Cordillera Andina, o por otras razones sociales como los centro de privación de libertad y espacios para personas discapacidad, generando así la formación y capacitación a través de Tecnologías de Información y Comunicación (*TIC*) con el fin de contribuir a la consolidación de poder Comunal.

Infomóvil cuenta con 1 coordinador nacional, 4 coordinadores regionales y 48 entre promotores y promotoras dispuestos a llevar la formación socio tecnológica a las comunidades en cualquier parte del país.

Actualmente el Infomóvil cuenta con las 28 unidades terrestres, cada una tiene entre 10 y 14 computadores con contenidos multimedia y conexión a Internet mediante el satélite Simón Bolívar, como ejemplo de la soberanía nacional, donde más de 700 mil personas están aprendido a utilizar el computador.

¿Qué Realiza el proyecto Infomóvil?

Con las unidades móviles se realizan actividades culturales, deportivas y formativas; como la proyección de video educativos, foros y alfabetización tecnológica en las comunidades. A través del infomóvil se organizan operativos y planes de contingencia, la movilidad y la conectividad resultan ser elementos bastante estratégicos.

Las tomas socio tecnológicas en la comunidades ya son un referente entre las diversas actividades desarrolladas por la Fundación Infocentro. Estas tomas son organizadas con el pueblo, días previos a la llegada del Infomóvil. Los coordinadores del Infomóvil junto a los líderes de las comunidades, los Consejos Comunales y Comunas, en asambleas de campesinos diagnostican las realidades sociales y planifican las actividades que desarrollaran durante los días que dura la toma. (*Se llama "tomas" cuando el Infomóvil se dirige a una zona del territorio nacional).*

El Equipo de Infomóvil trabaja como un solo bloque, presto a recorrer el país sin importar condiciones atmosféricas o geográficas, para llevar las Tecnologías de la Información y Comunicación a cada Venezolano y Venezolana disminuyendo la exclusión del uso de estas tecnologías en los sectores populares.

Para encontrar más información referente al “Proyecto Infomóvil” y la “Fundación Infocentro” más cercano a tu zona de residencia le invito visitar su sitio web en la siguiente url: http://www.infocentro.gob.ve/ o llenar el formulario que está en el siguiente vínculo: http://www.infocentro.gob.ve/atencion.php

Quiero dar un especial agradecimiento al personal de “Fundación Infocentro” por toda la colaboración e información suministrada, la cual está ubicada en la Av. Universidad, Esquina El Chorro, Torre MCT, Piso 11 Fundación Infocentro. La Hoyada, Caracas.Venezuela.

Artículo de:
Naudy Villarroel Urquiola
Twitter:**@naudyu**

Curso PHP Parte II

En esta sección estudiaremos las Variables.

Variables

Entendemos por variable al nombre que se le da aun valor para que sea asignado en memoria. En PHP las variables se representan como un signo de dólar (*$*) seguido por el nombre de la variable. El nombre de la variable es sensible a minúsculas y mayúsculas.

Un nombre de variable valido tiene que empezar con una letra o una raya (*underscore*), seguido de cualquier número de letras, números y caracteres alfanuméricos.

En PHP no es necesario establecer el tipo de dato, debido PHP es lo que se denomina como un lenguaje de tipos permisivos (*no requieren una declaración de tipo de variable y convierte automáticamente el tipo de variable en función del contexto en el que se utilice y las operaciones que se realicen en sus valores*), para crear la variable solo es necesario incluirlas en una expresión y establecer su valor.

Los tipos de datos sencillos son los que contienen un rango de valores que se puede ordenar en una dimensión (*cadenas, números, valores boleanos, etc*); mientras que los tipos de datos estructurados incluyen matrices y objetos.

Ejemplo:

```
<?php

 $nombre = "VaSlibre";
 $Nombre = "Valencia Software Libre";
 echo "$nombre, $Nombre";

?>
```

La salida sería algo como esto:

VaSlibre, Valencia Software Libre

Como pueden observar PHP toma las 2 variables como diferente, aunque para nosotros podrían ser iguales, esto es debido a que PHP hace distinción entre mayúsculas y minúsculas.

Las variables siempre se asignan por valor. Esto significa que cuando se asigna una expresión a una variable, el valor íntegro de la expresión original se copia en la variable de destino.

Podemos usar comillas simples (') o comillas dobles (") para mostrar valores, pero hay diferencia entre una y la otra, ejemplo:

Si observamos el anterior ejemplo estamos imprimiendo

```
echo "$nombre, $Nombre";
```

lo que en efecto nos enviará por la pantalla VaSlibre, Valencia Software Libre, pero si usamos:

```
echo '$nombre, $Nombre';
```

Nos enviará por la pantalla $nombre, $Nombre es decir, que los valores de la variables son omitidos y muestra en forma literal el contenido que se envía a impresión.

PHP proporciona una gran cantidad de variables predefinidas. De todas formas, muchas de esas variables no pueden estar completamente documentadas ya que dependen de sobre qué servidor se esté ejecutando, la versión y configuración de dicho servidor, y muchos otros factores.

Las variables predefinidas del sistema NO pueden usarse como nombres de variables.

Algunas de las variables globales comunes son:

$_GET, $_POST, $_FILE, $_SESSION, $_REQUEST, $_COOKIE, $_SERVER.

El ámbito de una variable es el contexto dentro del que la variable está definida. La mayor parte de las variables PHP sólo tienen un ámbito sencillo.

Ejemplos de Variables y contenido

$variable = 5;
$variable = '5';
$variable = NULL;
*$variable = 25 * 3;*
$variable = $variable2
$variable ="este texto \"lleva comillas dobles \"";
$variable ='este texto "lleva comillas dobles "';

Alcance de las Variables

El alcance de una variable es el contexto dentro del cual la variable está definida y donde la variable puede ser usada, por defecto las variables son locales, es decir son vistas solo donde se están usando.

Tipos de Variables

PHP soporta ocho tipos primitivos.

tipos escalares:
* boolean (*verdadero/falso, sí/no, 1/0*) se trata como TRUE o FALSE (*1 o 0*)
* integer (número entero)
* float (*número en coma-flotante, también conocido como 'double'*)
* string (*cadena de texto*)

tipos compuestos:
* array (*matriz de valores*)
* object (*objetos*)

tipos especiales:
* resource (*variable especial, que contiene una referencia a un recurso externo*)
* NULL (*variable no tiene valor*).

NOTA:
Escalar: significa que los valores que contiene se pueden ordenar en función a una escala.
Compuesto: Significa que los datos contienen varios elementos.
Especial: Significa que un número o valor especial tiene un significado concreto para la aplicación.

Asignación de Valores por referencia:

Es cuando la variable a la que se le asigna la referencia se convierte en un a "*alias*" o "apunta" a la variable asignada, los cambios a la nueva variable afectan tanto a la original como a la referencia. Esto no conlleva una copia de valores. Para hacer una variable de referencia se usa el símbolo &

Ejemplo

```
<?php

 $org = 'Valor Variable original';
 $ref = &$org;
 $ref = 'Nuevo valor asignado a variable referencia';
 echo "Variable \$org: ", $org;
 echo "Variable \$ref: ", $ref;

?>
```

Convertir tipos de datos

Función **gettype**: Obtener el tipo de una variable
Formato: *gettype (variable)*

Los posibles valores retornados serían: boolean integer double string array object resource NULL

Función **settype**: Define el tipo de una variable
Formato: *settype (variable, tipo)*

Los posibles valores de tipo son: boolean integer float string array object null

Devuelven TRUE si todo se llevó a cabo correctamente, FALSE en caso de fallo.

Ejemplo

```
<?php

 $variable = 2007;
 echo 'La variable $variable es de tipo: '.gettype($variable);

 $variable = (string)$variable;
 echo '<br /> La variable $variable ahora es de tipo: '. gettype($variable);

 $variable = settype($variable,"boolean");
 echo '<br /> La variable $variable ahora es de tipo: '. gettype($variable);

?>
```

Seguiremos en la próxima edición.

Artículo por AWVEN
http://www.awven.com
Twitter: @awvene
Identi.ca: awven

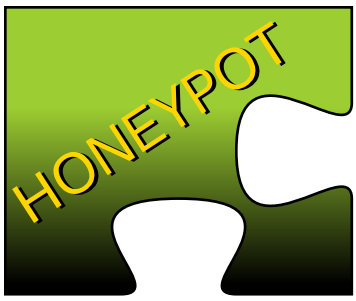
HoneyPot - a la caza de los Piratas Informáticos

En el año 2005 escribí un articulo con el nombre de "*Honeypots - Servidores Trampas*" en mi sitio web, el cual tuvo bastante receptividad, de hecho posee una llamada por enlace externo desde la wikipedia en español y ern algunas universidades fue usado como fuente de consulta. Desde esa fecha para aca ha corrido bastante agua y han habido mejoras importantes en las herramientas usadas para la aplicación de este tipo de sistema. Este nuevo artículo está dirigido hacia las nuevas generaciones de administradores.

El papel de la tecnología del sistema de detección de intrusos basado en señuelos o "*honeypots*" han evolucionado, demostrando su enorme potencial para la comunidad informática. En un principio fueron utilizados principalmente por los investigadores aunque actualmente esta siendo usado por empresas, entes gubernamentales y particulares, con el fin de atraer a los piratas informáticos a un sistema de redes para estudiar y evaluar sus movimientos y comportamiento. En efecto, al brindar detección temprana de actividades no autorizadas en las redes, los honeypots son ahora más útiles que nunca para los profesionales de seguridad de TI, proporcionando data para securizar los sistemas reales.

Los primeros conceptos fueron creados por varios iconos en la seguridad informática, especialmente por Cliff Stoll en el libro "*The Cuckoo's Egg*" y el trabajo de Bill Cheswick "*An Evening with Berferd*".

Dan Adams expresó un concepto bastante acertado: "*Consisten en activar un servidor y llenarlo de archivos tentadores, hacer que sea difícil, pero no imposible penetrarlo y sentarse a esperar que aparezcan los intrusos*"



Los honeynets (*conjuntos de honeypots*) dan a los crackers un gran espacio para recorrer. Presentan obstáculos que poseen el nivel de complejidad suficiente para atraerlos, pero sin irse al extremo para no desalentarlos... Ellos juegan con los archivos y conversan animadamente entre ellos sobre todo los fascinantes programas que encuentran, mientras el personal de seguridad observa con deleite cada movimiento que hacen. Francamente, siento una combinación de sentimientos con respecto a espiar a la gente, aunque no sean buenas personas".

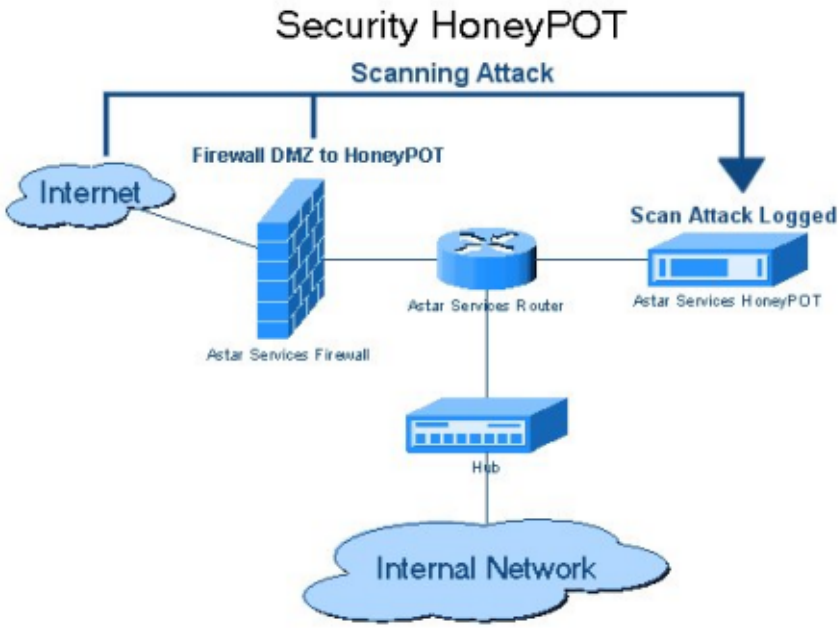
Algunos administradores TI usan los Honeypots para distraer a los atacantes de las máquinas realmente importantes del sistema, y advertir rápidamente un posible ataque al sistema, además de permitir un examen en profundidad del atacante, durante y después del ataque al honeypot. También algunos administradores de sistemas han creado honeypots que imitan un servidores de correo abierto y un proxie abierto, lo cual es empelado para identificar a spammers.

Honeypots son seguidos de cerca las trampas de la red que sirve a varios propósitos: que pueden distraer a los adversarios de máquinas más valiosas en una red, pueden proporcionar una alerta temprana sobre el ataque de nuevo y las tendencias de la explotación y permiten un examen en profundidad de los adversarios durante y después de la explotación de un honeypot.

Un servidor Honeypot es a veces un simple computador en el cual se levantan varios "*servicios*" virtuales para registrar

o recavar la actividad de posibles intrusos.

Algo que se debe tener en cuenta que el honeypot es una "*trampa*" y hay que mantenerlo alejado de la red real, ya que un verdadero "*hacker*" experimentado podría usarlo como puente de ingreso a la red real... haciendo estragos dentro de ella. Todos los

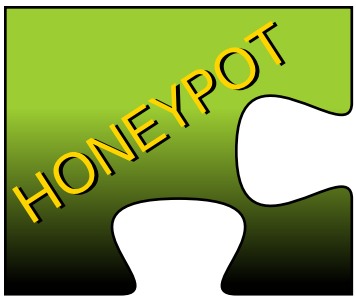


honeypots comparten el mismo concepto, y **JAMÁS** deben usarse como servidores de producción, pues podría afectar los servicios críticos de la red y comprometer la data.

En teoría, cualquier honeypot debería detectar cualquier actividad legítima o no entre su tráfico, claro dependiendo del o de los servicios levantados, por lo que se presupone que cualquier interacción es maliciosa, esto podría conllevar una ventaja o desventaja según el investigador y/o administrador que haya montado el servidor.

Un honeypots es una herramienta invaluable a la seguridad, debido a su flexibilidad en uso y configuración, permitiendo en una misma máquina establecer:

- prevención
- detección
- recopilación de información.



En general, existen dos tipos de honeypots, para la producción y para la investigación.

a) **Honeypots para la investigación**:
Gran parte de la atención actual se centra en los honeypots para la investigación, que se utilizan para recolectar información sobre las acciones de los intrusos. El proyecto Honeynet, por ejemplo, es una organización para la investigación sobre seguridad voluntaria, sin ánimo de lucro que utiliza los honeypots para recolectar información sobre las amenazas del ciberespacio.

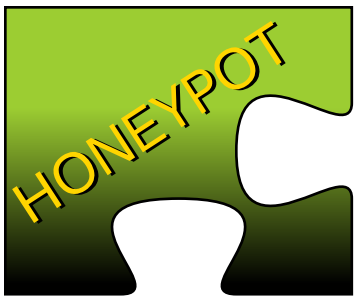
b) **Honeypots para la producción**:
Se les ha prestado menor atención a los honeypots para la producción, que son los que se utilizan para proteger a las organizaciones. Sin embargo, se les concede cada vez más importancia debido a las herramientas de detección que pueden brindar y por la forma cómo pueden complementar la protección en la red y en el host.

También existe otra forma de atraer a los atacantes, es lo que se conoce como "*Honeynet*", que es un conjunto de Honeypots, así se abarca un mayor rango de información para un estudio más especifico. Incluso este tipo de trampa, hace más fascinante y jugoso el ataque del intruso, lo cual incrementa el número de ataques y aumenta el esfuerzo del posible pirata informático.

La función principal a parte de la de estudiar las herramientas de ataque, es la de desviar la atención del atacante de la red real del sistema y la de capturar nuevos virus o gusanos para su posterior estudio. Una de las múltiples aplicaciones que tiene es la de poder formar perfiles de atacantes y ataques.

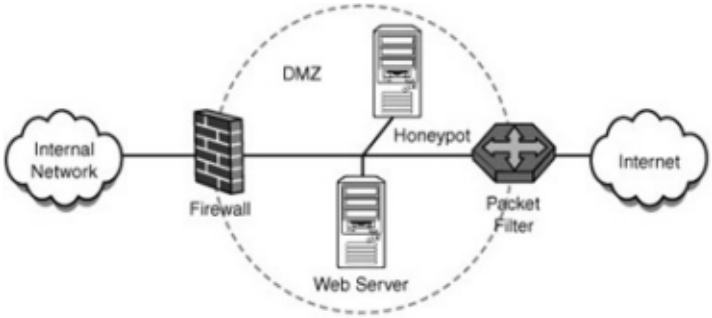
Los honeypots también se pueden describir como de alta o baja interacción, distinción que se basa en el nivel de actividad que le permiten al atacante. Un sistema de baja interacción ofrece actividad limitada; la mayoría de las veces funciona al emular (en algunos casos virtualizar) los servicios y sistemas operativos. Las principales ventajas de los honeypots de baja interacción es que son relativamente fáciles de instalar y mantener; también implican un riesgo mínimo porque el atacante nunca tiene acceso a un sistema operativo real para perjudicar a otros sistemas.

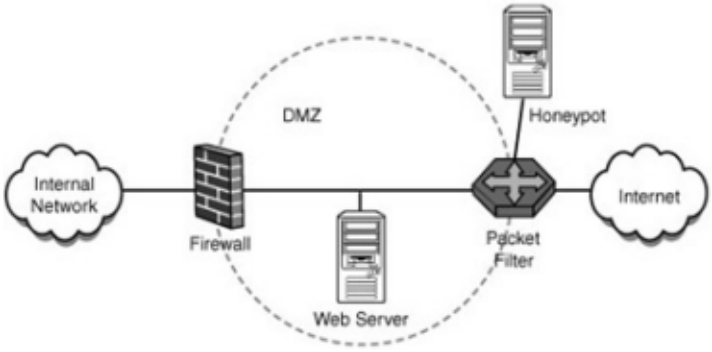
"*Honeyd*" es un ejemplo de honeypot de baja interacción, cuya función principal es monitorizar el espacio de direcciones IP no utilizado. Cuando un Honeyd detecta un intento de conectarse a un sistema que no existe, intercepta la conexión, interactúa con el atacante fingiendo ser la víctima para captar y registrar al ataque.



Por el contrario, los honeypots de alta interacción utilizan sistemas operativos reales y aplicaciones reales y no emulan nada. Al ofrecerles a los atacantes sistemas reales para que interactúen, las organizaciones pueden aprender mucho sobre su comportamiento. Los honeypots de alta interacción no imaginan como se comportará un atacante y proporcionan un ambiente que rastrea todas las actividades, lo que les permite a las organizaciones conocer un comportamiento al que de otra manera no tendrían acceso.

Los sistemas de alta interacción también son flexibles y los profesionales de la seguridad de TI pueden implementarlos en la medida que quieran. Además, este tipo de honeypot proporciona un objetivo más realista, capaz de detectar atacantes de mayor calibre. Los honeypots de alta interacción pueden ser complejos de instalar. Sin embargo, requieren que se implementen tecnologías adicionales para evitar que los atacantes los utilicen para lanzar ataques a otros sistemas.





Siendo más directos en el enfoque anteriormente expresado:

a) **Baja Interacción** -> Solución que emula sistema operativos y servicios.

b) **Alta Interacción** -> No hay emulación, son suministrados sistemas operativos y servicios reales.

Sistemas de Detección de Intrusos.

Los conocidos como sistemas de detección de intrusos se han convertido

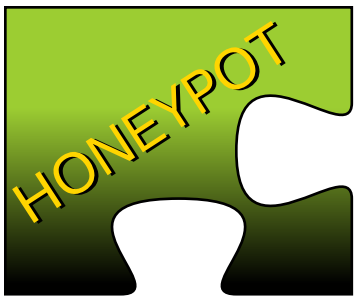
en un componente importante en la caja de herramientas de un buen administrador de seguridad, ya que es posible detectar actividades inapropiadas, incorrectas o anómalas. Algunos aún no lo tienen claro, y piensan que un IDS (*Intrusión Detection System*) puede ser la panacea que nos lleve a la más absoluta tranquilidad y a una irreal sensación de seguridad, más peligrosa en muchos casos que la inseguridad en sí misma. Un detector de intrusos no es más que una de las medidas de seguridad que necesariamente hay que tomar para proteger una red. Una herramienta bastante bueno de detección de intrusos y avisos de alertas es ConfigServer Security & Firewall puedes observarlo en: http://www.configserver.com

En resumen, un sistema de detección de intrusos hace exactamente eso. Detectar posibles intrusiones. Específicamente, pretende detectar ataques o abusos al sistema, alertando con los pormenores del ataque. Proporciona una seguridad parecida a la que un sistema de alarma instalado en casa puede suponer. Mediante varios métodos, ambos detectan si un intruso, atacante o ladrón está presente, y en consecuencia disparan una alarma. Por supuesto, la alarma puede saltar sin motivo, al igual que el administrador puede no llegar a oírla. Ante estar irregularidades, solo resta estar atento y revisar con regularidad y con cautela los avisos recibidos.

Tradicionalmente, hay dos tipos generales de sistemas de detección de intrusos:

a) **Anfitrión sistemas de detección de intrusiones (HIDS)**: sistemas de IDS que operan en una máquina para detectar actividad maliciosa en esa máquina.

b) **Red de sistemas basados en detección de intrusos (NIDS)**: sistemas de IDS que operan en los flujos de datos de la red.



Aunque últimamente se está empleando el Sistema de Prevención de Intrusos o IPS. Este es un sistema que supervisa de forma activa una red o host de los ataques y previene los ataques que se produzcan de y desde una IP especifica.

Que ventajas nos trae implementar un Honeypot:

1- **Conjunto de datos pequeños pero de gran importancia:** Los Honeypots recolectan pequeñas cantidades de información. En lugar de loguear 1 Gb por día, loguean sólo 1 Mb de datos por día. En vez de generar 10.000 alertas por día, pueden generar sólo 10 alertas por día. Recuerden, los honeypots sólo capturan actividad sospechosa ya que cualquier interacción con un honeypot es muy probablemente actividad no autorizada o una actividad maliciosa, en otras palabras los honeypots reducen el "ruido" recogiendo sólo datos indispensables, de gran valor, los producidos únicamente por los píratas informáticos. Esto significa que es mucho más fácil y económico el analizar los datos que un honeypot recoge.

2.- **Nuevas herramientas y tácticas**: Los honeypots son diseñados para capturar cualquier cosa que interactúa con él, incluyendo herramientas o tácticas nunca vistas. Permitiendo expandir la evaluación de un posible ataque.

3.- **Mínimos recursos**: Los honeypots requieren mínimos recursos, sólo capturan actividad irregular. Esto significa que un viejo Pentium II con 128 mb de RAM puede manejar fácilmente una entera red clase B en una red OC-12.

4.- **Cifrado o IPv6**: A diferencia de la mayoría de las tecnologías para la seguridad, como los sistemas IDS, los honeypots trabajan bien en entornos cifrados ("*encriptados*") como IPv6. No importa lo que los piratas informáticos lancen hacia y en contra del honeypot, este los detectará y los capturará.

5.-**Información**: Los honeypots pueden recoger información "*en profundidad*" como pocos, si es que existen tecnologías que se le parezcan.

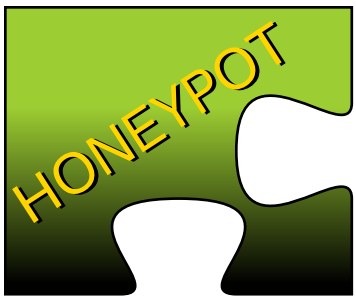
6.-**Simplicidad**: Finalmente, los honeypots son conceptualmente simples. No hay por qué desarrollar algoritmos raros, ni complejas tablas que mantener, o firmas que actualizar. Mientras más simple sea la tecnología, menos posibilidades de errores o desconfiguraciones habrá.

Cuales desventajas podemos encontrar:

Como en cualquier tecnología, los honeypots tienen su debilidad. Esto es debido a que no reemplaza a la actual tecnología, sino que trabaja con las que actualmente existen y podemos nombrar:

1.- **Visión Limitada**: Los honeypots pueden sólo rastrear y capturar actividad que interactúen directamente con ellos. Los Honeypots no podrán capturar ataques a otros sistemas vecinos o adjuntos, al menos que el atacante o la amenaza interactúe con el honeypot al mismo tiempo.

2.- **Riesgo**: Todas las tecnologías de seguridad tienen un riesgo. Los firewalls tienen el riesgo de que sean penetrados, el cifrado "*encriptación*" tiene el riesgo de que sean rotos *(hoy día casi todos han sido violentados*), sensores IDS tienen el riesgo de que fallen al detectar ataques. Los Honeypots no son diferentes, poseen su riesgo igualmente.



Específicamente, los honeypots tienen el riesgo de que sean apoderados y controlados por los piratas informáticos y lo utilicen para dañar otros sistemas dentro de la red. El riesgo es variado para los diferentes honeypots. Dependiendo en el tipo de honeypots puede haber un riesgo no mayor a la de una falla del sensor IDS, mientras que en otros honeypots puede que haya que enfrentarse a una situación crítica.

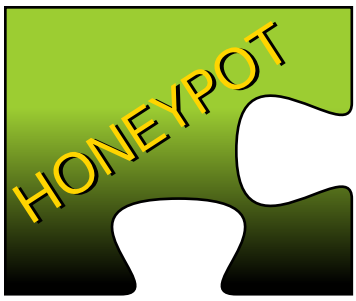
Ahora, ¿Cual es la posible utilidad de este tipo de tecnología?. Los honeypots pueden ayudar a prevenir ataques en varias formas. El primero es contra ataques automatizados (*botnet*), como gusanos (*worms*) o auto-rooters. Estos ataques son basados en herramientas que aleatoriamente escanean redes enteras buscando sistemas vulnerables. Si un sistema vulnerable es encontrado, estas herramientas automatizadas atacarán y tomarán el sistema (*con gusanos que se auto-copian a sí mismo en la víctima*). Uno de las métodos para protejer de tales ataques es bajando la velocidad de su escaneo y potencialmente detenerlos. Llamados "*sticky honeypots*" (*Tarros de miel "pegajosos"*), estas soluciones monitorizan el espacio IP no utilizado. Cuando los sistemas son escaneados, estos honeypots interactúan con él y disminuyen la velocidad del ataque. Hacen esto utilizando una variedad de trucos TCP, como poniendo el "Window size" a cero o poniendo al atacante en un estado de espera continua. Esto es excelente para bajar la velocidad o para prevenir la diseminación de gusanos que han penetrado en la red interna.

Un ejemplo de un sticky honeypot es el LaBrea Tarpit. Los "*Honeypots pegajosos*" son más comunes encontrarlos entre soluciones de baja interacción

(hasta podría llamársela soluciones "*no interactivas*", ya que reducen tanto la velocidad que hacen gatear al atacante. Los Honeypots pueden también proteger su organización de perpetradores humanos. Este concepto se conoce como engaño o disuasión. La idea es confundir al atacante, hacerle perder el tiempo y recursos interactuando con honeypots. Mientras tanto, su organización habría detectado la actividad del atacante y tendría tiempo para reaccionar y detener el ataque. Hasta se puede dar un paso más allá: si un atacante sabe que su organización está utilizando honeypots pero no sabe cuales son los sistemas honeypots y cuales son sistemas legítimos, quizás tenga miedo de ser capturado por honeypots y decida no atacarlo. Por lo tanto, honeypots disuaden al atacante. Un ejemplo de honeypot diseñado para hacer esto, es el **Deception Toolkit**, un honeypot de baja interacción.

Spam-IP.com es un sitio web dedicado a informar y luchar contra el Spam, esta gente han creado un Honeypot que se trata de un formulario cualquiera como los de blogs, páginas de contacto, etc. Cuando un bot envía información en dicho formulario, su IP es automaticamente incluida en la lista. Permitiendo luego que loas Administradores de red usen esa información para ser incluidas en la lista negra de Spam.

MalwareBlacklist.com, tal como su nombre lo indica, es una lista negra de malwares y URLs maliciosas. El servicio es ofrecido por la empresa de seguridad ParetoLogic y constantemente se está actualizando gracias a su honeypot y a las muestras que envían usuarios de todo el mundo.



unos de los honeypots más empleados son -**Jackpot-** escrito en Java (http://jackpot.uk.net/), y -**smtpot.py**- escrito en Python (http://llama.whoi.edu /smtpot.py). -**Proxypot**- es un honeypot que imita un proxy abierto (http://www.proxypot.org/). Kit de herramientas Honeyd para Linux" (**Honeyd Linux Toolkit** http://www.tracking-hackers.com/solutions/honeyd/ igualmente puedes usar un sencillo script hecho en perl (en realidad son 2) hecho por la gente de activeware.com el cual puedes descargar desde:

http://www.megamultimedia.com/arroba/arroba78/descargas/intrusos.zip

Te invito a que pruebes esta interesante herramienta, puedes aprender mucho de ella.

Enlaces Importantes:
http://www.honeypots.net/
http://his.sourceforge.net/trad/honeynet/

Artículo por
Héctor A. Mantellini (Xombra)
http://www.xombra.com
Twitter: @xombra
Identi.ca: xombra
GNU/Linux User: #414452

XOMBRA

Android el futuro del móvil.

Primero antes que todo, debemos hablar de **Andy Rubin**, la cabeza que detrás de bastidores creo la revolución que actualmente se conoce como uno de los mejores sistemas operativos móviles del mercado, aunque actualmente es el Vicepresidente de Ingeniería en Google y Director de Plataformas Móviles su historia se viene gestando desde hace más de 22 años, es decir el pasado de este hombre ha estado ligado a Android como meta existencial.

Cuando Andy egreso de la Universidad se mudo a Suiza y trabajo en Carl Zeiss (*si,la misma de las lentes ópticas de Nokia*) para luego viajar a las Islas Caimán donde Bill Caswell lo recluto para Apple Inc, con el pasar de los años específicamente en 1992 se integra a un proyecto llamado General Magic, un proyecto para desarrollar un SO para smartphones. Obviamente y como muchos inventos adelantados a su época el proyecto no calo y este se fue disolviendo.

Para terminar la pequeña síntesis de la bio de Andy, este termino uniéndose a Artemis (*adquirida por microsoft*) Research donde formo parte del proyecto WebTV, que intentaba unir la experiencia Internet-TV, luego fundo Danger (adquirida también por Microsoft) para finalmente fundar Android.inc que fuera adquirido por Google y nos lleva a la historia que hoy nos incumbe.

Android es un sistema operativo basado en Linux diseñado originalmente para dispositivos móviles, tales como teléfonos inteligentes, pero que posteriormente se expandió su desarrollo para soportar otros dispositivos tales como tablets, reproductores MP3, netbooks, PCs, televisores, lectores de e-books e incluso, se han llegado a ver en el CES, microondas y en lavadoras .

¿Porque es un sistema que podría acabar con la hegemonía de Apple y su Iphone?

Debo reconocer que el Iphone desde su lanzamiento ha sido un excelente producto en cuanto a innovación y desempeño, pero ¿cual es su principal problema?, El target al cual va dirigido y su plataforma cerrada, el cual hace que todo el control del sistema pertenezca a una sola compañía. Algunos dirán que esto es una ventaja ya que significa homogeneidad en el producto, pero conlleva a que las mejoras del sistema operativo en cuanto a hardware y software sean responsabilidad total de Apple Inc, no asi con Android pasa lo siguiente, cualquier fabricante puede hacer su propia versión de Android y compilarla para cualquier producto hardware elaborado por ellos, no importando si el dispositivo sea de gama baja, alta o media.

Perfecto ¿Pero?, he oído que el control de versiones de Android es un caos.

Si uno de los problemas del cual se les acusa a Android es la fragmentación, y esto conlleva a que los fabricantes prefieran no disponer de las ultimas versiones en sus dispositivos.

Con respecto a esto Google hizo un acuerdo con los principales fabricantes y operadoras del mundo a que los futuros lanzamientos de sus dispositivos dispondrán de las ultimas actualizaciones apenas salgan al mercado para que esto conlleve a una hegemonía en el uso del sistema operativo, esto sera una realidad con el próximo lanzamiento de Ice Cream Sándwich que podrá ejecutarse en cualquier teléfono, tablet, pc o televisión.

De la compra de Motorola por parte de Google.

La compra de Motorola Mobility permite a Google salvaguardar Android, lo mantendrán como sistema libre, debido al litigio de patentes con Apple y Microsoft que ponía en grave peligro al proyecto y Google adquiere con su compra mas de 17.000 patentes otorgadas y 7.500 pendientes de aprobación, 18 de ellas cubren tecnología esencial para el desarrollo de smartphones y ya fueron utilizadas en un litigio pasado con el fabricante de Cupertino.

Google registró durante este año alrededor de 1.000 patentes relacionadas con su Android.

Son 80 años los que tiene Motorola, y esta compra los pone en igualdad de condiciones con Apple y Microsoft en un sector, el de las patentes, donde la legislación americana se había desviado de su inicial objetivo, proteger al pequeño frente a los grandes y Google en este aspecto, era el pequeño. Solo por eso el precio pagado ya valdría la pena para Google, el crecimiento de Android en los últimos años lo había puesto en el punto de mira de Apple y Microsoft, Google ahora esta en igualdad en esa lucha, la propia Motorola tiene demandas presentadas contra Apple por el tema de las patentes.

Pero no solo sale ganando en el tema de las patentes, Google también adquiere el conocimiento y experiencia en la fabricación de hardware, de nuevo aparecen sus principales competidores Apple y Microsoft. Ahora puede desarrollar sus propios productos para acompañarlos de su software.

Con esa división Google intentara solucionar un tercer problema, la integración de Internet en los aparatos domésticos como la televisión, es el futuro y Google TV no llevaba el camino adecuado, con esta división eso puede corregirse. Google poseerá hardware capaz de servirnos Internet a la carta, potenciar su TV e incluso darle mas uso a YouTube que el que ahora tiene.

Artículo por
Eduardo Echeverria (echevemaster)
Twitter: @echevemaster

Curso Phyton Parte II

Programar con Python es algo satisfactorio. Un lenguaje simple, poderoso, y muy ordenado. Veamos algunas cosas:

Indentación:

Indentar el código es la práctica de "acomodar" el texto de manera que se entienda que código pertenece a cada bloque. Es decir agregar las tabulaciones necesarias al inicio de cada línea para que el texto quede más "adentro" del bloque. En Python indentar el código es estrictamente necesario, pues es la única forma que tiene el interprete de saber que instrucciones pertenecen a cada bloque; esto es porque en Python no hay delimitadores de bloque. Los delimitadores de bloques, son esos símbolos que nos permiten "encerrar" el código por bloques. En C++ y Java los delimitadores de bloque son las llaves. Así un condicional en C++ y en Java tiene la siguiente forma:

if (condicion){sentenciasDentroCondicional;}
SentenciasFueraCondicional;

Varios puntos que deben tomarse en cuenta:

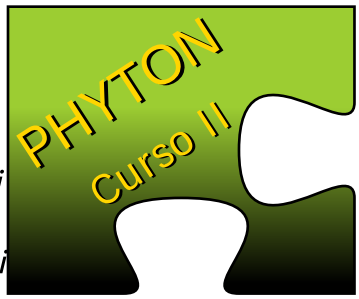
C/C++ y Java obligan a utilizar "*paréntesis*" en el predicado de la estructura condicional.

C/C++ y Java obligan a utilizar "*corchetes*" para marcar el inicio y el fin del bloque.

C/C++ y Java obligan a utilizar "*punto y coma*" para indicar el final de la instrucción.

En Python se utilizan los "*dos puntos*" sólo para marcar el inicio del bloque, ya que el bloque _debe_ estar indentado.

Mientras que en Python tiene la siguiente forma:



if condicion:

sentenciasDentroCondicional
sentenciasFueraCondicional

Cómo pueden ver, lo que diferencia un código de otro es el indentación, el margen o la sangría. Ahora ¿por qué es tan importante saber esto? Pues simple, porque por costumbre los desarrolladores utilizamos la tabulación para indentar el código y los editores de texto avanzado, que están preparados para codificar, e incluso los que no son tan avanzados, tienen distintas formas de hacer esa tabulación en el texto (*por ejemplo, gedit por defecto coloca lo que en una cadena de texto es un "\t" o una tabulación ASCII, algunos coloca 3 ó 4 espacios en blanco*), y esto suele traer problemas a la hora de codificar en distintas computadores, distintos editores y/o en la edición colaborativa de código. Así que asegurate de establecer bien claro en las opciones de tu IDE, o editor de textos la forma en la que se deben comportar las tabulaciones; yo recomiendo utilizar 4 espacios en blanco, pero es decisión de cada quién.

Una última cosa antes de empezar con las estructuras condicionales y bucles: el separador de sentencias es la nueva línea, es decir que una sentencia se separa de otra sólo si está en otra línea. Por ejemplo:

sentencia1
sentencia2
sentencia3

En Java y C/C++ el delimitador es el punto y coma (";"), pudiendo tener (*aunque no sea lo que se estila normalmente):*
sentencia1;sentencia2;sentencia3;

Las Estructuras condicionales

Ya les dí un abreboca de lo que son los condicionales en Python. La estructura es la siguiente:

if condicion:

En esta estructura <*condición*> es cualquier expresión booleana, es decir, una expresión que devuelva verdadero o falso, o incluso una variable de tipo booleana. En Python la tabla de comparadores es muy similar a la de otros lenguajes e incluso los operadores booleanos:

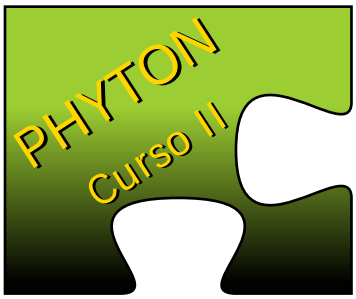
Operadores relacionales
== igual
!= Distinto
< Menor
> Mayor
<= Menor o Igual
>= Mayor o Igual

Operadores Booleanos
AND "Y" lógico
OR "O" lógico
NOT "No" lógico

Después de la condición se colocan dos puntos ("*:*"), para dar comienzo en la siguiente línea a las sentencias que se ejecutaran si la condición es verdadera ("*True*"). Luego del "*if*" tenemos el "*else*", para lo cual se le coloca la palabra reservada "else" seguida de los "*dos puntos*" y las sentencias que se ejecutarán en caso de que la condición no se cumpla deben ir indentadas. La estructura queda así:

```
if condicion:
   sentenciasQueEjecutanSiCondicionVerdadera

else:
   sentenciasQueEjecutanSiCondicionFalsa
```



Y para terminar con las estructuras condicionales, en python contamos con el "*elif*" para hacer estructuras condicionales anidadas, es decir, cuando necesitamos condiciones dentro de condiciones.

```
if condicion1:
   sentenciasQueEjecutanSiCondicion1Verdadera

elif condicion2:
   sentenciasQueEjecutanSiCondicion2Verdadera

elif condicion3:
   sentenciasQueEjecutanSiCondicion3Verdadera

else:
   sentenciasQueEjecutanSiNingunaCondicionVerdadera
```

En C/C++ y en Java esas estructuras no existen, sino que para hacer estructuras condicionales con varias condiciones se anidan las instrucciones "*if*" unas dentro de otras:

```
if (condicion1){
  sentenciasQueEjecutanSiCondicion1Verdadera;
}else{
  if (condicion2){
  sentenciasQueEjecutanSiCondicion2Verdadera;
  }else{
    if (condicion3){
  sentenciasQueEjecutanSiCondicion3Verdadera;
    }else{
sentenciasQueEjecutanSiNingunaCondicionEsVerdadera;
    }
  }
}
```

Recordando siempre que en estos lenguajes el indentado no es obligatorio, ni el cambiar de línea; y que las llaves no tienen que estar alineadas, siempre que estén completas. Pudiendo quedar el código de la siguiente manera:

```
if
(condicion1){sentenciasQueSeEjecutanSiCondicion1E
sVerdadera;}
else{if
(condicion2){sentenciasQueSeEjecutanSiCondicion2E
sVerdadera;}
else{if
(condicion3){sentenciasQueSeEjecutanSiCondicion3E
sVerdadera;}
else{sentenciasQueSeEjecutanSiNingunaCondicionEs
Verdadera;}}}
```

Así el código es menos legible.

Los Bucles ó Ciclos:

Los ciclos son el último tópico que tocaré en este artículo. Dividimos esta sección en dos partes: "*while*" y "for". El "*repeat*" no existe en python, por eso no lo trataremos. Realmente un ciclo"*repeat*" se puede escribir como un ciclo "while", asegurando que la condición se cumpla por lo menos la primera vez.

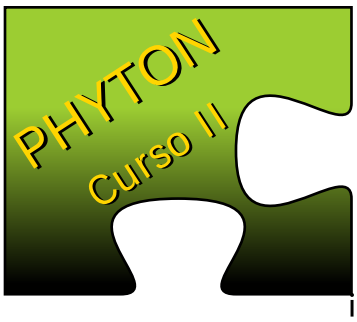
Ciclo "while"

El ciclo "*while*" tiene una construcción muy sencilla, consta de la palabra reservada while seguida de la condición y luego dos puntos ("*:*"). El código queda así:

```
while condicion:

sentenciasQueSeEjecutanMientrasLaCondicionEsVerd
adera
sentenciasQueSeEjecutanCuandoLaCondicionDejeDe
Cumplirse
```

Cabe destacar que siempre es necesario que la condición cambie en algún punto del programa porque sino no dejará de ejecutarse la parte nterna de un ciclo.



También hay que tomar en cuenta los mismos tópicos de comparación que con el "*if*" cuando comparas el "*while*" de python con el "while" de C/C++ o Java, es decir, en Python no se requiere que la condición esté entre paréntesis, es necesario que los dos puntos ("*:*") sigan a la condición y que las sentencias que se ejecuten dentro del ciclo estén indentadas.

Ciclo "for"

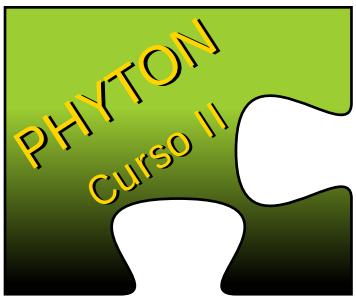
El ciclo "*for*" es algo un poco más difícil de entender al principio para los que venimos de lenguajes "*estilo-C*" como C++, Java o PHP. En Python el ciclo "*for*" es un iterador sobre listas (*las listas las veremos en otro artículo*), pero para hacer la comparación contra C/C++ y Java, haremos uso de la función "*range()*" para recorrer una secuencia de números (*sólo mostraré un uso simple, la función tiene más usos que los que veremos acá*).

Un ciclo "*for*" en Python se hace así:

```
for variable in range(10):

sentenciasQueSeEjecutaranDuranteLasVueltasDelCic
lo
```

Cómo podemos ver el ciclo es algo más natural. Podríamos leer el ciclo de la siguiente forma: "*Para el valor de la variable dentro del rango*". La función "*range()*" genera una secuencia de números desde 0 (cero) hasta el valor que se coloca como parámetro.

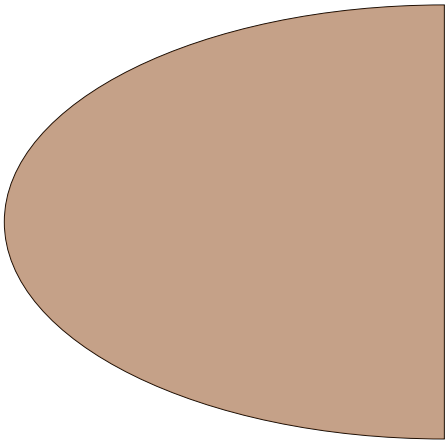


En C/C++ y Java, el "*for*" tiene una construcción un poco diferente:

for (variable=valorInicial; condicionDeVariable; cambioDeVariable)
{sentenciasDentroDelCiclo;}

En esta construcción se debe recordar que <*condicionDeVariable*> es una condición donde normalmente se indica que la variable es menor o mayor a algo, y <*cambioDeVariable*> es un cambio que se ejecuta en la variable del ciclo, puede ser un incremento o un decremento, depende de lo que se quiere hacer. Esa diferencia es importante, en Python nos movemos en un rango incremental, para ir en una secuencia decremental hay que proporcionarle más parámetros a la función range(), pero eso lo veremos en otro artículo.

En conclusión Python es un lenguaje bastante simple, y con una forma de codificar muy limpia y ordenada que es obligada por el lenguaje.



Espero que puedan engancharse al igual que me he enganchado yo.

Artículo de:
Eduard "X3MBoy" Lucena
Be Free, Be Linux…

Safe Creative es la primera plataforma de registro, información y gestión de Propiedad Intelectual para la realidad digital.

Información

Información en tiempo real de la propiedad intelectual, tanto para personas, a través de interfaz web y etiquetas informativas, como para programas y sistemas, a través de la tecnología semántica desarrollada por Safe Creative: Semantic Copyright.

¿Quién es el autor y cómo contactar con él?
¿Qué uso puedo hacer de esta obra?
¿Qué obras hay de este tipo con esta licencia de uso?

Seguridad

Para los autores: la prueba de autoría de un registro con garantías tecnológicas irrefutables (depósito de la obra, registro de múltiples huellas digitales y doble sellado de tiempo).

Para los usuarios de las obras: certeza sobre la licencia y usos permitidos y seguridad frente a posibles futuros cambios de licencia por parte del autor.
Autogestión

La tecnología e Internet hacen posible la auto-producción, la auto-distribución… Safe Creative proporciona la primera plataforma que permite la autogestión independiente y directa para las nuevas realidades creativas de creadores que necesitan modelos diferentes de gestión de derechos:

. Música para videojuegos.
. Modelado virtual 3D para software.
. Fotografía freelance.
. Auto-publicación de libros y música.
. Etc.

Garantía de fecha de registro

La fecha de los registros queda registrada con tecnología de sellado de tiempo, que se aplica de forma redundante para ofrecer también una garantía absoluta en la fecha del registro

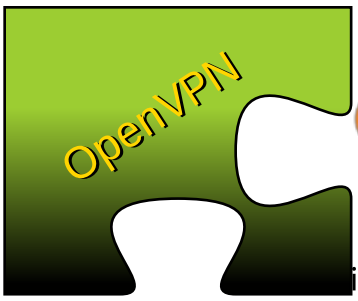
Plataforma de licenciamiento

Los autores pueden gestionar de forma directa e independiente la concesión y cobro de licencias de uso y distribución de sus trabajos. Colocar Un sistema de licenciamiento y pago "*Incrustable*" en cualquier página web.

Servicios y coste

Los servicios básicos de registro, información y certificación son gratuitos.

Los servicios "*Premium*" y de asistencia letrada por el equipo de abogados especialistas en propiedad intelectual de Safe Creative se pueden contratar por periodos anuales y con tarifas muy reducidas. Que espera, registra tu obra, protege tu trabajo. Ingresa a:
http://www.safecreative.org



OPENVPN

OpenVPN es una implementación de VPN SSL la cual usa las extenciones OSI layer 2 o 3 para asegurar redes la cual usa los protocolos SSL/TSL, soporta diferentes medios de autenticacion como certificados, smartcards, y/o usuario y contraseñas, y permite politicas de control de acceso para usuarios o grupos cuando usamos reglas de firewall *(iptables en mi caso*) aplicadas a las interfaces virtuales de la VPN.

OpenVPN 2.0 permite contectar multiples clientes a un solo servidor (*proceso*) OpenVPN sobre un simple puerto TCP o UDP.

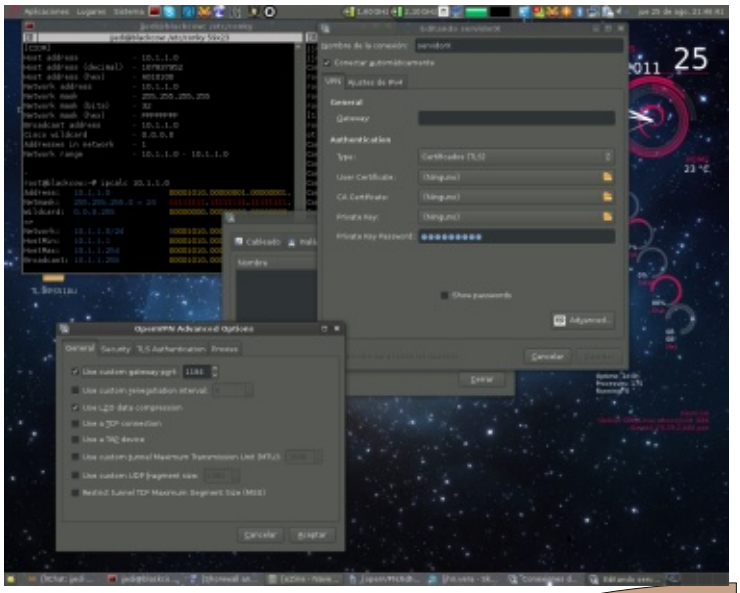
Porqué Surgen las VPNs?

- Intercambio flexible, rapido y seguridad de informacion.
- Sucursales en distintas ubicaciones.
- Administradores remotos.
- Necesidad de altos estandares de seguridad: autenticidad, integridad y disponibilidad.

Arquitecturas básicas:

- Acceso remoto (*roadwarriors*): Usuario que se contectan de manera remota (*domicilios, hoteles . .*)
- Utilizando internet como via de acceso.
- Punto a Punto: Conexión entre diversos puntos de una organización a través de internet.
- Interna VLAN: utiliza la LAN de la organizacion como via de acceso. Sirve para aislar zonas y servicios de la red interna.

Buscando en la web podemos hallar abundante información, alguno de ellos muy interesantes, en este artículo dejaré una serie de pasos esenciales para instalar y configurar OpenVPN en sistema operativo GNU/Linux Debia utilizando PKI (*infraestructura de claves*



públicas) en modalidad Roadwarrior.

lLo primero es instalar **OpenVPN**

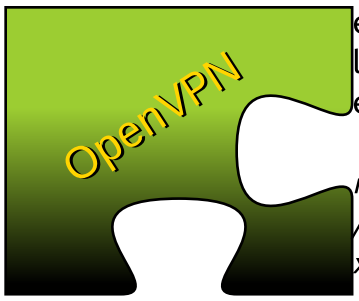
apt-get install openvpn openssl

aptitude install openvpn openssl

Cuando instalamos el metapaquete con aptitude este resuelve las siguientes dependencias y las necesita para realizar ciertas acciones criptográficas como:

* Creación de parámetros de claves RSA, DH y DSA.
* Creación de certificados X.509, CSR y CRL.
* Cálculo de resúmenes de mensaje.
* Cifrado y descifrado con cifradores.
* Pruebas de cliente y servidor SSL/TLS.
* Manejo de correos electrónicos firmados con S/MIME o cifrados.

Una vez instalado se debe generar las claves del servidor y de los clientes, pero antes se deben organizar algunos scripts de generación de estas mismas claves,



en Debian vamos a localizar los scripts en este directorio.

root@blackcow:~# /usr/share/doc/openvpn/examples/easy-rsa/

se puede copiar estas llaves a otra ubicacion (*opcional*)

root@blackcow:~# cp -r /usr/share/doc/openvpn/examples/easy-rsa/ /etc/openvpn/

trabajando en el directorio donde se copiaron los scripts

root@blackcow:~#cd /etc/openvpn/easy-rsa/2.0/

ahora, vamos a editar el archivo var, esto para personalizar la futura generacion de certificados y de identificar nuestra VPN. En si solo lo que vamos a editar son algunos datos de estos scripts.

-rw-r--r-- 1 root root 1678 ago 22 23:07 vars
root@blackcow:/etc/openvpn/easy-rsa/2.0# vim vars

 # These are the default values for fields
 # which will be placed in the certificate.
 # Don't leave any of these fields blank.
export KEY_COUNTRY="VE"
export KEY_PROVINCE="CCS"
export KEY_CITY="Altamira"
export KEY_ORG="MiOrganizacion"
export KEY_EMAIL="jjedi@miorganizacion.me"

Inicializar Variables de Ambiente de Trabajo

../vars
./clean-all
./built_ca

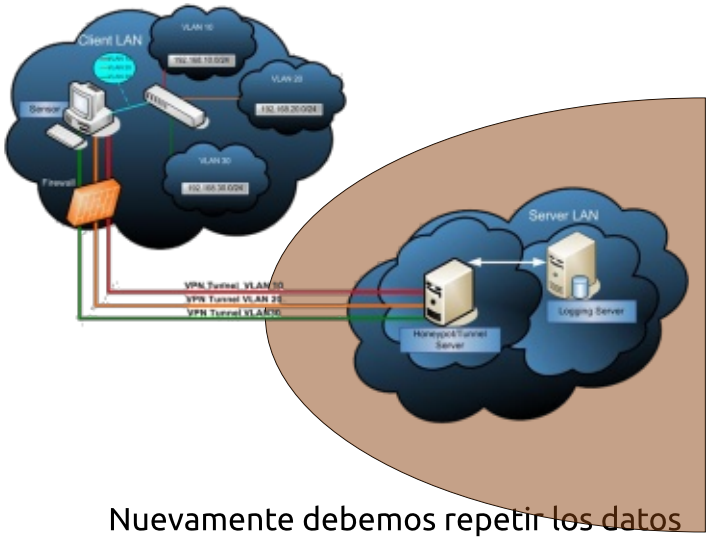
Con esto inicializamos el directorio de las claves (*borrando potenciales archivos viejos*)

Built_ca procede a generar el certificado CA

Debemos ingresar los datos correctos propios o de la Organización reales y teniendo especial atención con el parámetro Common Name el cual debera ser distinto para el caso de la CA, Servidor y los clientes.

Generamos las credenciales para el Servidor

root@blackcow:~#./build-key-server servidor



Nuevamente debemos repetir los datos ingresados anteriormente, recordando utilizar utilizando una denominación diferente de la que usamos para la CA

Generamos las credenciales de los clientes

root@blackcow:~#./build-key cliente1
Esto genera los archivos cliente1.key (la llave) y cliente1.crt (certificado).
Copiamos las credenciales del Servidor en /etc/openvpn
root@blackcow:~#cp ca.key ca.crt servidor.key

"*esto depende del Common Name que damos cuando creamos las credenciales para el servidor*"

Pasamos credenciales a los clientes

Esto archivos son *cliente1.crt cliente1.key* y *ca.crt* lo podemos pasar via sftp, ssh o scp la idea es usar una herramienta que sea segura.

Los Archivos *cliente1.crt* y *cliente1.key* dependen del Common Name que ingresamos cuando creamos las credenciales de los clientes.

Creamos la configuracion del Servidor OpenVPN

root@blackcow:~#vim /etc/openvpn/server.conf

port 1194
proto udp
dev tun
persist-tun
ca ca.crt
cert servidor.crt
key servidor.key
dh dh1024.pem
#Direcciones IP que se le asignaran a los clientes
#El server es el .1
server 10.1.1.0 255.255.255.0
ifconfig-pool-persist ipp.txt
#Ruta para que los clientes alcancen la red local del server
push "route 192.168.1.0 255.255.255.0"
#Para que los clientes se visualicen entre ellos
#esto debe ir junto con la opcion routeback si usas un shorewall
keepalive 10 120
comp-lzo
user nobody
group nogroup
persist-key
persist-tun
status openvpn-status.log
verb 4

Nota: *la red local de la VPN esta bajo la direccion 192.168.1.0/24 y la red del servidor VPN es 10.1.1.0/24 Si el usuario remoto o administrador esta detras de un NAT debe tener una direccion local distinta no perteneciente a la red local de la Red VPN.*

Configuramos el cliente OpenVPN

dentro del directorio */openvpn/config* creamos un archivo llamado *cliente1.ovpn*

tls-client
client
dev tun
proto udp
remote host.midominio.me
float
#ya que la ip de arriba es dinamica
resolv-retry infinite
nobind
persist-key
persist-tun
ca "rutadelarchivo ca.crt"
cert "rutadelArchivo cliente1.crt"
key "Archivo cliente1.key"
comp-lzo
verb 4

Nota: se puede copiar cada uno de estos archivos en una misma carpeta [*ca.crt cliente1.crt* y *cliente1.key*]

Arrancamos el demonio OpenVPN en el Servidor

root@blackcow:~#/etc/init.d/openvpn start

y luego creamos la conexion con el *GUI de gnome* para *openvpn* o con *gadmin-openvpn-client*

John M. A. Vera F.
VaSLibre Valencia-Venezuela
Linux Counter# 467192