Raza MEXICANA

RAZA MEXICANA • MEXICO • NUMERO 17 • FEBRERO 2005 • WWW.RAZA-MEXICANA.ORG

Hacker tras las rejas

Raza-Mexicana

Se que lo siguiente sonará muy personal y que mi posición de editor no me permite hacerlo pero siendo honestos es algo que me hizo reflexionar mucho y me gustaría compartirlo con la raza.

Hace como dos años mi Madre compró un perico (bueno, la verdad nunca supimos si era perico o cotorro o pájaro silvestre, pero nosotros creíamos que era perico por lo verde) en un tianguis popular (mercado sobre ruedas) con la intención de que la casa tuviera vida y que el animal repitiera palabras obscenas; al principio parecía enfermo, casi no comía y no producía ningún ruido, solo se la pasaba en su palo mirando al piso, pasarían como tres semanas cuando ya dejó de mirar al piso y comenzó a moverse mas, ya se posaba en los barrotes de su jaula y ya comía mas y hacia algo que me llamó mucho la atención, se posaba en los barrotes y sacaba la cabeza, se estiraba lo mas que podía, luego se atoraba con sus alas y lo intentaba de lado, todos los días lo hacia, a cualquier hora del día que yo lo fuera a ver estaba haciendo lo mismo, sacando la cabeza tratando de salirse. Mi madre dio la orden de que se le pusiera alambre alrededor de la jaula para minimizar sus ganas de sacar la cabeza y así fue, le puse alambre pero el perico seguía intentándolo, picaba y picaba el alambre hasta que un día logró romperlo y así siguió saque y saque la cabeza. Yo era de la idea de dejarlo salir y ver que hacia, me imaginaba que solo quería sentir el suelo bajo sus patas o que quizá quería volar un rato, o que había dejado a su familia el día que lo capturaron. Abogué por él varias veces, pero mi Madre se obstinó a que se quedara en la jaula y que algún día dijera obscenidades, pero no fue así, lo mas que pudo hacer fueron ruidos escandalosos en las mañanas y ya. Mi Madre en las noches lo metía porque al gato del vecino le gustaba ir a molestarlos por las noches, y yo en las mañanas que pasaba a la cocina lo veía haciendo lo que mejor sabía hacer: sacar la cabeza e impulsarse para salirse. Así pasaron los días y nunca dejó de internarlo, día a día trataba y trataba, ya nunca recuperó su plumaje que se le fue desgastando al rozar con los barrotes; así fue hasta que un día lo logró.

Esa mañana se me hacia tarde para salir de casa, pero no podía pasar por alto mi desayuno, así que me dirigí hacia la cocina, y fue entonces cuando lo vi, afuera de su jaula, me miró y si hubiera tenido labios para gesticular una sonrisa se que lo hubiera hecho. Por mi cabeza pasó la idea de atraparlo y meterlo a la jaula, pero hubo algo en él que se me hizo familiar… Lo único que hice fue salirme sin desayunar y dejar la puerta abierta, lo que fue de él después de ese día ya no importa lo que importa es que lo que vi en él fue lo que he visto en varias personas; es esa lucha diaria por lograr los objetivos, es no darse por vencido y seguir intentando las cosas no importando si alguien o algo se nos pone en el camino, no importando cuantas veces se llegue al mismo resultado erróneo. Se que los ortodoxos dirán que era puro instinto, pero yo creo que era mas que eso. Toda cadena es tan fuerte como lo es su eslabón más débil, se que nunca hubiera podido romper los barrotes con su pico, pero la puerta creo que era la mejor opción para salir de allí, hay gente que se la vive luchando contra los barrotes sin darse cuenta que la salida es por otro lugar y peor aun, hay gente que los barrotes solo los tiene en su mente y esos barrotes imaginarios no los dejan salir.

Una vez El Viejo (deadsector) me dijo algo muy cierto: “Ser hacker no tiene nada que ver con computadoras o sistemas operativos, ser hacker es un modo de vida y una manera de enfrentar los problemas; un non-hacker se encuentra un problema y dice: ‘no se pudo’, un hacker se pregunta porque no se pudo y busca la manera de solucionar el problema y no se da por vencido. Un hacker en computadoras es también una persona que le gusta solucionar problemas, que va a investigar primero para descubrir el problema y luego para encontrar la solución, hay gente que no saben ni siquiera identificar los problemas y mucho menos les interesa buscar solución. Un hacker nace no se hace, no puedes ser hacker solo aprendiendo a debugear programas o codeando o hackeando páginas.” Creo que ese perico tenia algo de hacker, ya se que van a decir que estoy bajo el influjo de una droga, que estoy denigrando a los hackers del mundo, que se me caerá el pene por blasfemar, pero no me importa lo que digan, la verdad es que el perico nunca se rindió, luchó y venció, su actitud frente a la vida nunca fue de derrota. Coincido con lo que dice El Viejo, el hack es un estilo de vida, una forma de enfrentar los problemas, es algo que no está ligado con las computadoras o el internet, es algo que tiene que ver con lo cotidiano, y si los muy ortodoxos dicen que estoy malempleando el término de hacker pues entonces pongámosle otro nombre como jaquer o jaquero, el título no importa.

Este ezine va para las mentes inquietas, revolucionarias, innovadoras y creativas de este planeta, para los jaqueros que luchan día a día por lograr el cambio, por aquellos jaquers que se levantan en la mañana esperando que el día traiga nuevos retos, para esas mentes inquietas que buscan su lugar, por los jaqueros que se fueron y por los que están por venir y que este documento sirva un poco para quitar los barrotes que no nos dejan ser libres.

P.D. Ya había editado la ezine cuando El Viejo me roló un URL que reafirma lo aquí descrito: http://www-mitpress.mit.edu/catalog/item/default.asp?sid=0B984A34-C8D5-4FB0-A158-358C3A591CF3&ttype=2&tid=9559

Vlad

Contenido.

ANALISIS DE RUBY

Por ^NetXinG^ (Netxing@spymac.com)

Joven, potente y sencillo son las características que definen ha este nuevo lenguaje de programación, nacido en 1993 por el japonés Yukihiro Matsumoto que trató de combinar el poder de Perl, Lisp... etc. En este escrito se mostrarán algunas de sus características y por que se ha llevado tantos elogios en sus pocos años de vida.

¿Qué es Ruby?

Es un lenguaje de programación de guiones (scripts) interpretado, de muy alto nivel y orientado a objetos, su diferencia de los demás lenguajes se hace presente cuando comprendemos sus amplias características:

- Te permite realizar directamente llamadas al sistema operativo
- Potentes operaciones sobre cadenas de caracteres y expresiones regulares
- Retroalimentación inmediata durante el proceso de desarrollo

Rápido y sencillo.

Esto lo consigue al ser un lenguaje débilmente tipificado, ya que las variables carecen de tipo, y no es necesaria su declaración. También colabora a este objetivo el que su sintaxis sea clara y simple, además de tener recolector de basura (la gestión de la memoria se realiza de forma automática).

Programación orientada a objetos:

- Todo es un objeto
- Clases, herencia, métodos, ...
- Métodos singleton
- Mixins por módulos
- Iteradores y cierres

Ruby aporta también otras cualidades que son de agradecer como:

- Enteros de precisión múltiple
- Modelo de procesamiento de excepciones
- Carga dinámica
- Hilos
- Es completamente de código abierto
- Hay ports para todos los sistemas operativos populares
- Tiene librerías para todo lo imaginable

Ahora realicemos una comparación con el lenguaje Python (es cuando todo lenguaje se hace imperfecto o no actual).

Python es de esos lenguajes que no pasan de moda, su documentación es muy extensa y comprensiva, por supuesto cada día su comunidad de usuarios se extiende por todo el mundo, así como sus bindings para todas las bibliotecas creadas o no creadas, pero su orientación a objetos es muy pobre (tipo mi vecindad), talvez cuando salga alguna versión Parrot/Phyton se arreglará (xDD) si no comparten mi idea investiguen un poco de los atributos privados y me entenderán.

Ruby puede ser utilizado donde Perl, Python y otros lenguajes entran al ataque. Tareas diarias de scripting, programación para el web, multitasking y sockets entre otros, son modalidades en las que Ruby tiene el poder de intervenir. (Parecería que lo estoy vendiendo)

Inconvenientes en Ruby.

Hay muy poca información en inglés o español, la mas sobresaliente esta en japonés, pero esperemos y pronto haya buen documental.

Instalación y programación de Ruby.

Las versiones de Ruby se pueden conseguir en ftp://ftp.ruby-lang.org/pub/ruby/ para saber si ya lo tenemos instalado tecleamos desde la línea de comandos de la shell:

```
$ruby -v
(-v le indica al intérprete que imprima la versión de Ruby), utilizaremos
$ para indicar la shell
```

Si está instalado Ruby, aparecerá el siguiente mensaje o algo similar:

```
$ruby -v
ruby 1.8 (2001-11-23) [i586-linux]
```

Comencemos a programar en Ruby, se puede introducir directamente en la línea de comandos un programa Ruby utilizando el parámetro -e

```
$ruby -e 'print "hola mundo\n"'
hola mundo
```

Algunas cosas sorprendentemente complejas y útiles se pueden hacer con programas miniatura que caben en la línea de comandos. Por ejemplo, el siguiente programa reemplaza la cadena foo por bar en todos los ficheros cabecera y fuentes C del directorio de trabajo, realizando una copia de seguridad del fichero original a la que añade ".bak"

```
$ruby -i .bak -pe 'sub "foo", "bar"' *.[ch]
```

El siguiente programa funciona como el comando “cat” de UNIX (aunque es más lento):

```
$ruby -pe 0 file
```

Hay que tener en cuenta que la sintaxis de Ruby reproduce con más exactitud la del lenguaje Eiffel, también se puede apreciar la falta de la sentencia return en sus programas ya que es innecesaria debido a que una función Ruby devuelve lo último que haya evaluado. La utilización de return es factible aunque innecesaria. Ruby también puede tratar cualquier entero que quepa en la memoria del ordenador, incluye un programa llamado eval.rb que permite la introducción de código desde el teclado a través de un bucle iterativo que muestra los resultados a medida que se obtienen ejemplo:

```
$ruby eval.rb
ruby> print "hola mundo\n"
hola mundo.
  nil
ruby> exit
```

Print produce hola mundo. La siguiente línea, es este caso nil informa sobre lo último que se ha evaluado; Ruby no distingue entre sentencias y expresiones, por lo tanto la evaluación de una pieza de código significa básicamente lo mismo que ejecutarla. Aquí nil, indica que print no devuelve ningún valor significativo. Obsérvese que se puede salir del bucle de interpretación con exit, aunque también funciona ^D (ctrl.+d), por algo llega a ser útil este pequeño programa eval.rb.

El manejo de las cadenas es más inteligente e intuitivo que en C. Por ejemplo, se pueden concatenar cadenas con + y se puede repetir una cadena varias veces con *:

```
ruby> "foo" + "bar"
"foobar"
ruby> "foo" * 2
"foofoo"
```

Estos son simples ejemplos de los cuales se puede ver una potencia muy aceptable a la hora de estar programando, todo lo que lees aquí fue sacado de libros (japoneses), foros, programadores e internet... saludos a ellos, este es mi primer artículo, solo quiero agradecer raza-mexicana, y saludos a todos los que la conforman.

GPS, SISTEMA DE POSICIONAMIENTO GLOBAL

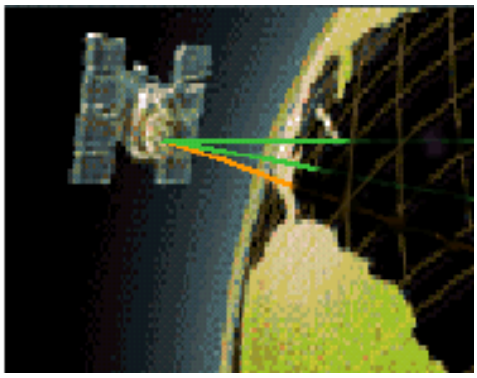
Por piojero (piojero@gmail.com)

En este pequeño artículo escribiré un poco de lo que es GPS, y los usos que actualmente tiene tanto civil como militar. Solo quiero decir que este es un texto introductorio hacia el tema, a medida que aprenda más sobre GPS, iré avanzando en cuanto a dificultad con posteriores artículos ya que el tema es muy interesante.

GPS, o Global Positioning System, fue creado en la década de los 70 por el DoD, (Departamento de Defensa de los Estados Unidos), los cuales utilizaron este sistema con fines puramente militares como herramienta de logística y como complemento para el guiado de armas (todavía se utilizan en todos los conflictos actuales).

Si nos situamos en el plano histórico (década del 70, mundo bipolar), podemos deducir que el bloque socialista liderado por la Unión Soviética también poseía su propio sistema de posicionamiento satelital. Este se llama GLONASS, no fue lo suficientemente difundido por la caída de este bloque.

GPS, fue el desarrollo posterior a OMEGA, un sistema de navegación instalado en tierra. Las ventajas de un sistema de navegación satelital son muchísimas en comparación al sistema de navegación terrestre, por eso el cambio.



Disposición en el espacio.

El sistema GPS, consta de 24 satélites distribuidos en distintos puntos, de esos 24, 21 satélites son los que funcionan y los 3 restantes quedan de reserva. Como es de esperar, estos satélites son geoestacionarios. No tienes idea de que es esto?, tranquilo que todo tiene una respuesta ;). Los satélites se clasifican según su distancia a la tierra, en tres tipos:

Satélite de orbita baja: trabajan en una altura de entre los 800 a 7000 Km. de altura. Este tipo de satélites son muy utilizados por empresas de telefonía para transmisión de voz.
Satélite de orbita Media: Están dispuestos a 19300 Km. De altura con respecto a la tierra.

Satélite Geoestacionario (o de orbita alta): Estos son los satélites que utiliza GPS. Están estacionados a 35890 Km. de altura, en la orbita geoestacionaria, lo que les permite dar la vuelta entera en su orbita en 24 horas. Son los únicos que se mantienen en algún lugar geográfico (lógicamente desde el punto de vista de coordenadas). Esta es una cualidad sumamente importante, ya que de la distancia entre los satélites, dependerá la triangulación, y por consecuente la ubicación de algún objeto con receptor GPS.

GPS posee 6 sectores equidistantes de 4 satélites cada uno, que se desplazan siguiendo la rotación de la tierra. Por otra parte, GLONASS se distribuye con 3 sectores equidistantes de 8 satélites

Si hablamos de disposición en el espacio, no debemos olvidarnos también de las estaciones de control. En una base aérea en Colorado esta ubicada la estación principal y las otras están distribuidas en algunas islas ubicadas en distintos lugares (Hawai, Ascensión, Diego García, Kwajalein) con el objeto de regular la estabilidad de los satélites en las orbitas.

Ubicación de las estaciones.

Señales del GPS.

En GPS, el sistema de señal se basa en la utilización de dos códigos diferentes en la misma portadora de señal más uno adicional que tiene la función de reserva al código preciso. Para ser un poquito más claro, estas líneas de código son:

- Código C/A (Course Acquisition): Este es el código con el que trabajan los usuarios civiles. Tiene una frecuencia de 1.023 Mhz y posee una longitud de 1023 bits.
- Código P (Precision Code): Al ser éste el código preciso, creo que no hace falta decir que tipos de usuarios lo utilizan. Trabaja con una frecuencia de 10.23 Mhz (si, es 10 veces más grande que el código C/A). Este código utiliza 2 subtramas adicionales que nos permiten mejor precisión. Las subtramas mencionadas tienen 15.345.037 y 15.345.000 bits (este tema lo tengo poco claro así que prefiero dejarlo así, cuando aprenda bien como esta compuesto escribiré un poco más...).

- Código Y (Antispoofing code): Este código es también de uso estrictamente militar. Consiste básicamente en un código preciso de uso únicamente clasificado, con la función de encriptarse para evitar engaños por parte de otra persona. Está vigente desde 1994.

Perfecto, una ves visto los códigos que tiene GPS, es necesario determinar las frecuencias con las cuales son enviadas. Podemos distinguir dos frecuencias diferentes por donde son enviados los códigos:

- Frecuencia L1: Frecuencia a 1575.42 Mhz. En esta frecuencia se envían los códigos C/A y P. Posee una longitud de onda de aproximadamente 190 mm.
- Frecuencia L2: con una intensidad de 1227.60 Mhz. Esta es la frecuencia que hace trabajar al código Y. Posee una longitud de onda de aproximadamente 240 mm.

Niveles de utilización de GPS

El sistema de posicionamiento consta básicamente de 2 niveles de servicios aplicables, una para uso civil (C/A) y otro para servicio militar (P). Ahora pasaré a explicar un poco las características de cada uno:

- SPS (Standard Positioning Service) : Para uso civil. Consiste básicamente en un servicio disminuido de GSP dispuesto por el DoD debido a medidas de seguridad. El uso de este servicio no tiene restricción ni cargo alguno, con lo que podemos utilizar constantemente un receptor de GPS sin pagarle absolutamente nada a nadie. El rango de error es de aproximadamente 100 metros en el plano horizontal (ejes X y Z) y 160 metros en el plano vertical (eje Y). La precisión en cuanto al tiempo es de 170 nanosegundos.
- PPS (Precise Positioning Service) : Restringidos únicamente para uso militar o para uso civil autorizado. Es aproximadamente 7 veces más preciso que el SPS, tiene una precisión horizontal de 18 metros (ejes X y Z) y una precisión vertical de 27 metros (eje Y). La exactitud temporal es de 100 nanosegundos.

Como ven, el tema hasta ahora viene muy entendible, (o por lo menos eso es lo que trato de hacer) solo espero que entiendan esto y cualquier duda mail me.

GPS Meridian de uso civil.

Tramas y subtramas en GPS, el envío de datos.

GPS está formado por tramas de 1500 bits, los cuales a su vez, están compuestos por cinco subtramas de 300 bits. Un mensaje de GPS esta compuesto por 25 tramas, que se van generando cada 12,5 minutos. A partir de aquí, deducimos que cada trama se genera cada 30 segundos.

Bien, ahora explicaré un poco como están compuestas las subtramas para determinar como se envían los datos, tanto desde el punto de vista del usuario, como el punto de vista del satélite.

Trama Completa = 1500 bits	
Primera subtrama	Datos de telemetría (TLM): primeros 8 bits en subtrama. El handover (HOW): que permite el cambio de C/A a P. Básicamente en esta subtrama se realizan las correcciones previas (reloj, atmósfera, etc). El final de todas las subtramas están compuestas por unos bits de paridad que corrigen datos finales.
Segunda subtrama	TLM, HOW, mas datos de efemérides (orbita exacta del satélite) + datos de paridad.
Tercera subtrama	TLM, HOW, mas datos de efemérides + datos de paridad. (espero que para la próxima pueda exponer más información sobre la segunda y tercera subtrama)
Cuarta subtrama	TLM, HOW. En esta trama se arma el dato de navegación una vez completada las 25 tramas (con 125 subtramas). Datos de paridad al final.
Quinta subtrama	TLM, HOW. Almanaque (con este dato se clasifican los satélites mejores posicionados para realizar la transferencia de datos) para los satélites y estado de la constelación de los mismos.

Estos son los datos utilizados por GPS. Vale aclarar que dichos datos son procesados por el MCC, (Master Control Center) el cual procesa las subtramas, calcula las efemérides y en caso error la corrige con los datos de paridad. Solo queda decir que el envío de paquetes de información se da en un ancho de banda de 50 bps.

Errores en GPS.

Y si, GPS no es perfecto y puede jugarnos malas experiencias si no tenemos en cuenta que por ciertos factores, la información que nos muestra nuestro receptor puede ser no del todo correcta.

- Errores provocados por la Atmósfera: La naturaleza de la atmósfera está compuesta por muchas partículas que desgraciadamente atentan contra una medición exacta de nuestro GPS. Se destacan los errores por perturbación ionosférica, en donde se pueden apreciar errores en precisión de aproximadamente 10 metros.
- Errores en procesamiento de datos: el mal calculo de las efemérides, la microscópica pero segura imprecisión de los relojes atómicos en los satélites, y los errores en la posición de orbita, implican errores de calculo que dan imprecisiones de 1 metro aproximadamente.
- Errores provocados por Disponibilidad Selectiva: Es un error impuesto a propósito por la DoD, el rango de error es de 70 metros aproximadamente.
- Error del receptor GPS: si nuestro recepto tiene alguna falla interna, por ende no nos dará información precisa.

Estos rangos de errores es lo que forma para los creadores de GPS el CEP (Error Circular Probable) el cual nos da un rango circular del dato en el receptor, con una probabilidad de acierto de un 95 %.

Elevando la precisión: DGPS.

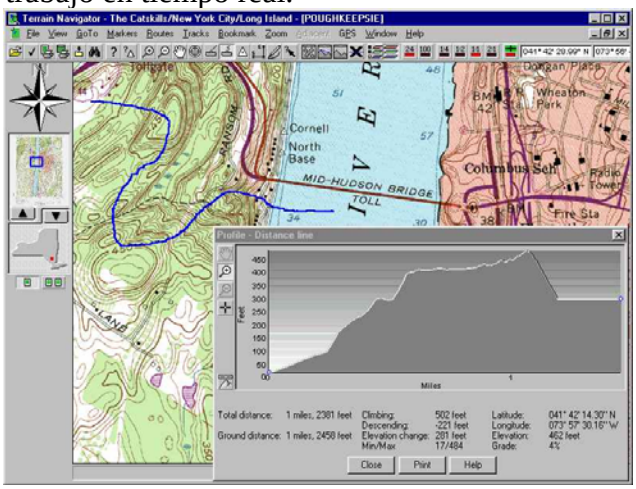
Existen una serie de técnicas que nos permiten elevar el rango de precisión de nuestro GPS. Estas se llaman DGPS (GPS diferencial) y consisten básicamente en el uso de software especializado que permite calcular y mejorar las pseudo-distancias que recibió nuestro GPS.

Para obtener mayor precisión, lo único que haremos será comparar los datos obtenidos en una central la cual reciba señales GPS y conozcamos su ubicación. Una vez que tengamos los datos de esta central, nos fijamos en los datos que recogió nuestro receptor y allí sacamos los parámetros necesarios para determinar la diferencia que provoca el error en la medición.

Software para GPS.

Con la aparición de software en GPS, el rango de exactitud se ha elevado considerablemente, estos programas permiten mejorar las pseudo-distancias que producen errores en la medición. A continuación, haré mención sobre algunos programas de GPS:

- Software de ruteo: Son aquellos programas que utilizan GPS en conjunto para determinar waypoints en algún trazado de ruta. Es utilizado principalmente por aviones y barcos. Muchos de estos programas incrementan considerablemente la precisión del código C/A y algunos hasta hacen uso del código P (GPS en aviones de combate, etc).
- Software de topografía: Son programas que recogen información sobre la forma de la topografía utilizando GPS. Un programa muy conocido en este genero es el MapTech Terrain Navigator, el cual nos muestra datos de topografía en 2D y en 3D, corre en pc´s de escritorio y en laptops y acepta la conexión de equipos GPS para trabajo en tiempo real.



MapTech en modo 2d.

Bien, creo que mejor lo dejamos acá, solo quiero decirles que si tienen alguna duda sobre algo que esté poco claro, por favor escriban a la dirección de mail que esta arriba que con gusto los ayudaré (en lo que pueda claro está).

Webs de interés:

http://www.maptech.com/land/index.cfm (página de MapTech)
http://www.gpsmundo.com/index.asp?DocumentID=745 (manuales de hardware GPS)

WINDOWS XP SP 2 LIVE CD

Por DeadSector (deadsector@raza-mexicana.org)

Aquí aprenderemos a crear un livecd de Windows XP SP2 usando bartPE (Bart Preinstalled Environment). Un LiveCD es un CD-ROM booteable de Windows XP o Windows 2003 Server que te permite correr el sistema operativo desde RAM sin tener que instalar nada en disco duro. Tendrás el ambiente completo win32, interfaz grafico, soporte para filesystem FAT o NTFS y volúmenes hasta mas de 2 TB, tendrás acceso a la red, active directory y podrás usar herramientas como vnc o remote desktop para controlar otras maquinas. Es la manera perfecta para limpiar virus de una maquina, cambiar el password de administrador local si lo perdiste, recuperar archivos borrados del disco duro o recuperar archivos de disco duro en caso de que hayas formateado tú maquina por error. Todo depende de los plugins que vayas agregando a tu livecd

Necesitaras una copia de WindowsXP con SP2 y los siguientes archivos:

- Bartpe versión 3.1.3 http://www.nu2.nu/pebuilder/
- WindowsXPE plugin http://oss.netfarm.it/winpe/
- HWPnP http://www.paraglidernc.com

Voy a copiar los archivos de bartPE v3.1.3 a una carpeta llamada d:\bartpe. De mi CD de Windows XP con sp2 integrado voy a copiar la carpeta i386 a d:\i386

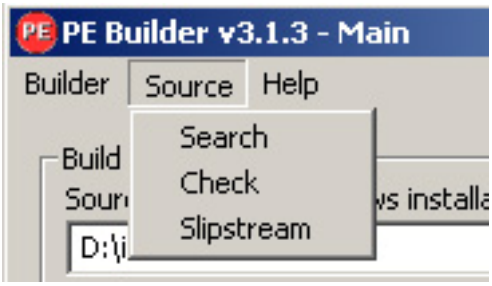
Lo primero es ejecutar pebuilder.exe. En la ventana principal te pide "Source:" donde tienes tu carpeta i386 en mi caso seria d:\ sin el i386

"Custom:" es una carpeta en tu disco duro donde tienes programas que quieras copiar a tu cd.

"Output:" es el lugar donde bartpe copiara todos los archivos para generar la imagen ISO de tu cd, usare los defaults que en mi caso serian d:\bartpe\bartpe

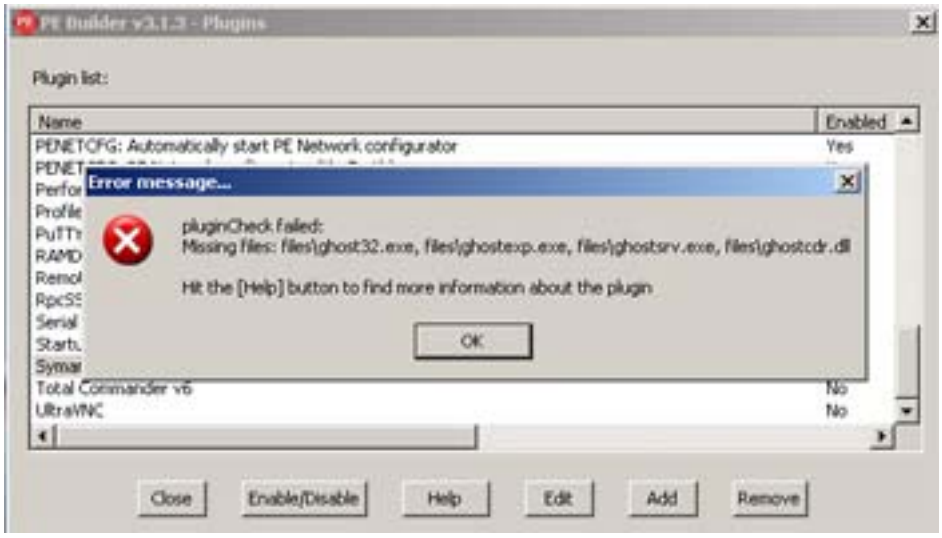
Hay que seleccionar la opción de crear una imagen ISO y si quieres quemar tu cd cuando termines entonces selecciona la opción de Burn to CD.

En caso de que no tengas Windows XP con sp2 slipstreamed, pero ya tienes copia de sp2 en tu disco duro bartpe te puede ayudar a hacer el trabajo.



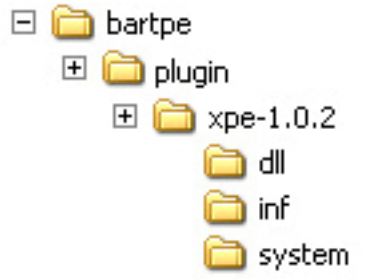
Seleccionas la carpeta de tu windowsXP source, y la carpeta donde tienes el archivo de SP2. Si estas usando un cdrom tienes que seleccionar la opción de Source is readonly y en output seleccionar la carpeta en donde quieres copiar tus archivos actualizados. Con esto tu carpeta de instalación de Windows XP tendrá integrados todos los parches que vienen con SP2.

Ahora viene lo bueno, en pantalla principal oprime PLUGINS para agregar nuestros plugins y tendrás el siguiente menú. Seleccionas ADD y agregas xpe-1.0.2.cab. Este archivo se supone que ya lo bajaste de links que puse a principio de artículo. Cuando agregas un plugin se genera una carpeta en bartpe\plugins y ahí tienes que copiar los archivos necesarios. Para saber que necesitas puedes seleccionar el plugin y el botón de HELP. Te dará un listado de los archivos necesarios y donde debes copiarlos.



Por ejemplo selecciona el plugin mcafee stinger y pícale al botón de HELP, saldrá una ventana con las instrucciones de archivos que necesitas bajar de Internet y copiar a la carpeta correcta. Ya que tengas los archivos copiados correctamente presiona botón de ENABLE/DISABLE. Si te falta algún archivo el mensaje de error te dirá cuales son. Una ves agregado el plugin XPE hay que deshabilitar el plugin STARTUP GROUP y PE LOADER y nu2Shell v1.0 .

Algunos plugins tendrás que bajar para poder agregarlos a bartpe como Chntpw Local Password Changer, Easy Recovery Pro 6 [AvB], FoldersReport v1.10, Registry Editor PE v0.2b, passware. http://www.raza-mexicana.org/programas/bartpe tendrá algunos plugins que mencionamos en este artículo. Si el plugin viene en formato CAB lo agregas con opción ADD, si viene en zip y no contiene cab entonces solo extraes los archivos a la carpeta d:\bartpe\plugins



Una vez instalados los plugins hay que crear 3 carpetas (DLL, INF y SYSTEM) en la carpeta del plugin XPE (D:\bartpe\plugin\xpe-1.0.2)

D:\bartpe\plugin\xpe-1.0.2\DLL
D:\bartpe\plugin\xpe-1.0.2\INF
D:\bartpe\plugin\xpe-1.0.2\SYSTEM

Busca el archivo d:\i386\driver.cab y copia todos los archivos *.dll y *.exe a la carpeta D:\bartpe\plugin\xpe-1.0.2\DLL puedes usar winzip para sacar los archivos de driver.cab. Copia todos los archivos *.sys a la carpeta D:\bartpe\plugin\xpe-1.0.2\SYSTEM.

Busca el archivo d:\i386\sp2.cab y copia todos los archivos *.dll y *.exe a la carpeta D:\bartpe\plugin\xpe-1.0.2\DLL puedes usar winzip para sacar los archivos de sp2.cab. Copia todos los archivos *.sys a la carpeta D:\bartpe\plugin\xpe-1.0.2\SYSTEM.

Edita el archivo D:\bartpe\plugin\xpe-1.0.2\xpe-defaults.inf y cambia las líneas 42 y 43. Quítales el “;”

```
; Required
 0x2,"ControlSet001\Control\Session
Manager\Environment","USERPROFILE","%temp%\@ProfilesDir@\Default User"
 0x2,"ControlSet001\Control\Session
Manager\Environment","ProfilesDir","%temp%\@ProfilesDir@"
```

Estarás diciendo WOW eso fue fácil! Ya terminamos?? Pues no. Todavía falta configurar tu desktop con tu wallpaper de britney spears, paris hilton, los simpson, matrix, lord of the rings, imágenes de revolucionarios mexicanos o algún logotipo de Raza Mexicana.

Copia tu imagen de britney.bmp a la carpeta d:\bartpe y renómbrala a bartpe.bmp. Tendrás que borrar la imagen original que existe en esa carpeta. En mis pruebas no funciono con una imagen jpg pero al convertirla a bmp con paint funciono sin problemas.

Para modificar tu interfaz grafico tendrás que editar el archivo llamado D:\bartpe\plugin\xpe-1.0.2\z_xpe-custom.inf.sample y una vez terminados tus cambios renombrarlo a z_xpe-custom.inf.

Lo primero que vamos a modificar en z_xpe-custom.inf es la línea 22 de loaderprompt y le pondremos algo elite como “recalibrando sensores y capacitores Flux..”

```
[SetValue]
"txtsetup.sif","SetupData","loaderprompt","""recalibrando sensores y
capacitores Flux..."""
```

Para cambiar la barra de START y ponerla abajo sin que se esconda cambiamos las siguientes líneas: ponle comentarios a las líneas 427,428 y 429 “;”

```
; TaskBar on Top - Autohide
;0x3,"Software\Microsoft\Windows\CurrentVersion\Explorer\StuckRects2","Se
ttings",\
;
28,00,00,00,ff,ff,ff,ff,03,00,00,00,01,00,00,00,3c,00,00,00,1e,00,00,00,f
e,\
;  ff,ff,ff,fe,ff,ff,ff,02,04,00,00,1c,00,00,00
```

Y quítaselos a las líneas 447,448 y 449 para tener la barra de start abajo y que no se esconda

```
; TaskBar on Bottom - No Autohide
0x3,"Software\Microsoft\Windows\CurrentVersion\Explorer\StuckRects2","Set
tings",\

28,00,00,00,ff,ff,ff,ff,02,00,00,00,03,00,00,00,3f,00,00,00,1e,00,00,00,f
e,\
  ff,ff,ff,e4,02,00,00,02,04,00,00,02,03,00,00
```

Ahora falta crear shotcuts para cada plugin y esto también se hace modificando el archivo D:\bartpe\plugin\xpe-1.0.2\z_xpe-custom.inf. Por ejemplo agrega las líneas:

```
[Software.AddReg]
0x2,"Sherpya\XPEinit\Programs","Adaware","%SystemDrive%\Programs\adawares
e\ad-aware.exe"
0x2,"Sherpya\XPEinit\Programs","McAfee
Antivirus","%SystemDrive%\Programs\mcafee\scangui.exe"
0x2,"Sherpya\XPEinit\Programs","EasyRecovery","%SystemDrive%\Programs\Ont
rack\EasyRecovery Professional\easyrecovery.exe"
0x2,"Sherpya\XPEinit\Programs","Messenger2","%SystemDrive%\Programs\Messe
nger2\messenger2.cmd"
0x2,"Sherpya\XPEinit\Programs","Remote                                Desktop
Connection","%systemroot%\system32\mstsc.exe"
```

Para agrupar shortcuts dentro de un subfolder solo agrégale el nombre del fólder:

```
[Software.AddReg]
0x2,"Sherpya\XPEinit\Programs","Passware\1Key","%SystemDrive%\Programs\Pa
ssware\1key.exe"
0x2,"Sherpya\XPEinit\Programs","Passware\2Key","%SystemDrive%\Programs\Pa
ssware\2key.exe"
```

Así saldrán en START , Programs , Passware , 1key y 2key. Para poner un icono en desktop :

```
[Software.AddReg]
0x2,"Sherpya\XPEinit\Desktop","Command Prompt","%comspec%"
```

Y para terminar solo presiona el botón de BUILD en la pantalla principal de bartpe. Si seleccionaste la opción de BURNCD tendrás tu cd booteable. Yo prefiero crear solo el ISO y uso programas alcohol 120% y vmware para hacer las pruebas. Cuando estoy contento con el cd entonces lo quemo.

Ahora te toca buscar plugins o crear los tuyos. Aquí están algunas páginas donde tienen herramientas o plugins que pueden servir para agregar a tu cd.

http://www.nu2.nu/pebuilder/
http://www.dirk-loss.de/win-tools.htm
http://xpe.collewijn.info/
http://www.pecd.net/modules.php?name=Forums&file=index

Cualquier pregunta que tengas me la puedes hacer por email o entrando a IRC Server irc.raza-mexicana.org puerto 30003. Nos vemos.

EVOLUCIÓN

Por data_gate (finefine@gmail.com)

Después de visitar en los foros del CUM, y de visitar algunas páginas mexicanas decidí escribir este artículo en el cual expondré una ayuda para todos aquellos que inician en la informática.

Antes que nada, al mencionar “hacker”, me refiero al significado que las personas comunes sin conocimiento de informática, le dan, el cual podría ser: una persona que entra a otros sistemas y ataca o hace algo malo. Tengan en cuenta que hay muchos significados y que nosotros podemos decir que es un buen programador u otra cosa, pero al fin y al cabo para las personas sin conocimiento de informática será el mismo, gracias a las películas y la publicidad en la TV, cine u otros medios.

La Historia

Hace varios años empecé en todo este ambiente de los sistemas informáticos, y en ese momento creí que lo indicado era crearme un nick y tratar de hacerme conocido bajo ese nick. El nick que elegí en ese entonces era algo tonto y contenía corchetes.

Al iniciar, conocí lo que eran los troyanos como Subseven, netbus y demás utilerías que había en ese tiempo, al momento de usar subseven me sentía emocionado por aquello que creí estar haciendo, creía que por fin era un “hacker” hecho y derecho. Al principio no notaba cuan equivocado e inmaduro me encontraba, hasta que conocí el IRC, y me encontré con personas que sabían mas que yo (siempre habrá alguien mas listo que tú) y aprendí nuevos términos en la red, como “lamer” y “l33t”, en ese momento me sentía la persona mas lista del mundo en cuanto al Internet se relacionaba.

En ese entonces estaba de moda la película “Hackers”, dicha película influyó en mi vida de una manera marcadamente estúpida. Nunca en esa etapa de mi vida me di cuenta que simplemente era una película de ciencia ficción, y que no era para adoptar un “modus-vivendi” en razón de ella.

Para ese entonces, al pasar el tiempo, me di cuenta de que el usar Troyanos no me hacia mas que ser una persona “lamer”, y de lo peor en la red. Fue entonces cuando decidí, cegado por la emoción de una película de ciencia ficción, dejar de usar los Troyanos y comencé a leer información acerca del significado y uso de las “shells” y la manera de entrar a otras computadoras a base de algo que le decían 3xpl01tz.

Otra vez, creí que ya sabia todo acerca de la red, sentía que era lo mejor de lo mejor y que nadie podría superarme, y si alguien lo hacia, lo negaba rotundamente. Fue entonces cuando inicie un grupo de “Hackers”, entre los cuales elegiría a los mejores de la red y así formaríamos un grupo al cual volvería famoso y conocido en la “comunidad”.

Para ese entonces sabia como bajar un exploit de alguna pagina de vulnerabilidades y sin leerlo, sin saber nada de cómo estaba compuesto, lo ejecutaba en alguna shell que había logrado conseguir a cambio de un software o una tarjeta de crédito generada por un programa el cual me había pasado otra persona que lo había conseguido gracias al amigo de su amigo de su tío, y así obtenía una shell mas en otro sistema.

¿Qué hacer con la shell?

En realidad no hacia nada, no sabia que hacer, daba acceso a otras personas, o borraba algo, o cambiaba el index.html en la carpeta del servicio Web. Así conocí lo que eran los 'defaces' o más correctamente "defacements".

En el momento que conocí los deface's, empecé a ver a la demás gente como personas incultas en el ámbito de la informática, sin importar quien era. Y así, empecé a hacer deface's, sin sentido, mandando saludos, o diciendo comentarios aun mas vagos e insípidos, sin una idea real. Entre los demás de mi grupo era lo máximo, en ese tiempo estaba en su apogeo todo eso…

Al poco tiempo de ello, caí en un ciclo del cual es muy difícil salir, y en el cual muchas personas se quedan encerradas toda su vida, ese ciclo es el de hacer lo mismo y subir su ego, complacer esa necesidad narcisista que a todos nos ha tocado sentir, unos mas, otros menos.

Pasó algún tiempo y me encontré con otro grupo de personas las cuales tenían aún más conocimiento de lo que yo podía tener, y me empecé a juntar con ellas…

Para ese entonces había ya pasado un tiempo usando un sistema operativo llamado Linux. Había conocido ese sistema operativo gracias a una revista que había comprado sin querer. En ese entonces no conocía nada de Linux, y en realidad no sabía que uso podía darle. Solo sabia que el usar Linux "me haría una persona diferente a las demás".

Tiempo después de haber convivido con esas personas las cuales tenían aun mas conocimiento de lo que creí haber sabido en ese entonces, me di cuenta que hacer deface's estaba mal, y que era una estupidez.

Al poco tiempo me dedique a aprender de Linux, y dejar de lado todo lo que eran esas cosas de "hackers".

En ese entonces volví a creer que estaba en la cima de todo, por el simple hecho de que sabia usar un sistema operativo llamado Linux, en el cual podía compilar archivos con extensión .c, y manejar todo en una pantallita negra.

Pero ahí empezó otro error... empecé a odiar a Microsoft sin motivos claros, en realidad era solo por decir que Microsoft era lo peor y que Linux era lo mejor. (Léase "Verdades" por DeadSector)

En esta etapa me burlaba de la gente que usaba Microsoft Windows, y le decía que “usar Windows solo provoca errores”.

En realidad por dentro era muy diferente... Me peleaba mucho con Linux y me quedaba con las ganas de hacer tantas cosas por el simple hecho de usar un sistema operativo llamado Linux.

Al poco tiempo de pasar esa etapa, me di cuenta de que esos ideales eran estúpidos, así que instale Microsoft Windows, y así me dedique aprender de Linux nada más, y verlo como una herramienta más.

Empecé asistir a congresos y a disfrutar de todo este mundo de la informática, en este entonces me di cuenta que no sabía todo de la informática y que nunca lo iba a saber todo, y que siempre habría alguien mas listo que yo.

Actualmente estoy metido en la programación y en mi empleo, disfruto tratando de aprender todo acerca de la informática, como esta compuesto un sistema operativo, los protocolos existentes en los diferentes tipos de topología de redes así como su funcionamiento.

La realidad…

Ahora bien…¿Alguien ha pasado esas etapas? ¿Les suena esa historia? Viendo toda esta historia, y dando una mirada hacia el pasado, me dan pena algunas etapas que pasé. Hay tantas personas que se quedan en un ciclo y nunca salen de esa inmadurez…

Hubiese dado todo por conseguir las verdades de este mundo informático desde la primera vez que use la computadora, saber que esos inicios no me llevarían a ningún lado bueno e iniciar aprendiendo de verdad, con este texto quiero que se percaten, aquellos que apenas inician, lo que realmente vale la pena…APRENDER.

SACAR CUENTAS PRODIGY

Por abadia(a_b_a_d_i_a@hotmail.com)

Paso 1

Veámonos al portal de prodigy www.prodigy.com.mx en la parte superior derecha nos da la opción de checar el famoso correo de 25 MB de regalo por ser parte de su servicio basura, ahí en los campos de usuario y contraseña pones el mismo usuario y contraseña, es decir:

Usuario : juan
Contraseña : Juan

Claro que vamos a utilizar un método de fuerza bruta o diccionario, ahora bien en el momento que nuestras claves de usuario y contraseña sean validas se va abrir el correo de prodigy de este usuario.

Paso 2

Nos vamos a la opción de dar de alta nuestros correos adicionales http://www.prodigy.com.mx/correos_adicionales/index.html (paso uno : aviso ¡¡ no tener adeudo en telmex etc,) paso dos: te pedirá que ingreses tu contraseña y usuario o sea el que descubrimos JUAN, si al activar este correo sale el siguiente mensaje: "Felicidades, da click para activar tus correos adicionales", automáticamente sabemos que este usuario y contraseña es también de acceso a internet y si en el ejemplo JUAN fuera un correo adicional ? saldría este mensaje : "CBE7 debe capturar la información de la cuenta que tiene acceso a Internet", así sabremos que esa cuenta sirve para internet y para correo y cual es simplemente un correo adicional (fácil no?).

Paso 3

¿Cómo saber que tipo de cuenta es? ok en algunas cuentas de correo viene aun el mensaje de bienvenida a prodigy, así sabemos en que fecha dio de alta su clave, solo los de prodigy infinitum dicen : "mail de bienvenida prodigy infinitum" ahí sabremos que esta cuenta es de prodigy infinitud, pero si solo dice miembro prodigy pues nos vamos a la opción de servicios prodigy hogar http://www.prodigy.com.mx/hogar/hogar_quees.html ahí encontramos una opción que dice: "si ya eres miembro del servicio consulta minutos adicionales fuera del servicio", ahí nuevamente ponemos nuestra contraseña JUAN y usuario JUAN si logramos entrar nos dará información de los minutos que has navegado fuera de el horario de prodigy hogar y automáticamente sabremos que esta cuenta de internet es prodigy hogar, si no rechaza la solicitud saldrá un mensaje que dice el usuario no tiene contratado prodigy hogar, ok esto es para prodigy hogar, para prodigy por minuto hacemos el mismo procedimiento pero en la pagina http://www.prodigy.com.mx/porminuto/index.html ahí encontramos la opción de cambiar usuario, es en la que vamos a trabajar ya que si vamos a la opción de consulta e incremento

ahí nos pide el teléfono de la cuenta (es mas difícil saber el teléfono, pero no imposible ya estoy en eso), pero en cambio de usuario solo nos pide usuario y contraseña, ponemos JUAN (ojo es bien importante esto) cuando lo pongas te dice que pongas el nuevo usuario y contraseña, DEBES PONER EL MISMO USUARIO Y CONTRASEÑA, ¿Por qué? si la cuenta es de prodigy por minuto te dirá que el usuario ya existe y si no es de prodigy por minuto te dirá que el usuario no tiene contratado prodigy por minuto. NUNCA CAMBIES EL USUARIO Y CONTRASEÑA YA QUE SI LO HACES LA PERSONA DUEÑA DE ESA CUENTA NO PODRA ENTRAR A INTERNET Y LLAMARA PARA CAMBIARLA Y TU PUES YA TE LA PELASTE OK. Bueno si entramos a prodigy hogar y por minuto y nos dijo que no tienes contratado esos planes las cuentas o son infinitum o prodigy ilimitado, así sabemos que plan tiene contratado (ojo también hay que saber los planes prodigy de servicio para no cagarla y no entrar con una cuenta de prodigy por minuto ya que su sistema registra el tiempo y el teléfono de donde se conectó y se lo cobra al usuario, trata siempre de encontrar cuentas de acceso ilimitado u hogar de 6pm a 8 am del otro día.

Bueno pues para que no se quiebren la cabeza con usuarios y contraseñas ahí les va el tip: día /mes/año ejemplo : 210578, agárrate un año y dale todos los días y los 12 meses y por lo menos obtienes 15 contraseñas NO SABEN CUANTA GENTE PONE ESTO... JA JA JA JA JA. Aquí les dejo algunos que son buenos, chequenlos :

Usuario y contraseña

eliasayub
slimcarlos
sanborns
aeromexico
anaguevara
110250
230350
280350
150550
290550

Prodigy tiene cobertura en toda la republica así que si tienen un contraseña de Guadalajara solo tienen que configurarla a su estado, en la misma página de prodigy están los teléfonos de área local y para conectar de cada estado y ciudad, así navegarás con una cuenta de otro estado pero configurada a tu estado, así la llamada será local. Bueno pues me despido de ustedes, suerte.

UN POQUITO DE ESTEGANOGRAFÍA

Por piojero (piojero@gmail.com)

Consideremos a la esteganografía como el arte de esconder datos de importancia bajo otra forma determinada de información que despiste a cualquiera que desee ver el contenido secreto.

Actualmente, un gran porcentaje de la comunidad que navega por Internet considera poco probable el hecho de que otra persona pueda estar mirando una determinada información que se ha enviado a otro destinatario. Y no es de sorprender, ya que la mayoría de estas personas desconocen totalmente el recorrido que tiene dicho dato (sea importante o no) hasta la llegada al destinatario. Es así como día a día nos encontramos con noticias en donde se muestra un fraude por carding gracias al descuido del usuario victima. Ahora, ¿ no es seguro enviar un mail en donde el contenido no este encriptado o "escondido"?. Y la respuesta a esta pregunta es muy variada, ya que si sos Don Nadie de la Nada, es muy difícil que alguien este interesado en saber que envíes; pero si tu eres el administrador de algún servidor "jugoso", mejor toma algunas medidas para evitarte problemas.

Gracias a un grupo de personas que se dio cuenta del riesgo que corremos al toparnos con un sniffer cuando enviamos algo por mail, nace el software encargado de encriptar archivos, y los programas encargados de "esconder" la información de modo que esta no parezca importante. Personalmente considero que una persona tiene que hacer el uso en conjunto de este tipo de programas para estar seguro a la hora de enviar algo importante, ya que si alguien te escaneo cierta información encriptada, ya se sabe que hay contenido oculto allí, pero si tu información va como una imagen o como un sonido, es mas complicado identificar que hay algo allí dentro.

Entonces, ¿qué necesitamos para ocultar nuestros archivos de miradas entrometidas? Los ingredientes son pocos y están al alcance de cualquier mortal (a si que a no aflojar que esto es bien básico y te aseguro que te evitaras mas problemas de los que ya tienes) solo necesitamos alguna versión del programa Steganos Security Suite, un poco de tiempo y muchas ganas de aprender. ¿cómo? ¿en donde consigues bajarte Steganos? Ve a http://www.vollversion.de/files/sss4de_vv.exe, http://www.donwload.com y listo...(mas claro que el agua no?)

Ahora si nos meteremos un poco en el asunto. La esteganografía puede utilizarse para imágenes digitales y para sonido digital. Ahora entraremos un poco en lo que es cada una de ellas.

Personalmente considero que no existe ventaja alguna entre ocultar datos en imágenes o en sonido, esto es por que a pesar de ser mínima, se puede apreciar una baja en la calidad de la imagen o el sonido, no obstante, esa disminución es mínima y muy difícil de detectar.

Esteganografía con imágenes.

Para ocultar un dato en una imagen, tenemos que tener en cuenta que la imagen debe mantener si o si su integridad como archivo, es decir, que posea sus valores binarios reales. Esto lo apreciamos por ejemplo si tenemos dos archivos (uno .GIF o .BMP, y otro .JPG) los cuales muestran la misma imagen. Seguramente el archivo .jpg pesara menos, ya que este es un recurso utilizado por el tipo de archivo en donde se comprimen los valores binarios, pero se pierde su integridad original. Este tipo de archivos, a pesar de ser los que mas nos hubieran convenido, no sirven por una razón lógica, cualquier alteración de su estructura binaria en donde agreguemos valores determinados, corrompería la compresión de los valores, haciendo al archivo ilegible y por lo tanto, inutilizable. Otra cosa importante a tener en cuenta, es la calidad de la imagen que servirá de despiste. A medida que la calidad de la imagen crezca, podremos introducir más cantidad de datos para ocultar.

Calidad de imagen	Cantidad de bytes por pixel	Prestación para ocultamiento
Imagen de 8 bits (256 colores)	1 byte	baja
Imagen de 16 bits	2 bytes	buena
Imagen de 24 bits	3 bytes	excelente

Prestemos atención al siguiente dato: si tenemos una imagen en formato GIF o BMP, el cual tiene un tamaño de 800x600 pixels nuestro GIF pesara aproximadamente 174 Kb, entre tanto nuestro BMP pesara algo así como 1,37 Mb con lo cual, nos conviene el primer formato ya que ocupa menos espacio y sirve para esconder información, pero hay que tener en cuenta que tenemos menos espacio a disposición para nuestro uso. Estos valores mencionados anteriormente, están dados con una calidad de 24bits, con lo cual el usuario verá si esta en condiciones o no de utilizar 24, 16 u 8 bits.

LSB, inserción del último bit significante.

El LSB (least significant bit, o inserción del último bit significante al castellano) es el método estándar para ocultar datos en la imagen. Como su nombre lo indica, en este método se utiliza el último bit de todas las cadenas de bytes por cada píxel empleado. Para aclarar un poco esto, daré un ejemplo:

Tengo mi imagen en formato BMP de 800x600 pixels de 24 bits y quiero saber cuanto espacio tengo disponible con este método, pues es simple: Sabiendo que en un píxel hay un byte de información, que en una imagen de 24 bits hay 3 pixels y que nuestra imagen tiene un tamaño de 800x600 pixels, calculamos el tamaño de la imagen total y después se divide por los bits en cada byte, de modo que así nos da el tamaño disponible:

800 x 600 = 480000 bytes (pero como tenemos una imagen de 24 bits)
480000 x 3= 1440000 bytes (peso total del archivo)
1440000 % 8 = 180000 bytes (cantidad disponible para usar y esconder datos)

Espero que se te hayan aclarados las cosas con esta explicación. Ahora sigamos ¿en que estábamos? Ha. si, el LSB. Como dije antes, usa el último bit de cada cadena, es decir que en un píxel tenemos a disposición 3 bits para hacer lo que se nos antoje, ahora ¿hay algún cambio en la imagen? Para ser precisos si, pero no xD. Es decir, se cambia la estructura binaria, pero la imagen no cambia para el ojo humano. Entonces, tanto el programa Steganos, como el S-tools hace esto para que tengas una idea básica...

Esquema de una imagen:

```
______________________________
| pixel pixel pixel pixel pixel píxel |
| pixel pixel pixel pixel pixel píxel |
| pixel pixel pixel pixel pixel píxel |
| pixel pixel pixel pixel pixel píxel |
| pixel pixel pixel pixel pixel píxel |
| pixel pixel pixel pixel pixel píxel |
| pixel pixel pixel pixel pixel píxel |
| pixel pixel pixel pixel pixel píxel |
“”””””””””””””””””””””””””””””””””
```

Como podemos observar, tenemos nuestra supuesta imagen de 800x600 pixel de 24 bits. Ahora si vemos a la imagen con sus valores binarios, encontramos esto...

```
______________________________
| 00101010-10100001-01100101 |
| 00100110-11010011-01011010 |
| 00101010-11000100-00110110 |
| 00101111-01101011-10100010 |
| 10010100-11010010-11010100 |
| 10011011-01101110-11010100 |
| 00101010-11010101-00100111 |
| 01010011-10001010-10100011 |
“”””””””””””””””””””””””””””””””
```

Bien, solo queda decir que cuando el programa procede a la utilización del LSB, se cambian los últimos bits de cada cadena de binarios.

Por ejemplo, escondemos el valor binario para la letra “a”(cuyo valor binario es) 10000011 y nos queda como en el grafico, ¿esta claro? Espero que si...

Esteganografia con sonido.

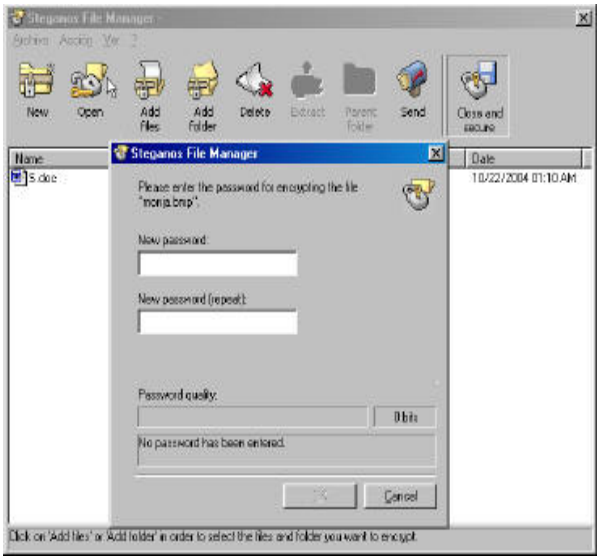
Si te quedo clarito el tema de la esteganografia en imágenes, pues alégrate por que con sonido la cosa no cambia mucho, ya que el método utilizado para ocultar archivos es el LSB. Ahora si a tener en cuenta esto, son poco los formatos en donde se pueden ocultar archivos por la misma razón de que necesitamos un archivo con su estructura original, sin que este comprimido, así que mejor te vas olvidando de pasar mp3 a tus amigos con mensajes adentro. El formato comúnmente mas utilizado es el WAV, ya que la mayoría del software puede trabajar con este tipo de archivos.

Utilizando el Steganos Security Suite 4.15

Steganos Security Suite 4.15, es una completa suite con varias funciones que nos permitirán proteger nuestros datos de manera confiable y segura. Funciona bajo windows y no necesita de grandes requisitos de sistema. Al ejecutar SSS 4.15, nos aparecerá una ventana en donde el programa muestra todas sus funciones, como encriptar mail, hacer discos protegidos, bloquear la pc, destruir datos que nos dejan cuando navegamos en Internet y la mas importante, la de ocultar archivos. El uso del programa es tremendamente

simple, ya que es muy intuitivo y gráfico. Personalmente es una estupenda herramienta en estos días en donde hay muchos lammers que solo quieren cagarte los datos que envías para después jactarse de que son lo mas... historia vieja que todos conocemos. Para cerrar este articulito (muy básico por cierto) les dejo un par de imágenes del Steganos que es el programa que uso y que hasta el día de hoy me ha funcionado a las mil maravillas.

ejecutando SSS 4.15



introduciendo archivos ocultos y asegurando con contraseña

Bien, creo que con esto ya esta por ahora, solo espero que el artículo les haya gustado. Como vieron, todo lo que se mostró aquí es MUY BÁSICO, así que a quejase menos y aprender mas. Hasta la próxima.

LINUX NOOB GUIDE.

Por 0x60 (0x60@hax0rs.biz)

Introducción.

Me decidí en escribir este texto porque he visto a demasiadas personas preguntar lo mismo y han sido flameados en muchos foros por respuestas tontas o contestan personas que no tienen ni la más mínima idea solo por verse kool, y a lo que me refiero es a la constante pregunta de "¿qué distro escoger?" o "¿por qué cambiar a linux?" para muchos puede resultar una pregunta muy sencilla pero para alguien empezando es bastante pesado recibir muchas respuestas y muchas flameadas.

¿Qué es Linux?

Básicamente linux es un sistema operativo, parte del revolucionario movimiento del open source. El open source significa que los programadores liberan el código de fuente para que sus clientes puedan verlo y modificarlo a sus necesidades y así el programa hará lo que ellos exactamente quieren. Las distribuciones son la recreación o modificación personal de alguien, como linux es open source cada quien lo puede modificar y crear su propia distro. Muchos de sus creadores no venden su distribución, se puede descargar gratis como es el caso de Debían, Fedora, Slackware, etc.

Pero que es Linux? es el kernel, aquel que sirve como modo de comunicación entre el disco duro, el cpu, etc. www.kernel.org

¿Qué Distro escoger?

Antes que nada quiero que pienses cuales son tus necesidades actuales, repasemos algunas de las distros disponibles actualmente:

Mandrake: es una distro muy flexible a lo que se refiere en configuración para el newbie, instalación grafica y magnifica forma de reconocer tu hardware. Su sistema de paquetes RPM hace muy fácil la instalación de estos mismos, lo único en contra sería los problemas financieros que esta pasando actualmente la compañía.

Red Hat: una de las distros mas reconocidas en el mundo y que es actualmente la dominante en servidores. Que quiere decir que encontrarás soporte en muchas comunidades, lo único en contra sería que red hat ahora solo se vende, pero podemos encontrar a su sucesor que es Fedora.

Debian: es un proyecto totalmente no comercial, en lo personal es mi favorita y actualmente es la que uso, es perfecta para servidores con funciones críticas, muchos usuarios prefieren usar las ramas de pruebas o inestable más actualizadas. Para instalar no es tan sencilla pero esto es compensado con el apt-getun maravilloso instalador de paquetes, su sistema de paquetes es .deb.

Knoppix: es un live-cd (no requiere de instalación), en menos de 5 minutos tendrás un sistema linux totalmente funcional. Tiene mas de 2 GB en software, gran detección de hardware, una vez instalado se convierte en Debian sid.

Slackware: es una distro muy antigua pero sigue existiendo, esta no es recomendable para principiantes porque no tiene interfaces graficas para usuarios, todo esta basado en modo de texto así como la configuración de hardware, se puede recomendar para administradores que conocen de linux ya que es bastante segura.

Así podría estar escribiendo durante horas, pero creo que te has dado cuenta en general y ya tendrás tu propia idea sobre linux y qué distro escoger.

Nota: estas no son todas las distros, son solo unas cuantas, busca en google.

Ahora donde puedes encontrar los ISOS, www.linuxiso.org - encontrarás las distros más populares. Bueno eso es todo, si has leído todo es porque en realidad tienes interés.

INTRODUCCIÓN A LA CRIPTOGRAFÍA MODERNA

Por beck (beck@badc0ded.org.ar)

La criptografía podría atreverme a clasificarla en tradicional y moderna por el hecho de que la tradicional (la cual no quiero mencionar mucho aquí) se refería a rotaciones cíclicas en alfabetos, polialfabetos y esas cosas que siempre explican en las charlas de criptografía pero ahora quiero enfatizar un poco lo que es la criptografía moderna la cual utiliza las capacidades de cómputo para generar nuevos algoritmos usando otras álgebras como álgebra de bool, y usando el poder de el álgebra moderna (campos anillos relaciones de equivalencia). Pero bueno, iremos por partes.

Llave.

Una duda usual generalmente es a que se refiere la gente cuando dice que lleva cifrado a 8n bits (digo 8n porque siempre es múltiplo de 8 ya que no creo que encuentres algoritmos que usen 1/2 byte (4 bits)) esto puede ser por 2 cosas, supongamos que me dijeron que es de 56 bits

1. El tamaño de la llave soporta máximo 56 bits (7 bytes)
o
2. El tamaño de encripción por bloque es de 56 bits (cifra de 56 bits en 56 bits)

Que significa el punto numero 1?

Que la llave solamente soporta 7 bytes (7*8), si ponen de llave aX49m20Kl9 <-- esta llave es de 80 bits así que solamente tomará aX49m20 los últimos 24 bits son inútiles.

Antes de empezar a mostrar ejemplos veamos cuando es una llave segura o insegura, tú aplica tu criterio.

Si tu llave es X9rK3f, eso parece una buena llave... pero pues muchos de ustedes tal vez dijeron que no lo es, y están en lo correcto, veamos con combinatoria porque no es segura.

Supongamos que cuando en un sistema metemos un password hay veces que este nos dice que la llave tiene elementos repetidos, si nosotros sabemos que no tiene elementos repetidos sabemos que podemos usar una ecuación sencilla las cuales me muestran las ordenaciones

$$\frac{n!}{(n-m)!}$$

n es el número de elementos que existen en el conjunto en nuestro caso es la tabla ASCII, y m es de cuantos en cuantos se van a tomar, en nuestro caso es X9rK3f y esto son 6 bytes entonces tenemos que n=256 y m=6 y podemos decir que "de n posibles elementos

tomamos de m en m" y lo que buscamos es cuantas posibles permutaciones u ordenaciones podemos crear con ese conjunto.

Esto es n factorial entre n - m factorial, donde factorial para los que aun no hayan visto esto en la escuela es por ejemplo: 6! = 720 = 6*5*4*3*2*1 esto algunas personas lo podrían haber visto usando notación de producto con PI mayúscula pero eso eso es diferente.

Es factorial por el hecho de que como las ordenaciones no se repiten en este entonces multiplica n*(n-1) para no incluir el elemento repetido y así sucesivamente después se divide entre la diferencia de el universo y la cardinalidad de el conjunto que tomamos de este y lo hacemos factorial. Por lo tanto sería:

```
  256!
--------
(256-6)!
```

Esto con algebrita de baldor sabemos que es 256*254*253*252*251 el cual es 265343617566720 este es el número de posibilidades que hay de encontrar un password de 6 bytes en un conjunto de 256 elementos, si el password SI repite bytes es más sencillo:

```
n^m  (n elevado a la m)

256^6 = 281474976710656
```

Que es lo mismo que 2^48 ya que si tomamos un universo de 2 (binario 1 y 0) y transformamos la cardinalidad del password en bits tenemos que son 48 bits por lo tanto es equivalente y es la forma de ver todos los subconjuntos que existen A = { x,y,z }. El número de subconjuntos que se pueden formar es 2^3 = 8 y es 2 por el hecho de que solo puede ser de 2 formas , o SI esta, o NO esta.

Si se fijan esto nos da un poco de mas posibilidades ya que incluye que se repitan bytes, esto puede ser abcdXX o XabXcd o XXabcd etc. Donde X es el elemento repetido y pertenece a A el cual es la tabla ASCII. Pero si ustedes saben que el password solo tiene letras mayúsculas y minúsculas y/o números esto se puede reducir.

26+26+10 serían las letras minúsculas + letras mayúsculas + números enteros de el 0 al 9 esto nos da un universo de 62 y tenemos n=62 y m=6

Sin bytes repetidos sería: 62!/56! el cual sería: 62*61*59*58*57 = 737694228
Con bytes repetidos sería: 62 a la 6 el cual sería: 56800235584

Como podemos ver estos números son pequeños y 100% computables en poco tiempo simplemente teniendo una computadora Pentium 4 a mas de 1.2 gHZ obtendremos estas combinaciones en menos de 8 horas y podremos probar TODAS localmente en menos de 11 horas así que recuerden que una llave difusa es igual de insegura que beck123 si estas miden poco, entre una llave de 6 bytes y una de 7 bytes hay una gran diferencia ya que

estamos hablando de exponenentes y pues tenemos que checar todos los elementos que se formarían con otro byte este sería el doble más si se repiten.

Veamos un poco de como se organiza la criptografía moderna y que es eso de los bloques. Veamos un poco de lógica de conjuntos.

A Intersección B indica los elementos que están en A y en B al mismo tiempo y en cómputo la operación Y es denotada por AND (&) esto también se puede ver como conjuntos. Por ejemplo: si tengo el conjunto de bits A = 101011 y B = 111110

A & B sería la intersección de bits entre A y B

```
101011
111110

101010
```

1 y 1 es 1 ya que si tienen en común al 1, 1 y 0 es 0 ya que no tienen en común nada. A UNION B son los elementos que están en A o B y en cómputo la operación O es denotada por OR (|)

A | B sería la unión de los bits entre A o B y este por supuesto tiene que ser un conjunto más grande

```
101011
111110

111111
```

1 y 0 es 1 porque hay una unión de bits entonces digamos que este "complementa" con 1 (lo puse entre comillas porque complementa no esta del todo bien dicho) 1 y 1 es 1 tendría que ser algo diferente mas grande que 1 pero estamos en base 2 por lo tanto ya esta hecha la unión. 0 y 0 es 0 ya que no existe y no se hace la unión y como vemos 111111 es un número mas grande que la intersección que fue 101010

Existe otra operación importante que es el O exclusivo o XOR, esta tiene una propiedad importante la cual. Esta vendría siendo como "la negación de la intersección" o complemento de la intersección", por ejemplo A XOR B

```
101011
111110
```

010101 si se fijan se parece a A & B pero solo que estos están invertidos pero cual es la propiedad importante?, llamémosle C a A XOR B

```
C XOR A = B
B XOR C = A
```

Tenemos que:

```
010101 XOR 101011 = 111110

010101
101011
111110
```

Esto nos lleva a algo interesante, que tal si uso esta propiedad de la forma

```
PASSWORD XOR TEXTO = TEXTOCIFRADO
TEXTOCIFRADO XOR PASSWORD = TEXTO
```

El proceso de cifrado y descifrado es el mismo. Si no me creen compilen este programa y hagan la prueba es un código en C donde "^" denota la operación XOR http://www.badc0ded.org.ar/files/C/xor.c

La parte mas importante es esta:

```
  while (read(fileno(archivo),&byte,sizeof(char)) > 0)
      {
         byte ^= contraseña[i % tamaño];
         write(fileno(cifrado), &byte,sizeof(char));
         i++;
      }
```

Pero que pasa si mi contraseña es más pequeña que mi archivo (como es generalmente)? Si se fijan en contraseña[i % tamaño] ando delimitando el índice al tamaño de I usando a contraseña como si fuera un anillo conmutativo, por ejemplo en Zn donde n = tamaño por lo tanto si tamaño = 3 entonces 6 = 0 ya que 6 = 0 modulo 3 (6/3 = 2 y sobran 0) entonces si i=839 (si se esta leyendo el byte 839 de un archivo) y mi contraseña es de 3 bytes entonces 839 = 2 modulo 3 ya que 839/3=279 y el residuo es 2 por lo tanto i = x mod t y x siempre es menor a t.

Por lo tanto si tengo ABCDXYZW y tengo de contraseña PASS entonces

```
ABCDXYZW
PASSPASS
```

Así le corresponderían las letras.

Desventajas.

1 XOR 0 es 1. Por lo tanto PASSWORD XOR 0 = PASSWORD entonces si ustedes usan ese programa para cifrar un binario con ceros si lo abren con su notepad o en Vi verán su contraseña en algunas partes de un archivo esto lo pueden hacer con un .exe en Windows por ejemplo o con cualquier ejecutable en linux o unix (/bin/ls, /bin/cp etc.).

Como vimos aquí las operaciones de cifrado y descifrado son las mismas:

```
./xor archivo.txt archivo.txt.enc password
./xor archivo.txt.enc archivo_nuevo.txt password
```

En resumen en esta parte de lógica de bool AND te puede servir para delimitar un espacio ya que siempre te dará la intersección por lo tanto x & y siempre te dará que x <= y.

OR para evadir ceros en procesos estocásticos (probabilístico) y sacar números más grandes en un espacio de bits igual.

NOT para sacar la negación de algo.

XOR pues ya vimos también se puede ver como para "guardar" dos valores en una variable, por ejemplo digamos que queremos intercambiar los valores de a y b (a=b, b=a), para eso muchos pensarían que necesitan una variable temporal para hacer:

```
t = a
a = b
b = t
```

Pero con xor:

```
a = a^b
b = a^b
a = a^b
```

Con numeritos: a=2 b=3

```
a = 2^3   (esto es 1)
b = 1^3   (esto es 2)
a = 1^2   (esto es 3)
```

Entonces tenemos que a=3 y b=2 y ya nos evitamos un espacio de memoria inútil usando lógica sencilla.

¿Cómo tienen que ser las funciones de cifrado?

Veámoslo un poco desde un enfoque matemático, y tenemos que tienen que ser funciones biyectivas para que podamos sacar el inverso, por ejemplo si tenemos:

f:R+ -> R+
f(x) = ln(((PI * x^2))) esto es logaritmo natural de x al cuadrado multiplicado por PI el inverso de f es :

invf(x) = raiz_cuadrada(e^x / PI) esto es raíz cuadrada de e a la x entre PI. Ahora que tal si hacemos f(invf(x)) si hemos ido a la escuela sabremos que eso nos regresa la identidad en este caso x, y en cifrado eso es lo que buscamos

```
f(5.2) = 4.442047
```

```
invf(5.2) = 7.5961
f(invf(5.2)) = 5.2
```

Ya que f(7.5961) = 5.2

Si lo quieres en C sería:

```
float f(float x)
{
return log((x*x)*M_PI);
}

float invf(float x)
{
return sqrt(exp(x)/M_PI);
}
```

Estas serían las 2 funciones, ya vimos que son biyectivas y tienen inversa obviamente biyectivas en el dominio y codominio definido anteriormente (reales positivos a reales positivos) pero bueno eso fue en matemáticas, ahora esto lo tenemos que usar de f:A -> A donde A es el conjunto de los números 0 al 255 (tabla ASCII), en este caso 8 bits (2^8), 0xff, etc.

En arreglos computacionales es sencillo hacer funciones inversas usando teoría básica de conjuntos, por ejemplo:

```
unsigned char A = { 4, 3, 0, 2, 6, 1, 5 };
```

A es un conjunto pero también es una función Inyectiva, suprayectiva y esto implica que es biyectiva. Saquemos el inverso de A:

```
unsigned char invA = { 2, 5, 3, 1, 0, 6, 4 };
```

A[invA[x]] = x tal que x pertenece Im { 1,2,3, ... , m } done m es la cardinalidad o tamaño del conjunto A en este caso x tiene que estar entre el 0 al 6, hagan la prueba.

El elemento invA[2] de A, A[invA[2]] sería A[3] y A[3] = 2 como vieron A[invA[2]] = 2;

Esto es la teoría muy básica ya que los métodos actuales usan una mayor complejidad (mucho mayor) y usan la llave para generar tablas, o localizar espacios que contienen información hacia otra parte del algoritmo.

Por último, un algoritmo de encripción por bloques, aunque suene redundante es cuando usa bloques de datos para cifrar por ejemplo si voy a usar bloques de 32 bits (4 bytes) entonces tomare de 4 bytes en 4 bytes para cifrar el pedazo de dato por ejemplo:

ABCDAAEE

Tomo los primeros 4 bytes y los represento numéricamente, por ejemplo un número de 32 bits (4 bytes) sería 0x01020304 ya que contiene { 0x01, 0x02, 0x03, 0x04} en el ejemplo la A representa en hexadecimal 0x41, la B 0x42 etc. Entonces 0x41424344 sería mi primer bloque y mi password supón que es 0x33474a32 entonces ya solo tiene que hacer 0x41424344 operación 0x33474a32 y ya con eso solo aplicas una operación una vez y cifras 32 bits de golpe después tomas AAEE el cual sería 0x41414545

Nota: ABCD es 0x41424344 si tu byte order del procesador es big endian, si es little endian como en Intel es 0x44434241

```
sh-2.05b$ ./bloque
error:
./bloque TEXT PASS
Solo usa 32 bits en TEXT y PASS ya que solo tomo bloques de 32 bits (4
bytes)
sh-2.05b$ ./bloque ABCD PASS
0x41424344 XOR 0x50415353 = 0x11031017
sh-2.05b$
```

El código es: http://www.badc0ded.org.ar/files/C/bloque.c

La parte importante en C es la siguiente:

```
  unsigned int *plano,*pass;
  plano = (int *) argv[1]; // transformo el char * de argv[1] a puntero
entero
  pass = (int *) argv[2];  // lo mismo con el password
  printf ("0x%08x XOR 0x%08x = 0x%08x\n", *plano, *pass, *plano ^
*pass); // tomo la dirección a la que apunta el puntero int con "*" y
hago las operaciones
```

Archivos que he hecho de interés (todos son libres)

http://www.badc0ded.org.ar/files/C/ISprime.c
http://www.badc0ded.org.ar/index.php?gadget=filebrowser&action=display&path=C/lea-1.0.3/crypt-essentials
http://www.badc0ded.org.ar/files/C/cook/cook.c.txt
http://www.badc0ded.org.ar/files/C/xhsc/cook.c
http://www.badc0ded.org.ar/index.php?gadget=filebrowser&action=display&path=C/xhsc
http://www.badc0ded.org.ar/files/C/vigenere.c
http://www.badc0ded.org.ar/index.php?gadget=filebrowser&action=display&path=C/hslc

Hay varias cosas más en mi página de matrices y números primos may chequenle. Espero les haya gustado, faltaron muchas cosas pero creo que de todos modos ya me extendí demasiado en mi página tengo mas información de algoritmos que he hecho y métodos con números primos y procesos estocásticos, la página es

http://www.badc0ded.org.ar (soy mexicano pero .ar es gratuito hehe). Comentarios, críticas y correcciones.

DESPEDIDA

Gracias a todos aquellos que se dieron tiempo para redactar su artículo y enviárnoslo, me gustaría recordarles que si se apegaran a las normas que se dictaron para la recepción de sus documentos pues facilitarían la edición de la ezine.

Recibí un documento que mas bien parecía un anuncio publicitario, para todo decía que lo contactaras, que él era bueno en equis o en ye, y la verdad la ezine todavía no llega a esos niveles, si en verdad quieres aportar algo pues bienvenido, no esperes nada a cambio, los que aquí colaboramos no recibimos nada a cambio, el fin es difundir la cultura informática sin un fin de lucro.

Como se habrán dado cuenta en esta ezine solo parece un artículo de un miembro, así que como se dice por ahí, esta ezine es suya, ustedes la hacen con sus aportaciones, sigan enviando sus documentos apegándose a las normas de redacción.