

Castillo Medina Aarón Martín

Documentación de la parte de autenticación

Introducción

Para la autenticación se ha elegido el lenguaje de programación C para la elaboración del CGI, la idea del CGI es que el formulario del login envíe los datos a éste y posteriormente éste decida (con ayuda de la base de datos) si el usuario tiene o no permitido entrar al sistema y de ser así bajo qué modo (usuario o administrador).

Requerimientos

Primero que nada es importante destacar que se requiere de la biblioteca libpq-dev, para ello se instala con el siguiente comando:

```
aptitude install libpq-dev
```

Ó

```
apt-get install libpq-dev
```

También es importante que en el archivo `/etc/apache2/sites-enabled/000-default` se le introduzca una nueva directiva.

```

dom 30 de mar, 04:40
user@debian: /etc/apache2

Archivo Editar Ver Buscar Terminal Pestañas Ayuda

user@debian: /var/www x user@debian: /etc/apache2 x user@debian: ~/Documentos x
<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Dav On
        Options Indexes +ExecCGI FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory \"/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn

    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
...
"000-default" 32L, 710C

1,1 Todo

```

Es importante que en ese mismo lugar se le coloque la directiva +ExecCGI, esto nos garantiza que se podrán ejecutar archivos .gi en el directorio /var/www.

Nota: adicionalmente me pidió en el servidor frontendmod02 el ServerName, entonces debajo de donde dice VirtualHost (hasta arriba) se añade la instrucción ServerName frontendmod02

(De acuerdo a las especificaciones se recomendaba no colocarlo aquí pero hubo muchos problemas con el script y con el alias cgi-bin ya que en muchas ocasiones tomaba la ruta completa.

Procedimiento

Con base a las credenciales del login.php, éstas se le envían al CGI para que proceda a analizarlas y a validarlas.

De acuerdo a lo que leí el CGI en el caso de C es un simple programa de C que ha sido compilado con extensión CGI (por la bandera -o), por lo que para tener este CGI funcionando únicamente se le compila y el código objeto es el que se coloca en el action del form.

Ahora lo que hace el CGI es ir leyendo las entradas y separarlas en tokens (con ayuda del método strtok)

Se verifican los tokens en la base de datos, si los datos resultan ser correctos se mandan a escribir al archivo tokens1 (en caso de ser un administrador) y tokens2(en caso de ser un usuario) y además se envían por la URL, si los datos resultan incorrectos o son un intento de hacer algún tipo de sentencia maliciosa simplemente se sanitizan y el resultado de éstos se envía por medio de la URL SIN enviarse al archivo, así se garantiza que, al momento de realizar la búsqueda, como las credenciales no estarán en los archivos, entonces no será válida.

Más preciso, antes de enviar las credenciales autenticadas por URL, a cada token se le aplica una función hash aplicada entre 10 y 2010 veces (esto es para incrementar la seguridad del token) y después se envían.

Cabe mencionar que el número de veces de la iteración de la función hash es un número aleatorio, lo cual incrementa más (incluso para el usuario), las probabilidades de “adivinar” la cadena correspondiente al hash.

Para evitar tener que estar acarreado también un token del tipo (usuario o admin) se optó por guardar los tokens en archivos diferentes de acuerdo al tipo de usuario (archivo tokens1 es para los admins mientras que el archivo tokens2 es para los usuarios). De eso se habló un poco hace algunas líneas.

Aquí cabe mencionar que si bien esta medida resulta ser un poco descuidada en realidad no hay tanto problema ya que para este método se tiene como refuerzo el envío de las credenciales a través de la URL.

Posteriormente un archivo de confirmación php tomará las credenciales enviadas por URL y tomará las credenciales de los archivos tokens, si existen en ambos lado entonces se procede a reenviar al usuario a la ventana que dejó en su sesión anterior (se recuerda que el hecho de ser usuario o administrador depende sobremanera del archivo en que se guardó).

Y así termina la ejecución de la autenticación vía CGI.

Ejecución del programa

La forma de ejecutar el programa es el siguiente:

Se coloca el archivo en el directorio `/var/www`, si bien se aconsejó al equipo de no hacerlo no quedó de otra ya que al guardarlo en el `/cgi-bin/` marcaba muchos errores, en especial errores de redireccionamiento, por ello se optó por guardarla ahí.

Se debe ejecutar el siguiente comando:

```
gcc -lssl -lpq -o validar.cgi validar.c datos.c
```

Y posteriormente el siguiente comando:

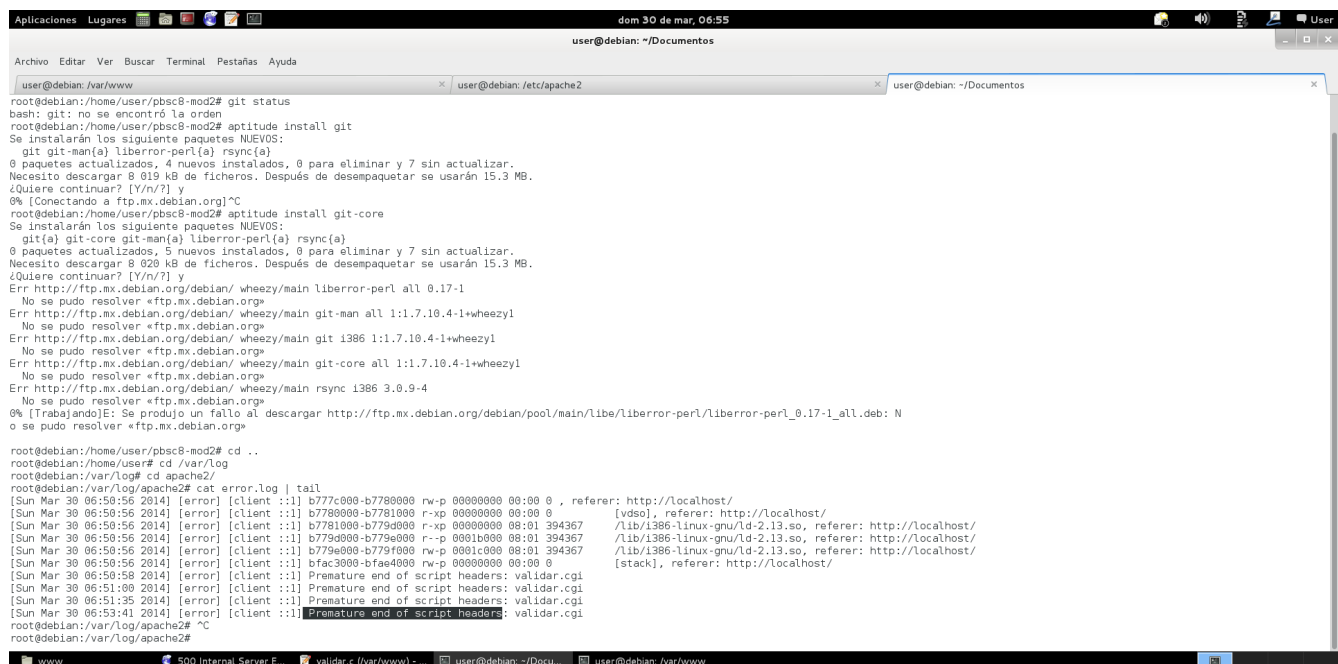
```
cp validar.cgi /usr/lib/cgi-bin/validar.cgi
```

Este último es sólo para garantizar que el programa correrá adecuadamente aún después de haberlo ocolocado en `/var/www`.

Observaciones

Algunas observaciones a realizar son:

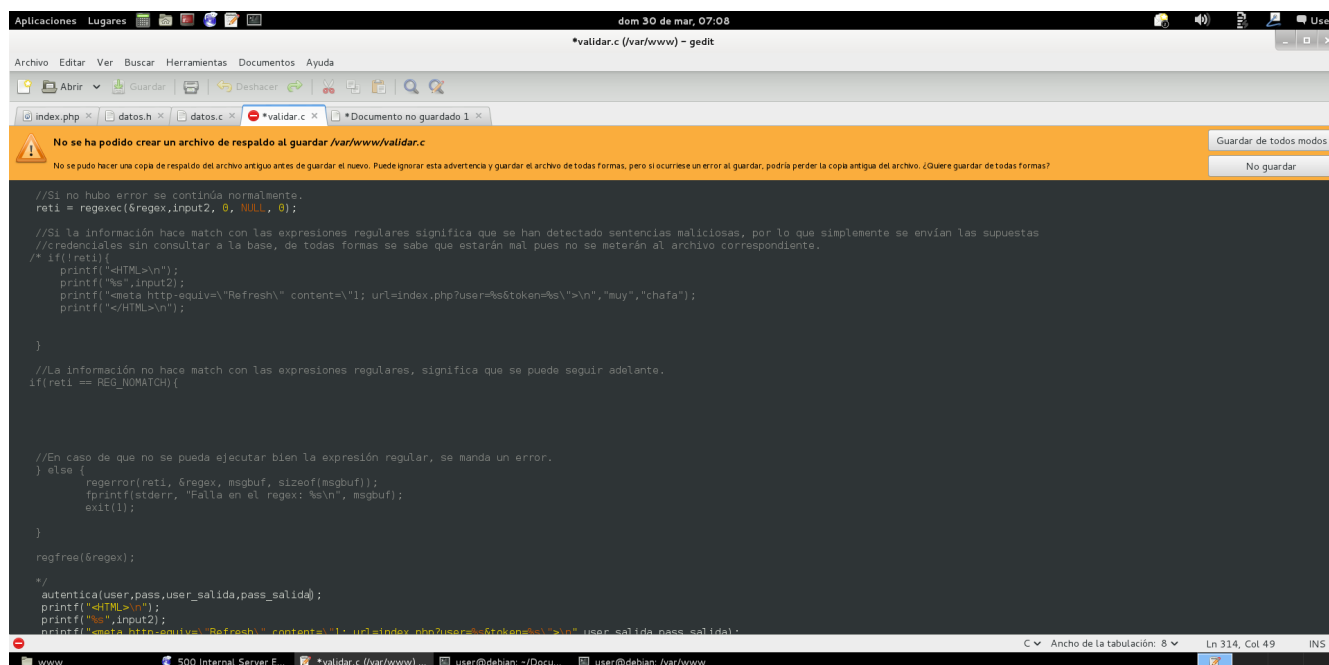
He tenido muchos problemas para realizar la autenticación, he revisado el `error.log` y lo que encontré fue lo siguiente:



```
user@debian: ~/Documentos
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
user@debian: /var/www
root@debian:/home/user/pbsc8-mod2# git status
bash: git: no se encontró la orden
root@debian:/home/user/pbsc8-mod2# aptitude install git
Se instalarán los siguiente paquetes NUEVOS:
git git-man(a) liberror-perl(a) rsync(a)
0 paquetes actualizados, 4 nuevos instalados, 0 para eliminar y 7 sin actualizar.
Necesito descargar 8 019 kB de ficheros. Después de desempaquetar se usarán 15.3 MB.
¿Quiere continuar? [Y/n/?] y
0% [Conectando a ftp.mx.debian.org]^C
root@debian:/home/user/pbsc8-mod2# aptitude install git-core
Se instalarán los siguiente paquetes NUEVOS:
git(a) git-core git-man(a) liberror-perl(a) rsync(a)
0 paquetes actualizados, 5 nuevos instalados, 0 para eliminar y 7 sin actualizar.
Necesito descargar 8 020 kB de ficheros. Después de desempaquetar se usarán 15.3 MB.
¿Quiere continuar? [Y/n/?] y
Err http://ftp.mx.debian.org/debian/ wheezy/main liberror-perl all 0.17-1
No se pudo resolver «ftp.mx.debian.org»
Err http://ftp.mx.debian.org/debian/ wheezy/main git-man all 1:1.7.10.4-1+wheezy1
No se pudo resolver «ftp.mx.debian.org»
Err http://ftp.mx.debian.org/debian/ wheezy/main git 1386 1:1.7.10.4-1+wheezy1
No se pudo resolver «ftp.mx.debian.org»
Err http://ftp.mx.debian.org/debian/ wheezy/main git-core all 1:1.7.10.4-1+wheezy1
No se pudo resolver «ftp.mx.debian.org»
Err http://ftp.mx.debian.org/debian/ wheezy/main rsync 1386 3.0.9-4
No se pudo resolver «ftp.mx.debian.org»
0% [Trabajando]E: Se produjo un fallo al descargar http://ftp.mx.debian.org/debian/pool/main/libe/liberror-perl/liberror-perl_0.17-1_all.deb: N
o se pudo resolver «ftp.mx.debian.org»

root@debian:/home/user/pbsc8-mod2# cd ..
root@debian:/home/user# cd /var/log
root@debian:/var/log# cd apache2/
root@debian:/var/log/apache2# cat error.log | tail
[Sun Mar 30 06:50:56 2014] [error] [client ::1] b777c000-b7780000 rw-p 00000000 00:00 0 , referer: http://localhost/
[Sun Mar 30 06:50:56 2014] [error] [client ::1] b7780000-b7781000 r-xp 00000000 00:00 0 [vdso], referer: http://localhost/
[Sun Mar 30 06:50:56 2014] [error] [client ::1] b7781000-b7790000 r-xp 00000000 00:01 394367 /lib/i386-linux-gnu/ld-2.13.so, referer: http://localhost/
[Sun Mar 30 06:50:56 2014] [error] [client ::1] b7790000-b779f000 r-p 00010000 00:01 394367 /lib/i386-linux-gnu/ld-2.13.so, referer: http://localhost/
[Sun Mar 30 06:50:56 2014] [error] [client ::1] b779f000-b779f000 r-p 0001c000 00:01 394367 /lib/i386-linux-gnu/ld-2.13.so, referer: http://localhost/
[Sun Mar 30 06:50:56 2014] [error] [client ::1] bfac3000-bfac4000 rw-p 00000000 00:00 0 [stack], referer: http://localhost/
[Sun Mar 30 06:50:58 2014] [error] [client ::1] Premature end of script headers: validar.cgi
[Sun Mar 30 06:51:00 2014] [error] [client ::1] Premature end of script headers: validar.cgi
[Sun Mar 30 06:51:35 2014] [error] [client ::1] Premature end of script headers: validar.cgi
[Sun Mar 30 06:53:41 2014] [error] [client ::1] Premature end of script headers: validar.cgi
root@debian:/var/log/apache2# ^C
root@debian:/var/log/apache2#
```

Sin embargo eso no me dio mucha información, aunque definitivamente fue algo que realicé en el método de autenticación ya que antes de ésto me funcionaba a la maravilla. De hecho después de este error ya no me dejó siquiera guardarlo.



```
//Si no hubo error se continúa normalmente.
reti = regexexec(&regex,input2, 0, NULL, 0);

//Si la información hace match con las expresiones regulares significa que se han detectado sentencias maliciosas, por lo que simplemente se envían las supuestas
//credenciales sin consultar a la base, de todas formas se sabe que estarán mal pues no se meterán al archivo correspondiente.
/* if(reti){
    printf("<html>\n");
    printf("%s",input2);
    printf("<meta http-equiv='Refresh' content='1'; url=index.php?user=%s&token=%s'\n",\"muy\", \"chafa\");
    printf("</html>\n");
}

//La información no hace match con las expresiones regulares, significa que se puede seguir adelante.
if(reti == REG_NOMATCH){

//En caso de que no se pueda ejecutar bien la expresión regular, se manda un error.
} else {
    regerror(reti, &regex, msgbuf, sizeof(msgbuf));
    fprintf(stderr, "Falla en el regex: %s\n", msgbuf);
    exit(1);
}

regfree(&regex);
*/
autentica(user,pass,user_salida,pass_salida);
printf("<html>\n");
printf("%s",input2);
printf("<meta http-equiv='Refresh' content='1'; url=index.php?user=%s&token=%s'\n",user_salida,pass_salida);
```

Ya no supe qué hacer, la verdad le intenté de todo pero el tiempo se me agotó.

Por cierto, al hacer la configuración de añadir +ExecCGI en el servidor me está dando muchos peros, más que si lo instalara en una máquina mía.

Con respecto del buen funcionamiento del programa se implementó un módulo de sanitización de entradas que funciona decentemente.

También se implementó una política de acceder a la base de datos sólo cuando fuese necesario, es decir, si se descubría alguna señal de ingreso malicioso se pasaban credenciales falsas para garantizar que ese usuario nunca iniciaría sesión

Con respecto del guardado de tipo de usuarios en dos diferentes archivos lo tomamos así por dos razones, por falta de tiempo y por sencillez, esperando que haya sido una opción sensata.

Finalmente lo que más causó problema es ese error del script que termina inesperadamente, tristemente no se le pudo dar una solución adecuada por lo que sólo esperamos la penalización por ello no sea muy grande, si bien en términos de trabajo la autenticación resultaba lo más pequeño de todos, al hacerla en C resultó en un dolor de cabeza, en especial por los hash ya que para convertirse a char* se requería de un método especial.

Nota adicional: tuve algunos problemas con mi git y llegué a un punto en que ya no pude echarlo a andar y para evitar perder tiempo lo que hice fue pedirle a mis compañeros que subieran mi parte en sus cuentas, por ello se verá que soy el que menos subió cosas sin embargo los códigos en C tienen mi nombre y cada uno viene minuciosamente documentado.

Fuentes de consulta

<http://stackoverflow.com/questions/308695/c-string-concatenation>

<http://www.linuxjournal.com/content/accessing-postgresql-cc>

<http://stackoverflow.com/questions/18864112/getting-error-while-installing-postgresql-gem>

<http://www.webthing.com/tutorials/cgifaq.1.html#2>

http://httpd.apache.org/docs/2.2/mod/mod_mime.html#addhandler

<http://stackoverflow.com/questions/5499504/shared-c-constants-in-a-header>

<http://serverfault.com/questions/524477/options-execcgi-is-off-in-this-directory-var-www-index-py>

<http://stackoverflow.com/questions/4063314/c-cgi-how-to-get-form>

<http://stackoverflow.com/questions/19552360/what-means-standard-input-in-c-language>

<http://www.linuxquestions.org/questions/programming-9/writing-a-cgi-to-redirect-to-another-webpage-6703/>