D

Código del plugin "escaneo"

Módulo “escaneo.pm”

```
#/usr/bin/perl
#######          ****************************************************************
#######          ****************************************************************
#######          **********                Plugin escaneo.pm           *********
#######          **********  Creado por: Aldo Iván Girón Capistrán  *********
#######          **********  Última fecha de modificación: 15/10/2010 *********
#######          ****************************************************************
#######          ****************************************************************

#######                    Objetivo del plugin
####### Verificar si se encuentran anomalías en la red que presenten el comportamiento de una
computadora o computadoras infectadas por algún malware.

###### El plugin verifica comportamientos anómalos verificando si se ha realizado en la red:
######           * Escaneo de puertos
######           * Escaneo de Ip'S
######           * Envió de información hacia el exterior y que presente un comportamiento anormal
######           * Ataque de negación de servicios (DoS)

######  El plugin escaneo.pm consta de las siguientes subrutinas.
######      * run:            Encargada de ejecutar el plugin cada 5 minutos.
######      * init:           Encargada de cargar el plugin al momento de iniciar el software nfsen.
######      * epuertos:       Encargada de verificar si se realizó un escaneo de puertos
######      * eips:           Encargada de verificar si se realizó un escaneo de ip's
######      * exterior:       Encargado de verificar si hubo fuga información hacia el exterior dela
red con un comportamiento anormal
#####       * DoS:            Encargado de verificar si se ha efectuado un ataque denegación de
servicios (DoS) o denegación de servicios distribuido (DDoS).
######      * guarda:         En caso de encontrar un comportamiento anormal, esta subrutina se
encargará de guardar los flujos anormales detectados en el archivo “anomalías.txt”
######      * envia_correo:  Esta subrutina verificará si existe el archivo “anomalías.txt”. Si
existe este archivo la subrutina se encargará de notificar mediante él envió de un correo
electrónicomostrando el contenido de este archivo
######      * compara_ip:    Subrutina ocupada por “eips”. Esta subrutina se encarga de verificar si
se presenta un escaneo de IP’S en los registros analizados.
######      * compara:    Subrutina ocupada por “epuertos”Esta subrutina se encarga de verificar si
se presenta un escaneo de puertos en los registros analizados.
######      * compara_exterior: Subrutina ocupada por “exterior”Esta subrutina se encarga de
verificar si se presenta una fuga de información en los registros analizados.
######      * compara_dos:   Subrutina ocupada por “DoS”Esta subrutina se encarga de verificar si se
presenta un ataque de negación de servicios en los registros analizados.
######      * separa:     Subrutina encargada de separar la información recibida por el colector
nfdump de la siguiente forma:     (protocolo) , (ip ori : pto ori) , (ip dst : pto dst)
######      * agrupa:     Subrutina ocupada por “epuertos” y “DoS”. Esta subrutina se encarga de
separar  la información en un formato que permita la detección de un escaneo de puertos o ataque de
negación de  servicios. La subrutina “agrupa” separa la información de la siguiente forma:
######                    xxx.xxx.xxx.xxx :  xxxxx    ->  xxx.xxx.xxx.xxx :  xxxxx        xx
######                             (ip ori)    , (pto ori)  ,     (ip dst)     , (pto dst) , (trafico)
######      * agrupa_ip:  Subrutina ocupada por “eips” y “exterior”. Esta subrutina se encarga de
separar  la información en un formato que permita la detección de un escaneo de IP’S o detectar fuga
de información. La subrutina “agrupa_ip” separa la información de la siguiente forma:
######   xxx.xxx.xxx.xxx :  xxxxx     ->   xxx.xxx  .    xxx   .   xxx    :  xxxxx        xx
######    (ip ori)        , (pto ori)  ,   (ip dst1) , (ip dst2), (ip dst3), (pto dst)  (trafico)
######      * Cleanup:    Subrutina especial, utilizada por el software nfsen para permitir al plugin
limpiar datos cuando se finaliza la ejecución del software Nfsen

package escaneo;
### Declaración de modulos a utilizar.
use strict;
use Switch;
use NfSen;
use NfConf;
use Sys::Syslog;

Sys::Syslog::setlogsock('unix');
use Mysql;
use Notification;
### Declaración de variables locales dentro del plugin, que actuaran como variables globales para el
uso de las demás subrutinas
my ( $nfdump, $PROFILEDATADIR, $LOGFILE, $NOTIFY );
my (@registros,@guarde,@anomalias,@encontre);
my @registros =();

#############################################################################################
######      Subrrutina run:           Encargada de ejecutar el plugin cada 5 minutos.

sub run {
### Guardamos información recibida por subrutina init y notifícanos en syslog el que seestá
ejecutando el plugin.
my $profile = shift;
my $timeslot = shift;
syslog('debug',"Plugin escaneo run: Profile: $profile, Time: $timeslot");

### Inicializamos el arreglo encontre con -1: En este arreglo se guardará el número de la línea en el
cual se encontró alguna anomalía para evitar revisar información repetida
```

```
        @encontre =(-1);

### Leemos información proporcionada por una módulo de nfsen
        my %profileinfo      = NfProfile::ReadProfile('live');
        #my $netflow_sources ="$PROFILEDATADIR/live/Anexo";

### Accedemos a la ruta donde se instaló nfdump
        my $netflow_sources ="$PROFILEDATADIR/live/$profileinfo{'sourcelist'}";

### El arreglo “registros” contendrá toda la información que ha sido interpretada por el colector
nfdump, y ha sido guardada en un archivo nfcapd, este arreglo se modificará cada 5 minutos,
almacenando en él la nueva información capturada por nfdump y guardada en el archivo nfcapd
@registros = `$nfdump -M $netflow_sources -r nfcapd.$timeslot`;

### Guardamos la información contenida en el arreglo registros en un archivo llamada salida.txt
        if (open (LOG, "> /Listry-AIGC/salida.txt")){ ;
                print LOG "$nfdump -M $netflow_sources -r nfcapd.$timeslot\n\nDatos guardados.\n\n";
        print LOG @registros;
        close LOG ;
        } else {
        syslog('debug', "Escaneo: unable to open $LOGFILE") ;
        }
### Si en ejecuciones posterioresdel plugin se detectaron anomalías, se creará un archivo llamado
“anomalias.txt”. Al realizar un nuevo análisis en caso de tener este archivo se borrará con el
objetivo de no mostrar información repetida
        if (open (VERIFICA,"</Listry-AIGC/anomalias.txt")){
                close VERIFICA;
                system ("rm /Listry-AIGC/anomalias.txt");
        }
        if (open (VERIFICA,"</Listry-AIGC/anomalias_php.txt")){
close VERIFICA;
system ("rm /Listry-AIGC/anomalias_php.txt");
}
### Mandamos llamar a las subrutinas epuertos, eips, exterior y DoS, cada vez que alguna de estas
subrutinas termine volvemos a iniciar el arreglo encontre en -1
        &epuertos(@registros);
        @encontre =(-1);
        &eips(@registros);
        @encontre =(-1);
&exterior(@registros);
        @encontre =(-1);
&DoS(@registros);

### Al haber terminado todas las subrutinas su función especifica verificamos si existen anomalías
que se han guardado en el archivo anomalias.txt, en caso de encontrar este archivo llamamos a la
subrutina envia_correo
        if (open (VERIFICA,"</Listry-AIGC/anomalias.txt")){
                close VERIFICA;
                &envia_correo();
        }

} # Fin de la subrutina run

#############################################################################################################
###  Subrutina init:         Encargada de cargar el plugin al momento de iniciar el software nfsen.

sub Init {
syslog("info", "Escaneo: Init");
# Inicializamos variables que contendrán información necesaria parala ejecución de la subrutina run
$nfdump          = "$NfConf::PREFIX/nfdump";
    $PROFILEDATADIR =  "$NfConf::PROFILEDATADIR";
    $LOGFILE         = "$NfConf::VARDIR/tmp/escaneo.log" ;
    $NOTIFY          = 1 ;
return 1;
} # Fin subrrutina Init

#############################################################################################################
###  Subrutina epuertos:     Encargada de verificar si se realizó un escaneo de puertos

sub epuertos{
### Declaración variables locales ocupadas en esta subrutina.
        my ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico);
        my ($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$trafico_tmp);
        my ($var,$valor, $linea, $contador, $contador_anom,$cont_tmp,$anomalo);
        my @tmp;
        $contador=0;

### Obtenemos la información enviada por la subrutina run
        @registros=@_;

### Inicializamos arreglos vacíos, el arreglo “anomalias” contendrá la información encontrada del
análisis y que se ha considerado como anormal.El arreglo “guarde” contiene elementos separados por la
subrutina separa.
        @guarde=();
        @anomalias=();
        $linea=0;
```

```
## Obtenemos el número total de elementos almacenados en el arreglo “registros” y se guarda esta
referencia en la variable “valor”
        $valor=$#registros;
        #print "$#registros";

### Recorremos todos los datos contenidos en el arreglo “registros”, haciendo referencia a cada uno
de ellos en la variable “linea”
foreach $linea (@registros){
                #print "$linea\t $contador\n";

## Llamamos a la subrutina “separa”,  pasando como argumento la variable “linea”, además el arreglo
“guarde”. Esta subrutinadevuelveel arreglo “guarde” que contiene la información separada
                @guarde=&separa($linea,@guarde);
                #print "\nHola imprimo \t@guarde\n";
                #print "$linea";
        }   ###Fin ciclo foreach

### Inicializamos contadores requeridos
        $contador =0;
        $contador_anom=0;
        $valor=$#guarde;
        my ($cont_encontre, $cont_linea ,$banderin);

### Este ciclo while se encargara de verificar si existe algún número guardado en arreglo
“encontre”que corresponda con alguna línea del arreglo “registros”, en caso de encontrar algún número
igual no permitirá que se realice el análisis, debido a que esa línea ya ha sido revisada y se
encontró con una anomalía
### Este ciclo while recorrerá todos los elementos que se tengan en nuestro arreglo registros
        ## Inicio 1er while
        while($contador<=$valor){
                $cont_encontre=$#encontre;
                $cont_linea=0;
                $banderin=1;
                while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                $anomalo=$encontre[$cont_linea];
                #print"$cont_tmp == $anomalo \t";
                        if ($cont_tmp == $anomalo){
                                #print "\n No entre\n";
                                $banderin=0;
                        }
                $cont_linea ++;
                }   #### Fin ciclo lineas encontradas

### En caso de que el ciclo anterior haya encontrado alguna línea ya revisada y con comportamiento
anómalo no se entra al if, Si el registro no ha sido revisado se procederá a revisar esta línea con
las demáselementosdel arreglo.
                if ($banderin==1){

## Guardamos el valor temporal del arreglo “guarde” en la variable “linea”
                $linea=$guarde[$contador];
                #print "\n\n$contador\t$linea  \n";

## Mandamos llamar a la subrutina “agrupa”, que encargará de separar los elementos de la forma
indicada en el inicio del plugin
                ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico)=&agrupa($linea);
                #print "\n\n$contador\t$ip_ori:$port_ori --> $ip_dest:$port_dest\n";

## Asignamos el varal de la variable “contador” a la variable “cont_tmp”, este contador temporal se
encargara de buscar alguna anomalía desde este elemento en adelante
                $cont_tmp=$contador;

### En este arreglo se almacenaran todos los elementos temporales que hayan sido encontrados como
anómalos
                @tmp=();

### Cada que se encuentre un elemento anormal en este arreglo se incrementará la variable
“contador_anom”
                $contador_anom=0;

### En este ciclo iremos comparando los elementos que se encuentren por debajo del contador
“cont_tmp”
                ### Inicio 2 ciclowhile
                while($cont_tmp<=$valor){
                        $cont_tmp ++;
                        $cont_encontre=$#encontre;
                        $cont_linea=0;
                        $banderin=1;

#Esta parte del codigo ya se ha explicado.
                        while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                        $anomalo=$encontre[$cont_linea];
                        #print"$cont_tmp == $anomalo \t";
                                if ($cont_tmp == $anomalo){
                                        #print "\n No entre\n";
                                        $banderin=0;
                                }
                        $cont_linea ++;
                        }   #### Fin ciclo lineas encontradas
```

```
                        if ($banderin==1){
                        #print "$contador = $cont_tmp\t";

## Guardamos el elemento temporal a comparar
$linea=$guarde[$cont_tmp];
                        #print "\t$cont_tmp\t$linea";

## Llamamos a la subrutina agrupa. Esta subrutina se encargará de agrupar la línea que contiene el
elemento temporal. Esta subrutina nos devuelve variables que contiene la información
agrupada($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$trafico_tmp)=&agrupa($linea);
                        #print"\t$cont_tmp =-= $ip_ori_tmp:$port_ori_tmp -->
                        $ip_dest_tmp:$port_dest_tmp";

## Mandamos llamar a la subrutina “compara” que se encargará de verificar si hay un incremento en el
puerto destino del elemento actual con el elemento temporal comparado, en caso de encontrar esta
anomalía, se guardara esta línea en el arreglo “tmp”, además de incrementar el contador_anom.Esta
subrutina devuelve la variable “port_dst” para no perder referencia de este elemento.
                        ($port_dest,$contador_anom,@tmp)=&compara($ip_ori, $ip_dest, $port_dest,
$ip_ori_tmp, $ip_dest_tmp, $port_dest_tmp,$linea,$cont_tmp,$contador_anom,$contador,@tmp);
                #$cont_tmp ++;
                        }## Fin if coincide
                }   #### Fin 2 ciclo while iteración con valores temporales
                #print "\n\n";

### En caso de que la variable“contador_anom” sea mayor o igual que diez, nos indica que se ha
detectado un escaneo de puertos.Al detectarse esta anomalía se guardara el contenido del arreglo“tmp”
en el arreglo “anomalias”
                if ($contador_anom >= 10){
                        push (@anomalias,@tmp);
                }
                }   #Fin if si coincide línea con alguna anomalía
        $contador ++;
        }   ###Fin ciclo 1er while recorre todos los datos encontrados

### Si existe el arreglo “anomalias” se ejecutará la subrutina guarda,pasando como argumento a esta
subrutina la bandera “impirmo”, esta bandera indicaráque anomalía se ha detectado. Además se pasa
como argumento el arreglo “anomalias”.
        if (@anomalias){
                my $imprimo=0;
                &guarda($imprimo,@anomalias);
        }
}   ### Fin subrutina e_puertos

#############################################################################################
###     Subrutina eips:      Encargada de verificar si se realizó un escaneo de IP'S
sub eips{
### Declaración variables locales ocupadas en esta subrutina.
        my ($ip_ori,$port_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_dest,$trafico);
my ($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,$port_dest_tmp,$trafico_tmp);
        my ($var,$valor, $linea, $contador, $contador_anom,$cont_tmp,$anomalo);
        my @tmp;
        $contador=0;

### Obtenemos la información enviada por la subrutina run
        @registros=@_;

### Inicializamos arreglos vacíos, el arreglo “anomalias” contendrá la información que se ha
detectado anormal del análisis realizado, el arreglo guarde contiene elementos separados por la
subrutina separa.
        @guarde=();
        #@anomalias=();
        $linea=0;

## Obtenemos el número total de elementos almacenados en la arreglo “registros”
        $valor=$#registros;
        #print "$#registros";

### Recorremos todos los datos contenidos en el arreglo “registros”.haciendo referencia a cada uno de
ellos en la variable “linea”
foreach $linea (@registros){
                #$contador ++;

## Llamamos a la subrutina “separa”,  pasando como argumento la variable “linea”, además el arreglo
“guarde”. Esta subrutinadevuelveel arreglo “guarde” que contiene la información separada
                @guarde=&separa($linea,@guarde);
                #print "\nHola imprimo \t@guarde\n";
                #print "$linea";
        }   ###Fin ciclo foreach

### Inicializamos contadores requeridos
        my $i;
        my ($cont_encontre, $cont_linea ,$banderin);

## Este ciclo hará que todo el procedimiento se repita dos veces por este motivo:
##      * En la 1er iteración buscara escaneo de IP's con modificación en el 4 octeto
##      * En la 2 iteración buscara escaneo de IP's con modificación en el 3er octeto
        ## Inicio ciclo for recorre 2 veces todo el procedimiento
```

```
        for ($i=1; $i<=2; $i++){
                @anomalias=();

### Inicializamos contadores requeridos
                $contador =0;
                $contador_anom=0;
                $valor=$#guarde;
                #print "$i\n";
### Este ciclo while recorrerá todos los elementos que se tengan en nuestro arreglo registros
                ### Incio primer ciclo while
while($contador<=$valor){

                        $cont_encontre=$#encontre;
                        $cont_linea=0;
                        $banderin=1;
### Este ciclo while se encargara de verificar si existe algún número guardado en arreglo
“encontre”que corresponda con alguna línea del arreglo “registros”, en caso de encontrar algún número
igual no permitirá que se realice el análisis, debido a que esa línea ya ha sido revisada y se
encontró con una anomalía
### Este ciclo while recorrerá todos los elementos que se tengan en nuestro arreglo registros
                        while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                                $anomalo=$encontre[$cont_linea];
                                #print"$cont_tmp == $anomalo \t";
                                if ($cont_tmp == $anomalo){
                                        #print "\n No entre\n";
                                        $banderin=0;
                                }
                        $cont_linea ++;
                        }   #### fin ciclo lineas encontradas
### En caso de que el ciclo anterior haya encontrado alguna línea ya revisada y con comportamiento
anómalo no se entra al if, Si el registro no ha sido revisado se procederá a revisar esta línea con
las demáselementosdel arreglo.
                        if ($banderin==1){

## Guardamos el valor temporal del arreglo “guarde” en la variable “linea”
                                $linea=$guarde[$contador];

## Mandamos llamar a la subrutina “agrupa_ip”, que encargará de separar los elementos de la forma
indicada en el inicio del plugin
        ($ip_ori,$port_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_dest,$trafico)=&agrupa_ip($linea);
                                #print "\n$contador\t$ip_ori:$port_ori -->
                                $ip_dest1.$ip_dest2.$ip_dest3:$port_dest\n";

## Asignamos el varal de la variable “contador” a la variable “cont_tmp”, este contador temporal se
encargara de buscar alguna anomalía desde este elemento en adelante
                                $cont_tmp=$contador;

### En este arreglo se almacenaran todos los elementos temporales que hayan sido encontrados como
anómalos
                                @tmp=();

### Cada que se encuentre un elemento anormal en este arreglo se incrementará la variable
“contador_anom”
                                $contador_anom=0;

### En este ciclo iremos comparando los elementos que se encuentren por debajo del contador
“cont_tmp”
                        ### Inicio 2 while
                                while($cont_tmp<=$valor){
                                        $cont_tmp ++;
                                        $cont_encontre=$#encontre;
                                        $cont_linea=0;
                                        $banderin=1;

#Esta parte del codigo ya se ha explicado.
                                        while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                                                $anomalo=$encontre[$cont_linea];
                                                #print"$cont_tmp == $anomalo \t";

                                                if ($cont_tmp == $anomalo){
                                                        #print "\n No entre\n";
                                                        $banderin=0;
                                                }
                                        $cont_linea ++;
                                        }   #### fin ciclo lineas encontradas

                                if ($banderin==1){

## Guardamos el elemento temporal a comparar
                $linea=$guarde[$cont_tmp];

## Llamamos a la subrutina agrupa_ip. Esta subrutina se encargará de agrupar la línea que contiene el
elemento temporal. Esta subrutina nos devuelve variables que contiene la información agrupada
($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,$port_dest_tmp,$trafico_tmp)=&ag
rupa_ip($linea);

#print"\t$cont_tmp =-= $ip_ori_tmp:$port_ori_tmp --> $ip_dest_tmp1 a.$ip_dest_tmp2 b.$ip_dest_tmp3
c:$port_dest_tmp";
```

```
## Mandamos llamar a la subrutina “compara_ip” que se encargará de verificar si hay un incremento en
el tercer o cuarto octeto de la IP destino (dependiendo de la iteración del ciclo for) del elemento
actual con el elemento temporal comparado, en caso de encontrar esta anomalía, se guardara esta línea
en el arreglo “tmp”, además de incrementar el contador_anom.Esta subrutina devuelve la variable
“ip_dest2” e “ip_dest3” para no perder referencia de estos elementos.

($ip_dest2,$ip_dest3,$contador_anom,@tmp)=&compara_ip($i,$ip_ori, $ip_dest1,$ip_dest2, $ip_dest3,
$port_dest, $ip_ori_tmp, $ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,
$port_dest_tmp,$linea,$cont_tmp,$contador_anom,$contador,@tmp);
                                } ## Fin if
                }   #### Fin 2 ciclo while iteración con valores temporales

### En caso de que la variable“contador_anom” sea mayor o igual que diez, nos indica que se ha
detectado un escaneo de IP’S.Al detectarse esta anomalía se guardara el contenido del arreglo“tmp” en
el arreglo “anomalias”
                                if ($contador_anom >= 10){
                                        push (@anomalias,@tmp);
                                }
                        }   #fin if si conincide con linea con alguna anomalia
                        $contador ++;

                }   ###Fin 1er ciclo while recorre todos losdatos encontrados

### Si existe el arreglo “anomalias” se ejecutará la subrutina guarda,pasando como argumento a esta
subrutina la bandera “impirmo”, esta bandera indicaráque anomalía se ha detectado. Además se pasa
como argumento el arreglo “anomalias”.
        if (@anomalias){
my $imprimo=1;
&guarda($imprimo,@anomalias);
                }
        }#fin ciclo for  recorre 2 veces
} #######fin sub escaneo ips

#############################################################################################
## Subrutina exterior:       Encargado de verificar si se ha enviado información hacia el exterior
de la red con un comportamiento anormal.
### No se comenta esta subrutina debido a que tiene una estructura similar eips, solamente cambia en
llamar  a la subrutina“compara_exterior”.

sub exterior{
        my ($ip_ori,$port_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_dest,$trafico);
        my
($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,$port_dest_tmp,$trafico_tmp);
        my ($var,$valor, $linea, $contador, $contador_anom,$cont_tmp,$anomalo);
        my @tmp;
        $contador=0;
        @registros=@_;
        @guarde=();
        @anomalias=();
        $linea=0;
        $valor=$#registros;
        #print "$#registros";

foreach $linea (@registros){
                #print "$linea\t $contador\n";
                @guarde=&separa($linea,@guarde);
                #print "\nHola imprimo \t@guarde\n";
                #print "$linea";
        }###Fin ciclo for

        $contador =0;
        $contador_anom=0;
        $valor=$#guarde;
        my ($cont_encontre, $cont_linea ,$banderin);

        ### Inicio 1er ciclo while recorre todos los datos encontrados
        while($contador<=$valor){
                $cont_encontre=$#encontre;
                $cont_linea=0;
                $banderin=1;

                while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                        $anomalo=$encontre[$cont_linea];
                        #print"$cont_tmp == $anomalo \t";

                        if ($cont_tmp == $anomalo){
                                #print "\n No entre\n";
                                $banderin=0;
                        }
                        $cont_linea ++;
                }   #### Fin ciclo lineas encontradas

                ### Inicio if en caso de que el elemento actual no haya sido detectado como anomalo
                if ($banderin==1){
                                $linea=$guarde[$contador];
        ($ip_ori,$port_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_dest,$trafico)=&agrupa_ip($linea);
```

```
                                #print "\n$contador\t$ip_ori:$port_ori -->
$ip_dest1.$ip_dest2.$ip_dest3:$port_dest\n";
                                $cont_tmp=$contador;
                                @tmp=();
                                $contador_anom=0;

                                ## Inicio 2 ciclo while iteraccion con valores temporales
                                while($cont_tmp<=$valor){
                                        $cont_tmp ++;
                                        $cont_encontre=$#encontre;
                                        $cont_linea=0;
                                        $banderin=1;

                                        while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                                                $anomalo=$encontre[$cont_linea];
                                                #print"$cont_tmp == $anomalo \t";

                                                if ($cont_tmp == $anomalo){
                                                        #print "\n No entre\n";
                                                        $banderin=0;
                                                }
                                                $cont_linea ++;
                                        }   #### Fin ciclo lineas encontradas

                                if ($banderin==1){
                $linea=$guarde[$cont_tmp];

                                        #print "\t$cont_tmp\t$linea";

        ($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,$port_dest_tmp,$trafico
_tmp)=&agrupa_ip($linea);

                                        #print"\t$cont_tmp =-= $ip_ori_tmp:$port_ori_tmp -->
$ip_dest_tmp1 a.$ip_dest_tmp2 b.$ip_dest_tmp3 c:$port_dest_tmp";

                                        ($port_ori,$contador_anom,@tmp)=&compara_exterior($ip_ori,
$port_ori, $ip_dest1,$ip_dest2, $ip_dest3, $ip_ori_tmp, $port_ori_tmp,
$ip_dest_tmp1,$linea,$cont_tmp,$contador_anom,$contador,$trafico,$trafico_tmp,@tmp);
                                } ## Fin if
                }#### fin 2 ciclo while iteraccion con valores temporales

                                if ($contador_anom >= 10){
                                        push (@anomalias,@tmp);
                                }
                        }    #Fin if si conincide con linaco con alguna anomalia
                        $contador ++;
                        }     ###Fin 1er ciclo while recorre todos losdatos encontrados

        if (@anomalias){
my $imprimo=2;
&guarda($imprimo,@anomalias);
}
}   ##### Fin subrrutina exterior.

#############################################################################################
## Subrrutina DoS:           Encargado de verificar si se ha efectuado un ataque denegación de
servicios (DoS) o denegación de servicios distribuido (DDoS).
### No se comenta esta subrutina debido a que tiene una estructura similar a la subrutina “epuertos”,
solamente cambia en llamar a la función “compara_dos”.

sub DoS{
        my ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico);
        my ($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$trafico_tmp);
        my ($var,$valor, $linea, $contador, $contador_anom,$cont_tmp,$anomalo);
        my @tmp;
        $contador=0;
        @registros=@_;
        @guarde=();
        @anomalias=();
        $linea=0;
        $valor=$#registros;
        #print "$#registros";
foreach $linea (@registros){
                #print "$linea\t $contador\n";
                @guarde=&separa($linea,@guarde);
                #print "\nHola imprimo \t@guarde\n";
                #print "$linea";
        }   ###Fin ciclo foreach
        $contador =0;
        $contador_anom=0;
        $valor=$#guarde;
        my ($cont_encontre, $cont_linea ,$banderin);

        ## Inicio 1er while
        while($contador<=$valor){
                $cont_encontre=$#encontre;
                $cont_linea=0;
                $banderin=1;
```

```
                while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                        $anomalo=$encontre[$cont_linea];
                        #print"$cont_tmp == $anomalo \t";
                        if ($cont_tmp == $anomalo){
                                #print "\n No entre\n";
                                $banderin=0;
                        }
                        $cont_linea ++;
                }   #### Fin ciclo lineas encontradas
                if ($banderin==1){
                $linea=$guarde[$contador];
                #print "\n\n$contador\t$linea  \n";

                ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico)=&agrupa($linea);
                #print "\n\n$contador\t$ip_ori:$port_ori --> $ip_dest:$port_dest\n";
                $cont_tmp=$contador;
                @tmp=();
                $contador_anom=0;
                ### Incio 2 while
                while($cont_tmp<=$valor){
                        $cont_tmp ++;
                        $cont_encontre=$#encontre;
                        $cont_linea=0;

                        $banderin=1;
                        while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                        $anomalo=$encontre[$cont_linea];
                        #print"$cont_tmp == $anomalo \t";
                                if ($cont_tmp == $anomalo){
                                        #print "\n No entre\n";
                                        $banderin=0;
                                }
                        $cont_linea ++;
                        }   #### fin ciclo lineas encontradas
                        if ($banderin==1){
$linea=$guarde[$cont_tmp];
                        #print "\t$cont_tmp\t$linea";
                ($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$trafico_tmp)=&agrupa($linea);
                        #print"\t$cont_tmp =-= $ip_ori_tmp:$port_ori_tmp -->
$ip_dest_tmp:$port_dest_tmp";
        ($port_dest,$contador_anom,@tmp)=&compara_dos($port_dest,$ip_dest,$trafico,$port_dest_tmp,$i
p_dest_tmp,$trafico_tmp,$linea,$cont_tmp,$contador_anom,$contador,@tmp);
                        }  ## Fin if
                }   #### Fin 2 ciclo while iteraccion con valores temporales
                 if ($contador_anom >= 50){
                 #if ($contador_anom >= 25){
                        push (@anomalias,@tmp);
                 }
                 }   #Fin if si conincide con linaco con alguna anomalia
        $contador ++;
        }   ###Fin ciclo 1er while recorre todos los datos encontrados
        if (@anomalias){
                 my $imprimo=3;
                 &guarda($imprimo,@anomalias);
        }
}       ### Fin subrrutina DoS

#############################################################################################
### Subrutina obten_fecha_id encargada de obtener fecha e ID para insertar en  BD Mysql

sub obten_fecha_id{
        my ($segundos, $minutos, $horas, $dia_mes, $mes, $anyo, $fecha_unix, $fecha);
## Obtenemos fecha y le asignamos el formato correcto
        $fecha_unix = time ();
        ($segundos, $minutos, $horas, $dia_mes, $mes, $anyo, undef, undef, undef) = localtime
($fecha_unix);
        $mes ++;
if (($dia_mes >= 0)&&($dia_mes < 10)){
                $dia_mes ='0'.$dia_mes;
        }
if (($mes >= 0)&&($mes < 10)){
                $mes ='0'.$mes;
        }
if (($segundos >= 0)&&($segundos < 10)){
                $segundos ='0'.$segundos;
        }
if (($minutos >= 0)&&($minutos < 10)){
                $minutos ='0'.$minutos;
        }
if (($horas >= 0)&&($horas < 10)){
                $horas ='0'.$horas;
        }
if ($anyo < 1900){
                $anyo += 1900
        }
## Guardamos en variable “fecha”: año, mes, día, hora y minutos
        $fecha="$anyo-$mes-$dia_mes $horas:$minutos:$segundos";
        #print "$fecha\n";
```

```
## Generamos un número aleatorio que servirá como referencia del registro en las tablas de MySQL,
este número lo asignamos a la variable "ID"
        my $rango=999999999;
        my $ID=int(rand($rango));
        #print "$ID\n";
return ($ID, $fecha)
}

#################################################################################################
### Subrutina guarda:        En caso de encontrar un comportamiento anormal, esta rutina se encarga
de guardar los datos encontrados en el archivo "anomalias.txt" y generar el archivo
"anomalias_php.txt". Este archivo genera el formato observado en la interfaz web del módulo
escaneo.php

sub guarda{

        my($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes);
        my($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$trafico_tmp);
        my($linea,$imprimo, $fecha,$contador,$var,$espe,$red_serv);
## Se almacenan en variables datos proporcionados por la subrutina que ha mandado a llamar a esta
subrutina
        $imprimo=shift;
        $contador =0;
        my @tmp=@_;
        my $var_tmp="";
        my $query="";
        my($ID,$fecha);

# MYSQL CONFIG VARIABLES
my ($host,$database,$user,$pw,$connect,$execute);
 $host = "localhost";
 $database = "anomalias";
 $user = "root";
 $pw = "Aldeano";
# PERL MYSQL CONNECT()
$connect = Mysql->connect($host, $database, $user, $pw);
# SELECT DB
 $connect->selectdb($database);

## Abrimos archivos
open (ESCRIBE,">>/Listry-AIGC/anomalias.txt");
open (PHP,">>/Listry-AIGC/anomalias_php.txt");
### Mediante switch verificaremos los contenidos de la variable "imprimo", dependiendo del valor
contenido en esta variable, seejecutara el case que le corresponda y guardará datos del arreglo "tmp"
en los archivos, además de indicar que anomalía se ha detectado.
        switch ($imprimo){
                ## Se encontró un escaneo de puertos
                case 0{
                        ($ID,$fecha)=obten_fecha_id();
                        print ESCRIBE "\n\t\t\t\tEscaneo de puertos encontrado:\n\t\t\t\t$fecha\n";
                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip destino    \t:
Pto destino\tPaquetes\tTrafico\n";
                        print PHP"<h3><font color='red'>\t\t\tEscaneo de puertos
encontrado:\n\t\t\t\t$fecha</font></h3>";
                        print PHP"<table with border=4, width='90%'>";
                        print PHP"<tr aling='center' background BGCOLOR='#F54C58'><td><h3><font
color='white', aling='center'>Protocolo</font></h3></td><td><h3><font color='white',
aling='center'>IP origen</font></h3></td><td><h3><font color='white', aling='center'>Puerto
origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                        $espe=0;
                        $linea=$tmp[0];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        $var=$ip_ori;
                foreach(@tmp){
                $linea=$tmp[$contador];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                #$port_tmp="$port_tmp, $port_dest"
                                if ($var eq $ip_ori){
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                        if ($var_tmp){
                                                $var_tmp="$var_tmp, $port_dest";
                                        }
                                        else {
                                                $var_tmp=$port_dest;
                                        }
                                }# fin if
                                else {
```

```
                                        $var=$ip_ori;
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                        ($ID,$fecha)=obten_fecha_id();
                                        $query="insert into epuertos values
('$ID','$fecha','$protocolo', '$ip_ori', '$ip_dest', '$var_tmp')";
                                        $execute = $connect->query($query);
                                        #print "\n$query\n";
                                        print ESCRIBE "\nSe observa que la ip origen: $ip_ori esta
buscando algun puerto disponible dentro de la ip destino: $ip_dest que pueda infectar. \nEl escaneo
de puertos se esta realizando de forma secuencial\nSe ha insertado la anomalia en tabla 'epuertos'
con ID= $ID\n\n";
                                        print PHP"</table>";
                                        print PHP "<h4>\nSe observa que la ip origen: $ip_ori esta
buscando algun puerto disponible dentro de la ip destino: $ip_dest \nque pueda infectar. El escaneo
de puertos se esta realizando de forma secuencial\nSe ha insertado la anomalia en tabla 'epuertos'
con ID= $ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia de
dectectada.\n\n</h4>";
                                        $espe=$contador;
                                        print ESCRIBE "\n\t\t\t\tEscaneo de puertos
encontrado:\n\t\t\t\t\t$fecha\n";
                                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip
destino    \t:  Pto destino\tPaquetes\tTrafico\n";
                                        print PHP"<h3><font color='red'>\t\t\tEscaneo de puertos
encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                                        print PHP"<table with border=4, width='90%'>";
                                        print PHP"<tr aling='center' background
BGCOLOR='#F54C58'><td><h3><font color='white', aling='center'>Protocolo</font></h3></td><td><h3><font
color='white', aling='center'>IP origen</font></h3></td><td><h3><font color='white',
aling='center'>Puerto origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                        $var_tmp=$port_dest;
                                }  ## Fin else
                $contador++;
                }       ## Fin For
                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                         ($ID,$fecha)=obten_fecha_id();
                          $query="insert into epuertos values ('$ID','$fecha','$protocolo',
'$ip_ori', '$ip_dest', '$var_tmp')";
                         $execute = $connect->query($query);
                         #print "\n$query\n";
                         print ESCRIBE "\nSe observa que la ip origen: $ip_ori esta buscando algun
puerto disponible dentro de la ip destino: $ip_dest que pueda infectar. \nEl escaneo de puertos se
esta realizando de forma secuencial\nSe ha insertado la anomalia en tabla 'epuertos' con ID=
$ID\n\n";
                         print PHP"</table>";
                         print PHP "<h4>\nSe observa que la ip origen: $ip_ori esta buscando algun
puerto disponible dentro de la ip destino: $ip_dest \nque pueda infectar. El escaneo de puertos se
esta realizando de forma secuencial\nSe ha insertado la anomalia en tabla 'epuertos' con ID= $ID\nY
se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia de
dectectada.\n\n</h4>";
                }       ### Fin case 0

                ## Se encontró un escaneo de Ip's
                case 1{
                        ($ID,$fecha)=obten_fecha_id();
                        print ESCRIBE "\n\t\t\t\tEscaneo de Ip'S encontrado:\n\t\t\t\t\t$fecha\n";
                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip destino    \t:
Pto destino\tPaquetes\t Trafico\n";
                         print PHP"<h3><font color='purple'>\t\t\tEscaneo IP'S
encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                         print PHP"<table with border=4, width='90%'>";
                         print PHP"<tr aling='center' background BGCOLOR='#501287'><td><h3><font
color='white', aling='center'>Protocolo</font></h3></td><td><h3><font color='white',
aling='center'>IP origen</font></h3></td><td><h3><font color='white', aling='center'>Puerto
origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                         $espe=0;
```

```
                        $linea=$tmp[0];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        $var=$ip_ori;
                foreach(@tmp){
                $linea=$tmp[$contador];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                if ($var eq $ip_ori){
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5></td><td><h5 aling='center'>$trafico</h5></td></tr>";

                                        #$var_tmp="$var_tmp, $ip_dest";
                                        if ($var_tmp){
                                                $var_tmp="$var_tmp, $ip_dest";
                                        }
                                        else {
                                                $var_tmp=$ip_dest;
                                        }
                                }# fin if

                                else {
                                        $var=$ip_ori;
                                        $linea=$tmp[$espe];

        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                        ($ID,$fecha)=obten_fecha_id();
                                        $query="insert into eips values
('$ID','$fecha','$protocolo', '$ip_ori', '$var_tmp', '$port_dest')";
                                        $execute = $connect->query($query);
                                        #print "\n$query\n";
                                        print ESCRIBE "\nSe observa que la ip $ip_ori esta buscando
alguna otra ip dentro de la red que pueda infectar!\nEl escaneo de Ip's realizado es de forma
secuencial con una variacion en el tercer o cuarto octeto dependiendo \ndel caso encontrado\n Se ha
insertado la anomalia en tabla 'eips' con ID= $ID.\n\n";
                                        print PHP "</table>";
                                        print PHP "<h4>\nSe observa que la ip $ip_ori esta buscando
alguna otra ip dentro de la red que pueda infectar!\nEl escaneo de Ip's realizado es de forma
secuencial con una variacion en el tercer o cuarto octeto dependiendo\ndel caso encontrado\n Se ha
insertado la anomalia en tabla 'eips' con ID= $ID\nY se ha notificado via email a:
'aldo\@localhost.localdomain' sobre la anomalia de dectectada.\n\n<\h4>";

                                        $espe=$contador;
                                        print ESCRIBE "\n\t\t\t\tEscaneo de Ip'S
encontrado:\n\t\t\t\t\t$fecha\n";
                                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip
destino    \t:  Pto destino\tPaquetes\t Trafico\n";
                                        print PHP"<h3><font color='purple'>\t\t\tEscaneo IP'S
encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                                        print PHP"<table with border=4, width='90%'>";
                                        print PHP"<tr aling='center' background
BGCOLOR='#501287'><td><h3><font color='white', aling='center'>Protocolo</font></h3></td><td><h3><font
color='white', aling='center'>IP origen</font></h3></td><td><h3><font color='white',
aling='center'>Puerto origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5></td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                        $var_tmp=$ip_dest;
                                }  ## Fin else
                $contador++;
                }       ## Fin For

                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        ($ID,$fecha)=obten_fecha_id();
                        $query="insert into eips values ('$ID','$fecha','$protocolo', '$ip_ori',
'$var_tmp', '$port_dest')";
                        $execute = $connect->query($query);
                        #print"\n$query\n";
                        print ESCRIBE "\nSe observa que la ip $ip_ori esta buscando alguna otra ip
dentro de la red que pueda infectar!\nEl escaneo de Ip's realizado es de forma secuencial con una
variacion en el tercer o cuarto octeto dependiendo \ndel caso encontrado\nSe ha insertado la anomalia
en tabla 'eips' con ID= $ID.\n\n";
```

```
                        print PHP"</table>";
                        print PHP "<h4>\nSe observa que la ip $ip_ori esta buscando alguna otra ip
dentro de la red que pueda infectar!\nEl escaneo de Ip's realizado es de forma secuencial con una
variacion en el tercer o cuarto octeto dependiendo\ndel caso encontrado\nSe ha insertado la anomalia
en tabla 'eips' con ID= $ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la
anomalia de dectectada.\n\n</h4>";
                }       ### Fin case 1

                ## Se encontrófuga de información hacia el exterior con un comportamiento anormal
                case 2{
                        ($ID,$fecha)=obten_fecha_id();
                        print ESCRIBE "\n\t\t\t\tEnvio de informacion hacia el exterior
encontrado:\n\t\t\t\t\t$fecha\n";
                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip destino    \t:
Pto destino\tPaquetes\tTrafico\n";
                        print PHP"<h3><font color='cyan'>\t\t\tEnvio de informacion hacia el
exterior encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                        print PHP"<table with border=4, width='90%'>";
                        print PHP"<tr aling='center' background BGCOLOR='#24E1DB'><td><h3><font
color='white', aling='center'>Protocolo</font></h3></td><td><h3><font color='white',
aling='center'>IP origen</font></h3></td><td><h3><font color='white', aling='center'>Puerto
origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                        $espe=0;
                        $linea=$tmp[0];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        $var=$ip_ori;

                foreach(@tmp){
                $linea=$tmp[$contador];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                if ($var eq $ip_ori){
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";

                                        if ($var_tmp){
                                                $var_tmp="$var_tmp, $port_ori";
                                        }
                                        else {
                                                $var_tmp=$port_ori;
                                        }
                                        #$var_tmp="$var_tmp, $port_ori";
                                }# fin if

                                else {
                                        $var=$ip_ori;
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                        if ($ip_ori =~ /(\d+.\d+).(\d+).(\d+)/){
                                                $red_serv="$1.$2";
                                                #print "$red_serv ";
                                        }
                                        if (($red_serv eq '172.16.5')||($red_serv eq '172.16.10')){
                                                ($ID,$fecha)=obten_fecha_id();
                                                $query="insert into exterior values
('$ID','$fecha','$protocolo', '$ip_ori','$var_tmp', '$ip_dest')";
                                                $execute = $connect->query($query);
                                                #print "\n$query\n";
                                                print ESCRIBE "\nSe observa que la ip origen:
$ip_ori perteneciente a la red de servidores: $red_serv.0/24 esta enviando informacion con un trafico
mayor a 5 MB hacia Ip's fuera del rango permitido. \nTambien se detecta que la Ip origen $ip_ori
utiliza puertos de forma secuencial en el envio de la informacion\nSe ha insertado la anomalia en
tabla 'exterior' con ID= $ID.\n\n";
                                                print PHP"</table>";
                                                print PHP"<h4>\nSe observa que la ip origen:
$ip_ori perteneciente a la red de servidores: $red_serv.0/24 esta enviando \ninformacion con un
trafico mayor a 5 MB hacia Ip's fuera del rango permitido. \nTambien se detecta que la Ip origen
$ip_ori utiliza puertos origen de forma secuencial en el envio de la informacion\nSe ha insertado la
anomalia en tabla 'exterior' con ID= $ID\nY se ha notificado via email a:
'aldo\@localhost.localdomain' sobre la anomalia de dectectada.\n\n</h4>";
                                        }
                                        else {
                                                ($ID,$fecha)=obten_fecha_id();
                                                $query="insert into exterior values
('$ID','$fecha','$protocolo', '$ip_ori','$var_tmp', '$ip_dest')";
                                                $execute = $connect->query($query);
                                                #print "\n$query\n";
```

```
                                                print ESCRIBE "\nSe observa que la ip origen:
$ip_ori perteneciente a una red de usuarios esta enviando informacion con un trafico mayor a 5 MB
hacia Ip's fuera del rango permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos
de forma secuencial en el envio de la informacion\nSe ha insertado la anomalia en tabla 'exterior'
con ID= $ID.\n\n";
                                                print PHP"</table>";
                                                print PHP "<h4>\nSe observa que la ip origen:
$ip_ori perteneciente a una red de usuarios esta enviando \ninformacion con un trafico mayor a 5 MB
hacia Ip's fuera del rango permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos
origen de forma secuencial en el envio de la informacion\nSe ha insertado la anomalia en tabla
'exterior' con ID= $ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la
anomalia de dectectada.\n\n</h4>";
                                        }
                                        $espe=$contador;
                                        ($ID,$fecha)=obten_fecha_id();
                                        print ESCRIBE "\t\t\t\tEnvio de informacion hacia el
exterior encontrado:\n\t\t\t\t\t$fecha\n";
                                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip
destino    \t:  Pto destino\tPaquetes\tTrafico\n";
                                        print PHP"<h3><font color='cyan'>\t\t\tEnvio de informacion
hacia el exterior encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                                        print PHP"<table with border=4, width='90%'>";
                                        print PHP"<tr aling='center' background
BGCOLOR='#24E1DB'><td><h3><font color='white', aling='center'>Protocolo</font></h3></td><td><h3><font
color='white', aling='center'>IP origen</font></h3></td><td><h3><font color='white',
aling='center'>Puerto origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        #print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='10'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                $var_tmp=$port_ori;
                                }  ## Fin else
                $contador++;
                }       ## Fin For
                $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                        if ($ip_ori =~ /(\d+.\d+).(\d+).(\d+)/){
                                $red_serv="$1.$2";
                                #print "$red_serv ";
                        }

                        if (($red_serv eq '172.16.5')||($red_serv eq '172.16.10')){
                                ($ID,$fecha)=obten_fecha_id();
                                $query="insert into exterior values ('$ID','$fecha','$protocolo',
'$ip_ori','$var_tmp', '$ip_dest')";
                                 $execute = $connect->query($query);
                               #print "\n$query\n";
print ESCRIBE "\nSe observa que la ip origen: $ip_ori perteneciente a la red de servidores:
$red_serv.0/24 esta enviando informacion con un trafico mayor a 5 MB hacia Ip's fuera del rango
permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos de forma secuencial en el
envio de la informacion\nSe ha insertado la anomalia en tabla 'exterior' con ID= $ID.\n\n";
                                print PHP"</table>";
                                print PHP"<h4>\nSe observa que la ip origen: $ip_ori perteneciente
a la red de servidores: $red_serv.0/24 esta enviando \ninformacion con un trafico mayor a 5 MB hacia
Ip's fuera del rango permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos origen
de forma secuencial en el envio de la informacion\nSe ha insertado la anomalia en tabla 'exterior'
con ID= $ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia de
dectectada.\n\n</h4>";
                        }
                        else {
                                ($ID,$fecha)=obten_fecha_id();
                                $query="insert into exterior values ('$ID','$fecha','$protocolo',
'$ip_ori','$var_tmp', '$ip_dest')";
                                $execute = $connect->query($query);
                                #print "\n$query\n";
                                print ESCRIBE "\nSe observa que la ip origen: $ip_ori perteneciente
a una red de usuarios esta enviando informacion con un trafico mayor a 5 MB hacia Ip's fuera del
rango permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos de forma secuencial en
el envio de la informacion\nSe ha insertado la anomalia en tabla 'exterior' con ID= $ID.\n\n";
                                print PHP"</table>";
                                print PHP"<h4>\nSe observa que la ip origen: $ip_ori perteneciente
a una red de usuarios esta enviando \ninformacion con un trafico mayor a 5 MB hacia Ip's fuera del
```

```
rango permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos origen de forma
secuencial en el envio de la informacion\nSe ha insertado la anomalia en tabla 'exterior' con ID=
$ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia de
dectectada.\n\n</h4>";
                        }
                }       #### Fin case 2

                ## Ataque denegación de servicios encontradó
                case 3{
                        ($ID,$fecha)=obten_fecha_id();
                        print ESCRIBE "\n\t\t\tAtaque denegacion de servicios
encontrado:\n\t\t\t\t\t$fecha\n";
                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip destino    \t:
Pto destino\tPaquetes\t Trafico\n";
                        print PHP"<h3><font color='blue'>\t\t\tAtaque denegacion de servicios
encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                        print PHP"<table with border=4, width='90%'>";
                        print PHP"<tr aling='center' background BGCOLOR='#08088A'><td><h3><font
color='white', aling='center'>Protocolo</font></h3></td><td><h3><font color='white',
aling='center'>IP origen</font></h3></td><td><h3><font color='white', aling='center'>Puerto
origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                        $espe=0;
                        $linea=$tmp[0];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        $var=$ip_dest;
                foreach(@tmp){
                $linea=$tmp[$contador];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                if ($var eq $ip_dest){
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";


                                        if ($var_tmp){
                                                $var_tmp="$var_tmp, $ip_ori";
                                        }
                                        else {
                                                $var_tmp=$ip_ori;
                                        }
                                        #$var_tmp="$var_tmp, $ip_ori";
                                }# fin if

                                else {
                                        $var=$ip_dest;
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                        ($ID,$fecha)=obten_fecha_id();
                                        $query="insert into DoS values
('$ID','$fecha','$protocolo', '$var_tmp', '$ip_dest', '$port_dest')";
                                        $execute = $connect->query($query);
                                        #print "\n$query\n";
                                        print ESCRIBE "\nSe observa que la Ip destino $ip_dest Esta
recibiendo multiples conexiones sobre le puerto $port_dest de una o varias Ip origen.\nCon un trafico
mayor a 50 MB. Con el objetivo de saturar la conexion\nSe ha insertado la anomalia en tabla 'DoS' con
ID= $ID.\n\n";
                                        print PHP"</table>";
                                        print PHP"<h4>\nSe observa que la Ip destino $ip_dest Esta
recibiendo multiples conexiones sobre le puerto $port_dest de una <br/>o varias Ip origen. Con un
trafico mayor a 50 MB. Con el objetivo de saturar la conexion\nSe ha insertado la anomalia en tabla
'DoS' con ID= $ID \nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia
de dectectada..\n\n</h4>";
                                        $espe=$contador;
                                        print ESCRIBE "\n\t\t\tAtaque denegacion de servicios
encontrado:\n\t\t\t\t\t$fecha\n";
                                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip
destino    \t:  Pto destino\tPaquetes\t Trafico\n";
                                        print PHP"<h3><font color='blue'>\t\t\tAtaque de negacion
de servicios encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                                        print PHP"<table with border=4, width='90%'>";
                                        print PHP"<tr aling='center' background
BGCOLOR='#08088A'><td><h3><font color='white', aling='center'>Protocolo</font></h3></td><td><h3><font
color='white', aling='center'>IP origen</font></h3></td><td><h3><font color='white',
aling='center'>Puerto origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
```

```
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                                        $linea=$tmp[$espe];
        $ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                $var_tmp=$ip_ori;
                                }  ## Fin else
                               $contador++;
                }       ## Fin For

                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        ($ID,$fecha)=obten_fecha_id();
                        $query="insert into DoS values ('$ID','$fecha','$protocolo', '$var_tmp',
'$ip_dest', '$port_dest')";
                        $execute = $connect->query($query);
                        #print "\n$query\n";
                        print ESCRIBE "\nSe observa que la Ip destino $ip_dest Esta recibiendo
multiples conexiones sobre le puerto $port_dest de una o varias Ip origen. Con un trafico mayor a 50
MB.\nCon el objetivo de saturar la conexion\nSe ha insertado la anomalia en tabla 'DoS' con ID=
$ID.\n\n";
                        print PHP"</table>";
                        print PHP"<h4>\nSe observa que la Ip destino $ip_dest Esta recibiendo
multiples conexiones sobre le puerto $port_dest de una o<br/>varias Ip origen. Con un trafico mayor a
50 MB. Con el objetivo de saturar la conexion\nSe ha insertado la anomalia en tabla 'DoS' con ID=
$ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia de
dectectada.\n\n</h4>";
                }       #### Fin case 3
        }    #### Fin switch
close (ESCRIBE);
        close (PHP);
}      ### Fin subrrutina guarda

##############################################################################################################
### Subrutina envia_correo: Si existe el archivo“anomalias.txt” se notificarámedianteél envió de un
correo electrónico que contendrá el contenido este archivo.
sub envia_correo{
        my @arreglo;
        my $tmp;

### Abrimos una tubería hacia sendmail e imprimiremos cabeceras correspondientes para el envió del
correo, además de la información contenida en el archivo “anomalias.txt”
open (MAIL,"|/usr/sbin/sendmail -t");
print MAIL "Content-Type: text/plain; charset=iso-8859-1\n";
print MAIL "To: aldo\@localhost.localdomain\n";
print MAIL "From: aldeano_demon_nfsen\@localhost.localdomain\n";
print MAIL "Subject: Asunto del mensaje-> Alerta!!!!!    Anomalias detectadas\n";
print MAIL "Content-Type: text/plain; charset=iso-8859-1\n";

open (ARCHIVO,"</Listry-AIGC/anomalias.txt");
                        @arreglo=<ARCHIVO>;
close(ARCHIVO);

foreach $tmp (@arreglo){
print MAIL "$tmp";
                }
print MAIL "\n\n\n";
        close (MAIL);
}       ### Fin subrrutina envia correo

##############################################################################################################
## Subrutina compara_ip: Ocupada por subrutina eips. Esta subrutina verificará si se presenta un
escaneo de IP’S
## Ej.   xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy:zz    ->   xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy:zz
##       xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy+1:zz  ->xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy+1.yyy:zz
##       xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy+2:zz  ->   xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy+2.yyy:zz
##       xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy+3:zz  ->   xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy+3.yyy:zz
##                                                    .
##       xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy+n:zz  ->   xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy+n.yyy:zz

sub compara_ip {
### Inicializamos variables ocupadas y almacénanos en ellas los datos provenientes de la subrutina
eips
my($i,$ip_ori, $ip_dest1,$ip_dest2, $ip_dest3, $port_dest, $ip_ori_tmp,
$ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,
$port_dest_tmp,$linea,$cont_tmp,$contador_anom,$contador,@tmp);
my $linea;
my ($var_ipori, $var_ipori_tmp, $var_ipdst, $var_ipdst_tmp);

        $i=shift;
        $ip_ori= shift;
        $ip_dest1=shift;
```

```
        $ip_dest2=shift;
        $ip_dest3=shift;
        $port_dest=shift;
        $ip_ori_tmp=shift;
        $ip_dest_tmp1=shift;
        $ip_dest_tmp2=shift;
        $ip_dest_tmp3=shift;
        $port_dest_tmp=shift;
        $linea=shift;
        $cont_tmp=shift;
        my $contador_anom=shift;
        $contador=shift;
        my @tmp=@_;

### Creamos una variable que contendrá todala dirección IP (xxx.xxx.xxx.xxx) y se comparara esta
variable con255.255.255.255, en caso que se presente este dato salimos.
        my $var_ipdst="$ip_dest1.$ip_dest2.$ip_dest3";

        if($var_ipdst eq '255.255.255.255'){
                #print "No entre";
        }

## Después de comprobar que no sea un broadcast, comparamos con switch indicándonos que estamos
buscando.
#       * Case 1: Se buscara escaneo de IP’S con variación en el 4 octeto
#       * Case 2: Se buscara escaneo de IP’S con variación en el 3 octeto
        else {
                switch ($i){
                        case 1{
                                #print "\nEntre caso 1";
## En caso de encontrar que el valor del 4 octeto es 255, incrementamos en 1 el valor del 3er octeto
                                if ($ip_dest3 == 255) {
                                        $ip_dest2 ++;
                                        $ip_dest3=0;
                                #print"\n\n\n\t\t $ip_dest1.$ip_dest2.$ip_dest3:$port_dest   ";
                                }
### En caso de que sea un valor normal incrementaremos en 1 el valor del 4 octeto para proceder a la
comparación
                                else {
                                        $ip_dest3 ++;
                                }


### Creamos variables temporales que tendrán la dirección IP a comparar, además de la dirección IP
temporal.
                                $var_ipori=$ip_ori;
                                $var_ipori_tmp=$ip_ori_tmp;
                                $var_ipdst="$ip_dest1.$ip_dest2.$ip_dest3:$port_dest";

        $var_ipdst_tmp="$ip_dest_tmp1.$ip_dest_tmp2.$ip_dest_tmp3:$port_dest_tmp";
        #print "$var_ipdst:$port_dest  ===  $var_ipdst_tmp:$port_dest_tmp   ";

### En caso de que las variables IP normales y temporales coincidan en sus valores se aumenta
contador temporal y se guarda la línea que contiene la anomalía en el arreglo “tmp”, además de
guardar el número de la línea en el arreglo “encontre”
                                if (($var_ipori eq $var_ipori_tmp) && ($var_ipdst eq
$var_ipdst_tmp)){
                                        if(@tmp){
                                                push(@tmp,$linea);
                                                $contador_anom ++;
                                                push(@encontre,$cont_tmp);
                                        }
                                        else {
                                                push(@tmp,$guarde[$contador],$linea);
                                                $contador_anom +=2;
                                                push(@encontre,$cont_tmp);
                                        }
                                }

## En caso contrario decrementamos el cuarto octeto
                                else {
                                        #$continua=0;
                                        #print "Diferentes\t";
                                        $ip_dest3 --;
                                }
                        } #fin case 1

### Mismo procedimiento que case 1: La única diferencia es el incremento o decremento del 3er octeto,
dependiendo de la acción efectuada.
                        case 2{
                                #print "\nEntre caso 2";
                                $ip_dest2 ++;
                                #print" $ip_dest1.$ip_dest2.$ip_dest3:$port_dest ===
                                $ip_dest_tmp1.$ip_dest_tmp2.$ip_dest_tmp3:$port_dest_tmp  ";

                                $var_ipori=$ip_ori;
                                $var_ipori_tmp=$ip_ori_tmp;
                                $var_ipdst="$ip_dest1.$ip_dest2.$ip_dest3:$port_dest";
```

```
                        $var_ipdst_tmp="$ip_dest_tmp1.$ip_dest_tmp2.$ip_dest_tmp3:$port_dest_tmp";
                        #print "$var_ipdst:$port_dest  ===  $var_ipdst_tmp:$port_dest_tmp   ";
                                if (($var_ipori eq $var_ipori_tmp) && ($var_ipdst eq
$var_ipdst_tmp)){
                                        if(@tmp){
                                                push(@tmp,$linea);
                                                $contador_anom ++;
                                                push(@encontre,$cont_tmp);
                                        }
                                        else {
                                                push(@tmp,$guarde[$contador],$linea);
                                                $contador_anom +=2;
                                                push(@encontre,$cont_tmp);
                                        }
                                }
                                else {
                                        #$continua=0;
                                        #print "Diferentes\t";
                                        $ip_dest2 --;
                                }
                        }
                        default {

                                #print "\nopcion no valida";
                        }
                }   #fin switch
        }  #fin else
### Devolvemos valores
return ($ip_dest2,$ip_dest3,$contador_anom,@tmp);
}   #fin subrrutina compara_ip para el escaneo ip's

#################################################################################################
### Inicio subrutina compara: Ocupada por la subrutina “epuertos”. Esta subrutina se encarga de
verificar si se presenta un escaneo de puertos

##Ej             xxx.xxx.xxx.xxx:ww       =>       yyy.yyy.yyy.yyy:ww
##              xxx.xxx.xxx.xxx:ww       =>       yyy.yyy.yyy.yyy:ww+1
##              xxx.xxx.xxx.xxx:ww       =>       yyy.yyy.yyy.yyy:ww+2
##              xxx.xxx.xxx.xxx:ww       =>       yyy.yyy.yyy.yyy:ww+3
##                                       .
##              xxx.xxx.xxx.xxx:ww       =>       yyy.yyy.yyy.yyy:ww+n

sub compara {
### Declaración de variables locales y se almacenara en ellas valores obtenidos de la subrutina
“epuertos”.
my ($ip_ori,$ip_dest,$port_dest,$ip_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$cont_tmp,$contador);
        my (@anomalias,@tmp);
        my $linea;
        $ip_ori= shift;
        $ip_dest=shift;
        $port_dest=shift;
        $ip_ori_tmp=shift;
        $ip_dest_tmp=shift;
        $port_dest_tmp=shift;
        $linea=shift;
        $cont_tmp=shift;
        my $contador_anom=shift;
        $contador=shift;
        @tmp=();
        @tmp=@_;
        #print "\n$ip_ori:$port_ori-->$ip_dest:$port_dest\t\t$ip_ori_tmp:$port_ori_tmp -->
$ip_dest_tmp:$port_dest_tmp\n Linea =$linea";

## Incrementamos el puerto destino para proceder a su comparación con los valores temporales
        $port_dest ++;
        #print"\t$port_dest == $port_dest_tmp ";

### Verificamos si se presenta un incremento en el puerto destino, además si direcciones IP orígenes
y destinoson iguales, en caso de que esto se presente esto, guardamos la anomalía detectada en el
arreglo “tmp”, además guardamos en el arreglo“encontre” el número dela línea. También incrementamos
el valor de “contador_anomalo”.
        if (($ip_ori eq $ip_ori_tmp) && ($ip_dest eq $ip_dest_tmp) &&($port_dest eq
$port_dest_tmp)){
                #$continua=1;
                #print "Iguales\t $cont_tmp\t$linea \n";
                if(@tmp){
                        push(@tmp,$linea);
                        $contador_anom ++;
                        push(@encontre,$cont_tmp);
                }
                else {
                        #print "\n$contador Arreglo tmp vacio $guarde[$contador]\n";
                        push(@tmp,$guarde[$contador],$linea);
                        $contador_anom +=2;
                        #print "@tmp\n";
                        push(@encontre,$cont_tmp);
                }
        }
```

```
## En caso contrario todo normal, decrementamos el puerto destino
        else {
                #$continua=0;
                #print "Diferentes\t";
                $port_dest --;
        }
### Devolvemos valores
return ($port_dest,$contador_anom,@tmp);
}   #### Fin subrrutina compara (creada para verificar si existe un escaneo de puertos)

#################################################################################################
### Subrutina Compara_exterior: Ocupada por la subrutina “exterior”. Esta subrutina se encarga de
verificar si se presenta una fuga de información hacia el exterior con comportamiento anómalo
## Ej            xxx.xxx.xxx.xxx:ww       =>      aaa.aaa.aaa.aaa:zz
##               xxx.xxx.xxx.xxx:ww+1     =>      aaa.aaa.aaa.aaa:zz
##               xxx.xxx.xxx.xxx:ww+2     =>      aaa.aaa.aaa.aaa:zz
##               xxx.xxx.xxx.xxx:ww+3     =>      aaa.aaa.aaa.aaa:zz
##                                        .
##               xxx.xxx.xxx.xxx:ww+n     =>      aaa.aaa.aaa.aaa:zz
## Donde aaa.aaa.aaa.aaa Es una Ip exterior a la red normal

sub compara_exterior{
## Declaración de variables locales a ocupar y se les asigna la información recibida de subrutina
exterior.
my($ip_ori, $port_ori, $ip_dest1, $ip_dest2, $ip_dest3, $ip_ori_tmp, $port_ori_tmp,
$ip_dest_tmp1,$linea,$cont_tmp,$contador_anom,$contador,$trafico,$trafico_tmp,@tmp);
        $ip_ori= shift;
        $port_ori=shift;
        $ip_dest1=shift;
        $ip_dest2=shift;
        $ip_dest3=shift;
        $ip_ori_tmp=shift;
        $port_ori_tmp=shift;
        $ip_dest_tmp1=shift;
        $linea=shift;
        $cont_tmp=shift;
        my $contador_anom=shift;
        $contador=shift;
        $trafico=shift;
        $trafico_tmp=shift;
        my @tmp=@_;
        my $red_serv;

### Se incrementa el valor del puerto origen.
        $port_ori ++;

## Se crea una variable especial que contendrá la ip_dst yyy.yyy.yyy.yyy y se compara con
255.255.255.255, en caso de ser verdadero termina la subrutina y se decremento el puerto origen.
        my $var_ipdst="$ip_dest1.$ip_dest2.$ip_dest3";

        if ($ip_ori_tmp =~ /(\d+.\d+).(\d+).(\d+)/){
                $red_serv="$1.$2";
                #print "$red_serv ";
        }
        if ($trafico_tmp =~ /(\d+|\d+.\d+)(\s)(K|M|G|T)/){
                switch ($3){
                        case 'K'{
                                $trafico_tmp= $1*1024;
                                #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'M'{
                                $trafico_tmp= $1*1024*1024;
                                #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'G'{
                                $trafico_tmp= $1*1024*1024*1024;
                                #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'T'{
                                $trafico_tmp= $1*1024*1024*1024*1024;
                                #print "$trafico_tmp \t= $1 * $3\n";
                        }
                } #Fin switch
        } ### Fin if

        if($var_ipdst eq '255.255.255.255'){
                #print "\t $var_ipdst  --->>  No entre";
                $port_ori --;
        }

## En caso contrario se compara el 1 y 2 octeto con direcciones IP destino válidas y si se presenta
esto salimos de la subrutina.
        elsif ( ($ip_dest_tmp1 eq '255.255') || ($ip_dest_tmp1 eq '172.16') || ($ip_dest_tmp1 eq
'172.16') || ($ip_dest_tmp1 eq '172.17')) {

                #print "\t $var_ipdst --->><<----- EnTrE \t $ip_dest_tmp1 \n ";
                $port_ori --;
```

```
        }
## En caso que no se cumpla la condición anterior se tiene una dirección IP exterior, comparamos si
tiene un comportamiento mencionado en el ejemplo, de ser así almacenamos el valor de la línea que
presenta este comportamiento anormal, además de almacenar el número de esta línea en el arreglo
“encontre” e incrementamos el contador.
        #elsif (($ip_ori eq $ip_ori_tmp) && ($port_ori eq $port_ori_tmp)) {
        elsif ((($red_serv eq '172.16.5')||($red_serv eq '172.16.10'))&&($ip_ori eq
$ip_ori_tmp)&&($port_ori eq $port_ori_tmp)&&($trafico_tmp >= 5242880) || (($ip_ori eq
$ip_ori_tmp)&&($port_ori eq $port_ori_tmp)&&($trafico_tmp >= 5242880))) {
                #print"\t Encontre anomalia $var_ipdst --->> \t $ip_dest_tmp1   .   $ip_dest_tmp2 \n
";
                if(@tmp){
                        push(@tmp,$linea);
                        $contador_anom ++;
                        push(@encontre,$cont_tmp);
                }
                else {
                        push(@tmp,$guarde[$contador],$linea);
                        $contador_anom +=2;
                        push(@encontre,$cont_tmp);
                }
        }
## No se detectó nada, se decremento el valor del puerto origen
        else { $port_ori --;}
## Devolvemos valores requeridos.
return ($port_ori,$contador_anom,@tmp);
}  ### Fin subrrutina compara_exterior

#############################################################################################
### Subrutina Compara_dos: Ocupada por la subrutina“DoS”. Esta subrutina se encarga de verificar si
se presenta un ataque denegación de servicios o un ataque de negación de servicios distribuido.
## Ej
##              aaa.aaa.aaa.aaa:ww       =>      xxx.xxx.xxx.xxx:zz
##              aaa.aaa.aaa.aaa:ww       =>      xxx.xxx.xxx.xxx:zz
##              aaa.aaa.aaa.aaa:ww       =>      xxx.xxx.xxx.xxx:zz
##              aaa.aaa.aaa.aaa:ww       =>      xxx.xxx.xxx.xxx:zz
##                                       .
##              aaa.aaa.aaa.aaa:ww       =>      xxx.xxx.xxx.xxx:zz
##Donde:  aaa.aaa.aaa.aaa es cualquier ip.
##              ww cualquier puerto origen igual en todas las ip aaa.aaa.aaa.aaa
##              xxx.xxx.xxx.xxx. es cualquier ip destino igual.
##              zz cualquier puerto destino
##              Además debe de cumplir que el tráfico sea mayor a 50 M

sub compara_dos{
##Declaración de variables locales a ocupar y se les asigna la información recibida de subrutina DoS.

my ($port_dest,$ip_dest,$trafico,$port_dest_tmp,$ip_dest_tmp,$trafico_tmp,$cont_tmp,$contador);
        my (@anomalias,@tmp);
        my $linea;
        $port_dest=shift;
        $ip_dest=shift;
        $trafico=shift;
        $port_dest_tmp=shift;
        $ip_dest_tmp=shift;
        $trafico_tmp=shift;
        $linea=shift;
        $cont_tmp=shift;
        my $contador_anom=shift;
        $contador=shift;
        @tmp=@_;

## El valor almacenado del tráfico en cada flujo se convertirá a bytes.
        if ($trafico_tmp =~ /(\d+|\d+.\d+)(\s)(K|M|G|T)/){

                switch ($3){
                        case 'K'{
                            $trafico_tmp= $1*1024;
                            #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'M'{
                            $trafico_tmp= $1*1024*1024;
                            #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'G'{
                            $trafico_tmp= $1*1024*1024*1024;
                            #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'T'{
                            $trafico_tmp= $1*1024*1024*1024*1024;
                            #print "$trafico_tmp \t= $1 * $3\n";
                        }
                } #Fin switch
        } ### Fin if
        else {
                #print "$trafico_tmp \n";
        }
```

```
## Empezamos comparación. En caso de encontrar broadcats salimos.
        if ($ip_dest_tmp eq '255.255.255.255'){
                #print "No entro\n";
        }

##  Verificamos si presenta un comportamiento anormal: Dirección IP destino actual sea igual a la
dirección IP destino temporal, además de que los puertos origen sean los mismos y el tráfico sea
mayor a 50 M, en caso de presentarse esta anomalía se guarda esta linea como un comportamiento
anormal
        elsif (($ip_dest eq $ip_dest_tmp) &&($port_dest eq $port_dest_tmp) && ($trafico_tmp >=
52428800)){
                #$continua=1;
                #print "Iguales\t $cont_tmp\t$linea \n";
                if(@tmp){
                        push(@tmp,$linea);
                        $contador_anom ++;
                        push(@encontre,$cont_tmp);
                }
                else {
                        push(@tmp,$guarde[$contador],$linea);
                        $contador_anom +=2;
                        #print "Se incremento mi contador en =\t$contador_anom\n";
                        #push(@tmp,$linea);
                        #$contador_anom ++;
                        push(@encontre,$cont_tmp);
                }
        }

## En caso contrario todo normal, salimos
        else {
                #print "Diferentes\t";
        }

### Devolvemos valores
return ($port_dest,$contador_anom,@tmp);
#return (@anomalias);

}       ### Fin subrrutina compara_dos

#############################################################################################################
####                              Subrutina separa:
### Se encarga de separar la información recibida por el colector nfdump de la siguiente forma:
######                            (protocolo) , (ip ori : pto ori) , (ip dst : pto dst)
### Y almacenarla en el arreglo guarda.
sub separa{
        my ($linea, $var, $banderin);
        my @ip;

        $linea=shift;
        @ip=@_;
        $banderin=1;
        #print "$linea\n Recibi arreglo";
                        #print "\n$linea";

### Por medio de expresiones regulares separamos la información de la forma indicada y la guardamos
en el arreglo temporal llamado “ip”
                        if( $linea =~ /ICMP (\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)-
>(\s+)(\d+.\d+.\d+.\d+:\d+.\d+)(\s+)(\d+)(\s+)(\d+)/){

                $var= "ICMP $2 -> $5 $7 $9";
                                push(@ip,$var);
                                $banderin=0;
                }

                        if( $linea =~  /([a-zA-Z]+)(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)-
>(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s+)(\d+|\d+.\d+)(\s)(K|M|G|T)/){
                        #$var= "$1 $3 -> $6 $8 $10 $12 ";
                               $var= "$1 $3 -> $6 $8 $10 $12 $14";
                push(@ip,$var);
                                $banderin=0;
                }

                        if( ($banderin==1) && ($linea =~  /([a-zA-
Z]+)(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)-
>(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s+)(\d+)/)){
                        $var= "$1 $3 -> $6 $8 $10 $12 ";
                               #$var= "$1 $3 -> $6 $8 $10 $12 $14";
                push(@ip,$var);
                                $banderin=0;
                }

                        if( ($banderin==1) && ($linea =~  /([a-zA-
Z]+)(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)-
>(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)(\d+)(\s+)(\d+|\d+.\d+)(\s)(K|M|G|T)/)){
                        $var= "$1 $3 -> $6 $8 $10 $12 ";
                               #$var= "$1 $3 -> $6 $8 $10 $12 $14";
                push(@ip,$var);
                                $banderin=0;
```

```
                }

                        if( ($banderin==1) && ($linea =~  /([a-zA-
Z]+)(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)->(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)(\d+)(\s+)(\d+)/)){

                $var= "$1 $3 -> $6 $8 $10";
                                push(@ip,$var);
                        }
        #print "\nHola imprimo \t@ip";

### Regresamos el arreglo que contiene toda la infroamcion separada
        return (@ip);
} ####fin subrrutina.


#############################################################################################################
#####                             Subrutina agrupa
###Ocupada por la subrutina “epuertos” y “DoS”, la información obtenida por la subrutina “separa”, se
agrupará de la siguiente forma y será pasada como argumento a la subrutina “compara” o “compara_dos”
con el objetivo de verificar si se ha realizado un escaneo de puertos o un ataque de negación de
servicios.
######  xxx.xxx.xxx.xxx :  xxxxx    ->  xxx.xxx.xxx.xxx :  xxxxx        xx
######   (ip ori)    , (pto ori)  ,     (ip dst)     , (pto dst) , (trafico)

sub agrupa {
### Declaramos variables en donde se guardara información recibida y variables que almacenaran la
información separada por las expresiones regulares, con el formato indicado
        my ($linea,$ip_ori,$ip_dest,$port_ori,$port_dest,$trafico,$protocolo,$paquetes);
        $linea=shift;
        my $banderin=1;

        if( $linea =~ /ICMP(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+.\d+.\d+):(\d+.\d+)(\s)(\d+)(\s)(\d+)/){
                $ip_ori=$2;     $port_ori=$3;   $ip_dest=$6;    $port_dest=$7;  $trafico=$11;
        $protocolo="ICMP";       $paquetes=$9;
                #print "$ip_ori:$port_ori -> $ip_dest:$port_dest \tTrafico= $trafico\t";
                $banderin=0;
        }

        if( ($banderin==1) && ($linea =~  /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)/)){
                $ip_ori=$3;     $port_ori=$4;   $ip_dest=$7;    $port_dest=$8;  $trafico="$14 $16";
        $protocolo="$1"; $paquetes="$10 $12";
                 #print "$ip_ori:$port_ori -> $ip_dest:$port_dest \tTrafico= $trafico $14\t";
                $banderin=0;
         }

        if( ($banderin==1) && ($linea =~  /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s)(\d+)/)){
                $ip_ori=$3;     $port_ori=$4;   $ip_dest=$7;    $port_dest=$8;  $trafico="$14";
        $protocolo="$1"; $paquetes="$10 $12";
                 #print "$ip_ori:$port_ori -> $ip_dest:$port_dest \tTrafico= $trafico $14\t";
                $banderin=0;
         }

        if( ($banderin==1) && ($linea =~  /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)/)){
                $ip_ori=$3;     $port_ori=$4;   $ip_dest=$7;    $port_dest=$8;  $trafico="$12 $14";
        $protocolo="$1"; $paquetes=$10;
                 #print "$ip_ori:$port_ori -> $ip_dest:$port_dest \tTrafico= $trafico $14\t";
                $banderin=0;
         }

        if( ($banderin==1) && ($linea =~ /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)(\d+)(\s)(\d+)/)){
$ip_ori=$3;     $port_ori=$4;   $ip_dest=$7;    $port_dest=$8;  $trafico=$12;       $protocolo="$1";
        $paquetes=$10;
                 #print "$ip_ori:$port_ori -> $ip_dest:$port_dest \tTrafico= $trafico\t";
        }

## Devolvemos estas variables que contiene la informaciÃ³n agrupada.
return ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes);

}    #### fin subrrituna agrupa

#############################################################################################################
###                             Subrutina agrupa_ip
###   Ocupada por la subrutina “eips” y “exterior”, la información obtenida por la subrutina “separa”
se agrupará de la siguiente forma y será pasada como argumento a la subrutina
“compara_ip”o“compara_exterior” respectivamente  con el objetivo de verificar si se ha realizado un
escaneo de IP’S o se ha enviado información hacia el exterior.
######  xxx.xxx.xxx.xxx :  xxxxx     ->   xxx.xxx  .    xxx   .   xxx    :  xxxxx        xx
######  (ip ori)        , (pto ori)  ,   (ip dst1) , (ip dst2), (ip dst3), (pto dst)  (trafico)

sub agrupa_ip{
### Declaramos variables en donde se guardara información recibida y variables que almacenaran la
información separada por las expresiones regulares, con el formato indicado
```

```
        my ($linea, $ip_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_ori,$port_dest,$trafico);
        $linea=shift;
        my $banderin=1;

       if( $linea =~ /ICMP(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+).(\d+).(\d+):(\d+.\d+)(\s)(\d+)(\s)(\d+)/){
                $ip_ori=$2;     $port_ori=$3;   $port_dest=$9;  $trafico=$13;
                $ip_dest1=$6;    $ip_dest2=$7;    $ip_dest3=$8;

               #print "$ip_ori:$port_ori -> $ip_dest1.$ip_dest2.$ip_dest3:$port_dest \tTrafico=
$trafico\t";
                $banderin=0;
        }

        if( ($banderin==1) && ($linea =~ /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+).(\d+).(\d+):(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)/)){
               $ip_ori=$3;     $port_ori=$4;   $port_dest=$10;  $trafico="$16 $18";
                $ip_dest1=$7;    $ip_dest2=$8;    $ip_dest3=$9;
                #print "$ip_ori:$port_ori -> $ip_dest1.$ip_dest2.$ip_dest3:$port_dest \tTrafico=
$trafico\t";
                $banderin=0;
        }

        if( ($banderin==1) && ($linea =~  /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+).(\d+).(\d+):(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s)(\d+)/)){
               $ip_ori=$3;     $port_ori=$4;   $port_dest=$10;  $trafico="$16";
                $ip_dest1=$7;    $ip_dest2=$8;    $ip_dest3=$9;
                #print "$ip_ori:$port_ori -> $ip_dest1.$ip_dest2.$ip_dest3:$port_dest \tTrafico=
$trafico $16\t";
                $banderin=0;
       }

        if( ($banderin==1) &&($linea =~  /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+).(\d+).(\d+):(\d+)(\s)(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)/)){

               $ip_ori=$3;     $port_ori=$4;   $port_dest=$10;  $trafico="$14 $16";
                $ip_dest1=$7;    $ip_dest2=$8;    $ip_dest3=$9;
                #print "$ip_ori:$port_ori -> $ip_dest1.$ip_dest2.$ip_dest3:$port_dest \tTrafico=
$trafico $16\t";
                $banderin=0;
       }

        if( ($banderin==1) && ($linea =~ /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+).(\d+).(\d+):(\d+)(\s)(\d+)(\s)(\d+)/)){

               $ip_ori=$3;     $port_ori=$4;   $port_dest=$10;  $trafico=$14;
                $ip_dest1=$7;    $ip_dest2=$8;    $ip_dest3=$9;
                #print "$ip_ori:$port_ori -> $ip_dest1.$ip_dest2.$ip_dest3:$port_dest \tTrafico=
$trafico\t";

        }

return ($ip_ori,$port_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_dest,$trafico);
}    ################fin subbrutina agrupa_ip############################
# The Cleanup function is called, when nfsend terminates. It's purpose is to give the
# plugin the possibility to cleanup itself. It's return value is discard.
sub Cleanup {
syslog("info", "Plugin escaneo Cleanup");
# not used here
}

1;
```

Módulo “escaneo.php”

```
<?php
/*************************************************************
*************************************************************
**********           Plugin escaneo.php              ********
**********  Creado por: Aldo Iván Girón Capistrán      ********
**********  Última fecha de modificación: 1/08/2010  ********
*************************************************************
*************************************************************

*                   Objetivo del plugin
* Mostrar los resultados obtenidos por escaneo.pm en el navegador web
*/

function escaneo_ParseInput( $plugin_id ) {
/*
      Esta función no es ocupada en este plugin, pero debe de existir.
      */
} // Fin escaneo_ParseInput

/*
        Función que se ejecutará al momento de mandar llamar al plugin en el navegador web
*/
function escaneo_Run( $plugin_id ) {
```

```
global $VARDIR;
// Verificamos que estemos en el profile live
if ($_SESSION['profile'] == 'live') {

                print "<h2><b><font color='red', aling='center'>\t\t\tObjetivo del
plugin:<br/>Analizar el ultimo archivo nfcapd obtenido en busqueda de anomalias tipicas de algun
malware.</font></b></h2>";

// Abrimos el archivo a mostrar en caso de haber detectado anomalías y mostramos el contenido del
archivo
                if (fopen("/Listry-AIGC/anomalias_php.txt","r")){

                        print "<h2><b><font color='green', aling='center'>Se encontraron las
siguientes anomalias:</font></b></h2>";
                        echo "<pre>";

                        $lines = file("/Listry-AIGC/anomalias_php.txt");

                        foreach ($lines as $line){
                                echo "$line";
                        }
                        echo "</pre>";
                }
// En caso de no existir el archivo notificamos que no se encontró ninguna anomalía
                else {
                        print "<h2><b><font color='green', aling='center'>No se encontraron
anomalias en el analisis realizado</font></b></h2>";
                }
//Mostramos el archivo que se capturo con ayuda de nfcapd.
                print "\n\n";
                print "<h2><b><font color='green', align='center'>Se analizo el
archivo:</font></b></h2>\n";
        echo "<pre>";
        #$logfile = "$VARDIR/tmp/trackstats.log";
        $logfile="/Listry-AIGC/salida.txt";
        $lines = file($logfile) ;

        foreach ($lines as $line) {
echo htmlspecialchars($line) ;
         }
        echo "</pre>";
        } else {
print "<h3>plugin not applicable for profile: " .$_SESSION['profile'] . "</h3>" ;
       }
} // Fin Escaneo_Run
?>
```