# Universidad Nacional Autónoma de México

# Dirección General de Cómputo y de Tecnologías de Información y Comunicación

Plan de Becarios en Seguridad Informática
14° Generación

Documentación del proyecto de Perl

Equipo 7:

- Urbina Garrido Mauricio
- Valverde Martínez Alberto

# Repositorio

https://github.com/MauricioUrb/proyectoPerl.git

**Nota**: Se detalla completamente el funcionamiento de la herramienta el la documentación POD.

# Breve descripción Courier POP

## Archivo de configuración: courier-pop_eq7.conf

Este archivo contiene los parámetros con el cual funcionará el script.

```
# Configuraciones generales del programa en Perl (courier pop)
[ courierPop_eq7 ]
log = /var/log/courier-pop_eq7.log

# Configuración de servicio a monitorear
[ courierPop ]
enable = yes
log = /var/log/courier-pop.log
attempts = 10
time = 60
```

## Script en perl

### Parte inicial del script

Se comienza con la preparación de las variables globales a usar. También desde el inicio se revisa la existencia del directorio donde estarán los logs, de no existir, se crea y se cambian los permisos para tener una correcta escritura y lectura de éste.

```
$archivoLogs = "/var/log/mail.log";
$fechaGlobal = "";
@months = qw(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec);
$globalYear = 0;
$globalHour = 0;
$globalMin = 0;
while(1){
    unless (-d "/var/log/courier-pop_eq7"){
        system("sudo mkdir /var/log/courier-pop_eq7");
        system("sudo chmod 777 /var/log/courier-pop_eq7");
        system("sudo touch /var/log/courier-pop_eq7/courier-pop_eq7.log");
        system("sudo chmod 777 /var/log/courier-pop_eq7/courier-pop_eq7.log");
    }
    open (REGLOG, ">>", "/var/log/courier-pop_eq7/courier-pop_eq7.log") or die $!;
    #Se calcula la fecha de hoy
    ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime();
    $fechaGlobal = ($year+1900)." ".($mon+1)." ".$mday." ".$hour.":".$min.":".$sec;
    $globalYear = $year+1900;
    $globalHour = $hour;
    $globalMin = $min;
    print REGLOG "$fechaGlobal Se ha iniciado el servicio\n";
    open (LOGF, "<", $archivoLogs) or die $!;
    # Se limpia el arreglo
    @registros = ();
```

## Lectura de logs

Se abre el archivo "*/var/log/mail.log*" y se leen todos los registros que contengan "imapd: ", dado que estos son los que registran la actividad de intento de ingreso, y será mediante esto que se identificará si hay ataques en la subrutina "analisis".

```
#Apertura de archivo de logs
while (<LOGF>) {
    #Agregamos al arreglo y mandamos a la función
    chomp $_;
    if ($_ =~ /imapd: LOGIN/){
        push @registros, $_;
    }
}
close(LOGF);
analisis(@registros);
```

## Lectura del archivo de configuración

Se abre el archivo de configuración para la herramienta y se parsean los valores que se quieren leer, asignándolos a sus respectivas variables.

```perl
$archivoConf = "courier-pop_eq7.conf";
$config = Config::Tiny->read($archivoConf);

$logFile  = $config->{courierPop_eq7}{log};
$enable   = $config->{courierPop}{enable};
$log      = $config->{courierPop}{log};
$attempts = $config->{courierPop}{attempts};
$time     = $config->{courierPop}{time};
```

## Subrutina: analisis

Aquí se analizan todos los logs recibidos por la lectura del archivo. Se agrupan los resultados de otra expresión regular por la fecha y las direcciones ip.

```perl
sub analisis {
    %hosts = ();
    foreach $registro (@_){
        $registro =~ m#([A-Z][a-z]+ \d+ \d+:\d+:\d+).*\[(.*:\d+\.\d+\.\d+\.\d+)\]#;
        #$1 -> Fecha
        #$2 -> IP
        $horaReg = epoch($1) + (3600 * 5);
        if($horaReg >= (time - $time) && time >= $horaReg ){
            unless(exists($hosts{"$2"})){
                $hosts{"$2"} = epoch($1);
            }else{
                $valor = $hosts{"$2"};
                $hosts{"$2"} = "$valor ".epoch($1);
            }
        }
    }

    foreach $key (keys %hosts){
        @fechas = split " ", $hosts{$key};
        $size = scalar @fechas;
        if($size >= $attempts){
            bloqueo($key);
        }
    }
}
```

Se compara si la fecha del registro corresponde a un tiempo intermedio entre la fecha y hora actual menos el tiempo especificado en el archivo de configuración y esta misma hora local. Esto se logra mediante el formato epoch.

Si la condición anterior se cumple, entonces se crea un hash donde las llaves son las direcciones ip y sus valores una cadena que contiene todas las horas en las que se intentó un ingreso.

Posteriormente, para cada llave se cuenta la cantidad de internos que hubieron en el lapso de tiempo especificado en el archivo de configuración. Si éstos igualan o superan el el máximo de intentos, entonces se bloquea la dirección ip.

## Subrutina: epoch

Devuelve la fecha del registro en formato epoch.

```
sub epoch{
    $fecha = shift;
    $fecha =~ m#([A-Z][a-z]+) (\d+) (\d+):(\d+):(\d+)#;
    $index = first_index { $_ eq $1 } @months;
    #$sec,$min,$hour,$mday,$mon,$year
    return timegm($5,$4,$3,$2,$index,$globalYear);
}
```

## Subrutina: bloqueo

Se bloquea mediante iptables y se agrega al log junto con una impresión en pantalla para indicar que se bloqueó la ip.

```
sub bloqueo{
    $ip = shift;
    $ip =~ m#(.*):(\d+\.\d+\.\d+\.\d+)#;
    #$1 -> IPv6
    #$2 -> IPv4
    #Se bloquea la IP hasta las 23:59 UTC del dia en que se detectó
    $block_ipv6 = `sudo ip6tables -A INPUT -s $1 -j DROP -m time --timestart $globalHour:$globalMin --timestop 23:59`;
    $save_ipv6 = `sudo /sbin/ip6tables-save`;
    $block_ipv4 = `sudo iptables -A INPUT -s $2 -j DROP -m time --timestart $globalHour:$globalMin --timestop 23:59`;
    $save_ipv4 = `sudo /sbin/iptables-save`;
    print "$fechaGlobal  Se bloqueó la ip: $ip\n";
    print REGLOG "$fechaGlobal  Se bloqueó la ip: $ip\n";
}
```

# Pruebas

## Ejecución de programa

El programa se ejecuta y busca los registros con un tiempo menor a la hora actual - tiempo establecido en el archivo de configuración, es decir empieza a evaluar los registros a partir de la hora en la que se inicia el servicio y tiene un margen de tiempo para analizarlos. Una vez que se lee una IP que cumple con este tiempo se evalúa si ha sobrepasado los intentos establecidos, de ser así bloquea la IP hasta las 23:59 del día en el que se detectó. El servicio actualiza los registros del log por lo que sólamente se analizan los nuevos que cumplan con el tiempo establecido, esto lo hace hasta que se detiene el servicio.

```
alberto@kali:~/proyectoPerl$ perl proyecto.pl
2020 9 21 11:6:52  Se bloqueó la ip: 2001:DB8:2de::e13:168.224.5.1
```

## Bloqueo de iptables

```
alberto@kali:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  168.224.5.1          anywhere             TIME from 16:06:00 to 23:59:00 UTC

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
alberto@kali:~$ sudo ip6tables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all      2001:db8:2de::e13    anywhere             TIME from 16:06:00 to 23:59:00 UTC

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

## Log de la herramienta

```
alberto@kali:~$ sudo cat /var/log/courier-pop_eq7/courier-pop_eq7.log
2020 9 21 11:6:52  Se bloqueó la ip: 2001:DB8:2de::e13:168.224.5.1
2020 9 21 11:33:29  Se bloqueó la ip: 2005:DB8:2df::e13:168.23.5.6
```

Si la herramienta se ejecuta se pueden observar los print que indican cuando se bloqueó una dirección IP.

Si se ejecuta como servicio, se observa que se inicia correctamente, y al ver los logs también va bloqueando los casos que va leyendo.

```
alberto@kali:~/proyectoPerl$ perl courier-pop.service.pl start
/var/log/courier-pop_eq7/courier-pop_eq7.logCourier-pop                    [Started]
alberto@kali:~/proyectoPerl$ sudo cat /var/log/courier-pop_eq7/courier-pop_eq7.log
2020 9 21 11:6:52  Se bloqueó la ip: 2001:DB8:2de::e13:168.224.5.1
2020 9 21 11:33:29  Se bloqueó la ip: 2005:DB8:2df::e13:168.23.5.6
2020 9 21 12:36:8  Se bloqueó la ip: 2005:DC8:2df::e13:168.23.5.7
2020 9 21 12:42:49  Se bloqueó la ip: 2005:DC8:2ff::e13:168.21.5.7
2020 9 21 13:10:37  Se bloqueó la ip: 2015:aC8:2ff::e13:163.21.5.7
2020 9 21 13:29:21 Se ha iniciado el servicio
alberto@kali:~/proyectoPerl$ perl courier-pop.service.pl stop
/var/log/courier-pop_eq7/courier-pop_eq7.logCourier-pop                    [Stopped]
alberto@kali:~/proyectoPerl$ sudo cat /var/log/courier-pop_eq7/courier-pop_eq7.log
2020 9 21 11:6:52  Se bloqueó la ip: 2001:DB8:2de::e13:168.224.5.1
2020 9 21 11:33:29  Se bloqueó la ip: 2005:DB8:2df::e13:168.23.5.6
2020 9 21 12:36:8  Se bloqueó la ip: 2005:DC8:2df::e13:168.23.5.7
2020 9 21 12:42:49  Se bloqueó la ip: 2005:DC8:2ff::e13:168.21.5.7
2020 9 21 13:10:37  Se bloqueó la ip: 2015:aC8:2ff::e13:163.21.5.7
2020 9 21 13:29:21 Se ha iniciado el servicio
2020 9 21 13:29:43 Se ha detenido el servicio
```

Al detenerse la herramienta, se muestra de igual forma que funciona sin problemas.

# Servicio

```
use Daemon::Control;
use Time::Local;
use Config::Tiny;

$path = `pwd`;
chomp($path);

$archivoConf = "courier-pop_eq7.conf";
$config = Config::Tiny->read($archivoConf);
$logFile  = $config->{courierPop_eq7}{log};
print $logFile;
open (REGLOG, ">>", $logFile) or die $!;
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime();
$fechaGlobal = ($year+1900)." ".($mon+1)." ".$mday." ".$hour.":".$min.":".$sec;

if($ARGV[0] eq "start"){
    print REGLOG "$fechaGlobal Se ha iniciado el servicio\n";
}
elsif($ARGV[0] eq "stop"){
    print REGLOG "$fechaGlobal Se ha detenido el servicio\n";
}

close REGLOG;

exit Daemon::Control->new(
    name        => "Courier-pop",
    lsb_start   => "$syslog $remote_fs",
    lsb_stop    => "$syslog",
    lsb_sdesc   => "Courier-pop",
    lsb_desc    => "courier-pop_eq7.pl daemon",
    path        => $path."/daemon.pl",
    program     => "$path/courier-pop_eq7.pl",
    pid_file    => "/tmp/mydaemon.pid",
    stderr_file => "/tmp/mydaemon.out",
    stdout_file => "/tmp/mydaemon.out",
    fork        => 2,
)->run;
```

Se creó un script que funciona del mismo modo que un demonio, iniciando o deteniendo el servicio, registrándose de igual forma en el archivo de log.