[ANEXO Nº 1]

[Lista de sistemas SCADA propietarios]

Nombre	Sitio Web
AggreGate	http://aggregate.tibbo.com/
AutomationX	http://www.automationx.ca/
BroadWin WebAccess	http://www.broadwin.com/
Catalyst Pro / Easy HMI	http://www.bastools.com/
Circuitor Power Studio	http://www.circutor.es/
Complete SCADA	http://www.completescada.com
CSWorks (Control System Works)	http://www.controlsystemworks.com/
DEXMA	http://www.dexmatech.com/
e-FlowNet portal	http://www.pomiary.com.pl/
EnergoSCADA	http://rpk-su.info/
ICONICS GENESIS32 / GENESIS64	http://www.iconics.com/
ICSCADA	http://www.icscada.ca/
iDigi Solutions	http://www.idigi.com/
Ignition	http://www.inductiveautomation.com/
IGSS	http://www.7t.dk/igss/
Inductive Automation	http://www.inductiveautomation.com/
Indusoft Web Studio / CEView	http://www.indusoft.com/
IntegraXor	http://www.integraxor.com/
IntelliCom NetBiter webSCADA	http://www.intellicom.se/webscada.cfm
Matrikon Operational Insight	http://www.matrikon.com/
NetSCADA	http://www.scadalink.com/
Nortech Central iHost	http://www.webscada.co.uk/
OPC Systems.NET	http://www.opcsystems.com/
Opto 22	http://www.opto22.com/
PcVue Solutions	http://www.arcinfo.com/
PowerVise Web SCADA	http://www.energywebnetwork.com/
PQSCADA / Power IQ / Elspec Investigator	http://www.elspec-ltd.com/
PROASIS DCS-Win	http://www.desin.com/
Proficy iFix	http://www.ge-ip.com/
Rockwell Automation RSView32	http://www.rockwellautomation.com/
RTAP	http://www.rtapscada.com/
Schneider PowerLogic	http://www.powerlogic.com/
Siemens SIMATIC WinCC	http://www.automation.siemens.com/
SmartPower	http://www.smartpower.com.ve/
Storozh	http://storozh.com.ua/
Vista Control Systems (Vsystem)	http://www.vista-control.com/
Vista	http://www.vista-control.com/
WebSCADA	http://www.webscada.com/
Winlog Pro / Lite	http://www.sielcosistemi.com/
Wonderware	http://global.wonderware.com/

NOTA: la presente es una lista de soluciones propietarias para sistemas SCADA encontradas durante el desarrollo del presente trabajo. No representa una lista completa de soluciones disponibles.

[ANEXO Nº 2]

[STUXNET]

"Stuxnet" es el nombre de un gusano informático identificado por primera vez en Junio de 2010. Es un programa escrito originalmente para atacar un sistema de control industrial o un conjunto de sistemas similares y la evidencia sugiere que fue diseñado por los gobiernos de EE.UU. e Israel y estaba particularmente dirigido a las instalaciones del programa nuclear de Irán [1].

Su objetivo final era reprogramar sistemas de control industrial modificando el código en controladores lógicos programables (PLCs) para hacerlos trabajar de determinada manera y además ocultar los cambios del operador del sistema. El propósito final de Stuxnet era el sabotaje de instalaciones industriales mediante la reprogramación de PLCs [2]. Stuxnet incluye una serie de elementos, como el primer rootkit para un PLC conocido hasta la fecha, vulnerabilidades de día cero, Windows rootkits, técnicas de evasión de antivirus, complejos procesos de inyección de código, rutinas de propagación vía red, actualizaciones punto-a-punto (p2p) y una interfaz de comando y control. En particular, el blanco principal de Stuxnet era "Siemens SIMATIC Step 7", un software de programación de PLCs para Windows.

Symantec realizó un análisis exhaustivo del código de Stuxnet, así como de registros de acceso y otros elementos pertinentes. Como resultado, generaron un extenso reporte donde detallan el funcionamiento del sistema y ofrecen algunos detalles sobre la manera en que pudo lograrse la infección:

- La infección inicial probablemente fue lograda a través de un contratista de la instalación industrial. Este es un punto de fácil acceso, ya que un contratista

[1] Stuxnet. <http://en.wikipedia.org/wiki/Stuxnet> [Consulta: 2013]
[2] Symantec Security Resources, "W32.Stuxnet Dossier", Houston, Texas, USA, February 2011.

tendrá inherentemente menores niveles de seguridad que la instalación per se. Luego fue propagada a las instalaciones industriales, posiblemente mediante un dispositivo de almacenamiento extraíble (USB).

- Una vez en la red corporativa de la instalación, que era el blanco final del ataque, Stuxnet comenzó a propagarse de manera sigilosa en busca de los computadores usados para programar PLCs. En este caso en particular, por seguridad, dichos computadores no estaban conectados a Internet o la red corporativa. Eran básicamente estaciones de trabajo sin conectividad. A pesar de esto, debido a que Stuxnet puede no sólo propagarse por la red sino también vía dispositivos extraíbles, pudo finalmente infectarlas.

- La técnica para modificar el código de los PLCs fue la de sustituir un DDL del entorno de programación. El DLL es uno utilizado en la comunicación entre el entorno y el PLC al momento de escribir los bloques de programación. El DLL fue modificado para que monitorease los bloques escritos en el PLC e introdujese modificaciones directamente en el código compilado, que sería posteriormente ejecutado por los PLCs. Para ocultar el hecho de que el código fue modificado, también monitorea la lectura de los bloques de código y remueve las modificaciones insertadas antes de que sea recibido y desplegado ante el operador.

- La táctica de sabotaje usada fue la de modificar la frecuencia de trabajo de centrifugadoras. De esta manera se buscaba sacarlas de operación, bien sea haciendo pensar a los operadores que estaban funcionando indebidamente u ocasionando daños reales a las mismas.

Luego del descubrimiento de Stuxnet otras plataformas relacionadas han sido descubiertas, entre ellas: Duqu, Flame y Gauss.

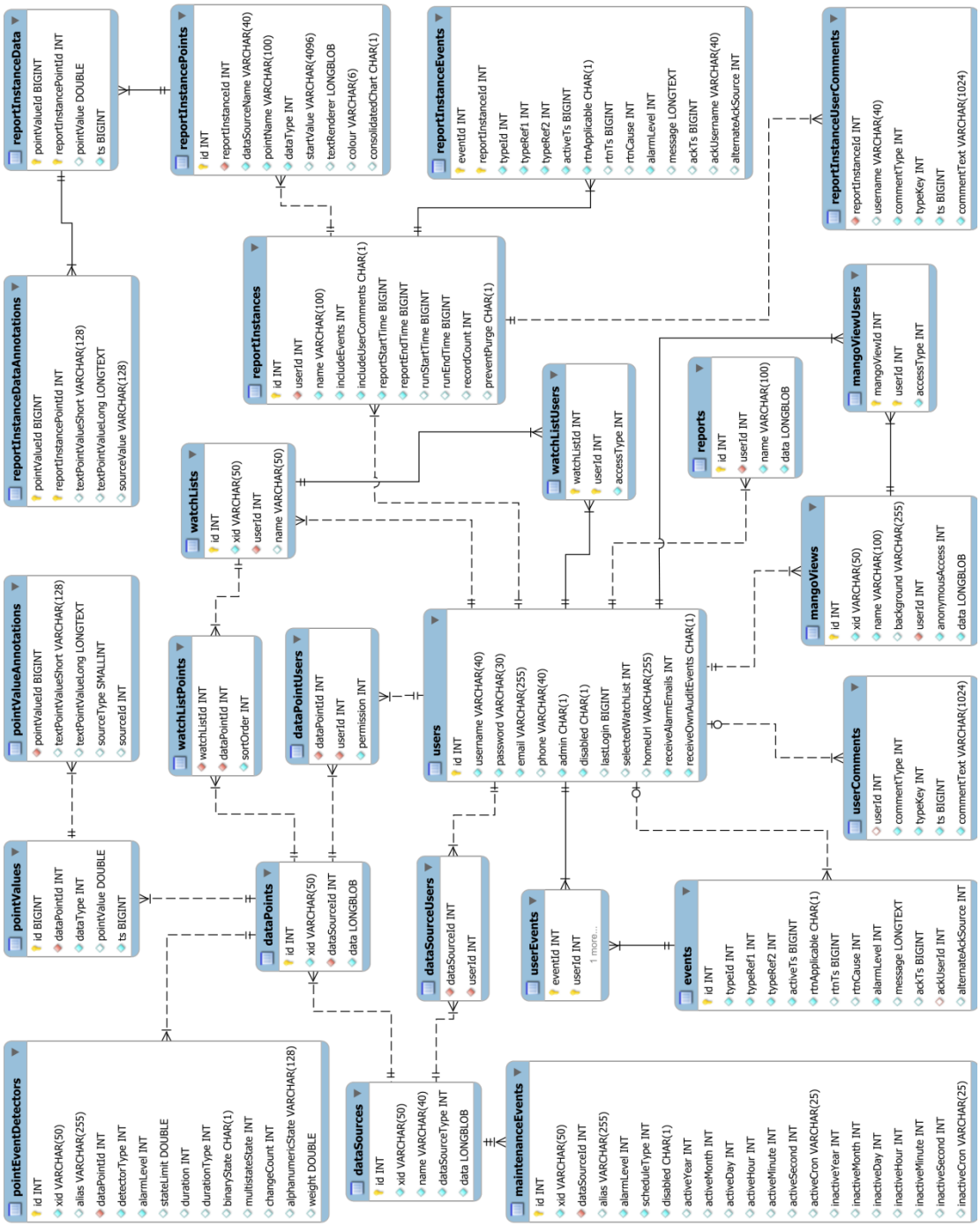
[ANEXO Nº 3]

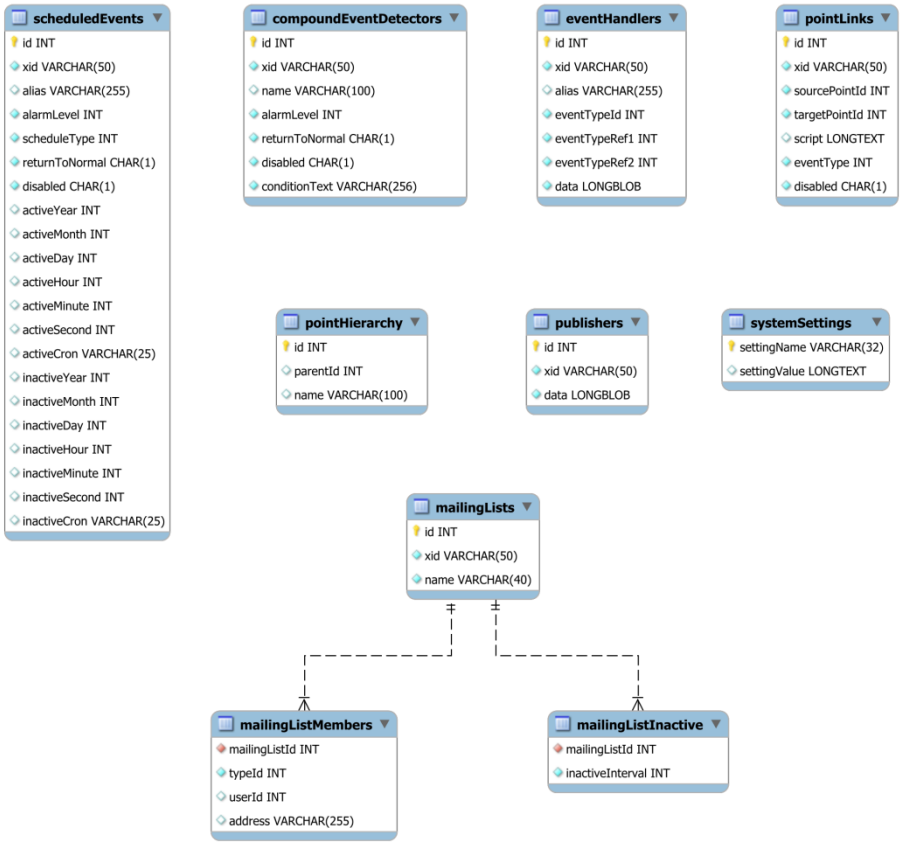
[Tabla comparativa de sistemas SCADA de código abierto]

Nombre	Sitio Web	Licencia	Copyleft	Web	Activo	Documenta-ción	Multi-plataforma	Notas
ARGOS	http://www.cintal.com.ve/argos/	GPL	Sí	No	No	No	Sí*	*Tiene problemas bajo Windows.
FreeSCADA	http://www.free-scada.org/	Common Public License 1.0	Sí	Sí	No	No	No*	*Desarrollado sobre plataformas .NET para Windows.
Likindoy	http://www.likindoy.org/ (no disponible)	GPLv2	Sí	Sí*	No	Sí	Sí	*El HMI es limitado.
Lintouch	http://lintouch.org/	GNU	Sí	No	No	No	Sí	
Mango M2M	http://mango.serotoninsoftware.com/	GPLv3	Sí	Sí	Sí*	Sí**	Sí	*Su desarrollo continúa bajo otra licencia. **No hay documentación para el código o la arquitectura de software.
OpenAPC	http://www.openapc.com/	propia	No	No	Sí	Sí	Sí	
OpenSCADA	http://www.openscada.org/	LGPLv3	Sí	No	Sí	Sí	Sí	
Proview	http://www.proview.se/	GPL	Sí	No	Sí	Sí	No	
ScadaBR	http://www.scadabr.com.br/	Apache	No	Sí	Sí	Sí*	Sí	Basado en Mango M2M. **No hay documentación para el código o la arquitectura de software.
ScadaBR CE (Comunity Edition)	http://www.scadabr.com.br/	GPLv3	Sí	Sí	Sí	Sí*	Sí	Basado en Mango M2M. **No hay documentación para el código o la arquitectura de software.
Stantor	http://stantor.free.fr/	GPL	Sí	Sí	Sí	Parcial	No	
SZARP	http://www.szarp.org/	GPLv2	Sí	No	Sí	Sí*	No**	*La mayoría de la documentación está disponible sólo en Polaco. **Sólo las aplicaciones cliente son multiplataforma.
VISUAL	http://visual.sourceforge.net/	GPLv2	Sí	No	No	Parcial	No*	*Sólo el cliente HMI es multiplataforma.

[ANEXO Nº 4]

[Modelo Entidad-Relación de Mango M2M v1.12.3]



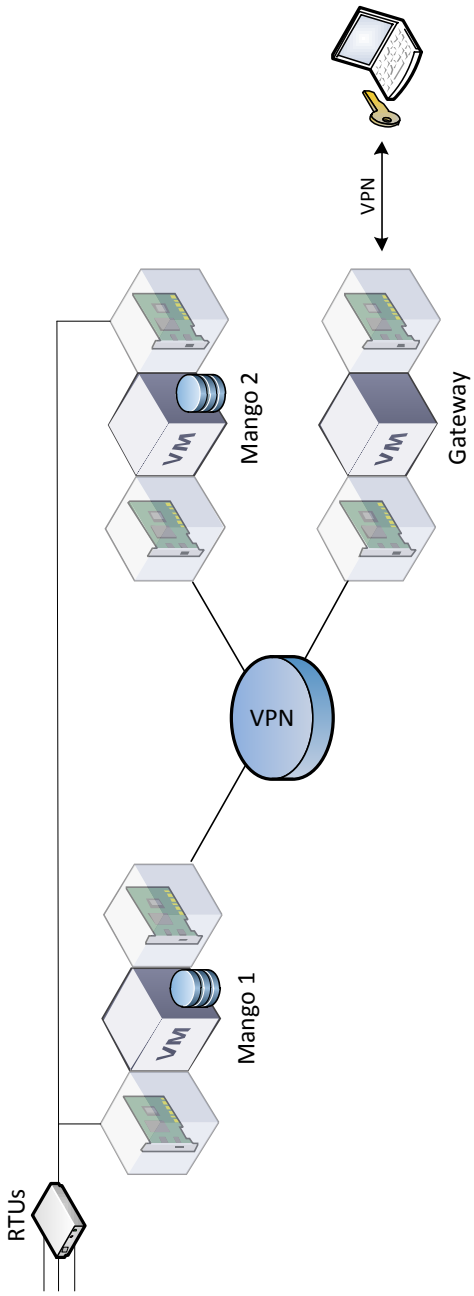

scheduledEvents
id INT
xid VARCHAR(50)
alias VARCHAR(255)
alarmLevel INT
scheduleType INT
returnToNormal CHAR(1)
disabled CHAR(1)
activeYear INT
activeMonth INT
activeDay INT
activeHour INT
activeMinute INT
activeSecond INT
activeCron VARCHAR(25)
inactiveYear INT
inactiveMonth INT
inactiveDay INT
inactiveHour INT
inactiveMinute INT
inactiveSecond INT
inactiveCron VARCHAR(25)
compoundEventDetectors
id INT
xid VARCHAR(50)
name VARCHAR(100)
alarmLevel INT
returnToNormal CHAR(1)
disabled CHAR(1)
conditionText VARCHAR(256)
eventHandlers
id INT
xid VARCHAR(50)
alias VARCHAR(255)
eventTypeId INT
eventTypeRef1 INT
eventTypeRef2 INT
data LONGBLOB
pointLinks
id INT
xid VARCHAR(50)
sourcePointId INT
targetPointId INT
script LONGTEXT
eventType INT
disabled CHAR(1)
pointHierarchy
id INT
parentId INT
name VARCHAR(100)
publishers
id INT
xid VARCHAR(50)
data LONGBLOB
systemSettings
settingName VARCHAR(32)
settingValue LONGTEXT
mailingLists
id INT
xid VARCHAR(50)
name VARCHAR(40)
mailingListMembers
mailingListId INT
typeId INT
userId INT
address VARCHAR(255)
mailingListInactive
mailingListId INT
inactiveInterval INT

[ANEXO Nº 5]

[Diagrama UML de la clase “PointComponent”]

[ANEXO Nº 6]

[Diagrama de interfaces de red del SCADA web implementado]



[ANEXO Nº 7]

[Perro guardián de MTUs (código fuente)]

```
#=================================================================================================
# Alejandro González [2013]
# Universidad Central de Venezuela - Facultad de Ingeniería - Escuela de Ingeniería Eléctrica
# MANGO BACKUP START
#=================================================================================================
use strict;
use warnings;
use IO::Socket::PortState qw(check_ports);
use IO::Socket;
use LWP::UserAgent;

#=================================================================================================
# CONFIGURACIÓN
#=================================================================================================
my $timeout = 1; # en segundos
my $main_mtu_ip = '192.168.2.136';
my $this_mtu = '192.168.2.137';
my $tick = 4; # en segundos
my $retries_default = 3; # cuantos reintentos
my $httpds_var = 'watchdog_respaldo';
my $httpds_port = 8080;
my $only_check_db = 0;
my $mysql_admin_user = 'root';
my $mysql_admin_pass = '';

#=================================================================================================
# CAMPOS
#=================================================================================================
my $retries = $retries_default;
my $ua = LWP::UserAgent->new;

my %check_main_mtu = (
    tcp => {
        8080 => {
            name => 'Tomcat',
        },
        3306 => {
            name => 'MySQL',
        }
    }
);

my %check_tomcat = (
    tcp => {
        8080 => {
            name => 'Tomcat',
        }
    }
);

#=================================================================================================
# FUNCIONES
#=================================================================================================

#-------------------------------------------------------------------------------
sub push_state {
    my ($state, $ip) = @_;

    check_ports($ip, $timeout, \%check_tomcat);

    if ($check_tomcat{"tcp"}->{8080}->{open}) {

        mylog("Actualizando estado en $ip.");

        my $req = HTTP::Request->new(GET => "http://$ip:$httpds_port/httpds?$httpds_var=$state");
        my $res = $ua->request($req);
    }
    else {
        mylog("Saltando $ip. No hay conectividad.");
    }
}
```

```
#------------------------------------------------------------------------------
sub get_formated_local_datetime {
    my ($second, $minute, $hour, $dayOfMonth, $month, $yearOffset) = localtime();
    my $year = 1900 + $yearOffset;
    return sprintf( "%0u/%02u/%02u %02u:%02u:%02u", $year, $month, $dayOfMonth, $hour, $minute,
$second);
}

#------------------------------------------------------------------------------
sub mylog {
    my ($text) = @_;

    my $date = get_formated_local_datetime();
    print "[$date] ".$text."\n";
}

#------------------------------------------------------------------------------
sub start_backup_mtu {
    check_ports($this_mtu, $timeout, \%check_tomcat);

    # arrancarlo sólo si aún no está corriendo
    if (not $check_tomcat{"tcp"}->{8080}->{open}) {

        # detener primero lo replicación (si vuelva a arrancar el MTU 01, no sincronizar)
        if ($mysql_admin_pass eq '') {
            system("mysql --user=$mysql_admin_user -e \"STOP SLAVE\"");
        }
        else {
            system("mysql    --user=$mysql_admin_user    --password=$mysql_admin_pass    -e    \"STOP
SLAVE\"");
        }

        system('/etc/init.d/tomcat7 start');
    }
}

#==============================================================================================================
# SCRIPT
#==============================================================================================================
my $state = 0;
while (1) {
    sleep $tick;

    check_ports($main_mtu_ip, $timeout, \%check_main_mtu);
    if (not $check_main_mtu{"tcp"}->{3306}->{open}) {
        $retries--;
        if ($retries < 0) {
            $retries = $retries_default;
            start_backup_mtu();
        }
    }
    elsif ((not $only_check_db) && (not $check_main_mtu{"tcp"}->{8080}->{open})) {
        $retries--;
        if ($retries < 0) {
            $retries = $retries_default;
            start_backup_mtu();
        }
    }

    push_state($state, $main_mtu_ip); # para saber si el watchdog está funcionando
    push_state($state, $this_mtu);    # para saber si el watchdog está funcionando
    $state++;
    if ($state > 256) {
        $state = 0;
    }
}
```

[ANEXO Nº 8]

[Perro guardián del enrutador (código fuente)]

```
#=============================================================================================================
# Alejandro González [2013]
# Universidad Central de Venezuela - Facultad de Ingeniería - Escuela de Ingeniería Eléctrica
# SCRIPT DEL ENRUTADOR
#=============================================================================================================
use strict;
use warnings;
use IO::Socket::PortState qw(check_ports);
use IO::Socket;
use LWP::UserAgent;

#=============================================================================================================
# CONFIGURACIÓN
#=============================================================================================================
my $timeout = 1; # en segundos
my $mtu01_ip = '192.168.2.136';
my $mtu02_ip = '192.168.2.137';
my $tick = 1 ; # en segundos
my $retries_default = 1; # cuantos reintentos
my $mtu01_sate = 0;
my $mtu02_sate = 1;
my $allways_push = 1; # útil para que Mango pueda determinar si el watchdog está funcionando.

#=============================================================================================================
# CAMPOS
#=============================================================================================================
my $current_mtu = $mtu01_ip;
my $backup_mtu = $mtu02_ip;
my $retries = $retries_default;
my $ua = LWP::UserAgent->new;

my %check_mysql = (
    tcp => {
        3306 => {
            name => 'MySQL',
        }
    }
);

my %check_tomcat = (
    tcp => {
        80 => {
            name => 'Tomcat',
        }
    }
);

my %check_all = (
    tcp => {
        80 => {
            name => 'Tomcat',
        },
        3306 => {
            name => 'MySQL',
        }
    }
);

#=============================================================================================================
# FUNCIONES
#=============================================================================================================

#-------------------------------------------------------------------------------
sub change_route {
    my ($ip) = @_;

    system('iptables -t nat -D PREROUTING 1');
    system('iptables -t nat -D PREROUTING 1');
```

```
    system('iptables -t nat -A PREROUTING -s 10.8.0.0/24   -p tcp --dport 80 -j DNAT --to-
destination '.$ip.':80');
    system('iptables -t nat -A PREROUTING -s 10.8.0.0/24   -p tcp --dport 8080 -j DNAT --to-
destination '.$ip.':8080');

    system('iptables -t nat -D POSTROUTING 1');
    system('iptables -t nat -A POSTROUTING -j MASQUERADE');
}

#-------------------------------------------------------------------------------
sub push_state {
    my ($state, $ip) = @_;

    check_ports($ip, $timeout, \%check_tomcat);

    if ($check_tomcat{"tcp"}->{80}->{open}) {

        mylog("Actualizando estado en $ip.");
        my $req = HTTP::Request->new(GET => "http://$ip/httpds?gatewaystate=$state");
        my $res = $ua->request($req);
    }
    else {
        mylog("Saltando $ip. No hay conectividad.");
    }
}

#-------------------------------------------------------------------------------
sub push_current_state {
    if ($current_mtu eq $mtu01_ip) {
        push_state($mtu01_sate, $mtu01_ip);
        push_state($mtu01_sate, $mtu02_ip);
    }
    else {
        push_state($mtu02_sate, $mtu01_ip);
        push_state($mtu02_sate, $mtu02_ip);
    }
}

#-------------------------------------------------------------------------------
sub get_formated_local_datetime {
    my ($second, $minute, $hour, $dayOfMonth, $month, $yearOffset) = localtime();
    my $year = 1900 + $yearOffset;
    return sprintf( "%0u/%02u/%02u %02u:%02u:%02u", $year, $month, $dayOfMonth, $hour, $minute,
$second);
}

#-------------------------------------------------------------------------------
sub mylog {
    my ($text) = @_;

    my $date = get_formated_local_datetime();
    print "[$date] ".$text."\n";
}

#-------------------------------------------------------------------------------
sub save_current_mtu {
    my ($ip) = @_;

    open LASTMTUFILE, '>', 'current_mtu';
    print LASTMTUFILE $ip;
    close LASTMTUFILE;
}

#-------------------------------------------------------------------------------
sub load_current_mtu {

    if (not -e 'current_mtu') {
        return;
    }

    open LASTMTUFILE, '<', 'current_mtu';

    my $line = <LASTMTUFILE>;

    if ($line =~ m/([\d.]*)$/m) {
        $current_mtu = $1;
    }
```

```
    close LASTMTUFILE;

    if ($current_mtu eq $mtu01_ip) {
        $backup_mtu = $mtu02_ip;
    }
    else {
        $backup_mtu = $mtu01_ip;
    }
}

#=============================================================================================================
# SCRIPT
#=============================================================================================================
load_current_mtu();

while (1) {
    sleep $tick;

    check_ports($current_mtu, $timeout, \%check_all);
    if ((not $check_all{"tcp"}->{3306}->{open}) || (not $check_all{"tcp"}->{80}->{open})) {
        $retries--;
        if ($retries < 0) {
            mylog("Base de datos o Mango M2M no disponible en '$current_mtu'.");

            check_ports($backup_mtu, $timeout, \%check_mysql);
            if ($check_mysql{"tcp"}->{3306}->{open}) {
                mylog("Cambiando servidor a '$backup_mtu'.");
                my $aux = $current_mtu;
                $current_mtu = $backup_mtu;
                $backup_mtu = $aux;
                change_route($current_mtu);
                save_current_mtu($current_mtu);
                push_current_state();
            }
            else {
                mylog("Servidor de respaldo no disponible ('$backup_mtu').");
                if ($allways_push) {
                    push_current_state();
                }
            }

            $retries = $retries_default;
        }
    }
    elsif ($allways_push) {
        push_current_state();
    }
}
```