

# Отказоустойчивое решение (кластер). Зеркальный вариант. Аутентификация на Preshared Key

## Описание стенда

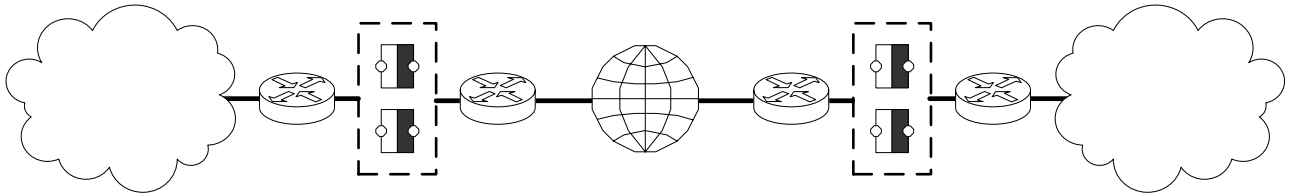


Рис. 1. Общая схема подключения

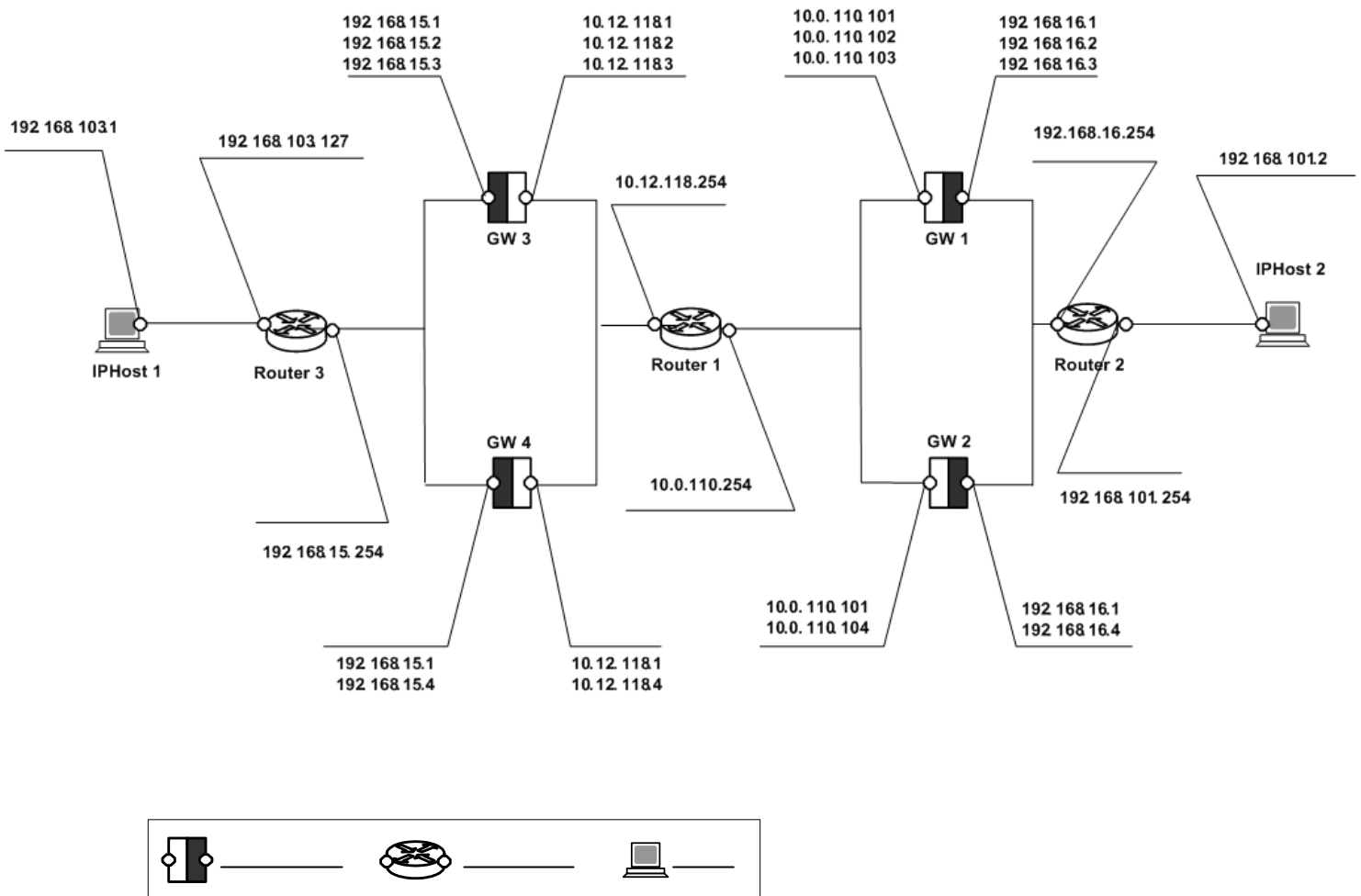


Рис. 2 Схема стенда.

В этом сценарии реализована возможность создания отказоустойчивого решения для двух подсетей на базе шлюзов безопасности CSP VPN Gate. В этом сценарии подсеть SN1 (192.168.103.0/24) защищена кластером, функции которого выполняют два шлюза безопасности GW1 и GW2, а вторая подсеть SN2 (192.168.101.0/24) – кластером из двух шлюзов безопасности GW3 и GW4. В схему включены три роутера: Router1, Router2 и Router3.

## Логика работы отказоустойчивого решения

В предложенном решении реализована следующая логика:

- В качестве кластера используются два шлюза безопасности CSP VPN Gate
- Шлюзы имеют различные роли: GW1 и GW3 – основные шлюзы безопасности, GW2 и GW4 – резервные шлюзы безопасности.
- Кроме шлюзов безопасности определены «надежные» устройства. На эти устройства будут отправляться ICMP-пакеты для диагностики работоспособности сетевых интерфейсов шлюзов безопасности.
- В нормальном режиме весь трафик обрабатывается на основных шлюзах безопасности. Резервные шлюзы безопасности в это время находятся в режиме ожидания.
- В случае выхода из строя любого основного шлюза безопасности его заменяет парный ему резервный шлюз безопасности и обрабатывает весь проходящий трафик.
- После восстановления работоспособности основного шлюза безопасности резервный шлюз переходит в режим ожидания и отдает управление основному шлюзу безопасности.
- Проверка работоспособности основного и резервного шлюзов безопасности, а также действия по активации и настройке интерфейсов производятся с помощью скриптов, устанавливаемых на шлюзы безопасности.

## Параметры защищенного соединения

Параметры защищенного соединения между подсетями SN1 и SN2:

- Аутентификация на Preshared Key.
- IKE parameters:
  - Encryption algorithm – GOST
  - Hash algorithm – GOST
  - DH-group – group2 (1024)
- IPSec parameters:
  - ESP encryption algorithm – GOST

## Основные шаги по настройке устройств стенда

Настройка устройств стенда:

- Сборку стенда и проверку соединений мы не описываем
- Конфигурирование каждого из шлюзов безопасности GW1-GW4 производится по следующему сценарию:
  - астройка адресов шлюза безопасности
  - Установка скриптов.
  - Настройка роутинга. Устанавливаем роутер по умолчанию
  - Создание политики безопасности, используя командную строку
- Настройка устройств Router1 – Router3

- Настройка устройств IPHost1 - IPHost2.

## Настройка шлюза безопасности GW1

### Настройка адресов шлюза безопасности

Для предотвращения конфликтов IP-адресов рекомендуется настроить адреса, которые будут назначаться интерфейсам при старте ОС. Для установленного продукта CSP VPN Gate версии 2.2 и выше это можно выполнить при помощи скрипта `/usr/sbin/ipsetup` (в ОС Linux - `/sbin/ipsetup`), входящего в состав продукта.

Для продукта CSP VPN Gate версии 2.1 такую настройку адресов можно выполнить в файле `/etc/hosts`.

Интерфейсу с именем `PHYS_1_1_NAME` должен быть назначен адрес `FIXED_1_1_PARAM`, а интерфейсу с именем `PHYS_1_2_NAME` – адрес `FIXED_1_2_PARAM`. Соответствие имен, адресов и конкретных значений смотрите в разделе ["Описание настраиваемых параметров основного скрипта устройства GW1"](#).

### Установка скриптов

На шлюз безопасности GW1 нужно установить два скрипта. Один скрипт будет отвечать за конфигурирование интерфейсов и оценку работоспособности шлюза, а второй скрипт будет отвечать за старт основного скрипта при перезагрузке операционной системы.

### Основной скрипт (`vpn_main_gate`)

Основной скрипт сохраняем в директории `/etc/init.d` под именем `vpn_main_gate` и выполняем для него

```
chmod a+x vpn_main_gate
```

Текст скрипта `vpn_main_gate`:

```
#!/bin/ksh

# Version 1.2

IP_RELIABLE1=10.0.110.254
IP_RELIABLE2=192.168.16.254

PHYS_1_1_NAME=iprb0
TUNNEL_1_PARAM=10.0.110.101/16

TO_CHECK_1_NAME=$PHYS_1_1_NAME:1
TO_CHECK_1_PARAM=10.0.110.102/16

FIXED_1_1_NAME=$PHYS_1_1_NAME:2
FIXED_1_1_PARAM=10.0.110.103/16

PHYS_1_2_NAME=iprb1

TUNNEL_2_PARAM=192.168.16.1/24
```

```
TO_CHECK_2_NAME=$PHYS_1_2_NAME:1
TO_CHECK_2_PARAM=192.168.16.2/24

FIXED_1_2_NAME=$PHYS_1_2_NAME:2
FIXED_1_2_PARAM=192.168.16.3/24

CHECK_N=3
RELAX_TIMEOUT=4

PING=/usr/sbin/ping

do_ping1()
{
    $PING $IP_RELIABLE1 1 > /dev/null
}

do_ping2()
{
    $PING $IP_RELIABLE2 1 > /dev/null
}

notify()
{
    logger -p local0.notice "$1"
    echo `date` "$1" > /dev/console
}

on_fail()
{
    ifconfig $PHYS_1_1_NAME down 2> /dev/null
    ifconfig $PHYS_1_2_NAME down 2> /dev/null
    ifconfig $TO_CHECK_1_NAME down 2> /dev/null
    ifconfig $TO_CHECK_2_NAME down 2> /dev/null
    ifconfig $FIXED_1_1_NAME $FIXED_1_1_PARAM broadcast + -deprecated 2>
/dev/null
    ifconfig $FIXED_1_2_NAME $FIXED_1_2_PARAM broadcast + -deprecated 2>
/dev/null
    /opt/VPNagent/bin/lsp_mgr reload
    notify 'Main gate is deactivated'
}

on_ok()
{
    ifconfig $TO_CHECK_1_NAME $TO_CHECK_1_PARAM broadcast + deprecated up 2>
/dev/null
```

```

    ifconfig $TO_CHECK_2_NAME $TO_CHECK_2_PARAM broadcast + deprecated up 2>
/dev/null

    sleep $RELAX_TIMEOUT

    ifconfig $PHYS_1_1_NAME $TUNNEL_1_PARAM broadcast + up 2> /dev/null
    ifconfig $PHYS_1_2_NAME $TUNNEL_2_PARAM broadcast + up 2> /dev/null
    ifconfig $FIXED_1_1_NAME deprecated 2> /dev/null
    ifconfig $FIXED_1_2_NAME deprecated 2> /dev/null

/opt/VPNagent/bin/lsp_mgr reload

    notify 'Main gate is activated'
}

ifconfig $TO_CHECK_1_NAME plumb 2> /dev/null
ifconfig $TO_CHECK_2_NAME plumb 2> /dev/null

ifconfig $FIXED_1_1_NAME plumb 2> /dev/null
ifconfig $FIXED_1_2_NAME plumb 2> /dev/null

ifconfig $PHYS_1_1_NAME $TUNNEL_1_PARAM down broadcast + 2> /dev/null
ifconfig $PHYS_1_2_NAME $TUNNEL_2_PARAM down broadcast + 2> /dev/null

ifconfig $FIXED_1_1_NAME $FIXED_1_1_PARAM up broadcast + 2> /dev/null
ifconfig $FIXED_1_2_NAME $FIXED_1_2_PARAM up broadcast + 2> /dev/null

if do_ping1 && do_ping2; then
    RETRY_COUNT1=$CHECK_N
    RETRY_COUNT2=$CHECK_N
    on_ok
else
    RETRY_COUNT1=0
    RETRY_COUNT2=0
    on_fail
fi

while true; do
    if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -gt 0 ]]; then
        if do_ping1; then
            RETRY_COUNT1=$CHECK_N
        else
            RETRY_COUNT1=$((RETRY_COUNT1-1))
            if [ $RETRY_COUNT1 -eq 0 ]; then
                on_fail
            fi
        fi
    fi

    if do_ping2; then

```

```

        RETRY_COUNT2=$CHECK_N
    else
        RETRY_COUNT2=$((RETRY_COUNT2-1))
        if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -eq 0 ]]; then
            on_fail
        fi
    fi

    if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -gt 0 ]]; then
        sleep 1
    fi

    else
        if do_ping1 && do_ping2; then
            RETRY_COUNT1=$CHECK_N
            RETRY_COUNT2=$CHECK_N
            on_ok
        fi
    fi
done

```

Запуск скрипта `vpn_main_gate` будет производиться с помощью вспомогательного скрипта при старте операционной системы. Запуск и остановку скрипта `vpn_main_gate` можно производить вручную, для чего следует выполнить:

```
/etc/init.d/vpn_main_gate_init start | stop
```

## Описание настраиваемых параметров основного скрипта устройства GW1

В основном скрипте настройте следующие параметры:

`IP_RELIABLE1` – IP-адрес надежного устройства из внешней подсети.

`IP_RELIABLE2` – IP-адрес надежного устройства из внутренней подсети.

`PHYS_1_1_NAME` – имя внешнего физического интерфейса.

`TUNNEL_1_PARAM` – параметры внешнего физического интерфейса: IP-адрес и длина префикса (количество единиц в сетевой маске).

`TO_CHECK_1_NAME` – имя внешнего виртуального интерфейса `Virtual (to check)`, который будет использоваться для определения работоспособности шлюза безопасности (на него будет отправлять ping резервный шлюз безопасности).

`TO_CHECK_1_PARAM` – параметры внешнего виртуального интерфейса: IP-адрес и длина префикса (количество единиц в сетевой маске).

`FIXED_1_1_NAME` – имя внешнего виртуального интерфейса `Virtual (fixed)`.

`FIXED_1_1_PARAM` – параметры внешнего виртуального интерфейса `Virtual (fixed)`: IP-адрес и длина префикса (количество единиц в сетевой маске).

`PHYS_1_2_NAME` – имя внутреннего физического интерфейса.

`TUNNEL_2_PARAM` – параметры внутреннего физического интерфейса: IP-адрес и длина префикса (количество единиц в сетевой маске).

TO\_CHECK\_2\_NAME – имя внутреннего виртуального интерфейса Virtual (to check), который будет использоваться для определения работоспособности шлюза безопасности (на него будет отправлять ring резервный шлюз безопасности).

TO\_CHECK\_2\_PARAM – параметры внутреннего виртуального интерфейса: IP-адрес и длина префикса (количество единиц в сетевой маске).

FIXED\_1\_2\_NAME – имя внутреннего виртуального интерфейса Virtual (fixed).

FIXED\_1\_2\_PARAM – параметры внутреннего виртуального интерфейса Virtual (fixed): IP-адрес и длина префикса.

CHECK\_N – количество неуспешных попыток ring, после которых считается, что связи нет. Чем меньше данное число, тем быстрее срабатывает переключение интерфейсов, но тем больше вероятность ложных срабатываний.

RELAX\_TIMEOUT – таймаут (в секундах) между поднятием виртуальных интерфейсов Virtual (to check) и поднятием туннельных интерфейсов iprb0 и iprb1. В этот таймаут резервный шлюз безопасности должен опустить туннельные интерфейсы iprb0 и iprb1.

notify – команды оповещения. Настройка описана ниже в разделе “Команды оповещения”.

## Команды оповещения

Еще одной настройкой основного скрипта является функция оповещения (notify).

Она позволяет задавать команды оповещения такие как вывод в системную консоль, syslog и т.п.

В основных скриптах эти команды пишутся в блоке:

```
notify()
{
# Команды оповещения.
}
```

В данную функцию передается строка сообщения для вывода. В командах эта строка задается как \$1.

Пример команды вывода на системную консоль. Вместе с сообщением выводится дата и время события:

```
echo `date` "$1" > /dev/console
```

Пример команды вывода в syslog:

```
logger -p local0.notice "$1"
```

**Примечание:** в аргументе –p задается syslog facility и severity генерируемого сообщения. Чтобы данное сообщение было получено, необходимо, чтобы syslog-daemon был соответствующим образом сконфигурирован в файле /etc/syslog.conf.

Пример вывода сообщений с facility local0 и severity notice в файл /var/adm/messages:

```
local0.notice /var/adm/vpnlog
```

Пример отправки сообщений на хост 192.168.101.2:

```
local0.notice @192.168.101.2
```

Пример функции вывода на системную консоль и в syslog:

```
notify()
{
logger -p local0.notice "$1"
echo `date` "$1" > /dev/console
}
```

## Вспомогательный скрипт (vpn\_main\_gate\_init)

Вспомогательный скрипт служит для обеспечения запуска и остановки основного скрипта.

Вспомогательный скрипт сохраняем в директории `/etc/init.d` под именем `vpn_main_gate_init` и выполняем для него

```
chmod a+x vpn_main_gate_init
```

Текст скрипта `vpn_main_gate_init`:

```
#!/bin/sh

# Version 1.2

VPN_MAIN_GATE_SCRIPT=/etc/init.d/vpn_main_gate
PS=/usr/bin/ps
GREP=/usr/bin/grep
KILL=/usr/bin/kill
AWK=/usr/bin/awk

case $1 in
'start')
    $VPN_MAIN_GATE_SCRIPT &
    ;;
'stop')
    PID=`$PS -ef | $GREP -v grep | $GREP $VPN_MAIN_GATE_SCRIPT | $AWK '{print $2}'`
    if [ ! -z "$PID" ] ; then
        /usr/bin/kill $PID 1> /dev/null 2>&1
    fi
    ;;
*)
    echo "Usage: vpn_main_gate_init { start | stop }"
    ;;

```

```
esac
```

Для того, чтобы скрипт `vpn_main_gate` вызывался при StartUp-е и останавливался при Shutdown-е, необходимо сделать линки на скрипт `vpn_main_gate_init`:

```
ln -s /etc/init.d/vpn_main_gate_init /etc/rc2.d/S22vpn_main_gate_init
ln -s /etc/init.d/vpn_main_gate_init /etc/rc0.d/K91vpn_main_gate_init
```

## Установка шлюза по умолчанию

После выполнения этих настроек необходимо установить шлюз по умолчанию. В качестве этого устройства следует выбрать адрес устройства Router1: 10.0.110.254.

## Создание политики GW1

Приступим к настройке CSP VPN Gate, установленного на шлюзе безопасности GW1.

Конфигурирование можно производить локально или удаленно, используя командную строку. Для этого перейдем в директорию `/opt/VPNagent/bin/` и запустим `cs_console`.

При помощи этого приложения необходимо набрать следующую конфигурацию:

```
crypto ipsec df-bit clear
crypto isakmp identity hostname
crypto isakmp keepalive 10 3
ip host cluster2 10.12.118.1
username cscons password csp
hostname GW1
enable password csp
!
crypto isakmp policy 1
  hash md5
  encryption des
  authentication pre-share
  group 2
!
crypto isakmp key 1234567890 hostname cluster2
!
crypto ipsec transform-set GOST esp-des
  mode tunnel
!
ip access-list extended CryptoACL
  permit ip 192.168.101.0 0.0.0.255 192.168.103.0 0.0.0.255
!
crypto map CMAP 1 ipsec-isakmp
  match address CryptoACL
  set transform-set GOST
  set pfs group2
  set peer 10.12.118.1
!
interface FastEthernet0/0
  ip address 10.0.110.101 255.255.0.0
  ip address 10.0.110.102 255.255.0.0 secondary
  ip address 10.0.110.103 255.255.0.0 secondary
  crypto map CMAP
!
crypto isakmp peer address 10.12.118.1
  set aggressive-mode client-endpoint ipv4-address 10.12.118.1
end
```

[Текст LSP конфигурации](#), которая получена в результате конвертирования cisco-like конфигурации, происходящего при выходе из конфигурационного режима `cs_console`, приведена в Приложении. Ее текст можно получить, используя утилиту `lsp_mgr show`, входящую в состав CSP VPN Gate.

## Настройка шлюза безопасности GW2

Шлюз безопасности GW2 в нашей схеме играет роль резервного шлюза. На нем также следует установить два скрипта – основной и вспомогательный.

### Настройка адресов шлюза безопасности

Для предотвращения конфликтов IP-адресов рекомендуется настроить адреса, которые будут назначаться интерфейсам при старте ОС. Для установленного продукта CSP VPN Gate версии 2.2 и выше это можно выполнить при помощи скрипта `/usr/sbin/ipsetup` (в ОС Linux - `/sbin/ipsetup`).

Для продукта CSP VPN Gate версии 2.1 такую настройку адресов можно выполнить в файле `/etc/hosts`.

Интерфейсу с именем `PHYS_2_1_NAME` должен быть назначен адрес `FIXED_2_1_PARAM`, а интерфейсу с именем `PHYS_2_2_NAME` – адрес `FIXED_2_2_PARAM`. Соответствие имен, адресов и конкретных значений смотрите в разделе [“Описание настраиваемых параметров основного скрипта устройства GW2”](#).

## Установка скриптов

### Основной скрипт устройства GW2 (`vpn_reserve_gate`)

Основной скрипт надо положить в директорию `/etc/init.d` под именем `vpn_reserve_gate` и выполнить для него

```
chmod a+x vpn_reserve_gate
```

Текст скрипта `vpn_reserve_gate`:

```
#!/bin/ksh
# Version 1.2

IP_RELIABLE1=10.0.110.254
IP_RELIABLE2=192.168.16.254

IP_TO_CHECK1=10.0.110.102
IP_TO_CHECK2=192.168.16.1

PHYS_2_1_NAME=iprb0

TUNNEL_1_PARAM=10.0.110.101/16

FIXED_2_1_NAME=${PHYS_2_1_NAME}:1
FIXED_2_1_PARAM=10.0.110.104/16

PHYS_2_2_NAME=iprb1

TUNNEL_2_PARAM=192.168.16.1/24
```

```
FIXED_2_2_NAME=$PHYS_2_2_NAME:1
FIXED_2_2_PARAM=192.168.16.4/24

CHECK_N=3

PING=/usr/sbin/ping

notify()
{
    logger -p local0.notice "$1"
    echo `date` "$1" > /dev/console
}

do_ping1()
{
    $PING $IP_RELIABLE1 1 > /dev/null
}

do_ping2()
{
    $PING $IP_RELIABLE2 1 > /dev/null
}

do_ping_g1()
{
    $PING $IP_TO_CHECK1 1 > /dev/null
}

do_ping_g2()
{
    $PING $IP_TO_CHECK2 1 > /dev/null
}

ifs_down()
{
    ifconfig $FIXED_2_1_NAME -deprecated 2> /dev/null
    ifconfig $FIXED_2_2_NAME -deprecated 2> /dev/null
    ifconfig $PHYS_2_1_NAME down 2> /dev/null
    ifconfig $PHYS_2_2_NAME down 2> /dev/null
    /opt/VPNagent/bin/lsp_mgr reload
    notify 'Reserve gate is deactivated'
}

ifs_up()
```

```
{
    ifconfig $PHYS_2_1_NAME $TUNNEL_1_PARAM broadcast + up 2> /dev/null
    ifconfig $PHYS_2_2_NAME $TUNNEL_2_PARAM broadcast + up 2> /dev/null
    ifconfig $FIXED_2_1_NAME deprecated 2> /dev/null
    ifconfig $FIXED_2_2_NAME deprecated 2> /dev/null
    /opt/VPNagent/bin/lsp_mgr reload
    notify 'Reserve gate is activated'
}

IFS_UP_FLAG=0

ifconfig $FIXED_2_1_NAME plumb 2> /dev/null
ifconfig $FIXED_2_1_NAME $FIXED_2_1_PARAM up broadcast + 2> /dev/null

ifconfig $FIXED_2_2_NAME plumb 2> /dev/null
ifconfig $FIXED_2_2_NAME $FIXED_2_2_PARAM up broadcast + 2> /dev/null

ifconfig $PHYS_2_1_NAME $TUNNEL_1_PARAM down broadcast + 2> /dev/null
ifconfig $PHYS_2_2_NAME $TUNNEL_2_PARAM down broadcast + 2> /dev/null

if do_ping_g1 && do_ping_g2; then
    RETRY_COUNT_G1=$CHECK_N
    RETRY_COUNT_G2=$CHECK_N
    RETRY_COUNT1=$CHECK_N
    RETRY_COUNT2=$CHECK_N
    ifs_down
else
    RETRY_COUNT_G1=0
    RETRY_COUNT_G2=0

    if do_ping1 && do_ping2; then
        RETRY_COUNT1=$CHECK_N
        RETRY_COUNT2=$CHECK_N
        ifs_up
        IFS_UP_FLAG=1
    else
        RETRY_COUNT1=0
        RETRY_COUNT2=0
        ifs_down
    fi
fi

while true; do
    CHECK_CONTINUE=0
```

```

if [[ $RETRY_COUNT_G1 -gt 0 ]] && [[ $RETRY_COUNT_G2 -gt 0 ]]; then
    if do_ping_g1; then
        RETRY_COUNT_G1=$CHECK_N
        RETRY_COUNT1=$CHECK_N
    else
        RETRY_COUNT_G1=$((RETRY_COUNT_G1-1))
    fi

    if do_ping_g2; then
        RETRY_COUNT_G2=$CHECK_N
        RETRY_COUNT2=$CHECK_N
    else
        RETRY_COUNT_G2=$((RETRY_COUNT_G2-1))
    fi

    if [[ $RETRY_COUNT_G1 -gt 0 ]] && [[ $RETRY_COUNT_G2 -gt 0 ]]; then
        sleep 1
    fi

else
    if do_ping_g1 && do_ping_g2; then
        RETRY_COUNT_G1=$CHECK_N
        RETRY_COUNT_G2=$CHECK_N
        RETRY_COUNT1=$CHECK_N
        RETRY_COUNT2=$CHECK_N
        ifs_down
        IFS_UP_FLAG=0
    fi
fi

if [[ $RETRY_COUNT_G1 -eq 0 ]] || [[ $RETRY_COUNT_G2 -eq 0 ]]; then
    if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -gt 0 ]]; then
        if do_ping1; then
            RETRY_COUNT1=$CHECK_N
        else
            RETRY_COUNT1=$((RETRY_COUNT1-1))
            if [ $RETRY_COUNT1 -eq 0 ]; then
                ifs_down
                IFS_UP_FLAG=0
            fi
        fi
    fi

    if do_ping2; then
        RETRY_COUNT2=$CHECK_N

```

```

else
    RETRY_COUNT2=$((RETRY_COUNT2-1))
    if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -eq 0 ]];
then
    ifs_down
    IFS_UP_FLAG=0
    fi
fi

if [[ $RETRY_COUNT1 -eq $CHECK_N ]] && [[ $RETRY_COUNT2 -eq
$CHECK_N ]] && [[ $IFS_UP_FLAG -eq 0 ]]; then
    ifs_up
    IFS_UP_FLAG=1
    fi

if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -gt 0 ]]; then
    sleep 1
fi

else
    if do_ping1 && do_ping2 && [[ $IFS_UP_FLAG -eq 0 ]]; then
        RETRY_COUNT1=$CHECK_N
        RETRY_COUNT2=$CHECK_N
        ifs_up
        IFS_UP_FLAG=1
    fi
fi
fi
done

```

## Описание настраиваемых параметров основного скрипта устройства GW2

В основном скрипте настройте следующие параметры:

IP\_RELIABLE1 – IP адрес надежного устройства из внешней подсети.

IP\_RELIABLE2 – IP адрес надежного устройства из внутренней подсети.

IP\_TO\_CHECK1 – адрес внешнего интерфейса основного шлюза безопасности, по которому производится проверка его работоспособности.

IP\_TO\_CHECK2 – адрес внутреннего интерфейса основного шлюза безопасности, по которому производится проверка его работоспособности.

PHYS\_2\_1\_NAME – имя внешнего физического интерфейса резервного шлюза безопасности, на котором установлен туннельный адрес (общий для обоих шлюзов безопасности).

PHYS\_2\_2\_NAME – имя внутреннего физического интерфейса резервного шлюза безопасности, на котором установлен туннельный адрес (общий для обоих шлюзов безопасности).

TUNNEL\_1\_PARAM – параметры внешнего физического интерфейса резервного шлюза безопасности: IP-адрес и длина префикса (количество единиц в сетевой маске).

TUNNEL\_2\_PARAM – параметры внутреннего физического интерфейса резервного шлюза безопасности: IP-адрес и длина префикса (количество единиц в сетевой маске).

FIXED\_2\_1\_NAME – имя внешнего виртуального интерфейса, с которого осуществляется ping для проверки работоспособности основного шлюза безопасности.

FIXED\_2\_2\_NAME – имя внутреннего виртуального интерфейса, с которого осуществляется ping для проверки работоспособности основного шлюза безопасности.

FIXED\_2\_1\_PARAM – параметры внешнего виртуального интерфейса: IP-адрес и длина префикса (количество единиц в сетевой маске).

FIXED\_2\_2\_PARAM – параметры внутреннего виртуального интерфейса: IP-адрес и длина префикса (количество единиц в сетевой маске).

CHECK\_N – количество неуспешных попыток ping, после которых считается, что связи нет и основной шлюз безопасности вышел из строя. Чем меньше данное число, тем быстрее срабатывает переключение интерфейсов, но тем больше вероятность ложных срабатываний.

notify – команды оповещения.

## Команды оповещения

Настройка команд оповещения производится точно так же, как и для основного скрипта шлюза безопасности GW1.

## Вспомогательный скрипт устройства GW2 (vpn\_reserve\_gate\_init)

Вспомогательный скрипт служит для обеспечения запуска и остановки основного скрипта.

Вспомогательный скрипт сохраняем в директории `/etc/init.d` под именем **vpn\_reserve\_gate\_init** и выполняем для него

```
chmod a+x vpn_reserve_gate_init
```

Текст скрипта `vpn_reserve_gate_init`:

```
#!/bin/sh

# Version 1.2

VPN_RESERVE_GATE_SCRIPT=/etc/init.d/vpn_reserve_gate
PS=/usr/bin/ps
GREP=/usr/bin/grep
KILL=/usr/bin/kill
AWK=/usr/bin/awk

case $1 in
'start')
    $VPN_RESERVE_GATE_SCRIPT &
    ;;
'stop')
```

```

    PID=`$PS -ef | $GREP -v grep | $GREP $VPN_RESERVE_GATE_SCRIPT | $AWK
    '{print $2}'`
    if [ ! -z "$PID" ] ; then
        /usr/bin/kill $PID 1> /dev/null 2>&1
    fi
    ;;
*)
    echo "Usage: vpn_reserve_gate_init { start | stop }"
    ;;
esac

```

Для того, чтобы скрипт `vpn_reserve_gate` вызывался при `StartUp`-е и останавливался при `Shutdown`-е необходимо сделать линки на скрипт `vpn_reserve_gate_init`:

```

ln -s /etc/init.d/vpn_reserve_gate_init /etc/rc2.d/S22vpn_reserve_gate_init
ln -s /etc/init.d/vpn_reserve_gate_init /etc/rc0.d/K91vpn_reserve_gate_init

```

## Установка шлюза по умолчанию

После выполнения этих настроек необходимо установить шлюз по умолчанию. В качестве этого устройства следует выбрать адрес устройства `Router1` - `10.0.110.254`.

## Создание политики GW2

Настройки CSP VPN Gate устройства `GW2` ничем не отличаются от настроек устройства `GW1` и выполняются аналогично. Конфигурация этого устройства:

```

crypto ipsec df-bit clear
crypto isakmp identity hostname
crypto isakmp keepalive 10 3
ip host cluster2 10.12.118.1
username cscons password csp
hostname GW2
enable password csp
!
logging trap debugging
!
crypto isakmp policy 1
    hash md5
    encryption des
    authentication pre-share
    group 2
!
crypto isakmp key 1234567890 hostname cluster2
!
crypto ipsec transform-set GOST esp-des
    mode tunnel
!
ip access-list extended CryptoACL

```

```
permit ip 192.168.101.0 0.0.0.255 192.168.103.0 0.0.0.255
!
crypto map CMAP 1 ipsec-isakmp
match address CryptoACL
set transform-set GOST
set pfs group2
set peer 10.12.118.1
!
interface FastEthernet0/0
ip address 10.0.110.101 255.255.0.0
ip address 10.0.110.104 255.255.0.0 secondary
crypto map CMAP
interface FastEthernet0/1
ip address 192.168.16.1 255.255.255.0
ip address 192.168.16.4 255.255.255.0 secondary
!
crypto isakmp peer address 10.12.118.1
set aggressive-mode client-endpoint ipv4-address 10.12.118.1
!
end
```

[Текст LSP конфигурации](#), которая получена в результате конвертирования cisco-like конфигурации, приведена в Приложении. Ее текст можно получить, используя утилиту **lsp\_mgr show**, входящую в состав CSP VPN Gate.

## Настройка шлюза безопасности GW3

Шлюз безопасности GW3 в нашей схеме играет роль основного шлюза. На нем также следует установить два скрипта – основной и вспомогательный.

### Установка скриптов

#### Основной скрипт устройства GW3 (vpn\_main\_gate)

Основной скрипт надо положить в директорию */etc/init.d* под именем **vpn\_main\_gate** и выполнить для него

```
chmod a+x vpn_main_gate
```

Текст скрипта **vpn\_main\_gate**:

```
#!/bin/ksh

# Version 1.2

IP_RELIABLE1=10.12.118.254
IP_RELIABLE2=192.168.15.254

PHYS_1_1_NAME=iprb0
```

```
TUNNEL_1_PARAM=10.12.118.1/24

TO_CHECK_1_NAME=$PHYS_1_1_NAME:1
TO_CHECK_1_PARAM=10.12.118.2/24

FIXED_1_1_NAME=$PHYS_1_1_NAME:2
FIXED_1_1_PARAM=10.12.118.3/24

PHYS_1_2_NAME=iprb1

TUNNEL_2_PARAM=192.168.15.1/24

TO_CHECK_2_NAME=$PHYS_1_2_NAME:1
TO_CHECK_2_PARAM=192.168.15.2/24

FIXED_1_2_NAME=$PHYS_1_2_NAME:2
FIXED_1_2_PARAM=192.168.15.3/24

CHECK_N=3
RELAX_TIMEOUT=4

PING=/usr/sbin/ping

do_ping1()
{
    $PING $IP_RELIABLE1 1 > /dev/null
}

do_ping2()
{
    $PING $IP_RELIABLE2 1 > /dev/null
}

notify()
{
    logger -p local0.notice "$1"
    echo `date` "$1" > /dev/console
}

on_fail()
{
    ifconfig $PHYS_1_1_NAME down 2> /dev/null
    ifconfig $PHYS_1_2_NAME down 2> /dev/null
    ifconfig $TO_CHECK_1_NAME down 2> /dev/null
    ifconfig $TO_CHECK_2_NAME down 2> /dev/null
}
```

```

    ifconfig $FIXED_1_1_NAME $FIXED_1_1_PARAM broadcast + -deprecated 2>
/dev/null

    ifconfig $FIXED_1_2_NAME $FIXED_1_2_PARAM broadcast + -deprecated 2>
/dev/null

    /opt/VPNagent/bin/lsp_mgr reload

    notify 'Main gate is deactivated'
}

on_ok()
{
    ifconfig $TO_CHECK_1_NAME $TO_CHECK_1_PARAM broadcast + deprecated up 2>
/dev/null

    ifconfig $TO_CHECK_2_NAME $TO_CHECK_2_PARAM broadcast + deprecated up 2>
/dev/null

    sleep $RELAX_TIMEOUT

    ifconfig $PHYS_1_1_NAME $TUNNEL_1_PARAM broadcast + up 2> /dev/null
    ifconfig $PHYS_1_2_NAME $TUNNEL_2_PARAM broadcast + up 2> /dev/null
    ifconfig $FIXED_1_1_NAME deprecated 2> /dev/null
    ifconfig $FIXED_1_2_NAME deprecated 2> /dev/null

    /opt/VPNagent/bin/lsp_mgr reload

    notify 'Main gate is activated'
}

ifconfig $TO_CHECK_1_NAME plumb 2> /dev/null
ifconfig $TO_CHECK_2_NAME plumb 2> /dev/null

ifconfig $FIXED_1_1_NAME plumb 2> /dev/null
ifconfig $FIXED_1_2_NAME plumb 2> /dev/null

ifconfig $PHYS_1_1_NAME $TUNNEL_1_PARAM down broadcast + 2> /dev/null
ifconfig $PHYS_1_2_NAME $TUNNEL_2_PARAM down broadcast + 2> /dev/null

ifconfig $FIXED_1_1_NAME $FIXED_1_1_PARAM up broadcast + 2> /dev/null
ifconfig $FIXED_1_2_NAME $FIXED_1_2_PARAM up broadcast + 2> /dev/null

if do_ping1 && do_ping2; then
    RETRY_COUNT1=$CHECK_N
    RETRY_COUNT2=$CHECK_N
    on_ok
else
    RETRY_COUNT1=0
    RETRY_COUNT2=0
    on_fail
fi

while true; do

```

```

if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -gt 0 ]]; then
    if do_ping1; then
        RETRY_COUNT1=$CHECK_N
    else
        RETRY_COUNT1=$((RETRY_COUNT1-1))
        if [ $RETRY_COUNT1 -eq 0 ]; then
            on_fail
        fi
    fi
fi

if do_ping2; then
    RETRY_COUNT2=$CHECK_N
else
    RETRY_COUNT2=$((RETRY_COUNT2-1))
    if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -eq 0 ]]; then
        on_fail
    fi
fi

if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -gt 0 ]]; then
    sleep 1
fi

else
    if do_ping1 && do_ping2; then
        RETRY_COUNT1=$CHECK_N
        RETRY_COUNT2=$CHECK_N
        on_ok
    fi
fi

done

```

### Вспомогательный скрипт устройства GW3 (vpn\_main\_gate\_init)

Вспомогательный скрипт сохраняем в директории `/etc/init.d` под именем `vpn_main_gate_init` и выполняем для него

```
chmod a+x vpn_main_gate_init
```

Текст скрипта `vpn_main_gate_init`:

```

#!/bin/sh

# Version 1.2

VPN_MAIN_GATE_SCRIPT=/etc/init.d/vpn_main_gate
PS=/usr/bin/ps
GREP=/usr/bin/grep

```

```

KILL=/usr/bin/kill
AWK=/usr/bin/awk

case $1 in
'start')
    $VPN_MAIN_GATE_SCRIPT &
    ;;
'stop')
    PID=`$PS -ef | $GREP -v grep | $GREP $VPN_MAIN_GATE_SCRIPT | $AWK '{print $2}'`
    if [ ! -z "$PID" ] ; then
        /usr/bin/kill $PID 1> /dev/null 2>&1
    fi
    ;;
*)
    echo "Usage: vpn_main_gate_init { start | stop }"
    ;;
esac

```

## Установка шлюза по умолчанию

На это устройство скрипты не устанавливаются, нужно только указать шлюз по умолчанию - 10.12.118.254.

## Создание политики GW3

Для этого необходимо перейти в директорию `/opt/VPNagent/bin` и запустить `cs_console`.

В консоли нужно набрать конфигурацию:

```

crypto ipsec df-bit clear
crypto isakmp identity hostname
crypto isakmp keepalive 10 3
ip host cluster1 10.0.110.101
username cscons password csp
hostname GW3
enable password csp
!
logging trap debugging
!
crypto isakmp policy 1
    hash md5
    encryption des
    authentication pre-share
    group 2
!
crypto isakmp key 1234567890 hostname cluster1
!
crypto ipsec transform-set GOST esp-des

```

```
mode tunnel
!
ip access-list extended cryptoACL
 permit ip 192.168.103.0 0.0.0.255 192.168.101.0 0.0.0.255
!
crypto map CMAP 1 ipsec-isakmp
 match address cryptoACL
 set transform-set GOST
 set pfs group2
 set peer 10.0.110.101
!
interface FastEthernet0/0
 ip address 10.12.118.1 255.255.255.0
 ip address 10.12.118.2 255.255.255.0 secondary
 ip address 10.12.118.3 255.255.255.0 secondary
 crypto map CMAP
interface FastEthernet0/1
 ip address 192.168.15.1 255.255.255.0
 ip address 192.168.15.2 255.255.255.0 secondary
 ip address 192.168.15.3 255.255.255.0 secondary
!
crypto isakmp peer address 10.0.110.101
 set aggressive-mode client-endpoint ipv4-address 10.0.110.101

end
```

[Текст LSP конфигурации](#), которая получена в результате конвертирования cisco-like конфигурации, приведена в Приложении. Ее текст можно получить, используя утилиту `lsp_mgr show`, входящую в состав CSP VPN Gate.

## Настройка шлюза безопасности GW4

### Установка скриптов

#### Основной скрипт устройства GW4 (`vpn_reserve_gate`)

```
#!/bin/ksh
# Version 1.2

IP_RELIABLE1=10.12.118.254
IP_RELIABLE2=192.168.15.254

IP_TO_CHECK1=10.12.118.2
IP_TO_CHECK2=192.168.15.2

PHYS_2_1_NAME=iprb0
```

```
TUNNEL_1_PARAM=10.12.118.1/24

FIXED_2_1_NAME=$PHYS_2_1_NAME:1
FIXED_2_1_PARAM=10.12.118.4/24

PHYS_2_2_NAME=iprb1

TUNNEL_2_PARAM=192.168.15.1/24

FIXED_2_2_NAME=$PHYS_2_2_NAME:1
FIXED_2_2_PARAM=192.168.15.4/24

CHECK_N=3

PING=/usr/sbin/ping

notify()
{
    logger -p local0.notice "$1"
    echo `date` "$1" > /dev/console
}

do_ping1()
{
    $PING $IP_RELIABLE1 1 > /dev/null
}

do_ping2()
{
    $PING $IP_RELIABLE2 1 > /dev/null
}

do_ping_g1()
{
    $PING $IP_TO_CHECK1 1 > /dev/null
}

do_ping_g2()
{
    $PING $IP_TO_CHECK2 1 > /dev/null
}

ifs_down()
{
```

```

    ifconfig $FIXED_2_1_NAME -deprecated 2> /dev/null
    ifconfig $FIXED_2_2_NAME -deprecated 2> /dev/null
    ifconfig $PHYS_2_1_NAME down 2> /dev/null
    ifconfig $PHYS_2_2_NAME down 2> /dev/null
/opt/VPNagent/bin/lsp_mgr reload
    notify 'Reserve gate is deactivated'
}

ifs_up()
{
    ifconfig $PHYS_2_1_NAME $TUNNEL_1_PARAM broadcast + up 2> /dev/null
    ifconfig $PHYS_2_2_NAME $TUNNEL_2_PARAM broadcast + up 2> /dev/null
    ifconfig $FIXED_2_1_NAME deprecated 2> /dev/null
    ifconfig $FIXED_2_2_NAME deprecated 2> /dev/null
/opt/VPNagent/bin/lsp_mgr reload
    notify 'Reserve gate is activated'
}

IFS_UP_FLAG=0

ifconfig $FIXED_2_1_NAME plumb 2> /dev/null
ifconfig $FIXED_2_1_NAME $FIXED_2_1_PARAM up broadcast + 2> /dev/null

ifconfig $FIXED_2_2_NAME plumb 2> /dev/null
ifconfig $FIXED_2_2_NAME $FIXED_2_2_PARAM up broadcast + 2> /dev/null

ifconfig $PHYS_2_1_NAME $TUNNEL_1_PARAM down broadcast + 2> /dev/null
ifconfig $PHYS_2_2_NAME $TUNNEL_2_PARAM down broadcast + 2> /dev/null

if do_ping_g1 && do_ping_g2; then
    RETRY_COUNT_G1=$CHECK_N
    RETRY_COUNT_G2=$CHECK_N
    RETRY_COUNT1=$CHECK_N
    RETRY_COUNT2=$CHECK_N
    ifs_down
else
    RETRY_COUNT_G1=0
    RETRY_COUNT_G2=0

    if do_ping1 && do_ping2; then
        RETRY_COUNT1=$CHECK_N
        RETRY_COUNT2=$CHECK_N
        ifs_up
        IFS_UP_FLAG=1

```

```
else
    RETRY_COUNT1=0
    RETRY_COUNT2=0
    ifs_down
fi
fi

while true; do
    CHECK_CONTINUE=0

    if [[ $RETRY_COUNT_G1 -gt 0 ]] && [[ $RETRY_COUNT_G2 -gt 0 ]]; then
        if do_ping_g1; then
            RETRY_COUNT_G1=$CHECK_N
            RETRY_COUNT1=$CHECK_N
        else
            RETRY_COUNT_G1=$(( $RETRY_COUNT_G1-1 ))
        fi

        if do_ping_g2; then
            RETRY_COUNT_G2=$CHECK_N
            RETRY_COUNT2=$CHECK_N
        else
            RETRY_COUNT_G2=$(( $RETRY_COUNT_G2-1 ))
        fi

        if [[ $RETRY_COUNT_G1 -gt 0 ]] && [[ $RETRY_COUNT_G2 -gt 0 ]]; then
            sleep 1
        fi
    else
        if do_ping_g1 && do_ping_g2; then
            RETRY_COUNT_G1=$CHECK_N
            RETRY_COUNT_G2=$CHECK_N
            RETRY_COUNT1=$CHECK_N
            RETRY_COUNT2=$CHECK_N
            ifs_down
            IFS_UP_FLAG=0
        fi
    fi

    if [[ $RETRY_COUNT_G1 -eq 0 ]] || [[ $RETRY_COUNT_G2 -eq 0 ]]; then
        if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -gt 0 ]]; then
            if do_ping1; then
                RETRY_COUNT1=$CHECK_N
            else
```

```

        RETRY_COUNT1=$((RETRY_COUNT1-1))
        if [ $RETRY_COUNT1 -eq 0 ]; then
            ifs_down
            IFS_UP_FLAG=0
        fi
    fi

    if do_ping2; then
        RETRY_COUNT2=$CHECK_N
    else
        RETRY_COUNT2=$((RETRY_COUNT2-1))
        if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -eq 0 ]];
then
            ifs_down
            IFS_UP_FLAG=0
        fi
    fi

    if [[ $RETRY_COUNT1 -eq $CHECK_N ]] && [[ $RETRY_COUNT2 -eq
$CHECK_N ]] && [[ $IFS_UP_FLAG -eq 0 ]]; then
        ifs_up
        IFS_UP_FLAG=1
    fi

    if [[ $RETRY_COUNT1 -gt 0 ]] && [[ $RETRY_COUNT2 -gt 0 ]]; then
        sleep 1
    fi

else
    if do_ping1 && do_ping2 && [[ $IFS_UP_FLAG -eq 0 ]]; then
        RETRY_COUNT1=$CHECK_N
        RETRY_COUNT2=$CHECK_N
        ifs_up
        IFS_UP_FLAG=1
    fi
fi
fi
done

```

### Вспомогательный скрипт устройства GW4 (vpn\_reserve\_gate\_init)

```

#!/bin/sh

# Version 1.2

VPN_RESERVE_GATE_SCRIPT=/etc/init.d/vpn_reserve_gate

```

```

PS=/usr/bin/ps
GREP=/usr/bin/grep
KILL=/usr/bin/kill
AWK=/usr/bin/awk

case $1 in
'start')
    $VPN_RESERVE_GATE_SCRIPT &
    ;;
'stop')
    PID=`$PS -ef | $GREP -v grep | $GREP $VPN_RESERVE_GATE_SCRIPT | $AWK
'{print $2}'`
    if [ ! -z "$PID" ] ; then
        /usr/bin/kill $PID 1> /dev/null 2>&1
    fi
    ;;
*)
    echo "Usage: vpn_reserve_gate_init { start | stop }"
    ;;
esac

```

## Установка шлюза по умолчанию

На это устройство скрипты не устанавливаются, нужно только указать шлюз по умолчанию - 10.12.118.254.

## Создание политики GW4

Для этого необходимо перейти в директорию `/opt/VPNagent/bin` и запустить `cs_console`.

В консоли нужно набрать конфигурацию:

```

crypto ipsec df-bit clear
crypto isakmp identity hostname
crypto isakmp keepalive 10 3
ip host cluster1 10.0.110.101
username cscons password csp
hostname GW4
enable password csp
!
logging trap debugging
!
!
crypto isakmp policy 1
    hash md5
    encryption des
    authentication pre-share
    group 2

```

```
!  
crypto isakmp key 1234567890 hostname cluster1  
!  
crypto ipsec transform-set GOST esp-des  
mode tunnel  
!  
ip access-list extended cryptoACL  
permit ip 192.168.103.0 0.0.0.255 192.168.101.0 0.0.0.255  
!  
crypto map CMAP 1 ipsec-isakmp  
match address cryptoACL  
set transform-set GOST  
set pfs group2  
set peer 10.0.110.101  
!  
interface FastEthernet0/0  
ip address 10.12.118.1 255.255.255.0  
ip address 10.12.118.4 255.255.255.0 secondary  
crypto map CMAP  
interface FastEthernet0/1  
ip address 192.168.15.1 255.255.255.0  
ip address 192.168.15.4 255.255.255.0 secondary  
!  
crypto isakmp peer address 10.0.110.101  
set aggressive-mode client-endpoint ipv4-address 10.0.110.101  
end
```

[Текст LSP конфигурации](#), которая получена в результате конвертирования cisco-like конфигурации, приведена в Приложении. Ее текст можно получить, используя утилиту `lsp_mgr show`, входящую в состав CSP VPN Gate.

## Настройка устройства Router1

Дополнительных настроек для этого устройства не требуется.

## Настройка устройства Router2

Настройка этого устройства сводится к установке в качестве шлюза по умолчанию «общего» адреса устройств GW1 и GW2 в этом сегменте сети - 192.168.16.1.

## Настройка устройства Router3

Настройка этого устройства сводится к установке в качестве шлюза по умолчанию «общего» адреса устройств GW3 и GW4 в этом сегменте сети - 192.168.15.1.

## Настройка устройства IPHost1

Настройка этого устройства сводится к установке в качестве шлюза по умолчанию «внутреннего» адреса устройства Router3: 192.168.103.127.

## Настройка устройства IPHost2

Настройка этого устройства сводится к установке в качестве шлюза по умолчанию «внутреннего» адреса устройства Router2: 192.168.101.254.

## Приложение

### LSP устройства GW1

```
# This is automatically generated LSP
# Conversion Date/Time:   Wed May  3 18:36:28 2006

GlobalParameters(
    Title                = "This LSP was automatically generated by
CSP Converter at Wed May  3 18:36:28 2006"
    Version              = "2.1"
    LDAPLogMessageLevel = INFO
    SystemLogMessageLevel = INFO
    PolicyLogMessageLevel = INFO
    CertificatesLogMessageLevel = INFO
)

SyslogSettings(
    Server = 127.0.0.1
    Facility = LOG_LOCAL7
)

IKETransform IKETransform_1(
    CipherAlg   *= "G2814789CPR01-K256-CBC-65534"
    HashAlg     *= "GR341194CPR01-65534"
    GroupID     *= MODP_1024
    LifetimeSeconds = 86400
)

ESPProposal ESP_GOST(
    Transform* = ESPTransform(
        CipherAlg*      = "G2814789CPR01-K256-CBC-254"
        LifetimeSeconds = 3600
        LifetimeKilobytes = 4608000
    )
)

AuthMethodPreshared IKE_auth_cs_key_cluster2(
    LocalID = IdentityEntry( KeyID *= "73746e6467617465" )
    RemoteID = IdentityEntry(
```

```

        IPv4Address *= 10.12.118.1
        KeyID *= "636c757374657232"
    )
    SharedIKESecret = "cs_key_cluster2"
)

IKERule IKE_CMAP_1(
    Transform* = IKETransform_1
    AggrModeAuthMethod *= IKE_auth_cs_key_cluster2
    MainModeAuthMethod *= IKE_auth_cs_key_cluster2
    AggrModePriority = TRUE
    DoAutopass = TRUE
    DPDIdleDuration = 10
    DPDResponseDuration = 3
    DPDRetries = 5
)

IPsecAction CMAP_1(
    TunnelingParameters *= TunnelEntry(
        PeerIPAddress = 10.12.118.1
        DFHandling=CLEAR
    )
    ContainedProposals *= ( ESP_GOST )
    GroupID *= MODP_1024
    IKERule = IKE_CMAP_1
)

FilteringRule Filter_nil_acl_CMAP_1(
    LocalIPFilter *= FilterEntry( IPAddress *= 192.168.101.0/24 )
    PeerIPFilter *= FilterEntry( IPAddress *= 192.168.103.0/24 )
    NetworkInterfaces *= "iprb0"
    Action *= ( CMAP_1 )
)

FilteringRule Filter_nil_acl(
    LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    PeerIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    NetworkInterfaces *= "iprb0"
    Action *= ( PASS )
)

```

## LSP устройства GW2

```

# This is automatically generated LSP
# Conversion Date/Time: Wed May 3 17:46:50 2006

GlobalParameters(
    Title = "This LSP was automatically generated by
    CSP Converter at Wed May 3 17:46:50 2006"
)

```

```
Version = "2.1"
LDAPLogMessageLevel = DEBUG
SystemLogMessageLevel = DEBUG
PolicyLogMessageLevel = DEBUG
CertificatesLogMessageLevel = DEBUG
)
SyslogSettings(
  Server = 127.0.0.1
  Facility = LOG_LOCAL7
)

IKETransform IKETransform_1(
  CipherAlg *= "G2814789CPR01-K256-CBC-65534"
  HashAlg *= "GR341194CPR01-65534"
  GroupID *= MODP_1024
  LifetimeSeconds = 86400
)

ESPProposal ESP_GOST(
  Transform* = ESPTransform(
    CipherAlg* = "G2814789CPR01-K256-CBC-254"
    LifetimeSeconds = 3600
    LifetimeKilobytes = 4608000
  )
)

AuthMethodPreshared IKE_auth_cs_key_cluster2(
  LocalID = IdentityEntry( KeyID *= "73746e6463737067617465" )
  RemoteID = IdentityEntry(
    IPv4Address *= 10.12.118.1
    KeyID *= "636c757374657232"
  )
  SharedIKESecret = "cs_key_cluster2"
)

IKERule IKE_CMAP_1(
  Transform* = IKETransform_1
  AggrModeAuthMethod *= IKE_auth_cs_key_cluster2
  MainModeAuthMethod *= IKE_auth_cs_key_cluster2
  AggrModePriority = TRUE
  DoAutopass = TRUE
  DPDIdleDuration = 10
  DPDResponseDuration = 3
  DPDRetries = 5
)

IPsecAction CMAP_1(
  TunnelingParameters *= TunnelEntry(
```

```

        PeerIPAddress = 10.12.118.1
        DFHandling=CLEAR
    )
    ContainedProposals *= ( ESP_GOST )
    GroupID *= MODP_1024
    IKERule = IKE_CMAP_1
)
FilteringRule Filter_nil_acl_CMAP_1(
    LocalIPFilter *= FilterEntry( IPAddress *= 192.168.101.0/24 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 192.168.103.0/24 )
    NetworkInterfaces *= "iprb0"
    Action *= ( CMAP_1 )
)
FilteringRule Filter_nil_acl(
    LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    NetworkInterfaces *= "iprb0"
    Action *= ( PASS )
)
FilteringRule Filter_nil_acl_1(
    LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    NetworkInterfaces *= "iprb1"
    Action *= ( PASS )
)

```

## LSP устройства GW3

```

# This is automatically generated LSP
# Conversion Date/Time: Wed May 3 18:45:06 2006

GlobalParameters(
    Title = "This LSP was automatically generated by
CSP Converter at Wed May 3 18:45:06 2006"
    Version = "2.1"
    LDAPLogMessageLevel = DEBUG
    SystemLogMessageLevel = DEBUG
    PolicyLogMessageLevel = DEBUG
    CertificatesLogMessageLevel = DEBUG
)
SyslogSettings(
    Server = 127.0.0.1
    Facility = LOG_LOCAL7
)
IKETransform IKETransform_1(
    CipherAlg *= "G2814789CPRO1-K256-CBC-65534"
    HashAlg *= "GR341194CPRO1-65534"

```

```
        GroupID      *= MODP_1024
        LifetimeSeconds = 86400
    )
    ESPProposal ESP_GOST(
        Transform* = ESPTransform(
            CipherAlg*      = "G2814789CPR01-K256-CBC-254"
            LifetimeSeconds = 3600
            LifetimeKilobytes = 4608000
        )
    )
    AuthMethodPreshared IKE_auth_cs_key_cluster1(
        LocalID = IdentityEntry( KeyID *= "475731" )
        RemoteID = IdentityEntry(
            IPv4Address *= 10.0.110.101
            KeyID *= "636c757374657231"
        )
        SharedIKESecret = "cs_key_cluster1"
    )
    IKERule IKE_CMAP_1(
        Transform* = IKETransform_1
        AggrModeAuthMethod *= IKE_auth_cs_key_cluster1
        MainModeAuthMethod *= IKE_auth_cs_key_cluster1
        AggrModePriority = TRUE
        DoAutopass = TRUE
        DPDIIdleDuration = 10
        DPDResponseDuration = 3
        DPDRetries = 5
    )
    IPsecAction CMAP_1(
        TunnelingParameters *= TunnelEntry(
            PeerIPAddress = 10.0.110.101
            DFHandling=CLEAR
        )
        ContainedProposals *= ( ESP_GOST )
        GroupID *= MODP_1024
        IKERule = IKE_CMAP_1
    )
    FilteringRule Filter_nil_acl_CMAP_1(
        LocalIPFilter *= FilterEntry( IPAddress *= 192.168.103.0/24 )
        PeerIPFilter  *= FilterEntry( IPAddress *= 192.168.101.0/24 )
        NetworkInterfaces *= "iprb0"
        Action *= ( CMAP_1 )
    )
    FilteringRule Filter_nil_acl(
        LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
```

```

PeerIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
NetworkInterfaces *= "iprb0"
Action *= ( PASS )
)
FilteringRule Filter_nil_acl_1(
LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
PeerIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
NetworkInterfaces *= "iprb1"
Action *= ( PASS )
)

```

## LSP устройства GW4

```

# This is automatically generated LSP
# Conversion Date/Time: Wed May 3 16:48:27 2006

GlobalParameters(
Title = "This LSP was automatically generated by
CSP Converter at Wed May 3 16:48:27 2006"
Version = "2.1"
LDAPLogMessageLevel = DEBUG
SystemLogMessageLevel = DEBUG
PolicyLogMessageLevel = DEBUG
CertificatesLogMessageLevel = DEBUG
)
SyslogSettings(
Server = 127.0.0.1
Facility = LOG_LOCAL7
)
IKETransform IKETransform_1(
CipherAlg *= "G2814789CPR01-K256-CBC-65534"
HashAlg *= "GR341194CPR01-65534"
GroupID *= MODP_1024
LifetimeSeconds = 86400
)
ESPProposal ESP_GOST(
Transform* = ESPTransform(
CipherAlg* = "G2814789CPR01-K256-CBC-254"
LifetimeSeconds = 3600
LifetimeKilobytes = 4608000
)
)
AuthMethodPreshared IKE_auth_cs_key_cluster1(
LocalID = IdentityEntry( KeyID *= "475732" )
RemoteID = IdentityEntry(
IPv4Address *= 10.0.110.101
KeyID *= "636c757374657231"
)
)

```

```

    )
    SharedIKESecret = "cs_key_cluster1"
)
IKERule IKE_CMAP_1(
    Transform* = IKETransform_1
    AggrModeAuthMethod *= IKE_auth_cs_key_cluster1
    MainModeAuthMethod *= IKE_auth_cs_key_cluster1
    AggrModePriority    = TRUE
    DoAutopass          = TRUE
    DPDIdleDuration    = 10
    DPDResponseDuration = 3
    DPDRetries         = 5
)
IPsecAction CMAP_1(
    TunnelingParameters *= TunnelEntry(
        PeerIPAddress = 10.0.110.101
        DFHandling=CLEAR
    )
    ContainedProposals *= ( ESP_GOST )
    GroupID *= MODP_1024
    IKERule = IKE_CMAP_1
)
FilteringRule Filter_nil_acl_CMAP_1(
    LocalIPFilter *= FilterEntry( IPAddress *= 192.168.103.0/24 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 192.168.101.0/24 )
    NetworkInterfaces *= "iprb0"
    Action *= ( CMAP_1 )
)
FilteringRule Filter_nil_acl(
    LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    NetworkInterfaces *= "iprb0"
    Action *= ( PASS )
)
FilteringRule Filter_nil_acl_1(
    LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    NetworkInterfaces *= "iprb1"
    Action *= ( PASS )
)

```

## Таблица маршрутов устройства GW1

Routing Table: IPv4

Destination	Gateway	Flags	Ref	Use	Interface
-------------	---------	-------	-----	-----	-----------

192.168.13.0	10.0.110.1	UG	1	2324	
192.168.101.0	192.168.16.254	UG	1	5	
192.168.16.0	192.168.16.1	U	1	62	iprb0
192.168.16.0	192.168.16.1	U	1	0	iprb0:1
192.168.16.0	192.168.16.1	U	1	0	iprb0:2
10.0.0.0	10.0.110.101	U	1	91	iprb0
10.0.0.0	10.0.110.101	U	1	0	iprb0:1
10.0.0.0	10.0.110.101	U	1	0	iprb0:2
default	10.0.110.254	UG	1	30	
127.0.0.1	127.0.0.1	UH	3	91931	lo0

### Таблица маршрутов устройства GW2:

Routing Table: IPv4

Destination	Gateway	Flags	Ref	Use	Interface
192.168.13.0	10.0.110.1	UG	1	83	
192.168.101.0	192.168.16.254	UG	1	1	
192.168.16.0	192.168.16.4	U	1	33	iprb1:1
10.0.0.0	10.0.110.104	U	1	63	iprb0:1
default	10.0.110.254	UG	1	14	
127.0.0.1	127.0.0.1	UH	2	1148	lo0

### Таблица маршрутов устройства GW3:

Routing Table: IPv4

Destination	Gateway	Flags	Ref	Use	Interface
192.168.15.0	192.168.15.1	U	1	7	iprb1
192.168.15.0	192.168.15.1	U	1	0	iprb1:1
192.168.15.0	192.168.15.1	U	1	0	iprb1:2
192.168.103.0	192.168.15.254	UG	1	7	
10.12.118.0	10.12.118.1	U	1	7	iprb0
10.12.118.0	10.12.118.1	U	1	0	iprb0:1
10.12.118.0	10.12.118.1	U	1	0	iprb0:2
224.0.0.0	10.12.118.1	U	1	0	iprb0
default	10.12.118.254	UG	1	310	
127.0.0.1	127.0.0.1	UH	2	1551	lo0

### Таблица маршрутов устройства GW4:

Routing Table: IPv4

Destination	Gateway	Flags	Ref	Use	Interface
192.168.15.0	192.168.15.4	U	1	59	iprb1:1

---

192.168.103.0	192.168.15.254	UG	1	4	
10.12.118.0	10.12.118.4	U	1	69	iprb0:1
default	10.12.118.254	UG	1	99	
127.0.0.1	127.0.0.1	UH	2	1325	lo0

## Вывод команды `ifconfig` –а устройства GW1

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000

iprb0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 8
    inet 10.0.110.101 netmask ffff0000 broadcast 10.0.255.255
    ether 0:a0:c9:85:6f:17

iprb0:1: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu
1500 index 8
    inet 10.0.110.102 netmask ffff0000 broadcast 10.0.255.255

iprb0:2: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu
1500 index 8
    inet 10.0.110.103 netmask ffff0000 broadcast 10.0.255.255

iprb1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 7
    inet 192.168.16.1 netmask ffffffff00 broadcast 192.168.16.255
    ether 0:e:a6:78:83:2d

iprb1:1: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu
1500 index 7
    inet 192.168.16.2 netmask ffffffff00 broadcast 192.168.16.255

iprb1:2: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu
1500 index 7
    inet 192.168.16.3 netmask ffffffff00 broadcast 192.168.16.255
```

## Вывод команды `ifconfig` –а устройства GW2

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000

iprb0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 8
    inet 10.0.110.101 netmask ffff0000 broadcast 10.0.255.255
    ether 0:50:fc:90:7f:1c

iprb0:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 8
    inet 10.0.110.104 netmask ffff0000 broadcast 10.0.255.255

iprb1: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 7
    inet 192.168.16.1 netmask ffffffff00 broadcast 192.168.16.255
    ether 4c:0:10:71:5c:b6

iprb1:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 7
    inet 192.168.16.4 netmask ffffffff00 broadcast 192.168.16.255
```

## Вывод команды `ifconfig` –а устройства GW3

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000

iprb0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 10.12.118.1 netmask ffffffff00 broadcast 10.12.118.255
    ether 0:d:61:12:9a:50
```

```
iprb0:1: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu
1500 index 2
    inet 10.12.118.2 netmask ffffffff broadcast 10.12.118.255
iprb0:2: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu
1500 index 2
    inet 10.12.118.3 netmask ffffffff broadcast 10.12.118.255
iprb1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.168.15.1 netmask ffffffff broadcast 192.168.15.255
    ether 0:d:61:12:9a:51
iprb1:1: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu
1500 index 3
    inet 192.168.15.2 netmask ffffffff broadcast 192.168.15.255
iprb1:2: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu
1500 index 3
    inet 192.168.15.3 netmask ffffffff broadcast 192.168.15.255
```

## Вывод команды `ifconfig` –а устройства GW4

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
iprb0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 10.12.118.1 netmask ffffffff broadcast 10.12.118.255
    ether 0:d:61:a:7b:3c
iprb0:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 10.12.118.4 netmask ffffffff broadcast 10.12.118.255
iprb1: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.168.15.1 netmask ffffffff broadcast 192.168.15.255
    ether 0:d:61:a:7b:3d
iprb1:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.168.15.4 netmask ffffffff broadcast 192.168.15.255
```