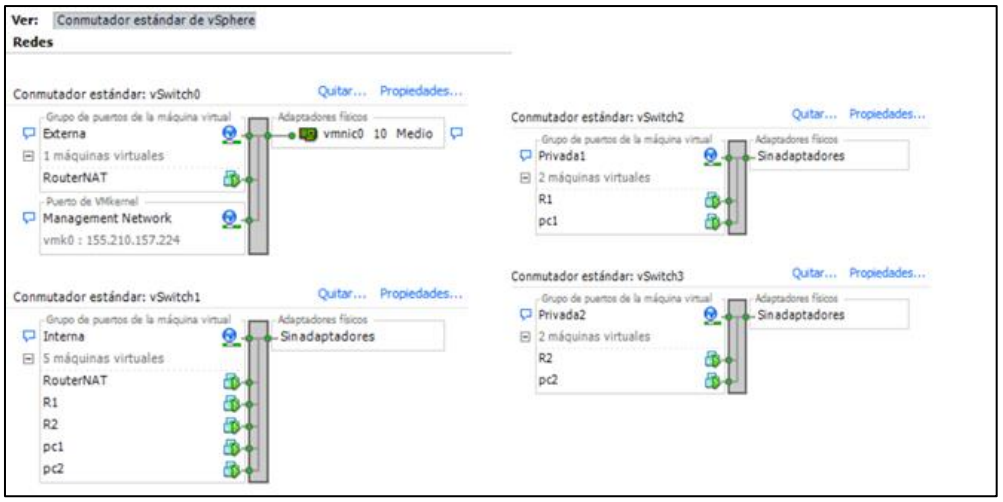
ANEXO A: MONTAJE E INSTALACIÓN

Para montar el entorno de trabajo, se utilizó en su momento VMware vSphere. Para este TFG, no ha sido necesario un uso avanzado de esta herramienta, pero sí tener unos mínimos conocimientos para establecer correctamente las conexiones o reiniciar las máquinas en caso de fallo. En la siguiente imagen, podemos comprobar la interconexión de las máquinas virtualizadas.



Captura de la configuración del escenario en vSphere

Como podemos ver en las imágenes, tenemos 4 *switch*. Dos que forman las dos redes LAN privadas, otro que conecta la máquina *routerNAT* (que hace de MS/MR) con el exterior, y otro *switch* que interconecta internamente todas las máquinas para tener acceso.

En cuanto al *software* usado en las máquinas virtuales, veamos cómo se instala cada uno de ellos:

LISP-SM: a través de su repositorio de GitHub:

- git clone https://github.com/Simplemux/lispmob-with-simplemux
- make
- sudo make install

D-ITG: el generador de tráfico se instaló en los *host*, PC1 y PC2, de la siguiente forma:

1. Descargar el paquete usando los siguientes comandos:
   - sudo apt-get update
   - sudo apt-get install d-itg
2. Desempaquetar los archivos descargados:
   - Ir al siguiente directorio:
     - cd /var/cache/apt/archives/
   - Desempaquetar:
     - dpkg -i d-itg_2.8.1-r1023-3_i386.deb

Para ejecutar el programa, los comandos se lanzan desde el directorio de los binarios, que se encuentran en: /usr/bin. Para que la ejecución sea correcta, primero es necesario lanzar ITGRecv en el *host* receptor antes que ITGSend.

IPsec-tools: su instalación se realizó tanto en los *host* como en los *router*:

- sudo apt-get update
- sudo apt-get install ipsec-tools

Tcpdump:

- sudo apt-get update
- sudo apt-get install tcpdump

ANEXO B: ARCHIVOS USADOS EN LAS PRUEBAS

Para todas las pruebas realizadas, la configuración de los parámetros de LISP del archivo *lispd.conf* se mantiene constante, mientras que serán las configuraciones de Simplemux las que irán variando. A continuación se muestran la configuración de los campos de LISP.

- En xTR1, *lispd_xTR1.conf*:

```
# General configuration

debug                   = 3
map-request-retries     = 2
log-file                = /var/log/lispd.log

# Define the type of LISP device LISPmob will operate as
#
# operating-mode can be any of:
# xTR, RTR, MN, MS, xTRSM
#
# xTRSM is the mode used for adding Simplemux functionalities. It uses
the same parameters as
#xTR, but it adds the possibility of using Simplemux features.
#
operating-mode          = xTRSM

rloc-probing {
    rloc-probe-interval             = 30
    rloc-probe-retries              = 2
    rloc-probe-retries-interval     = 5
}

map-resolver        = {
        192.168.100.254
}

map-server {
        address         = 192.168.100.254
        key-type        = 1
        key         = proyecto
        proxy-reply     = off
}

database-mapping {
    eid-prefix          = 192.168.1.0/24
    rloc-address {
        address         = 192.168.100.3
        priority        = 1
        weight          = 0
    }
}
```

- En xTR2, *lispd_xTR2.conf:*

```
# General configuration

debug                   = 3
#map-request-retries    = 2
#log-file               = /var/log/lispd.log

# Define the type of LISP device LISPmob will operate as
#
# operating-mode can be any of:
# xTR, RTR, MN, MS, xTRSM
#
# xTRSM is the mode used for adding Simplemux functionalities. It uses
the same parameters as
#xTR, but it adds the possibility of using Simplemux features.
#
operating-mode          = xTRSM


rloc-probing {
    rloc-probe-interval             = 30
    rloc-probe-retries              = 2
    rloc-probe-retries-interval     = 5
}

map-resolver        = {
        192.168.100.254
}

map-server {
        address         = 192.168.100.254
        key-type        = 1
        key         = proyecto
        proxy-reply     = off
}


database-mapping {
    eid-prefix          = 192.168.2.0/24
    rloc-address {
        address         = 192.168.100.4
        priority        = 1
        weight          = 0
    }
}
```

MODO = 0

Para las pruebas del modo 0, la configuración de los parámetros de Simplemux para la multiplexión de 2 paquetes es la siguiente:

```
simplemux {
        ipsrc           = 192.168.1.1
        ipdst           = 192.168.2.1
        lispsrc         = 192.168.100.3
        lispdst         = 192.168.100.4
        netsrc          = 192.168.1.0/24
        netdst          = 192.168.2.0/24
        num-pkt         = 2
        mtu-user        = 1500
        mtu-int         = 1500
        threshold       = 600
        timeout         = 10000000
        period          = 10000000
        ROHC-mode       = 0
        port-dst        = 8888
        port-src        = 8888
        IPSEC-mode      = 0
        min-rate-secure-packets = 1
}
```

Esta sección de configuración de Simplemux, es la continuación del archivo presentado anteriormente con los valores necesarios de LISP. Para la multiplexión de 4 paquetes, basta con cambiar en ambos xTR el campo *num-pkt* por un 4 y volver a ejecutar el archivo mediante:

- En xTR1 : /root/lispmob-with-simplemux/lispd/lispd -f /etc/lispd_xTR1.conf
- En xTR2 : /root/lispmob-with-simplemux/lispd/lispd -f /etc/lispd_xTR2.conf

Como apunte: en xTR2 al archivo es el mismo, cambiando las direcciones correspondientes. La elección de puertos, no está implementada por el momento, por lo tanto no tiene repercusión en la ejecución. Al elegir un modo de seguridad diferente al 2, el campo *min-rate-secure-packets* no tiene repercusión.

Para la prueba con tráfico seguro desde PC1 hacia PC2, es necesario configurar el archivo de *ipsec-tools.conf* correspondiente en cada *host.* Ya que el tráfico generado es UDP, se configura en la SP que dicho protocolo sea securizado. Los archivos son simétricos en cuanto a los sentidos del tráfico entrante y saliente.

En los PC, ejecutar:

- setkey –f /etc/ipsec-tools.conf

El archivo usado es:

```
## Flush the SAD and SPD
#
 flush;
 spdflush;

# SAs para ESP empleando claves largas de 192 bits (168 + 24 paridad)

add 192.168.1.1 192.168.2.1 esp 0x101 -m transport -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.2.1 192.168.1.1 esp 0x102 -m transport -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

#Politicas transporte

spdadd 192.168.1.1 192.168.2.1 UDP -P out ipsec
  esp/transport//require;

spdadd 192.168.2.1 192.168.1.1 UDP -P in ipsec
  esp/transport//require;
```

En PC2:

```
## Flush the SAD and SPD
#
 flush;
 spdflush;


# SAs para ESP empleando claves largas de 192 bits (168 + 24 paridad)

add 192.168.1.1 192.168.2.1 esp 0x101 -m transport -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.2.1 192.168.1.1 esp 0x102 -m transport -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

#Politicas transporte

spdadd 192.168.1.1 192.168.2.1 UDP -P in ipsec
  esp/transport//require;

spdadd 192.168.2.1 192.168.1.1 UDP -P out ipsec
  esp/transport//require;
```

MODO = 1

Para probar el modo 1, lo primero es borrar las políticas de seguridad ejecutadas anteriormente en los PC:

- setkey –FP

Posteriormente, configurar las políticas de seguridad de xTR1 Y xTR2, en sus respectivos archivos *ipsec-tools.conf.* Los archivos usados fueron los siguientes:

En xTR1:

```
## Flush the SAD and SPD
#
 flush;
 spdflush;

# SAs para ESP empleando claves largas de 192 bits (168 + 24 paridad)

add 192.168.100.3 192.168.100.4 esp 0x202 -u 101 -m transport -E 3des-
cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.100.4 192.168.100.3 esp 0x302 -u 102 -m transport -E 3des-
cbc 0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

#AÑADIDO ULTIMO
add 192.168.100.3 192.168.100.4 esp 0x203 -m tunnel -E 3des-cbc
0xcf4a15ba057c3d90cb66701a3ff0d428b483e65a0e1dfcba;
add 192.168.100.4 192.168.100.3 esp 0x303 -m tunnel -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;

spdadd 192.168.100.3[4344] 192.168.100.4[4344] any -P out ipsec
  esp/transport//unique:101;

spdadd 192.168.100.4[4344] 192.168.100.3[4344] any -P in ipsec
  esp/transport//unique:102;
```

En xTR2:

```
## Flush the SAD and SPD
#
 flush;
 spdflush;

# SAs para ESP empleando claves largas de 192 bits (168 + 24 paridad)

add 192.168.100.3 192.168.100.4 esp 0x202 -u 101 -m transport -E 3des-
cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.100.4 192.168.100.3 esp 0x302 -u 102 -m transport -E 3des-
cbc 0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;
```

```
#Politicas transporte

spdadd 192.168.100.3[4344] 192.168.100.4[4344] any -P in ipsec
  esp/transport//unique:101;

spdadd 192.168.100.4[4344] 192.168.100.3[4344] any -P out ipsec
  esp/transport//unique:102;
```

Tras la ejecución de ambos archivos, cambiar los campos de *IPsec-mode* a 1 y *num-pkt* al número de paquetes correspondientes a cada caso. Para realizar la prueba con tráfico seguro desde PC1, el archivo *ipsec-tools.conf* de los *host* es el mismo que usado en las pruebas del modo 0.

```
simplemux {

ipsrc           = 192.168.1.1
ipdst           = 192.168.2.1
lispsrc         = 192.168.100.3
lispdst         = 192.168.100.4
netsrc          = 192.168.1.0/24
netdst          = 192.168.2.0/24
num-pkt         = 2
mtu-user        = 1500
mtu-int         = 1500
threshold       = 600
timeout         = 10000000
period          = 10000000
ROHC-mode       = 0
port-dst        = 8888
port-src        = 8888
IPSEC-mode      = 1
min-rate-secure-packets = 1
}

simplemux {
ipsrc           = 192.168.1.1
ipdst           = 192.168.2.1
lispsrc         = 192.168.100.3
lispdst         = 192.168.100.4
netsrc          = 192.168.1.0/24
netdst          = 192.168.2.0/24
num-pkt         = 4
mtu-user        = 1500
mtu-int         = 1500
threshold       = 600
timeout         = 10000000
period          = 10000000
ROHC-mode       = 0
port-dst        = 8888
port-src        = 8888
IPSEC-mode      = 1
min-rate-secure-packets = 1
}
```

MODO = 2

En el caso del modo 2 de seguridad, se ejecutan en PC1 y PC2 sendos archivos *ipsec-tools.conf* que se mantienen iguales a las anteriores pruebas realizadas. No es necesario cambiar este archivo, ya que se cambiará en la generación de paquetes el número de paquetes UDP que se securizarán. A su vez, tampoco se modifican los *ipsec-tools.*conf de xTR1 o xTR2. En cuanto a la multiplexión, se hará con 4 paquetes.

- Ratio 0:

```
simplemux {
        ipsrc           = 192.168.1.1
        ipdst           = 192.168.2.1
        lispsrc         = 192.168.100.3
        lispdst         = 192.168.100.4
        netsrc          = 192.168.1.0/24
        netdst          = 192.168.2.0/24
        num-pkt         = 4
        mtu-user        = 1500
        mtu-int         = 1500
        threshold       = 600
        timeout         = 10000000
        period          = 10000000
        ROHC-mode       = 0
        port-dst        = 8888
        port-src        = 8888
        IPSEC-mode      = 2
        min-rate-secure-packets = 0
}
```

- Ratio 1:

```
simplemux {
        ipsrc           = 192.168.1.1
        ipdst           = 192.168.2.1
        lispsrc         = 192.168.100.3
        lispdst         = 192.168.100.4
        netsrc          = 192.168.1.0/24
        netdst          = 192.168.2.0/24
        num-pkt         = 4
        mtu-user        = 1500
        mtu-int         = 1500
        threshold       = 600
        timeout         = 10000000
        period          = 10000000
        ROHC-mode       = 0
        port-dst        = 8888
        port-src        = 8888
        IPSEC-mode      = 2
        min-rate-secure-packets = 1
}
```

ANEXO C: DIPLOMA DE GIT Y GITHUB

Durante la realización del curso *Gestión de proyectos Software con Git y GitHub*, se enseña cómo usar la herramienta *git* para realizar el control de versiones de un código, y el uso del repositorio GitHub. A continuación se adjunta el diploma de superación del mismo:


miríada

17-03-2018

Alberto Parrra

Ha completado con éxito el siguiente curso de la Universidad Politécnica de Madrid:

Gestión de proyectos Software con Git y GitHub

El objetivo principal de este MOOC es conocer y controlar Git y el servicio o plataforma GitHub (http://github.com) para poder gestionar los proyectos y coordinar a los miembros integrantes de los equipos de trabajo con éxito.

Duración del curso: 4 semanas (24 horas de estudio estimadas)

Telefónica | EDUCACIÓN DIGITAL

Con este reconocimiento no se afirma, de ninguna manera , que la persona indicada sea estudiante matriculado en la universidad responsable del curso en MiriadaX. Tampoco se le están reconociendo créditos, ni concediendo títulos, grados o acreditaciones de ningún tipo por parte de la universidad.

Acreditación de la superación del curso