

Proyecto de Grado 2010

CERTIFICACION DE IP4JVM

Anexo II - Instalación y configuración de v6eval

Autor: Daniel Enrique Rosano Lorenzo

Tutores:

Ariel Sabiguero Yawelak

Leandro Scasso

Instituto de Computación

Facultad de Ingeniería

Universidad de la República

Octubre de 2011

1 Índice

1 Índice.....	2
2 Instalación de v6eval	3
3 Configuración de v6eval.....	5
3.1 Introducción	5
3.2 Archivos de configuración.....	5
3.2.1 Formato de tn.def.....	6
3.2.2 Formato de nut.def.....	7
3.3 Instalación de Self Test	8
3.4 Como correr los tests de prueba	8
3.5 Configuraciones particulares.....	9
4 Glosario	12

2 Instalación de v6eval

Antes de comenzar la instalación de v6eval, es necesario tener instalado Perl 5.005_02 o superior y los siguientes módulos de Perl:

- Expect
- IO-Stty
- IO-Tty
- Digest-MD5
- YAML

Para obtener los módulos desde la web, puede ser necesario crear una conexión de red para descargar los archivos. Suponiendo que le0 es el nombre de la interfaz de red, se deben correr los siguientes comandos:

```
# ifconfig le0 up
# dhclient le0
```

Los módulos de Perl antes nombrados, se pueden instalar ejecutando los siguientes comandos desde shell:

```
# (cd /usr/ports/lang/p5-Expect && make install)
# (cd /usr/ports/security/p5-Digest-MD5 && make install)
# (cd /usr/ports/textproc/p5-YAML && make install)
```

Para obtener los archivos de instalación de v6eval desde un bash FreeBSD, es útil instalar wget:

```
# cd /usr/ports/ftp/wget
# make install clean
# rehash
```

Luego descargamos el paquete de instalación:

```
# wget http://www.tahi.org/release/v6eval/v6eval-3.3.1.tar.gz
```

A continuación se compila e instala v6eval:

```
% cd $SOMEWHERE/v6eval-X.Y.Z
% make
# make install
```

v6eval queda instalado en /usr/local/v6eval

Luego hay que crear un dispositivo especial BPF.

```
# vi /etc/rc.conf
# vi /etc/devfs.rules
```

```
/etc/rc.conf
-----
devfs_system_ruleset="devfsrules_unhide_bpf"
-----
```

```
/etc/devfs.rules
-----
[devfsrules_unhide_bpf=1025]
add path 'bpf*' user root group wheel mode 0660 unhide
-----
```

El usuario que ejecuta los tests necesita pertenecer al grupo wheel. Para agregar el usuario al grupo Wheel se ejecuta el comando de Unix:

```
# pw useradd usuario -G wheel
```

Configurar la línea serial

```
# touch /var/log/aculog
# chown uucp:dialer /var/log/aculog
# chmod 660 /var/log/aculog
```

El usuario que ejecuta los tests también necesita pertenecer al grupo dialer y se realiza con el comando:

```
# pw useradd usuario -G dialer
```

Configuración de la interfaz de red

En primer lugar se debe deshabilitar el soporte IPv6 en la interfaz de red a ser usada para testing.

```
# vi /etc/rc.conf
/etc/rc.conf
-----
ipv6_enable="NO"
-----
```

Al menos, una o dos interfaces de red que serán usadas deben estar en estado 'up'.

```
# vi /etc/rc.conf
/etc/rc.conf
-----
ifconfig_<Link0_device_name>="up"
-----
```

Si el NUT es un router, se debe especificar también Link1.

```
# vi /etc/rc.conf
/etc/rc.conf
-----
ifconfig_<Link0_device_name>="up"
ifconfig_<Link1_device_name>="up"
-----
```

3 Configuración de v6eval

A continuación se explica cómo configurar v6eval, así como los principales archivos de configuración de dicha herramienta.

3.1 Introducción

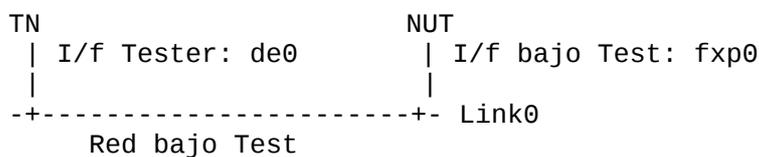
Conectar la interfaz de testing del TN a la red del NUT. Esta interfaz se puede configurar tanto para IPv4 como para IPv6.

Conectar la interfaz bajo test del NUT a la red bajo test. La interfaz del NUT debe estar configurada para IPv4 ya que cualquier paquete de IPv6 puede alterar el resultado de los tests.

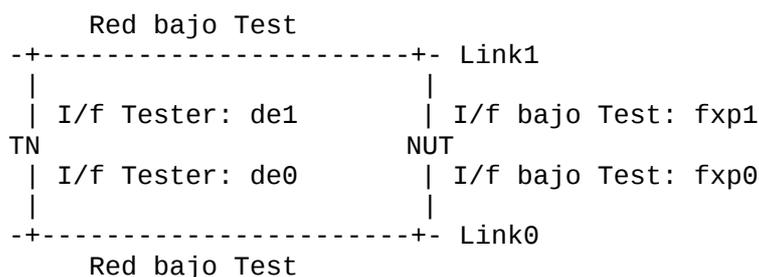
Asegúrese que ningún otro nodo esté conectado a la red, ya que sus paquetes podrían interferir en los tests.

Ejemplo:

1) NUT es un host



2) NUT es un router



En este documento suponemos que NUT es un host por defecto, y señalamos las modificaciones necesarias si NUT es un router.

3.2 Archivos de configuración

Existen dos archivos de configuración en v6eval: *tn.def* y *nut.def* que deben estar definidos en */usr/local/v6eval/etc/*. Se dan como ejemplo dos archivos *tn.def.sample* y *nut.def.sample* en dicho directorio.

Estos archivos tienen el siguiente formato:

- Las líneas vacías y aquellas que comienzan con '#' se consideran comentarios.
- En otro caso, una línea cumple el formato descrito en "keyword arguments".

3.2.1 Formato de tn.def

El archivo *tn.def* define la información acerca de TN.

Este archivo tiene las siguientes entradas:

RemoteCuPath	Especifica la ruta al comando <i>cu</i> . Valor por defecto: <i>/usr/bin/cu</i>
RemoteDevice	Especifica el dispositivo que será usado para control remoto. Valor por defecto: <i>cuaa0</i> Si el TN es FreeBSD/i386 6.0-RELEASE o superior, tal vez deba especificar <i>cuad0</i> como RemoteDevice.
RemoteDebug	Especifica el nivel de debug para el control remoto. Valor por defecto: 0
RemoteLog	Especifica el nivel de log para el control remoto. 0: no loggear mensajes de la línea serial 1: loggear mensajes de la línea serial Valor por defecto: 0
RemoteSpeed	Especifica el intervalo de tiempo de envío de caracteres. Valor por defecto: 0 (seg.)
RemoteLogout	Habilitar/deshabilitar logout. 0: omitir logout 1: hacer logout cada vez que hay una invocación remota. Valor por defecto: 0
RemoteMethod	Habilitar/deshabilitar invocación de comandos remotos. serial: habilita. none: deshabilita. Valor por defecto: serial
LinkX	Especifica el nombre y la MAC address de una interfaz de testing, de acuerdo al siguiente formato: <pre>Link0 de0 00:00:00:00:01:00 ~~~~~ ~~~ ~~~~~</pre> <p>(1) (2) (3)</p> <p>(1): el nombre de un link. En la versión actual de v6eval, únicamente se acepta "Link0" y "Link1", y no es aceptada ninguna otra opción.</p> <p>(2): nombre exacto de la interfaz de red de testing (ej: de0, le0, etc).</p> <p>(3): la MAC address de la interfaz de testing.</p> <p>Si el NUT es un router, también se necesita especificar Link1.</p>

3.2.2 Formato de nut.def

El archivo *nut.def* define la información acerca de NUT.

Este archivo tiene las siguientes entradas:

System	Especifica el tipo de sistema del NUT. Valor por defecto: manual
TargetName	Especifica la información de sistema del NUT como ser versión, tipo de CPU, etc. Esta entrada será desplegada en el archivo resultado. Valor por defecto: null
HostName	Especifica el nombre del NUT. Valor por defecto: null
Type	Especifica el tipo del NUT. Se acepta "host", "router" o "special". No puede ser omitido. Valor por defecto: host
User	Especifica el nombre del usuario del súper usuario del NUT. Si seleccionamos manual en "System", este valor puede omitirse. Valor por defecto: root
Password	Especifica la contraseña del usuario del súper usuario del NUT. Si seleccionamos manual en "System", este valor puede omitirse. Valor por defecto: v6eval
LinkX	Especifica el nombre y la MAC address de una interfaz bajo testing, de acuerdo al siguiente formato: <pre>Link0 de0 00:00:00:00:01:00 ~~~~~ ~~~ ~~~~~~~~~~~~~~~~~~~~~~ (1) (2) (3)</pre> <p>(1): el nombre de un link. En la versión actual de v6eval, únicamente se acepta "Link0" y "Link1", y no es aceptada ninguna otra opción.</p> <p>(2): nombre exacto de la interfaz de red de testing (ej: de0, le0, etc).</p> <p>(3): la MAC address de la interfaz de testing.</p> <p>Si el NUT es un router, también se necesita especificar Link1.</p>

3.3 Instalación de Self Test

Self Test (última versión 5.0.0) es un conjunto de scripts de prueba que implementa los tests de conformidad de Ipv6 Ready Logo.

Para descargar Self Test, puede ser útil el siguiente comando:

```
# wget http://www.tahí.org/logo/release/Self_Test_5-0-0.tgz
```

Luego se descomprime el tgz en alguna carpeta de nuestro agrado, y ubicados dentro de la carpeta raíz de Self Test, podemos correr los scripts de los tests:

Los tests se dividen en cinco sub-carpetas que describen las diferentes categorías:

- spec.p2 - RFC 2460: Internet Protocol, Version 6 (IPv6) Specification
- nd.p2 - RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
- addr. p2 - RFC 4862: IPv6 Stateless Address Autoconfiguration
- pmtu.p2 - RFC 1981: Path MTU Discovery for IP version 6
- icmp.p2 - RFC 4443: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification

3.4 Como correr los tests de prueba

Para correr los tests de prueba de Self Test, ejecute los siguientes comandos:

Para probar los tests para host.

```
# make ipv6ready_p2_host
```

Para probar los tests para router.

```
# make ipv6ready_p2_router
```

Si estos comandos se corren en la carpeta raíz de Self Test, se correrán todos los scripts para las cinco sub-categorías. En caso de querer correr únicamente los tests para una cierta categoría, correr los comandos antes comentados, dentro de la categoría elegida.

A veces además se desea correr un único test (o conjunto de tests) para poder verificar con rapidez un cambio. Para lograrlo, disponemos de los siguientes comandos:

Para correr los tests a partir del test *init* hasta el final.

Ejemplo: Si init=3; corre todos los tests a partir del 3.

```
# make AROPT='-s init' ipv6ready_p2_host (ó ipv6ready_p2_router)
```

Para correr los tests a partir del primero hasta el test *fin*.

Ejemplo: Si fin=8; corre los tests del 1 al 8.

```
# make AROPT='-e fin' ipv6ready_p2_host (ó ipv6ready_p2_router)
```

Para correr los tests a partir del test *init* hasta el test *fin*.

Ejemplo: Si init=4 y fin=7; corre los tests 4, 5, 6 y 7.

```
# make AROPT='-s init -e fin' ipv6ready_p2_host
```

3.5 Configuraciones particulares

Cada grupo de tests contiene configuraciones particulares para esos tests. Estas configuraciones se encuentran disponibles dentro de cada subdirectorio en el archivo config.pl.

IPv6 Specification (spec.p2)

`$wait_dadns` = 5;

Tiempo de espera para que un DADNS responda a un RA (default: 5 seg.)

`$wait_after_dadns` = 5;

Tiempo de espera para que se envíe un paquete tras recibir un DADNS (default: 5 seg.)

`$wait_reply` = 5;

Tiempo de espera por respuesta de un paquete del NUT a un paquete enviado por el TN (echo reply, icmp error message...- default: 5 seg.)

`$exceed_max` = 50;

Tiempo de espera para un ICMP Time Exceeded (default: 65 seg.)

`$exceed_min` = 40;

Tiempo de espera para enviar un ICMP Time Exceed tras recibir el primer paquete del fragmento (default: 55 seg.)

Neighbor Discovery (nd.p2)

Variables de configuracion (para routers)

`$min_MaxRtrAdvInterval` = 0;

`$max_MaxRtrAdvInterval` = 1800;

`$min_MinRtrAdvInterval` = 0;

`$max_MinRtrAdvInterval` = 1350;

`$min_AdvLinkMTU` = 0;

`$max_AdvLinkMTU` = 1500;

`$min_AdvReachableTime` = 0;

`$max_AdvReachableTime` = 3600000;

`$min_AdvRetransTimer` = 0;

`$max_AdvRetransTimer` = 0xffffffff;

`$min_AdvCurHopLimit` = 0;

`$max_AdvCurHopLimit` = 255;

\$min_AdvDefaultLifetime = 0;
\$max_AdvDefaultLifetime = 9000;

\$min_AdvValidLifetime = 0;
\$max_AdvValidLifetime = 0xffffffff;

\$min_AdvPreferredLifetime = 0;
\$max_AdvPreferredLifetime = 0xffffffff;

SAA & DAD(addr.p2)

\$wait_dadna = 5;
Tiempo de espera entre DADNS y DADNA (default: 5 seg.)

\$wait_solna = 5;
Tiempo de espera entre NS y NA (default: 5 seg.)

\$wait_rs = 5;
Tiempo de espera entre recepción DAD para LLA y recibir un RS (default: 5 seg.)

\$wait_dadns{'ra'} = 5;
Tiempo de espera para recibir un DAD NS para un global address después de recibir un RA (default: 5 seg.)

\$DupAddrDetectTransmits = 1;
DupAddrDetectTransmits: Número de veces que el NUT envía paquetes DADNS.
(default: 1 vez)

\$lla_autoconf = "YES";
Especifica si el NUT configura su link-local address automáticamente ("YES" o "NO" - default: YES)

Path MTU Discovery (pmtu.p2)

\$wait_dadns = 10;
Tiempo de espera de un DADNS en respuesta a un RA. (default: 5 seg.)

\$wait_after_dadns = 10;
Tiempo de espera para transmitir paquetes después de la llegada de DADNS (default: 5 seg.)

\$wait_reply = 5;
Tiempo de espera por respuesta de un paquete del NUT a un paquete enviado por el TN (echo reply, icmp error message...- default: 5 seg.).

\$default_mtu = 1500;
Default MTU (default: 1500)

\$wait_incomplete = 20;
Tiempo de espera para que una entrada del cache pase a estado INCOMPLETE.

(default: 10 seg.)

ICMPv6 (icmp.p2)

`$wait_reply` = 5;

Tiempo de espera por respuesta de un paquete del NUT a un paquete enviado por el TN (echo reply, icmp error message...) default: 5 seg.

`$wait_time_exc` = 50;

Tiempo máximo de espera para el envío de mensaje Time exceeded (default: 65 seg.)

`$reboot_incleanup` = "NO";

Realizar reboot después de hacer cleanup() (YES: reboot, NO: not reboot - default: NO)

4 Glosario

BPF (Berkeley Packet Filter): Interfaz a la capa de acceso a datos que permite que se envíen/reciban paquetes.

DAD (Duplicate Address Detection): Mecanismo por el cual un nodo que está obteniendo su dirección de red puede detectar si esa dirección es única o no.

DADNS (Duplicate Address Detection Neighbor Solicitation): Mensaje que se intercambia en el proceso de DAD.

FreeBSD: Sistema operativo utilizado en el TN.

ICMPv6 (Internet Control Message Protocol): Sub protocolo de IP utilizado para el envío y gestión de mensajes de control.

I/f (Interface): Acrónimo dado por TAHI para referirse a una interfaz de red.

LLA (Link Local Address): Es un tipo de dirección IPv6 utilizado para comunicaciones en una red local o en conexiones punto a punto. El prefijo utilizado es fe80::/64

MTU (Maximun Transmission Unit): Es la cantidad máxima de bytes que se pueden enviar en un paquete IP (usualmente 1492 bytes).

NA (Neighbor Advertisement): Mensaje enviado en respuesta a un NS.

NS (Neighbor Solicitation): Es un mensaje para averiguar la dirección IP de un vecino.

NUT (Node Under Test): Nodo sometido a testing.

RA (Router Advertisement): Mensaje enviado por un router para promocionar su dirección de red.

RS (Router Solicitation): Mensaje enviado por un nodo para descubrir routers en su red.

SAA (stateless Address Autoconfiguration):

TN (Tester Node): Nodo que realiza el testing.

v6eval: Herramienta con la cual se llevan a cabo los test de conformidad.