

Tutorial de DNSSEC: Firmado de zonas

Sebastian Castro
LACNIC 30
Rosario, Argentina, Septiembre 2018

Agenda

Mantenimiento de zonas firmadas

- Generación de llaves
- Firma de zona
- Rotación de llaves
- Cambios a la zona
- Regeneración de firmas
- Actualización de cadena de confianza

Agenda

¿Qué son los HSM?

- ¿Por qué usar un HSM?

¿Cómo firmar?

- Usando OpenDNSSEC y HSM
- Usando BIND y HSM

Agenda

Monitoreo de zonas firmadas

- Integridad
- Presencia de llaves
- Expiración de firmas
- Validación de firmas
- Cadena de confianza completa
- Validación desde la raíz

Mantenimiento de zonas firmadas

Generación de llaves

- Al decidir firmar una zona, necesitas generar llaves para KSK y ZSK
- Con los parámetros adecuado: algoritmo, largo de la llave
- Para todas las zonas a firmar
- Y llaves extras para rotación futura

Mantenimiento de zonas firmadas

Firma de la zona

- Tomar el contenido de la zona original
- Ordenar en orden lexicográfico
- Canonizar registros
- Generar secuencias NSEC/NSEC3
- Agregar registros DNSKEY
- Generar registros RRSIG

Mantenimiento de zonas firmadas

Rotación de llaves

- Introducir nuevas llaves (KSK o ZSK) en el momento correcto
- Mantención de estado de las llaves
- Remover llaves ya usadas al final de la rotación
- Usar las llaves correctas para firmar

Mantenimiento de zonas firmadas

Cambios a la zona

- Editar o regenerar zona
- Refirmar

Regeneración de firmas

- Si la zona no cambia, los registros RRSIG tienen que refrescarse.
- Refresco en el momento adecuado
- No refrescar todos los registros a la vez

Mantenimiento de zonas firmadas

Actualización de cadena de confianza

- Enviar el registro DS a la zona padre en el momento adecuado
- Verificar que el registro DS calza con el DNSKEY
- Durante la rotación de llaves, cambiar el registro DS en el padre

HSM

Hardware Security Modules

- Almacenamiento seguro de llaves
- Previene la extracción de las llaves privadas
- En algunos casos, aceleración de operaciones criptográficas
- Interface de acceso bien definida (PKCS#11)

HSM

Protección vía software y hardware

- Si el dispositivo es comprometido, las llaves se borran

Generalmente incluyen un generador de números aleatorios

NLnetLabs implementó un HSM en software, llamado SoftHSM.

- Muy útil para probar el uso de un HSM antes de comprar

¿Por qué usar un HSM?

Manejo de riesgos

- La llave puede ser comprometida
 - Intrusos en el servidor
 - Personal comprometido o poco feliz
 - Factorización de la llave
- Reducción de riesgos
 - Proteger el servidor
 - Proteger las llaves
 - HSM siguen estándares de seguridad bien definidos, ejemplo FIPS 140-2

Firmado usando OpenDNSSEC

OpenDNSSEC se encarga de

- Mantención de zonas firmadas
- Mantención de las llaves asociadas

Creado para proveer

- Buen manejo de llaves
- Manejo de parámetros vía políticas
- Soporte para HSM

Firmado usando OpenDNSSEC

OpenDNSSEC opera como caja negra

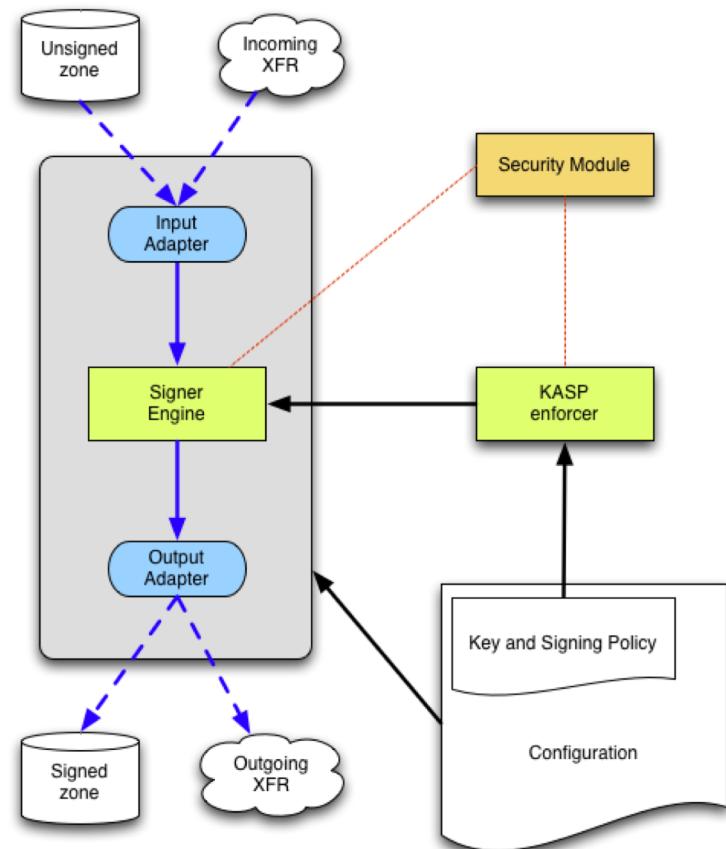
- Le alimentas zonas sin firmar
 - Vía archivo o transferencia de zona
- Produce zonas firmadas
 - Vía archivo o transferencia de zona
- Pensado para operar entre el origen de la zona y un primario o servidor de distribución

Architectura de OpenDNSSEC

HSM: Almacén de llaves

Enforcer: administra zonas, llaves, eventos. Rotación de llaves

Signer: recibe zonas sin firmar, firma zonas, entrega zonas firmadas



OpenDNSSEC: configuración

- Archivos de configuración en XML
 - conf.xml
 - kasp.xml
 - addns.xml
- Archivos de zona
- Archivos temporales
- kasp.db: base SQLite con el estado de las llaves. Crítico, no olvide respaldar!

OpenDNSSEC: tutorial

- Instrucciones basadas en Raspberry PI with Raspbian y NitroKey HSM

```
sudo apt-get install opensc
```

- OpenSC provee acceso al HSM

```
pkcs11-tool --show-info
Cryptoki version 2.20
Manufacturer      OpenSC Project
Library           OpenSC smartcard framework (ver 0.16)
Using slot 0 with a present token (0x0)
```

- HSM proveen diferentes "slots" para almacenar llaves.

OpenDNSSEC: tutorial

- Un HSM requiere ser inicializado.
Generalmente tienen una clave por omisión

```
sc-hsm-tool --initialize --so-pin 3537363231383830 --pin  
648219 --label "NitroKey HSM"
```

OpenDNSSEC: tutorial

- Podemos ver el slot por omisión, #0

```
pkcs11-tool -L
Available slots:
Slot 0 (0x0): Nitrokey Nitrokey HSM
(010000000000000000000000) 00 00
    token label          : NitroKey HSM (UserPIN)
    token manufacturer   : www.CardContact.de
    token model           : PKCS#15 emulated
    token flags            : rng, login required, PIN initialized,
token initialized
    hardware version      : 24.13
    firmware version       : 2.6
    serial num             : DENK0101304
```

OpenDNSSEC: tutorial

- Instalamos OpenDNSSEC

```
sudo apt-get install opendnssec libhsm-bin
```

OpenDNSSEC: tutorial

- Configurar HSM como repositorio de llaves. Editamos
`/etc/opensssec/conf.xml`

```
<Repository name="nitrokey">                                         Provisto por OpenSC
    <Module>/usr/lib/arm-linux-gnueabihf/opensc-pkcs11.so</Module>
    <TokenLabel>NitroKey HSM (UserPIN)</TokenLabel>                 Definido en la
    <PIN>648219</PIN>                                               inicialización
    <Capacity>20</Capacity>                                         Capacidad del
    <RequireBackup/>                                              HSM/Slot
    <SkipPublicKey/>
</Repository>
```

OpenDNSSEC: tutorial

- Verificamos acceso usando OpenDNSSEC

```
sudo -u opendnssec ods-hsmutil info
```

Repository: nitrokey

Module:	/usr/lib/arm-linux-gnueabihf/opensc-pkcs11.so
Slot:	0
Token Label:	NitroKey HSM (UserPIN)
Manufacturer:	www.CardContact.de
Model:	PKCS#15 emulated
Serial:	DENK0101304

OpenDNSSEC: tutorial

- Generamos una llave usando OpenDNSSEC

```
sudo -u opendnssec ods-hsmutil generate nitrokey rsa 1024
Generating 1024 bit RSA key in repository: nitrokey
Key generation successful: 0bd3b444ce9aae5eeef541c170bf41f5
```

- Verificamos que la llave existe

```
sudo -u opendnssec ods-hsmutil list nitrokey
```

```
Listing keys in repository: nitrokey
```

```
1 key found.
```

Repository	ID	Type
-----	--	----
nitrokey	0bd3b444ce9aae5eeef541c170bf41f5	RSA/1024

OpenDNSSEC: tutorial

- Borramos la llave de prueba

```
sudo -u opendnssec ods-hsmutil remove  
0bd3b444ce9aae5eeef541c170bf41f5  
Key remove successful.
```

- Estamos listos para configurar OpenDNSSEC para firmado

OpenDNSSEC: tutorial

- Borramos la llave de prueba

```
sudo -u opendnssec ods-hsmutil remove  
0bd3b444ce9aae5eeef541c170bf41f5  
Key remove successful.
```

- Estamos listos para configurar OpenDNSSEC para firmado

OpenDNSSEC: inicializar DB

- OpenDNSSEC almacena politicas y estado de las zonas en SQLite.

```
sudo ods-enforcer-db-setup
*WARNING* This will erase all data in the database; are you
sure? [y/N] y
Database setup successfully.
```

OpenDNSSEC: política

- Definimos una política de firmado
 - Parámetros para las firmas, en que HSM se almacenan
 - NSEC o NSEC3
 - Parámetros para las llaves: algoritmos y tamaño para KSK y ZSK, tiempos de introducción
 - Parámetros del registro SOA para la zona a firmar
 - Parámetros de la zona padre

OpenDNSSEC: verificación

- Una vez editado `kasp.xml`, verificar

```
sudo -u opendnssec ods-kaspcheck
INFO: The XML in /etc/opendnssec/conf.xml is valid
INFO: The XML in /etc/opendnssec/kasp.xml is valid
WARNING: In policy default, Y used in duration field for
Keys/KSK Lifetime (P1Y) in /etc/opendnssec/kasp.xml - this
will be interpreted as 365 days
WARNING: In policy lab, Y used in duration field for Keys/KSK
Lifetime (P1Y) in /etc/opendnssec/kasp.xml - this will be
interpreted as 365 days
INFO: The XML in /etc/opendnssec/zonelist.xml is valid
```

OpenDNSSEC: ejecutar componentes

- Con la base de datos inicializada, y políticas definidas, podemos iniciar los servicios

```
sudo ods-control start
Starting enforcer...
OpenDNSSEC key and signing policy enforcer version 2.0.4
Engine running.
ctrl completed in 0 seconds.
Starting signer engine...
OpenDNSSEC signer engine version 2.0.4
Engine running.
```

OpenDNSSEC: zona mínima

- Preparamos una versión minima de una zona en el archivo
/var/lib/openssl/unsigned/dnsseclab.nz.zone

```
dnsseclab.nz.      3600    IN      SOA      pri.dnsseclab.nz.  
hostmaster.dnsseclab.nz. 2018091200 21600 7200 2592000 3600  
dnsseclab.nz.      1800    IN      NS       puck.nether.net.  
dnsseclab.nz.      3600    IN      NS       pri.dnsseclab.nz.  
dnsseclab.nz.      43200   IN      MX       1 aspmx.l.google.com.  
dnsseclab.nz.      43200   IN      MX       5 alt1.aspmx.l.google.com.  
dnsseclab.nz.      43200   IN      MX       5 alt2.aspmx.l.google.com.  
pri.dnsseclab.nz. 43200   IN      A        54.218.206.177  
z-token.dnsseclab.nz. 43200   IN      TXT     "MARK"
```

OpenDNSSEC: agregar zona

- Le indicamos a ODS que queremos mantener una zona

```
sudo ods-enforcer zone add --zone dnsseclab.nz --policy lab  
  --input /var/lib/opendnssec/unsigned/dnsseclab.nz.zone  
  --in-type file  
  --output /var/lib/opendnssec/signed/dnsseclab.nz.zone  
input is set to /var/lib/opendnssec/unsigned/dnsseclab.nz.zone.  
output is set to /var/lib/opendnssec/signed/dnsseclab.nz.zone.  
Zone dnsseclab.nz added successfully  
zone add completed in 81 seconds.
```

OpenDNSSEC: tras bambalinas

- La zona es mantenida ahora

```
sudo ods-enforcer zone list
Database set to: /var/lib/opendnssec/kasp.db
Zones:
Zone:           Policy:       Next change:
Signer Configuration:
dnsseclab.nz      lab        Wed Sep 12 16:01:40
2018  /var/lib/opendnssec/signconf/dnsseclab.nz.xml
zone list completed in 0 seconds.
```

OpenDNSSEC: tras bambalinas

- ods-enforcer creo las llaves necesarias para la zona

```
ods-hsmutil list softhsm
```

```
Listing keys in repository: softhsm
6 keys found.
```

Repository	ID	Type
-----	--	-----
softhsm	2575ab4b87f57c1d1aa8c59290578408	RSA/2048
softhsm	6aad1f30bdc83650ada1db2ec2561910	RSA/1024
softhsm	34a5e2e9a4e92092b4058939be3f1523	RSA/2048
softhsm	cfbe3877fdf0e8dd338f7eaa956793b9	RSA/1024
softhsm	d78c7f4b0a1986d13876ec4d9ddeb217	RSA/1024
softhsm	a470392f7d794f968e230eb473001c14	RSA/1024

OpenDNSSEC: tras bambalinas

- Y el archivo `/var/log/syslog` muestra

```
Sep 17 12:09:22 raspberrypi ods-enforcerd: [zone_add_cmd] zone dnsseclab.nz added [policy: lab]
Sep 17 12:09:22 raspberrypi ods-enforcerd: [hsm_key_factory_generate]
3 keys needed for 1 zones covering 86400 seconds, generating 3 keys
for policy lab
Sep 17 12:09:22 raspberrypi ods-enforcerd: 3 new ZSK(s) (1024 bits)
need to be created.
Sep 17 12:09:24 raspberrypi ods-enforcerd: [hsm_key_factory_generate]
1 keys needed for 1 zones covering 86400 seconds, generating 1 keys
for policy lab
Sep 17 12:09:24 raspberrypi ods-enforcerd: 1 new KSK(s) (2048 bits)
need to be created.
Sep 17 12:09:29 raspberrypi ods-signerd: [STATS] dnsseclab.nz
2018091201 RR[count=8 time=0(sec)] NSEC[count=3 time=0(sec)]
RRSIG[new=8 reused=0 time=1(sec) avg=8(sig/sec)] TOTAL[time=1(sec)]
```

OpenDNSSEC: Otras tareas

- Rotación de llaves: enforcer se encarga
- Refirmado de la zona: signer se encarga
- Actualización de firmas: signer se encarga
- Backup de llaves: depende del HSM
- Publicación de la zona firmada: depende de la arquitectura
- Actualización de la cadena de confianza

OpenDNSSEC: Zona firmada

- `/etc/openssl/conf.xml`

```
<!-- the <NotifyCommand> will expand the following variables:
```

```
        %zone      the name of the zone that was signed  
        %zonefile  the filename of the signed zone  
-->  
<NotifyCommand>/usr/local/bin/my_nameserver_reload_command %zone  
%zonefile</NotifyCommand>
```

- Una vez que la zona se firma, este comando se ejecuta
- Es el lugar perfecto para verificar que la zona firmada este correcta antes de publicar

OpenDNSSEC: Registros DS

- `/etc/opendnssec/conf.xml`

```
<DelegationSignerSubmitCommand>/usr/sbin/simple-dnskey-  
mailer.sh</DelegationSignerSubmitCommand>
```

- Cuando se generan nuevos registros DS, se ejecuta el comando
 - Pueden actualizar el padre directamente
 - O recibir una notificación por correo
 - En .NZ, el script manda un correo firmado con PGP para los registros DS de .nz que se envían a IANA
 - Para otras zonas, los registros se inyectan en la correspondiente zona padre.

Firmando con BIND

BIND

- Firmar con BIND y un HSM es desafiante
- Si van a usar un HSM, necesitaran
 - Parchar y recompilar OpenSSL
 - Recompilar BIND para usar OpenSSL
 - Mantener las versiones alineadas a mano
 - Una buena dosis de fe y oraciones

Después de horas de intentar, recompilar, y leer, ni SoftHSM ni Nitrokey funcionó.

BIND

- Puede mantener una zona firmada
 - Zona estática
 - Zona dinámica usando dynamic updates
- Partiendo con BIND 9.11
 - Administración de llaves usando dnssec-keymgr
- No hace bien
 - Generación de llaves la primera vez

BIND: Configuración base

- Una configuración mínima para mantener zonas firmadas en named.conf

```
options {  
    directory "/var/cache/bind";  
  
    dnssec-validation no; // No recursion  
    recursion no;  
    dnssec-enable yes;  
  
    auth-nxdomain no;    # conform to RFC1035  
    listen-on-v6 { any; };  
};
```

BIND: Generar llaves

```
sudo mkdir -p /var/cache/bind/keys/dnsseclab.nz
sudo chown -R bind:bind /var/cache/bind/keys/
cd /var/cache/bind/keys
# Generate a KSK
sudo -u bind dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -T DNSKEY -f
KSK dnsseclab.nz.
# Generate a ZSK
sudo -u bind dnssec-keygen -a RSASHA256 -b 1024 -n ZONE -T DNSKEY
dnsseclab.nz.
```

BIND: Firmar zona

- Creamos una zona de juguete para firmar

```
sudo -u bind mkdir /var/cache/bind/zones
# Create test zonefile as before
sudo cp /var/lib/opendnssec/unsigned/dnsseclab.nz.zone
/var/cache/bind/zones
```

- Configuramos BIND para firmar

```
zone dnsseclab.nz {
    type master;
    file "zones/dnsseclab.nz.zone";
    auto-dnssec maintain;
    update-policy local;
    key-directory "keys/dnsseclab.nz";
};
```

BIND: Firmar zona

- BIND no es muy conservador

```
Sep 20 12:56:57 raspberrypi named[25492]: zone dnsseclab.nz/IN:  
reconfiguring zone keys
```

```
Sep 20 12:56:57 raspberrypi named[25492]: zone dnsseclab.nz/IN: next  
key event: 20-Sep-2018 13:56:57.693
```

- Pero si consultamos el servidor

```
dig SOA dnsseclab.nz @localhost +dnssec  
dnsseclab.nz.          3600      IN      SOA      pri.dnsseclab.nz.  
hostmaster.dnsseclab.nz. 2018091202 21600 7200 2592000 3600  
dnsseclab.nz.          3600      IN      RRSIG    SOA 8 2 3600  
20181020005657 20180919235657 64053 dnsseclab.nz.  
JMw359LPbqtV74cvF2IbBFL4qnYZuBgnWt1dzXH0eRluIonbz29hFBAE  
r2YEqdqgCXVkyIu4qgQ1nk/Rwrgjuc80rrt6MCaCBCWlnlNC09gnTwgF  
X84wD5POKmYWq0MqMBmjg/LzbGD70tezJlPCuT04RrkrskaJJu61BUZa QTY=
```

BIND: Mantener zona firmada

- BIND firmara la zona mientras encuentre llaves asociadas.
- Cambios a la zona necesitan ser vía dynamic updates

BIND: Rotación de Llaves

- Veamos los tiempos de las llaves que tenemos

```
cd /var/cache/bind/keys/dnsseclab.nz
sudo -u bind dnssec-settime -p all Kdnsseclab.nz.+008+64053.key
Created: Thu Sep 20 12:43:58 2018
Publish: Thu Sep 20 12:43:58 2018
Activate: Thu Sep 20 12:43:58 2018
Revoke: UNSET
Inactive: UNSET
Delete: UNSET
```

- Creamos una nueva ZSK

```
sudo -u bind dnssec-keygen -a RSASHA256 -b 1024 -n ZONE -T DNSKEY -A
now+1h dnsseclab.nz.
Generating key
pair.....+++++
++ ....+++++
Kdnsseclab.nz.+008+56373
```

BIND: Rotación de llaves

- Creamos una nueva ZSK

```
sudo -u bind dnssec-keygen -a RSASHA256 -b 1024 -n ZONE -T DNSKEY -A
now+1h dnsseclab.nz.
Generating key
pair.....+++++
++ ....+++++
Kdnsseclab.nz.+008+56373
```

- Le decimos a BIND sobre la nueva llave

```
sudo rndc loadkeys dnsseclab.nz
```

- Y vemos en el log

```
Sep 20 13:42:14 raspberrypi named[25492]: zone dnsseclab.nz/IN:
reconfiguring zone keys
Sep 20 13:42:14 raspberrypi named[25492]: zone dnsseclab.nz/IN: next
key event: 20-Sep-2018 14:38:32.850
```

BIND: Cadena de confianza

- Para completar la cadena de confianza, necesitamos los registros DS

```
sudo -u bind dnssec-dsfromkey -K . -1 Kdnsseclab.nz.+008+60820.key
dnsseclab.nz. IN DS 60820 8 1
8EAC710E126747C3CA32257B6A1632CC96FA9B07
```

```
sudo -u bind dnssec-dsfromkey -K . -2 Kdnsseclab.nz.+008+60820.key
dnsseclab.nz. IN DS 60820 8 2
683CF1B1C37D049BDF8BC882A3891D5BCC06E12B5AE0BC2068EF90E44A553D02
```

BIND: Otras tareas

- Publicar la zona firmada?
 - BIND provee transferencia de zona y soporta NOTIFY cuando la zona ha sido firmada
- Ver la zona firmada?
 - En el lugar de la zona original!
 - Se puede usar inline-signing para preservar la zona original
- Mi zona es regenerada cada vez, como la firmo?
 - No hemos probado esa opción :(
- Se pueden automatizar los rollovers?

BIND: dnssec-keymgr

- En BIND 9.11 se introduce **dnssec-keymgr**
 - Diseñada para ejecutar vía cron
 - Utiliza un archivo de políticas
 - Crea y actualiza archivos de llaves dependiendo de ciertos eventos
 - Automatiza la creación y rotación de llaves
 - Escrito en Python como un *wrapper* para **dnssec-keygen** y **dnssec-settime**
- BIND 9.11.0 Release Notes

(Many thanks to Sebastián Castro for his assistance in developing this tool at the IETF 95 Hackathon in Buenos Aires, April 2016.)

BIND: dnssec-keymgr

- Usando otro laboratorio con BIND 9.11

```
cd /var/cache/bind/keys
# Generate a KSK
sudo -u bind dnssec-keygen -a RSASHA256 -b 4096 -n ZONE -T DNSKEY -f
KSK -A now+1h dnsseclab.nz.
# Generate a ZSK
sudo -u bind dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -T DNSKEY -A
now+1h dnsseclab.nz.
```

BIND: DNSSEC policy

```
policy global {  
    algorithm rsasha256;  
    key-size ksk 4096;  
    key-size zsk 2048;  
    roll-period ksk 1w;  
    roll-period zsk 1d;  
    pre-publish ksk 1d;  
    pre-publish zsk 1h;  
    post-publish ksk 1d;  
    post-publish zsk 1h;  
    standby ksk 1;  
    standby zsk 1;  
    keyttl 5mi;  
    coverage 2w;  
};  
  
policy default { policy global; };
```

BIND: aplicar política

- Instruimos a dnssec-keymgr que aplique política

```
sudo -u bind dnssec-keymgr -c /etc/bind/dnssec-policy.conf -K  
/var/cache/bind/keys/dnsseclab.nz/  
# /usr/sbin/dnssec-settime -K /var/cache/bind/keys/dnsseclab.nz/ -P  
20180920053833 -A 20180920053833 -I 20180921063826 -D 20180921073826  
Kdnsseclab.nz.+008+10308  
# /usr/sbin/dnssec-keygen -q -K /var/cache/bind/keys/dnsseclab.nz/ -S  
Kdnsseclab.nz.+008+10308 -L 300 -i 3600
```

- Fijara los parámetros de tiempo y creara las llaves que sean necesarias.
 - Con esto BIND hará los rollovers automáticamente

Otras opciones

- Knot DNS es un servidor de DNS autoritativo creado por CZ.NIC
- Soporta DNSSEC
- Parte de DNSSEC appliances de Secure64
- Elegido por RIPE para mantener sus zonas
 - <https://labs.ripe.net/Members/anandb/dnssec-signer-migration>

Preguntas

iGRACIAS!

sebastian@internetnz.net.nz