

Desplegando servicios federados en eduroam con Moonshot

Grupos de trabajo de RedIRIS 2014, Madrid

Alejandro Pérez, Rafael Marín, Gabriel López

Departamento de Ingeniería de la Información y las Comunicaciones
Universidad de Murcia

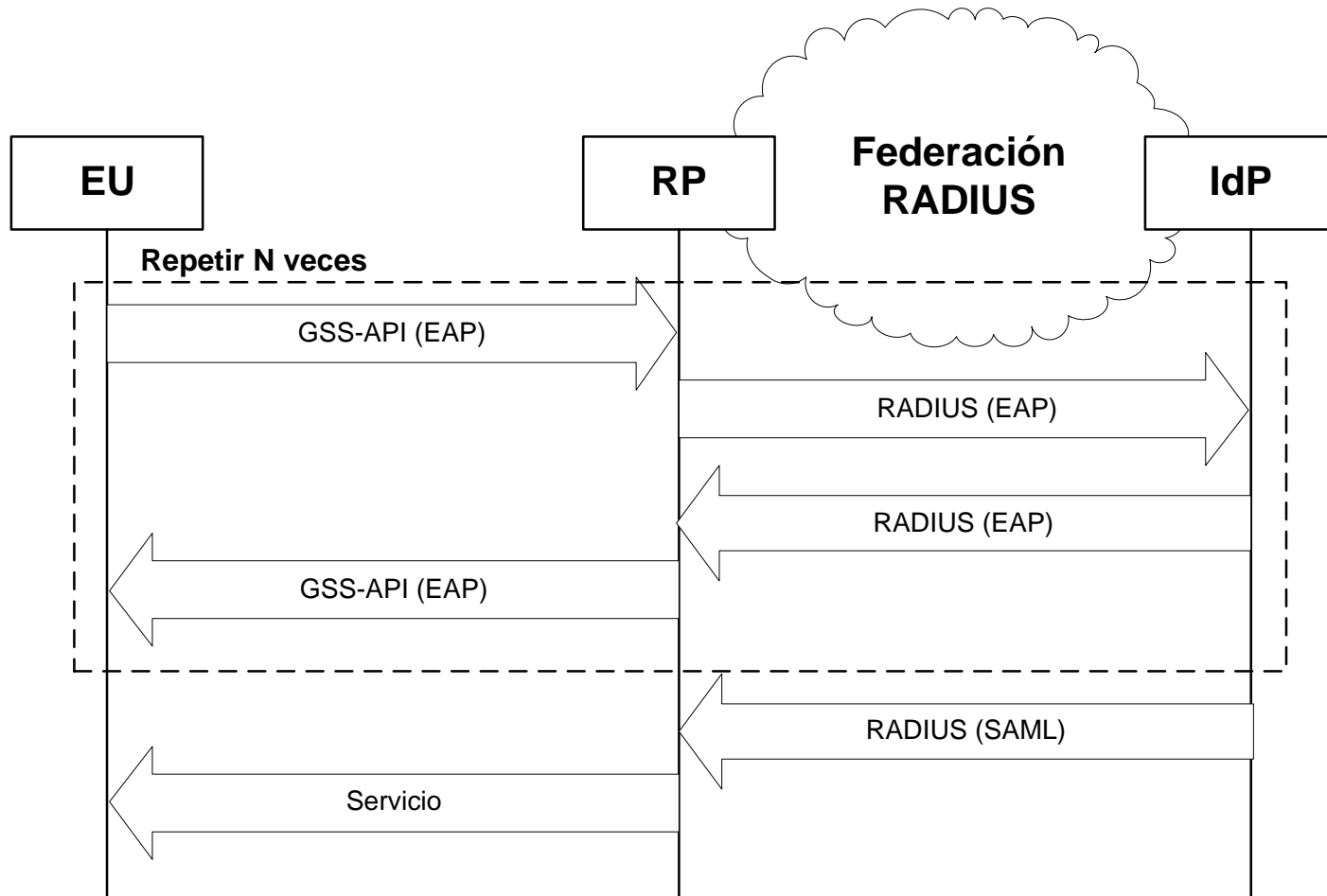
Motivación

- Federaciones de identidad
 - Relaciones de confianza para identificar usuarios
 - Usabilidad y menor coste de despliegue
- Inconvenientes
 - Definidas para tipos de servicio específicos
 - Uso de tecnologías diferentes
 - Acceso a la red (p.ej. eduroam) → RADIUS, Diameter...
 - Servicios web → SAML, OpenID, OAuth...
 - Algunos servicios no disponen de soluciones de federación
 - Correo electrónico, acceso remoto a ficheros, acceso a terminal remoto...

¿Qué es Moonshot?

- Moonshot
 - Desarrollo de una tecnología para llevar el concepto de identidad federada a cualquier tipo de servicio (cloud, ftp, http, ssh...)
- Elementos clave:
 - EU → Quiere acceder a un servicio
 - RP → Proporciona el servicio
 - IdP → Autentica al usuario y proporciona información de autorización al RP
- Tecnologías clave:
 - GSS-API → Control de acceso a servicios (entre EU y RP)
 - RADIUS → Federación (entre RP e IdP)
 - SAML → Autorización (entre RP e IdP)
 - EAP → Autenticación de usuario (entre EU e IdP)

¿Qué es Moonshot?



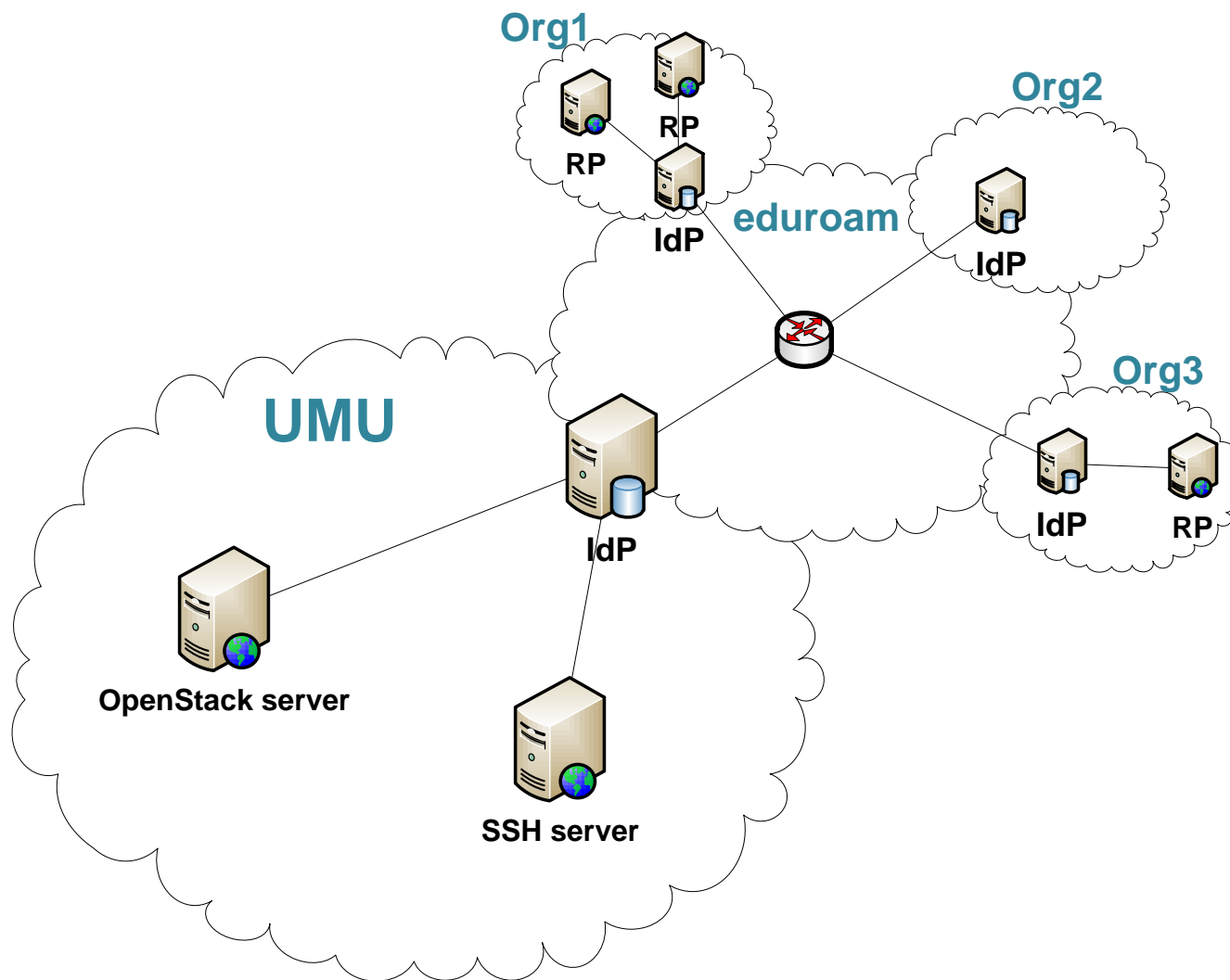
¿Qué es Moonshot?

- Realizado en parte dentro del proyecto GN3
 - Participado por RedIRIS y la UMU
- En proceso de estandarización dentro del IETF (ABFAB WG)
- Completamente implementado, documentado y mantenido por la comunidad Moonshot
 - <https://community.ja.net/groups/moonshot>

Desplegando Moonshot

- La infraestructura RADIUS de eduroam es una candidata ideal para el despliegue de Moonshot
 - Las relaciones de confianza ya están establecidas
 - Multitud de organizaciones interconectadas
- Veremos cómo desplegar los diferentes componentes de Moonshot usando esta infraestructura
- Dos ejemplos prácticos, desplegados en UMU:
 - Servidor SSH que permite el acceso a la cuenta *federated* a cualquier miembro de la comunidad eduroam (**GN3Plus**)
 - Servidor OpenStack que permite el acceso al tenant *swifttenanttest1* sólo a los miembros de UMU, y el acceso a *swifttenanttest2* a cualquier miembro de la comunidad eduroam (**CLASSe**)

Desplegando Moonshot



Desplegando Moonshot - IdP

- Cualquier servidor RADIUS actual de eduroam pueden actuar como IdPs Moonshot...
 - Pero no envían la sentencia SAML
- Para configurar un IdP nuevo:
 1. Instalar FreeRADIUS y conectarlo a la infraestructura eduroam
 2. Crear las cuentas de usuario deseadas
 3. Configurar FreeRADIUS para que genere una sentencia SAML
 - Estática → Plantilla de sentencia fija, rellena con variables FreeRADIUS
 - Dinámica → Sentencia generada con OpenSAML, rellena con valores obtenidos de diferentes bases de datos (en desarrollo)

Ejemplo 1 y 2: IdP

- Configuramos un nuevo servidor RADIUS
 - Subdominio de la UM
 - moonshot.um.es
- Creamos una cuenta de pruebas
 - test@moonshot.um.es
- Configuramos la plantilla para la sentencia SAML de forma estática
 - Sección post-auth del fichero sites-enabled/default

```
update reply {  
  SAML-AAA-Assertion = "<saml:Assertion xmlns:saml='urn:oasis:names:tc:.....'"  
  SAML-AAA-Assertion += "<saml:Conditions NotOnOrAfter='2015-03-19T08:30:00Z'/>"  
  SAML-AAA-Assertion += "<saml:Issuer>moonshot.inf.um.es</saml:Issuer>"  
  .....
```

Ejemplo 1 y 2: IdP

```
<saml:Assertion xmlns:saml='urn:oasis:names:tc:SAML:2.0:assertion' ...>
  <saml:Conditions NotOnOrAfter='2015-03-19T08:30:00Z' />
  <saml:Issuer>moonshot.um.es</saml:Issuer>
  <saml:Subject>
    <saml:NameID Format='urn:oasis:names:tc:SAML:2.0:nameid-format:transient'>
      %{%reply:User-Name}:-{%request:User-Name}}
    </saml:NameID>
  </saml:Subject>
  <saml:AttributeStatement>
    <saml:Attribute Name='studentcard' ...>
      <saml:AttributeValue>Student</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute Name='affiliation' ...>
      <saml:AttributeValue>umu</saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

Ejemplo:
test@moonshot.um.es

Desplegando Moonshot - RP

- Cualquier aplicación que soporte GSS-API debe funcionar con Moonshot
 - Aunque algunas están mal programadas y requieren pequeños ajustes (ej. OpenSSH)
- Será necesario:
 1. Instalar código de Moonshot en el RP
 2. Configurar un proxy RADIUS conectado a la infraestructura eduroam
 3. Configurar mapeo de atributos (autorización)
 - Convertir atributos RADIUS y/o SAML en atributos específicos de la aplicación

Ejemplo 1: Servidor SSH

- Configuramos la máquina
 - moonshot-ssh.inf.um.es
- Instalamos Moonshot
- Instalamos servidor OpenSSH parcheado para Moonshot
 - Disponible con el propio código de Moonshot
- Configuramos el proxy RADIUS
 - moonshot.um.es
- Configuramos el mapeo de atributos
 - Si TRUE →
 - OpenSSH.local_login_user := federated
 - No requiere sentencia SAML
 - Esto autoriza a cualquier usuario a acceder a la cuenta federated@moonshot-ssh.inf.um.es

Ejemplo 2: Servidor OpenStack

- Configuramos la máquina
 - classe1.qalab.geant.net
- Instalamos Moonshot
- Instalamos servidor OpenStack con soporte para GSS-API
 - <https://github.com/kwss/keystone>
- Configuramos el proxy RADIUS
 - moonshot.um.es
- Configurar mapeo de atributos
 - Si *SAML.affiliation* == umu →
 - OpenStack.tenant := swifttenanttest1
 - Si no →
 - OpenStack.tenant := swifttenanttest2

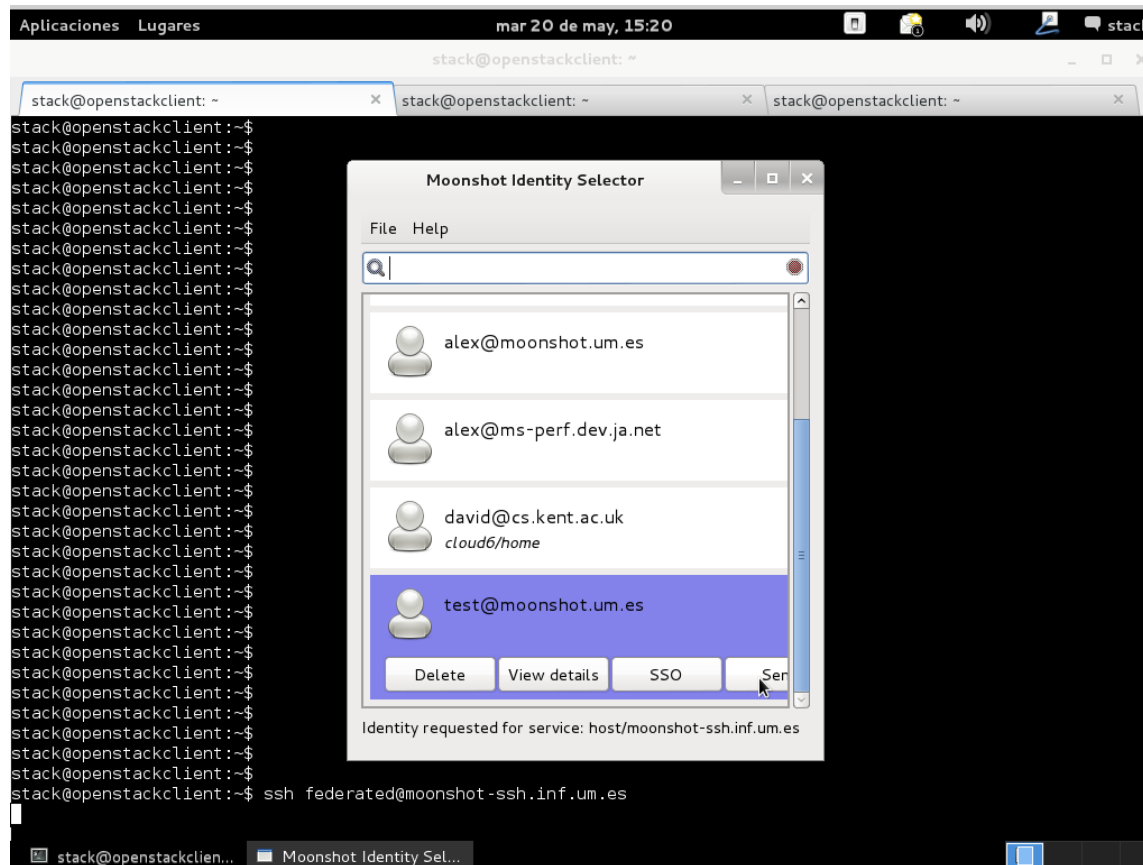
Desplegando Moonshot - EU

- Cualquier aplicación que soporte GSS-API debe funcionar con Moonshot
- Será necesario:
 1. Instalar código de Moonshot
 2. Intentar acceder al servicio
 3. Introducir o seleccionar la identidad a usar

Ejemplo 1: Cliente SSH

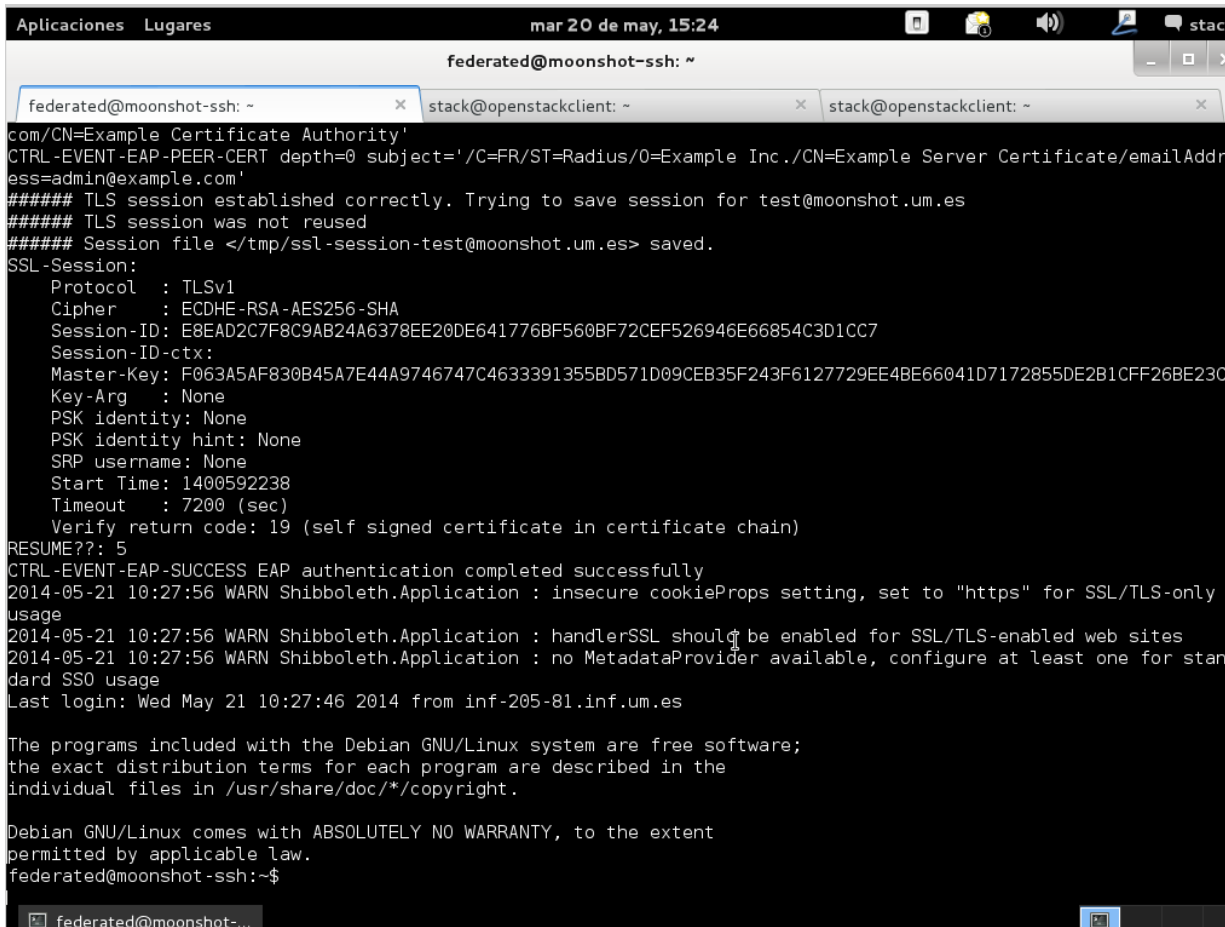
3. Seleccionamos la identidad a utilizar

- test@moonshot.um.es
- O cualquier otra identidad válida en eduroam



Ejemplo 1: Cliente SSH

4. Accedemos al servicio solicitado



```
mar 20 de may, 15:24
federated@moonshot-ssh: ~
com/CN=Example Certificate Authority'
CTRL-EVENT-EAP-PEER-CERT depth=0 subject='/C=FR/ST=Radius/O=Example Inc./CN=Example Server Certificate/emailAddress=admin@example.com'
##### TLS session established correctly. Trying to save session for test@moonshot.um.es
##### TLS session was not reused
##### Session file </tmp/ssl-session-test@moonshot.um.es> saved.
SSL-Session:
  Protocol : TLSv1
  Cipher   : ECDHE-RSA-AES256-SHA
  Session-ID: E8EAD2C7F8C9AB24A6378EE20DE641776BF560BF72CEF526946E66854C3D1CC7
  Session-ID-ctx:
  Master-Key: F063A5AF830B45A7E44A9746747C4633391355BD571D09CEB35F243F6127729EE4BE66041D7172855DE2B1CFF26BE23C
  Key-Arg  : None
  PSK identity: None
  PSK identity hint: None
  SRP username: None
  Start Time: 1400592238
  Timeout  : 7200 (sec)
  Verify return code: 19 (self signed certificate in certificate chain)
RESUME??: 5
CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
2014-05-21 10:27:56 WARN Shibboleth.Application : insecure cookieProps setting, set to "https" for SSL/TLS-only usage
2014-05-21 10:27:56 WARN Shibboleth.Application : handlerSSL should be enabled for SSL/TLS-enabled web sites
2014-05-21 10:27:56 WARN Shibboleth.Application : no MetadataProvider available, configure at least one for standard SSO usage
Last login: Wed May 21 10:27:46 2014 from inf-205-81.inf.um.es

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

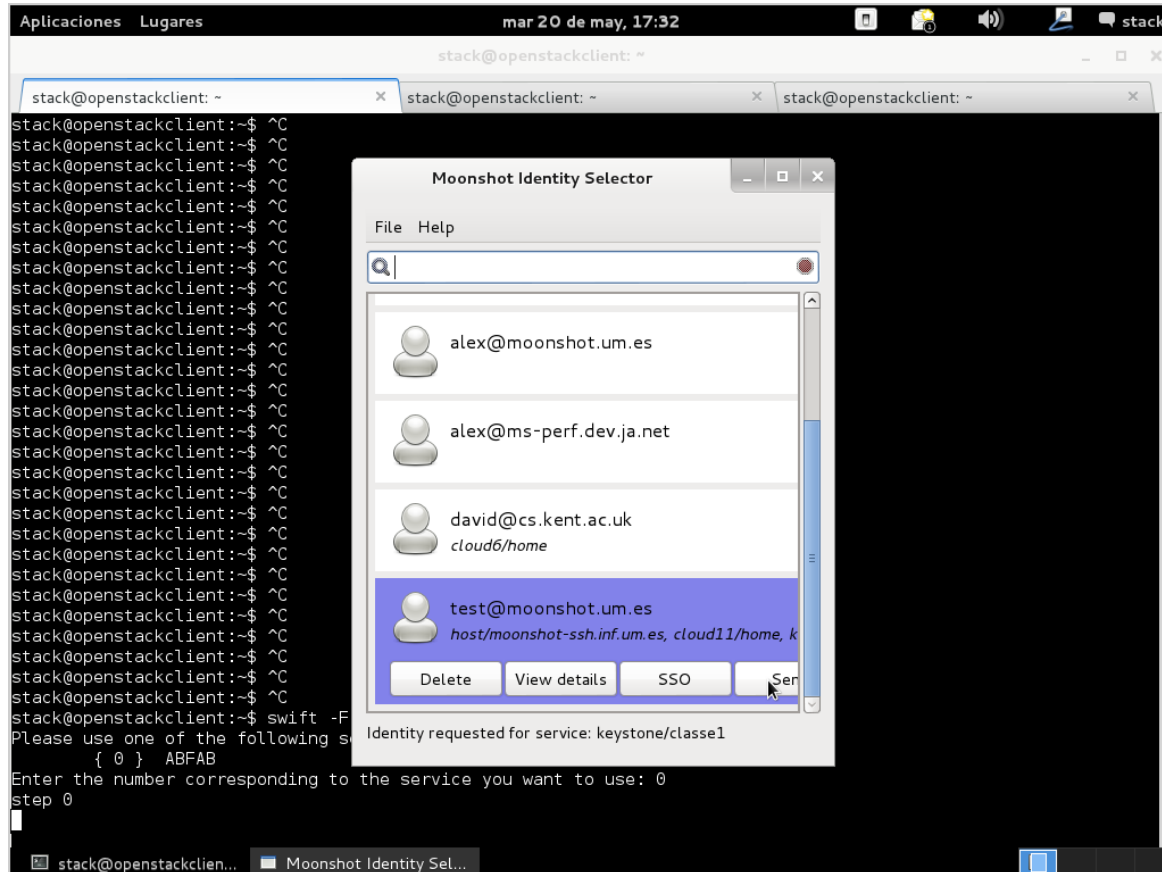
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
federated@moonshot-ssh:~$
```

Ejemplo 2: Cliente OpenStack

1. Instalamos Moonshot
2. Instalamos el cliente OpenStack con soporte para GSS-API
 - <http://sec.cs.kent.ac.uk/demos/keystone.html>

Ejemplo 2: Cliente OpenStack

5. Seleccionamos la identidad a utilizar
 - test@moonshot.um.es
 - O cualquier otra identidad válida en eduroam



Ejemplo 2: Cliente OpenStack

6. Obtenemos acceso al *tenant* correspondiente

```

Aplicaciones Lugares mar 20 de may, 17:33
stack@openstackclient: ~
stack@openstackclient: ~
stack@openstackclient: ~
CTRL-EVENT-EAP-METHOD EAP vendor 0 method 21 (TTLS) selected
##### Initializing TLS connection. Trying to load pre-established SSL session for: test@moonshot.um.es
##### Session file </tmp/ssl-session-test@moonshot.um.es> found. Loading TLS session.
{'negotiation': u'YIG9BgrBgEFBQ8BARIGAAoAAAAUAACoAQIAqBwAAAAAnhYDAQBeAgAAWgMBU3yAA7rgfhwLnSC7MquSUSKERDwK0udL9q
FG0tpST4YgpoiwhX1NuC1UEGvEgRfIGeS5rNKwLZYHDNI/fLTWP7DAFAAAEv8BAEAAAsABAMAAQIADwABARQDAQABARYDAQAw3mD9vtt2DljX/W
jT/9nnVGoIrr0olp0AsWSSiZKbyQMj7Hxav/Py6TX75/+YFH2t', 'cid': u'146637fca78d4ab9a372b254b3b9324b'}
step 3
##### TLS session established correctly. Trying to save session for test@moonshot.um.es
##### TLS session was reused
##### Session file </tmp/ssl-session-test@moonshot.um.es> saved.
SSL-Session:
  Protocol      : TLSv1
  Cipher       : ECDHE-RSA-AES256-SHA
  Session-ID  : A688B0857D4DB82D54106BC48117C819E4B9ACD2B09596070CD23F7CBB563FB0
  Session-ID-ctx:
  Master-Key  : C47E61D406180CB045901DAEA2A014A1946B3F41A5DB3779A6667DB677D66E331B83508DD02D5ABEED0BE267DD8FB64F
  Key-Arg     : None
  PSK identity: None
  PSK identity hint: None
  SRP username: None
  Start Time  : 1400599810
  Timeout     : 7200 (sec)
  Verify return code: 19 (self signed certificate in certificate chain)
{'negotiation': u'YBkGCSsGAQUFDwEBEGYCAAAABQAAAAQDAgAE', 'cid': u'146637fca78d4ab9a372b254b3b9324b'}
step 4
CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
{'negotiation': u'YDkGCSsGAQUFDwEBEGYCAAAAaAwAAABBrZXlzdG9uZS9jbGZzc2UxgAAADgAAAAzu/KKtxL2ZXieq7n0=', 'cid': u'14
6637fca78d4ab9a372b254b3b9324b'}
step 5
Authentication successful using "test@moonshot.um.es" moonshot identity.

You have access to the following tenant(s)and domain(s):
  { 0 } swifttenanttest1
Enter the number corresponding to the tenant you want to use: █
stack@openstackclien...

```

Muchas gracias por su
atención

¿Alguna pregunta?