



Federated SSH access

Is that possible?

Cándido Rodríguez

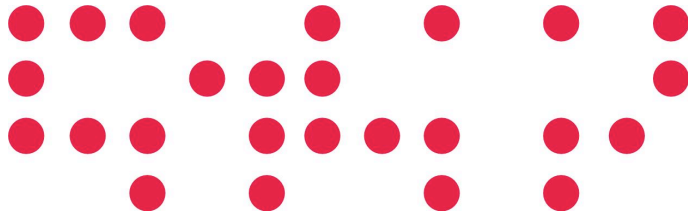
candido.rodriguez@rediris.es

1. What and why?

2. Architecture

3. Web application for access management

4. Conclusions & future work



- **Objective**

- Federated SSH access to our hosts
 - Compatible with different AAI technologies: SAML, PAPI, X.509,...

- **Why?**

- Current methods for SSH access are based on the user himself, not on his attributes
- There is no interconnection with enterprise directories
- The updates of the authorization_keys files are not simple

- **Acknowledgement**

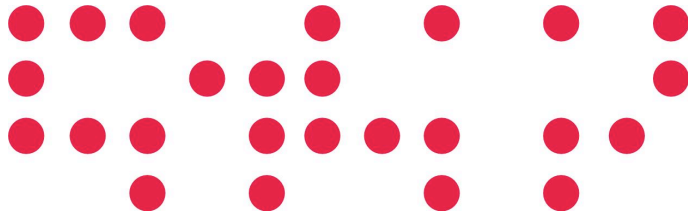
- Some ideas has been taken from Andreas Solberg (FEIDE) and Daniel García (University of Seville)

1. What and why?

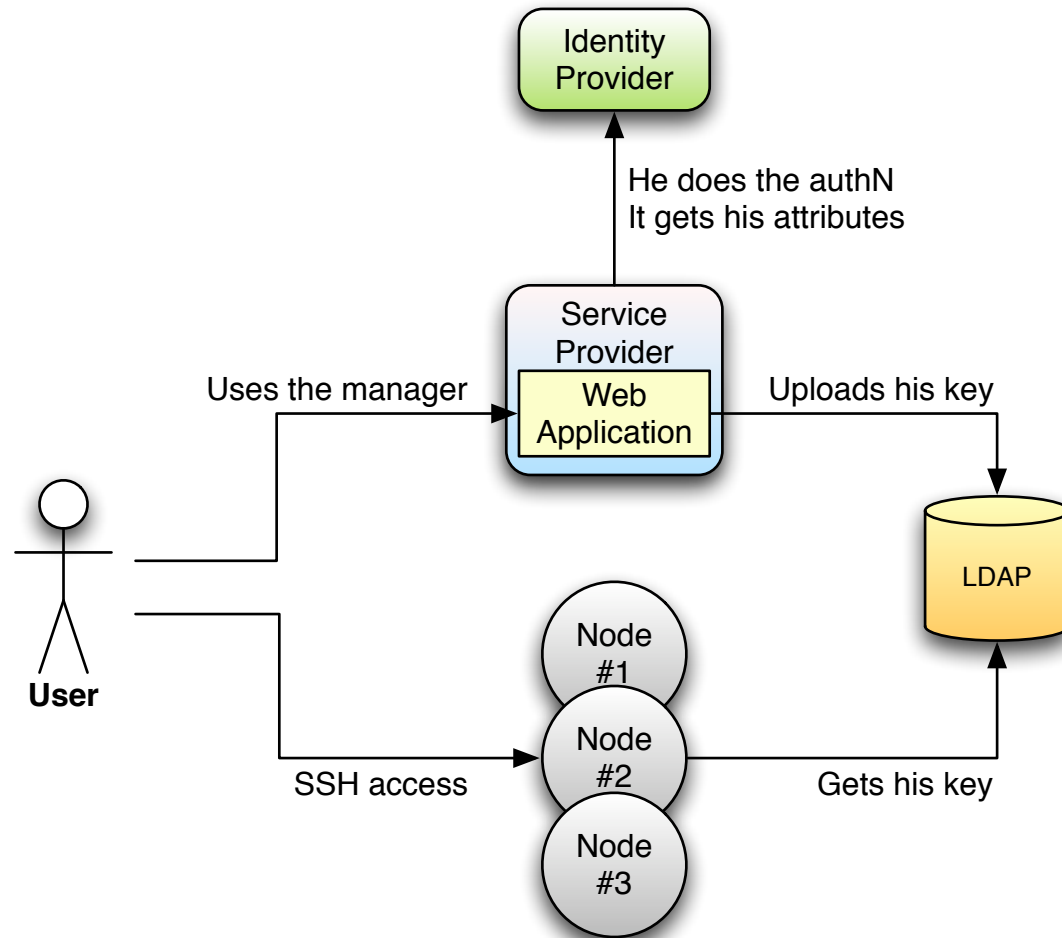
2. Architecture

3. Web application for managing the access

4. Conclusions & future work



- The scenario...



- Storing keys into a LDAP server

- It requires the following (not so common) schemas:
 - SCHAC
 - Available at <http://www.terena.org/activities/tf-emc2/schacreleases.html>
 - iris
 - Available at <http://www.rediris.es/ldap/schema/iris.schema>
 - openssh-lpk
 - Available at <http://code.google.com/p/openssh-lpk/downloads/list>

```
attributetype ( 1.3.6.1.4.1.24552.500.1.1.1.13 NAME  
'sshPublicKey'  
DESC 'MANDATORY: OpenSSH Public key'  
EQUALITY octetStringMatch  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
```

```
objectclass ( 1.3.6.1.4.1.24552.500.1.1.2.0 NAME  
'ldapPublicKey' SUP top AUXILIARY  
DESC 'MANDATORY: OpenSSH LPK objectclass'  
MAY ( sshPublicKey $ uid )  
)
```

- The following information is kept by the LDAP server:
 - User's rights for accessing a node

Attribute	LDAP attribute
Requester ID	uid
Public key	sshPublicKey
SSH URI access	schacUserStatus
Expiry date	shachExpiryDate

- SSH URI is defined by an URN:
 - urn:mace:terena.org:schac:userStatus:es:rediris.es:sshaccess:**node.rediris.es**
%3A22+user=root

- How to get the keys through LDAP?
 - OpenSSH doesn't support LDAP connections
 - Two different approaches:
 - A patched OpenSSH which is able to get keys through LDAP
 - Develop a software which somehow connects OpenSSH to LDAP
- Patched OpenSSH
 - There is one available at <http://code.google.com/p/openssh-lpk/>
 - It's a big patch (60 Kb)
 - You need to store additional information about the user
 - **This solution dropped!**

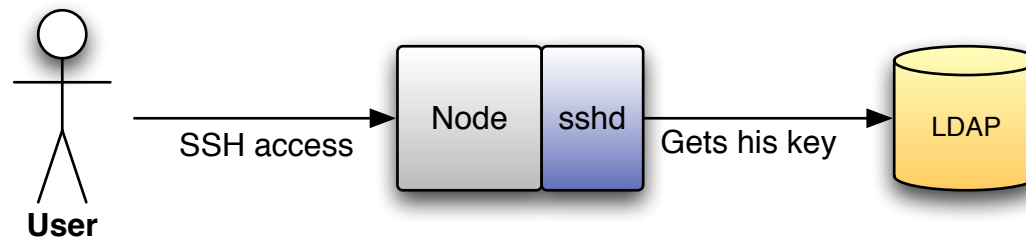
- Proposed patch for OpenSSH

- We've developed one
 - Much lighter: only 10 Kb!
 - Add more configuration parameters
 - Small function for getting the keys
- We've made available...
 - The patch
 - A Debian package containing a patched openssh-server
- Advantages
 - sshd is able to get keys through LDAP in real time
- Disadvantages
 - What if you **cannot** or you **do not want to** replace an standard SSH installation

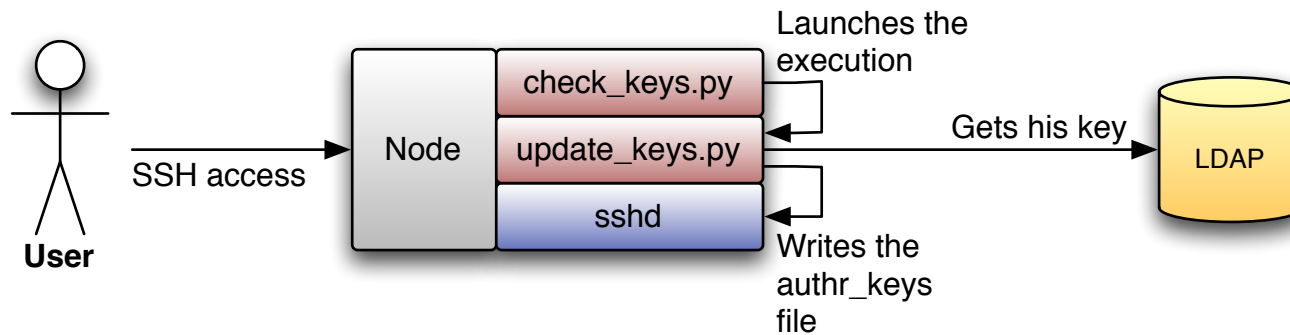
- A solution without modifying OpenSSH?
 - update_keys.py
 - Script which gets valid keys from LDAP
 - It updates the authorized_keys file
 - You should add a cron entry for this script
 - Executed every 1 or 2 minutes
 - check_keys.py
 - Script which starts the first one when a HTTP message is received
 - RESTful service through an embedded mini-webserver
 - When it receives a request for \$webserver:\$port/setkeys
 - It calls the first script

- Comparing two models...

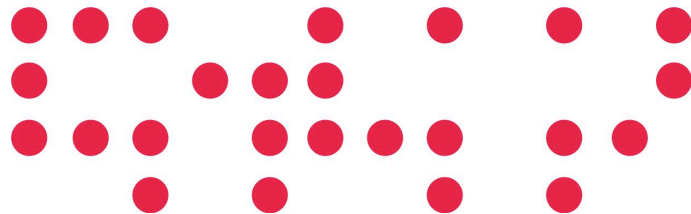
- Patched OpenSSH



- Non patched OpenSSH



1. What and why?
2. Architecture
- 3. Web application for access management**
4. Conclusions & future work



- Web 2.0 application developed in PHP and JQuery
- It is a federated application
 - Designed to be easily compatible with different AAI technologies
 - SAML 1.1, SAML 2, OpenID, X.509, PAPI...
 - It gets user's **attributes** from the user's IdP
 - Even the public keys for SSH!
 - And users can directly upload their public keys
- It allows users to upload their keys into LDAP
 - Not only based on who is... but also based on his attributes
 - A **policy** for defining who is allowed to upload keys
- It supports groups of nodes
 - Uploading public keys for all nodes of a **group**

- How does a policy look like?

- Definition for a single node:

```
<SSHServer host="172.16.202.128" port="22">  
  <CheckStatus value="urn:mace:rediris.es:check:ws:http%3A//  
172.16.202.128%3A8888/setkeys+cron=10/0" />  
  <Policies>  
    <Policy user="root" timeout="30">  
      <Attributes check="none" ><!-- all, any or none -->  
        <Attribute name="cn" value="John" />  
      </Attributes>  
    </Policy>  
    <Policy user="test" timeout="30">  
      <Attributes check="all" ><!-- all, any or none -->  
        <Attribute name="cn" value="John" />  
      </Attributes>  
    </Policy>  
  </Policies>  
</SSHServer>
```

- How does a policy look like?

- Definition for a group:

```
<SSHGroup id="admin">
  <Backend value="ldap">
    <Server host="localhost" port="389" />
    <BaseDN value="dc=groups,dc=myserver,dc=org" />
    <BindDN value="uid=admin,dc=myserver,dc=org" password="x" />
  </Backend>
  <Policies>
    <Policy user="root" timeout="20">
      <Attributes check="all" ><!-- all, any or none -->
        <Attribute name="organization" value="rediris.es" />
        <Attribute name="eduPersonAffiliation" value="student" />
      </Attributes>
    </Policy>
  </Policies>
</SSHGroup>
```

- How are groups represented in LDAP?
 - Group definition

Attribute	LDAP attribute
ID group	cn

- Node in a group
 - Its parent entry is the group entry

Attribute	LDAP attribute
Server name	ipHostNumber
Port	ipProtocolNumber
Check mode	schacUserStatus

FedSSH: the web application

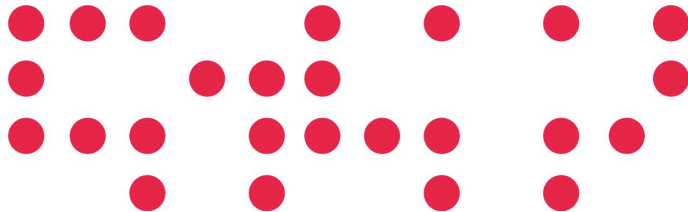


- Demo!

The screenshot shows the FedSSH web application interface. At the top, there is a navigation bar with 'Fed.SSH' and links for 'Home', 'Servers', 'Keys', and 'Help'. On the right, a user profile for 'kan' is shown with '2 keys'. Below the navigation bar is a table with the following data:

Server	Status	Available connections
homer.rediris.es	Fail	Send key for user perfsnar
172.16.96.128	Patched	Send key for user root
Group admin	3 (green) 0 (red)	You're not allowed to log in as user root You're not allowed to log in as user infoadmin You're not allowed to log in as user test
node01.rediris.es	Patched	
node02.rediris.es	Patched	
node03.rediris.es	Patched	

1. What and why?
2. Architecture
3. Web application for access management
- 4. Conclusions & future work**



- **New tool for administration the SSH access of our hosts**
 - A policy for defining the user's rights
 - Grouped hosts for making their administration easier
 - Decisions based on user's attributes
- **OpenSSH is able to connect through LDAP without modifying its source code**
 - Less time maintaining the authorized_keys file in our infrastructure
- **Federated environment**
 - Allowing access to external users
 - Even if we don't know them!

- Future

- All information available at <http://www.rediris.es/sshfed/> on 1st March
- More patched distributions packages
 - CentOS, Red Hat, Ubuntu...
- An administration area
 - Policy editor
 - Group editor
- An schedule component
 - Supporting **slot reservations**
- Thinking on new ways of connecting OpenSSH and LDAP



MINISTERIO
DE INDUSTRIA, TURISMO
Y COMERCIO

red.es



MINISTERIO
DE INDUSTRIA, TURISMO
Y COMERCIO

red.es

Edificio CICA, Campus Universitario
Avenida Reina Mercedes s/n
41012 Sevilla. España

Tel.: 95 505 66 00
Fax: 95 505 66
www.red.es

