



SPANKEY SSH KEY MANAGEMENT QUICK START

The specifications and information in this document are subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. This document may not be copied or distributed by any means, in whole or in part, for any reason, without the express written permission of RCDevs.

Copyright (c) 2010-2017 RCDevs SA. All rights reserved.

<http://www.rcdevs.com>

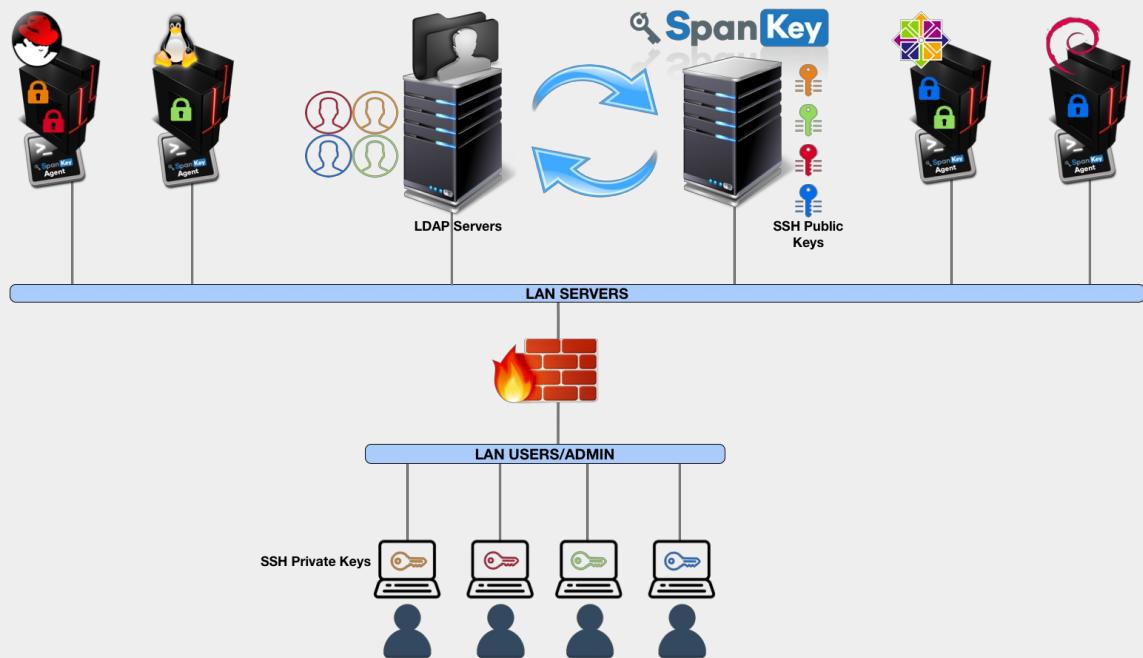
WebADM and OpenOTP are trademarks of RCDevs. All further trademarks are the property of their respective owners.

Limited Warranty

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to info@rcdevs.com.

SpanKey SSH Key Management Quick Start

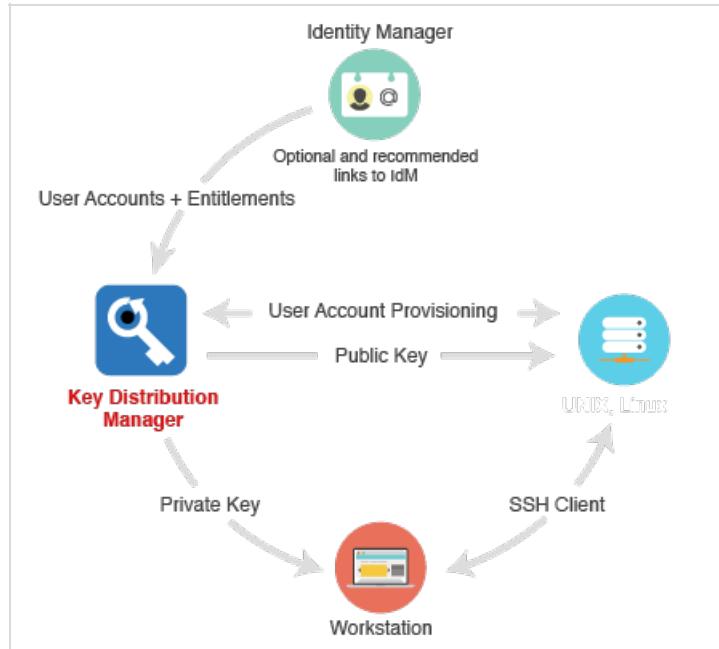
PAM OpenSSH NSS



1. Overview

SpanKey is a centralized SSH key server for OpenSSH, which stores and maintains SSH public keys in a centralized LDAP directory (i.e. Active Directory). With SpanKey there is no need to distribute, manually expire or maintain the public keys on the servers. Instead, the SpanKey agent is deployed on the servers and is responsible for providing the users' public keys on-demand. The SpanKey server provides per-host access control with “server tagging”, LDAP access groups, centralized management from the RCDevs WebADM console, shared accounts, privileged users (master keys), recovery keys... It supports public key expiration with automated workflows for SSH key renewal (via Self-Services). For information on SpanKey, please visit [RCDevs Website](#).

For this recipe, you will need to have WebADM installed and configured. Please, refer to [WebADM Installation Guide](#) and [WebADM Manual](#) before installing SpanKey server. SpanKey server should be installed on the WebADM server.



2. Packages Installation

2.1 RHEL & CentOS through RCDevs Repository

2.1.1 Add RCDevs Repository

On a RedHat, CentOS or Fedora system, you can use our repository, which simplifies updates. Add the repository:

```
yum install https://www.rcdevs.com/repos/redhat/rcdevs_release-1.0.0-0.noarch.rpm
```

Clean yum cache:

```
yum clean all
```

You are now able to install RCDevs packages on your system.

2.1.2 SpanKey Server Installation

```
yum install spankey
```

After the Spankey server installation, you need to restart WebADM services:

```
/opt/webadm/bin/webadm restart
```

To enable SpanKey web service, you need to login on the WebADM GUI. Under **Applications** tab, click **Authentication** in category box and you should find **SSH Public Key Server (SpanKey)**. Click on **REGISTER** button.

2.1.3 SpanKey Client and NSCD Installation

```
yum install spankey_client nsqd
```

The SpanKey client requires nsqd and OpenSSH. NSCD is the Linux name service caching daemon which is required for caching NSS information on the Linux client. Without NSCD, any user or group ID resolution will trigger SpanKey NSS requests. Caching on the client side will prevent your servers from being overloaded with NSS requests.

Note

Be aware that at least OpenSSH 6.2 is needed. (Added a sshd_config option AuthorizedKeysCommand to support fetching authorized_keys from a command in addition to (or instead of) from the filesystem.)

2.2 Debian & Ubuntu through RCDevs Repository

2.2.1 Add RCDevs Repository

On a Debian system, you can use our repository, which simplifies updates. Add the repository:

```
wget https://www.rcdevs.com/repos/debian/rcdevs-release_1.0.0-0_all.deb  
apt-get install ./rcdevs-release_1.0.0-0_all.deb
```

Clean apt cache:

```
apt-get update
```

You are now able to install RCDevs packages on your system with apt-get command.

2.2.2 SpanKey Server Installation

```
apt-get install spankey
```

After the Spankey server installation, you need to restart WebADM services:

```
/opt/webadm/bin/webadm restart
```

To enable SpanKey web service, you need to login on the WebADM GUI. Under **Applications** tab, click **Authentication** in category box and you should find **SSH Public Key Server (SpanKey)**. Click on **REGISTER** button.

2.2.3 SpanKey Client and NSCD Installation

```
apt-get install spankey-client nscd
```

The SpanKey client requires nscd and OpenSSH. NSCD is the Linux name service caching daemon which is required for caching NSS information on the Linux client. Without NSCD, any user or group ID resolution will trigger SpanKey NSS requests. Caching on the client side will prevent your servers from being overloaded with NSS requests.

Note

Be aware that at least OpenSSH 6.2 is needed. (Added a sshd_config option AuthorizedKeysCommand to support fetching authorized_keys from a command in addition to (or instead of) from the filesystem.) With Ubuntu servers, depending on your OS setup, you may need to install libldap as well.

2.3 Installation Using the Self-Installer

You first need to download the Spankey software package. You can download the latest package on the [RCDevs Website](#). Download and copy the SpanKey server self-installer package to your server. You can copy the package file to the server with WinSCP or SCP. Then connect via SSH to your server, uncompress and run the self-installer package with:

```
gunzip spankey-2.0.x-x.sh.gz  
bash spankey-2.0.x-x.sh
```

Follow the installer.

For the SpanKey client:

```
gunzip spankey_client-2.1.x.sh.gz  
bash spankey_client-2.1.x.sh
```

Follow the installer and don't forget to install the NSCD package.

3. Configurations

3.1 SpanKey Server

Once SpanKey server package is installed, you have to enable SpanKey service in WebADM. Go to the WebADM Administrator console, click on **Applications** tab > **Authentication** and click on **Register** button for **SSH Public Key Server**. The default configuration is ready and suited for most Linux environments, but for initial tests, it is recommended to click on **CONFIGURE** button and set the following options in SSH Public Key Server (SpanKey server):

The screenshot shows the WebADM Freeware Edition interface. On the left, there's a sidebar titled 'LDAP Server (OpenLDAP)' with a refresh icon. It lists 'OpenLDAP (2)', 'dc=WebADM', and 'o=Root (3)' which contains 'cn=admin', 'cn=ppolicy', and 'cn=test_user'. Below these are 'Create / Search' and 'Details / Check' links for each item. On the right, the main area is titled 'WebADM Freeware Edition v1.6.8-4' with the copyright notice 'Copyright © 2010-2018 RCDevs SA, All Rights Reserved'. A navigation bar includes Home, Admin, Create, Search, Import, Databases, Statistics, Applications, About, and Logout. Below the navigation is a section titled 'Misc Options' containing two settings: 'SSH Cache Time' (checkbox checked, value 0) and 'NSS Cache Time' (checkbox checked, value 0). Each setting has a descriptive note below it.

Misc Options
<input checked="" type="checkbox"/> SSH Cache Time 0 Key cache time for authorized and master group members (in minutes). SSH cache is stored in WebADM Session Server. You need to clear session data to purge SSH cache. Set '0' to disable caching (not recommended).
<input checked="" type="checkbox"/> NSS Cache Time 0 Cache time for NSS users and groups (in minutes). NSS cache is stored in WebADM Session Server. You need to clear session data to purge NSS cache. Set '0' to disable caching (not recommended).

This will disable server caching, generally helpful during configuration stage and tests.

⚠ Important note

For production server caching is highly recommended.

LDAP Server (OpenLDAP)

WebADM Freeware Edition v1.6.8-4
Copyright © 2010-2018 RCDevs SA, All Rights Reserved

Home | Admin | Create | Search | Import | Databases | Statistics | Applications | About | Logout

Server Policy

SSH Key Format RSA (Default)

RSA is recommended because other key types cannot be exported for use with PuTTY.
ECC (Elliptic Curve) is a new standard which uses much smaller key sizes.
DSA support is limited to 1024 bits keys and is deprecated in OpenSSH servers.

RSA Key Length 2048 (Default)
2048 bits is recommended for SSH usage.

ECC Key Length 256 (Default)
256 bits is recommended for SSH usage.

Key Lifetime 360
Time after which a key expires and must be re-registered (in days).
Set '0' to disable the expiration on newly registered keys.

Enable Offline Mode Yes No (default)
Cache authorized keys and NSS data for offline use when SpanKey server is down.

Allow Password Change Yes No (default)
Allow self LDAP password change with the usual 'passwd' Linux command.
This feature will be implemented in SpanKey client v2.0.2.

Require Extra Login Factors OTP
Enable additional multi-factor authentication with OpenOTP.
Note: SCP and non-interactive sessions support OTP with Push only.

Allowed Local Users root
Comma-separated list of users for which the usual SSH authorized keys files are allowed.
For these users both centrally-managed public keys and local authorized keys files can be used.

Authorized Key File(s) .ssh/authorized_keys
Comma-separated list of authorized keys file(s) on the SSH hosts for the local users.

- > The SSH Key format can be defined here.
- > RSA Key Length can also be settled here.
- > The SSH Key Lifetime can be adjusted too.
- > Send Self-Registration: This option can be enabled if you want to have a new self-registration request when the SSH key has expired.
- > Enable Offline Mode: Offline mode can be enabled in case of the SpanKey server is unavailable.
- > Require Extra Login Factors: An OTP validation can be added during the authentication workflow.

Some other settings can be enabled on Spankey server:

LDAP Server (OpenLDAP)

- OpenLDAP (2)
 - dc=WebADM
- o=Root (3)
 - cn=admin
 - cn=ppolicy
 - cn=test_user

WebADM Freeware Edition v1.6.8-4
Copyright © 2010-2018 RCDevs SA, All Rights Reserved

[Home](#) | [Admin](#) | [Create](#) | [Search](#) | [Import](#) | [Databases](#) | [Statistics](#) | [Applications](#) | [About](#) | [Logout](#)

UNIX Account Options

Create Home Directories Yes No (default)
Automatically create the user's home directory if not present.

Minimum UID Number
Users with UID number below the value are ignored.

Minimum GID Number
Groups with GID number below the value are ignored.

Session Options

Record Session Data Yes No (default)
Stores the terminal and SCP session information in WebADM Record database.
- Terminal sessions are recorded as replayable videos.
- SFTP sessions are recorded as event logs.

Max Session Time
Automatically close SSH sessions after the configured time (in minutes).
Use '0' to disable automatic session expiration.

Screen Lock Time
Automatically lock SSH screen if idle for the configured time (in minutes).
Use '0' to disable session lock time.

Welcome Message

Message to be displayed in the terminal session.

- › **Create Home Directory:** If enabled, the user home directory will be automatically created during the first login if not present.
- › **Record Session Data:** This is a new feature of SpanKey! This setting allows you to record and store in SQL database, terminal sessions, SFTP sessions. Sessions are replayable video which can be found in **Databases** tab > **Recorded Sessions** under WebADM Admin Console.

LDAP Server (OpenLDAP)

- OpenLDAP (2)
 - dc=WebADM
- o=Root (3)
 - cn=admin
 - cn=ppolicy
 - cn=test_user

WebADM Freeware Edition v1.6.8-4
Copyright © 2010-2018 RCDevs SA, All Rights Reserved

[Home](#) | [Admin](#) | [Create](#) | [Search](#) | [Import](#) | [Databases](#) | [Statistics](#) | [Applications](#) | [About](#) | [Logout](#)

SQL Data Tables

- Localized Messages**
Message translations for applications
- Inventoried Devices**
Tokens and other inventories
- Recorded Sessions**
SpanKey session records and command logs

- › **Max Session Time:** This setting can be settled if you want to define a maximum session time.

Under SSH Public Key Server configuration, you can find various configurations options to set access controls to your SSH key-based logins, such as Master Group, Backup Keys, Authorized Group, Tagging... Some of these settings are described in the chapter “Advanced Configuration”.

⚠ Important Note

Require client certificate for SpanKey client is highly recommended for production use!

⚠ Important Note

If you enable this option, every SpanKey client who actually works without a client certificate will stop working. To solve this, you can generate a client certificate through WebADM Admin GUI > Admin tab > Issue Server or Client SSL Certificate and import the generated certificate in /opt/spankey/conf/ folder of your SpanKey client.

LDAP Server (OpenLDAP)

- OpenLDAP (2)**
 - dc=WebADM**
 - o=Root (3)**
 - cn=admin**
 - cn=ppolicy**
 - cn=test_user**

Create / Search

Details / Check

Main information

Client Name or Description:

Certificate Type: ⓘ

Restricted Application: ⓘ

Certificate validity (in days): ⓘ

Private Key Password (optional):

Additional information

Organization Name:

Organizational Unit:

Country Name: ⓘ

Locality Name:

State or Province:

Street Address:

Email Address:

Ok **Cancel**


LDAP Server (OpenLDAP)

WebADM Freeware Edition v1.6.8-4

Copyright © 2010-2018 RCDevs SA, All Rights Reserved

 Home
 Admin
 Create
 Search
 Import
 Databases
 Statistics
 Applications
 About
[Logout](#)

Create Third-party SSL Server Certificate

Creating private key... Success

Certificate details:

- commonName: **test.domain.com**
- description: **CLIENT:spankey**
- organizationName: **RCDevs**
- organizationalUnitName: **IT**
- countryName: **LU**
- localityName: **Belval**
- stateOrProvinceName: **Luxembourg**

Creating a certificate request based on the above details... Success

Calling WebADM CA for certificate request signing... Success

Private Key (PEM format):

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiGw0BAQFASCBKcwggsJAgEAAoIBAQDBZ6o40tG5iyiq
Y7p5RLtTl/+B/jaQFamgdjfzjcklKEFuqkGrh/lKyfhaenRxeBa1gRz09gNKi5
vgrbRJ1FRucdpuzvyxSxXSndtj0cm+zuWfH2XjrtTaYn4Pat62WgoNu6eNkpVq10z
1Ji31Ey6D8FyquzO2x5JVNgNmN+paLd/LxsIyj8DSj9x0f0vlpMT6zYIraliHE
Ezm/rW9T329oibqUSvWOMhPg1rPfmhvRtTSIKooSk3MVgEJnsfoP8F7f/139umb
KcZED0VaPrCOHQii2oo4VGstKGf+7LGdC8jKzQdjd2zjzbnd9rg2d/ZFciQmRl
jNUJ41IJAgMBAAECggEAoSMbNkyL3WgfWSfi+zluzQ85in16hnyOa7j0aeQ95ld
iNSNph6WuwSpbi7FqYTpskEam4HwmQJm4UhkdAp7pxjLz66P/qttNh2SgS6AJc/
fYvgd4yxwtomGU5jc4fcgY1LlrMdWtp3f2SCPT9+uei1BqoS9afHdszYpmgGu7
wGbclrTNncTDXONRfH2EA/XYqkip46xd9uQawDPNyoy2Vb76zw+/E4aLBpXga
qRedMbOE0KA17v2b0yzhDqngGA5/dwOWehb3Kj/0Lcp1d6F6iD0w7af8NgknUbC
acc7VsU7e1VPHU4Q0RJdYM+4tP2LV27o65j6aVjoQKBqQDq7/Yhp/p5aAtoOU
X09sIFFj9fL6pm1W30H7way/Zttq1kgYhsGPno/jitUQBhlajqf1AG/aact3dfS
AzeY8729JwwbGkellLzSuzsNTsXTeGck5DcKD6anc5wCvm0yulFrXTx5snek5WM7
08NOLUh82ZEZKACdzjdzjzWTwKBgDectedfna5zE7NNTfd1QVW/NWQ61rjoxA
yDXV6fpvgnY3kZf2Lzt+CayftSxwLydYTrxwSVQGqc9Nk6eyHer6B4AyvxCR8e
ndvUHUGzHKQZk1Dj6JCpcRQaYnMjKbc72WiJMYlkmA574b4mN/jBq+HD8OzQj4Qz
bZH1h8sY5wKBgEsgubNzRww9Y1Cc/4j378ggNvGviSNTH5+80H5y12Njkhnvw0UK
Zieh3gw+1EEH+ule+hdY/A+Gy6Z8zbu0/D+apeeH+a6Lw4h1H11jwdpjxgxxevp
D17YHCbUNN2t5Or+hqqiUGwOwX272FPv2KqdMawTEUpNTSa4dTZi+scHaoCBAJu7
```

Certificate (PEM format):

```
-----BEGIN CERTIFICATE-----
MIIDIjCCAgggAwIBAgIBAzANBgkqhkiGw0BAQsFADAKMRIwEAYDVQDDAlXZWJB
RE0gQ0ExDjAMBgNVBAoMBUxvY2FsMB4XDTE4MT1xNzE1Mzg1M1oXDTE5MTIxNzE1
Mzg1M1owgYQxGDAWBgNVBAMMD3Rlc3Qu2G9tYWluLmNvbzdjzjdGA1UEDQwOQ0xJ
RU5UOnNwY5rZXkxdzAnBgNVBAoMB1JDRGVczcELMAkGAIUECwvCSVQxZcA1BgNV
BAYTAkxVMQ8wQDYDVQHQDAZCZwx2YwxEzARBgNVBAgMCxk1eGvTYm91cmcwggEi
MA0GCSqGSIb3DQEBehrnhtBDwAwggEKAoIBAQDBZ6o40tG5iyiqY7p5RLtTl/+B/jaQFamBLeeewucklKEFuqkGrN/lKyfhaenRxeBa1gRz09gNKi5vgrbRJ1FRucd
puzvyxSxXSndtj0cm+zuWfH2XjrtTaYn4Pat62WgoNu6eNkpVq10z1Ji31Ey6D8Fy
quz02x5JVNgNmN+paLd/LxsIyj8DSj9x0fonkezT6zYIraliHEzm/rW9T329o
ibqUSvWOMhPg1rPfmhvRtTSIKooSk3MVgEJnsfoP8F7f/139umbKcZED0VaPrC
OHQii2oo4VGstKGf+7LGdC8jKzQdjd2zjzbnd9rg2d/ZFciQmRljNUJ41IJAgMB
AAEwDOQJKoZihvcNAQELBQAQdgEBAkbsotTXJXvepl9itiG+AJR6zvbJNMCG9x
OF767d9BI+X3+bplTU5Hf8yFc+3wBKCBs7dzuzz9uWGN/gtK3x8hLPmRLv6nVYoA
UIKbmz+h3Ksz1mdsP+wMeexQ7W05sVSCY7gnbeX0WkqmBhuj19zMfdBzIC1kdaixF
VU903csfEOGql2uCrH+rbaqTFHMVudcGygJN8FUgJpW3W6SbkUTnETCmXMG9njRCb
mgyo5064ivs+zdfFOatSH1MCszydbTLhyK1EWfncCWQi20k5v2/xsGJn7UdrDrSz
awpX79wSF4vy+Ro61CLqif0uwupEB4kf2yfDQI4sBGc1lQ71NiQ=
-----END CERTIFICATE-----
```

[Download Cert & Key File](#)
Ok

3.2 SpanKey Client

The SpanKey client consists of two components activated at setup time.

- > SSH component - provides a user login with public keys stored within a directory server (Active Directory, OpenLDAP, Open Directory...).
- > NSS component - provides a native mapping of your directory users and groups to those in Linux.

3.2.1 SpanKey Client Setup Script

At the end of the installation of the SpanKey package, run the following command to launch setup wizard:

`/opt/spankey/bin/setup` The wizard will prompt you for the details similar to below:

```
[root@spankey_client ~]# /opt/spankey/bin/setup
Setup has already been run for this installation. Overwrite (y/n)?: y
Overwriting...
Enter one of your running WebADM node IP or hostname []: 192.168.3.117
Do you want to enable SpanKey Client for OpenSSH server (y/n)? [N]: y
Do you want to enable SpanKey Client NSS plugin (y/n)? [Y]: y
Do you want to register SpanKey Client logrotate script (y/n)? [Y]: y
Do you want SpanKey Client to be automatically started at boot (y/n)? [Y]: y

Primary OpenOTP service URL is: 'https://192.168.3.117:8443/spankey/'
Enable SpanKey Client for OpenSSH server: 'YES'
Enable SpanKey Client NSS plugin: 'YES'
Register SpanKey Client logrotate script: 'YES'
SpanKey Client must be automatically started at boot: 'YES'

Do you confirm (y/n)?: y

Applying SpanKey Client settings from default configuration files... Ok
Retrieving WebADM CA certificate from host '192.168.3.117'... Ok
The setup needs now to request a signed 'SpanKey' client certificate.
This request should show up as pending in your WebADM interface and an administrator
must accept it.
Waiting for approbation...
```

At this step, you have to log in on the WebADM Administration GUI to approve the SSL certificate request.

WebADM Freeware Edition v1.6.8-4
Copyright © 2010-2018 RCDevs SA, All Rights Reserved

Home | Admin | Create | Search | Import | Databases | Statistics | Applications | About | Logout

Hello Admin ([cn=admin,o=Root](#))
Connected as Super Administrator to rcvm7.local

Application Status

OpenID & SAML Provider:	Not Configured
Secure Password Reset:	Ok (v1.0.12)
User Self-Service Desk:	Ok (v1.1.8)
User Self-Registration:	Ok (v1.1.8)
MFA Authentication Server:	Ok (v1.4.2)
Single Sign-On Server:	Ok (v1.0.8)
SMS Hub Server:	Ok (v1.1.2)
SSH Public Key Server:	Ok (v2.0.2-1)
QR Login & Signing Server:	Ok (v1.2.5-3)

Configurations Objects

Local Domains: 1 (Details)	Trust Domains: 0 (Details)
Mount Points: 0 (Details)	Option Sets: 1 (Details)
Client Policies: 0 (Details)	Admin Roles: 1 (Details)

[WebADM] [2018-12-17 17:44:30] [rcvm7.local] New pending server/client certificate requests (1)

Click Here For Details

Click on the red button at the end of the home page.

On the next screen, you can show the SSL certificate request is pending:

The screenshot shows the WebADM interface. On the left sidebar, there's a tree view under 'LDAP Server (OpenLDAP)' with nodes like 'dc=WebADM', 'o=Root (3)', and 'cn=admin'. Below the sidebar, the main content area has a title 'SSL Certificate Requests' and a message: 'Find below the pending certificate requests send to the WebADM certificate generation API. Found 1 pending server SSL certificate requests:'. A table lists one request: 'ubuntu18client-virtual-machine' (Client type, 192.168.3.178 source, 17:12:30 received, 250 secs expires in, SpanKey application, Pending status). Buttons for 'Accept' and 'Reject' are shown. At the bottom, a yellow banner says '[WebADM] [2018-12-17 17:44:30] [rcvm7.local] New pending server/client certificate requests (1)' with a 'Click Here For Details' button.

Click on the Accept button and the Spankey-client setup will continue.

This screenshot is similar to the previous one but shows the 'Accepted' status for the pending request. The table now shows 'ubuntu18client-virtual-machine' with a 'Status' column value of 'Accepted'. The 'Accept' and 'Reject' buttons are still present.

```
Waiting for approbation... 0k
Updating entry 'client_id' in file '/opt/spankey/conf/spankey.conf'... 0k
Updating file '/etc/ssh/sshd_config'... 0k
Updating file '/etc/nsswitch.conf'... 0k
Updating file '/etc/pam.d/password-auth'... 0k
Registering SpanKey Client service...
Registering SpanKey Client service... 0k
Adding logrotate script... 0k
```

SpanKey Client has successfully been setup.

IMPORTANT: Do not forget to perform the following actions before you exit this session:

- Start SpanKey (/opt/spankey/bin/spankey start)
- Restart 'sshd'
- Restart 'nscd'

The configuration of the SpanKey client is done, you have to restart sshd, nscd and spankey-client:

```
[root@spankey_client ~]# systemctl restart sshd
[root@spankey_client ~]# systemctl restart nscd
[root@spankey_client ~]# systemctl start spankey
```

SpanKey client setup is done.

3.2.2 SpanKey Client silent installation

Since WebADM 1.7.1, a new feature is now available for the automatic certificate approval. This setting can be useful when you massively deploy SpanKey Client. To enable this feature, log in on the **WebADM Admin GUI > Admin tab > Runtime Actions > Issue Server or Client SSL Certificate > Auto Confirm Mode**.

Create Third-party SSL Server Certificate

Auto Confirm Mode

Enable Auto Confirm: Yes No [i](#)

Auto Confirm Time: 30 Minute [i](#)

Auto Confirm App: SpanKey [i](#)

Auto Confirm IPs: 192.168.3.0/24 [i](#)

Main information

Server Hostname (FQDN):

Certificate Type: Server [i](#)

Certificate validity (in days): [i](#)

Private Key Password (optional): [i](#)

Additional information

Alternative Name(s): [i](#)

Organization Name:

Organizational Unit:

Country Name: [i](#)

Locality Name:

State or Province:

Street Address:

Email Address:

Ok Cancel

In the Auto Confirm mode, you can specify the time, application and the clients IPs where auto confirms will works. On the previous screenshot, I have configured the auto confirm valid 30 minutes for every Spankey clients coming from the network 192.168.3.0/24. To enable the auto-confirm, switch the **Enable Auto Confirm** button to **Yes**. The auto confirm is now

enabled.

The SpanKey client can now be installed silently. Once the package is installed, run the following command to run the SpanKey Client setup with your parameters.

- > `192.168.3.117` is my WebADM/SpanKey server IP,
- > `my_client_id` is the client_id value configured in `/otp/spankey/conf/spankey.conf`
- > `ENABLE_SSH__DEFAULT=Y` is to enable SpanKey_client for OpenSSH (by default, this setting is set to `No` for other scenarios)

```
[root@spankey_client ~]# ENABLE_SSH__DEFAULT=Y /opt/spankey/bin/setup silent
192.168.3.117 my_client_id
Primary OpenOTP service URL is: 'https://192.168.3.117:8443/spankey/'
Enable SpanKey Client for OpenSSH server: 'YES'
Enable SpanKey Client NSS plugin: 'YES'
Register SpanKey Client logrotate script: 'YES'
SpanKey Client must be automatically started at boot: 'YES'

Applying SpanKey Client settings from default configuration files... Ok
Retrieving WebADM CA certificate from host '192.168.3.117'... Ok
The setup needs now to request a signed 'SpanKey' client certificate.
This request should show up as pending in your WebADM interface and an administrator
must accept it.
Waiting for approbation... Ok
Updating entry 'client_id' in file '/opt/spankey/conf/spankey.conf'... Ok
Updating file '/etc/nsswitch.conf'... Ok
Updating file '/etc/pam.d/password-auth'... Ok
Registering SpanKey Client service...
Registering SpanKey Client service... Ok
Adding logrotate script... Ok

SpanKey Client has successfully been setup.

IMPORTANT: Do not forget to perform the following actions before you exit this session:
- Start SpanKey (/opt/spankey/bin/spankey start)
- Restart 'sshd'
- Restart 'nscd'
```

The configuration of the SpanKey client is done, you have to restart sshd, nscd and Spankey client:

```
[root@spankey_client ~]# systemctl restart sshd
[root@spankey_client ~]# systemctl restart nscd
[root@spankey_client ~]# systemctl start spankey
```

4. Advanced Configurations

4.1 SpanKey Client

4.1.1 Files and Folders

SpanKey client is installed under `/opt/spankey/` folder.

Find below the SpanKey client software installation file structure and important files.

- › `/opt/spankey/bin/` : Location for SpanKey service binaries and startup scripts.
 - › `spankey` : SpanKey executable control script for starting and stopping the service process. To start SpanKey from the command line, issue `./spankey start`. To stop SpanKey, issue `./spankey stop`.
 - › `setup` : Initial SpanKey setup script run by the self-installer. The setup can be re-run manually at any time.
- › `/opt/spankey/doc/` : Location for spankey documentation resources.
- › `/opt/spankey/conf/` : Location for SpanKey configuration files.
 - › `spankey.conf` : Main configuration file. Defines the basic SpanKey client parameters.

```
#-#-#-#
#
#  SpanKey's main configuration file.
#
#-#-#-#
#
#  The entry below tells the daemon where the log file must be.
#  At the very early stage (when the daemon started but did not read yet this
configuration file)
#  logs are sent to the standard output. Anyway, since the launcher script use a
redirection, you won't even see them.
#
log_file          /opt/spankey/logs/spankeyd.log
#
#  When log level is set to 'Normal', all components will log both errors and warnings
only.
#  'Verbose' will make all components just log everything.
#
log_level        Normal
#
#
#-#-#-#


#-#-#-#
#
#  Where to produce the daemon's pid file.
#
#pid file        /opt/spankey/temp/spankeyd.pid
```

```
#  
#  
#---#---#  
  
#---#---#  
#  
# The daemon needs this CA file to trust SpanKey servers it will talk to.  
#  
ca_file          /opt/spankey/conf/ca.crt  
#  
#  
#---#---#  
  
#---#---#  
#  
# An optional client certificate and password spankeyd will use to communicate with  
SpanKey servers.  
#  
client_cert_file    /opt/spankey/conf/spankey.pem  
#client_cert_password PaSsWoRd  
#  
#  
#---#---#  
  
#---#---#  
#  
# The section below contains a list of backend servers the daemon should connect to.  
# It must contains one or two target OTP server.  
# Any additional server in the list will just be ignored.  
#  
server_urls {  
  url1 https://192.168.3.117:8443/spankey/  
  #url2 https://<server2>:8443/spankey/  
}  
#  
#  
#---#---#  
  
#---#---#  
#  
# How spankeyd will relay request to the WebADM backend.  
# - "balanced" means the request will be balanced between server 1 and server 2 in a  
round-robin fashion.  
# - "ordered" means server 2 is kept as a hot spare in case the primary server stops  
answering requests properly.  
#  
#server_policy      BaLaNcEd  
#
```

```

#
#-#-#-#


#-#-#-#
#
# The default domain name to pass when the requester only provided a username.
# It typically overrides the default domain in the SpanKey server configuration.
#
#default_domain_name Default
#
# To let backends know how to extract fields 'domain' and 'username' correctly from
the username string the client entered.
#
#domain_separator    \\
#
#
#-#-#-#


#-#-#-#
#
# Requested Tags (user must present all the tags).
#
#requested_tags      TAG1,TAG2
#
#
#-#-#-#


#-#-#-#
#
# User settings (better configure settings in client policies).
# Fixed list of SpanKey policy settings to be passed via the SpanKey API.
#
#user_settings      SpanKey.KeyExpire=10
#
#
#-#-#-#


#-#-#-#
#
# The client identifier to be sent to OpenOTP servers along authentication requests.
# This allows to apply per client contextual policies on the WebADM server while
running an authentication workflow.
#
client_id          my_client_id
#
#
#-#-#-#

```

```

#-#-#-#
#
# The SOAP request TCP timeout is by default 30.
# Just keep it as it unless you really understand all the possible consequences a
change could have.
#
#soap_timeout      30
#
#
#-#-#-#
#
#
#-#-#-#

```

- > `/opt/spankey/lib/` : Location for SpanKey system libraries.
- > `/opt/spankey/libexec/` : Location for SpanKey system executables.
- > `/opt/spankey/logs/` : Location for log files produced by SpanKey client.
- > `/opt/spankey/temp/` : Location for SpanKey temporary data files. Under this directory, you will find service PID files.

4.1.2 SpanKey Client and Auditd

Since Spankey client v2.1.0 and SpanKey server v2.0.4-1, you can use Auditd with SpanKey. Auditd will allow you to record executed commands, SCP actions (copy, remote execution) in WebADM record database. To enable Auditd with SpanKey client and Auditd packages must be installed and running on the target machine. By default, Auditd for SpanKey client is disabled. To enable it, after the Spankey client installation and configuration, edit the following file:

```
/etc/audisp/plugins.d/spankey.conf
```

```

# This file controls the configuration of the SpanKey Client plugin.
# It simply takes events and forwards them to the SpanKey daemon.

active = no
direction = out
path = /opt/spankey/libexec/audisp_plugin
type = always
#args =
format = string

```

Change the `active` setting from `no` to `yes`:

```
# This file controls the configuration of the SpanKey Client plugin.  
# It simply takes events and forwards them to the SpanKey daemon.  
  
active = yes  
direction = out  
path = /opt/spankey/libexec/audisp_plugin  
type = always  
#args =  
format = string
```

To changes takes effect, a restart of spankey client is required. Logs are now sent to auditd and auditd forwards logs to SpanKey client daemon. The daemon will forward logs to SpanKey server.

```
systemctl restart spankey
```

⚠ Important Note

Be aware, if you enable Auditd with SpanKey then all Auditd rules that have been set before on that machine will be erased. Therefore, if you are using your own Auditd rules for monitoring your machine then you can not use SpanKey with the **Record Audit Logs** feature.

Please refer to step [4.2.7 Audit logs and SSH Sessions recording](#) of this documentation to enable auditd logs on the SpanKey server side and to know how to consult recorded logs.

4.2 SpanKey Server

Below are described some of the most relevant SSH Public Key Server configuration options.

4.2.1 Master Group

In SpanKey you can define master groups where the members of the group are considered as super users and can use their SSH key to access any other SpanKey account. A master group can be configured in SpanKey global configuration or in a client policy. To configure a master group, go on SpanKey global configuration or client policy and configure your Master Group.

For example, my master group is `cn=master, o=Root` and the member of this group is my `cn=admin, o=Root` who has a public key enrolled on his account:

Register / Unregister SSH Public Key for `cn=admin,o=Root`

An SSH public key is already registered for user and is **VALID**.
The key does not have an expiration date and will not automatically expire!

Public Key:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIBCgKCAQEAjopaf08+UKF07rA2Ktia5
mq9LkjHPcVKx44S0p/YXZNF0Vlr+x+Xhb2SSdROG1IzN9GwqYjkuvqz49PCq
/XLfh/Q2gLePxVwIvncJ4tjqH2TR+T1E31AK6nv+8HiKzMpbfQObx9cetaGMCoW
N6vkS9N5Bq
/WoJP9uaNwuzfFR20NFKk3tUYPeSHXc2791BYTndnv6BCIjp4FXGDFT
/Wci2PMJlr3LgE+mKb5yTTm3Wb85Wdpn7JWnf0YBMAKwo3y3QTN3KVBs7bsEQ8oD9
H6mdCjVKeuNhigYKmLqyEpIg+2XI2zP2i+7cafokGfQhZtY4YBckWW
/cwF2X+xxwTDAAoAR
```

Authorized Key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQCoilp/Tz5QoXTusDYq0hrmar0uSMc9xUrHj
hLSn9hdk0XRUuv7H5eFvZJJ3pE4bUjM30zapiOS6+rpj08Kr9ct+H9DaAt4
/FXAi+dwni2COofZNH5PUTfUAqjqe
/7weKRkyIt9A5vH1x6loWxI7a3q+RL03kGr9ag8k
/25o3C7N8VHbQ0UqTe1Rg951ddzbv2UFh0d2e/oEIIM
/gVcYMVP9ZYJk8wkuvcuAT6YpvnJNObdZvzlZ2mfslad
/RaFwArCiftdRM3cnISztIwRhvaP0faZ0KNIIp642GRBaayurtSkid7zcibM
```

Key Format: RSA

Key Length: 2048 Bits

Remove **Cancel**

That means the admin's account is able to login on every account with his own private key. The public key of the admin account is added to every user account. If I call the `authorized_key` command for different users I should see the administrateur public key and the public key of the user:

```
[root@ubuntu18client-virtual-machine ~]# /opt/spankey/libexec/authorized_keys test_user
environment="ONE_TIME_AUTHENTICATION_TOKEN=C6578D1DCE1FFAA29F7C3F092957DF96",command="/opt
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQJCQuTF0MSmLUZ4iCpxBS/6D/nITkfkILuS00cTC3BR3tC2lhqjvxZXW070C
test_user@Default
environment="ONE_TIME_AUTHENTICATION_TOKEN=C6578D1DCE1FFAA29F7C3F092957DF96",command="/opt
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQCoilp/Tz5QoXTusDYq0hrmar0uSMc9xUrHjhLSn9hdk0XRUuv7H5eFvZJJ:
admin@Default
```

We can see 2 public keys for test_user account, his own public key and admin's public key.

```
[root@ubuntu18client-virtual-machine ~]# /opt/spankey/libexec/authorized_keys yoann
environment="ONE_TIME_AUTHENTICATION_TOKEN=010EF8A1110F7503DD4AC04F325E52F1",command="/opt
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQCTLE6WCDDi/gknvCpWKNXBgCZ8eZeFfYN/MJ7PBv90lWlk/puUEwC2lmWQv
yoann@Default
environment="ONE_TIME_AUTHENTICATION_TOKEN=010EF8A1110F7503DD4AC04F325E52F1",command="/opt
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQCoilp/Tz5QoXTusDYq0hrmar0uSMc9xUrHjhLSn9hdk0XRUuv7H5eFvZJJ:
admin@Default
```

It's the same for yoann's account...

Now, trying to log in with test_user and Yoann's account with the admin's private key:

```
11:56 $ ssh -i admin.pem test_user@192.168.3.178
Hello, SpanKey Tester!

Session recording is enabled.
Session lock is disabled.
Session's max duration is 30 minutes.

test_user@ubuntu18client-virtual-machine:~$ whoami
test_user
test_user@ubuntu18client-virtual-machine:~$ pwd
/home/test_user
test_user@ubuntu18client-virtual-machine:~$ exit
exit

>>> Session's duration was aprox 11 seconds <<<

Connection to 192.168.3.178 closed.
```

```
11:56 $ ssh -i admin.pem yoann@192.168.3.178
```

Hello, SpanKey Tester!

Session recording is enabled.
Session lock is disabled.
Session's max duration is 30 minutes.

```
yoann@ubuntu18client-virtual-machine:~$ whoami  
yoann  
yoann@ubuntu18client-virtual-machine:~$ pwd  
/home/yoann  
yoann@ubuntu18client-virtual-machine:~$ exit  
exit
```

>>> Session's duration was aprox 6 seconds <<<

Connection to 192.168.3.178 closed.

4.2.2 Backup/Recovery Keys

By default, the SpanKey agents will erase users' authorized_keys file at runtime to prevent users from adding rogue public keys. If recovery keys are configured, then these keys are automatically written to the user's authorized_keys file, for recovery purposes (to be used in the event where SpanKey client cannot communicate with the SpanKey server).

To configure a backup key, go on the WebADM Admin GUI, click on **Applications** tab, in **Authentication** category, you can find **SSH Public Key Server**, click on **CONFIGURE** button. You are now in SpanKey server configuration. Find the **Power Users & Recovery** section, check the box **Backup Keys** and put the public key to have an access on the target server even if SpanKey client or SpanKey server is down. Put the public key in the authorized key format here:

The screenshot shows the WebADM Freeware Edition v1.6.8-4 interface. On the left, there is a sidebar with a tree view of the LDAP structure under 'LDAP Server (OpenLDAP)'. The tree includes 'dc=WebADM', 'o=Root (5)' which contains 'cn=admin', 'cn=master', 'cn=ppolicy', 'cn=test_user', and 'cn=yoann'. Below the tree are 'Create / Search Details / Check' buttons. The main right panel has a header 'WebADM Freeware Edition v1.6.8-4' and a sub-header 'Power Users & Recovery'. It features two sections: 'Master Group' (unchecked) and 'Backup Keys' (checked). Under 'Master Group', it says 'All the members of the selected group are allowed to login with any account.' Under 'Backup Keys', there is a text input field containing the public key: `ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCOilp/Tz5QoXTusDYq0hr`. At the bottom, a note states: 'List of SSH authorized keys (one key per line in Authorized Keys format). These recovery keys are automatically written in the user's authorized_keys files.'

That means the private key associated with this public key will be able to log in on the target server even if SpanKey server or SpanKey client is down.

The public key can be found when you click on the user on the left tree, in **Application Actions** box, click on

SSH Public Key Server and Register/Register SSH Public Key.

The screenshot shows the WebADM Freeware Edition v1.6.8-4 interface. On the left, there is a sidebar with a tree view of LDAP objects under "LDAP Server (OpenLDAP)". The tree includes "dc=WebADM", "o=Root (5)" which contains "cn=admin", "cn=master", "cn=ppolicy", "cn=test_user", and "cn=yoann". Below the tree are "Create / Search" and "Details / Check" buttons. On the right, the main content area displays the "Object cn=admin,o=Root (Super Administrator)" details. It shows "Object class(es): person, webadmAccount, po...", "Account is unique: Yes (in o=root)", "WebADM settings: None [CONFIGURE]", "WebADM data: 2 data [EDIT]", "User activated: Yes Deactivate", and "Logs and inventory: WebApp, WebSrv, Inventory". Under "LDAP Actions", there are links for "Delete this object", "Copy this object", "Move this object", "Export to LDIF", "Change password", "Create certificate", "Unlock WebApp access", and "Advanced edit mode". Under "Application Actions", there are links for "Secure Password Reset (1 actions)", "User Self-Registration (1 actions)", "MFA Authentication Server (13 actions)", "SMS Hub Server (1 actions)", "SSH Public Key Server (3 actions)", and "QR Login & Signing Server (8 actions)".

The second part of the screenshot shows the "SpanKey User Actions for cn=admin,o=Root (3)". It lists three actions: "Register / Unregister SSH Public Key" (key icon), "Set or Change Key Expiration" (clock icon), and "Test Authorized Keys" (key icon). Each action has a descriptive text below it. At the bottom right is a "Cancel" button.

I can see the public key enrolled for this user in SSH key format and in authorized key format.

Now, we will do a test to see if the backup key is returned by the authorized key command for the yoann user on a SpanKey client:

```
[root@ubuntu18client-virtual-machine ~]# /opt/spankey/libexec/authorized_keys yoann
environment="ONE_TIME_AUTHENTICATION_TOKEN=CF6CC2389B99374FBB92E76D58EF891",command="/opt/spankey/bin/ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCTLE6WCDDi/gknvCpWKNXBgCZ8eZeFfYN/MJ7PBv90lWlk/puUEwC2lmWQv
yoann@Default
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCoilp/Tz5QoXTusDYq0hrmar0uSMc9xUrHjhLSn9hdk0XRUuv7H5eFvZJJ:
```

As you can see, yoann user has his own public key returned by SpanKey server and the Admin recovery key previously configured.

```
12:59 $ ssh -i admin.pem yoann@192.168.3.178
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-42-generic x86_64)
Last login: Tue Dec 18 12:57:16 2018 from 192.168.3.233

yoann@ubuntu18client-virtual-machine:~$ exit
logout
Connection to 192.168.3.178 closed.
```

Below are the logs from the SpanKey server side for the authorized key request:

```
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] New spankeyAuthorizedKeys SOAP request
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] > Username: yoann
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] > Client ID: SpanKey
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] Registered spankeyAuthorizedKeys request
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] Resolved LDAP user: cn=yoann,o=Root (cached)
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] Found user fullname: yoann
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] Found 23 user settings: EnableLogin=Yes,X11Forwarding=Yes,PortForwarding=Yes,AgentForwarding=Yes,PTYAllocation=Y[1 Items],AllowKeyFiles=No,KeyFiles=.ssh/authorized_keys,MinUID=500,MinGID=100,MailSubject=% Access Notification
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] Found 1 user data: PublicKey
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] Found 2048 bits RSA public key
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] Returning 1 authorized public key
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] Returning 1 backup public key
[2018-12-18 12:59:01] [192.168.3.178] [SpanKey:MN3K614Y] Sent success response
```

4.2.3 Shared Account/Authorized Group

Authorized Groups operate on the principle of a shared account. Shared accounts are a common practice in Enterprise use of SSH. A shared account (i.e. ‘webmaster’ user) is a system account which is used concurrently by several administrators. In SpanKey you can transform any generic LDAP user into a shared SSH account simply by linking this account to a ‘shared access LDAP group’. Then all the members of that group can gain access to the shared account with their own SSH key. For example, my shared account is `webmaster` and I want to allow access to `webmaster` account by `IT` group members.

Member of this group are `test_user` and `yoann` accounts:

The screenshot shows the WebADM Freeware Edition v1.6.8-4 interface. On the left, a sidebar titled "LDAP Server (OpenLDAP)" displays a tree structure of LDAP objects. The "o=Root" node has six children: "cn=IT", "cn=admin", "cn=master", "cn=ppolicy", "cn=test_user", and "cn=yoann". Below the tree are two buttons: "Create / Search" and "Details / Check". On the right, the main panel is titled "Object cn=IT,o=Root". It contains an "LDAP Actions" section with options like "Delete this object", "Copy this object", "Move this object", "Export to LDIF", "Add members", and "Advanced edit mode". The "Object Details" section shows "Object class(es): groupOfNames" and "Group activated: No Activate Now!". Below these are input fields for "Object Name" (set to "IT") and "Description / Note". There are also sections for "Add Attribute" (with a dropdown menu) and "Add Extension" (set to "UNIX Group"). Under "Group Member", there are two entries: "cn=test_user,o=Root" and "cn=yoann,o=Root", each with a "Goto" button and a checkbox. At the bottom is a large blue "Apply Changes / Delete Selected" button.

After that, I click on my `webmaster` account on the left tree. In `Object Details` box, I click on `CONFIGURE` button.

Choose SpanKey application and in **Shared Account** section, I configure my **IT** group like below:

Now, I'm able to log into my SpanKey_client with Yoann private key on the shared account **webmaster**:

```
16:43 $ ssh -i yoann.pem webmaster@192.168.3.178
```

```
Hello, SpanKey Tester!
```

```
Session recording is enabled.  
Session lock is disabled.  
Session's max duration is 30 minutes.
```

```
webmaster@ubuntu18client-virtual-machine:~$ whoami  
webmaster  
webmaster@ubuntu18client-virtual-machine:~$ pwd  
/home/webmaster  
webmaster@ubuntu18client-virtual-machine:~$ exit  
exit
```

```
>>> Session's duration was aprox 8 seconds <<<
```

```
Connection to 192.168.3.178 closed.
```

Logs on the SpanKey server side:

```
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:K6I3YWBV] New spankeyAuthorizedKeys SOAP request
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:K6I3YWBV] > Username: webmaster
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:K6I3YWBV] > Client ID: SpanKey
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:K6I3YWBV] Registered spankeyAuthorizedKeys request
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:K6I3YWBV] Resolved LDAP user: cn=webmaster,o=Root (cached)
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:K6I3YWBV] Found user fullname: webmaster
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:K6I3YWBV] Found 23 user settings: EnableLogin=Yes,X11Forwarding=Yes,PortForwarding=Yes,AgentForwarding=Yes,PTYAllocation=Y[1
Items],AllowKeyFiles=No,KeyFiles=.ssh/authorized_keys,MinUID=500,MinGID=100,MailSubject=% Access Notification
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:K6I3YWBV] Allowed group 'IT' with 2 member public keys
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:K6I3YWBV] Returning 2 authorized public keys
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:K6I3YWBV] Sent success response
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] New spankeySessionStart SOAP request
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] > Username: webmaster
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] > Identity: yoann
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] > Command: /bin/bash
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] > Terminal: Yes
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] > Client ID: SpanKey
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] > Source IP: 192.168.3.233
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] Registered spankeySessionStart request
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] Resolved LDAP user: cn=yoann,o=Root (cached)
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] Resolved LDAP groups: it
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] Found user fullname: yoann
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] Found 16 user settings: WelcomeText=Hello, SpanKey Tester!,MaxSessionTime=30,LockSessionTime=0,RecordSessions=Yes,CreateHomedir=Yes,MailSubject=% Access Notification,OfflineMode=Yes,EnableLogin=Yes
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] Started interactive terminal session of ID cmIRB5Es0dfsx4rC valid for 600 seconds
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] Sent success response
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:G2MDIYQF] New spankeySessionUpdate SOAP request
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:G2MDIYQF] > Session: cmIRB5Es0dfsx4rC
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] Found terminal session started 2018-12-18 14:41:04
[2018-12-18 14:41:04] [192.168.3.178] [SpanKey:HLTYITW4] Sent success response
```

4.2.4 TAGs

All hosts managed by SpanKey Server can be tagged in the SpanKey client configuration. For example, all web servers could be tagged with the acronym «WEB» in the configuration file of SpanKey client. Then you can add this Tag for all Webmaster accounts to ensure SSH access to every web server. To configure a Tag, click on a user account and in the section **Object Details** there is WebADM Settings. Click on the **CONFIGURE** button. Go on the SpanKey application and there are the options Allowed Server Tags.

TAGs can be configured on an LDAP account or an LDAP group. To set a tag on an account or a group, go on the WebADM Admin GUI, click on your account/group, in the **Object Details** box, you can find WebADM settings, click on **CONFIGURE**. In applications box on the left, select SpanKey. You are now in SpanKey configuration for your user or your group. In **Access Restriction** category, check the box **Allowed Server Tags** and configure your TAGs. On my side, I configured **web** TAG for my test_user.

The screenshot shows the WebADM Freeware Edition v1.6.8-4 interface. On the left, a sidebar displays the LDAP structure under 'LDAP Server (OpenLDAP)'. Under 'o=Root (5)', it lists 'cn=admin', 'cn=master', 'cn=ppolicy', 'cn=test_user', and 'cn=yoann'. Below this, there are 'Create / Search' and 'Details / Check' buttons. The main panel shows the 'Object cn=test_user,o=Root' details. The 'Object Details' section includes fields for Object class(es) (person, webadmAccount, po...), Account is unique (Yes), WebADM settings (2 settings [CONFIGURE]), WebADM data (2 data [EDIT]), User activated (Yes), and Logs and inventory (WebApp, WebSrv, Inventory). The 'LDAP Actions' section contains links for Delete this object, Copy this object, Move this object, Export to LDIF, Change password, Create certificate, Unlock WebApp access, and Advanced edit mode. The 'Application Actions' section lists various services like Secure Password Reset, User Self-Registration, MFA Authentication Server, SMS Hub Server, SSH Public Key Server, and QR Login & Signing Server. At the bottom of the main panel, the 'Access Restrictions' section shows a checked checkbox for 'Allowed Server Tags' with the value 'web' entered. A note below says 'Comma-separated list of allowed server tags.' The 'Session Options' section is also visible at the bottom.

Now, I just have to TAG my servers where SpanKey client is configured. TAG should be configured in

```
/opt/spankey/conf/spankeyd.conf .
```

```
[root@ubuntu18client-virtual-machine ~]# vi /opt/spankey/conf/spankeyd.conf
#-#-#-#
#
# spankeyd's main configuration file.
#
...
#
#-#-#-#
#
# Requested Tags (user must present all the tags).
#
#           requested_tags      web
#
#
#-#-#-#
...
#
#
#-#-#-#
```

Please, restart SpanKey Client after editing the configuration file.

```
[root@ubuntu18client-virtual-machine ~]# /opt/spankey/bin/spankey restart
```

After tagging my server, I perform a login with an account which has the same TAG configured.

```
15:39 $ ssh -i test_user.pem test_user@192.168.3.178
```

```
Hello, SpanKey Tester!
```

```
Session recording is enabled.  
Session lock is disabled.  
Session's max duration is 30 minutes.
```

```
test_user@ubuntu18client-virtual-machine:~$ whoami  
test_user  
test_user@ubuntu18client-virtual-machine:~$ pwd  
/home/test_user  
test_user@ubuntu18client-virtual-machine:~$ exit  
exit
```

```
>>> Session's duration was aprox 7 seconds <<<
```

```
Connection to 192.168.3.178 closed.
```

See below the result of the authentication:

```
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] New spankeyAuthorizedKeys SOAP request
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] > Username: test_user
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] > Tags: web
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] > Client ID: SpanKey
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Registered spankeyAuthorizedKeys request
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Resolved LDAP user: cn=test_user,o=Root
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Found user fullname: test_user
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Found 23 user settings: EnableLogin=Yes,X11Forwarding=Yes,PortForwarding=Yes,AgentForwarding=Yes,PTYAllocation=Y[1 Items],BackupKeys=[1 Items],AllowKeyFiles=No,KeyFiles=.ssh/authorized_keys,MinUID=500,MinGID=100,MailSubject=Access Notification
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Found 2 user tags: WEB,SQL
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Found 1 user data: PublicKey
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Found 2048 bits RSA public key
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Validated authorization for server tag 'WEB'
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Returning 1 authorized public key
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Returning 1 backup public key
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:CC7ZTR8Q] Sent success response
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] New spankeySessionStart SOAP request
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] > Username: test_user
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] > Identity: test_user
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] > Command: /bin/bash
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] > Terminal: Yes
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] > Client ID: SpanKey
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] > Source IP: 192.168.3.233
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] Registered spankeySessionStart request
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] Resolved LDAP user: cn=test_user,o=Root (cached)
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] Found user fullname: test_user
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] Found 13 user settings: WelcomeText=Hello, SpanKey Tester!,MaxSessionTime=30,LockSessionTime=0,RecordSessions=Yes,CreateHomedir=Yes,MailSubject=Access Notification,OfflineMode=Yes
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] Started interactive terminal session of ID Md618XfBrP1Mnkmq valid for 600 seconds
[2018-12-18 15:39:36] [192.168.3.178] [SpanKey:TM789F0W] Sent success response
```

It works well for the test_user, I will try now an authentication with the account Yoann which doesn't have the **web** TAG.

```
15:40 $ ssh -i yoann.pem yoann@192.168.3.178
```

See below the result of the authentication:

```
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] New spankeyAuthorizedKeys SOAP request
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] > Username: yoann
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] > Tags: web
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] > Client ID: SpanKey
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] Registered spankeyAuthorizedKeys request
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] Resolved LDAP user: cn=yoann,o=Root
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] Found user fullname: yoann
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] Found 23 user settings: EnableLogin=Yes,X11Forwarding=Yes,PortForwarding=Yes,AgentForwarding=Yes,PTYAllocation=Y[1
Items],AllowKeyFiles=No,KeyFiles=.ssh/authorized_keys,MinUID=500,MinGID=100,MailSubject=Access Notification
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] Found 1 user data: PublicKey
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] Found 2048 bits RSA public key
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] Account is missing authorization for server tag 'WEB'
[2018-12-18 15:40:18] [192.168.3.178] [SpanKey:8JSB1WK0] No authorized public key found
[2018-12-18 15:40:19] [192.168.3.178] [SpanKey:8JSB1WK0] Sent failure response
```

As you can see, the authentication failed because the account is missing an authorization for server TAG **web**.

4.2.5 Guest Account

Another feature of SpanKey is the Guest Account. A Guest account can be used by a consultant for example. If enabled, the user's home directory will automatically be created and deleted after logout. The account is deleted after the last opened session is closed. In my example, I will configure an account named **Oracle_Guest**. To configure this account as a Guest Account, click on your user on the left tree, in **Object Details** box, you can find **WebADM Settings**, click on **CONFIGURE**. In applications box on the left, select **SpanKey**. You are now in SpanKey configuration for your users. In **UNIX Account Options** category, check the box **Guest Account Mode** and set this feature to **Yes**.



In that scenario, I can also configure a TAG for this Guest User, **SQL** TAG, for example, to allow the access to every **SQL** tagged servers by my Oracle consultant through the Guest account.

4.2.6 Allow local users and local Authorized Keys File(s) usage

The SpanKey server allows you to configure local users who will be able to use the local authorized keys file(s) configured. In the SpanKey server configuration, you will find the following setting under Server Policy:

The screenshot shows the WebADM Freeware Edition v1.6.8-4 interface. On the left, there is a sidebar titled "LDAP Server (OpenLDAP)" with a refresh icon. It lists "OpenLDAP (2)" and "dc=WebADM". Under "dc=WebADM", there is a section titled "o=Root (6)" containing user entries: "cn=admin", "cn=master", "cn=oracle_guest", "cn=ppolicy", "cn=test_user", and "cn=yoann". Below this are "Create / Search Details / Check" links. On the right, the main panel has a header "WebADM Freeware Edition v1.6.8-4" with a copyright notice. It includes navigation links: Home, Admin, Create, Search, Import, Databases, Statistics, Applications, About, and Logout. There are also icons for API, desktop, mobile, globe, and help. The main content area contains three configuration sections: 1) "Require Extra Login Factors" with a dropdown set to "LDAP" and a note about OpenOTP support. 2) "Allowed Local Users" with a list box containing "root, admin, webmaster". 3) "Authorized Key File(s)" with a list box containing ".ssh/authorized_keys, .ssh/authorized_keys2". A note below specifies comma-separated lists for both users and files.

Configure your users who are able to use the local authorized keys file(s) first and after that, configure the authorized keys file(s) that your users will be able to use for local login.

4.2.7 Audit logs and SSH Sessions recording

For security audit, Spankey provide 2 kinds of audit logs.

The first one is the graphical session recording. All SSH sessions can be recorded and that allow you to replay every SSH sessions at any moment through the WebADM Admin interface. The **Record Session Data** setting must be enabled for session recording.

Another kind of audit is the **Record Audit Logs**. The setting will allow you to store audit event (commands and file events) in the WebADM Record databases.

These 2 settings can be enabled under SpanKey Server configuration:

The screenshot shows the "Session Options" configuration screen. It contains two settings: 1) "Record Session Data" with a checked checkbox and a radio button set to "Yes" (selected). A note below states: "Stores the graphical terminal sessions in WebADM Record database. SCP and SFTP sessions cannot be recorded." 2) "Record Audit Logs" with a checked checkbox and a radio button set to "Yes" (selected). A note below states: "Stores Auditd events in WebADM Record database (commands and file events)."

Recorded sessions and audit logs can be replayed under

WebADM Admin GUI > Databases > Recorded Sessions

Database Viewer for Recorded Sessions (1000 results out of 2272 log items)														
Filters (1)														
Client		Equals		spankey_shell		Remove								
Application		Equals				Add Filter								
This Minute		This Hour		Today		This Week		This Month						
Display Options		Log Actions		Statistic Options		Database Pruning								
Retrieve max		1000				Delete selected items		Show first		ALL				
Page results		35				Re-encrypt all records								
						Statistics as CSV / XML		Group by		None				
												Draw source map		
												Clean		
<input type="checkbox"/>	<input type="radio"/> Application	<input type="radio"/> Client	Start Time		Stop Time		<input type="radio"/> User DN	<input type="radio"/> User IP	<input type="radio"/> Host IP	Session ID		Type	Size	Action
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:39		2019-04-03 18:15:39		cn=spankey_ubuntu19.ou=Loic.o...	192.168.3.233	78.141.172.206	RKGRL567E	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:39		2019-04-03 18:15:39		cn=spankey_ubuntu19.ou=Loic.o...	192.168.3.233	78.141.172.206	RKGRL567E	TERM	92 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:38		2019-04-03 18:15:38		cn=spankey_fedora29.ou=Loic.o...	192.168.3.233	78.141.172.206	TMHXT05W	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:38		2019-04-03 18:15:38		cn=spankey_fedora29.ou=Loic.o...	192.168.3.233	78.141.172.206	TMHXT05W	TERM	89 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:36		2019-04-03 18:15:36		cn=spankey_scientific.ou=Loic.o...	192.168.3.233	78.141.172.206	KYACYRGH	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:35		2019-04-03 18:15:35		cn=spankey_centos7.ou=Loic.o...	192.168.3.233	78.141.172.206	57PYPUJDN	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:35		2019-04-03 18:15:35		cn=spankey_centos7.ou=Loic.o...	192.168.3.233	78.141.172.206	57PYPUJDN	TERM	87 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:33		2019-04-03 18:15:33		cn=spankey_centos6.ou=Loic.o...	192.168.3.233	78.141.172.206	58ROU550	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:33		2019-04-03 18:15:33		cn=spankey_centos6.ou=Loic.o...	192.168.3.233	78.141.172.206	58ROU550	TERM	87 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:32		2019-04-03 18:15:32		cn=spankey_debian9.ou=Loic.o...	192.168.3.233	78.141.172.206	U0ZBRB05	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:32		2019-04-03 18:15:32		cn=spankey_debian9.ou=Loic.o...	192.168.3.233	78.141.172.206	U0ZBRB05	TERM	92 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:32		2019-04-03 18:15:32		cn=spankey_ubuntu18.ou=Loic.o...	192.168.3.233	78.141.172.206	K5J0S4AO	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:32		2019-04-03 18:15:32		cn=spankey_ubuntu18.ou=Loic.o...	192.168.3.233	78.141.172.206	K5J0S4AO	TERM	94 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:31		2019-04-03 18:15:32		cn=spankey_ubuntu16.ou=Loic.o...	192.168.3.233	78.141.172.206	T3Y7QZLG	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 18:15:31		2019-04-03 18:15:32		cn=spankey_ubuntu16.ou=Loic.o...	192.168.3.233	78.141.172.206	T3Y7QZLG	TERM	94 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:10		2019-04-03 17:44:10		cn=spankey_ubuntu19.ou=Loic.o...	192.168.3.233	78.141.172.206	H4909RVJ	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:09		2019-04-03 17:44:10		cn=spankey_fedora29.ou=Loic.o...	192.168.3.233	78.141.172.206	TIICE2X0	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:07		2019-04-03 17:44:07		cn=spankey_scientific.ou=Loic.o...	192.168.3.233	78.141.172.206	YSKDPJ76	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:07		2019-04-03 17:44:07		cn=spankey_scientific.ou=Loic.o...	192.168.3.233	78.141.172.206	YSKDPJ76	TERM	99 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:06		2019-04-03 17:44:06		cn=spankey_centos7.ou=Loic.o...	192.168.3.233	78.141.172.206	XVE8PQ9A	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:06		2019-04-03 17:44:06		cn=spankey_centos7.ou=Loic.o...	192.168.3.233	78.141.172.206	XVE8PQ9A	TERM	87 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:04		2019-04-03 17:44:04		cn=spankey_centos6.ou=Loic.o...	192.168.3.233	78.141.172.206	V9RT5KOV	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:04		2019-04-03 17:44:04		cn=spankey_centos6.ou=Loic.o...	192.168.3.233	78.141.172.206	V9RT5KOV	TERM	87 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:04		2019-04-03 17:44:04		cn=spankey_debian9.ou=Loic.o...	192.168.3.233	78.141.172.206	I2WSYJL0	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:04		2019-04-03 17:44:04		cn=spankey_debian9.ou=Loic.o...	192.168.3.233	78.141.172.206	I2WSYJL0	TERM	92 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:03		2019-04-03 17:44:03		cn=spankey_ubuntu18.ou=Loic.o...	192.168.3.233	78.141.172.206	Y5DDGLZ	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:03		2019-04-03 17:44:03		cn=spankey_ubuntu18.ou=Loic.o...	192.168.3.233	78.141.172.206	Y5DDGLZ	TERM	88 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:03		2019-04-03 17:44:03		cn=spankey_ubuntu16.ou=Loic.o...	192.168.3.233	78.141.172.206	JW60GXK9	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:44:03		2019-04-03 17:44:03		cn=spankey_ubuntu16.ou=Loic.o...	192.168.3.233	78.141.172.206	JW60GXK9	TERM	88 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:17:37		2019-04-03 17:17:41		cn=spankey_ubuntu16.ou=Loic.o...	192.168.3.233	78.141.172.206	KK0UGCYK	AUDIT	11 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell	2019-04-03 17:16:52		2019-04-03 17:16:52		cn=spankey_ubuntu19.ou=Loic.o...	192.168.3.233	78.141.172.206	GEH3MGR	AUDIT	163 Bytes	View	
<input type="checkbox"/>	SpanKey	<input checked="" type="checkbox"/> spankey_shell												

Under the Recorded Sessions databases, 2 types of record are available:

- > **TERM** : This is a graphical session record
- > **AUDIT** : This is the command and file events record

Click on view button to see the recorded sessions/logs

Other informations like client, Session duration, User DN, User IP, Host IP and Session ID are also useful here.

This is an example of auditd logs available through WebADM Admin GUI under databases > Recorded Sessions. Click on **View** button on an **AUDIT** log type to consult auditd logs:

[2019-04-15 14:49:34] [1234] Executed command '/bin/bash' (pid 25851) in '/home/yoann' as 501:100
[2019-04-15 14:49:34] [1234] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1235] Executed command '/usr/bin/id -gn' (pid 25859) in

```
'/home/yoann' as 501:100
[2019-04-15 14:49:34] [1235] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1236] Executed command '/usr/bin/id -un' (pid 25861) in
'/home/yoann' as 501:100
[2019-04-15 14:49:34] [1236] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1238] Executed command 'ls /etc/bash_completion.d' (pid 25865) in
'/home/yoann' as 501:100
[2019-04-15 14:49:34] [1238] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1239] Executed command 'uname -o' (pid 25867) in '/home/yoann' as
501:100
[2019-04-15 14:49:34] [1239] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1240] Executed command 'pkg-config --variable=completionsdir bash-
completion' (pid 25869) in '/home/yoann' as 501:100
[2019-04-15 14:49:34] [1240] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1241] Executed command '/bin/sh /usr/libexec/grepconf.sh -c' (pid
25870) in '/home/yoann' as 501:100
[2019-04-15 14:49:34] [1241] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1242] Executed command 'grep -qsi ^COLOR.*none /etc/GREP_COLORS'
(pid 25871) in '/home/yoann' as 501:100
[2019-04-15 14:49:34] [1242] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1243] Executed command '/usr/bin/tty -s' (pid 25873) in
'/home/yoann' as 501:100
[2019-04-15 14:49:34] [1243] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1244] Executed command '/usr/bin/tput colors' (pid 25874) in
'/home/yoann' as 501:100
[2019-04-15 14:49:34] [1244] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1245] Executed command '/usr/bin/dircolors --sh
/etc/DIR_COLORS.256color' (pid 25876) in '/home/yoann' as 501:100
[2019-04-15 14:49:34] [1245] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1246] Executed command '/usr/bin/grep -qi ^COLOR.*none
/etc/DIR_COLORS.256color' (pid 25877) in '/home/yoann' as 501:100
[2019-04-15 14:49:34] [1246] > Event 'execve' returned success with code 0
[2019-04-15 14:49:34] [1247] Executed command '/usr/bin/id -u' (pid 25879) in
'/home/yoann' as 501:100
[2019-04-15 14:49:34] [1247] > Event 'execve' returned success with code 0
[2019-04-15 14:49:39] [1248] Executed command 'ps faux' (pid 25880) in '/home/yoann' as
501:100
[2019-04-15 14:49:39] [1248] > Event 'execve' returned success with code 0
[2019-04-15 14:49:41] [1249] Executed command 'sh /tmp/test.sh' (pid 25886) in
'/home/yoann' as 501:100
[2019-04-15 14:49:41] [1249] > Event 'execve' returned success with code 0
[2019-04-15 14:50:05] [1250] Executed command 'scp /tmp/test.sh
yoann@192.168.3.181:/Users/yoann/Desktop/' (pid 25907) in '/home/yoann' as 501:100
[2019-04-15 14:50:05] [1250] > Event 'execve' returned success with code 0
[2019-04-15 14:50:05] [1251] Executed command '/usr/bin/ssh -x -oForwardAgent=no -
oPermitLocalCommand=no -oClearAllForwardings=yes -l yoann -- 192.168.3.181 scp -t
/Users/yoann/Desktop/' (pid 25908) in '/home/yoann' as 501:100
[2019-04-15 14:50:05] [1251] > Event 'execve' returned success with code 0
```

4.2.8 Sudoers Policy Plugin

Since SpanKey Client for Linux v2.2.0 and SpanKey Server v2.0.5-1, you can use Sudo Commands with SpanKey. There is an advanced section that you may use in WebADM to apply the full syntax of the sudoers file (global options, global aliases and rules). Then, the rules coming from Spankey policies (global, user, and client policy) will be appended. So the priority order of the rules are:

1. Client policy
2. User policy
3. Global policy
4. Rules from the advanced section

Run the following command `sudo -V` to check if SpanKey sudoers policy plugin has been successfully loaded:

```
$ ssh -i centos7 centos7@192.168.3.120

Welcome to SpanKey SSH Server.
This is a demonstration by RCDEVS SA.

Session recording is enabled.
Audit logs recording is enabled.
Session lock idle time is 10 minutes.
Session's max duration is 30 minutes.

[centos7@centos7-client ~]$ sudo -V
Sudo version 1.8.23

SpanKey sudoers policy plugin version 2.3.0
Copyright 2010-2019 RCDevs SA, All rights reserved.

Sudoers file grammar version 46
Sudoers I/O plugin version 2.3.0
[centos7@centos7-client ~]$ exit
exit

>>> Session's duration was aprox 6 seconds <<<

Connection to 192.168.3.120 closed.
$
```

Authorized sudo commands can be set in `WebADM GUI > Applications >`

`SSH Public Key Server (SpanKey) v2.0.5-1 > Configure > Privilege Elevation:`

Privilege Elevation

Sudo Commands

```
(ALL) /bin/ls
(ALL) /usr/bin/yum
(ALL) /usr/bin/apt-get
(ALL) /usr/bin/package-cleanup
(ALL) /sbin/reboot
(ALL) /usr/sbin/reboot
(ALL) /sbin/shutdown
```

Sudo Advanced

Simple sudo authorized commands in the 'sudoers' format.
A command should be configured without the left part (before '=').

SSH Options

Source Address Filter

Run the following command `sudo -l` to check the rights and the set of rules:

```
$ ssh -i centos7 centos7@192.168.3.120

Welcome to SpanKey SSH Server.
This is a demonstration by RCDEVS SA.

Session recording is enabled.
Audit logs recording is enabled.
Session lock idle time is 10 minutes.
Session's max duration is 30 minutes.

[centos7@centos7-client ~]$ sudo -l
User centos7 may run the following commands on centos7-client:
(ALL) /bin/ls
(ALL) /usr/bin/yum
(ALL) /usr/bin/apt-get
(ALL) /usr/bin/package-cleanup
(ALL) /sbin/reboot
(ALL) /usr/sbin/reboot
(ALL) /sbin/shutdown
(ALL) /usr/sbin/shutdown
[centos7@centos7-client ~]$ exit
exit

>>> Session's duration was aprox 4 seconds <<<
```

```
Connection to 192.168.3.120 closed.
$
```

4.3 OpenSSH

The SpanKey client setup script asks us during the setup if we want to enable SpanKey for OpenSSH and we reply **Yes** to this question.

This action involves changing `/etc/ssh/sshd_config` configuration file. The script edit the following parameters:

```
AuthorizedKeysCommand /opt/spankey/libexec/authorized_keys
AuthorizedKeysCommandUser root
PermitUserEnvironment yes
UsePAM yes
```

Depending on the SSHd version, you might need to use `AuthorizedKeysCommandRunAs` instead of `AuthorizedKeysCommandUser`. Restart SSHd if you change the configuration.

```
service sshd restart
```

4.4 NSS Provider

4.4.1 RHEL & CentOS

The SpanKey client setup script asks us during the setup if we want to enable SpanKey for NSCD and we reply **Yes** to this question.

This action involves changing `/etc/nsswitch.conf` configuration file.

The script edit the following parameters:

```
passwd: files spankey sss
shadow: file sss
group: files spankey sss
```

Restart NSCD to apply the configuration:

```
service nscd restart
```

4.4.2 Debian & Ubuntu

The SpanKey client setup script asks us during the setup if we want to enable SpanKey for NSCD and we reply **Yes** to this question.

This action involves changing `/etc/nsswitch.conf` configuration file.

The script edits the following parameters:

```
passwd: compat spankey
shadow: compat
group:  compat spankey
```

4.4.3 getent passwd/group tests

To check if your LDAP users are well returned on your spankey_client, you can use the following command:

```
getent passwd
```

This command should return all LDAP accounts allowed for this host. An LDAP account can be returned only if the account is extended to UNIX. Please refer to step [5 . 0 Users / Groups Management](#) to know how to activate/extend an LDAP account for SpanKey usage).

```
[root@webadm temp]# getent passwd  
#### The following accounts are local accounts  
  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin  
games:x:12:100:games:/usr/games:/sbin/nologin  
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin  
nobody:x:99:99:Nobody:/sbin/nologin  
dbus:x:81:81:System message bus:/sbin/nologin  
polkitd:x:999:998:User for polkitd:/sbin/nologin  
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin  
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin  
postfix:x:89:89::/var/spool/postfix:/sbin/nologin  
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin  
admin:x:1000:1000:admin:/home/admin:/bin/bash  
nscd:x:28:28:NSCD Daemon:/sbin/nologin  
systemd-bus-proxy:x:998:996:systemd Bus Proxy:/sbin/nologin  
systemd-network:x:192:192:systemd Network Management:/sbin/nologin  
tss:x:59:59:Account used by the trousers package to sandbox the tcscd  
daemon:/dev/null:/sbin/nologin  
webadm:x:997:995::/opt/webadm:/bin/bash  
mysql:x:27:27:MariaDB Server:/var/lib/mysql:/sbin/nologin  
ntp:x:38:38::/etc/ntp:/sbin/nologin  
tcpdump:x:72:72::/sbin/nologin  
radiusd:x:95:95:radiusd user:/var/lib/radiusd:/sbin/nologin  
spankey:x:996:1001:SpanKey Client System User:/opt/spankey:/sbin/nologin
```

The following accounts are LDAP accounts

```
Administrateur:x:1111:111::/home/administrateur:/bin/bash  
quick:x:500:100::/home/quick:/bin/bash  
yoann:x:1010:100::/home/yoann:/bin/bash  
test_user:x:500:100::/home/test_user:/bin/bash
```

Note

« getent passwd » command may take few seconds to yield results.

After the getent passwd command, you should have the following result in `/opt/webadm/logs/webadm.log` (server

side) if the command has worked successfully:

```
[2018-05-22 17:11:25] [192.168.3.178] [SpanKey:AFA5ES1I] New spankeyNSSList SOAP request
[2018-05-22 17:11:25] [192.168.3.178] [SpanKey:AFA5ES1I] > Database: user
[2018-05-22 17:11:25] [192.168.3.178] [SpanKey:AFA5ES1I] > Client ID: my_client_id
[2018-05-22 17:11:25] [192.168.3.178] [SpanKey:AFA5ES1I] Registered spankeyNSSList request
[2018-05-22 17:11:25] [192.168.3.178] [SpanKey:AFA5ES1I] Found 4 posix users
[2018-05-22 17:11:25] [192.168.3.178] [SpanKey:AFA5ES1I] Sent success response
```

To check if your LDAP groups are well returned on your spankey client machine, you can use the following command:

```
getent group
```

Note that only activated LDAP groups will be returned with this command. Please refer to step

[5 . 0 Users/Groups Management](#) to know how to activate/extend an LDAP group for SpanKey usage).

```
[root@we2yo tmp]# getent group

##### The following groups are local groups

root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mem:x:8:
kmem:x:9:
wheel:x:10:
cdrom:x:11:
mail:x:12:postfix
man:x:15:
dialout:x:18:webadm
floppy:x:19:
games:x:20:
tape:x:30:
video:x:39:
ftp:x:50:
lock:x:54:
audio:x:63:
nobody:x:99:
users:x:100:
avahi-autoipd:x:170:
```

```
avahi-autodiscover:x:77:  
utmp:x:22:  
utempter:x:35:  
ssh_keys:x:999:  
input:x:998:  
systemd-journal:x:190:  
systemd-bus-proxy:x:997:  
systemd-network:x:996:  
dbus:x:81:  
polkitd:x:995:  
dip:x:40:  
tss:x:59:  
postdrop:x:90:  
postfix:x:89:  
chrony:x:994:  
sshd:x:74:  
mysql:x:993:  
webadm:x:1000:  
ldap:x:55:  
slocate:x:21:  
nscd:x:28:  
tcpdump:x:72:  
cgred:x:992:  
docker:x:991:  
radiusd:x:990:  
toto:x:1003:  
apache:x:48:  
stapusr:x:156:  
stapsys:x:157:  
stapdev:x:158:
```

The following groups are LDAP groups

```
Administrateurs de l'entreprise:x:100:Administrateur  
Admins du domaine:x:101:Administrateur,yoann,vagrant  
ITWeb:x:103:vagrant  
Invités du domaine:x:110:  
testgroup:x:100:testadfs,vagrant  
webadm admins:x:102:yoann  
yotesting:x:10000:
```

After the getent group command, you should have the following result in /opt/webadm/logs/webadm.log (server side) if the command has worked successfully:

```
[2019-04-15 14:49:33] [192.168.3.178] [SpanKey:GMX0P188] New spankeyNSSList SOAP request
[2019-04-15 14:49:33] [192.168.3.178] [SpanKey:GMX0P188] > Database: group
[2019-04-15 14:49:33] [192.168.3.178] [SpanKey:GMX0P188] > Client ID: my_client_id
[2019-04-15 14:49:33] [192.168.3.178] [SpanKey:GMX0P188] Registered spankeyNSSList request
[2019-04-15 14:49:33] [192.168.3.178] [SpanKey:GMX0P188] Found 7 NSS groups
[2019-04-15 14:49:33] [192.168.3.178] [SpanKey:GMX0P188] Sent success response
```

5. Users/Groups Management

5.1 Users Management (Activation)

To enable your LDAP users to be propagated as Linux accounts, and to work with the SpanKey, they must be extended with “Unix Account” object class. This is done in the WebADM graphical interface (can be done as a batch jobs as well) as follows:

1. Choose LDAP account that you like to extend.
2. Make sure the account is a WebADM account. If not, you must first extend the account with WebADM object class.
3. Choose WebADM Account in Add Selector. Click **Add**.
4. Choose UNIX Account in the Add Extension selector. Click **Add**.

LDAP Server (OpenLDAP)

- OpenLDAP (2)
 - dc=WebADM
 - o=Root (3)
 - cn=admin
 - cn=ppolicy
 - cn=test_user
- Create / Search Details / Check

WebADM Freeware Edition v1.6.8-4
Copyright © 2010-2018 RCDevs SA, All Rights Reserved

Home | Admin | Create | Search | Import | Databases | Statistics | Applications | About | Logout

Object **cn=test_user,o=Root**

LDAP Actions

- Delete this object
- Copy this object
- Move this object
- Export to LDIF
- Change password
- Create certificate
- Unlock WebApp access
- Advanced edit mode

Object Details

Object class(es): person, webadmAccount
 Account is unique: Yes (in o=root)
 WebADM settings: 1 settings [CONFIGURE]
 WebADM data: None [EDIT]
 User activated: Yes Deactivate *i*
 Logs and inventory: WebApp, WebSrv, Inventory

Application Actions

- Secure Password Reset (1 actions)
- User Self-Registration (1 actions)
- MFA Authentication Server (13 actions)
- SMS Hub Server (1 actions)
- SSH Public Key Server (3 actions)
- QR Login & Signing Server (8 actions)

Object Name test_user Rename

Add Attribute (10) Description / Note Add

Add Extension (1) UNIX Account Add

Last Name test_user
[\[add values\]](#)

Login Name test_user
[\[add values\]](#)

WebADM Settings Edit Application Settings
[\[delete attribute\]](#)
 OpenOTP.Login Mode: LDAPOTP

Apply Changes / Delete Selected

1. Enter the following information and click **Proceed**. Click on **Extend Object**.

LDAP Server (OpenLDAP)

- OpenLDAP (2)
 - dc=WebADM
 - o=Root (3)
 - cn=admin
 - cn=ppolicy
 - cn=test_user
- Create / Search Details / Check

WebADM Freeware Edition v1.6.8-4
Copyright © 2010-2018 RCDevs SA, All Rights Reserved

Home | Admin | Create | Search | Import | Databases | Statistics | Applications | About | Logout

Add Extension **UNIX Account** to **cn=test_user,o=Root**

In order to add the objectclass **UNIX Account**
you must specify at least 3 new mandatory attribute(s).

Mandatory attributes

UID Number: 500

GID Number: 100

Home Directory: /home/test_user

Optional attributes

Login Shell: /bin/bash

General Information:

Description / Note:

Proceed **Cancel**

Now, the LDAP Account is extended for UNIX Authentication.

5.2 Groups Management (Activation)

To enable your LDAP groups to be propagated as Linux groups, and to work with the SpanKey, it must be extended with “Unix Group” object class. This is done in the WebADM graphical interface (can be done as a batch jobs as well) as follows:

1. Choose LDAP group that you like to extend.
2. Choose UNIX Group in the Add Extension selector. Click **Add**.
3. Enter the required information and click **Proceed**. Click on **Extend Object**.

Now, the LDAP group is extended for UNIX usage.

5.3 Auto increment UIDnumber and GIDnumber during user/group activation

In order to auto increment UID and GID numbers during user/group activation, you have to create an **LDAP Option Sets** object. Login on the **WebADM Admin GUI > Admin tab > LDAP Option Sets > Add OptionSet**. On the next screen, name your OptionSet :

Click **Proceed** button and on the next page click on **Create Object** :

Create Configuration Object of Type OptionSet

Confirm object creation for cn=UID_GUID auto_increment,dc=OptionSets...

Attribute	Value
DN	cn=UID_GUID auto_incremen...
Common Name	UID_GUID auto increment
WebADM Object Type	OptionSet

Create Object

You are now on the **Option Set** configuration page :

Object Settings for cn=UID_GUID auto_increment,dc=OptionSets,dc=WebADM

Disable Option Set Yes No (default)

Target Subtree

The LDAP tree the optionset applies to.

Tree Root Context

Set a forced LDAP tree view base for any administrators existing inside the target subtree.
The tree root context will filter SQL audit logs entries based on the user DN in every entry.
Note: Does not apply for super administrators.

Unicity.Check.Context

Context within which unique attributes unicity is verified.

Certificate Signing Mode Rsign (Default)

Rsign: Use embedded WebADM Rsign PKI to sign certificate requests (recommended).
External: Use HTML forms with copy/paste (needed for using an external CA).

WebADM Account Quota

The quota represents the maximum number of activated WebADM accounts the subtree may contain.
Quotas can be defined at several levels in the LDAP tree.
WebADM will recursively check the number of activated accounts honors any quota in the chain.
Note: Does not apply for super administrators.

LDAP Creation Defaults

Comma-separated list of default attribute values automatically filled when creating LDAP objects.
Syntax: Attr1=Value1, Attr2=Value2...

Apply **Cancel** **Reset**

Configure the root LDAP treebase for the 3 first settings and click **Apply**. For Active Directory it should be something like dc=domain,dc=com according to your domain.

The OptionSet configuration is done and UIDnumber and GIDnumber will be automatically increased during user/group activation.

5.4 Active Directory Permissions

If you are working with Active Directory and during the UNIX extension you have a failure, it's probably due to rights permissions. That means your super_admin doesn't have enough rights to add the UNIX class to the user and/or to write values on UNIX attributes. To fix it, login on the Active Directory server and run the following command through Powershell:

```
dsacls "CN=Users,DC=test,DC=local" /I:T /G 'TEST\webadm_admins:WPRP;objectClass'  
dsacls "cn=users,dc=test,dc=local" /I:T /G 'TEST\webadmadmin:WPRP;gidnumber'  
dsacls "cn=users,dc=test,dc=local" /I:T /G 'TEST\webadmadmin:WPRP;uidnumber'  
dsacls "cn=users,dc=test,dc=local" /I:T /G 'TEST\webadmadmin:WPRP;unixhomedirectory'  
dsacls "cn=users,dc=test,dc=local" /I:T /G 'TEST\webadmadmin:WPRP;loginshell'  
dsacls "cn=users,dc=test,dc=local" /I:T /G 'TEST\webadmadmin:WPRP;description'  
dsacls "cn=users,dc=test,dc=local" /I:T /G 'TEST\webadmadmin:WPRP;gecos'
```

Note that `cn=users,dc=test,dc=local` is the user search base defined in WebADM Local Domain, `TEST` is my NetBIOS domain name and `webadmadmin` is my `super_admin` account.

For writing on AD administrators, rights previously settled are not enough because AdminSDHolder overwrites these rights every hour. So we need also to apply these rules on AdminSDHolder object and wait one hour that it's applied on all admin users and groups of the domain:

```
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /I:T /G  
'TEST\webadm_admins:WPRP;objectClass'  
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /I:T /G  
'TEST\webadmadmin:WPRP;gidnumber'  
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /I:T /G  
'TEST\webadmadmin:WPRP;uidnumber'  
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /I:T /G  
'TEST\webadmadmin:WPRP;unixhomedirectory'  
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /I:T /G  
'TEST\webadmadmin:WPRP;loginshell'  
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /I:T /G  
'TEST\webadmadmin:WPRP;description'  
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /I:T /G  
'TEST\webadmadmin:WPRP;gecos'
```

Now, you should be able to perform the UNIX extension through WebADM GUI.

Within the extended LDAP object, click on SSH Public Key Server (Actions box) to generate a SSH Private Key for the user:

1. In Application Action box, click on `SSH Public Key Server (3 actions)`, and select the first item `Register / Unregister SSH Public Key`.

WebADM Enterprise Edition v1.5.2
Copyright © 2010-2016 RCDevs SA, All Rights Reserved

Home | Admin | Cluster | Create | Search | Import | Databases | Applications | About | Logout

SpanKey User Actions for CN=test,CN=Users,DC=yorcdevs,DC=com (3)

Find below the user actions supported by **SSH Public Key Server (SpanKey)**.

 **Register / Unregister SSH Public Key**
You can use this action to generate an SSH key pair or register an inventoried PIV device.

 **Set or Change Key Expiration**
You can use this action to update the expiration date for a registered SSH public key.

 **Test Authorized Keys**
You can use this action to test public key retrieval with SpanKey.

[Cancel](#)

2. Configure your preferred Key Format and Key Length.
3. Configure key expiration (optional).
4. Click on **Register**.

WebADM Enterprise Edition v1.5.2
Copyright © 2010-2016 RCDevs SA, All Rights Reserved

Home | Admin | Cluster | Create | Search | Import | Databases | Applications | About | Logout

Register / Unregister SSH Public Key for CN=test,CN=Users,DC=yorcdevs,DC=com

You can use this form to create a new SSH private key. Please click 'Register' to start generating your key pair.
Warning: Only RSA private keys can be exported as PPK file for use with PuTTY.

Username:

 Generate a new SSH key private key
 Register a hardware key (Inventoried)

Key Format:

Key Length:

Expiration: [Edit](#)

[Register](#) [Cancel](#)

Your Public and Private Key are now generated by SpanKey server. Choose the format of the Private Key (OpenSSH or Putty) and click on Download Private Key button.



Register / Unregister SSH Public Key for CN=test,CN=Users,DC=yorcdevs,DC=com

The following private key can be used with your SSH client(s). Note that it will not be available anymore after quitting this page.
Please copy the private key block below or click the 'Download' button to save the private key file.

Private Key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEALynFYD54RDXUjhkDSRf3EiguGFGm+5J4D9UIbEORadweZrjY
kDCUD2h/Z/rBe2cu/EVNYQOp9z13YHpoIaojoXU7ddtiKhq9vPWUdupizY+85l6f
jwV2vautjimoBR1Tx2UCmxXa2IAbxS2+RemG7n61AwNx9x/tpmrbXmx3VMAcF1a2
L5sRUAdJNjn7y7D1UNnm2dQdwmcC07BhWei452RNSLgYYzXXQrxhsOy823KjdUo0
buowsPehlk14aUnJ816kUoRwI39HMLwGqV/TrrSyEJbbgL/N/MabS8CnWBAXM3R
bF02qz23/odm+kgpU25ZEvDTCaM2sON/vySL0wIDAQABAcIBAAzGzjHodH8kBm6Q
4Yp7vTT0cOfMBi/ldlOn+IXDiiYe0pI1Rf+f3veK2qPoNWLVUJrmUvOwi/N75ki8
fqNVDKMNcKyjt3HwgfikpPAE64IDv3ht3xUpuFKNCtGkV1GEjXVyaHQWvHGWFI1W
3LR9fPVgxegUjryo+v9okuhidqYRHyeR1XfFPqnlikvdK5B9+yD9wkuvXYUWADK
HuPngm02yq5Ns1V+jS9bwuAPUb/fuwNMQV3ipbR0drpR9S4PovE8c2kEF1Q9MIxr
ZY2MON15HMwb4TkG7-wlp4AXLsfhN9qHqu8ZK/GbyoLdxMfngeiJ0cZyiTXjxdt
TJ2MQOUcgYEAvtrXbvXQtYkah/MW9u5mZatNCZvyq9XRdr/tqzfqb15OKn4PJ5P
k7ybUMWh22gJFmmeen74Rk4G40op/RxtUBJ/JVrongXU7SB4k7kg5gSxcqr3j/u
Lm85alb+MsUUckXxuQLsE+6538bGV00WQotG6jJrgYM/7WhEuThvlHfUCgYEA6uTe
SMDFzVhkKibMTVxRAQ0+DViqEdTv2gqnhKnlyA+2Ky/hiffAibslxJDxsOp/QRcm
np87pS2rzFoOsYNRrdsLgjl93aci0hjfzvUFBu7tGjX6ifr/hqBNqwPmRTxMvKPlhr
1DNyE5Kr9iuEx5henqgPme+qQysj+J12sa1cXacCgYEAhfkGNb/7AD7yTA6XigR
2R7y4Zt26bwY1+OdsBwK/wlnjBx3Rd0crN9VNOBAX1r4bA4QVcl0j4qbhIKFcOp5
zpQEPVbWU4dEJ2zqL6evQLP+zBr5zERacQ69KE-i+sp0uYYD9Yof02641PL246ra5
RZoxKgt1nh9ZDoBw/+QQ0sECgYE09wbJtJCWEEL3lgwt5sVj64s7kUDT0w6//gG0
6Htt7NLG9M3Yson68ACPQ2rdpg8Nsz9XhVlpMcia57WGheauCyFCHja4NdGYc87
fjzeoqkS7CjYmz3i+Nzfx1td2+/jlafynBQ4Bi25Pd7+qexggV6tCDScyayH/3
dKfdYBcCgYA33ww6GKGIRv9W1hOK1lzzmmGv5yfxjJ1YxuKlxZ6c8pdhwY2hR3bdp
D9cTtQy0aM7AXT1iwX5bf2n1T3Xaxdk1bPE50lfeyKY49ewtCJ9knYavz0QgL6Cc
JEpSzEDbsCKMGgAzoR8E67+lg9jPq4GcVV2v5ap5Ryr0LwKKy5wuRQ==

-----END RSA PRIVATE KEY-----
```

Export Format: OpenSSH (PEM)
 PuTTY (PPK)

Export Password:

[Download Private Key](#)

Ok

Note

Register or Unregister of SSH Key can also be done through WebADM User Self-Services UI.

Now you can use the generated private key with your LDAP account, through SSH client or Putty and on any server where SpanKey Client is installed on. Without needing to deploy the user's public keys in authorized_keys files. To test, connect with your private key on a server managed by SpanKey client, like below:

```
ssh -i MyPrivateKey.pem test@192.168.3.178
[test@192.168.3.178 ~]#
```

6. Video Tutorial


[Play Video on Youtube](#)

without the prior written permission of RCDevs S.A. The latter especially applies for data processing systems. RCDevs S.A. also reserves all communication rights (lectures, radio and television). The hardware and software names mentioned in this manual are most often the registered trademarks of the respective manufacturers and as such are subject to the statutory regulations. Product and brand names are the property of RCDevs S.A. © 2019 RCDevs SA, All Rights Reserved