



SMART CARD - PIV

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 Security.

Copyright (c) 2010-2023 RCDevs Security 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.

Smart Card - PIV

[smartcard](#) [yubikey](#) [PIV](#)

Authentication with a Yubikey Smart Card / PIV

In this How-To we will configure a user in WebADM for using a PIV key. We need a WebADM server already configured.

1. Import the Inventory

We need to create an inventory file like this:

```
"Type", "Reference", "Description", "DN", "Data", "Status"  
"PIV Device", "<ID1>", "PIV Yubikey", "", "PublicKey=<pub_key1>", "Valid"  
"PIV Device", "<ID2>", "PIV Yubikey", "", "PublicKey=<pub_key2>", "Valid"  
"PIV Device", "<ID3>", "PIV Yubikey", "", "PublicKey=<pub_key3>", "Valid"
```

For my test, I have a Yubikey Nano with a PIV certificate and I use [yubico-piv-tool](#) for the management of the Yubikey, but it can work with other PIV keys.

We need to extract the public key. I do it with `yubico-piv-tool` and `openssl`:

```
[john@Mac-mini ~]$ yubico-piv-tool -aread-cert -s9a | openssl x509 -pubkey -noout  
-----BEGIN PUBLIC KEY-----  
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwjYEZhuhF9rrxHdCDstG  
J2ibVVrJhrZIfz4wwjrXtwEACJP2wWRe9dvNw5h3CrbguSc1l8mkKrfNwxAkGM0p  
MIx5KgNBaDMc0ggmjJFT0BIK4muJjdUZKhR3oFwBD/jjR701lGinYK873lYz01aS  
nf7j00wgTl4kU3V+sJEbI9t3cQHfE6DMMWeG8w3Q03z+fVkNN9f30TvvBDua95Qg  
G9m5eMtGqlrnPuovErHagfg8kd5lZFKy0akaoAhb0W6oQ8s8YKzCP1evcjLYe/o  
8K4br8vwp0jnBaKNKbVp08iAn1A0UTXWaKUytb3cYqMvzp9UYh5Vyfl4MtMh8ULP  
wwIDAQAB  
-----END PUBLIC KEY-----
```

Another way that works with other keys/cards (Feitian, electronic ID, ...) is to do this with `opencsc` and `pcsc-lite`. Once they are installed, you need to run these commands:

```
[root@fedora28 ~]# pkcs15-tool --list-key
Using reader with a card: Yubico Yubikey 4 OTP+CCID 00 00
Private RSA Key [PIV AUTH key]
Object Flags   : [0x1], private
Usage          : [0x2E], decrypt, sign, signRecover, unwrap
Access Flags   : [0x1D], sensitive, alwaysSensitive, neverExtract, local
ModLength     : 2048
Key ref       : 154 (0x9A)
Native        : yes
Auth ID       : 01
ID            : 01
```

```
[root@fedora28 ~]# pkcs15-tool --read-public-key 1
Using reader with a card: Yubico Yubikey 4 OTP+CCID 00 00
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwjYEZhuhF9rrxHdCDstG
J2ibVVrJhrZIfz4wwjrXtwEACJP2wWRe9dvNw5h3CrbguSc1l8mkKrfNwxAkGMOp
MIx5KgNBaDMcOggmjJFT0BIK4muJjdUZKhR3oFwBD/jjR701lGinYK873lYz01aS
nf7j00wgTl4kU3V+sJEbI9t3cQHfE6DMMWeG8w3Q03z+fVkNN9f30TvvBDua95Qg
G9m5eMtGqlrnPuovErHagfg8kd5lZFkY0akaoAhb0W6oQ8s8YKzCP1evcjfLYe/o
8K4br8vvp0jnBaKNKbVp08iAn1A0UTXWaKUytb3cYqMvzp9UYh5Vyfl4MtMh8ULP
wwIDAQAB
-----END PUBLIC KEY-----
```

We can create a file called `piv.csv` with the serial number as ID and the right public key:

```
"Type","Reference","Description","DN","Data","Status"
"PIV Device","8671120","PIV
Yubikey","","PublicKey=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwjYEZhuhF9rrxHdCDstGJ2
```

We import the file. Under the `Import` tab, we click on `Import Inventory File`:

We choose the `piv.csv` file and click on `Import`:

Now, the PIV key is present in the inventory:

2. Assign the Yubikey

We select the user in the LDAP tree on the left and add the `UNIX Account` extension:

We click on `Proceed`:

We `Extend Object`:

We click on `SSH Public key server`:

We click on `Register/Unregister SSH key`:

We select `Register a hardware key (Inventoried)`, enter the `Serial Number` (Reference) and `Register`:

Now, the PIV key is well registered.

3. Test with SSH

We'll try with a CentOS 7 as an ssh server.

We install and configure `spankey_client` on it:

```
[root@test_vm ~]$ yum install https://repos.rcdevs.com/redhat/base/rcdevs_release-1.1.1-1.noarch.rpm
[root@test_vm ~]$ yum clean all
[root@test_vm ~]$ yum install spankey_client -y
[root@test_vm ~]$ spankey_setup
This is the configuration tool for RCDevs SpanKey Agent.
It will configure SpanKey Server URL(s), SSH helper and NSS.

Do you have a WebADM cluster or standalone server (c/s)? s
Enter hostname or address for SpanKey server: my_webadm
Do you want to enable SpanKey for OpenSSH server (y/n)? y
Do you want SpanKey agent to auto-create home directories (y/n)? y
Do you want to enable SSH session management options (y/n)? y
Do you want to enable SpanKey NSS plugin (y/n)? y
SpanKey Agent for SpanKey standalone Server
Server URL: https://192.168.3.202:8443/spankey/ (Server Ok)
Enable SpanKey for OpenSSH server: Yes
Auto-create home directories: Yes
SSH session management options: Yes
Enable SpanKey NSS plugin: Yes

Do you confirm (y/n)? y

Updating /etc/spankey/spankey.conf... Ok
Updating /etc/ssh/sshd_config... Ok
Updating /etc/nsswitch.conf... Ok
Updating /etc/pam.d/password-auth... Ok
Created symlink from /etc/systemd/system/multi-user.target.wants/nscd.service to
/usr/lib/systemd/system/nscd.service.
Created symlink from /etc/systemd/system/sockets.target.wants/nscd.socket to
/usr/lib/systemd/system/nscd.socket.

SpanKey Agent has been successfully configured.
```

For the ssh client, we use a mac mini. We configure it for using the smartcard:

```
[John@Mac-mini ~]$ brew install opencsc
[John@Mac-mini ~]$ export OPENCSC_LIBS=$(brew --prefix opencsc)/lib
```

We try the authentication:

```
[John@Mac-mini ~]$ ssh -I $OPENCSC_LIBS/opencsc-pkcs11.so John@test_vm
Enter PIN for 'PIV Card Holder pin (../piv_II)':
bash-4.2$
```

I'm connected to the server with a user from the LDAP database and authenticated with my PIV key.

This manual was prepared with great care. However, RCDevs Security S.A. and the author cannot assume any legal or other liability for possible errors and their consequences. No responsibility is taken for the details contained in this manual. Subject to alteration without notice. RCDevs Security S.A. does not enter into any responsibility in this respect. The hardware and software described in this manual is provided on the basis of a license agreement. This manual is protected by copyright law. RCDevs Security S.A. reserves all rights, especially for translation into foreign languages. No part of this manual may be reproduced in any way (photocopies, microfilm or other methods) or transformed into machine-readable language without the prior written permission of RCDevs Security S.A. The latter especially applies for data processing systems. RCDevs Security 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 Security. © 2023 RCDevs Security S.A., All Rights Reserved