SMART CARDS IN LINUX

AND WHY YOU SHOULD CARE

Jakub Jelen

Red Hat

jjelen@redhat.com

PRIVATE KEYS, CERTIFICATES: WHAT ARE THEY USED FOR?

PRIVATE KEYS, CERTIFICATES: WHAT ARE THEY USED FOR?

- Email signatures & encryption
- SSH authentication
- Git commit/tag signing
- TLS client authentication (eGovernment)
- More secure password replacement

WHERE ARE THEY STORED?

- Hard drive
- Computer memory

ARE THEY SECURE?

DIRTY COW

SMART CARD: DEDICATED HW

THE MOST OBVIOUS CREDITCARD-SIZE FORM

"SMART CARD": HW TOKEN

MORE PRACTICAL FORM

IN THIS TALK

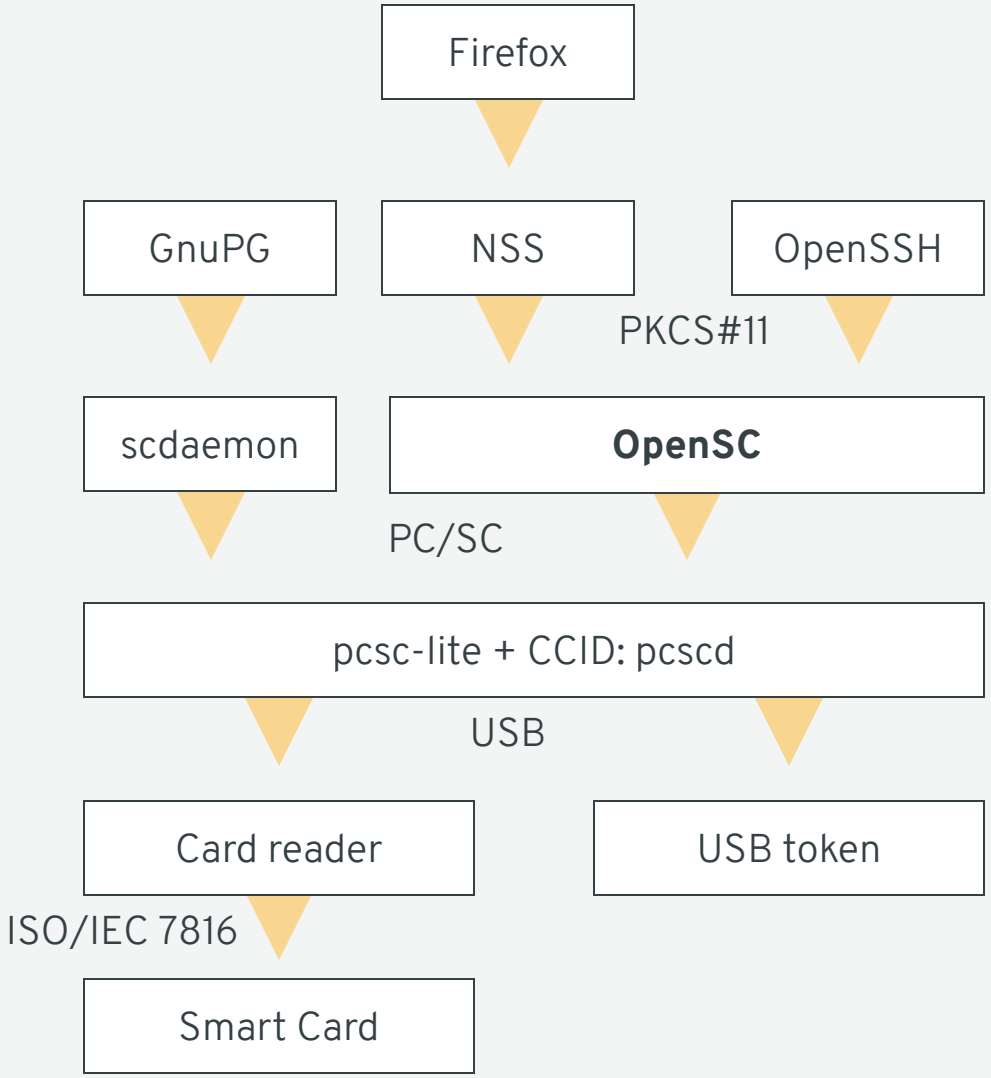
- Anatomy of smart card and software
- OpenSC project
- Practical examples
  - Smart Card
  - Other features
- Troubleshooting

ANATOMY OF SMART CARD

AND ITS SOFTWARE STACK

ANATOMY

- Smartcard
- ISO/IEC 7816 - closed :(
  - Electrical specification
  - Commands (APDU)
- pcsc-lite, CCID
  - PC/SC protocol
  - Chip card interface device
  - pcscd system daemon
- OpenSC
  - drivers for cards
  - exposing PKCS#11 interface
- PKCS#11 interface
  - for applications/libraries
- Applications, Libraries



OPENSC PROJECT

OPEN SOURCE SMART CARD TOOLS AND MIDDLEWARE

OpenSC

OPENSC PROJECT

- Card drivers
  - Most of current cards (almost 40)
  - PIV, OpenPGP, CardOS, myEID
  - Contributions: CAC, Coolkey (RHCS)
- Multiplatform (Linux, Mac, Windows, ...)
- Exposes PKCS#11 interface for other applications
  - Way to read, write and operate on keys
  - Prevents reading private data
- Testing
  - Mostly manual
  - CI running PKCS#11 testsuite for "our" cards

EXAMPLES

HOW CAN I DO ... WITH A SMART CARD?

(assuming already provisioned card with preloaded keys)

EXAMPLES

- Card inspection
- Atomic operations
- OpenSSH client
- sudo
- TLS Client Authentication
- Concurrent access
- GnuPG

CARD INSPECTION

PC/SC level (pcsc-tools)

```
$ pcsc_scan
PC/SC device scanner
V 1.4.25 (c) 2001-2011, Ludovic Rousseau
<ludovic.rousseau@free.fr>
Compiled with PC/SC lite version: 1.8.22
Using reader plug'n play mechanism
Scanning present readers...
0: OMNIKEY AG CardMan 3121 00 00

Thu Jan 11 15:52:13 2018
Reader 0: OMNIKEY AG CardMan 3121 00 00
  Card state: Card inserted, Shared Mode,
  ATR: 3B FF 14 00 FF 81 31 FE 45 80 25 A0 00 00 00 56 57 53 43
36 35 30 03 03 38
[...]
```

CARD INSPECTION

PKCS#11 level: Token (opensc)

```
$ pkcs11-tool --list-slots
Available slots:
Slot 0 (0x0): OMNIKEY AG CardMan 3121 00 00
  token label        : jjelen (jjelen)
  token manufacturer : 534e SafeNet
  token model        : PKCS#15 emulated
  token flags        : login required, token initialized, PIN
initialized
  hardware version   : 0.0
  firmware version   : 0.0
  serial num         : 4e06500042005002
  pin min/max        : 4/32
```

CARD INSPECTION

PKCS#11 level: Objects (opensc)

```
$ pkcs11-tool --list-objects --login
Using slot 0 with a present token (0x0)
Logging in to "jjelen (jjelen)".
Private Key Object; RSA
  label:      signing key for jjelen
  ID:         01
  Usage:      sign
Public Key Object; RSA 1024 bits
  label:      signing key for jjelen
  ID:         01
  Usage:      verify
Certificate Object; type = X.509 cert
  label:      signing key for jjelen
  ID:         01
[...]
```

ATOMIC OPERATIONS

Download the certificate from a card and show its content

```
$ pkcs11-tool --read-object --id 01 --type cert \
    --output-file cert.der
Using slot 0 with a present token (0x0)
$ openssl x509 -inform DER -in cert.der > cert.pem
$ openssl x509 -in cert.pem -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 41 (0x29)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: O = sjc.redhat.com Security Domain, CN = CA
Signing Certificate
        Validity
            Not Before: Jul 15 20:57:58 2016 GMT
            Not After : Jul 14 20:57:58 2021 GMT
        Subject: CN = Jakub Jelen, O = Token Key User, UID =
jjelen
```

ATOMIC OPERATIONS

Signature & Verification from command-line

```
$ pkcs11-tool --sign --id 01 --mechanism RSA-PKCS --login \
    --input-file data --output-file data.sig
Using slot 0 with a present token (0x0)
Logging in to "jjelen (jjelen)".
Please enter User PIN:
Using signature algorithm RSA-PKCS
$ openssl rsautl -verify -certin -inkey cert.pem \
    -in data.sig
[original signed data]
```

OPENSSH CLIENT

- List public keys on the smart card in OpenSSH format

```
$ ssh-keygen -D /usr/lib64/pkcs11/opensc-pkcs11.so
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQCXBle8[...]2YuRJF6AuwrpQ==
```

- Install the keys to the server
- Connect to server

```
$ ssh -I /usr/lib64/pkcs11/opensc-pkcs11.so example.com
Enter PIN for 'PIV_II (PIV Card Holder pin)':
```

- Store permanent configuration in client configuration

```
$ cat ~/.ssh/config
Host example.com
  PKCS11Provider /usr/lib64/pkcs11/opensc-pkcs11.so
```

- RSA keys only (OpenSSH bug #2474)

OPENSSH CLIENT (SSH-AGENT)

- Start ssh-agent (does not work with gnome-keyring):

```
$ test -e "$SSH_AUTH_SOCK" || eval $(ssh-agent)
```

- Add a card:

```
$ ssh-add -s /usr/lib64/pkcs11/opensc-pkcs11.so
Enter passphrase for PKCS#11:
Card added: /usr/lib64/pkcs11/opensc-pkcs11.so
```

- Connect to server:

```
$ ssh example.com
```

SUDO (PAM_SSH_AGENT_AUTH)

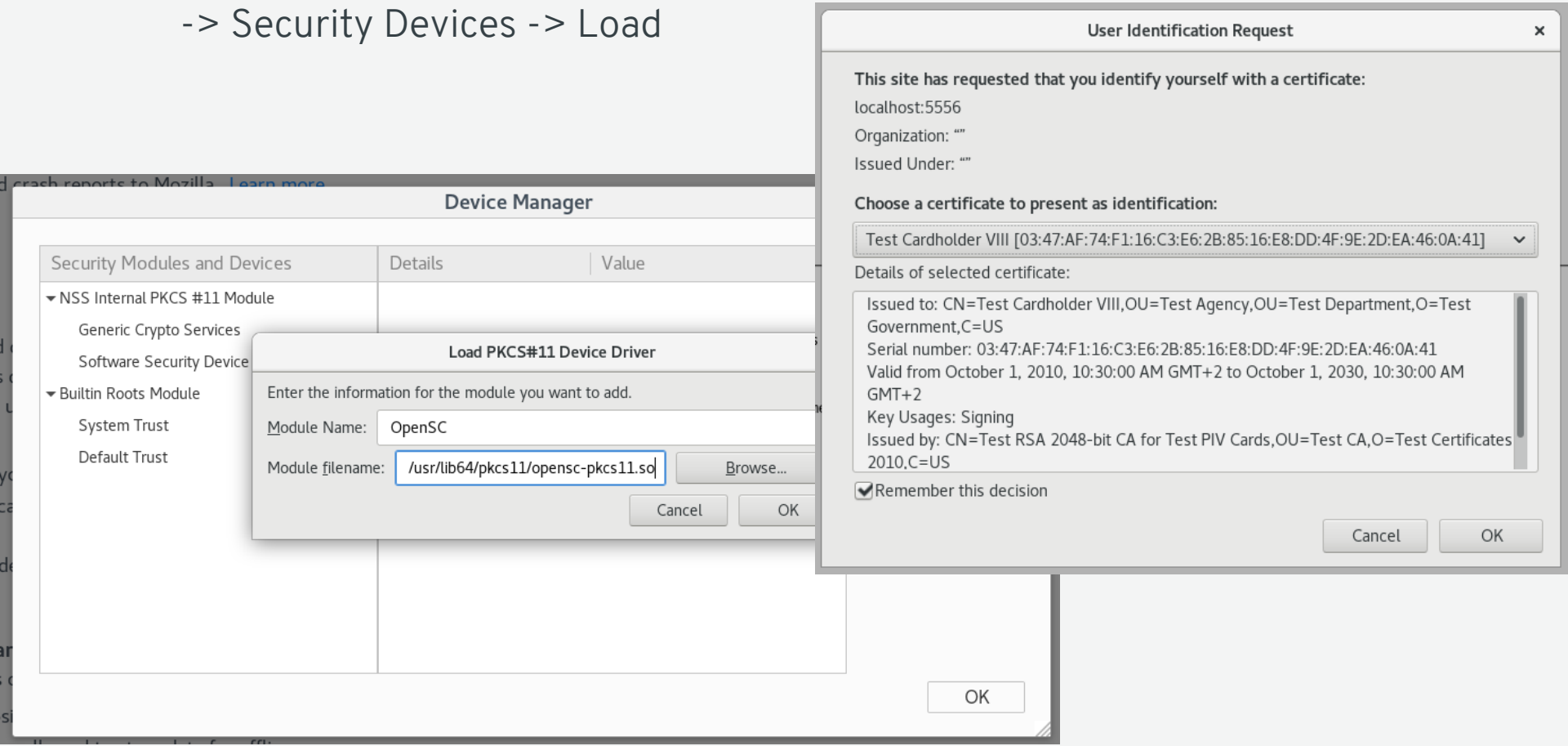
- Set up ssh-agent as in previous slide
- Store public key in
  - `/etc/security/authorized_keys`
- Configure sudo through pam:

```
$ cat /etc/pam.d/sudo
...
auth sufficient pam_ssh_agent_auth.so \
     file=/etc/security/authorized_keys
```

- Even on remote hosts (forwarded ssh-agent)

TLS CLIENT AUTHENTICATION

- Firefox -> Preferences -> Privacy&Security -> Security -> Security Devices -> Load



CONCURRENT ACCESS

- Configuration: opensc.conf

```
drivers = PIV-II; # speed up detection and avoid mismatches
reader_driver pcsc {
    disconnect_action=leave; # do not break concurrent sessions
}
```

- OpenSSH ssh-agent: long-running session

```
eval `ssh-agent` && ssh-add -s /usr/lib64/opensc-pkcs11.so
ssh example.com
```

- pkcs11-tool: ad-hoc commands

```
pkcs11-tool --login --sign --id02 -mRSA-PKCS -i data -o data.sig
```

- Some applications require exclusive access (GnuPG sdaemon) :(
- More applet on a single card = problems

GNUPG

- Email, git commit signing
- GnuPG's scdaemon
  - not using PKCS#11 to access OpenPGP applets
  - directly accessing PC/SC with **exclusive** access
  - preventing other applications to use the card
- gnupg-pkcs11-scd
  - Accessing cards using PKCS#11
  - More complicated configuration

KERBEROS

- pkinit: pre-authentication (RFC 4556)
  - Certificate and signature from PKCS#11
  - krb5.conf
  - `pkinit_identity = PKCS11:`
- FreeIPA 4.5: Mapping certificates to users
  - Whole blobs X.509 blobs
  - Flexible mapping rules
  - replacing pam_pkcs11

EXAMPLES

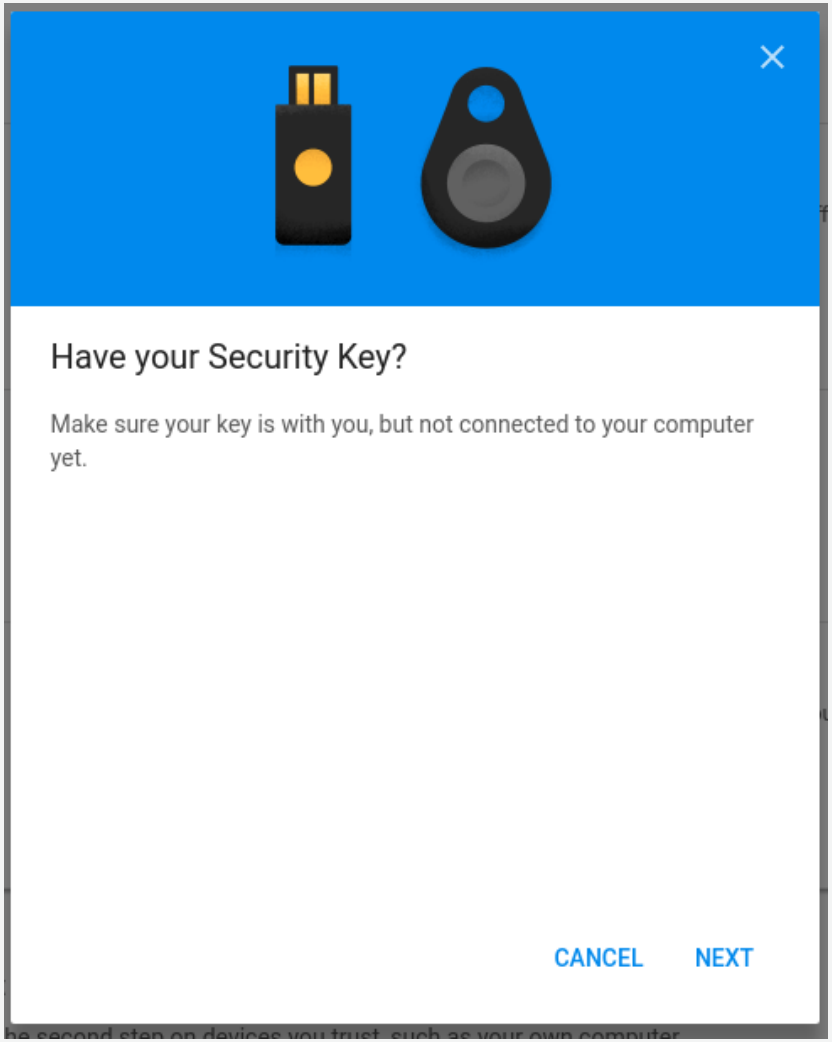
WHAT CAN I DO WITH OTHER HARDWARE TOKENS?

EXAMPLES

- Yubikey, Nitrokey, Feitian
- 2nd factor authentication
  - FIDO U2F
  - OTP
  - Yubico OTP
- Does not verify PIN
  - can not be the only factor!

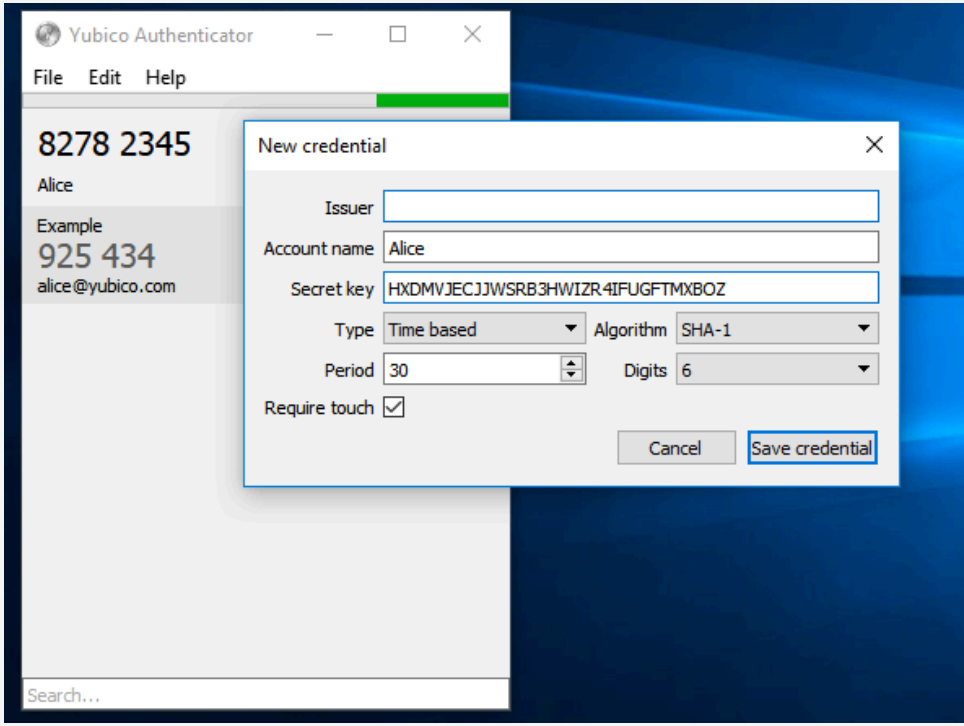
FIDO U2F

- FIDO Universal 2nd Factor
- Support:
  - Chromium out of box
  - Firefox 57: about:config
    - security.webauth.u2f = true
- Use cases
  - Fortify authentication to websites
  - Local login (pam_u2f)
- Alternative to SMS or OTP apps
- Physical verification with touch



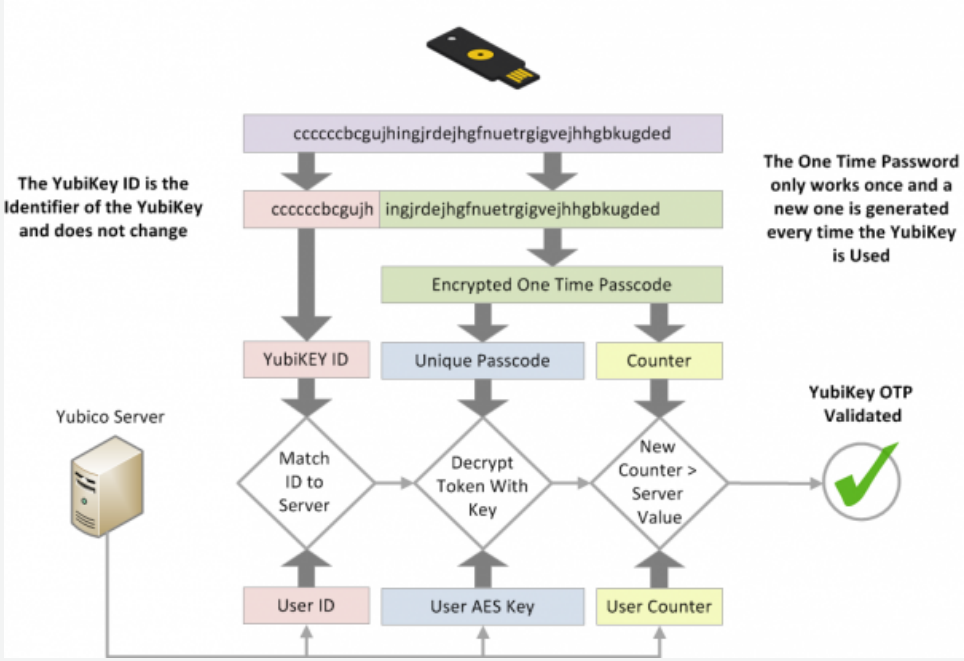
OATH-HOTP/TOTP

- One-Time Password
  - Standard OATH
  - HMAC hash-based
  - Securely stored secret key
- Client:
  - Yubikey Authenticator
  - + Android version
- Server (verification):
  - Usually with PAM module
- Physical verification with touch



YUBICO OTP

- One-Time Password
  - Yubico-version
- Client:
  - no drivers needed
  - USB HID keyboard
- Server (verification):
  - Usually with PAM module
- Physical verification with touch
- AES encryption



TROUBLESHOOTING

WHAT COULD GO WRONG?

TROUBLESHOOTING SMART CARD

- Is the reader/USB device detected?
  - $ lsusb
- Is the card detected in pcsc-lite?
  - $ pcsc_scan
- PCSC trace (APDU messages)
  - ```
    $ systemctl stop pcscd
    $ sudo LIBCCID_ifdLogLevel=0x000F pcscd --foreground --debug
    --apdu --color
    ```
- Is the card detected in OpenSC?
  - `pkc11-tool -L`
- PKCS#11 level trace:
  - `export PKCS11SPY=/usr/lib64/pkcs11/opensc-pkcs11.so`
  - `pkcs11-tool -L /usr/lib64/pkcs11-spy.so`
- OpenSC debug logs:
  - `OPENSC_DEBUG=9 pkc11-tool -L`
    ```

SMART CARDS SUMMARY

- Not only cards!
- Stores private keys securely
- PKCS#11 interface for developers
- Can replace passwords
- Can strengthen passwords:
  - U2F or OTP for second factor

Thank you for your attention