Michael'grisu'Bramer

Login per USB-Stick, etc

2. März 2008

Hinweis

- Diese Folien sind nach dem Vortrag leicht erweitert worden. So sind für die Demo-Einschübe entsprechende Folien dazu gekommen, wo die wichtigsten Zeilen drauf stehen.
- Sonst wird auf die Quellen verwiesen.
- Im Vortrag hat die Vorführung von pam_usb ja nicht funktioniert. Das Problem war, das pro User maximal ein USB-Stick eingerichtet werden darf. Und ich hatte vorher schon einen Stick eingerichtet und daher gab es das Problem im Vortrag.

Agenda

- Warum?

Agenda

- Warum?
- PAM-System...

Agenda

- Warum?
- PAM-System...
- Login per USB-Storage...

Agenda

- Warum?
- PAM-System...
- Login per USB-Storage...
- Login per bluetooth...

Agenda

- Warum?
- PAM-System...
- Login per USB-Storage...
- Login per bluetooth...
- Asymmetrisches Kryptosystem...

Agenda

- Warum?
- PAM-System...
- Login per USB-Storage...
- Login per bluetooth...
- Asymmetrisches Kryptosystem...
- Login per cryptokeys...

Warum?

- Zugang ist an HW gebunden (Token, passwortloses Login)

Warum?

- Zugang ist an HW gebunden (Token, passwortloses Login)
- Passwörter sind oft zu einfach

Warum?

- Zugang ist an HW gebunden (Token, passwortloses Login)
- Passwörter sind oft zu einfach
- bequeme Sicherheitsrichtlinien

Warum?

- Zugang ist an HW gebunden (Token, passwortloses Login)
- Passwörter sind oft zu einfach
- bequeme Sicherheitsrichtlinien
- sichere Key-Verwaltung

Warum?

- Zugang ist an HW gebunden (Token, passwortloses Login)
- Passwörter sind oft zu einfach
- bequeme Sicherheitsrichtlinien
- sichere Key-Verwaltung
- Fun

PAM-System

- Modulares Zugangsystem

PAM-System

- Modulares Zugangsystem
- Beispiel shadow-System, Karten-Systeme

Login per USB-Storage

- Prinzip

Login per USB-Storage

- Prinzip
- Wie geht es?
  - Stick vorbereiten
  - Nutzer auf dem Stick einrichten
  - PAM einrichten

Login per USB-Storage

- Prinzip
- Wie geht es?
  - Stick vorbereiten
  - Nutzer auf dem Stick einrichten
  - PAM einrichten
- Demo

der Code

- `apt-get install libpam-usb pamusb-tools`
- `pamusb-conf --add-device MyDevice`
- `pamusb-conf --add-user root`
- `pamusb-check root`
- `/etc/pam.d/common-auth:`
- `auth sufficient pam_usb.so`
- `auth required pam_unix.so nullok_secure`

Login per USB-Storage, pamusb-agent

- Prinzip

Login per USB-Storage, pamusb-agent

- Prinzip
- pamusb-agent

Login per USB-Storage, pamusb-agent

- Prinzip
- pamusb-agent
- Demo

der Code

- `/etc/pamusb.conf:`
- `<user id="USER">`
- `<device>MyDevice</device>`
- `<agent event="lock">dcop kdesktop KScreensaverIface lock</agent>`
- `<agent event="unlock">dcop kdesktop KScreensaverIface quit</agent>`
- `</user>`

Login per USB-Storage, Boot-Token

- Prinzip

Login per USB-Storage, Boot-Token

- Prinzip
- config
  - pamusb-vorbereiten (1 User pro Stick)

Login per USB-Storage, Boot-Token

- Prinzip
- config
  - pamusb-vorbereiten (1 User pro Stick)
  - framebuffer (lilo/grub)
  - apt-get install fbi
  - ein Bild
  - kdm/gdm aus init entfernen
  - eigenes Script, mit kdm/gdm-Autologin


debian
Willkommen
Bitte identifizieren Sie sich mit Ihrem USB-Stick!
warte auf USB-Stick...

der Code

- `http://www.pro-linux.de/berichte/automatisches-login-per-usb-stick.html`

Login per USB-Storage, Vor-/Nachteile

- einfach

Login per USB-Storage, Vor-/Nachteile

- einfach
- ohne besondere Hardware

Login per USB-Storage, Vor-/Nachteile

- einfach
- ohne besondere Hardware
- sehr flexibel

Login per USB-Storage, Vor-/Nachteile

- einfach
- ohne besondere Hardware
- sehr flexibel
- keine Schlüsselsicherheit

Login per USB-Storage, Vor-/Nachteile

- einfach
- ohne besondere Hardware
- sehr flexibel
- keine Schlüsselsicherheit
- one time pad

Login per bluetooth

- Prinzip

Login per bluetooth

- Prinzip
- Wie geht es?
  - lauffähiges bluetooth-subsystem
  - Config
  - PAM einrichten

Login per bluetooth

- Prinzip
- Wie geht es?
  - lauffähiges bluetooth-subsystem
  - Config
  - PAM einrichten
- Demo?

der Code

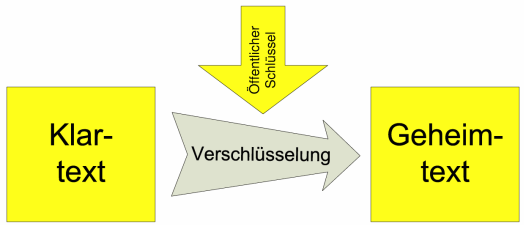
- `apt-get install libpam-blue`
- `hcitool scan`
- `/etc/security/bluesscan.conf:`
- `/etc/pam.d/common-auth:`
- `auth sufficient pam_blue.so`
- `auth required pam_unix.so nullok_secure`

Asymmetrisches Kryptosystem

- Public-Key Privat-Key

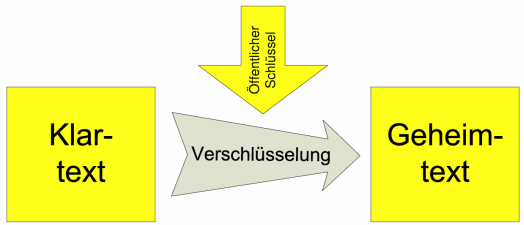
Asymmetrisches Kryptosystem

- Public-Key Privat-Key
- Verschlüsselung

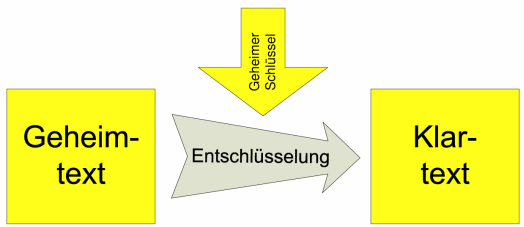


Asymmetrisches Kryptosystem

- Public-Key Privat-Key
- Verschlüsselung



- Entschlüsselung



Login per Crypto-Stick

- Prinzip

Login per Crypto-Stick

- Prinzip
- privat Key auf der Karte (sicher)

Login per Crypto-Stick

- Prinzip
- privat Key auf der Karte (sicher)
- pub Key öffentlich

Login per Crypto-Stick

- Prinzip
- privat Key auf der Karte (sicher)
- pub Key öffentlich
- intelligente Karte (smart-card)

Login per Crypto-Stick

- Prinzip
- privat Key auf der Karte (sicher)
- pub Key öffentlich
- intelligente Karte (smart-card)
- Wie geht es?
  - Stick vorbereiten

Login per Crypto-Stick

- Prinzip
- privat Key auf der Karte (sicher)
- pub Key öffentlich
- intelligente Karte (smart-card)
- Wie geht es?
  - Stick vorbereiten
  - Nutzer einrichten

Login per Crypto-Stick

- Prinzip
- privat Key auf der Karte (sicher)
- pub Key öffentlich
- intelligente Karte (smart-card)
- Wie geht es?
  - Stick vorbereiten
  - Nutzer einrichten
  - Key erzeugen

Login per Crypto-Stick

- Prinzip
- privat Key auf der Karte (sicher)
- pub Key öffentlich
- intelligente Karte (smart-card)
- Wie geht es?
  - Stick vorbereiten
  - Nutzer einrichten
  - Key erzeugen
  - pam einrichten (Pam_p11: simple, Pam-PKCS11)

theoretische Praxis

- Stick vorbereiten: `pkcs15-init --create-pkcs15`

theoretische Praxis

- Stick vorbereiten: `pkcs15-init --create-pkcs15`
- Nutzer einrichten:
- `pkcs15-init --store-pin --auth-id 01 --label "Michael Bramer"`

theoretische Praxis

- Stick vorbereiten: `pkcs15-init --create-pkcs15`
- Nutzer einrichten:
- `pkcs15-init --store-pin --auth-id 01 --label "Michael Bramer"`
- Key erzeugen:
- `pkcs15-init --generate-key rsa/1024 --auth-id 01`

theoretische Praxis

- Stick vorbereiten: `pkcs15-init --create-pkcs15`
- Nutzer einrichten:
- `pkcs15-init --store-pin --auth-id 01 --label "Michael Bramer"`
- Key erzeugen:
- `pkcs15-init --generate-key rsa/1024 --auth-id 01`
- key-export:
- `pkcs15-tool --read-ssh-key 45`
- nach .ssh/authorized_keys

theoretische Praxis

- Stick vorbereiten: `pkcs15-init --create-pkcs15`
- Nutzer einrichten:
- `pkcs15-init --store-pin --auth-id 01 --label "Michael Bramer"`
- Key erzeugen:
- `pkcs15-init --generate-key rsa/1024 --auth-id 01`
- key-export:
- `pkcs15-tool --read-ssh-key 45`
- nach .ssh/authorized_keys
- pam-config:
- `auth sufficient pam_p11_openssh.so /usr/lib/opensc-pkcs11.so`

misc

- PGP

misc

- PGP
- OpenSSL

misc

- PGP
- OpenSSL
- SSH

misc

- PGP
- OpenSSL
- SSH
- …

Quellen

1. http://www.pamusb.org/doc/quickstart
2. http://www.pro-linux.de/berichte/automatisches-login-per-usb-stick.html
3. http://pam.0xdef.net/doc.html
4. http://www.opensc-project.org/
5. man-Pages, READMEs