Safer Netboot

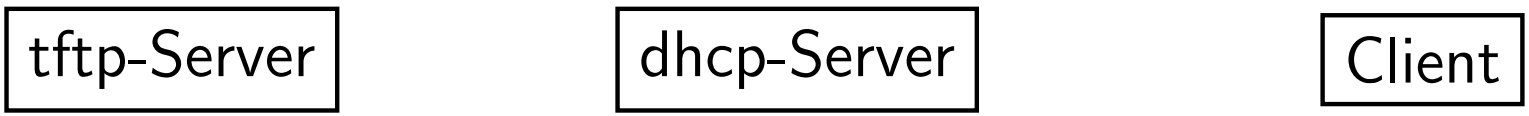
Absicherung des Preboot eXecution Enviromemts mit iPXE-ROM

Janik Jungjohann

Institut für Informatik
Humboldt-Universität zu Berlin

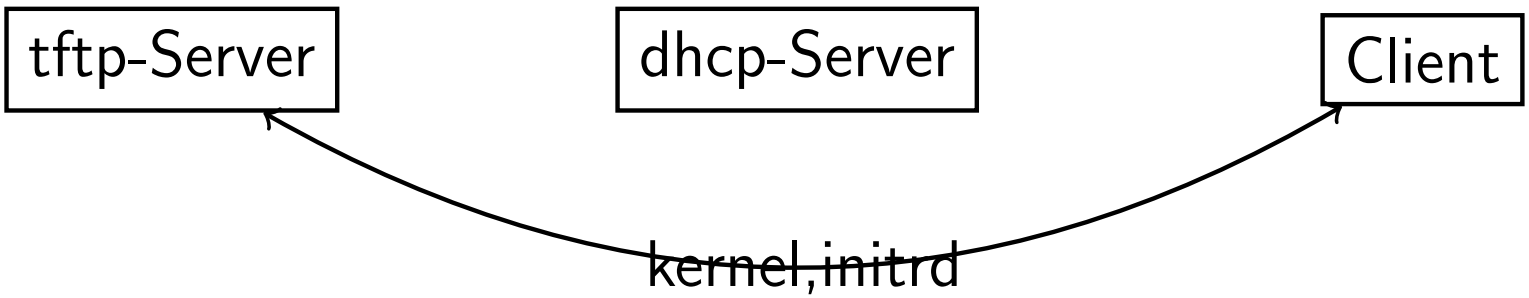
11. Oktober 2019

Motivation

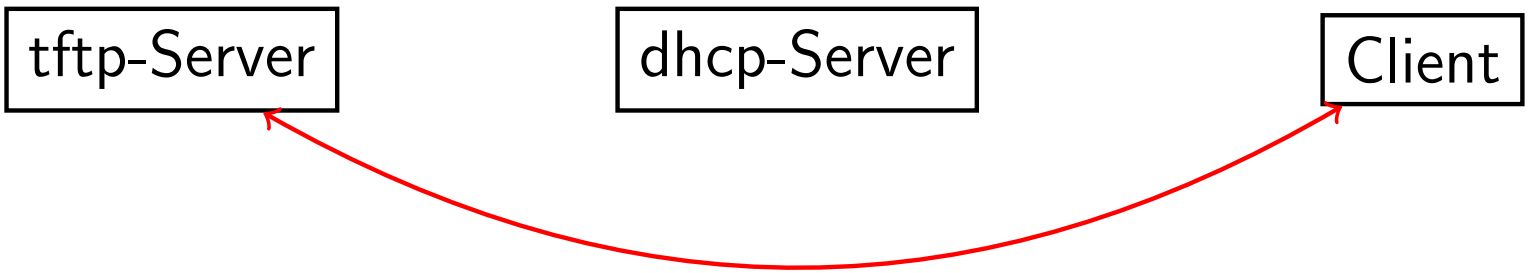


Motivation

Motivation

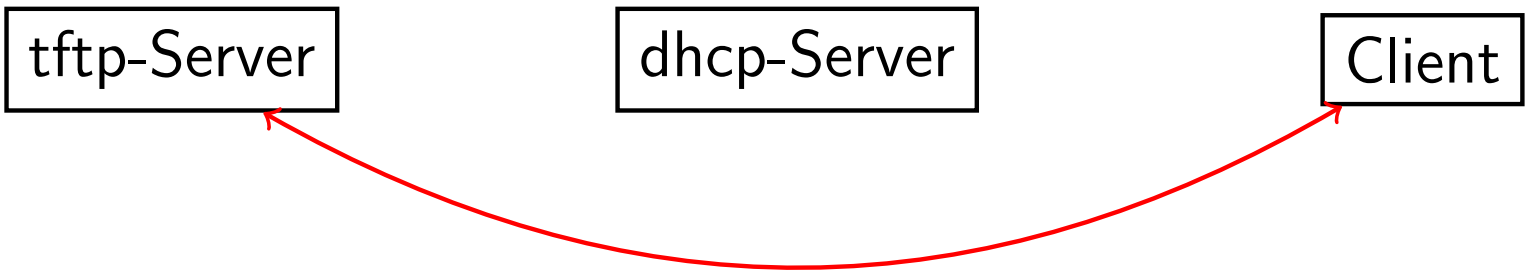


Motivation



- unverschlüsselt
- Man-in-the-Middle

Motivation



- unverschlüsselt
- Man-in-the-Middle
- keine Autorisierung des Clients
- prinzipiell jeder im Netzwerk darf booten
- Wunsch aus der RGB

Wie besser?

Wie besser?

- verschlüsselte Kommunikation

→ kein Man-in-the-Middle

Wie besser?

- verschlüsselte Kommunikation

→ kein Man-in-the-Middle

- Mechanismus für Authentisierung

Wie besser?

- verschlüsselte Kommunikation

→ kein Man-in-the-Middle

- Mechanismus für Authentisierung
- Lösung: HTTPS

Wie geht es weiter?

1. Motivation
2. Vorbereitung
3. Bauen
4. Booten

Was wird gebraucht?

Daten über HTTPS beziehen.

Was wird gebraucht?

Daten über HTTPS beziehen.

- HTTP-Server

Was wird gebraucht?

Daten über HTTPS beziehen.

- HTTP-Server
- HTTP-Client

Was wird gebraucht?

Daten über HTTPS beziehen.

- HTTP-Server
- HTTP-Client
- TLS-Layer: $\rightarrow$ Zertifikate

iPXE (I)

"iPXE is the leading open source network boot firmware. It provides a full PXE implementation enhanced with additional features such as:"

iPXE (I)

"iPXE is the leading open source network boot firmware. It provides a full PXE implementation enhanced with additional features such as:"

- HTTP(S)
- iSCSI
- FCoE
- AoE
- wireless

iPXE (I)

"iPXE is the leading open source network boot firmware. It provides a full PXE implementation enhanced with additional features such as:"

- HTTP(S)
- iSCSI
- FCoE
- AoE
- wireless
- ...

iPXE (II)

- Bootvorgang scriptabel
- Boot-ROM für physische Hardware
- chainloading möglich
- Bootprompt

Packen wir es an!

Wir brauchen Zertifikate:

- für den Server: Verschlüsselung
- für den Client: Autentisierung

Root CA

```
openssl req -x509 -newkey rsa:2048 -out ca.crt
-keyout ca.key -days 1000
```

Server

```
openssl req -newkey rsa -keyout server.key -out server.req
```

```
openssl ca -config ca.cnf -in server.req -out server.crt
```

Client

```
openssl req -newkey rsa -keyout client.key -out client.req
```

```
openssl ca -config ca.cnf -in client.req -out client.crt
```

Serverkonfiguration

```
<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile /etc/apache2/server.crt
    SSLCertificateKeyFile /etc/apache2/server.key
    SSLVerifyClient require
    SSLVerifyDepth 4
    SSLCACertificateFile /etc/apache2/ca.crt
</VirtualHost>
```

iPXE-Image

- Kompilieren der Quellen
- Optionen:
  - script ("EMBED=")
  - Zertifikate ("CERT=")
  - Vertrauen ("TRUST=")
  - privaten SChlüssel ("PRIV=")

iPXE-Script

```
#!ipxe

dhcp
kernel https://192.168.122.2/linux
initrd https://192.168.122.2/initrd
boot
```

make it!

```
make -j 4 bin/8086107c.mrom EMBED=boot.ipxe
          TRUST=ca.crt
          CERT=ca.crt,client.crt
          PRIVKEY=client.key
```

Intel Netzwerkkarte

ROM flashen

- verschiedene Werkzeuge (Intel, RealTek)
- "generischer" Ansatz mit flashrom (open Source)
- padden bis Chipgröße
- Bsp.: 131072 Byte - 85504 Byte = 45568 Byte Padding

Flashen (I)

- lspci -nn

Flashen (I)

- lspci -nn
- flashrom -p nicintel_spi:pci=07:01.0

Flashen (I)

- lspci -nn
- flashrom -p nicintel_spi:pci=07:01.0
- flashrom -p nicintel_spi:pci=07:01.0 -c "AT25F1024(A)" -r backup_"AT25F1024(A)".rom

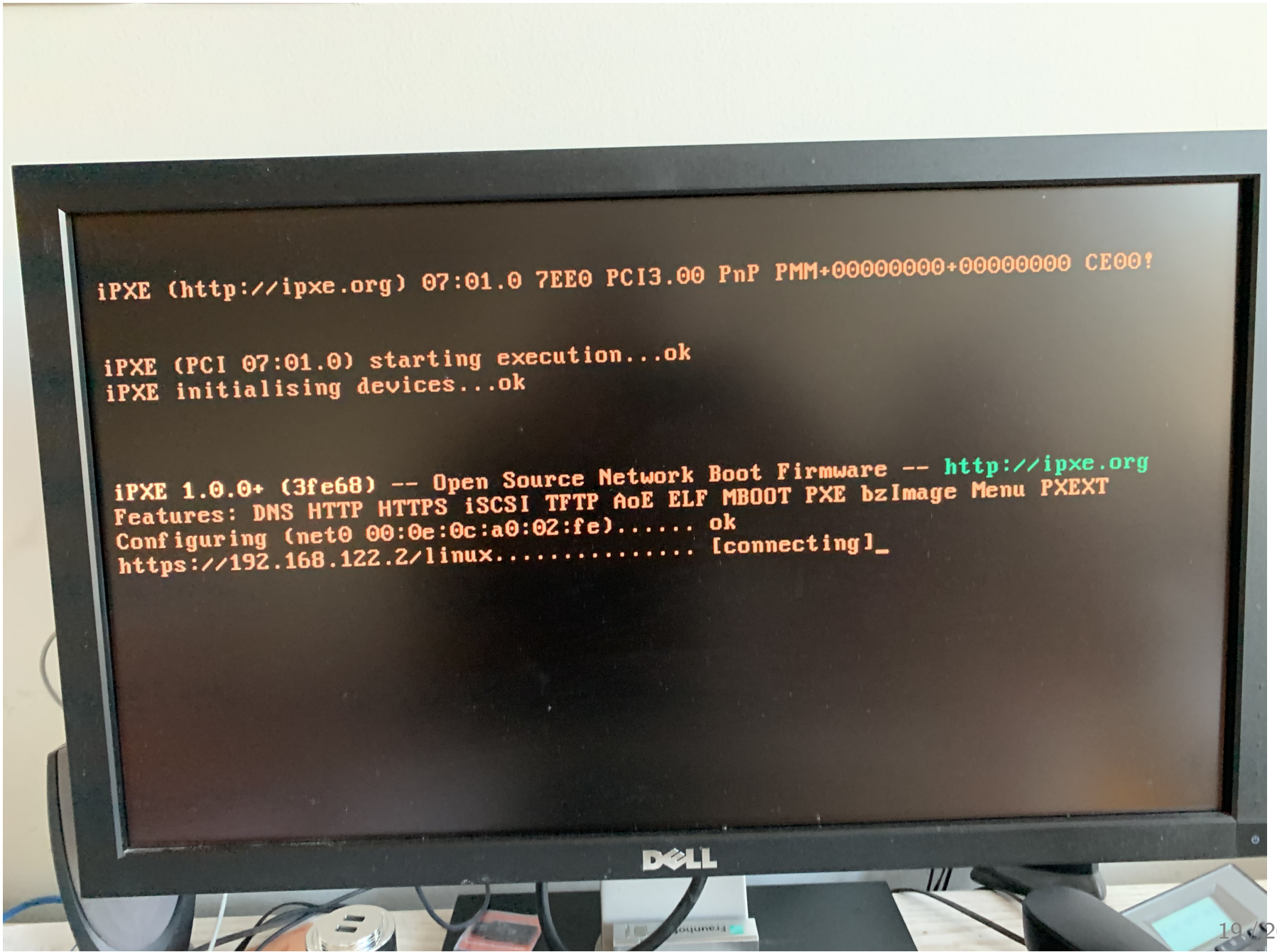
Flashen (I)

- lspci -nn
- flashrom -p nicintel_spi:pci=07:01.0
- flashrom -p nicintel_spi:pci=07:01.0 -c "AT25F1024(A)" -r backup_"AT25F1024(A)".rom
- flashrom -p nicintel_spi:pci=07:01.0 -c "AT25F1024(A)" -w ipxe.rom

Flashen (II)

```
flashrom v0.9.9-r1954 on Linux 4.19.0-1-grml-amd64 (x86_64)
flashrom is free software, get the source code at
https://flashrom.org

Calibrating delay loop... OK.
Found Atmel flash chip "AT25F1024(A)" (128 kB, SPI) on
nicintel_spi.
Reading old flash chip contents... done.
Erasing and writing flash chip... Erase/write done.
Verifying flash... VERIFIED.
```


iPXE (http://ipxe.org) 07:01.0 7EE0 PCI3.00 PnP PMM+00000000+00000000 CE00!
iPXE (PCI 07:01.0) starting execution...ok
iPXE initialising devices...ok
iPXE 1.0.0+ (3fe68) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP HTTPS iSCSI TFTP AoE ELF MBOOT PXE bzImage Menu PXEXT
Configuring (net0 00:0e:0c:a0:02:fe)..... ok
https://192.168.122.2/linux..... [connecting]_
DELL

Gibt es Fragen??