FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Forensik von DSL-Routern

Sebastian Braun, B.Sc.

Person

- Abschluss B.Sc. Ende Februar 2015

- Titel der Bachelorarbeit:

  „Systemanalyse von DSL-Routern als Grundlage einer forensischen Untersuchung“

  - Kooperation mit dem LKA NRW

Person

- Seit Mai 2015: Consultant für IT-Forensik, IT-Audit und eDiscovery bei Warth & Klein Grant Thornton

Gliederung

- Motivation
- Ziel
- Der DSL-Router
- Vorgehensweise
- Best Practice
- Fazit
- Ausblick

Motivation

- DSL-Router Anbindung an das Internet
- Nutzung des Internets in privaten Haushalten nimmt zu
    - 2013 80%
- Steigende Anzahl internetfähiger Geräte
    - PC, Smartphone, Tablet, Smart-TV...
- Moderne DSL-Router verarbeiten viele sensible Daten
    - Verbundene Geräte, DECT-Basisstation, VoIP-Funktion

Ziel

- Forensische Untersuchung des DSL-Routers
- Möglichkeiten zur Untersuchung und Sicherung des Systems mit Hilfe der gegebenen Komponenten
    - Was für Komponenten sind vorhanden?
    - Welche Möglichkeiten ergeben sich?
    - Was kann gesichert werden?
    - Wie kann es gesichert werden?

Der DSL-Router

- Eingebettetes System
  - Bestandteil eines technischen Systems
  - Funktionsumfang auf Nutzen reduziert
    - Betriebssystem mit eingeschränkten Funktionen
    - Mögliche Funktionen sind herstellerspezifisch
- Relevante Hardwarekomponenten
  - System-on-a-Chip
  - Arbeitsspeicher
  - Flash-Speicher

Der DSL-Router

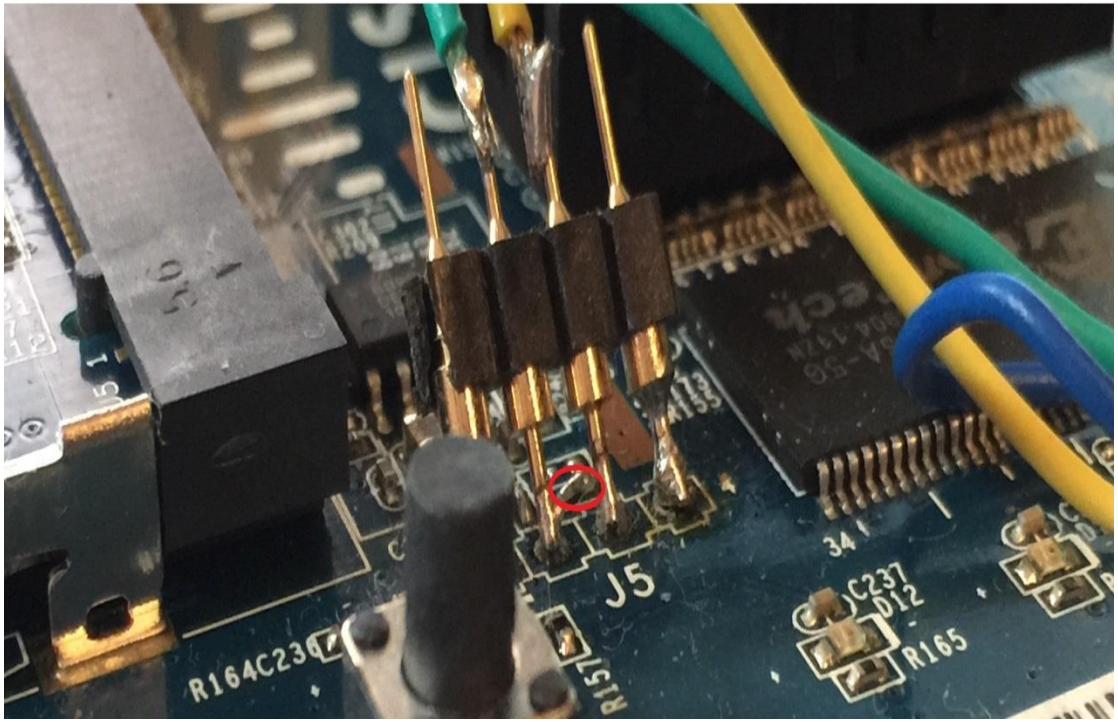
- Relevante Softwarekomponenten
  - Bootloader
  - Betriebssystem
- Kommunikationsmöglichkeiten
  - Telnet
  - UART-Schnittstelle

Vorgehensweise

- Vorbereitung der DSL-Router
- Forensische Möglichkeiten des Betriebssystems
- Forensische Möglichkeiten des Bootloaders
- Hardwarebasierte Möglichkeiten

Vorbereitung der DSL-Router

- Lokalisierung und Kontaktierung der UART-Schnittstelle
    - Muss teilweise erst aktiviert werden



Betriebssystemmöglichkeiten

- In der Regel Embedded Linux
- Untersuchung mittels Shell (falls vorhanden)
    - Login benötigt?
    - Aktivierung mittels „sh“-Befehl nötig?
    - Ohne Shell Untersuchung nicht möglich
- Informationserhalt aus bestimmten Dateien
    - /proc/mtd
    - /proc/meminfo
- Überprüfung vorhandener Befehle
- Sichern flüchtiger Systemdaten

Betriebssystemmöglichkeiten

```
# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00400000 00020000 "reserved-kernel"
mtd1: 03000000 00020000 "reserved-filesystem"
mtd2: 00400000 00020000 "kernel"
mtd3: 03000000 00020000 "filesystem"
mtd4: 00200000 00020000 "config"
mtd5: 01600000 00020000 "nand-filesystem"
mtd6: 00040000 00001000 "urlader"
mtd7: 00060000 00001000 "tffs (1)"
mtd8: 00060000 00001000 "tffs (2)"
```

```
# cat /proc/meminfo
MemTotal:         114976 kB
```

Unix-Tool dd

- Externes Speichermedium und Shell notwendig
  - Shell muss dd unterstützen
- Syntax: dd if=INPUT of=OUTPUT
- Erzeugt bitgenaue Kopie der Quelle und speichert diese am angegebenen Ort
- Arbeitsspeicher kann gesichert werden

```
# dd if=/dev/mtd0 of=/var/media/ftp/Kingston-DTRubber3-0-01/mtd0.dd
8192+0 records in
8192+0 records out
```

Bootloadermöglichkeiten

- Teilweise Userinterface zur Verfügung gestellt
  - Bootvorgang muss unterbrochen werden
  - Evtl. Passwort notwendig
  - Welche Befehle stehen zur Verfügung?
- Evtl. stellt das Bootloaderuserinterface einen Befehl zur byteweisen Ausgabe des Speichers auf der Konsole zur Verfügung
- Ausgabe muss geparst und in eine Datei geleitet werden

Bootloadermöglichkeiten

- Problem: Startadresse muss bekannt sein
- brntool.py automatisiert dies

```
[DANUBE Boot]:r

Enter the Start Address to Read....0xb0000000
Data Length is (1) 4 Bytes (2) 2 Bytes (3) 1 Byte...
Enter the Count to Read....(Maximun 10000)64

------------------------------------------------------------
 Address   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
------------------------------------------------------------
0xB0000000 10 00 01 3B 00 00 00 00 00 00 00 00 00 00 00 00
0xB0000010 68 8C 68 8C 00 00 00 00 31 2E 30 00 00 00 00 00
0xB0000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0xB0000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

[DANUBE Boot]:
```

```
CFE> dm bfc00070 32
bfc00070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
bfc00080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
bfc00560: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
bfc00570: 63 66 65 2d 76 01 00 25 66 09 00 00 00 00 00 00
```

Hardwaremöglichkeiten - JTAG

- Eigentlich Schnittstelle zum Testen von Hardware
  - z.B.: CPU
- Flash-Speicher ist mit der CPU verbunden
- Über CPU kann der Flash-Speicher ausgelesen werden
- Kenntnis über verwendete CPU notwendig
- Keine standardisierte Belegung

Hardwaremöglichkeiten - Chip-Off

- Entfernen des Flash-Speichers aus dem Router und Auslesen mittels spezieller Hard- und Software
- Flash-Speicher kann beim Vorgang zerstört werden
- Zusammensetzen der Daten aufgrund von Wear-Leveling problematisch

Best Practice

- Kontaktierung der UART-Schnittstelle
- Betriebssystemmöglichkeiten
- Bootloadermöglichkeiten
- JTAG
- Chip-Off

Fazit

- Viele verschiedene Hersteller und viele verschiedene Spezifikationen gestalten eine allgemeine Lösung als äußerst schwierig
    - Shell mit nötigen Befehlen vorhanden?
    - Shell geschützt?
    - Bootloaderuserinterface?
    - Firmwareupdate
- Shell häufig verfügbar, dd selten unterstützt
- Sicherung mittels Bootloaderinterface am häufigsten unterstützt, jedoch gehen flüchtige Daten verloren

Ausblick

- Sicherungen untersuchen und darin enthaltene Daten verfügbar machen
- Interpretation der Sicherungen
- Entwicklung eines Tools zur automatisierten Sicherung/Analyse

Ende

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?

Kontakt: sebastian.braun03@gmail.com