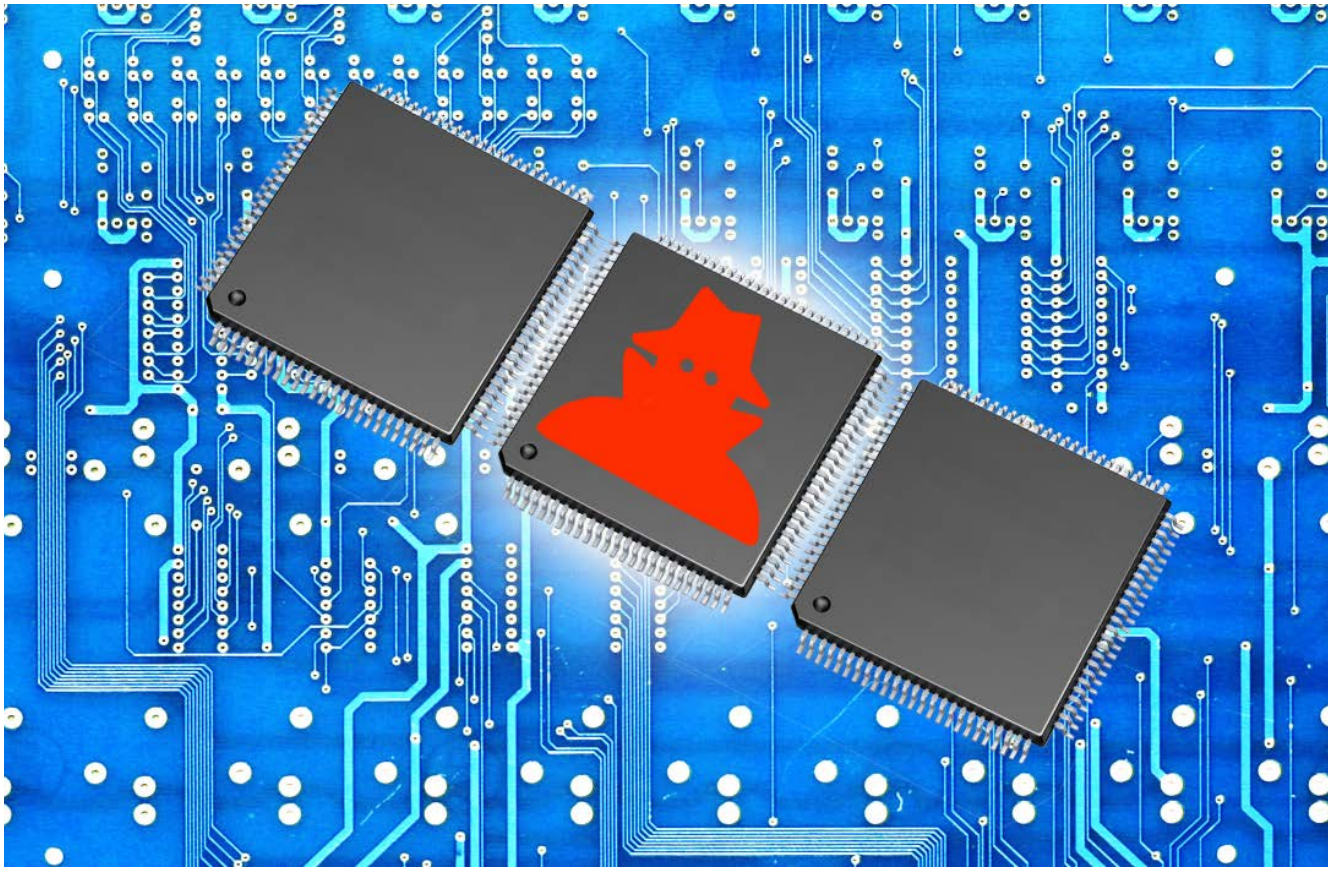


IIoT-Hacking: Speicher-ICs auslesen

Dumping Jack Flash

Alexander Poth

Auch Flashspeicher aller Art sind ein beliebtes Angriffsziel. Deshalb bilden sie Teil 2 der losen *iX*-Serie zum Thema IIoT-Hacking.

iX-TRACT

- Mit wenigen Werkzeugen, etwas Know-how und ein bisschen Recherche lassen sich auch fest verlötete Flash-Chips auf Embedded-Geräten auslesen.
- Auf ihnen befinden sich nicht nur sämtliche Systemdaten, sondern darin auch Hinweise auf Schwachstellen.
- Ist ein Dump eines Flash-Chips erst einmal erstellt, steht der Analyse nichts mehr im Wege. Die kann aber unterschiedlich kompliziert ausfallen.

Nachdem sich der erste Teil der neuen Artikelreihe zum IIoT-Hacking mit Angriffen auf IoT-Geräte beschäftigte, soll der zweite zeigen, mit welchen Mitteln sich Daten aus einem Speicherchip extrahieren lassen. Ein Flashmodul zu untersuchen beziehungsweise dessen Inhalt auszulesen, kann in Situationen notwendig sein, in denen die Firmware des Systems entweder nicht öffentlich verfügbar ist oder aus rechtlichen Gründen nicht bereitgestellt werden kann. Durch den Zugriff auf die Firmware erhält man einen umfassenden Einblick in die tatsächliche Funktionsweise des Geräts.

Für das Auslesen des Flash-Chips benötigt man lediglich einen Raspberry Pi oder ein alternatives Microboard sowie eine geeignete Messklemme für die Flash-Pins. In diesem Fall kommt die Messklemme Pomona SOIC Clip, Modell 5250, in der 8-Pin-Variante zum Einsatz.

Das Microboard muss über SPI-Pins (Serial Peripheral Interface) beziehungsweise über einen SPI-Bus verfügen und sich mit einem Linux bestücken lassen. Wer etwas Löterfahrung mitbringt, kann zur Not auf die Messklemme verzichten und sich über angelötete Leitungen direkt Zugang zu den IC-Pins verschaffen. Um Schäden am Flash oder gar am Board zu

vermeiden, empfiehlt sich dennoch die Nutzung einer solchen Klemme.

Zudem ist sicherzustellen, dass das Auslesen des Speichers nicht durch anhängende Komponenten beeinflusst wird, denn die notwendige Spannungsversorgung des IC muss das lesende Gerät, in diesem Fall der Raspberry Pi, übernehmen. Findet eine Beeinflussung statt, sollte man den Flashspeicher vor dem Auslesen von seiner Platine lösen.

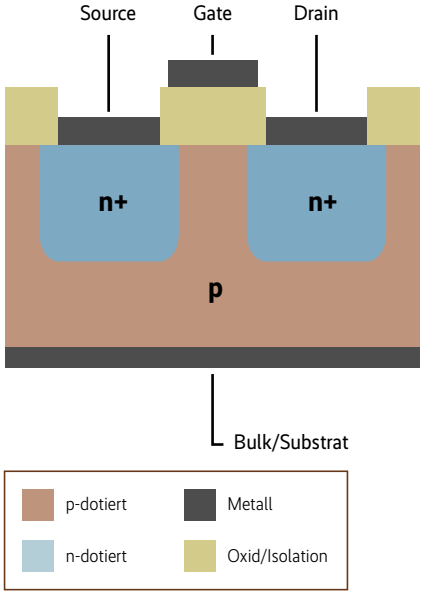
Zwar ist Flashspeicher heute in fast alles Geräten zu finden, dennoch gibt es Unterschiede. Die Flashspeicherzelle selbst unterscheidet sich kaum vom herkömmlichen FET (Feldeffekttransistor). Dieser verfügt über drei Anschlüsse. Die Source oder Quelle bildet den Zufluss, das Gate die Steuerelektrode und die Drain oder Senke den Abfluss (siehe Abbildung 1).

Source und Gate sind in den Arbeitsstromkreis eingebunden, quasi als Plus- und Minuspol. Liegt am Gate eine Spannung an, entsteht ein elektrisches Feld, das einen Elektronenfluss durch einen quantenmechanischen Tunnel von der Source zur Drain ermöglicht. Andernfalls können keine Ladungen respektive Elektronen von der Source zur Drain fließen. Verallgemeinert liegt also ein einfacher Schalter vor: Wenn Gate = 0, dann Drain = 0; wenn Gate = 1, dann Drain = 1.

Spannungen halten, Zellen verschalten

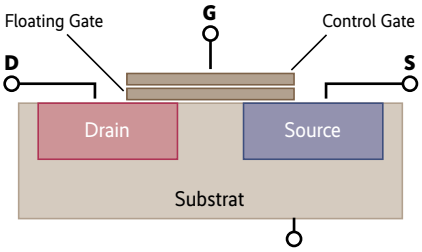
Soll ein Bit aber nichtflüchtig sein, muss eine elektrische Ladung am Gate dauerhaft anliegen. Deshalb benötigt man ein Element, das die Gate-Funktion speichert und den fließenden Tunnel erzeugt. Hier kommt das Floating Gate ins Spiel, das als Ladungsfalle dient, in der elektrische Ladung gespeichert bleibt (siehe Abbildung 2). Beim Schreiben und Löschen eines Bits, das die logischen Zustände NULL und EINS bildet, werden die Elektronen in das Floating Gate unter Spannung, der Schreibspannung, eingebracht und mit einer davon unabhängigen Spannung, der Löschspannung, wieder entnommen.

Allerdings kann man Flashspeicherzellen unterschiedlich verschalten. Bei Flash unterscheidet man generell zwischen NOR- und NAND-Flash. Die Unterschiede liegen vor allem in der Speicherdichte und den Zugriffszeiten.

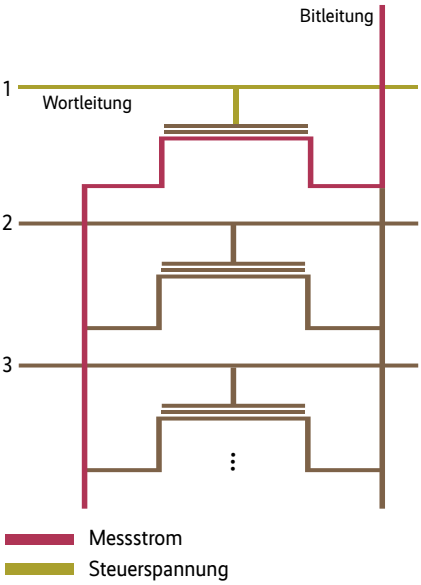


Jeder Feldtransistor verfügt über drei Anschlüsse: die Source für den Zufluss, das Gate als Steuerelektrode und die Drain als Abfluss (Abb. 1).

Bei NOR-Flash (NOT OR) werden die einzelnen Speicherzellen parallel verschaltet (siehe Abbildung 3). Der Zugriff auf die Speicherzellen erfolgt in diesem Fall wahlfrei und direkt, was schnelle Zugriffszeiten ermöglicht. NOR-Flash findet man etwa als Programmspeicher bei Mikro-



Das Floating Gate bildet eine Ladungsfalle, die elektrische Ladung hält (Abb. 2).

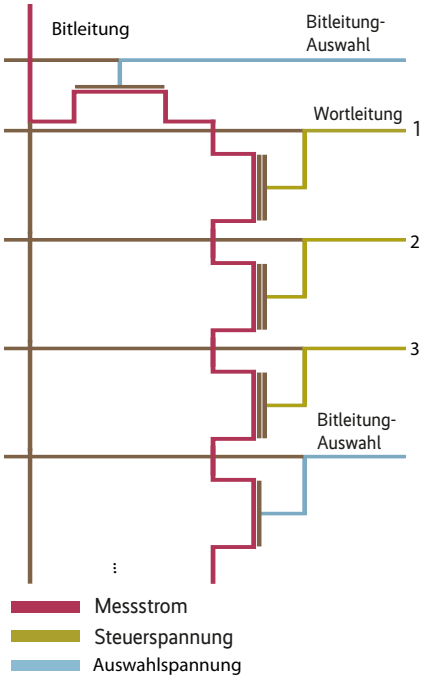


NOR-Flash-Zellen sind parallel verschaltet und linear adressierbar (Abb. 3).

controllern, da diese meist einen linear adressierbaren Speicher sowie wahlfreien Zugriff benötigen.

Anders hingegen sind bei NAND-Flash (NOT AND) die Speicherzellen in Serie geschaltet. Dadurch ist Lesen und Schreiben nur in Blöcken möglich (siehe Abbildung 4). Da bei den heute üblichen Datenträgern mit Dateisystem die Daten sowieso blockweise gelesen und geschrieben werden, eignet sich NAND-Flash für große Datenmengen. Da bei dieser Verschaltung mehrere Datenleitungen wegfallen, benötigen sie weniger Platz als bei NOR-Speichern. Die Datendichte ist dadurch höher als bei NOR-Speichern. Allerdings ist ein gewisser Softwareaufwand nötig, um NAND-Speicher richtig anzusteuern, der bei NOR-Flash entfällt.

In Embedded-Systemen sind Flashspeicher meist an den SPI-Bus angeschlossen. Das oft in Mikrocontrollern eingesetzte Serial Peripheral Interface überträgt die Daten seriell und kann etwa integrierte Schaltkreise nach dem Master-Slave-Prinzip anbinden. Der SPI-Bus verfügt über drei gemeinsame Leitungen, an die alle Teilnehmer angeschlossen sind:



NAND-Flash-Zellen sind in Serie verschaltet und nur blockweise beschreib- und lesbar (Abb. 4).

- MOSI (Master Output, Slave Input),
- MISO (Master Input, Slave Output),
- SCLK (Serial Clock), die zur Synchronisation mit dem Master dient.

Zudem gehören zum SPI-Bus eine oder mehrere Chip-Select-Leitungen namens SS, CS oder CE, die der Master steuert. Für jeden Slave ist eine eigene solche Leitung vorgesehen (siehe Abbildung 5). SPI ist vollduplexfähig und erlaubt variable Taktraten bis in den MHz-Bereich.

Auf dem OP-Tisch

Für den Versuch muss die Platine des DSL-Routers TP-Link TL-WR841N aus dem vorherigen Artikel herhalten. Dort verrät ein relativ kleiner Chip Hersteller und Modellbezeichnung. Da die Aufdrucke meist schwer zu erkennen sind, bietet sich zur Identifizierung derartiger Komponenten ein Digitalmikroskop an (siehe Abbildung 6).

Eine kurze Suche im Internet nach dem Begriff „25Q32CSIG Datasheet“ führt zum passenden Datenblatt mit gleichem Herstellerlogo und allen wichtigen Informationen zum Chip. Dem Datenblatt lässt sich entnehmen, dass es sich um einen Flash-Chip der Firma GigaDevice mit einem Takt von 140 MHz und einer Speicherkapazität von 4 MByte handelt, der über den SPI-Bus angesteuert wird. Zudem verrät das Datenblatt in einem Connection Diagram die einzelnen Pinbelegungen (siehe Abbildung 7). Laut Datenblatt verfügt der Chip über zusätzliche Modi wie Dual- oder Quad-SPI. Solche erweiterten Funktionen sollen etwa das Schreiben von Daten beschleunigen, benötigen aber mehr als die bekannten Standardanschlüsse. Zum Auslesen des Chips sind lediglich die oben beschriebenen Anschlüsse für die Standard-SPI-Funktion notwendig, außerdem die Pins für die Stromversorgung:

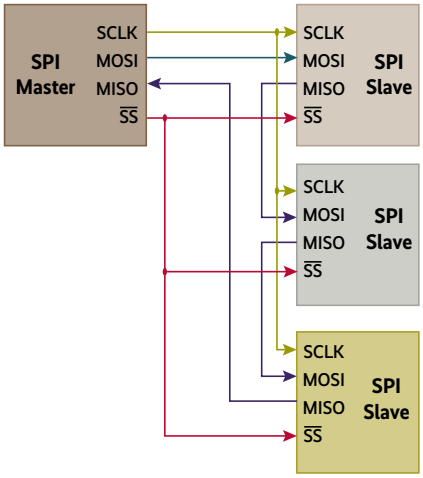
- CS#: Auf Pin 1 in der mit dem Orientierungspunkt markierten Ecke liegt die Chip-Select-Leitung, die oft auch mit SS oder CE beschriftet ist.
- SO: Der Slave-Output auf Pin 2 bildet die Schnittstelle zur MISO-Leitung.
- VSS: Die negative Versorgungsspannung auf Pin 4 ist identisch mit GND, dem Masseanschluss.
- SI: Der Slave-Input, Pin 5, entspricht dem MOSI-Pin.
- SCLK: Auf Pin 6 sitzt die Serial Clock.
- VCC: Pin 8 bildet den Anschluss für die positive Versorgungsspannung.

Nun fehlt noch das Pin-out-Diagramm für den Mikrocomputer, in diesem Fall ein Raspberry Pi 3 B+. Meist stellen die Hersteller bereits umfangreiche Dokumentationen zu Pinbelegungen auf ihrer Homepage zur Verfügung (siehe Abbildung 8).

Alles bereit für den Dumping-Prozess

Beim Verbinden der Pins auf dem Raspberry Pi mit der Messklemme ist auf die richtige Anordnung zu achten (siehe Tabelle „Verbindung der Pins“).

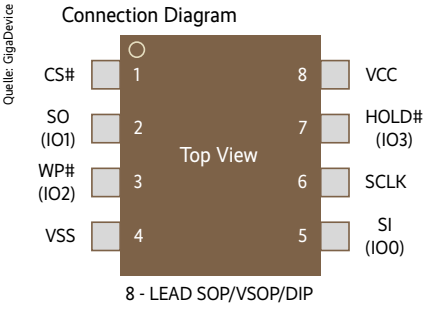
Anschließend ist die Messklemme nur noch auf den Flashspeicher zu setzen. Zur



Zu den drei gemeinsamen Leitungen MOSI, MISO und SCLK gesellt sich für jeden Slave eine eigene Chip-Select-Leitung (Abb. 5).

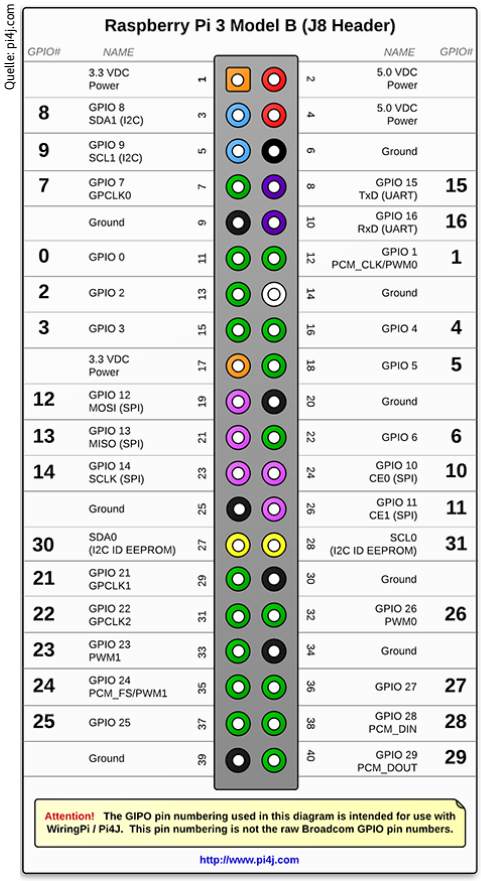
Unumgänglich, aber nicht immer einfach ist das Entziffern des Aufdrucks auf dem Chip (Abb. 6).

Quelle: GigaDevice



Ein Connection Diagram im Datenblatt des Flash-Chips GigaDevice 25Q32CSIG gibt die Pinbelegungen preis (Abb. 7).

Quelle: pi4j.com



Die umfangreiche Pinbelegung des Raspberry Pi ist im Internet schnell gefunden (Abb. 8).

korrekten Ausrichtung orientiert man sich an dem auf dem Flashspeicher eingebetteten Punkt, der auch im Connection Diagram Pin 1 markiert (siehe Abbildung 9).

Sind alle Kabel verbunden, startet man den Raspberry Pi. Dort öffnet man die Konfiguration mit dem Befehl `raspi-config` in einem Terminal. Um die benötigten Pins des Raspberry Pi überhaupt nutzen zu können, ist zunächst der SPI-Bus zu aktivieren (siehe Abbildung 10).

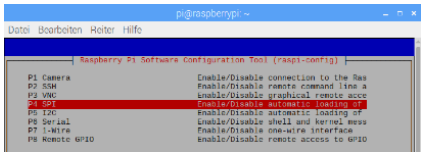
Zudem muss der Mikrocomputer einen Internetzugang haben. Mit dem Befehl `sudo apt-get install flashrom` installiert man die zum Auslesen des Flash-Chips nötige Software flashrom. Sie kann Flash-Chips identifizieren, auslesen und beschreiben und erkennt mittlerweile fast 500 Flash-ICs. Zunächst gilt es, den Chip zu identifizieren (siehe Listing 1).

Den Linux-eigenen SPI-Treiber liefert das Kernelmodul linux_spi. Der Parameter `dev=/dev/spidev0.0` steuert den Zugriff auf den SPI-Bus des Raspberry Pi als `spidev0.0`. Die zu verwendende SPI-Bus-Geschwindigkeit gibt der Parameter

Sind die Kabel den Pins auf dem Raspberry Pi richtig zugeordnet und ist auf dem Flash-Chip Pin 1 identifiziert, muss nur noch die Messklemme auf den Chip gesetzt werden (Abb. 9).

Verbindung der Pins

Pins Raspberry Pi	Pins Flashspeicher
3.3V	VCC
Ground	VSS
MOSI	SI
MISO	SO
SCLK	SCLK
CE0	CS#



Mit dem Raspberry Pi Software Configuration Tool raspi-conf lässt sich der SPI-Bus aktivieren (Abb. 10).

`spispeed=8000` in kHz an. Listing 1 zeigt auch, dass flashrom den Chip erfolgreich mit seiner Größe von 4 MByte identifiziert.

Chip-Inhalt lesen und auslesen

Der nächste Befehl soll den kompletten Inhalt des Chips auslesen und in einer Dump-Datei speichern. Hierzu erweitert man lediglich den vorherigen Befehl um den Read-Parameter `-r` und die Angabe der Zieldatei dump.bin (siehe Listing 2). Danach sollte sich im aktuellen Verzeichnis die 4 MByte große Binärdatei dump.bin befinden.

Aus solcherart generierten Firmware-Images lassen sich unter Umständen sehr sensible Informationen auslesen. Ist die Firmware weder vollständig verschlüsselt noch komprimiert, kann man gegebenenfalls Details mit dem Kommando `strings` extrahieren, das nach für den Menschen lesbaren Zeichen innerhalb einer Datei sucht.

Das Beispiel kombiniert `strings` mit `grep`. Da Firmware-Dateien meist einen Bootloader beinhalten, beginnt hier die Suche nach Hinweisen (siehe Listing 3). Damit gelingt es bereits, den U-Bootloader, Version 1.1.3, zu identifizieren.

Von großem Interesse sind zudem gespeicherte Keys oder Passwörter: Das Filtern nach Begriffen wie „Sharedkey“ oder einfach nur „Key“ ergibt eine achtstellige Zahlenkombination, die dem aufgedruckten WPS-Key auf der Rückseite des Routers entspricht (siehe Listing 4). Ein Angreifer wäre also tatsächlich in der Lage, aus einer kopierten Firmware Passwörter und weitere sensible Daten zu extrahieren und diese für sich zu nutzen.

Allerdings sind die Möglichkeiten zum Datenauslesen mit solchen Standardtools schnell erschöpft. Ein Grund ist etwa die Komprimierung von Dateisystemen auf solchen Embedded-Geräten. Deshalb soll der folgende Artikel im Detail zeigen, wie sich Kernel- und Dateisysteme einer „Embedded Firmware“ mit klassischen Linux-Werkzeugen vollständig extrahieren lassen.

Listing 1: Flash-Chip identifizieren

```
pi@raspberrypi:~ $ flashrom -p linux_spi:dev=/dev/spidev0.0,spispeed=8000
flashrom v0.9.9-r1954 on Linux 4.19.42-v7+ (armv7l)
flashrom is free software, get the source code at https://flashrom.org

Calibrating delay loop... OK.
Found GigaDevice flash chip "GD25Q32(B)" (4096 kB, SPI) on linux_spi.
No operations were specified.
```

Listing 2: Flash-Chip auslesen

```
pi@raspberrypi:~ $ flashrom -p linux_spi:dev=/dev/spidev0.0,spispeed=8000 -r dump.bin

flashrom v0.9.9-r1954 on Linux 4.19.42-v7+ (armv7l)
flashrom is free software, get the source code at https://flashrom.org

Calibrating delay loop... OK.
Found GigaDevice flash chip "GD25Q32(B)" (4096 kB, SPI) on linux_spi.
Reading flash... done.
```

Listing 3: Suche nach dem Begriff „Boot“ im Dump

```
root@kali:~/Desktop/TP-LINK-Experimente# strings dump.bin | grep -i boot
bootcmd=tftp
bootdelay=1
eU-Boot 1.1.3 (Mar 19 2018 - 15:36:42)
    %d: Boot system code via Flash (default)
    %d: Load Boot Loader code then write to Flash via TFTP
            Input Uboot filename
uboot.bin
```

Listing 4: Suche nach dem Begriff „Sharedkey“ im Dump

```
root@kali:~/Desktop/TP-LINK-Experimente# strings dump.bin | grep -i sharedkey
          <X_TP_PreSharedKey val=13810078 />
          <X_TP_PreSharedKey val=13810078 />
```

Ausblick

Nach etwas Vorarbeit auch mit den technischen Unterlagen ließ sich mit dem Werkzeug flashrom der Beispiel-Flash-Chip erfolgreich auslesen. Der extrahierte Speicherinhalt enthält in der Regel den gesamten Code, den das System ausführt. Dazu zählen auch, je nach Art des Geräts, der Bootloader, der Kernel sowie das Root-Filesystem, das hin und wieder Schwachstellen wie fest eingeschriebene Zugangsdaten, schwache Konfigurationsdateien und sonstige vertrauliche Daten enthält.

Um den Flash-Dump der Binärdatei weiter zu analysieren und relevante Dateien zu extrahieren, benötigt man allerdings spezielle Linux-Tools, die Gegenstand des nächsten Artikels sein werden. Sie decken unter anderem eine eventuelle Command Injection auf, die einem Angreifer die vollständige Kontrolle über den Router ermöglicht. (sun@ix.de)

Alexander Poth

ist in der IT-Security-Branche tätig. Zu seinen Schwerpunkten zählt die Firmware-Schwachstellenanalyse.