6R-04

家庭向けネットワーク機器のホワイトボックス化の提案

藤島　久麿[†1]　　寺澤　卓也[†2]
東京工科大学メディア学部メディア学科[†1†2]

1. はじめに

昨今、いわゆる家庭用 IoT 機器のセキュリティーに関する話題が注目を集めている。

本研究では、特に IoT 機器のソフトウェア管理について注目した。常時接続時代となり、これらの IoT 機器が常にインターネットにアクセスできることを前提として、ソフトウェアの管理をメーカーのクラウド上で行い、機器のハードウェアが起動するたびにソフトウェアをダウンロードし起動するというシステムの検討を行った。こうすることで、クラウド上のソフトウェアを変更することによって機器の用途を変更できる。今回、Raspberry Pi を使用してプロトタイプを作成し、Web サーバー上でソフトウェアイメージを切り替えることにより異なる機器として使用できるか実験を行ったので報告する。

2. クラウド化ソフトウェアと汎用ハードウェアの可能性

2.1　ソフトウェアのクラウド化

IoT 機器のセキュリティー問題として、脆弱性が発見れた時、運用されている機器の問題をどのように修正するかという問題がある。この問題は、２つが考えられる。１つは、メーカーが修正情報を出したにも関わらず、ユーザーの設定知識や認識の低さから、ソフトウェアの更新が行われないという問題である。もうひとつは、メーカーが既に保守を行っていない機器で脆弱性が発見された場合、修正情報がない状態でも IoT 機器が使用され続けてしまうことである。家庭向けの専用機器が IoT 化によって次々と、インターネットに接続される一方、PC 用 OS のようにソフトウェアの自動更新の方法が確立されている機器はまだまだ少ない。

本研究では、このような IoT 機器のソフトウェア部分をライセンス管理とし、クラウド上に保存する方法を検討した。

IoT 機器には、クラウド上に存在するソフトウェアをダウンロードし起動する機能のみを保存し、ハードウェアが再起動するたびにクラウド上からソフトウェアをダウンロードしはじめて IoT 機器として使用することができるようにする。そして、機器は一定の頻度で再起動される仕掛けとする。このようにすれば、ユーザーは機器のソフトウェアの情報更新を意識する必要なく、ソフトウェアの保守が行われなくなった機器はソフトウェアがダウンロードできなくなるため使用できなくなる。

2.2　ハードウェアのホワイトボックス化

現在多くの家庭向けネットワーク機器はコンピュータの小型化、低価格化により単純な機能のみを提供する機器であっても、専用のハードウェアを用意せず、汎用的な小型コンピュータに汎用的な OS を前提として製品の開発ができるようになりつつある。汎用的な OS によってハードウェアが抽象化されているということは、現在の PC やスマートフォンのように、用途によってソフトウェアを交換する、もしくはさらに周辺機器を追加することによって、さまざまな用途で使用できるように汎用的なハードウェアを販売し、使用用途によってソフトウェアを販売するということを実現できないか考え、このような機器として利用できる汎用的なハードウェアのことをハードウェアのホワイトボックス化と称することとした。

2.3　クラウド化ソフトウェアと汎用ハードウェアの可能性

ソフトウェアをライセンス管理とし、クラウド上に保管することだけを考えれば、保守は確実となるかもしれないが、メーカーがソフトウェアの保守管理を行わなくなった機器はハードウェアも使用できなくなってしまう。そこで、ハードウェアもホワイトボックス化しほかの用途に転用できるような機器として考えれば、同様の機能としても別の用途としても、ソフトウェアを切り替えて使用することができる。

A Proposal of 'white-boxing' for consumer network equipments
†1 Hisamaro Fujisima, †2 Takuya Terasawa
†1†2 School of Media Science,Tokyo University of Technology

3. 家庭向けネットワーク機器のプロトタイプの実装

3.1 本研究での範囲

本研究ではRaspberry Pi[1]をホワイトボックス化したIoT機器と見立て、ソフトウェアがクラウド化された機器を作成し、実験環境の中で評価を行い、このような提案が実現可能かどうかを探った。

3.2 条件

プロトタイプを作成するにあたって、ネットワーク機器は、家庭向けを想定し、ローカルネットワークの中に別途機器の設置すること、ネットワーク自体に仕掛けをするとはできないものとした。

すなわち、DHCP サーバーに設定が必要な、PXE boot[2]のような仕掛けは使用しないものとする。

3.3 実装方法

Linux のブートローダーとして一般的に使われる高機能ブートローダーの１つに、オープンソースのソフトウェアDas U-Boot[3]がある。U-Bootは様々なコンピュータに対応するブートローダーで Raspberry Pi もサポートしている。U-Bootには TFTP プロトコルを使用してファイルをメモリにダウンロードできる機能がある。この機能は IP とファイル名を指定して、ファイルをダウンロードでき、U-Boot が認識できるファイルに変換しておけば、ダウンロードしたメモリのアドレスを指定するだけで、アプリケーションの実行が可能である。

クラウド上に、TFTP サーバーを用意した。さらに、U-Boot にダウンロードされるファイルがWEBブラウザから変更できるようにHTTPサーバーを用意し、WEB アプリケーションを作成した。それぞれ FTP サーバーとして htpd-hpa[4]、HTTP サーバーとしてApache httpd[5]を使用した。

4. 評価

以上の方法でネットワークブートを行ったところ、U-Boot から、TFTP を使って、メモリ上にアプリケーションをダウンロードすることができた。U-Boot 上で動作する、１ファイル構成の簡易なアプリケーションの動作は確認できたが、Lunix カーネルの起動はできなかった。これが、U-Bootの設定の問題であるのか、Rasberry Piの仕様によるもので、実行しようとしているカーネルが正しくないのかは現在調査中である。

5. まとめ

今回、Raspberry Pi を使って、IoT機器のプロトタイプの作成を試みたが、Linux カーネルのブートはできなかった。U-Boot は月１度ほどのペースで最新版がリリースされ、バージョンによって動作が不安定なことも多いことから、他のハードウェアとの相性や、使用するバージョンの検討する必要があるだろう。

汎用 OS が使用できるということは、ソフトウェア開発の低コスト化が実現でき、汎用ハードウェアを使って機器を開発する大きな理由になりえる。実用化においては汎用 OS のブートは必要不可欠である。

本研究の目的である、ハードウェアのホワイトボックス化とソフトウェアのクラウド化によるIoT機器のセキュリティーと機能向上にはさらなる調査と実験が必要である。

6. 今後の課題

残る課題として、TFTPはセキュアでないUDPプロトコルであるため、SFTP などセキュアな通信が可能なプロトコルが使用可能なブートローダーの検討が必要である。また、U-BootではDNSを使った名前解決が行えないため、IP を直接指定する必要があるが、実用を考えれば、DNS での名前解決も必要である。停電時やネットワーク障害時の対策の検討も必要である。

参考文献

[1] Raspberry Pi,
https://www.raspberrypi.org/,
(2017.07.31 閲覧)

[2] PXE boot,
http://www.pix.net/software/pxeboot/,
(2017.07.31 閲覧)

[3] Das U-Boot Wiki,
https://www.denx.de/wiki/U-Boot,
(2017.10.14 閲覧)

[4] TFTP Official Ubuntu Documentation
https://help.ubuntu.com/community/TFTP
(2018.1.12 閲覧)

[5] The Apache HTTP Server Project
https://httpd.apache.org/
(2018.1.12 閲覧)