Vol.2013-CDS-7 No.8
2013/5/24

HTML5 から ECHONET Lite ネットワークへの簡易認証つきクロスドメインアクセス方式の提案

大和田　茂[†1]　　徳久　文彬[†2]

HTML5 で実装された Web サービスから宅内の ECHONET Lite ベースのホームサーバに簡易的な認証つきでクロスドメインアクセスするための一手法を提案する.この方式は通信を仲介する iFrame（サーバーパネルと呼ぶ）への PostMessage によって実装されたフレームワークであり,家電プロトコルの詳細を知らない Web アプリ開発者が容易に家電アプリを開発できるようにする一方,操作の危険性に応じて情報の流れをホームサーバの側から遮断できる機能を持っている.基本的なアイデアは,iFrame で実装されるサーバーパネルを Web サービスとは別ドメインに配置し,ここを仲介せずにホームサーバにアクセスすることを禁じること,危険性の高い操作については逐一ユーザーに問い合わせること,および通信路の暗号化である.

キーワード: ホームネットワーク, クロスドメインアクセス, 認証,HTML5, ECHONET Lite

HTML5 Cross Domain Access Method for ECHONET Lite-based Home Server with Simple Authentication Mechanism

SHIGERU OWADA[†1]　FUMIAKI TOKUHISA[†2]

We propose a framework to access ECHONET Lite-based home network from untrusted HTML5 web application with a simple authentication mechanism. Our system is based on message passing using PostMessage between third party app and trusted iFrame (called a server panel.) It allows web app programmers to easily access home network while suspicious operations can be blocked by user's manipulation. Basic ideas are three-fold: 1. The server panel is located on different domain from third party app to capture all access to home network, 2. Dubious operations can be blocked by user's intension, and 3. Encrypted WebSocket connection.

Keywords: Home network, cross-domain access, authentication, HTML5, ECHONET Lite

1. はじめに

近年のスマートハウス市場の盛り上がりにより,ECHONET Lite[1]に代表されるような宅内家電ネットワークインフラが急速に整備されてきており,様々なアプリが生まれる下地が整ってきている.しかし,これらはまだ登場して間もなく,解決すべきいくつもの問題が存在している.特に,アプリケーションの多くがブラウザ内で動作する Web アプリとして実装されつつある現状にあって,信頼できない Web アプリからの宅内ネットワークへのアクセスをどのように許していくかは大きな問題である.

Web アプリを実行したユーザーに,本人が意図しないアクセスを発行させるという攻撃手法は CSRF または XSRF（クロスサイト・リクエストフォージェリ）と呼ばれており宅内ネットに限った話ではないが,家電が勝手に動作させられるということは現実の住環境を破壊する可能性があるという意味でこれまでにない深刻さを持った問題であると言える.

†1 株式会社ソニーコンピュータサイエンス研究所
Sony Computer Science Laboratories, Inc.
†2 東京工業大学
Tokyo Institute of Technology

この問題に対する各 HEMS メーカーの通常のアプローチは,宅外から宅内にアクセスできるアプリを自社製のもののみに制限し,その範囲内においてのみセキュリティを確保する方法,すなわち囲い込みである.このアプローチであれば,情報が流れる部分のほとんどすべてを一社が管理することができ,責任の所在もはっきりするため会社としてはアプローチが容易である一方,アプリのバリエーションはその会社の開発力の範囲に制約される.大まかな傾向として,HEMS システムを製造するメーカーが Web アプリの開発にも長けていることはまれであり,ここが消費者の関心を引くアプリが出現しにくい一因になっている.

このような現状を打開するため,我々は信頼できるとは限らない Web 上のサードパーティアプリから JavaScript ベースの API を用いて容易に宅内ネットワークへアクセスすることを許しつつも,同時にその挙動を制約し,ある程度の安全性を確保する方式の研究開発を行っている.本稿で提案するフレームワークの基本的なアイデアは以下の 3 点である.

1. 信頼できない Web アプリからホームサーバへの直接

アクセスは禁じるが,信頼できる特定のサーバを経由したアクセスは許可する.この特定のサーバに置かれた html モジュールはサードパーティアプリと同一画面内に iFrame として配置される.通常の web ブラウザであれば,異なるドメイン間の通信は強く制約されており,この場合 iFrame との通信手段は,PostMessage を用いたものに限られる.

2. 危険と思われる操作をホームサーバが検出した際には,そのアクセスを許すかどうかを iFrame 内からユーザーに逐一問い合わせる.これにより,家電を目の前にした人間が操作するのと同程度の安全性を確保できる

3. 通信路を暗号化する.通信路の暗号化はすでに広く使われている技術であるためそれを用いる.

我々はこれらを,Kadecot[2]と呼ばれる Android 上で動作するホームサーバ上で実装した,サーバーパネルとホームサーバの間は WebSocket を用いて通信する.Android における SSL 通信は OS のバージョンによって制約があり,現時点では通信路の暗号化については一部サポートされないブラウザが存在している.

2. 提案システム

図 1 システムの構成
Figure 1 System architecture

本フレームワークは,HTML5 で実装されるサードパーティアプリと iFrame として実装されるサーバーパネル,および Android で実装されるホームサーバの 3 点からなる.

サードパーティアプリはサーバーパネルとの間で PostMessage のやりとりを行うことによって間接的にホームサーバにアクセスする.サーバーパネルはサードパーティアプリとは別ドメインにあるため,サードパーティアプリから直接その内部を覗くことは原則できない.

サーバーパネルは WebSocket を使ってホームサーバにアクセスする.この通信路は暗号化される.

ホームサーバは,サーバーパネル以外からの接続は許さないことにする.この遮断のための接続元情報は WebSocket のハンドシェイク時に流れる origin ヘッダにより特定することとする.origin ヘッダは偽装が可能だが,通常のブラウザであれば適切に設定されているので実用上問題はないものと考えられる.これにより,サードパーティアプリからの直接のホームサーバアクセスが禁じられる.

我々の実装で用いている Kadecot と呼ばれるホームサーバにはフリーの ECHONET Lite 実装である OpenECHO[3]が含まれている.ECHONET Lite とは 2011 年に仕様が公開された比較的新しいプロトコルで,多数の機器やセンサがサポートされている.スマートハウス市場の盛りあがりに合わせ,消費者が実際に購入可能な対応家電も次々と出てきている.このプロトコルに基いて,ホームサーバは機器の自動発見や情報取得・設定を行う.

2.1 ユーザーによるアクセス遮断

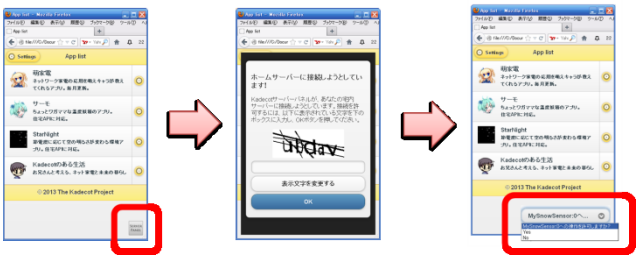

図 2 サーバーパネルの三形態
Figure 2 Three modes of the server panel

サーバーパネルは必要に応じて 3 通りの形態をとる,図 2 左の最小化状態の時は単なるボタンになっている.これを押せば,宅内機器の情報を閲覧したり,多少のリモコン操作を行ったり,ホームサーバの設定を行うことなどが可能である(図 3).

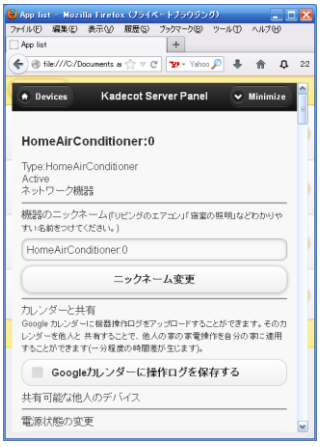

図 3 サーバーパネルでの機器設定画面の例
Figure 3 Device configuration panel in the server panel

サーバーパネルが WebSocket によってホームサーバに接続すると,ホームサーバは Captcha 画像を生成してサーバーパネルに送る.サーバーパネルは全画面形態に移行し,Captcha 内の文字列をユーザーに入力させるダイアログが出現する(図 2 中).この文字列をホームサーバ側で認証し,正しいとき

のみそれ以降の情報や操作性の提供を行うこととする.これは,不正なブラウザアドオンなどがインストールされたブラウザからアクセスされたときには,iFrame の中であっても不正操作がなされる可能性があり,そういった時にもプログラムから勝手に宅内家電にアクセスされるのを防ぐためである.
Captcha 認証によって人間が操作していることが確認できればあとはその接続を用いて情報提供を行ったり,機器操作を許したりする.ただし,その操作の中にはサードパーティアプリの制作者が意図するかしないかにかかわらず,危険をもたらすような操作も含まれる可能性がある.そのような操作をホームサーバが検知すると,それをユーザーにフィードバックし,許可されたときのみ当該操作を許すことにする.これもサーバーパネルが選択肢を提示し,ユーザーが操作を許可したときのみ実際の家電にコマンドが送られるようにする.これが図 2 右の状態で,ユーザーが操作の許可を選択しない限り実際の機器操作は起こらない.

2.2 通信路の暗号化について

WebSocket の暗号化は wss スキームを使い SSL を用いて行うのが一般的である.ただし,ホームサーバのドメイン名は確定できないので,認証局発行の証明書を用いることはできず自作の証明書を用いる必要がある.そのため通常のブラウザであれば最初に不審な証明書である旨の警告が出て,ユーザーがそれを承認する必要がある.

我々のシステムではこれを実装しているが,Android4.1 未満の端末では,OS 標準ソケットが TLS1.0 にしか対応していないため,それ以上のプロトコルを要求するクライアントとは通信ができない.我々の試した限りでは,最新の Internet Explorer と Firefox では暗号化通信可能であったが Chrome では不可能であった.

自作証明書による警告の表示と TLS のバージョン問題を嫌い,JavaScript の暗号化ライブラリを用いた自前暗号化も試してみたが,暗号化処理には大変大きな計算資源を必要とするところから実用に耐えるものはなかった.

ただし,これらの問題はサーバを Android としているために生じている制約であり本質的なものではない.事実 Linux での通信実験では何も問題は起こらなかった.

2.3 ECHONET Lite へのアクセス方式

前述のように,我々が実験に用いている Kadecot ホームサーバでは,フリーソフトである OpenECHO を用いて ECHONET Lite ネットワークにアクセスすることができる.ECHONET Lite では自動機器発見や状態変更通知なども行うことができるので,提案システムがサードパーティアプリに提供しているインターフェースでもそのような機能が扱えるようになっている.

具体的には,サードパーティアプリは kServerPanel.js というファイルを script タグで読みこむ.このファイルは現在の実装では 165 行しかない小さなファイルである.これを読みこむと kServerPanel というグローバルオブジェクトができるので,この中の初期化メソッドを呼び出すと,自動的に iFrame が生成される.

kServerPanel にはいくつかのメソッドが定義されているが,サードパーティアプリが触る必要があるのは初期化メソッドを除けば invoke(), onDeviceFound(), onPropertyChanged() の 3 つである.
invoke()は(ホームサーバ上の機能を表す)関数名,引数,コールバック関数の 3 つの引数を持つもので,代表的な関数名には”list”と”access”がある.これらはそれぞれ,機器一覧を取得するものと,個別の機器プロパティ(家電装置の内部変数)の値の取得や設定を行う関数である.繰り返し述べているように,このアクセスはホームサーバ側で危険性がないかどうかチェックされ,危険だと判断されればサーバーパネル内でユーザーに問いあわせがなされ,ユーザーが許可しない限りはサードパーティアプリにはエラー通知が返される.
onDeviceFound()と onPropertyChanged()は,ホームサーバ側がサードパーティアプリ側に情報を通知したい時に使われるもので,サードパーティアプリはこのメソッドを自分の望む処理を行うメソッドで上書きしておくことでサーバからの通知を受けとることができる.それぞれ,新たな機器が発見された時,機器の状態変更がアナウンスされた時に呼び出される.機器の状態変更については,ECHONET Lite の仕様書にて通知をしなければならない場合が規定されている.

2.4 ECHONET Lite 以外の機器

Kadecot ホームサーバでは,ECHONET Lite 対応機器だけでなく,他にも赤外線リモコンでコントロールする機器や DLNA で通信する機器なども制御することができる.赤外線の場合は,iRemocon[4]という外部機器と連携して既存の家電を動作させることができる.このパターンの場合,Kadecot は赤外線動作する機器をあたかも ECHONET Lite 機器であるかのようにネットワークに参加させ,ECHONET Lite プロトコルでコントロールする.従って,ECHONET Lite ネットワーク上に他のコントローラが存在したとすると,そのコントローラからは純粋な

ECHONET Lite 機器として操作することが可能である.
DLNA の場合は,ECHONET Lite では規定されていないので独自の制御方式となる.
また,Web 上の天気 API[5]などとも連携し,温度センサや湿度センサを ECHONET Lite センサとしてネットワークに参加させる機能も有している.

3. 関連システム

家電ネットワークの WebAPI を提供するパイオニアは大和ハウス工業株式会社による住宅 API[6]である.これは ECHONET/ECHONET Lite にとどまらず,接点装置やその他様々な機器を HTTP ベースで制御可能になっている.この API は宅内にある JavaScript アプリからのアクセスを前提としており,我々のようにサードパーティによるオンラインアプリからの直接アクセスを想定していないため,セキュリティの問題が少なく,クロスドメインアクセスはサポートされていない.

ブラウザ自体の機能を向上させることで,直接家電を操作しようという動きもある.例えば家電向けに開発されたブラウザである NetFront Browser[7] では,直接 DLNA を扱うことができる.また,Chrome ブラウザ拡張である Chrome Packeged Apps[8]を使うと,ブラウザから直接 uPnP/DLNA を実装することができる.このような流れは現在盛んに議論されており,将来様々な仕組みで家電を扱えるようになると考えられる.

4. 考察・将来課題

本稿で提案した簡易認証つきクロスドメインアクセス手法を用いることで,信頼できない Web サイトからのホームサーバーアクセスに一定の安全性をもたらすことができると考えられる.
ただし,将来のブラウザの機能拡張や,HTML の強化にともなって想定外の攻撃がなされる可能性はあり,我々の手法を用いれば常に安全というものでもない.今後も様々な手法が提案され,安全にアプリを開発・利用できる環境の構築が望まれる.
また,本稿ではホームサーバが危険を検出した際にユーザーに問いあわせを行う方法の提案を行ったが,実際にどのような操作が危険かについては言及していない.これについては既存研究[9]もあるので,今後実装していきたいと考えている.

参考文献

1) ECHONET Consortium http://www.echonet.gr.jp/
2) Kadecot by Sony CSL http://kadecot.net/
3) OpenECHO by Sony CSL https://github.com/SonyCSL/OpenECHO
4) iRemocon by グラモ http://i-remocon.com/
5) Open Weather Map API http://openweathermap.org/
6) 住宅 API by 大和ハウス工業株式会社 http://www.daiwahouse.co.jp/lab/HousingAPI/
7) NetFront Browser by ACCESS http://jp.access-company.com/products/browser/browser/
8) Chrome Packaged Apps http://developer.chrome.com/extensions/apps.html
9) 増田 耕一 「ホームネットワークにおける異常状態のモデル化とその見地手法に関する研究」,北陸先端科学技術大学院大学 情報科学研究科情報システム学専攻 修士論文, 2006 年 3 月