IoTデバイスの通信セキュリティ向上のための ホームネットワーク仮想化フレームワークの提案

塚崎 拓真[1]　滕 睿[2]　佐藤 健哉[1]

概要：近年，IoT(Internet of Things) が注目を集めるようになり，今後あらゆるモノがネットワークに接続され，利用されることが予想される．しかし，IoT の発展により利便性が高まる一方で，これまでネットワークに接続されていなかったモノが接続されることにより，セキュリティ上のリスクも高まっている．また，今後はホームネットワーク内で閉じたデバイス間の通信によって連携を行う形になることが想定される．デバイス間で直接通信を行う場合，各デバイスにおいてどのデバイスとの通信を受け入れるか，アクセス制御を行う必要がある．そこで本研究では，SDN(Software Defined Networks) の代表的プロトコルである OpenFlow を用いて，ホームネットワーク内の通信を監視するフレームワークの構築を検討した．また，提案システムでは，セキュリティ対策をオフロードした Proxy を仮想的に作成し．IoT デバイス間の通信を中継することで，本来 IoT デバイスに適用したいセキュリティ対策を実現した．そして，IoT デバイス間で閉じた通信を行うシミュレーションの評価を行い，ホームネットワークにおいてセキュリティ要件を保つことを示した．

Proposal of Home Network Virtualization Framework to Improve Communication Security of IoT Devices

TAKUMA TSUKASAKI[1]　RUI TENG[2]　KENYA SATO[1]

1. はじめに

近年，IoT(Internet of Things) が注目を集めるようになり，今後あらゆるモノがネットワークに接続され，利用されることが予想される．

しかし，IoT の発展により利便性が高まる一方で，これまでネットワークに接続されていなかったモノが接続されることにより，セキュリティ上のリスクも高まっている [1]．IoT デバイスは十分なセキュリティを考慮せずに開発されたものが多いため，悪意のある攻撃者によるサイバー攻撃の標的になりやすい．また，現在のスマートホームデバイスは，クラウド上のシステムと連携することで，デバイス間の連携を可能にしているが，今後はホームネットワーク内で閉じたデバイス間の通信によって連携を行う形になることが想定される．デバイス間で直接通信を行う場合，各デバイスにおいてどのデバイスとの通信を受け入れるか，アクセス制御を行う必要がある．しかし，IoT デバイスは従来の PC 等の既存機器と比較した場合，CPU 等のリソースを十分に保持していないため，デバイスの計算能力の制限やソフトウェア自体の脆弱性によって，適用できる機能が限られるという問題がある．そのため，ホームネットワーク内で通信するのであれば，どのデバイスも必ず利用するネットワークを利用したシステムを構築することが望ましい．

そこで本研究では，SDN(Software Defined Networks) の代表的プロトコルである OpenFlow を用いて，ホームネットワーク内の通信を監視するフレームワークの構築を検討した．提案システムでは，セキュリティ対策を適用可能なデバイスを Proxy と定義し，ルータ上に仮想的に作成する．ここに，IoT デバイスがリソース量の制限により適用できないセキュリティ対策をオフロードし，この Proxy が IoT デバイス間の通信を中継することで，本来 IoT デバイスに

[1] 同志社大学大学院 理工学研究科
Graduate School of Science and Engineering, Doshisha University
[2] 同志社大学モビリティ研究センター
Mobility Reserch Center, Doshisha University

適用したいセキュリティ対策を実現する．セキュリティ対策として，ホームネットワーク内の通信のトラフィック情報は既知であることを考慮し，フローの検証を OpenFlow コントローラで行う．

ルータ内にコンテナを配置し，そのコンテナ上に Proxy を作成する．そして，IoT デバイス間で閉じた通信を行うシミュレーションの評価を行い，ホームネットワークにおいてセキュリティ要件を保つことを示した．

2. 関連研究

2.1 ネットワークレベルの攻撃検知・防止

Sivanathan らは，SDN を用いてフローレベルでのトラフィックの動的な特性評価の使用を提案した [2]．これにより，データプレーンのトラフィックの一部のみを検査することになり，処理コストやネットワーク帯域のオーバヘッドの抑制を可能にした．また，提案システムにおいて，ノースバンド API を介して SDN コントローラと対話する解析エンジンを用い，IoT デバイスのネットワークを常に監視することを可能とした．パケットベースのモニタリングと比較し，セキュリティ上のメリットの大半を処理コストを大幅に削減しながら実現できることがわかった．

しかし，ホームネットワーク内の情報を外部で検査していることが問題点として挙げられる．現在のスマートホームデバイスは，クラウド上のシステムと連携することで，デバイス間の連携を可能にしているが，今後はホームネットワーク内で閉じたデバイス間の通信によって連携を行う形になることが想定される [3]．デバイス間で直接通信を行う場合，各デバイスでどういったデバイスとの通信を受け入れるか，アクセス制御を行う必要がある．しかし，全てのデバイスがアクセス制御に対応しているとは限らず，デバイスの計算能力の制限によって実現できるアクセス制御に制限があったり，デバイスのソフトウェア自体の脆弱性によってアクセス制御が機能しない場合が考えられる [4]．

2.2 ホームネットワーク運用の外部依存を避けた自己完結型システム

Zhang らは，クラウド上の遠隔サーバから制御されている現状のホームネットワークの問題点を挙げ，エンドユーザにシステムの完全な制御を提供するホームネットワークシステムを提案した [5]．提案システムは，IoT デバイスとアプリケーションがアプリケーション名付きのデータを介して通信し，データを直接保護することを可能にした．

しかし，各 IoT デバイスに対応したセキュリティ対策を施す柔軟性を持ち合わせていない．ホームネットワーク内には異なる規格のハードウェアや様々なアプリケーションが混在しているため，各デバイスに柔軟に対応できるシステムを構築することが望ましい．

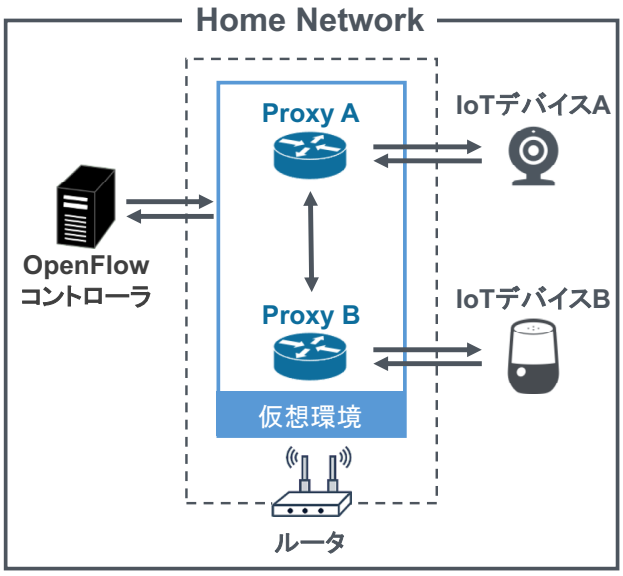

図 1 提案システムの構成

3. 提案システム

3.1 概要

提案システムでは，セキュリティ対策を適用可能なデバイスを Proxy と定義し，ルータ上に仮想的に作成する．ここに，IoT デバイスがリソース量の制限により適用できないセキュリティ対策をオフロードし，この Proxy が IoT デバイス間の通信を中継することで，本来 IoT デバイスに適用したいセキュリティ対策を実現する．セキュリティ対策として，ホームネットワーク内の通信のトラフィック情報は既知であることを考慮し，フローの検証を OpenFlow コントローラで行う．

3.2 システム構成

提案システムの構成を図 1 に示す．本提案システムの構成要素は，IoT デバイス，Proxy，ルータ，仮想環境から構成される．

- IoT デバイス
  本研究で扱う IoT デバイスは，センサーをはじめとした，CPU 等のリソースを十分に保持しておらず，直接セキュリティ対策を適用できないデバイスと定義する．
- Proxy
  IoT デバイスに要求されるセキュリティ対策を，仮想的に実現したものである．IoT デバイスからの通信を中継し，セキュリティ対策を適用する．セキュリティ対策ごとに作成し，IoT デバイスと紐づけることで，対象デバイスに応じた必要な対策を実現できる．
- ルータ
  IoT デバイス間通信の中継機器として用いる．ルータ上にコンテナを生成する．

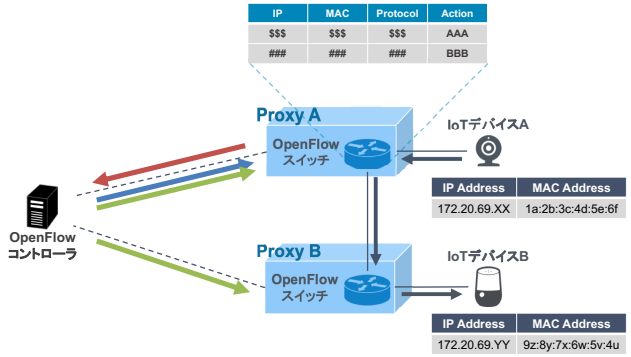

図 2　OpenFlow におけるフローチェック

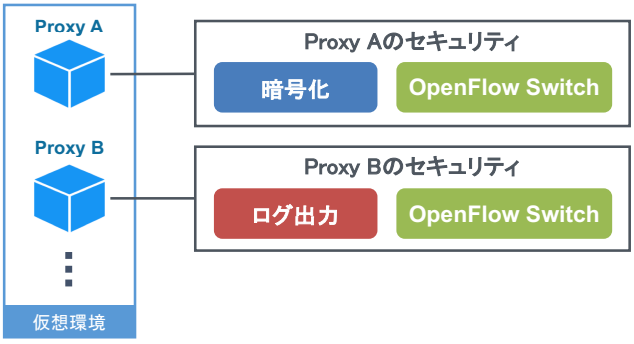

図 3　Proxy のセキュリティ対策

- 仮想環境
  Proxy の実行環境である．Proxy が作成される際に要求されるリソースを十分に提供することが可能である．

3.3　OpenFlow によるフローチェック

本研究におけるネットワーク監視を OpenFlow を用いて行う．OpenFlow によるフローチェックを図 2 に示す．一つの IoT デバイスに対し，Docker イメージからコンテナ上に OpenFlow スイッチの機能を生成する．IoT デバイスはこの OpenFlow スイッチを中継し，デバイス間通信を行う．OpenFlow コントローラは事前に IoT デバイスの情報を保持しており，デバイス間通信のフローテーブルを作成する．ホームネットワークの特性である通信形態が既知であることを考慮し，IP アドレスや通信頻度の確認を行い，ネットワークレベルにおける攻撃の検知を行う．

3.4　Proxy のセキュリティ対策

本研究におけるセキュリティ対策として，Proxy ごとに異なるセキュリティ対策を適用可能なことが挙げられる．仮想デバイスのセキュリティ対策の適用例を図 3 に示す．これによりリソースの都合上，IoT デバイスに直接適用できないセキュリティ対策を導入できることに加え，前述の IoT デバイスの問題点で述べたような様々なセキュリティ要件の変更に対しても柔軟に対応が可能となる．

また，コンテナ上で展開されるセキュリティ対策は適応

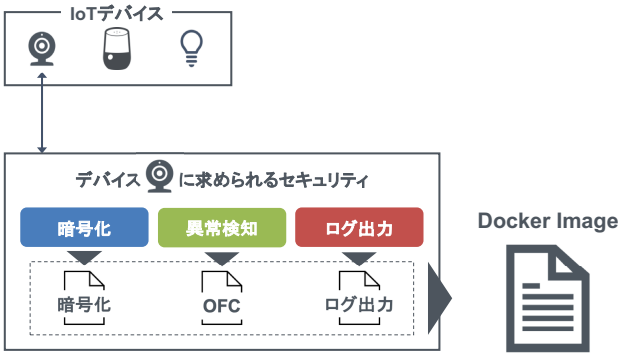

図 4　image ファイルの作成

表 1　実装環境

種類	項目	説明
Proxy	使用ソフト	Docker
OpenFlow	int	32

したいセキュリティ対策に対応したコンテナの image ファイルで定義される．image ファイルの作成図を図 4 に示す．この図のように IoT デバイスに対して適用したいセキュリティが複数ある場合においても，当該デバイスの規格に対応した対策をそれぞれ作成し，ソフトウェアモジュールのような形で組み合わせて定義することで，image ファイルを作成することが可能となる．

4.　実装

4.1　実装環境

本研究の実装環境，実装環境の構成をそれぞれに示す．Proxy の作成方法としては軽量なアプリケーション実行環境である Docker を利用した．Proxy を Docker で作成されるコンテナ状で稼働させることで複数の論理デバイスをリソース，オーバーヘッドを抑えて作成できることに加え，Docker Hub より配布される Docker Image を用いることで容易に作成可能となる．また今回扱う通信プロトコルとしては http(REST) を想定する．

4.2　動作手順

IoT デバイスの所有者であるユーザが本提案システムを利用する際の動作手順を図 5 と以下に示す．

(1) ユーザはデバイスを LAN 内に接続した後，Proxy の受付サーバへアクセス．
(2) Proxy はホームネットワーク内に IoT デバイスが接続されたことを確認．
(3) Proxy は OpenFlow コントローラへデバイス情報を送信．
(4) Proxy は Docker Hub から Docker Image を取得．
(5) 取得した Docker Image を基に仮想環境内に Docker Image を作成．
(6) Proxy はデバイス情報を基に IoT デバイスに接続を行

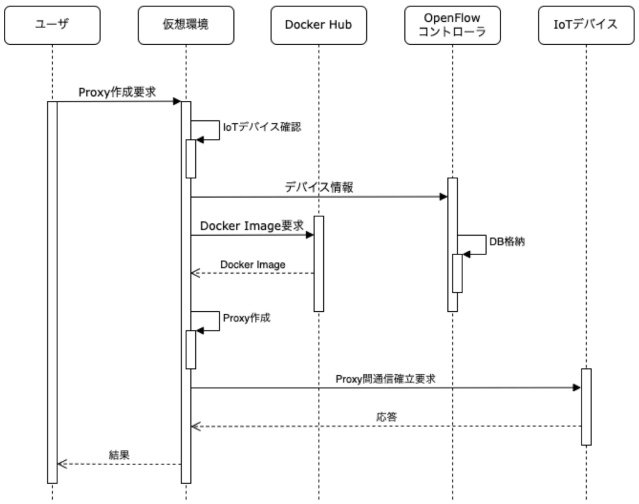

図 5 動作手順のシーケンス図

い，Proxy を経由して通信を行うように設定．

(7) 設定が完了し，IoT デバイス・Proxy 間の通信が確立された後，Proxy の作成・ネットワークの監視状況を報告．

4.3 想定ユースケース

本提案システムを用いた想定ユースケースを以下に示す．各 Docker イメージは Docker Hub というユーザが作成したコンテナをアップロードして公開・共有できるサービスを利用することを想定する．

- リソースを使うセキュリティ対策をあらかじめ提供する場合
  SSL による通信の暗号化など，IoT デバイスのリソースを多く利用するために適用できないセキュリティ対策をあらかじめ Docker イメージとして提供し，論理デバイス上で実現する．
- インシデント発生時などに対策を提供する場合
  事前に提供していたセキュリティ対策では想定していなかったインシデント等が発生した場合等に，当該デバイスの持つリソース量に依存せず，追加のセキュリティ対策を提供することが可能となる．

5. 評価

5.1 評価内容

本研究の評価として，まずセキュリティが適用されているかを検証した．今回は知らない IoT デバイスから通信があった場合と通信頻度が通常と異なる場合を想定した．それに対し，OpenFlow によるフローチェックが行われているかを検証した．

また，提案システムを適用した上で，IoT デバイス間で通信した際の EndtoEnd の時間を計測を行った．比較対象として，セキュリティ対策を適用していない場合についても計測を行った．

5.2 評価環境

今回適用するセキュリティ対策としては，鍵長が 1024bit の SSL による暗号化のイメージを Docker Hub より取得し，適用した．また，OpenFlow スイッチのイメージも取得し，OpenFlow によるフローチェックも行った．

6. 結果と考察

6.1 評価結果

登録済みの IoT デバイスかた通信要求が来た場合，登録していない IoT デバイスから通信要求が来た場合，通常の通信頻度と異なる通信がなされている場合のフローテーブルの結果を図 6 に示す．通常時は他のデバイスに対し，フローテーブルが作成されているが，異常時はパケットを Drop 処理するフローテーブルが作成されており，そのフローテーブルが削除されていることがわかる．

また，セキュリティ対策を施した提案システムとセキュリティ対策を施してないシステムにおける通信の比較結果を図に示す．

6.2 性能に関する考察

6.3 信頼性に関する考察

IoT デバイスを用いたシステムの安心安全を確保するための機能として，IPA により IoT 高信頼化昨日が定義されており，IoT 高信頼化要件として，IPA により IoT 高信頼化要件として，開始，予防，検知，回復，終了の 5 つの局面に分けてそれぞれセキュリティ要件が定義されている [7]．今回は前述の 5 つより，システム稼働中の局面である予防，検知，回復の 3 つにおける高信頼化要件に対し，提案システムの有効性について考察する．

- 予防の局面における考察
  予防の局面での高信頼化要件は，稼働中の異常発生を未然に防止できることである．これに対応する IoT 高信頼化機能としては，ログ取集機能，暗号化機能等があり，以上の予兆の把握，資産の保護を実現する．提案システムを用いることで，リソース量の関係で通常の IoT デバイスに適用できない機能であっても適用可能となる．
- 検知の局面における考察
  検知の局面での高信頼化要件は，稼働中の異常発生を早期に検知できることである．これに対応する IoT 高信頼化機能としては，状態監視機能，ログ収集機能があり，以上発生の検知や発生原因の特定を実現する．提案システムを用いることで，予防の局面同様，デバイスのリソース量に依存せず，求められる機能を実現できることに加え，Proxy は書く IoT デバイスごとに作成するため，個々のデバイスに応じた詳細な検知ルールを適用可能となる．
- 回復の局面における考察

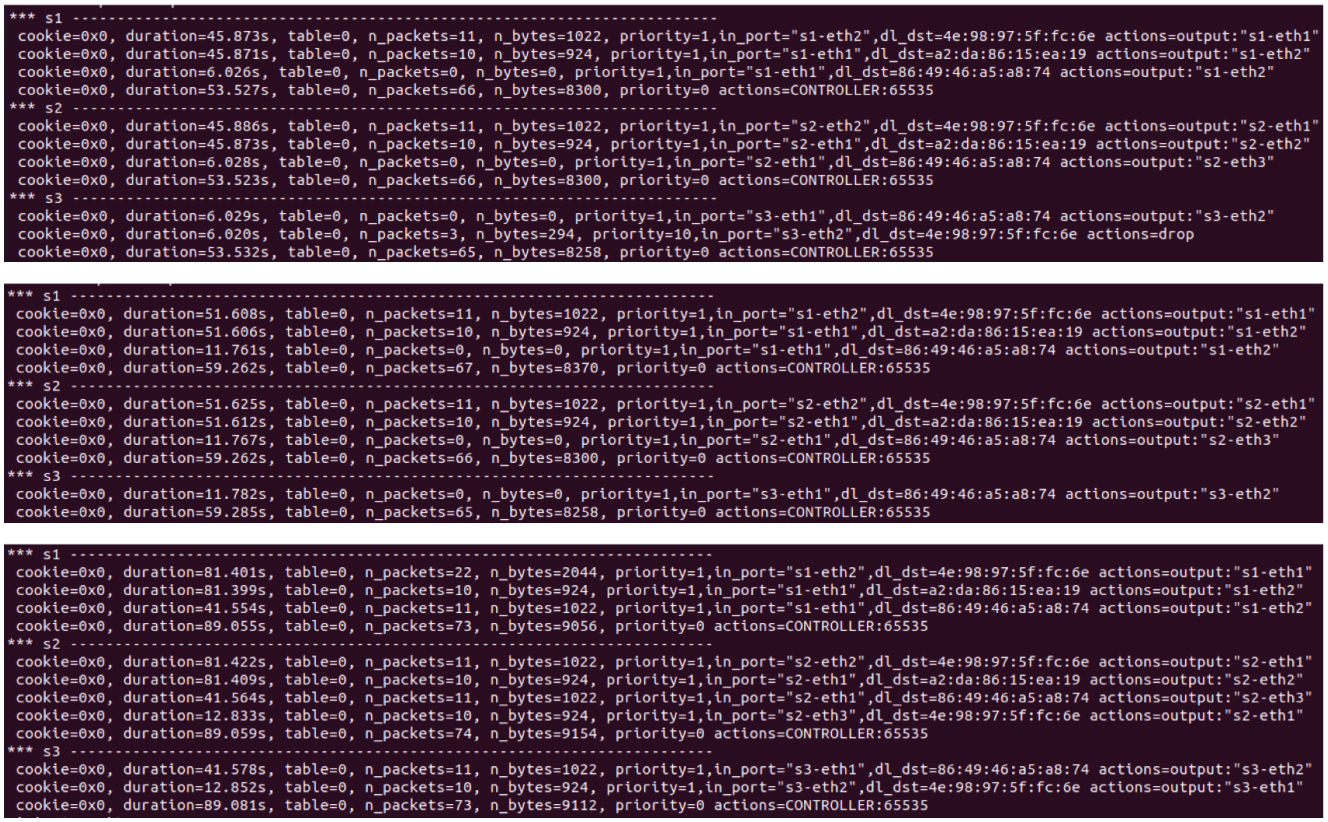

図 6 image ファイルの作成

回復の局面での高信頼化要件は，異常が発生した場合に稼働の復旧ができることである．特に IoT では，さまざまなデバイスが相互通信を行うため，事前に予測していなかった異常が発生することが考えられる．今回の環境では各セキュリティ対策は Docker Hub を通して Docker イメージとして提供することで，事前に作成したセキュリティ対策だけでなく，追加のセキュリティ対策も配布・適用が容易である．

7. まとめ

近年，IoT(Internet of Things) が注目を集めるようになり，今後あらゆるモノがネットワークに接続され，利用されることが予想される．しかし，IoT の発展により利便性が高まる一方で，これまでネットワークに接続されていなかったモノが接続されることにより，セキュリティ上のリスクも高まっている．また，今後はホームネットワーク内で閉じたデバイス間の通信によって連携を行う形になることが想定される．デバイス間で直接通信を行う場合，各デバイスにおいてどのデバイスとの通信を受け入れるか，アクセス制御を行う必要がある．そこで本研究では，SDN(Software Defined Networks) の代表的プロトコルである OpenFlow を用いて，ホームネットワーク内の通信を監視するフレームワークの構築を検討した．また，提案システムでは，セキュリティ対策をオフロードした Proxy を仮想的に作成し．IoT デバイス間の通信を中継することで，本来 IoT デバイスに適用したいセキュリティ対策を実現した．そして，IoT デバイス間で閉じた通信を行うシミュレーションの評価を行い，ホームネットワークにおいてセキュリティ要件を保つことを示した．

今後は，オーケストレータ等を用いて，新しい IoT デバイスがホームネットワークに追加された際に，自動的にコンテナが Proxy として配備される仕組みを検討する．また，Raspberry Pi 等の実機を用いた実験を行う予定である．

参考文献

[1] IoT 推進コンソーシアム, 総務省, 経済産業省, ”IoT セキュリティガイドライン ver 1.0”, 2016.

[2] A. Sivanathan, D. Sherratt, H. H. Gharakheili, V. Sivaraman and A. Vishwanath, ”Low-cost flow-based security solutions for smart-home IoT devices,” 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp. 1-6, 2016.

[3] C. Vallati et al., ”Mobile-Edge Computing Come Home Connecting things in future smart homes using LTE device-to-device communications”, IEEE Consumer Electronics Magazine, Vol.5, No.4, pp.77-83, 2016.

[4] M. Serror et al., ”Towards In-Network Security for Smart Homes”, Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES 2018), No.18, pp.1-8, 2018.

[5] Z. Zhang, T. Yu, X. Ma, Y. Guan, P. Moll and L. Zhang, ”Sovereign: Self-contained Smart Home with Data-centric Network and Security,” in IEEE Internet

of Things Journal, 2022.
[6] Nick McKeown et al., ”OpenFlow: enabling innovation in campus networks”, SIGCOMM Computer Communication Review, Vol.38, pp. 69 - 74, 2008.
[7] IPA 技術本部 ソフトウェア高信頼化センター (SEC), ”「つながる世界の開発指針」の実践に向けた手引き”, 2017.
[8] 情報処理学会: 情報処理学会論文誌 (IPSJ Journal) 原稿執筆案内，情報処理学会 (オンライン)，入手先 ⟨https://www.ipsj.or.jp/journal/submit/ronbun_j_prms.html⟩（参照 2022-03-01）.