Vol.2022-DPS-190 No.13
Vol.2022-CSEC-96 No.13
2022/3/10

自律型IoT システムのためのレジリエント・アーキテクチャに関する研究

白石 敬典[1,†1,a)]　橋本 正樹[1]　松井 俊浩[1]

概要：IoT システムはその普及に伴い，クラウドにデータが集中するモデルから，IoT とクラウドの間に中間層が存在するモデルへと変化しつつある．また，新技術を適用することで，将来的に IoT が自律的な処理機構を持つようになると考えられている．現状の IoT のセキュリティは，個別のセキュリティに基づいているが，アーキテクチャの変化や，IoT の自律的な処理機構が実現すると，デバイス同士が通信可能になることが予想されるため，IoT システムはその変化に対応する必要がある．特に攻撃されることを想定し，攻撃を前提とした回復力のあるアーキテクチャが重要になる．本研究では，「レジリエンス」に焦点を当て，攻撃を前提とする回復力を備えた自律型 IoT システムのためのレジリエント・アーキテクチャを提案する．IoT システムに対する代表的な脅威を分析し，レジリエンスに必要な要素を明確にした．また提案するアーキテクチャを基に，仮想化技術を用いたプロトタイプ実装を行い，仮想マシンの切り替えを平均 1 分 18 秒で実施可能なことを確認した．結論として，可用性を維持しつつ仮想マシンの切り替えを行い，異なる状態でシステムを構成可能であることを示した．

1. はじめに

近年，IoT は，家電製品といった身近なものから，自動車や医療・ヘルスケアといった様々な分野にも導入が進んでいる．また新しいネットワークモデルの提唱や，IoT デバイスへの新しい技術の適用も進んでおり，自律型 IoT システムの実現も期待されている．IoT を取り巻く環境は，私たちの生活をより良くするため変化し続けている反面，攻撃者にとっても魅力的な標的になっている．IoT セキュリティに目を向けると，脆弱な IoT が数多く存在しており，攻撃者の標的になっている [1]．このような IoT を標的とするマルウェアも増加傾向にある．例えば，2021 年 5 月には大手燃料パイプラインの Colonial Pipeline Co., が攻撃を受けたことにより，操業停止に追い込まれた．同社は，早期に操業を再開させる必要から攻撃者へ身代金を支払っている．

システムを安定的に運用するうえで，IoT システムの攻撃耐性を高めることは，社会的な課題であると共に急務である．そのためには，攻撃を受けても機能を止めることなく動き続けられる能力，また，被害からの回復能力としてのレジリエンスを IoT システムに備えることが重要である．本研究では，デバイス同士が繋がり，相互に通信するような自律型 IoT システムを想定し，そのためのレジリエント・アーキテクチャを提案する．

以降本稿は，第 2 章において自律型 IoT システムが，備えるべき特性について説明する．第 3 章では，IoT に関する研究状況について説明する．第 4 章では，IoT システムに対する脅威を整理，分析し，続く第 5 章で，分析した結果に対してレジリエンス視点で対策の整理，分析を行う．第 6 章で，回復と適応性に焦点を当てたレジリエント・アーキテクチャについて述べる．第 7 章では，実装したプロトタイプ及びその考察について説明する．第 8 章で，本稿の結論を述べる．

2. 自律型 IoT システム

本研究における自律型 IoT システムとは，IoT デバイス自身が状況を判断して，自分自身を制御するもので，人間による管理や判断が介在せずに動作し続けるシステムを指す．本章では，自律型 IoT システムについて，技術動向，社会的要請，備えるべき特性の 3 つの観点で調査した結果の，特に備えるべき特性を説明する．

2.1 備えるべき特性

備えるべき特性について説明する．国際標準規格として

[1] 情報セキュリティ大学院大学
IISEC, Yokohama, Kanagawa 221–0835, Japan
[†1] 現在，さくら情報システム株式会社
Presently with Sakura Information Systems Co., Ltd.
[a)] mgs204506@iisec.ac.jp

“ISO/IEC 30147:2021 Internet of Things (IoT) - Integration of IoT trustworthiness activities in ISO/IEC/IEEE 15288 system engineering processes[6]”に着目した．本規格の中心にあるのが，トラストワージネスである．トラストワージネスとは，セキュリティ，プライバシー，セーフティ，リライアビリティ，レジリエンスなどによって，システムがその関係者の期待に応える能力のことである．これらの特性の中で，レジリエンスは，ディペンダビリティにも含まれず，IoT のトラストワージネスに特徴的である．IT における PC やクラウドにおいては，情報漏洩がもっとも重要なリスクであるが，情報漏洩が発生した後に何らかの手段で被害を限定したり，回復手段を講ずることはむずかしく，いったん漏洩した情報に対するレジリエンスは考えにくい．一方，IoT デバイスが大量の機密情報を蓄えることはなく，情報漏洩よりも Bot 化，あるいは可用性への脅威に備える必要性が大きい．これらに対しては，レジリエンスの果たす役割は大きい．したがって，IoT のセキュリティをトラストワージネスから論ずる場合，レジリエンスに注目することは理にかなっている．

2.2 重要インフラにおけるレジリエンス

昨今の重要インフラに対する攻撃からレジリエンスに着目する．National Infrastructure Advisory Council (NIAC) のレポート [7] によると，「インフラのレジリエンスとは，破壊的なイベントの規模や期間を縮小する能力である．レジリエンスのあるインフラや企業の有効性は，破壊的なイベントに備え (anticipate)，緩和し (absorb)，それに適応し (adapt)，迅速に回復する (recover) 能力に依存する」とある．これら 4 つの能力を図示すると，図 1 のようなシーケンスになっている．

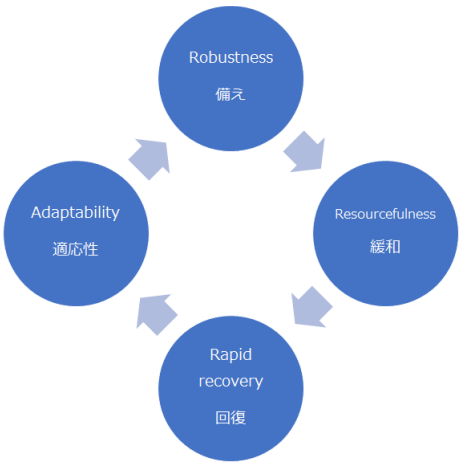

図 1 レジリエンスのシーケンス

各能力は，以下の通りである．

- Robustness: 備え (anticipate に相当する能力)
  災害に直面しても，稼働し続ける，あるいは立ち続けることができる能力のこと [7].
- Resourcefulness: 緩和 (absorb に相当する能力)
  災害が発生したときに，それを巧みに管理する能力のこと [7].
- Rapid recovery: 回復 (recover に相当する能力)
  災害後にできるだけ早く元の状態に戻す能力のこと [7].
- Adaptability: 適応性 (adapt に相当する能力)
  大惨事から得られる新たな教訓を吸収する手段のこと [7].

システムは，どれだけ事前に備えていても問題は発生し，それを避けることができない．如何に問題から回復し，次に活かすかが重要になる．本研究では，これら 4 つの目標のうち，攻撃を前提としたシステムにおいては，特に回復及び適応性に焦点を当てる．

3. 関連研究の状況

本章では，IoT に関する様々な関連研究を紹介する．

3.1 OS 再起動による OS イメージの更新とライフサイクル管理

須崎 et al. [8] は，Reboot-Oriented IoT(以降，RO-IoT) を提案している．RO-IoT は，攻撃者による乗っ取り・不正アクセスといった，脅威に侵害された IoT デバイスが自律的に OS の再起動を実行することで，システムの復旧を可能とするメカニズムである．その実現には，TEE を備えた IoT デバイスが使われ，同時に PKI ベースの証明書を用いたライフサイクル管理も実現している．

3.2 IoT マルウェアを使ったマルウェア対策

山口 [9] は，IoT マルウェアに対抗するアプローチとして White-Hat Worm を提案している．この提案は，2016 年 9 月に大規模な分散型サービス拒否 (DDoS) 攻撃を引き起こした IoT マルウェア Mirai に対抗するものである．

3.3 サイバーデセプションを用いたプロアクティブな防御

Ge et al.[10] は，SDN をベースとした IoT ネットワークのためのプロアクティブな新しい防御技術 NTS-MTD を提案している．著者等の研究では，IoT ネットワークにデコイノードと本番ノードを混在させたうえで，ネットワークトポロジーをシャッフルするために SDN 機能を利用している．

3.4 様々な監視方式

IoT デバイスの監視方式には様々なものが存在する．ホスト型侵入検知として“HADES-IoT”[11] やネットワークトラフィック分析による異常検出システム“IoT-KEEPER”[12]，クラウドベースのアンチマルウェアシステム“CloudEyes”[13] などである．

3.5 既存研究とレジリエンス

IoT セキュリティに関する様々な研究分野のサーベイを行った．IoT セキュリティでは，新しい技術を用いた防御や監視といった，「守り」の研究は盛んに行われている状況にある．反面，IoT の分野におけるレジリエンスの研究は，まだまだ不十分な状況にある．IoT デバイスやシステムが何らかの脅威によって侵害され，影響が生じる「事後」の研究は，未成熟な分野であると考える．将来的に本研究が想定する，IoT デバイス自身が状況を判断して，自分自身を制御する自律型 IoT システムが実現すると，守るだけでは不十分であり，IoT デバイス自身が脅威に対して回復し，同じ脅威に対して学習する必要があると考えている．

4. 脅威の整理と分析

本章では，IoT システムに対する代表的な脅威を整理し，脅威に対するレジリエンス視点での分析を行う．Vikas et al. は，既存の IoT システムにおける様々なアプリケーションや脅威，セキュリティソリューション等に関して詳細な調査を行い，4 つの層と 1 つのベースとなる広義の層と定義している [14]．本章では，Vikas et al. が提唱する IoT システムに対する脅威を基に，脅威と対策の整理・分析を行う．

4.1 センシング層に対する脅威

センシング層に対する代表的な脅威を説明する．IoT デバイスはオープンな場合が多く，直接 IoT デバイスに対して攻撃を実行することが可能である他に，IoT デバイスの近距離から攻撃を実行可能な点がある．以下にセンシング層に対する脅威を示す．

- Booting Attacks
  ブートプロセスが保護されていない IoT ノードは様々な攻撃を受けやすい．攻撃者は，ブートプロセスが保護されていない IoT デバイスの再起動時に攻撃を実行する可能性がある [14]．また関連研究として，J. Alex et al. は，電源を失っても数秒間は DRAM に情報が残存することに言及している．調査によると攻撃者は，デバイスを再起動することで DRAM が保持している機密情報を窃取することができる可能性がある [15]．残る脅威は，Node Capturing，Malicious Code Injection Attack，False Data Injection Attack，Side-Channel Attacks (SCA)，Eavesdropping and Interference，Sleep Deprivation Attacks である．

4.2 ネットワーク層に対する脅威

ネットワーク層に対する代表的な脅威を説明する．IoT デバイスのクレデンシャル情報や，IoT デバイスが収集・生成するデータに対する攻撃，また IoT デバイスを踏み台とする攻撃である．以下にネットワーク層に対する脅威を示す．

- DDoS/DoS Attack[14]
  攻撃者は，標的に対して大量の不要なリクエストを送りつけることで，サービス拒否を引き起こす可能性がある．特に攻撃に使用するソースが複数ある場合は DDoS と呼ばれる [14]．また関連研究として，Monika et al. は，IoT ネットワークにおける DDoS の検出に関する調査を行っている．調査によると攻撃者は，IoT デバイスの脆弱性を利用してデバイスを侵害することで Bot 化する．これらの Bot は標的となるシステムに対して複数の攻撃を実施することで甚大な損害を与え，ユーザーがリソースやサービスにアクセスできなくする [16]．残る脅威は，Phishing Site Attack，Access Attack，Data Transit Attacks，Routing Attacks である．

4.3 ミドルウェア層に対する脅威

ミドルウェア層に対する代表的な脅威を説明する．Vikas et al. はミドルウェア層を，ネットワーク層とアプリケーション層の間にある抽象化層としており，IoT アプリケーションのための API を提供する．そのため既存の Web アプリケーションに対する脅威も対象となっている．以下にミドルウェア層に対する脅威を示す．

- Man-in-the-Middle Attack[14]
  MQTT プロトコルでは，MQTT ブローカーを用いて，クライアントとサブスクライバー間の通信にパブリッシュ・サブスクライブ・モデルが採用されている．攻撃者は，ブローカーを制御し中間者になることで通信を制御できる [14]．また関連研究として，Henry et al. は MQTT プロトコルを使用する IoT デバイスに対する中間者攻撃について調査を行っている．MQTT 自体はセキュリティ機能を有しておらず，SSL/TLS を使用することでメッセージを暗号化する．しかし調査から SSL/TLS を使用しない通信が数多く存在していることを指摘している．このようにメッセージを暗号化しない場合，中間者攻撃によって通信を盗聴されたり，改ざんされる可能性がある [17]．残る脅威は，SQL Injection Attack，Signature Wrapping Attack，Cloud Malware Injection，Flooding Attack in Cloud である．

4.4 ゲートウェイに対する脅威

ゲートウェイに対する代表的な脅威を説明する．Vikas et al. は IoT システムの 4 つの層以外にゲートウェイを定義している．ゲートウェイは，IoT デバイスや人，クラウドサービスなどを繋げる広い意味での層と，IoT デバイスを構成するハードウェアやソフトウェア，そして通信プロトコルを含んだものである．以下にゲートウェイ層に対す

る脅威を示す.

- Secure On-boarding[14]
  IoT システムに新規でデバイスやセンサーが導入される場合，暗号鍵の保護が必要になる．ゲートウェイでは全ての通信が通過する為，オンボーディング・プロセスは暗号鍵を窃取しようとする中間者攻撃や盗聴を受けやすい [14]．残る脅威は，Extra Interfaces, End-to-End Encryption，Firmware updates である．

4.5 アプリケーション層に対する脅威

アプリケーション層に対する代表的な脅威を説明する．アプリケーション層では，IoT デバイスが収集・生成したデータを基に様々な処理を実施する．特にコンピューティングパワーの潤沢なクラウドサービスでは AI 学習による分析などを行い，エンドユーザーに対して価値を提供する．以下にアプリケーション層に対する脅威を示す.

- Data Thefts[14]
  IoT アプリケーションでは多くの重要またはプライベートなデータが扱われ，転送中のデータは攻撃を受けやすい．攻撃者は，IoT アプリケーションが，データの暗号化や分離，ユーザーやネットワークの認証，プライバシー管理に関して脆弱な点がある場合，データ盗難のリスクがある [14]．また関連研究として，Yousef et al. によると，規制や認定されてない安全性の低い IoT デバイスを介すとデータ漏えいに繋がる可能性がある [18]．残る脅威は，Access Control Attacks, Service Interruption Attacks，Malicious Code Injection Attacks，Sniffing Attacks，Reprogram Attacks である．

5. 対策の整理と分析

前章では，Vikas et al. が提唱する 4 つの層 (センシング，ネットワーク，ミドルウェア，アプリケーション) とゲートウェイ毎の代表的な脅威を説明した．脅威を分析すると攻撃手法に類似点があることがわかった．本章では，類似点から脅威をまとめ，それに対してレジリエンス視点による対策を示す.

5.1 脅威に対するレジリエンス視点での対策の整理・分析

前項では，IoT システムの各層における代表的な脅威について説明した．本項では，代表的な脅威に対して 2.4.2 で説明したレジリエンスの 4 つの達成目標による対策例を示す.

5.2 センシング層における脅威への対策

センシング層における代表的な脅威から，乗っ取り，サービス拒否，情報窃取に分類した．これらの脅威に対するレジリエンス視点での対策例 (一部) を示す.

- Robustness (anticipate (備え))
  ・同一機能を提供する IoT デバイスを冗長（マスター/スレーブ）で構成する
  ・同一機能を提供する IoT デバイスを複数台（クラスタリング）で構成する
  ・IoT デバイスが収集，生成するデータのバックアップを取得する
- Resourcefulness (absorb (緩和))
  ・当該通信を遮断する
  ・当該 IoT デバイスを運用環境から隔離する
  ・IoT デバイスのサービスを停止する
- Rapidly recover (recover (回復))
  ・IoT デバイスの OS を再起動する
  ・IoT デバイスの OS を再インストールする
  ・正常な IoT デバイスにサービスを移行する
  ・バックアップから復旧する
- Adaptability (adapt (適応))
  ・侵害された時とは異なる構成を取る
  ・侵害された情報に関連するパラメータを変更する

5.3 ネットワーク層における脅威への対策

ネットワーク層における代表的な脅威から，乗っ取り，サービス拒否，情報窃取に分類した．これらの脅威に対するレジリエンス視点での対策例 (一部) を示す.

- Robustness (anticipate (備え))
  ・IoT デバイスが収集，生成するデータのバックアップを取得する
  ・ネットワークトポロジーを細かく構成する
  ・IoT デバイスの状態やトラフィックを監視する
- Resourcefulness (absorb (緩和))
  ・当該通信を遮断する
  ・当該 IoT デバイスを運用環境から隔離する
  ・IoT デバイスのサービスを停止する
- Rapidly recover (recover (回復))
  ・IoT デバイスの OS を再起動する
  ・IoT デバイスの OS を再インストールする
  ・正常な IoT デバイスにサービスを移行する
  ・バックアップから復旧する
- Adaptability (adapt (適応))
  ・侵害された時とは異なる構成を取る
  ・侵害された情報に関連するパラメータを変更する

5.4 ミドルウェア層における脅威への対策

ミドルウェア層における代表的な脅威から，乗っ取り，サービス拒否，情報窃取，情報改ざんに分類した．これらの脅威に対するレジリエンス視点での対策例 (一部) を示す.

- Robustness (anticipate (備え))
  ・複数のクラウドサービスを利用する

・CDN サービスを利用する
・ネットワークトポロジーを細かく構成する
- Resourcefulness (absorb (緩和))
・当該通信を遮断する
・当該 IoT デバイスを運用環境から隔離する
・IoT デバイスのサービスを停止する
- Rapidly recover (recover (回復))
・バックアップから復旧する
・IoT デバイスの OS を再起動する
・IoT デバイスの OS を再インストールする
- Adaptability (adapt (適応))
・侵害された時とは異なる構成を取る
・侵害された情報に関連するパラメータを変更する

5.5 ゲートウェイにおける脅威への対策

ゲートウェイ層における代表的な脅威から，乗っ取り，サービス拒否，情報窃取に分類した．これらの脅威に対するレジリエンス視点での対策例 (一部) を示す．

- Robustness (anticipate (備え))
・(FIDO FDO のような) セキュアなオンボーディングプロセスを採用する
・安全にファームウェアを導入する仕組みを導入する
- Resourcefulness (absorb (緩和))
・当該通信を遮断する
・当該 IoT デバイスを運用環境から隔離する
・IoT デバイスのサービスを停止する
- Rapidly recover (recover (回復))
・IoT デバイスの OS を再インストールする
・IoT デバイスの OS を再起動する
- Adaptability (adapt (適応))
・侵害された時とは異なる構成を取る
・侵害された情報に関連するパラメータを変更する

5.6 アプリケーション層における脅威への対策

アプリケーション層における代表的な脅威から，乗っ取り，サービス拒否，情報窃取に分類した．これらの脅威に対するレジリエンス視点での対策例 (一部) を示す．

- Robustness (anticipate (備え))
・IoT デバイスの操作や通信等のログを取得する
・IoT デバイスの状態やトラフィックを監視する
・ネットワークトポロジーを細かく構成する
- Resourcefulness (absorb (緩和))
・当該通信を遮断する
・当該 IoT デバイスを運用環境から隔離する
・IoT デバイスのサービスを停止する
- Rapidly recover (recover (回復))
・IoT デバイスの OS を再起動する
・IoT デバイスの OS を再インストールする
・正常な IoT デバイスにサービスを移行する
・バックアップから復旧する
- Adaptability (adapt (適応))
・侵害された時とは異なる構成を取る
・侵害された情報に関連するパラメータを変更する

5.7 対策に対する分析

各層で脅威を類似点でまとめ，それらに対してレジリエンスの達成目標毎の対策例を示した．対策に要するコストや難易度といった要員は考慮せず，対策例を提示している．すべての対策を取ることが望ましいが，IoT システムが提供するサービスの特性や性質，重要度等を考慮した上で，取捨選択する必要がある．

6. 提案するアーキテクチャ

回復と適応性に焦点を当てたレジリエント・アーキテクチャについて述べる．備え及び緩和についても最後に説明する．今回提案するアーキテクチャでは，IoT デバイスを Linux ベースに限定している．これは，Dominik et al. の研究において，IoT デバイスの主要な OS として Linux が 80%以上のシェアを有していることが言及されていることから，十分に一般的であると考えるからである [11].

6.1 レジリエント・アーキテクチャの概観図

図 2 は，提案するレジリエント・アーキテクチャの概観図である，各要素については，次項で説明する．

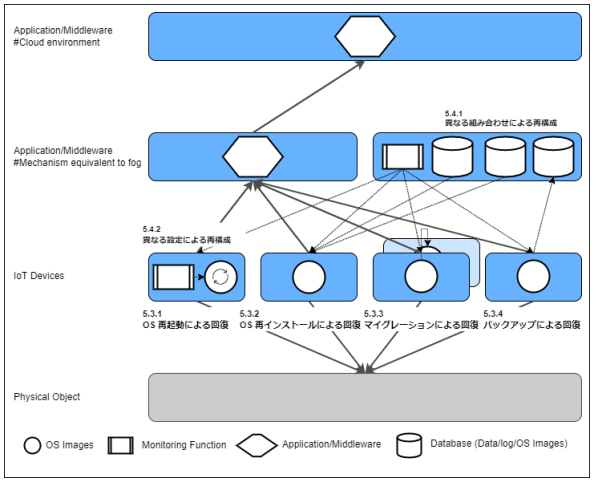

図 2 レジリエント・アーキテクチャ

6.2 回復

本項では，回復について説明する．回復とは，事象からできるだけ早く元の状態に戻す能力を指す．ここでは 4 つの回復手段について説明する．

6.2.1 OS 再起動による回復

レジリエント・アーキテクチャを検討するうえで，IoT デバイスの OS 再起動は，最も簡単な回復手段である．上述の通り，IoT デバイスの主要な OS として Linux が 80%以上のシェアを有している [11]．そのため，OS 再起動の契機としては，watchdog を利用することが可能である．watchdog とは，システムが正常に動作していることを kernel に伝えるデーモンである．watchdog の仕組みを利用することで，様々な障害を監視し，正常時とは異なる状態になった場合に OS 再起動を実行することができる．

6.2.2 OS 再インストールによる回復

マルウェアの中には，ファイルシステムやブートローダに感染して，OS 再起動後も持続するタイプのものがある [8]．その場合の回復手段として OS 再インストールがある．OS 再インストールは，ネットワーク経由で実行することが想定される．例えば，フォグ・コンピューティングを用いることで，ネットワーク経由で必要な処理を実行することが可能になる．

6.2.3 マイグレーションによる回復

この回復手段には，IoT デバイスにハイパーバイザーが利用されていることを前提にしている．既存のハイパーバイザーには，KVM や XEN などがあるが，IoT デバイス向けに設計されたハイパーバイザーが存在する．このような IoT デバイス向けに設計されたハイパーバイザーを搭載した IoT デバイス上で，複数の VM を提供することが可能であれば，VM をマイグレーションすることで回復することができる．

6.2.4 バックアップによる回復

IoT デバイスが収集・生成するデータのバックアップを取得しておくことで，回復させる手段である．この場合，データのバックアップ先として利用するのは，前項で説明したフォグ・コンピューティングなどの IoT デバイス以外の機構である．フォグ・コンピューティングは，IoT デバイスの至近距離に配置されることが想定されるため，フォグ・コンピューティングでデータのバックアップを取得する．

6.3 適応性

本項では，適応性について説明する．適応性とは，発生した事象から得られる新たな教訓を次に活かす能力を指す．ここでは 2 つ説明する．

6.3.1 異なる組み合わせによる再構成

IoT デバイス内で使われるソフトウェアや OS 等の脆弱性を突いた攻撃を受けたと仮定する．攻撃を受けた IoT デバイスは，攻撃を受けたときの組み合わせではない，異なる組み合わせで再構成される．これにより，同じ脆弱性を突いた攻撃によって影響を受ける可能性は低くなる．なお再構成する上で既存のファイル等で必要なもの (/etc/passwd 等) がある場合，予めバックアップを取るといった考慮をする必要がある．

6.3.2 異なる設定による再構成

不正アクセスによって IoT デバイスが侵害されたと仮定する．侵害された IoT デバイスが，OS 再起動によって回復したとして，ID やパスワードといったクレデンシャル情報が変更されていることで，少なくとも侵害されたときに使用された情報は効かなくなり，影響を受ける可能性は低くなる．

6.4 備え及び緩和

本項では，レジリエンスの 4 つの達成目標の残る二つ，備えと緩和について説明する．備えとは，事象に直面しても稼働し続ける能力を指す．また緩和とは，事象が発生した際に実施する能力を指す．これらは必要に応じて検討すべき達成目標である．

6.4.1 処理の冗長化

データを収集・生成する IoT デバイスを冗長化することは必要である．例えば，複数の IoT デバイスによって冗長化することで処理の途絶を防ぐことである．これはレジリエンス達成目標の備えに該当する．

6.4.2 影響範囲の最小化

当該通信の遮断，IoT デバイスの隔離，処理や OS の停止などによって被害範囲を可能な限り最小となるようにする．またシステム・ネットワーク構成を細かくすることで，予め被害範囲を小さくする．これはレジリエンス達成目標の備え及び緩和に該当する．

6.4.3 異常の検出

IoT システム内の様々なところで監視を行う．例えば，前項で説明した watchdog によって IoT デバイスによって自身を監視するだけでなく，ハイパーバイザーから VM への監視や，フォグ・コンピューティングから各 IoT デバイスに対する監視を行う．このように複数のポイントから監視を行い，異常を検出することで，被害箇所を早期に特定し，影響範囲を最小となるようにする．これはレジリエンス達成目標の備え及び緩和それぞれに該当する．

7. プロトタイプ実装と考察

本章では，実装したプロトタイプ及びその考察について述べる．今回のプロトタイプでは，回復として，『マイグレーションによる回復』，適応性として，『異なる組み合わせによる再構成』を用いた．Raspberry Pi を用い，仮想化技術等を組み合わせて，システムにおける回復と適応性について考察を加えた．プロトタイプ実装では，導入する上で，最も低コスト且つ導入難易度の低い実装を目標に，複数の IoT デバイスや外部機構 (フォグ・コンピューティング) を利用せず，単一の IoT デバイス内で全て完結するようにした．

7.1 実装したプロトタイプの概観とシーケンス

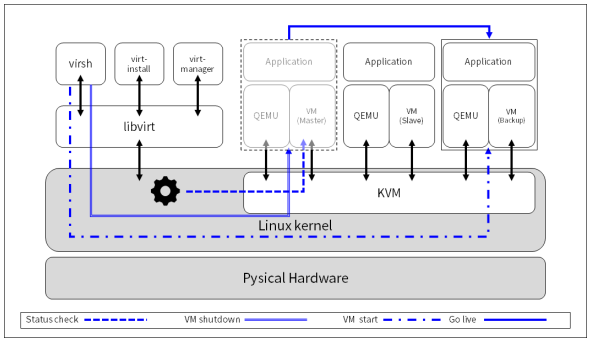

図 3 プロトタイプのシーケンス

7.2 実装

IoT デバイスでは，ホストマシンとして Ubuntu を使い，その上で動く仮想マシンを 3 台用意した．仮想マシンは，サービスを提供するマスターとそのスレーブ，そしてバックアップの 3 台で構成している．各仮想マシン間では，冗長化がとられており，基本的にはプライオリティ値の高い仮想マシンにリクエストが届くように設定を行った．またハイパーバイザーとして KVM を，仮想マシンの制御には libvirt を使用した．

7.3 考察

プロトタイプに対して想定するシナリオをもとに，動作確認を行った．本節では，想定シナリオ及び回復，適応性視点での考察を述べる．

7.3.1 想定シナリオ

サービスを提供する 1 台の IoT デバイスが，IoT マルウェア等によって侵害されることを想定している．プロトタイプでは，各仮想マシンで設定した vip (192.168.122.254) でリクエストを受け付ける．仮想マシンの実 IP は，Master を 192.168.122.166，Slave を 192.168.122.130，Backup を 192.168.122.184 とした．通常リクエストは vip で受け，keepalived の設定で priority が最も高い仮想マシン (Master) がサービスを提供する．

今回は Master が IoT マルウェア等に侵害され，サービスで使用しているポートが意図的な操作によって，閉塞されたものと仮定する．ホストマシンから毎分ポートの状態を監視しており，閉塞された場合，外部から接続不可能となるため，メッセージによって気付くことが可能である．その後，ホストマシンが libvirt を介して virsh を実行し，仮想マシン (Master) の停止と仮想マシン (Backup) の起動を実行する．

7.3.2 回復

プロトタイプ実装では，仮想マシンを切り替えることで回復することを示した．Web ブラウザを用いて正常性の確認を行った限り，目に見えるサービス断を確認することは出来ず，スムーズに切り替えを行うことを確認した．また仮に切り替えが正常に行えなかった場合でも，仮想マシン (Slave) によってサービス提供は可能であり，回復と可用性を実現したと言える．

7.3.3 適応性

プロトタイプ実装では，停止する仮想マシン (Master) とは異なる kernel，異なるソフトウェア (HTTP Server)，そして異なるバージョンのスクリプト言語で仮想マシン (Backup) を起動するように設定を行った．これは停止する仮想マシン (Master) はなんらかの脆弱性によって侵害されたと仮定しており，その状態とは異なる状態で仮想マシンを起動させることで，少なくともその脆弱性による侵害を受けないことを意味する．

7.3.4 切り替えに要する時間

30 回の切り替えを実行したところ，最短で 1 分 16 秒，最長でも 1 分 21 秒で切り替えを行えることがわかった．また 30 回平均だと 1 分 18 秒である．この間は，仮想マシン (Slave) がサービスを提供しており，可用性は保たれた状態にある．また今回のプロトタイプでは，仮想マシンの停止と起動の間に 60 秒間の遅延処理を加えている．遅延処理の時間を短く調整し，また不要なサービス等を停止することで切り替えに要する時間は十分に短縮可能であると考えている．なお今回はあくまでもプロトタイプで切り替えを 30 回実施しただけであり，切り替え回数自体は，統計的に有意な回数ではない．

7.3.5 切り替え実施時のホストマシンの負荷

今回のプロトタイプ実装では，IoT デバイス内で常時 2 台の仮想マシンが起動するように構成した．切り替え実施時のホストマシンの負荷を確認する限り，極端なハードウェアリソースの消費は確認することはできず，切り替え自体も想定通り実施可能であった．ただし仮想マシン上で IoT アプリケーションを実行してはいないため，実際に利用する際は，上記以上の負荷が生じる可能性が考えられる．

8. おわりに

本研究では，自律型 IoT システムのためのレジリエント・アーキテクチャを提案した．はじめに，IoT セキュリティに関する様々な研究を網羅的に調査することで，代表的な脅威に対抗するためのレジリエント・アーキテクチャに必要な要素を整理した．また，その結果を踏まえた上で，IoT システムのための回復と適応性に焦点を当てたアーキテクチャを示した．本アーキテクチャは，攻撃被害からの回復力を備えたもので，特に重要インフラといった，高い可用性が求められ，サービス断が許されない環境への適用を期待するものである．その後，そのアーキテクチャの一部をプロトタイプとして実装し，IoT デバイスに仮想化技術を適用することで，可用性を維持しつつサービスの切り

替えが可能なことを示した．同時に適応性についても，切り替え前後で異なるソフトウェア構成で同様のサービスを提供するシステムを実現可能であることを確認した．

今後の課題として，自律型 IoT システムでは，セキュリティやプライバシーといったレジリエンス以外の能力も当然必要になると考える．よって，レジリエンス以外のトラストワージネスを構成する能力 (セキュリティ，プライバシー，セーフティ，リライアビリティ) を考慮したアーキテクチャの検討がある．

参考文献

[1] Quoc-Dung Ngo, Huy-Trung Nguyen, Van-Hoang Le and Doan-Hieu Nguyen, A survey of IoT malware and detection methods based on static features, ICT Express Volume 6, Issue 4, December 2020, Pages 280-286, DOI: https://doi.org/10.1016/j.icte.2020.04.005

[2] IEEE Future Directions New Technology Proposal List – IEEE Future Directions, IEEE. https://cmte.ieee.org/futuredirections/ieee-future-directions-new-technology-proposal-list/ (参照 2022/01)

[3] The top technology trends — McKinsey, McKinsey & Company. https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-top-trends-in-tech (参照 2022/01)

[4] スマートシティプロジェクト｜スマートシティ官民連携プラットフォーム，スマートシティ官民連携プラットフォーム事務局. https://www.mlit.go.jp/scpf/projects/index.html (参照 2022/01)

[5] 「国家戦略特別区域法の一部を改正する法律」の成立について, スマートシティ官民連携プラットフォーム事務局. https://www.chisou.go.jp/tiiki/kokusentoc/kettei/r202005.html (参照 2022/01)

[6] ISO/IEC 30147:2021 — IEC Webstore, IEC 2021. https://webstore.iec.ch/publication/62644 (参照 2022/01)

[7] Alfred R. Berkeley III and Mike Wallace, A Framework for Establishing Critical Infrastructure Resilience Goals, National Infrastructure Advisory Council October 19, 2010

[8] Kuniyasu Suzaki, Akira Tsukamoto, Andy Green and Mohammad Mannan, Reboot-Oriented IoT: Life Cycle Management in Trusted Execution Environment for Disposable IoT devices, In Proceedings of Annual Computer Security Applications Conference (ACSAC '20). Association for Computing Machinery, New York, NY, United States, pp 428–441. December 2020. DOI: https://doi.org/10.1145/3427228.3427293

[9] Shingo Yamaguchi, White-Hat Worm to Fight Malware and Its Evaluation by Agent-Oriented Petri Nets †, In Proceeding of the IEEE 6th International Conference on Consumer Electronics – Taiwan (IEEE 2019 ICCE-TW), Yilan, Taiwan, pp 20–22. May 2019. DOI: https://doi.org/10.3390/s20020556

[10] Mengmeng Ge, Jin-Hee Cho, Dongseong Kim, Gaurav Dixit and Ing-Ray Chen, Proactive Defense for Internet-of-things: Moving Target Defense With Cyberdeception, ACM Transactions on Internet Technology Volume 22 Issue 1 February 2022 Article No.: 24 pp 1–31. DOI: https://doi.org/10.1145/3467021

[11] Dominik Breitenbacher, Ivan Homoliak, Yan Lin Aung, Nils Ole Tippenhauer and Yuval Elovici, HADES-IoT: A Practical Host-Based Anomaly Detection System for IoT Devices, In Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (Asia CCS '19). Association for Computing Machinery, New York, NY, United States, pp 479–484. July 2019. DOI: https://doi.org/10.1145/3321705.3329847

[12] Ibbad Hafeez, Markku Antikainen, Aaron Yi Ding and Sasu Tarkoma, IoT-KEEPER: Detecting Malicious IoT Network Activity Using Online Traffic Analysis at the Edge, IEEE Transactions on Network and Service Management, Volume 17, Issue 1. pp 45–59. March 2020. DOI: https://doi.org/10.1109/TNSM.2020.2966951

[13] Hao Sun, Xiaofeng Wang, Rajkumar Buyya and Jinshu Su, CloudEyes: Cloud-based malware detection with reversible sketch for resource-constrained internet of things IoT devices, Software—Practice & Experience, Volume 47, Issue 3. pp 479–484. March 2017. DOI: https://doi.org/10.1002/spe.2420

[14] Vikas Hassija, Vinay Chamola, Vikas Saxena, Divyansh Jain, Pranav Goyal and Biplab Sikdar, A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures, IEEE Access (Volume: 7), June 2019, Pages 82721 - 82743, DOI: http://dx.doi.org/10.1109/ACCESS.2019.2924045

[15] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum and Edward W. Felten, Lest we remember: cold-boot attacks on encryption keys, Communications of the ACM Volume 52 Issue 5. Association for Computing Machinery, New York, NY, United States, pp 91-98. May 2009. DOI: https://doi.org/10.1145/1506409.1506429

[16] Monika Roopak, Gui Yun Tian and Jonathon Chambers, Multi-objective-based feature selection for DDoS attack detection in IoT networks, IET Networks Volume 9, Issue 3, pp 120-127. May 2020. DOI: https://doi.org/10.1049/iet-net.2018.5206

[17] Henry T. Wong and Tie Luo, Man-in-the-Middle Attacks on MQTT-based IoT Using BERT Based Adversarial Message Generation, In conjunction with the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2020). San Diego, California, USA. August 2020. DOI: -

[18] Yousef Amar, Hamed Haddadi, Richard Mortier, Anthony Brown, James Colley and Andy Crabtree, An Analysis of Home IoT Network Traffic and Behaviour, arXiv: Networking and Internet Architecture. March 2018. DOI: -