NIST 内部報告書

NIST IR 8349

モノのインターネット（IoT） デバイスのネットワーク挙動を特徴付ける方法論

Paul Watrobski

ムルギア・スパヤ

ジョシュア・クロスターマン

ウィリアム・バーカー

本出版物は以下から無料で入手可能である：
https://doi.org/10.6028/NIST.IR.8349

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST 内部報告書

NIST IR 8349

モノのインターネット（IoT）デバイスのネットワーク挙動を特徴付ける方法論

ポール・ワトロブスキー

応用サイバーセキュリティディビジョン

情報技術研究所

ジョシュア・クロスターマン

MITRE 社

ムルギア・ソウッパヤ

コンピュータセキュリティディビジョン

情報技術研究所

ウィリアム・バーカー

ストラトビア合同会社

本出版物は以下から無料で入手可能である：

https://doi.org/10.6028/NIST.IR.8349 2025年 8 月

米国商務省

ハワード・ラトニック長官

国立標準技術研究所

クレイグ・バークハート、商務省標準技術担当次官代理兼国立標準技術研究所（*NIST*）所長代理

本論文では、実験手順を適切に特定するため、特定の機器、装置、ソフトウェア、または材料（商用・非商用を問わず）を識別している。こうした識別は、NIST による製品またはサービスの推奨・承認を意味するものではなく、また、識別された材料や機器が当該目的に必ずしも最適であることを示唆するものでもない。

本刊行物には、NIST が法定の責務に基づき現在開発中の他の刊行物への言及が含まれる場合がある。本刊行物の情報（概念や方法論を含む）は、関連刊行物が完成する前であっても連邦機関によって使用される可能性がある。したがって、各刊行物が完成するまでは、現行の要件、ガイドライン、手順（存在するもの）が引き続き有効である。計画および移行の目的上、連邦機関は NIST によるこれらの新規刊行物の開発状況を注視することが望ましい。

組織は、公開コメント期間中にすべてのドラフト出版物を確認し、NIST にフィードバックを提供することが推奨される。上記以外の多くの NIST サイバーセキュリティ出版物は、https://csrc.nist.gov/publications で入手可能である。

NIST 技術シリーズ方針

著作権、使用、およびライセンスに関する声明

NIST 技術シリーズ刊行物識別構文

刊行物履歴

NIST 編集審査委員会により 2025 年 8 月 4 日に承認

本 NIST 技術シリーズ刊行物の引用方法

Watrobski P, Souppaya M, Klosterman J, Barker W (2025) ネットワーク上のモノのインターネットデバイスの動作特性を評価する手法

(国立標準技術研究所、メリーランド州ゲイサーズバーグ)、NIST 内部報告書(IR) NIST IR 8349。

報告書 (IR) NIST IR 8349. https://doi.org/10.6028/NIST.IR.8349

著者 ORCID iD

Paul Watrobski: 0000-0002-6449-3030

ムルギア・ソウパヤ: 0000-0002-8055-8527 ジョシュア・クロスターマン: 0000-0002-0026-5522

ウィリアム・バーカー: 0000-0002-4113-8861

連絡先情報 iot-ddos-nccoe@nist.gov

国立標準技術研究所

宛先：応用サイバーセキュリティディビジョン、情報技術研究所

100 Bureau Drive (Mail Stop 2000) Gaithersburg, MD 20899-2000

追加情報

本出版物に関する追加情報（関連コンテンツ、更新情報、文書履歴など）は、https://csrc.nist.gov/pubs/ir/8349/final でご覧いただけます。

すべてのコメントは情報公開法（FOIA）に基づき公開される可能性がある。

要約

本報告書は、IoT デバイスのネットワークコミュニケーション動作を捕捉・文書化する手法について記述する。この情報に基づき、製造事業者、ネットワーク管理者等は、Manufacturer Usage Description（MUD）仕様に基づくファイルを作成・利用し、当該 IoT デバイスへのアクセスおよび当該デバイスからのアクセスを管理できる。また、本報告書では、本手法の現状の実装状況と将来の開発に向けた提案についても述べる。

キーワード

アクセス管理; デバイス特性評価; モノのインターネット（IoT）; 製造事業者使用説明（MUD）; ネットワークコミュニケーション

コンピュータシステム技術に関する報告書

米国国立標準技術研究所（NIST）の情報技術研究所（ITL）は、米国の測定・標準インフラにおける技術的リーダーシップの提供を通じて、米国経済と公共の福祉を促進している。ITL は、情報技術の開発と生産的利用を推進するため、試験、試験方法、基準データ、概念実証実装、技術分析を開発している。ITL の責任範囲には、連邦情報システムにおける国家安全保障関連以外の情報の費用対効果の高いセキュリティとプライバシーを確保するための、管理・運営・技術・物理的標準およびガイドラインの開発が含まれる。

補足情報

NIST の国立サイバーセキュリティ・センター・オブ・エクセレンス（NCCoE）は、MUD ファイル開発を支援する概念実証として開発されたオープンソースツール「MUD-PD」を作成した。このツールの詳細はセクション 3.2 で説明されている。GitHub 参照：https://www.github.com/usnistgov/MUD-PD。

対象読者

本報告書は、MUD ファイルの構築、作成、または利用を希望する以下の対象者向けに作成されている：

- MUD ファイルを生成する IoT デバイスメーカーおよび開発者
- MUD ファイルの展開を行うネットワーク管理者；
- IoT デバイスの脆弱性研究者およびアナリスト；
- ネットワーク機器の開発者および製造事業者；ならびに
- MUD 仕様に基づくコンポーネントを開発・利用するサービスプロバイダ

文書の慣例

本報告書では、文献において矛盾した定義や一般的な定義が存在する複数の用語を使用している。本稿の目的上、以下の定義を新たに作成または採用した：

特性化とは、デバイスに関連する動作や特性を記述するために使用される情報を収集、分析、および／または保存する行為を指す。こうした特性には、物理的特性やサポートされているコミュニケーション技術／プロトコルなどが含まれる。

フィンガープリンティングとは、デバイスタイプを一意に識別するのに役立つ情報を収集する行為である。

モノのインターネット（IoT）デバイスとは、物理世界とデジタル世界の双方と直接相互作用できるデバイスを指す。IoT デバイスにはセンサー、コントローラー、家電製品などが含まれる。NIST IR 8259 では、IoT デバイスを「物

理世界と直接相互作用するための少なくとも 1 つの変換器（センサーまたはアクチュエーター）と、デジタル世界と接続するための少なくとも 1 つのネットワークインターフェース（例：イーサネット、Wi-Fi、Bluetooth、Long-Term Evolution [LTE]、Zigbee、Ultra-Wideband [UWB]）を備えたデバイス」と定義している[1]。

MUD ファイルには、MUD 仕様書[2]で説明されているように、IoT デバイスとそのネットワーク動作を記述する情報が含まれる。「MUD プロファイル」という用語は既存文献で広く使用され、「MUD ファイル」と同義である。本論文では MUD 仕様書で定義された「MUD ファイル」の用法に従う。

MUD ファイルの正確性とは、MUD ファイルが IoT デバイスの全コミュニケーション要件をどの程度正確に捕捉しているかを示す。特に、デバイスが意図された機能を果たすために必要となる可能性のある全コミュニケーションをリストアップしている範囲（網羅性）と、デバイスが不要なコミュニケーションをリストアップすることを回避している程度（正確性）を指す。

（網羅性）と、デバイスが不要なコミュニケーションをリストから除外する程度（正確性）の両面から評価される。なお、デバイスの製造事業者が作成したファイルであっても、MUD ファイルの完全な正確性を保証することは不可能である場合がある。一部のデバイスでは、デバイスの動作に影響を与え得るあらゆる状況やネットワーク構成をテストすることは非現実的、あるいは不可能な場合がある。さらに、そうした状況によって明らかになる可能性のある潜在的なコミュニケーション要件が未知のままである場合もある。

商標情報

すべての登録商標または商標は、それぞれの組織に帰属する。

特許開示通知

注意：ITL は、本出版物のガイダンスまたは要件への準拠に必要な特許クレームの保有者に対し、当該クレームを ITL に開示すよう要請している。ただし、特許保有者は ITL の特許開示要請に対応する義務はなく、ITL は本出版物に適用される可能性のある特許を識別するための特許調査を実施していない。

本出版物の発行日時点および本出版物の指針または要件の遵守に必要な特許クレームの識別を求める要請後においても、ITL に対してそのような特許クレームは識別されていない。

ITL は、本出版物の使用において特許侵害を回避するためにライセンスが不要であることを表明または示唆するものではない。

目次

エグゼクティブサマリー 1

1. 序論 2

1.1. IoT デバイスとそのネットワークのセキュリティ確保における課題 2

1.2. 目的と範囲 3

1.3. 報告書の構成 4

2. ネットワークトラフィックキャプチャ手法 5

2.1. キャプチャ戦略 5

2.1.1. IoT デバイスのライフサイクル段階 5

2.1.1.1. セットアップ 5

2.1.1.2. 通常動作 5

2.1.1.3. 廃止/撤去 6

2.1.2. 環境変数 6

2.1.3. アクティビティベースおよびタイムベースのキャプチャ手法 7

2.1.4. ネットワークアーキテクチャとキャプチャ手法 8

2.1.5. キャプチャツール 8

2.1.5.1. tcpdump 8

2.1.5.2. Wireshark/tshark 9

2.2. キャプチャ手順 9

2.2.1. アクティビティベースのキャプチャ 9

2.2.2. 時間ベースのキャプチャ 10

2.3. ドキュメント戦略 10

3. 分析ユースケースとツール 13

3.1. 手動による MUD ファイル生成 (生成式的) 13

3.1.1. Wireshark 13

3.1.2. NetworkMiner 13

3.1.3. 手動による MUD ファイル生成プロセスの概要 13

3.2. MUD-PD 14

3.2.1. MUD-PD の機能セット 15

3.2.2. MUD-PD の用途 16

3.3. プライバシー分析のための MUD-PD サポート 16
4. 今後の研究課題 17
4.1. MUD-PD 機能の拡張 17
4.2. MUD パイプラインの開発 17
4.3. コミュニティへの未解決課題 20
参考文献 21
附属書 A. キャプチャ環境の例 23
附属書 B. 記号・略語・頭字語一覧 24
附属書 C. MUD-PD ツール GUI の概要 25
C.1. 新規パケットキャプチャの輸入事業者 27
C.2. デバイスの表示と輸入事業者 28
C.3. デバイスレポートの生成 (生成的) 30
C.4. MUD ファイルの生成 (生成的) 31

図の一覧

図 1. デバイスメーカーまたは開発者向けユースケースにおける MUD パイプライン 18
図 2. ネットワーク管理者ユースケース向け MUD パイプライン 18
図 3. 特に研究開発に活用される包括的な MUD パイプライン 19
図 4. キャプチャ構成例 23
図 5. ボタンとリストボックスがラベル付けされた MUD-PD メインウィンドウ 25
図 6. Fingerbank API キープロバイダプロンプト 26
図 7. 新規データベース作成プロンプト 26
図 8. 既存データベース接続プロンプト 26
図 9. パケットキャプチャをデータベースに輸入事業者するためのプロンプト 28
図 10. パケットキャプチャ輸入事業者処理中のラベル付き/未ラベルデバイス一覧ウィンドウ 29
図 11. デバイス輸入事業者のウィンドウプロンプト 29
図 12. データベースに記録されたファームウェアバージョンを更新するプロンプト 30
図 13. 人間が読めるデバイスレポート生成の生成的プロンプト 30
図 14. 単一パケットキャプチャの詳細を示すデバイスレポート例 31
図 16. サポート URL およびドキュメント URL を含むデバイス詳細入力プロンプト 32

図 17. インターネットコミュニケーションルールのプロバイダ入力プロンプト 32
図 18. ローカルコミュニケーションルールのプロバイダプロンプト 33
図 19. 生成予定の MUD ファイルのプレビュー 34

エグゼクティブサマリー

モノのインターネット（IoT）デバイスの期待されるネットワーク動作を特徴づけ理解することは、サイバーセキュリティにとって不可欠である。これにより、デバイスとそれが展開されるネットワークを保護するための適切なネットワークアクセス制御（例：ファイアウォールルールやアクセス制御リスト）の実装が可能となる。これには、デバイスのコミュニケーションを必要最小限のものに制限することも含まれる。また、デバイスが不正動作している可能性（侵害の兆候）を識別することも可能となる。IoT デバイスのネットワークコミュニケーションを制限する能力は、特にこれらのデバイスの増加を考慮すると極めて重要である。

大半の IoT デバイスのネットワーク動作は状況に依存する。例えば、多くの IoT デバイスは音声コマンド、人との物理的インタラクション、他のデバイス（スマートフォンや IoT ハブなど）、サービス（クラウドベースなど）といった複数のメカニズムを通じて操作・制御される。これらの各メカニズムは、同じ結果（例：音声コマンド、モバイルアプリ、物理的なスイッチ操作による電球の点灯）を達成する場合でも、異なるネットワーク動作を引き起こす可能性がある。さらに、特定のネットワーク動作パターンは、デバイスのライフサイクルの特定の段階（例：セットアップ、通常動作、廃棄）でのみ発生する場合がある。また、デバイスソフトウェアの更新に伴い、ネットワーク動作は時間とともに変化する可能性がある。これらの理由から、デバイスの想定されるネットワーク動作は、すべての想定シナリオおよびライフサイクルの各段階において特性化され、理解される必要がある。そうしなければ、デバイス設定、運用、または廃棄に必要な手順がネットワークアクセス管理によって妨げられ、完全な実行または実行自体が不可能になる可能性がある。

本出版物は、様々なユースケースや条件下における IoT デバイスのネットワーク動作の全範囲を正確に捕捉、文書化、特性化する推奨手法を説明する。この方法論を用いることで、IoT デバイスメーカーや開発者、ネットワーク事業者や管理者、クラウドプロバイダ、研究者は、Manufacturer Usage Description（MUD）に準拠したファイルを生成できる。MUD は、IoT デバイスが意図された機能を実行するために必要なネットワークコミュニケーションを規定する標準を提供する。MUD ファイルは、IoT デバイスを利用する組織に対し、各デバイスに適用すべきアクセス管理ルールを指示す。また、様々なセキュリティ技術によって自動的に消費・利用されることが可能である。MUD ファイルは特定のネットワーク展開に合わせて拡張できる。ネットワーク事業者、ネットワーク管理者、クラウドプロバイダは、セキュリティツールで収集した環境ベースのネットワークプロファイルと組み合わせて、デフォルトまたはカスタムの MUD ファイルを展開し、個々のデバイスだけでなくネットワーク全体を保護できる。

本出版物では、NIST 国立サイバーセキュリティ・センター・オブ・エクセレンス（NCCoE）が開発したオープンソースツール「MUD-PD」も紹介する。これは IoT デバイスの特性評価とそれに続く MUD ファイルの作成を自動化することを支援する。本ツールは収集データの分類・分析に加え、デバイスに関するレポートと展開可能な MUD ファイルの両方を生成するために使用できる。

1. 序論

米国国立標準技術研究所（NIST）傘下の国立サイバーセキュリティ・センター・オブ・エクセレンス（NCCoE）は、ネットワーク管理者および IoT ネットワーク運用者が、IoT デバイスのネットワークコミュニケーション要件を識別・理解・文書化する能力の向上に取り組んでいる。デバイスの種類や通信挙動を文書化することで、製造事業者使用説明（MUD）仕様に基づくファイルの作成が可能となり、ネットワーク管理者はこれを利用して当該デバイスへのアクセス管理を実施できる[2]。

「文書規約」セクションでは、本報告書全体で使用されるいくつかの用語を定義している。読者は先に進む前にこれらの定義を確認すること。

1.1. IoT デバイスとそのネットワークのセキュリティ確保における課題

ネットワークを適切に保護するには、ネットワーク管理者は、ネットワーク上に存在するデバイスと、各デバイスが意図した機能を実行するために必要なネットワークコミュニケーションを理解する必要がある。IoT デバイスを含むネットワークの場合、個々のデバイスを識別すること自体が困難な上、各デバイスが他のネットワーク構成要素に対して必要とするネットワークアクセスや、他のネットワーク構成要素が各デバイスに対して必要とするアクセスを把握することはさらに困難である。この課題に対処するため、多くの組織ではネットワーク上のデバイスタイプを識別する IoT デバイスフィンガープリンティングおよび特性評価手法を導入している。

各デバイスの IoT デバイス種別が判明すれば、ネットワーク管理者はそれらのセキュリティとネットワークアクセス管理を開始できる[3]。これにはデバイス特性や動作に関する情報の収集が含まれる。プリンストン大学 IoT インスペクター[4]や ProfilIoT の機械学習活用[5]といった手法が IoT デバイスの特性評価・識別に応用されており、各デバイスに関連するセキュリティ・プライバシー問題の洞察を提供可能である。ただし、全てのフィンガープリンティング・特性評価スキームが同等というわけではない。これらのスキームは、ネットワークトラフィックから導出された限定的なデータセットに基づいて構築されることが多く、デバイス種別の正確な識別のみを可能とする。これらの手法開発に用いられるネットワークトラフィック情報には、パケットヘッダー、ネットワークポート、パケットタイミング、ハンドシェイク、特定 IoT デバイス固有のその他の情報などが含まれる[6][7]。指紋開発に用いられるデータセットが限定的であるため、指紋にはデバイスの潜在的な行動範囲全体を判断するのに必要な情報が必ずしも含まれていない。本論文では、デバイスの行動をより包括的に特徴付ける手法を説明する。

IoT デバイスのネットワーク動作を包括的に特徴付けることは、いくつかの要因により困難である。例えば、IoT デバイスは動作に影響を与える内部変更を受けやすい。これらの変更はソフトウェア更新、ファームウェア更新、新規または追加ハードウェアによって引き起こされる可能性がある。ハードウェアの交換、他の IoT デバイスとの統合、新規ネットワークへの接続など、外部要因による変化も発生し得る。こうした変化は IoT デバイスの動作追跡の複雑さを増大させ、ひいては IoT デバイスの正確な特性評価の難易度を高める。ユーザーの活動も IoT デバイスの動作に大きく影響する。例えば、同一製造事業者製の同一カメラでも、使用目的が異なれば動作が著しく異なる場合がある。さらに、同一デバイスでもファームウェアやハードウェアのバージョンが異なれば動作が異なることもある。多くの IoT デバイスは既存デバイスの設計を基にバリエーションとして作成されるため、技術的には異なるデバイスであっても動作が類似して見えることがある。

MUD 仕様[2]の目的は、IoT デバイスが「正常に機能するために必要なアクセス権限とネットワーク機能をネットワークに通知する」標準プロバイダを提供することである。これは MUD ファイルを使用することで実現され、ネットワーク事業者、ネットワーク管理者、クラウドプロバイダが IoT デバイスに適用すべきアクセス管理ルールを把握できるように

する。しかし、ネットワーク事業者、ネットワーク管理者、またはクラウドプロバイダが不正確な MUD ファイルに基づいて IoT デバイスを展開した場合、デバイスの機能が著しく損なわれたり、脆弱性を引き起こす可能性がある。したがって、あらゆる MUD ファイルは可能な限り正確であることが不可欠である。

MUD ファイルの正確性は、以下の 2 つの概念に基づく：包括性――デバイスが意図された機能を実行するために必要となる可能性のある全てのネットワークコミュニケーションをリストアップする程度、および正確性――デバイスが不要なネットワークコミュニケーションをリストアップしない程度。ただし、製造事業者がデバイスが使用される全ての運用環境を予測できない可能性があるため、製造事業者プロバイダによる MUD ファイルが全て包括的である保証はない。デバイスが実行できるアクションの最終決定権は、デバイスの実装を担当するローカルネットワーク管理者[2]に帰属する。彼らは、デバイスの MUD ファイルを製造事業者プロバイダのものより制限的に、あるいは緩和的に設定することを決定できる場合がある。さらに、ネットワーク管理者は、製造事業者プロバイダの MUD ファイルがないデバイス用に独自に MUD ファイルを作成したいと考えるかもしれない。

1.2. 目的と範囲

MUD は IoT デバイスの動作を特徴付ける唯一の方法ではないが、デバイスの意図を文書化することで、IoT デバイスとネットワークの防御のための多くの選択肢とメカニズムをサポートする。本報告書は、IoT デバイスの潜在的なネットワーク動作に関する情報を明らかにするネットワークトラフィックデータに基づいて、正確な MUD ファイルを構築する方法を説明する。MUD ファイルの開発は、トラフィックキャプチャとトラフィック分析という 2 つの主要なステップから構成される。本報告書で説明する手法は、IoT デバイスの潜在的な動作を可能な限り多く捕捉し、正確なネットワークトラフィックデータセットを作成することを目的としている。この手法は、合理的に予測可能な IoT デバイスのネットワークトラフィック動作の全範囲を分析できるようにすることを目指す。これには、デバイスのライフサイクルの各段階で IoT デバイスの動作を変化させる可能性のある様々な要因の検証が含まれる。

製造事業者、開発者、ネットワーク事業者・管理者、クラウドプロバイダ、研究者は、この手法を活用して包括的なデータセットを開発できる。このデータセットは、MUD ファイルの生成、セキュリティ・プライバシー問題の調査、機械学習アルゴリズムの開発などに利用可能である。本手法はインターネットプロトコル（IP）ベースのネットワークで開発されましたが、他の種類のネットワークでも利用可能である。なお、この種の分析では以下の前提条件が重要である：

- IoT デバイスが分析プロセスのいかなる時点でも悪意ある主体による改ざんや侵害を受けていないこと、および
- IoT デバイスが製造事業者の意図通りに動作していること。

本報告書では、ネットワーク上の IoT デバイスの動作を捕捉する手法を提示すだけでなく、NCCoE が開発した MUD-PD ツールが、この動作情報を活用して MUD ファイルを作成する方法についても考察する。MUD-PD は正確な MUD ファイルを生成するため、多様なネットワークトラフィックキャプチャを必要とする。本ツールは関連情報を抽出・集約することで、大量のネットワークトラフィックデータを手動解析することなく正確な MUD ファイルの作成を可能にする。これにより、手動での MUD ファイル作成と比較して、生成に必要な時間と労力を大幅に削減できる。

MUD ファイルから生成されたルールの適用は、本報告書の範囲外である。NCCoE 実践ガイド特別刊行物(SP)1800-15『中小企業および家庭用 *IoT* デバイスのセキュリティ確保：製造事業者使用説明*(MUD)*を用いたネットワークベース攻撃の緩和』[8]では、いくつかの異なるアプローチが説明されている。当該プロジェクトでは、市販製品を用いた実環境ネットワークにおける MUD ベースのアクセス管理実装の実例を提供した。本報告書は、正

確な MUD ファイル開発のための構造化プロセスを提示すことで、この取り組みを支援・拡張する。これらのリソースを統合することで、IoT セキュリティ向け MUD ファイルの開発・運用基盤が構築される。

IoT デバイスの検知と識別も本報告書の対象外とする。ただし、手動によるデバイス識別と分析を支援する 2 つのツールについては、セクション 2.1.5 で説明を行う。

1.3. 報告書の構成

本報告書の残りの部分は、以下のセクションと附属書で構成される：

- セクション 2 では、トラフィックキャプチャ戦略、ツール、手順例、および文書化について論じる。
- セクション 3 では、ネットワークコミュニケーションの分析、プライバシーへの影響、および MUD ファイル生成について論じる。
- セクション 4 では、MUD-PD の機能強化や追加機能の開発、デバイス特性評価分野における継続的研究など、将来的な取り組みの可能性を探る。
- 参考文献セクションでは、本報告書全体で引用された参考文献を定義する。
- 附属書 A では、有線および無線 IoT デバイスの両方の分析をサポートするキャプチャ環境の例を示す。
- 附属書 B には略語一覧が含まれる。
- 附属書 C には、MUD-PD ツールの詳細な概要と、MUD ファイル生成への活用方法が記載されている。

2. ネットワークトラフィックキャプチャ手法

正確な MUD ファイルを適切に生成するには、各ネットワーク接続デバイスについて、意図された動作範囲を可能な限り広く反映した包括的なデータセットが必要である。ネットワークセキュリティやアクセス管理に使用される MUD ファイルの場合、生成される各ファイルが十分に正確であることが不可欠である。これにより、正当なネットワーク活動の誤った報告を防ぎ、デバイスの正常な機能に支障を来す可能性のある制限を課すことを回避できる。本節で説明する手法は、IoT デバイス分析と MUD ファイル生成に必要な情報の収集を支援するために設計されている。

この方法論はネットワークトラフィックに基づくものであり、ネットワークトラフィックからは観察できないデバイス動作は考慮していない。本方法論の範囲外で観察されたデバイス動作は、他の手段で文書化する必要がある。

2.1. キャプチャ戦略

意図した幅広いデバイス動作を捕捉するには、デバイスライフサイクル全体を通じて、多様な状態および環境条件下で IoT デバイスとのコミュニケーションを捕捉する必要がある。本節では、ネットワークトラフィック捕捉のアプローチ、戦略、およびツールについて説明する。本節に記載される情報（ライフサイクル段階、環境変数、キャプチャの種類）は、ネットワークキャプチャごとに文書化すべきである。ネットワーク上の接続デバイスに関する情報をすべて文書化することは、任意のデバイスの動作をより完全に分析するために重要である。

2.1.1. IoT デバイスのライフサイクル段階

IoT デバイスのライフサイクルを記述する分類法は様々であるが、本報告書では IoT デバイストラフィック分析のために、デバイスのライフサイクル構成要素を 3 つの広範なフェーズに分類する：セットアップ、通常運用、廃止/撤去である。

2.1.1.1. セットアップ

セットアップ段階には、IoT デバイスをネットワークに初期接続し、デバイスが完全に機能して通常動作を開始する準備が整うために必要な設定作業のすべてが含まれる。セットアップは通常、デバイスをネットワークに有線または無線で接続することから始まる。デバイスが接続されると、セットアッププロセスにはファームウェアの更新、スマートハブやスマートフォンなど他のデバイスへの接続、その他完了が必要な処理が含まれる。セットアップ動作に関する状況の大半では、製造事業者の指示に従うことで十分である場合が多いが、特定の状況下ではデバイスの動作を把握するために指示からの逸脱が必要となる場合がある（例：IoT デバイスを関連クラウドサービスに接続しない場合、製造事業者がクラウドサービス接続を前提としているデバイスにおいて特有の動作が生じる可能性がある）。一部のデバイスでは、クラウド／インターネットベースのサービスへの初回接続が必要となる場合がある。この段階では、IoT デバイスをスマートフォンや、そのデバイスを管理する予定の他のデバイス（コントローラー／スマートハブなど）に接続することも含まれる。

設定失敗時にも、製造事業者が想定したものと異なる接続挙動が生じ得る。例えば、コントローラー・スマートハブ・クラウドサービスへの接続が設定されたデバイスが、インターネット接続の欠如やポートブロックなど様々な理由で接続不能となる場合がある。

2.1.1.2. 通常動作

「通常動作」フェーズでは、IoT デバイスがセットアップされ、意図された機能を執行しているサービス寿命の大部分における動作を捕捉する。このフェーズでは、人間とデバイスの相互作用、コントローラーやスマートハブとデバイスの相互作用、クラウドサービスとデバイスの相互作用など、幅広い動作をカバーする。また、人間の操作なしに発生するデバイス主導の動作も対象となる。ソフトウェアおよびファームウェアの更新は、人間の開始または相互作用の有無にかかわらず発生し、デバイスの動作に意図した変更をもたらす可能性がある。人間が開始した更新と自動更新の両方を捕捉することが重要であるが、自動更新の捕捉はより困難な場合がある。通常動作中のその他の相互作用には、スマートフォンやクラウドベースのサービスを通じたリモート制御が含まれる。必要なリソースにアクセスできないといった通常動作の障害状況も、異常な動作を引き起こす可能性がある。必須デバイスの除去、コントローラー/スマートハブの除去、IoT デバイスのハードリセットなどの「予期せぬ」シナリオも、依然として正常動作とみなされ、同様に検証すべきである。

2.1.1.3. 廃止/撤去

IoT デバイスのライフサイクル最終段階（他での再利用または寿命終了前）には、コントローラー/スマートハブやクラウドサービスなど他のデバイスからの登録解除（廃止）およびネットワークからの除去（撤去）プロセスが含まれる。このプロセスに関する製造事業者の指示が存在する場合、可能であればキャプチャ計画プロセスの一部として含めるべきである。指示が存在しない場合、一般に推奨される廃止・撤去手順は工場出荷時設定へのリセットである。いずれの場合も、工場出荷時設定へのリセットはキャプチャ計画プロセスの一部として含めるべきである。工場出荷時設定へのリセットにより、デバイスは初期構成に戻る。（注：工場出荷時設定へのリセットプロセス中にファームウェア更新がロールバックされない場合がある。）

本報告書では、工場出荷時設定へのリセットを廃止/撤去フェーズの要素として扱う。工場出荷時設定へのリセットにより、クラウドサービスからの登録解除や IoT デバイスのネットワーク切断が行われる場合があるためである。また、IoT デバイスは事前の廃止措置なしにネットワークから撤去される場合もあるため、その他の撤去状況も対象に含めることを推奨する。デバイスが別の役割で使用される場合や新しい所有者が使用する場合、その後の操作は新規設定フェーズに該当すると扱われます。回収計画では、廃棄・撤去時のデバイス主導の動作と、人的操作によって引き起こされる動作の両方を網羅する必要がある。

2.1.2. 環境変数

IoT デバイスは、意図された動作範囲を可能な限り広く捉えるため、多様な環境条件下で検証されるべきである。例えば、IoT デバイスがインターネットへのアクセスを許可されていない場合、意図した機能を果たすために依存するコミュニケーションの一部（例：クラウドベースの製造事業者サポートサービスやネットワーク時刻サービス）を完了できない可能性がある。これにより、IoT デバイスはネットワーク上で、メーカーが当初想定または文書化したものとは異なる動作を示すことがある。セクション 1.1 で論じたように、現在、製造事業者プロバイダによる MUD ファイルがデバイスが示す可能性のある全てのコミュニケーションパターンを網羅している保証はない。例えば、サードパーティ製ライブラリの更新によりデバイスの動作が変更されている可能性がある。このようなコミュニケーションを捕捉することで、IoT デバイスのより正確な特性評価が可能となる。

本節では、セクション 2.1.1 で説明した 3 つのライフサイクルフェーズそれぞれで適用可能な環境変数の例を示す。これは完全なリストではないが、デバイスタイプや設計によっては、各変数が IoT デバイスの動作を変更する可能性がある。一貫性を保ち混乱を避けるため、これらの変数はネットワークトラフィックキャプチャプロセス全体を通じて維持され、キャプチャ開始後の追加・削除は避けるべきである。この規則には例外があり、例えばインターネット障害をエミュレートした際の動作をキャプチャする場合などが該当する。永続変数からの逸脱は明確に文書化すべきである。

- **インターネット接続なしとは**、IoT デバイスが接続されているローカルネットワークからインターネットアクセスを遮断することを指す。これにより IoT デバイスがリソースにアクセスする能力が制限される可能性がある。
- **優先 DNS サーバーのブロックとは**、優先ドメインネームシステム（DNS）サーバーがブロックされた状態でのデバイス動作をテストすることを指す。例えば、IoT デバイスが製造事業者管理の DNS サーバーに依存するよう設定されている場合、これらの DNS サーバーへのアクセスが制限されると、補償措置を講じない限り IoT デバイスの機能は低下する。
- **デバイスの分離は**、デバイスがローカルネットワーク上で単独であることを示す。つまり、IoT デバイスが正常に機能するために必要な必須のネットワークコンポーネントやその他のコミュニケーションコンポーネントを除き、他のデバイスは接続されていない。例えば、IoT デバイスがコントローラー/スマートハブやスマートフォンによって制御される必要がある場合、このデバイスもキャプチャ中に接続される可能性がある。
- **人的操作なしとは**、キャプチャ活動期間中にデバイスに対する人的操作や設定が行われていないことを意味する。キャプチャプロセス開始前に、アナリストによるデバイスの事前プログラム操作は一切行われない。
- **コントローラー/スマートハブ制御とは**、キャプチャ中にデバイスがコントローラー/スマートハブに接続済み、または接続予定であることを示す。コントローラー/スマートハブに接続された IoT デバイスは、未接続のデバイスとは通常異なる動作を示す。さらに、複数デバイスのコミュニケーションの一部または全部が、コントローラー/スマートハブとの間で送受信されるトラフィックとして観測される可能性がある。その結果、スマートハブコントローラー用に生成された MUD ファイルは、複数デバイスの MUD ファイルの集合となる可能性がある。
- **同一製造事業者は**、キャプチャ開始前に同一製造事業者のデバイスが少なくとも 1 台ネットワークに接続されていたことを意味する。同一製造事業者の IoT デバイスが 2 台存在するネットワークも考えられる。さらに、多くの製造事業者が独自の IoT「エコシステム」構築に取り組んでいる。一部の IoT デバイスは同一製造事業者の他デバイスとのコミュニケーションを前提に設計されているため、同一製造事業者の複数デバイスを接続すると、単体接続時には見られない追加の動作が明らかになる可能性がある。
- **完全なネットワーク**とは、キャプチャ開始前に IoT アプリケーションをシミュレートするのに十分な数のアクティブデバイスがローカルネットワークに接続されている状態を指す。IoT デバイスをサポートするネットワークの目的や範囲は大きく異なり、多くの場合アプリケーション依存であるため、完全なネットワークとみなすデバイスの数や種類を判断するのはアナリストの責任である。同一ネットワーク上の他のデバイスの存在は、特性評価対象の IoT デバイスの動作に影響を与える可能性がある。
- **物理環境の顕著な**変化とは、パケットキャプチャ前またはキャプチャ中に物理環境が大きく変化したことを示す。多くの IoT デバイスには、光・温度・音など物理環境の要素を追跡するセンサーが搭載されている。

2.1.3. アクティビティベースおよびタイムベースのキャプチャ手法

活動ベースのキャプチャは、特定の動作セット中のみに焦点を当てた IoT デバイスの動作を収集する。例えば、IoT デバイスのセットアップ動作をキャプチャする場合、特定の時間量を必要とせず、開始と完了はセットアッププロセスの所要時間によってのみ決定される。

時間ベースのキャプチャは、特定の時間枠における IoT デバイスの動作を捕捉することに焦点を当てる。例えば、通常の稼働状態にある 1 日全体の IoT デバイス動作をキャプチャすることで、幅広い動作（例：デバイス主導の動作）の観察と記録が可能となる。一部の動作は長期にわたってのみ観察可能である。この特性の一例として、ユーザーの動作を「学習」し、それに応じて機能を変更するデバイスが挙げられる。こうしたデバイスは、週末と平日では異

なる動作を示す可能性がある。また、学習したパターンが崩れる場合（例：休日やユーザーが長期旅行中）にも異なる動作を示すことがある。

2.1.4. ネットワークアーキテクチャとキャプチャ手法

デバイスのネットワーク行動を完全に捕捉するには、ローカルネットワーク上の全ホスト間の全ネットワークトラフィック、およびインターネット上の外部ソースとの間でローカルネットワークに出入りする全コミュニケーションを捕捉する必要がある。小規模かつ／または単純なネットワークの場合、単一のゲートウェイから直接ネットワークトラフィックを捕捉すれば十分である。ゲートウェイは外部ソースとの全コミュニケーション、およびローカルネットワーク上の全ネットワークデバイス間のコミュニケーションを受信するからである。単一ゲートウェイを使用したキャプチャ設定の例は附属書に記載されている。ネットワークトラフィックが単一ゲートウェイを経由しない大規模または複雑なネットワークでは、可能な限りネットワーク内の複数地点からトラフィックをキャプチャすることが推奨される。これらのキャプチャ地点は、関連する全トラフィックを適切に捕捉できるよう慎重に選定すべきである。

採用するキャプチャ手法は利用可能なハードウェアに依存する場合がある。キャプチャデバイスは、キャプチャした全トラフィックを保存するのに十分なリソースを必要とする。十分な処理能力、メモリ、またはストレージが不足していると、ネットワークパケットがドロップされる可能性があり、キャプチャの正確性と完全性が損なわれる恐れがある。

場合によっては、デバイスがローカルネットワークゲートウェイから監視できないコミュニケーションを可能にする追加のネットワークインターフェースを持つことがある。例えば、ZigBee ハブは ZigBee ネットワークと Wi-Fi ネットワークの両方に接続する可能性がある。理想的には、両ネットワークのトラフィックを分析のためにキャプチャすべきである。また、3G、4G、5G デバイスなど、デバイスの二次インターフェースが完全に外部ネットワークとのコミュニケーションを可能にするケースもある。このコミュニケーションも同様にキャプチャすることが理想的であるが、困難または不可能な場合がある。いずれにせよ、 ただし、デバイスが持つ可能性のある追加のネットワークインターフェースは、情報が伝送される代替経路となり得るため、文書化することが重要である。文書化手順についてはセクション 2.3 で詳細に説明する。

ネットワークキャプチャ位置を決定したら、キャプチャ方法を選択すべきである。ネットワークデバイスのリソースが十分であれば、選択したゲートウェイ/ルーター/スイッチ上で直接トラフィックをキャプチャするのが理想的である。これにより、ネットワークデバイスが管理するイーサネットポートおよび無線ラジオのいずれかまたは全てからのネットワークトラフィックをキャプチャし、取得した情報を直接保存できる。ネットワークデバイス上で直接トラフィックをキャプチャできない場合もあるが、この方法が不可能な状況では代替手段が存在する。例えば、有線 IoT デバイスにネットワークタップをインライン設置することで、目的のコミュニケーションにアクセスできる。別の代替手段として、ミラーリングポートまたはスイッチドポートアナライザ（SPAN）ポートを使用し、ポートまたは仮想 LAN からの全トラフィックを、選択したミラーポートまたは SPAN ポートで待機するキャプチャデバイスへ送信する方法がある。無線ネットワーク経由でコミュニケーションする IoT デバイスについては、プロミスキャスモードの無線ネットワークアダプタを使用することで無線トラフィックをキャプチャできる。ただし、無線キャプチャは常に理想的な選択肢とは限りません。無線トラフィックのキャプチャに干渉が避けられない場合（例：無線アイソレーションの使用による）があるためである。

2.1.5. キャプチャツール

ネットワークトラフィックをキャプチャするための様々なツールが利用可能である。最も広く使用されているのは tcpdump と Wireshark/tshark の 2 つであるが、組織はユースケースに最も適したツールを選択できる。

2.1.5.1. tcpdump

tcpdump は軽量なコマンドラインベースのツールであり、Cisco IOS、Junos OS、および多くの Linux ベースのルータやスイッチの OS で使用できる。パケットキャプチャ（pcaps）は、ネットワークトラフィックデータの保存に一般的に使用される標準的な pcap ファイル形式に保存できる。以下のコマンドは tcpdump の使用例を示している：

標準の pcap ファイル形式で保存でき、これはネットワークトラフィックデータの保存に一般的に使用される。以下のコマンドは tcpdump の使用例を示す：bash$ tcpdump -i eth0 -s0 -n -B 2000000 -w capture.pcap

- 「tcpdump」はキャプチャプログラムを開始する。
- 「-i eth0」は、インターフェース eth0 からパケットのキャプチャを開始するよう tcpdump に指示す。
- 「-s0」はスナップショット長を無制限に設定し、より大きなパケットのキャプチャを可能にする。tcpdump は通常、68 バイトを超える IPv4 パケットを切り捨てます。
- 「-n」はホスト名解決を無効化し、正常なキャプチャに必要な処理リソースとバッファリソースを削減する。
- 「-B 2000000」はオペレーティングシステムのキャプチャバッファサイズを 2,000,000 キビバイトに設定し、より多くのネットワークトラフィックをキャプチャ可能にする。ドライバやカーネルレベルでのパケット損失は依然発生し得るため、キャプチャハードウェアがタスクに十分対応できることを確認することが重要である。
- 「-w capture.pcap」はネットワークトラフィックを capture.pcap というファイルに保存する。

2.1.5.2. Wireshark/tshark

Wireshark は最も入手しやすいパケットキャプチャ・分析ツールの一つであり、オープンソースである。Wireshark はキャプチャと分析の両方にグラフィカルユーザーインターフェース（GUI）を提供する。また、tshark と呼ばれるコマンドラインベースのキャプチャユーティリティを備えており、キャプチャと分析の両機能を実行できる。tshark は多くの Linux ベースのルーターやスイッチ OS にインストール可能で、tcpdump を超える高度なパケットキャプチャ機能を追加できるが、本実験では実装されませんでした。

Wireshark は Windows、macOS、および Linux や Berkeley Software Distribution（BSD）を含む幅広い Unix および Unix 系プラットフォームでサポートされている。キャプチャツールとして Wireshark を使用する場合、可能な限り多くの関連ネットワークトラフィックをキャプチャできるように、ミラーリングポート/SPAN ポートまたはネットワークタップの設定が必要となることがよくある。Wireshark はネットワークインターフェースをプロミスキャスモードに設定することもサポートしており、これは無線ネットワークトラフィックを適切にキャプチャするためにしばしば必要となる。Wireshark/tshark は PCAP Next Generation Dump（PcapNg）ファイル形式をサポートしており、ネットワークトラフィックキャプチャにメタデータを追加することが可能である。詳細はセクション 2.3 を参照のこと。

2.2. キャプチャ手順

本節では、ネットワークトラフィックをキャプチャする手順例を列挙する。これらの例はルーターからの直接キャプチャに焦点を当てている。MUD ファイル生成に使用するデータセットには、デバイスのセットアップ、通常動作、廃止/撤去を含む、可能な限り広範なデバイス動作が含まれることが理想的である。手順は多くの状況に適用できるよう意図的に一般化されており、必要に応じて変更/カスタマイズ可能である。これらの手順が適用可能なネットワークの例については、附属書 A を参照すること。

2.2.1. アクティビティベースのキャプチャ

アクティビティベースのキャプチャには、デバイスの設定、廃棄/撤去、またはデバイスで実行されるその他のアクティビティなどのプロセスが含まれる場合がある。このキャプチャタイプの例となるプロセスは以下の通りである：

1. 本キャプチャに使用する環境変数を選択、実装、文書化する。環境変数に関する詳細はセクション 2.1.2 を参照のこと。
2. ルータでパケットキャプチャを開始する。
3. 事前に定めたデバイス操作を開始する。
4. デバイス操作を完了する。
5. パケットキャプチャを終了する。
6. パケットキャプチャファイルをルーターから外部ストレージに転送し、分析を行う。

2.2.2. 時間ベースのキャプチャ

時間ベースのキャプチャは、デバイスがユーザーアクティビティに関与していない場合も含め、より長い期間にわたるデバイスの動作に関する追加情報を提供する。このキャプチャタイプの例示的なプロセスは以下の通りである：

1. このキャプチャに使用する環境変数を選択、実装、文書化する。環境変数に関する詳細はセクション 2.1.2 を参照。
2. ルーターでパケットキャプチャを開始する。
3. 所定の時間が経過するまでデバイスの動作を継続させる。
4. パケットキャプチャを終了する。
5. パケットキャプチャファイルをルーターから外部ストレージに転送し、分析を行う。

2.3. ドキュメント戦略

各ネットワークトラフィックキャプチャ完了後、条件やその他の適用可能な詳細を徹底的に文書化し、各パケットキャプチャに関連付けることが重要である。ライフサイクルフェーズ、関連する環境変数、その他の重要な要素を文書化することは、ネットワークトラフィックのその後の分析に大いに役立ちます。この情報を記録するオプションには、ファイル名の編集、テキストドキュメントの使用、データベースへの情報保存、またはキャプチャファイル自体へのメタデータ記録が含まれる。

MUD 仕様[2]には特定の条件下でのトラフィック許可・遮断メカニズムは含まれていない点に留意されたい。しかしながら、ネットワーク管理者が特定のイベントにネットワーク活動を追跡できることは有用である。このような状況や、デバイスの動作をより深く理解するためには、各キャプチャ時の活動内容、実行されたアクション、環境変数をログとして記録することが重要である。

この情報を文書化する方法は複数存在する。最も単純なのは、各キャプチャに対して手動で説明文を作成し、テキスト文書をキャプチャデータと共に保存する方法である。この手法は拡張性に乏しく、キャプチャと文書が分離されるミスを招く可能性がある。代替手段として、PcapNg のコメントフィールドを利用する方法がある。PcapNg は libpcap フォーマットの機能を拡張する。Wireshark は pcap ファイルを PcapNg に変換でき、GUI からコメントを追加可能である。Wireshark のターミナルベースインターフェースである tshark では、ネットワークキャプチャ取得中

にコメントを含めることができる。以下のコマンドにより、キャプチャ環境と変数のテキスト説明を挿入できる。これにより情報はキャプチャ自体に格納される。

```
bash$ tshark -w capture.pcapng --capture-comment "Example comment."
```

- セクション 2.1.5.1（tcpdump）で使用した-i、-s、-n、-B オプションは、ここでも使用可能である。
- tshark キャプチャのデフォルトファイル形式は PcapNg である。
- --capture-comment オプションを使用すると、キャプチャ中にテキストコメントを追加できる。

PcapNg のコメントフィールドの使用は、依然として最適な解決策とは言えない場合がある。PcapNg は、関係者が情報を消費・利用するために追加の手動操作を必要とする点で制限がある。コメントフィールドは任意のテキスト入力が可能なため、JavaScript Object Notation（JSON）形式で情報を埋め込むことが可能である。JSON はコンピュータが解析/読み取り可能である。このため、NCCoE は必要な情報を JSON 形式でフォーマットし、PcapNg ファイルのコメントフィールドに挿入する Python ベースのツールを開発した。このツールはセクション 3.2 で説明する MUD-PD に含まれている。キャプチャ開始時に起動することも、後から挿入することも可能であるが、ツールは既存ファイルに情報を挿入する。JSON は人間が理解可能な形式であり、追加されるデータも比較的単純であるため、ユーザーは出力から必要な情報を把握できる。フォーマット例は以下の通りである：

```
{
"details": "キャプチャ詳細の例",
"lifecyclePhase": "normal operation",
"internet": true,
"humanInteraction": true,
"preferredDNS": true,
"isolated": false,
"controllerHub": false,
"mfrSame": true,
"fullNetwork": false,
"physicalChanges": false,
"durationBased": true,
"durationInteger": 60,
"durationUnits": "seconds",
"actionBased": false,
"action": ""
}
```

この形式は Python の辞書（dict）データ型に準拠しており、読み書きが容易である。したがって、辞書オブジェクト envi_vars が上記の例のように定義されている場合、パケットキャプチャファイルに次のように挿入できる：

```
python3&gt; import src.pcapng_comment
```

```
python3&gt; insert_comment(filename_in="./capture.pcap",          comment=envi_vars,
                filename_out="./capture_commented.pcapng")
```

- filename_in は既存の pcap または PcapNg キャプチャファイルである。
- comment はメタデータを含む Python 辞書オブジェクトである。
- filename_out は出力ファイル名を定義するオプション引数である。省略した場合、出力ファイル名は入力ファイル名から拡張子を削除し、末尾に「_commented.pcapng」を付加した形式になる。

本ツールは、メタデータを pcap ファイルを PcapNg に変換した後に挿入するか、PcapNg ファイルを直接コピーしたファイルに挿入できる。このツールは、セクション 3.2.2 で説明されているように、MUD-PD にグラフィカルに統合されている。

3. 分析ユースケースとツール

本節では、特性評価手法の複数のユースケースとサンプル分析ツールについて説明する。

3.1. 手動による MUD ファイル生成 (生成式的)

現在、MUD ファイルは手動で生成されることが多く、MUD Maker [9] のようなコンピュータの構文を気にせず必要な値を入力できるツールも存在するものの、ほとんどの MUD ファイルは依然として手作業で記述され、完成までに多大な労力を要する。ネットワークトラフィックのキャプチャ（セクション 2 で説明）を通じて必要なデータを取得した後、情報を抽出するには手動での分析が必要である。関連情報には、IoT デバイスがコミュニケーションしたネットワーク宛先、使用されたポートとプロトコル、デバイスの動作に関するその他のデータなどが含まれる。これは Wireshark や NetworkMiner などのネットワークトラフィック分析ツールを使用することで達成可能であり、MUD ファイルに必要な情報の抽出を可能にする。

3.1.1. Wireshark

Wireshark は、ネットワークトラフィック分析（およびセクション 2.1.5.2 で説明したパケットキャプチャ）のための著名なオープンソースツールである。Windows、macOS、Linux 上で動作可能。数百のプロトコルに対するディープパケットインスペクションをサポートしており、ユーザーがパケットバイトを精査して関連情報を抽出することを可能にする。多様な表示フィルタを用いた分析が可能で、結果は様々な形式でエクスポートできる。さらに、Wireshark は Transport Layer Security（TLS）および Wi-Fi Protected Access（WPA）/WPA2/WPA3 の復号化をサポートしている。これらの機能の組み合わせにより、セクション 2 で説明したように生成されたパケットキャプチャファイルから MUD ファイルを生成するために必要な分析が可能となる。

3.1.2. NetworkMiner

NetworkMiner は、Netresec によって構築・保守されているもう一つの人気のあるオープンソースのネットワークトラフィック分析ツールである。公式には Windows でのみサポートされているが、Mono を介して macOS でも実行できる。パケットキャプチャにも使用できるが、NetworkMiner の強みは、ネットワークトラフィックキャプチャを処理し、関連情報を迅速かつ簡単に表示すことにある。関与するネットワークホストを自動的に表示し、ファイル、画像、メッセージ、認証情報を抽出する。NetworkMiner は、ネットワークトラフィックキャプチャ全体におけるホスト間の個別セッションと DNS リクエストのリストも作成する。Wireshark のようなディープパケットインスペクション機能は備えていないが、Wireshark の深みを補完する迅速で有用なツールである。

3.1.3. 手動による MUD ファイル生成プロセスの概要

MUD ファイルの生成/開発プロセスは、一連のネットワークコミュニケーションキャプチャファイルから始まります。このセットには、セクション 2 で説明したような多様な動作が含まれていることが前提である。各ネットワークコミュニケーションキャプチャファイルに対して、以下の手順を実行できる：

1. パケットを検査し、以下を特定・記録する：
    a. IoT デバイス（送信元）アドレス（メディアアクセス制御[MAC]、IPv4/6）
    b. 宛先

i. 宛先アドレス（MAC、IPv4/6）

ii. ドメイン名

c. プロトコルとポート（伝送制御プロトコル［TCP］／ユーザーデータグラムプロトコル［UDP］）

プロトコル [UDP], IPv4/6)

i. 送信元開始型（対象の IoT デバイス） ii. 宛先開始型（対象の IoT デバイス以外のデバイス）

2. 宛先デバイスおよびサーバーを識別する：

a. デバイスの種類

b. 製造事業者

すべてのパケットキャプチャについてこの情報を収集したら、最終ステップとして情報を統合し MUD ファイルを作成する。複数のネットワークコミュニケーションキャプチャファイルに同一のデバイスやプロトコルが記録される場合や、同一デバイスが時間経過で異なる IP アドレスを割り当てられる可能性があるため、情報は一意のリストに統合すべきである。MUD ファイルでは IP アドレスは使用されないが、送信元と宛先のペアを追跡する上で記録は有用である。前述の通り、MUD ファイルの作成は簡易テキストエディタでの手動作業、または MUD Maker [9] へのテキスト入力で行える。MUD ファイルを展開する前には、手動で検証し、その内容が IoT デバイスの意図されたコミュニケーション要件および承認されたコミュニケーション要件を正確に反映していることを確認する必要がある。例えば、ドメイン名に変換されていない IP アドレスは MUD ファイルの実装を妨げる可能性があり、MUD 仕様書に記載されている通り、展開前に解決すべきである。

3.2. MUD-PD

NCCoE は、組織が自社のデバイス向けに正確な MUD ファイルを作成する際の参入障壁を低減する手法の概念実証として、オープンソースツール「MUD-PD」を開発した。MUD-PD は、Wireshark や NetworkMiner などのパケット検査ツールを用いた既存の MUD ファイル作成手法を補完するものである。現在、自動化された MUD ファイル生成には複数のアプローチが存在する。これには、トゥウェンテ大学の研究者によって考案された手法[10]、ニューサウスウェールズ大学（UNSW）が開発したオープンソースツール「MUDgee」[11]、およびインターネット技術タスクフォース（IETF）105 ハッカソンでルーカス・エスティエンヌとダニエル・イネスが開発したオープンソースツール「muddy」[12]が含まれる。

MUDgee ツールは単一のネットワークトラフィックキャプチャファイルを受け取り、観測されたネットワーク動作に基づいて MUD ファイルを生成する。MUDgee は観測された全活動が意図的かつ非悪意であると仮定する。MUD-PD における MUD ファイル生成機能の中核は当初 MUDgee を基盤として構築されたが、現在は muddy のフォークを基盤としている。別のプロジェクトである lstn/muddy は、MUD Maker の機能を再現する Python およびコマンドラインツールとして開発されました。muddy のフォークである usnistgov/muddy [13]は、lstn/muddy の豊富なコードベースを活用し、より移植性が高く統合しやすい Python オブジェクトを作成する。このフォークにより、ユーザーは任意の順序で必要なルールを入力でき、出力は IETF RFC 8520 [2]で規定された形式に従って整形される。この RFC のオプション機能（ ）は含まれていないが、コアデータとフォーマットはサポートされている。NCCoE はこの muddy のフォークを使用し、収集したネットワークトラフィックキャプチャに基づいて完全な MUD ファイルを生成している。

MUD-PD の初期バージョンでは、ネットワークキャプチャファイルの輸入事業者時にユーザーが全てのメタデータを手動で入力する必要がありました。この機能は依然として存在するが、強化されユーザーインターフェースが簡素化されている。MUD-PD が PcapNg ファイル形式をサポートするようになったため、キャプチャ環境に関する JSON 形式のデータがキャプチャファイル自体に埋め込まれ、そこから抽出可能になった。この強化により輸入事業者プロセスが簡素化され、メタデータの自動取り込みが可能になった。ネットワークキャプチャデータと文書化されたネットワーク環境メタデータを組み合わせることで、関連情報の移植性が向上すると同時に、MUD ファイル生成プロセスがより包括的かつ自動化可能となり、ユーザー入力が最小限で済むようになった。

MUD-PD はパケットキャプチャからデータを解析・抽出し、リレーショナルデータベースで整理する。GUI では、特定デバイスのネットワークコミュニケーションを検査する際、個々のパケットや任意のパケット組み合わせを検証できる。ネットワークキャプチャ中に発生した物理的な動作や活動に関するメタデータも保存されるため、ユーザーはネットワーク活動と物理世界との相関関係（例：ユーザー操作やデバイス作動に対応するネットワークトラフィック）についてより深い洞察を得られる。MUD ファイル開発を支援する探索ツールであることに加え、中核となるデータベースは任意の MySQL インターフェースを通じてクエリ可能であり、MUD-PD を超えた用途や分析を可能にする。

MUD-PD に組み込まれた追加機能には、デバイス（MAC アドレスや過去に記録されたデバイス名）と各ネットワークトラフィックキャプチャの一般的なメタデータを要約した、人間が読めるデバイスレポートの生成が含まれる。もう一つの重要な追加機能は、MUD ファイルの自動生成である。生成された MUD ファイルは、そのまま使用することも、開発者やネットワーク管理者がデバイスと MUD 対応ネットワークを防御するために必要に応じて修正することも可能である。MUD ファイルは、muddy のフォーク版へのカスタムユーザーインターフェースを通じて生成される。このインターフェースは、データベースに保存された豊富な前処理済みデータを利用しながら、muddy のデータパイプラインの強化版を活用している。

3.2.1. MUD-PD の機能セット

本節では MUD-PD の概要を説明する。詳細なツール説明は附属書 C を参照のこと。MUD-PD の主要機能は以下の 3 つである：

- **情報インポート：**最初の機能はネットワークトラフィックキャプチャのインポートである。このステップでは、キャプチャに関するメタデータを入力する機会がユーザーにプロバイダされる。ネットワークトラフィックキャプチャの輸入事業者目的は、パケットを解析し、送信元、宛先、ポート、プロトコルなどの関心のある特徴を抽出することである。この情報は MUD ファイルの中核を成する。ネットワークトラフィックキャプチャの解析と輸入事業者により、MUD-PD はローカルネットワークデバイスを識別し、それらを関心デバイスとしてラベル付けすることが可能になる。
- **データベース閲覧：**第二の機能は、データベースに輸入事業者によって輸入された関心情報のビューをユーザーに提示すことである。ユーザーは、全輸入事業者によるパケットキャプチャのリストと、選択したネットワークトラフィックキャプチャファイル内のデバイス一覧を閲覧できる。その後、デバイスまたは複数デバイスを選択し、それらから送信／受信されるパケットに関する情報を確認できる。詳細な調査のため、ユーザーは Wireshark や NetworkMiner などのパケットキャプチャ解析ツールでファイルを個別に開くことができる。
- **ファイル生成：**第三の、そして最も有用な機能は、デバイスレポートと MUD ファイルを生成することである。デバイスレポートは、そのデバイスが検出されたキャプチャを要約し、キャプチャ環境のメタデータやローカルネットワーク上でコミュニケーションしていた他のデバイスの概要を含みます。ウィザードがユーザーを導き、データベ

ース内のデータとユーザー入力から MUD ファイルを生成する。生成された MUD ファイルが運用に十分な精度を持つかどうかは、ユーザーが判断する必要がある。

3.2.2. MUD-PD の用途

MUD-PD は主に MUD を支援するツールとして設計されている。MUD の研究と展開を可能にするより大きなワークストリームの一要素となる可能性がある。特定のユースケースに依存する複数のパイプラインが存在し、それぞれについて詳細はセクション 4.2 で説明する。

MUD-PD には MUD 固有の機能があるものの、比較的用途に依存しない設計となっている。主な目的は IoT デバイス向け MUD ファイルの生成支援であるが、保存されたデータは他の目的でも様々な方法で分析可能である。データセットは必然的に大規模化かつラベル付きとなるため、機械学習技術を効果的に適用できる。機械学習と本データセットの応用例は豊富であり、予見されていない用途も含む。

セクション 3.3 で論じるように、MUD ファイル生成のために収集された同一データを用いて、これらのデバイスのプライバシー影響を検証できる。デバイスが単にどのようにコミュニケーションしているかではなく、何を通信しているか（コミュニケーション内容）を調査することで、スマートデバイスを家庭に導入することの影響について、より深い理解と認識を得ることができる。

3.3. プライバシー分析のための MUD-PD サポート

前述の通り、MUD-PD は IoT デバイス向け MUD ファイル生成以外の目的にも適用可能なツールである。MUD ファイルはデバイスの推奨動作を定義するものであり、コミュニケーションがデバイス動作の一要素であると言えるが、デバイスや関連ネットワークに付随するプライバシー影響を必ずしも捉えているわけではない。MUD-PD をプライバシー調査目的に使用する場合、NCCoE は本ツールを研究開発環境でのみ使用することを推奨する。保存データのセキュリティ保証が存在しないためである。管理されていない環境での使用は機密性の侵害につながる可能性がある。このような環境におけるプライバシーの影響を理解するには、デバイスから外部サービスへ送信されるデータ内容を把握する必要がある。デバイスや実装プロトコルによっては、ネットワークパケット内の内容が暗号化されている場合もあればされていない場合もあり（宛先 IP アドレスやポートなどのメタデータなど）、これは困難を伴う可能性がある。暗号化されている場合でも、分析対象のプロトコルは中間者攻撃（Man-in-the-Middle Attack）に脆弱であり、パケット内容の一部または全部が漏洩する可能性がある。このような攻撃手法はプライバシー調査に有用であるが、繰り返しになるが、管理された研究開発環境でのみ慎重に実施すべきである。

このような評価手法の実施には、道徳的・倫理的・プライバシーの問題が生じる可能性がある。これらの問題は、ツールの使用を管理された環境（実験室など）に限定し、NIST プライバシー枠組み[14]およびヒト被験者保護に関する共通規則[15]を遵守することで緩和すべきである。収集と記録のための同様の技術は、プライバシー調査にも有益である。つまり、潜在的にプライベートな情報がどのように送信されるかを追跡し、関与するすべてのデバイスや関係者へのリスクを追跡する。

4. 今後の研究課題

NCCoE が現在実施中の MUD 実装プロジェクトは終了した。本節では、コミュニティが実施可能なデバイス特性評価および MUD-PD に関する追加の研究開発領域について考察する。

既存の課題として、デバイス特性評価に採用される手法がセキュリティと信頼性の観点で堅牢であることを確信できるかどうかが挙げられる。代替デバイス特性評価手法（例：フィンガープリンティング）を含む追加の分析ユースケースやツールを実証・文書化することで、手法の有効性を拡大・確認できる。IoT デバイスの動作に影響を与え得る追加の状況や環境変数を特定・文書化する必要がある。キャプチャ環境変数を PcapNg ファイル内に公式オプションとして保存する機能のサポートも、パケットキャプチャ解析者のコミュニティにとって有益である。

4.1. MUD-PD 機能の拡張

NCCoE は、コミュニティの関心のあるメンバーに対し、機能強化や追加機能の開発継続を検討するよう奨励する。MUD-PD は主に概念実証（PoC）として機能することを目的としていました。既存機能の改善余地は多く、収集・記録・ファイル生成プロセスの効率化・簡素化・高速化が挙げられる。生成されるデバイスレポートの有用性も、コミュニティやユーザーからのフィードバックに基づき向上させられる。これらのレポートには、使用ポートの一覧に加え、デバイスがコミュニケーションした具体的なホストやサーバーのリストを追加できる。

MUD ファイル生成プロセス自体の使いやすさとユーザー体験に関する数々の強化も検討すべきである。これには、選択されたネットワークトラフィックキャプチャに基づいて生成される MUD ファイルの潜在的な品質について、ユーザーに大まかな推定値や警告を表示すことが含まれる。目的は、結果として得られる MUD ファイルに存在する可能性のある欠落や不備を強調することである。

MUD-PD を拡張し、キャプチャファイルから追加データを抽出・カタログ化することで、IoT デバイスのプライバシー影響を調査できるようにする。そのためには、パケットペイロード情報（文字列、画像、認証情報、証明書など）の抽出・保存が必要となる。パケットの暗号化有無、暗号アルゴリズムの種類と強度についてもログ記録する価値がある。

4.2. MUD パイプラインの開発

MUD ファイル開発に焦点を当てた追加ユースケースに対応するため、一連のパイプラインを検討できる。検討された 3 つのユースケースは以下の通り：(1) ユーザー向けに MUD ファイルを提供する必要がある製造事業者または開発者、(2) 公式 MUD ファイルまたはデバイスが当該ファイルに準拠しているかを確認し、許可された動作を拡張または変更したいネットワーク管理者、(3) 上記すべてに加え、既存の MUD ルールや提案された拡張の複雑性を調査することに関心を持つ研究者。

最初のユースケースでは、デバイスメーカーや開発者は、MUD ファイルの生成・検査・妥当性確認を容易かつ直感的に行う相互運用可能なツール群の利用が有用であると考えられる（図 1）。プロセスを開始するには、MUD Maker [9] などのツールを使用して手動で MUD ファイルを構築するか、MUD-PD や MUDgee を使用してネットワークトラフィックのキャプチャから生成するかの 2 つの選択肢がある。次のステップは、MUD Visualizer [16] を使用して視覚的に MUD ファイルを検査し、存在するべきルールが欠落していないこと、存在すべきでないルールが存在していないことを妥当性確認し、必要に応じてルールを編集することである。これらの手順を複数回繰り返した後（ ）、製造事業者は MUD ファイルに確信を持てる段階に達し、ユーザー向けに公開する可能性がある。図 1 に示したプロセスは、製造事業者提供の MUD ファイルがないデバイスの MUD ファイル生成にも利用できる。

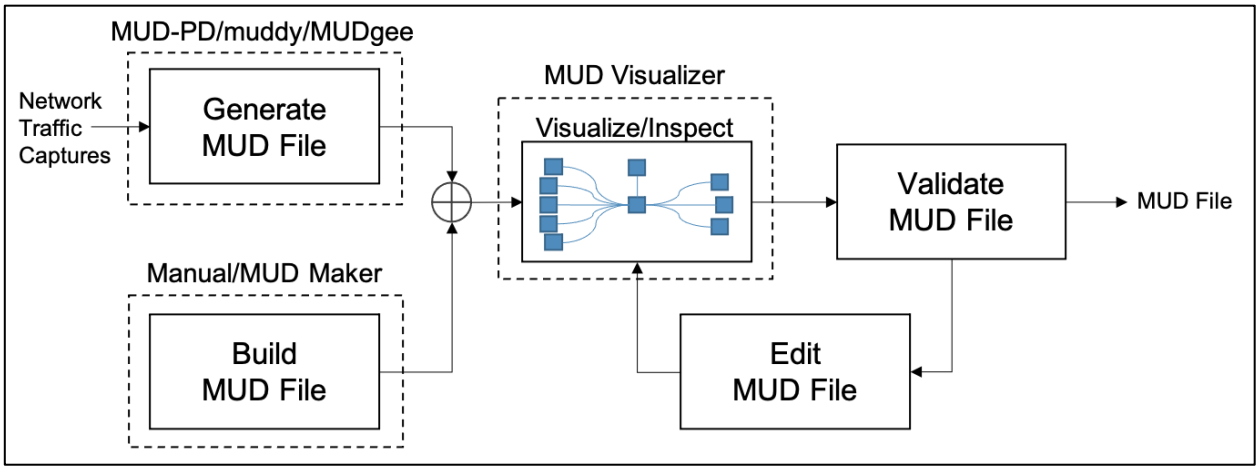

図 1. デバイスメーカーまたは開発者向けユースケースにおける MUD パイプライン

第二のユースケースでは、ネットワーク管理者が製造事業者定義の MUD ルールをオーバーレイ表示したネットワークビューを取得することが有用となる（図 2）。この機能を実現するには、MUD ファイルを取り込み、ネットワーク上で観測された動作と比較できる必要がある。MUD ファイルは製造事業者定義またはユーザー定義のいずれかである。MUD ファイルと観測された動作を検査・比較する際、ネットワーク管理者は、観測された動作が MUD ファイルに準拠していない箇所を強調表示した図表を提示される可能性がある。UNSW の研究者らは、プロバイダが提供した MUD ファイルと観測された活動を比較するツールを開発している[17]。MUD Visualizer ツールにこの機能を拡張することも想定できる。ネットワーク管理者はデバイスの機能削減や拡張（特定ネットワークへの適合）にも関心を持つため、MUD ファイルの作成・編集機能は望ましい。現在、MUD ファイルは MUD Maker を使用して構築/作成できるが、MUD ファイルを編集するための専用ツールは存在しません。ライブネットワークの管理と監視を支援するため、比較をライブネットワーク上でオンザフライで実行し、非準拠が検知された際にライブレポートや警告を発行することが有用かもしれない。

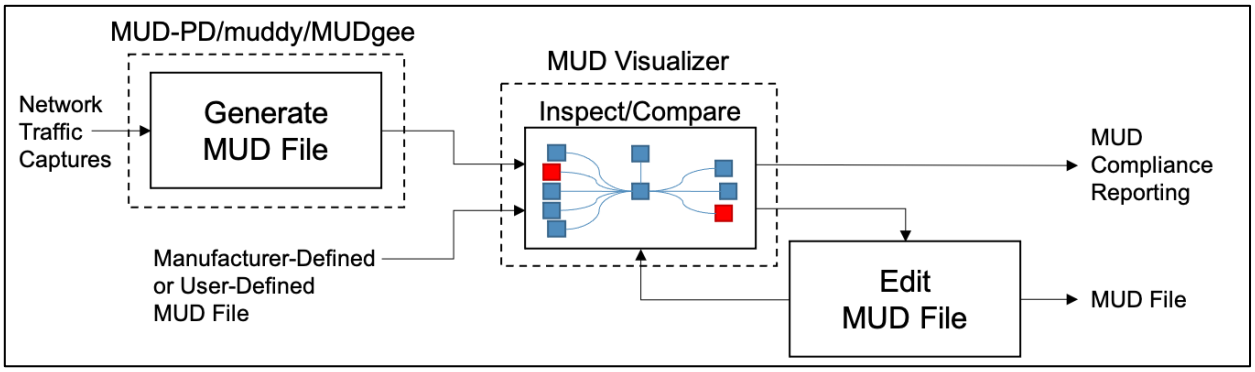

図 2. ネットワーク管理者ユースケース向け MUD パイプライン

3 つ目のユースケースはより自由度が高い。研究者は、製造事業者やネットワーク管理者に有用なツール全て、さらにはそれ以上の機能へのアクセスを望むかもしれない。既存の MUD ファイルの研究、様々な MUD ルールの影響調査、あるいは拡張機能の提供（図 3 参照）に関心を持つ可能性がある。研究者にとっては、MUD ファイルに基づくデバイスネットワークをエミュレートし、ネットワークの拡張性やデバイス間の相互作用を理解することが有用かもしれない。

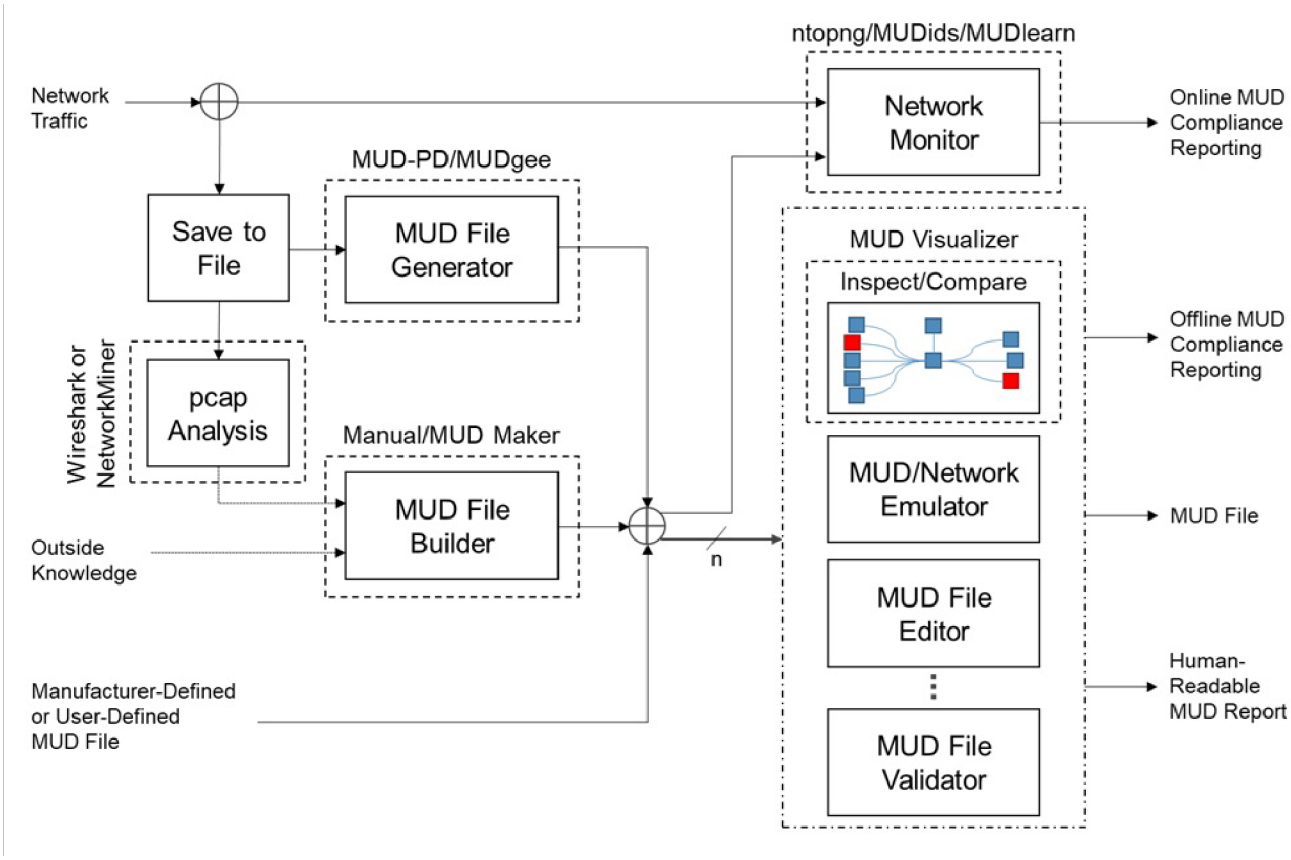

図 3. 特に研究開発に活用される包括的な MUD パイプライン

本図は、前述のユースケースにおける研究開発目標達成に向け、MUD に関連する既存および将来提案される多数のツールが如何に活用可能かを示している。図 3 の複数のボックスには、現状または将来の開発により関連役割を遂行し得る既存ツールがラベル付けされている。破線枠のないボックスは、当該役割を遂行し得る既存ツールが未関連付けの領域である。

MUD ファイルは様々な方法で生成または選択される。製造事業者提供のファイル、MUDPD や MUDgee によるネットワークキャプチャからユーザーが生成したファイル、MUD Maker や Wireshark、NetworkMiner を補助に手作業で作成したファイルなどがある。これらの MUD ファイルは複数の目的に使用されるか、様々な方法で処理される。図 3 の n が示すように、ある目的や処理では 1 つのバージョンの MUD ファイルを使用する必要がある一方、他の目的や処理では 2 つ以上のバージョンが必要となる場合がある。

ネットワーク活動のライブ分析に MUD ファイルを使用する場合、リアルタイムの MUD 準拠レポート作成が可能となる。さらに、ツール内で使用するための MUD 機能拡張が提案されている。他の MUD レポートツールの開発に関心も示されている。例えば、UNSW の研究者らは MUD をソフトウェア定義ネットワークと組み合わせて、侵入検知システムおよび ボリューム攻撃検知ツールを開発しており、いずれもリアルタイム報告の可能性がある。これらはそれぞれ MUDids および MUDlearn と呼ばれる[18][19]。MUD ファイルは、MUD Maker と対になる MUD Visualizer ツールを使用して可視化することもできる。このツールは、オフラインでの準拠性比較や手動妥当性確認のために 2 つの MUD ファイルを比較する機能へ拡張できる可能性がある。さらに、MUD ファイルの自動妥当性確認ツールや、これらのファイルに基づくネットワークエミュレーションの提案も進められている。これらのツール向けアプリケーションプログラミングインターフェース（API）の開発は、相互運用性と将来の開発を大幅に強化するだろう。

NCCoE は、IoT 製造事業者、開発者、ネットワーク管理者、研究者からなるコミュニティがこの分野の改善に引き続き貢献することを期待している。

4.3. コミュニティへの未解決課題

NCCoE は MUD-PD の機能開発を終了することを決定したが、この関心コミュニティのメンバーは、以下の未解決課題と疑問に取り組むことを推奨する：

- IoT デバイスの動作のあらゆる側面を捕捉することは不可能かもしれないため、MUD ファイルの正確性をどのように測定できるか？
    - デバイスの動作を変化させる可能性のあるその他の状況や環境変数は何か？
    - MUD ファイルの正確性を検証する方法（および不要な動作が含まれないようにする方法）は？
    - 包括的な MUD ファイルを作成するために必要なキャプチャの組み合わせは何か（許容されるべき動作が省略されないようにするには？）
- MUD-PD ツールまたはそのデータセットの他の応用例は何か？
- MUD パイプライン（または他のパイプライン）に接続するために考慮すべき他のツールは何か？
- このようなツールに望ましい機能は何であるか？
- 開発者、ネットワーク管理者、研究者にとって有用な、パケットキャプチャから抽出可能なその他の特徴は何でしょうか？
- ツールの品質と効率を改善するにはどうすればよいか？
- デバイス特性を評価する規定手法のセキュリティと信頼性を堅牢に保つにはどうすればよいであるか？
    - その有効性を客観的に実証するにはどうすればよいであるか？ 代替のデバイス特性評価手法（例：フィンガープリンティング）との比較はどうなるか？
- キャプチャ環境変数を PcapNg ファイル内に含める広範なユースケースは存在し、仕様の公式オプションとして含めるべきでしょうか？
    - そのようなオプションにはどのような環境変数を含めるべきか？

参考文献

[1] Fagan M, Megas K, Scarfone K, Smith M (2020) IoT デバイス製造事業者向け基礎的サイバーセキュリティ活動 (米国国立標準技術研究所, メリーランド州ゲイサーズバーグ). https://doi.org/10.6028/NIST.IR.8259

[2] Lear E, Droms R, Romascanu D (2019) 製造事業者 Usage Description Specification. (Internet Engineering Task Force (IETF)), IETF 意見要求 (RFC) 8520. https://doi.org/10.17487/RFC8520 Thangavelu V, Divakaran DM, Sairam R, Bhunia SS, Gurusamy M (2019) DEFT: A Distributed IoT Fingerprinting Technique. IEEE Internet of Things Journal 6(1):1-16.

[3] Thangavelu V, Divakaran DM, Sairam R, Bhunia SS, Gurusamy M (2019) DEFT: 分散型 IoT フィンガープリンティング技術. *IEEE Internet of Things Journal* 6(1):pp 940-952. https://doi.org/10.1109/JIOT.2018.2865604

[4] Huang DY, Apthorpe N, Acar G, Li F, Feamster N (2019) IoT Inspector: スマートホームデバイスからのラベル付きネットワークトラフィックを大規模にクラウドソーシング. *arXiv* 事前印刷. https://arxiv.org/abs/1909.09848

[5] Meidan Y, Bohadana M, Shabtai A, Guarnizo JD, Ochoa M, Tippenhauer NO, Elovici Y (2017) ProfilIoT: ネットワークトラフィック分析に基づく IoT デバイス識別のための機械学習アプローチ. 応用計算シンポジウム *(SAC '17)* 論文集 (ACM, モロッコ・マラケシュ), pp 506-509.https://doi.org/10.1145/3019612.3019878

[6] Bezawada B, Bachani M, Peterson J, Shirazi H, Ray I, Ray I (2018) IoT デバイスの行動指紋認証. *2018* 年ハードウェアセキュリティ攻撃と対策ワークショップ*(ASHES '18)*論文集 (ACM, カナダ・トロント), pp 41-50.https://doi.org/10.1145/3266444.3266452

[7] Aneja S, Aneja N, Islam MS (2018) ディープラーニングを用いた IoT デバイスのフィンガープリンティング. *2018 IEEE* モノのインターネットと知能システム国際会議 *(IOTAIS)* (IEEE, バリ, インドネシア), pp 174-179. https://doi.org/10.1109/IOTAIS.2018.8600824

[8] Dodson D, Montgomery D, Polk T, Ranganathan M, Souppaya M, Johnson S, Kadam A, Pratt C, Thakore D, Walker M, Lear E, Weis B, Barker W, Coclin D, Hojjati A, Wilson C, Jones T, Baykal A, Cohen D, Yelch K, Fashina Y, Grayeli P, Harrington J,クロスターマン J、ムルゲタ B、サイミントン S、シン J (2021) 特別刊行物 1800-15: 中小企業および家庭用モノのインターネット (IoT) デバイスのセキュリティ確保: 製造事業者使用説明 (MUD) を用いたネットワークベース攻撃の緩和 (国立標準技術研究所、メリーランド州ゲイサーズバーグ)。 https://doi.org/10.6028/NIST.SP.1800-15

[9] Lear E (2020) *MUD Maker Tool*. 参照先: https://mudmaker.org/mudmaker.html

[10] Schutijser CJTM (2018) MUD デバイスプロファイルを用いた自動化 DDoS 悪用防御への取り組み（トゥウェンテ大学、オランダ・エンスヘデ）。http://essay.utwente.nl/76207/

[11] Hamza A, Ranathunga D, Gharakheili HH, Roughan M, Sivaraman V (2018) Clear as MUD: IoT 行動プロファイルの生成、妥当性確認、適用. *2018* 年 *IoT* セキュリティ・プライバシーワークショップ*(IoT S&P '18)*論文集 (ACM, ブダペスト, ハンガリー), pp 8-14.https://doi.org/10.1145/3229565.3229566

[12] Estienne L, Innes D (2019) *muddy*. https://github.com/lstn/muddy で利用可能

[13] Watrobski P, Klosterman J (2021) *muddy*. Forked from *muddy* (https://github.com/lstn/muddy), Estienne L and Innes D. (国立標準技術研究所, Gaithersburg, MD).https://github.com/usnistgov/muddy で入手可能。

[14] 米国国立標準技術研究所 (2020) NIST プライバシー枠組み：エンタープライズリスクマネジメントを通じたプライバシーの改善のためのツール、バージョン 1.0。(米国国立標準技術研究所、メリーランド州ゲイサーズバーグ)。https://www.nist.gov/privacyframework で入手可能。

[15] 米国保健社会福祉省、ヒト研究防御局（2020）ヒト被験者防御に関する連邦政策（「共通規則」）。https://www.hhs.gov/ohrp/regulations-and-policy/regulations/common-rule/index.html で入手可能

[16] Andalibi V, Lear E (2020) *MUD Visualizer Tool*. 参照先: https://www.mudmaker.org/mudvisualizer.php

[17] Hamza A, Ranathunga D, Gharakheili HH, Benson TA, Roughan M, Sivaraman V (2019) MUD プロファイルを用いた IoT ネットワーク動作の検証と監視. *arXiv* 事前公開版. https://arxiv.org/abs/1902.02484

[18] Hamza A, Gharakheili HH, Sivaraman V (2018) IoT 侵入検知のための MUD ポリシーと SDN の統合.*Proceedings of the 2018 Workshop on IoT Security and Privacy (IoT S&P '18)* (ACM, Budapest, Hungary), pp 1-7. https://doi.org/10.1145/3229565.3229571

[19] ハムザ A、ガラケイリ HH、ベンソン TA、シバラマン V (2019) MUD 活動の SDN ベース監視による IoT デバイスへのボリューム攻撃の検知. *2019* 年 *ACM* シンポジウム「*SDN* 研究*(SOSR '19)*」論文集 (ACM, カリフォルニア州サンノゼ), pp 36-48.https://doi.org/10.1145/3314148.3314352

附属書 A. キャプチャ環境の例

本附属書では、有線および無線 IoT デバイスの両方の分析をサポートするキャプチャ環境の例を示す。キャプチャの手順例はセクション 2.2 で識別されている。以下のコンポーネントが例示環境を構成し、図 4 に示されている：

- 有線・無線を問わず全ネットワークトラフィックをキャプチャする tcpdump 機能を備えたホームルーター（OpenWRT を実行する Linksys WRT1900ACS）と、ホームルーターのキャプチャ保存容量を拡張するための外部ストレージ（例：フラッシュドライブ）
- Linux または macOS を実行するコンピュータ（必要に応じてキャプチャと分析の両方に使用可能）
- 特性評価対象の IoT デバイス（カメラ、スマートライト、スマート TV、スマートスイッチ）
- IoT デバイスと相互作用/コミュニケーションするその他のデバイス（スマートハブ/コントローラー/スマートフォンなど）

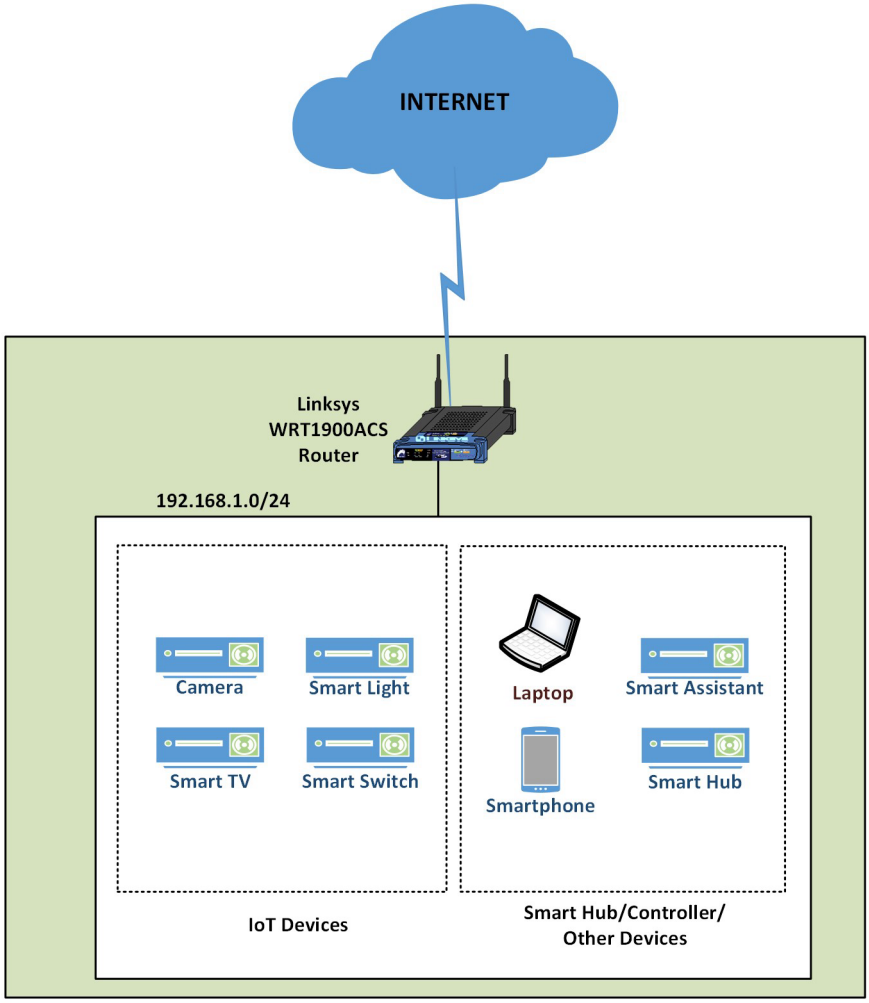

図 4. キャプチャ構成例

附属書 B. 記号・略語・頭字語一覧

API	Application Programming Interface	アプリケーションプログラミングインターフェース
BSD	Berkeley Software Distribution	バークレーソフトウェアディストリビューション
DHCP	Dynamic Host Configuration Protocol	ダイナミックホスト構成プロトコル
DNS	Domain Name System	ドメインネームシステム
E/W	East/West	東西
FOIA	Freedom of Information Act	情報公開法
GUI	Graphical User Interface	グラフィカルユーザーインターフェース
IETF	Internet Engineering Task Force	インターネット技術タスクフォース
IoT	Internet of Things	モノのインターネット
IP	Internet Protocol	インターネットプロトコル
IT	Information Technology	情報技術
ITL	Information Technology Laboratory	情報技術研究所
JSON	JavaScript Object Notation	JavaScript オブジェクト表記法
MAC	Media Access Control	メディアアクセス管理
MUD	Manufacturer Usage Description	製造事業者使用説明
NCCoE	National Cybersecurity Center of Excellence	国立サイバーセキュリティ・センター・オブ・エクセレンス
NIST	National Institute of Standards and Technology	国立標準技術研究所
N/S	North/South	南北
OUI	Organizationally Unique Identifier	組織固有識別
Pcap	Packet Capture	パケットキャプチャ
PcapNg	Packet Capture Next Generation Dump File Format	次世代パケットキャプチャダンプファイル形式
RFC	Request for Comments	意見要求
SDN	Software-Defined Networking	ソフトウェア定義ネットワーク
SPAN	Switched Port Analyzer	スイッチドポートアナライザ
TCP	Transmission Control Protocol	伝送制御プロトコル
TLS	Transport Layer Security	トランスポート層セキュリティ
UDP	User Datagram Protocol	ユーザーデータグラムプロトコル
UNSW	University of New South Wales	ニューサウスウェールズ大学
URL	Uniform Resource Locator	Uniform Resource Locator
WPA	Wi-Fi Protected Access	Wi-Fi 防御アクセス

附属書 C. MUD-PD ツール GUI の概要

本附属書では、公開時点における MUD-PD ツールの詳細な説明をプロバイダとして提供する。

MUD-PD を初めて起動すると（インストール手順は https://github.com/usnistgov/MUD-PD を参照）、MUD-PD メインウィンドウ（図 5）が表示される。図 5 内のラベルはこのウィンドウの構成要素を示している：

- (A) 既存データベースへの接続ボタン
- (B) データベースを作成・（再）初期化するボタン
- (C) キャプチャファイルを輸入事業者するボタン
- (D) MUD ファイル生成用の生成的ボタン
- (E) デバイスレポート生成(生成的)ボタン
- (F) 輸入事業者キャプチャファイルの一覧を表示すボックス
- (G) アクティブなローカルネットワークデバイスのリストを表示すボックス
- (H) コミュニケーションの一覧を表示すボックス
- (I) ボタン：以前に輸入事業者が輸入したキャプチャファイルを検査する
- (J) コミュニケーションの表示を南北（外部）トラフィックまたは東西（内部）トラフィックに制限するトグル
- (K) 後述の将来機能用トグル
- (L) コミュニケーションボックスに表示するパケット数を選択するボタン

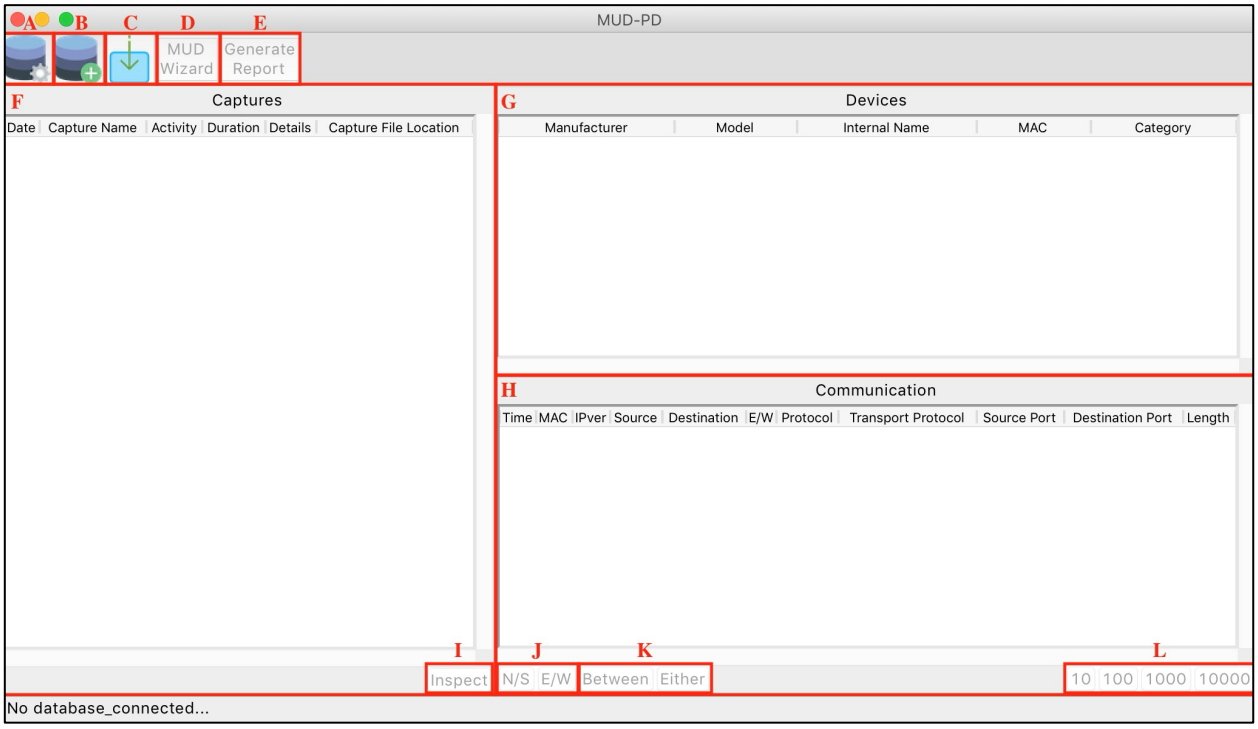

図 5. ボタンとリストボックスがラベル付けされた MUD-PD メインウィンドウ

MUD-PD を初回起動時、および終了または完了するまで、ユーザーはローカルに保存された Fingerbank API キーの任意プロバイダを促される（図 6）。Fingerbank および API キーが有効化する自動機能については附属書 C.2 で簡潔に説明する。

図 6. Fingerbank API キープロバイダプロンプト

次のステップは、B と表示されたボタンを選択し、新規データベース作成のプロンプトを開始することである（図 7）。

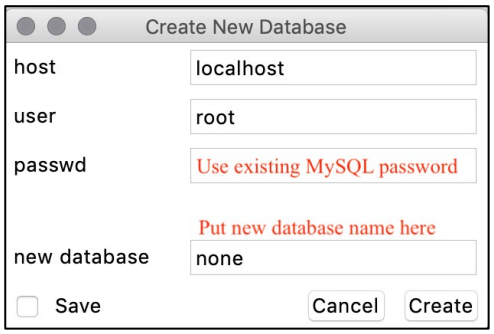

図 7. 新規データベース作成プロンプト

この時点以降、MUD-PD を実行するたびに、ユーザーは既存データベースに接続するために A と表示されたボタンを選択できる（図 5 および図 8 参照）。既存データベースに接続している場合、新規データベース作成ボタンはデータベースの再初期化にも使用でき、既存データをすべて消去する。この操作は元に戻せないため、注意して実行すること。

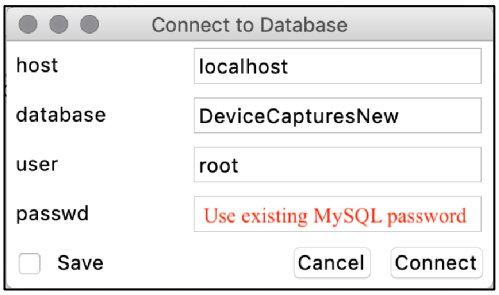

図 8. 既存データベース接続プロンプト

データベース接続後、ユーザーは内部のデータを閲覧できる。図 5 を参照すると、左側のキャプチャボックス(F)にこれまでに輸入事業者によって輸入されたパケットキャプチャファイル（pcap または PcapNg）の一覧が表示される。右上にはデバイス(G)セクションがあり、選択したキャプチャファイルでコミュニケーションしているローカルデバイスのリストが含まれる。右下の「コミュニケーション」（H）セクションには、キャプチャファイル内で選択されたデバイスが送信したパケットのリストが表示される。これらのボックスの上部には、いくつかのオプションを備えた短いツールバーがある。左から右へ、これらは次の通りである：データベースへの接続（A）、新規データベースの作成（B）、キャプチャファイルの輸入事業者（C）、MUD ファイルの生成（D）、デバイスレポートの生成（E）。

キャプチャリスト（F）には、インポートされたキャプチャファイルのメタデータ（キャプチャ時刻、キャプチャ対象イベント、キャプチャ時間（秒単位）、ファイル保存場所、インポート時に追加された詳細情報など）が表示される。リストの下には、現在選択されているパケットキャプチャを検査（I）するオプションがある。複数のキャプチャが選択されている場合、最上部に近いキャプチャのみが開かれます。パケットキャプチャを検査すると、キャプチャファイルをインポートする際に開くのと同じウィンドウが表示されるが、データベース内の詳細を更新/変更することができる。詳細はインポートプロセスと同一であり、その詳細は附属書 C.1 で詳しく説明されている。ユーザーは任意の数のキャプチャファイルを選択でき、これによりキャプチャ中にパケットを送受信したローカルデバイスをすべて表示すようにデバイスリストが変更される。

デバイス一覧(G)には、キャプチャ情報から推測可能な情報、またはインポート時にユーザーが入力した情報が含まれる。これには製造事業者、モデル、内部/ラボ用固有名、MAC アドレス、デバイスのカテゴリーが含まれる。デバイスリストのエントリ選択により、コミュニケーションボックスに表示される内容が決定される。ユーザーは「すべて…」を選択してネットワーク全体で通信された全パケットを表示すか、単一デバイスを選択してそのデバイスとのコミュニケーションのみを表示できる。複数デバイスを選択すると、選択したいずれかのデバイスとのコミュニケーションを表示できる。

コミュニケーションリスト(H)には、解析済みパケット情報（時刻、送信元 MAC アドレス、IP バージョン、送信元/宛先アドレス、トラフィック範囲、最内層プロトコル層、トランスポートプロトコル、送信元/宛先ポート、パケット長）が表示される。IP バージョンは 4 または 6 で示される。空白の場合は、パケットが IP 層（レイヤ 3）より下位層のものである。トラフィック範囲とは、値 1 で示される東西方向（内部/ローカルネットワーク）トラフィックか、値 0 で示される南北方向（外部アドレス/ネットワークとの間）トラフィックかを指す。送信元ポートと宛先ポートは TCP または UDP のポート番号である。ユーザーは南北（N/S）または東西（E/W）トラフィックでフィルタリングを選択でき、表示パケット数（J）を設定可能である。2 つのデバイスを選択した場合、追加の 2 つのボタン（K）により、両デバイス間のトラフィック、またはいずれかのデバイスに関連するトラフィックを表示できる。最後に、ユーザーは上記のフィルタ条件を満たす最初の 10、100、1000、または 10,000 パケットを表示す選択が可能である（L）。

C.1. 新規パケットキャプチャの輸入事業者

このツールの真価はパケットキャプチャファイルの輸入事業者時に発揮される。ユーザーは輸入事業者するファイルを選択するよう促される（図 9）。PcapNg ファイルの場合、MUD-PD は自動的に埋め込まれたメタデータを検索する。それ以外の場合は、キャプチャに関するメタデータ（デバイスライフサイクルのフェーズを含む）をユーザーが入力できる。ほとんどの場合、これは通常動作となる。他の 2 つのオプションはセクション 2.1.1 で説明したセットアップと除去である。ユーザーは適用されるすべての環境変数を選択することもでき、以下を含みます：

- インターネット接続が有効化されていたか
- デバイスの優先 DNS が有効化されていたか
- ・デバイスがネットワーク上で分離されていたか
- 物理環境の顕著な変化があったか
- キャプチャ対象がネットワーク全体のデバイスであったか
- コントローラーまたはハブが関与していた
- 対象となる主要デバイスと同じ製造事業者のデバイスが接続されていた
- デバイスに対する人的操作があった

キャプチャが時間ベースかアクションベースかも選択する必要がある。具体的な時間（秒単位）またはアクションを入力できる。監査可能性と使いやすさの観点から、この入力は強く推奨される。

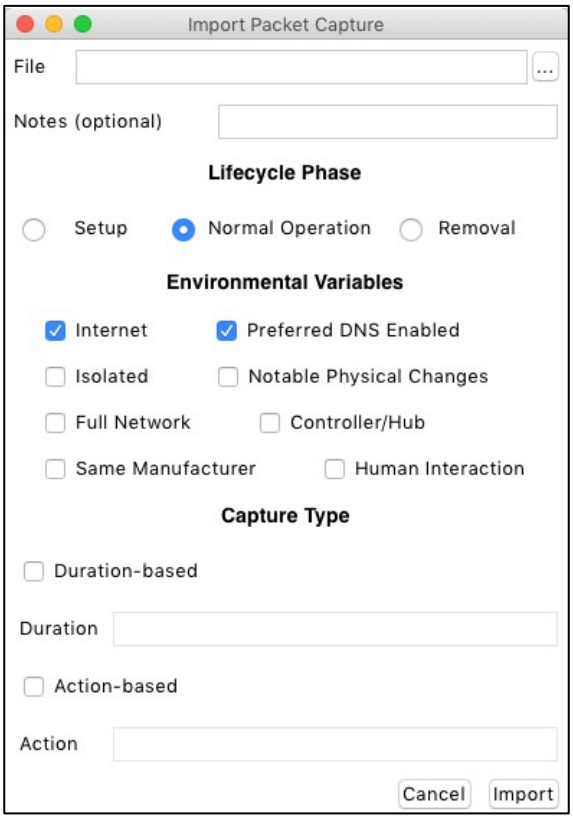

図 9. パケットキャプチャをデータベースに輸入事業者するためのプロンプト

C.2. デバイスの表示と輸入事業者

パケットキャプチャのインポートプロセス中、ユーザーにはキャプチャファイル内で検出されたラベル付きデバイスとラベルなしデバイスのリストが表示される（図 10）。ラベル付きデバイスとは、過去にインポートされたキャプチャで検出され、関連データがデータベースにインポート済みのデバイスを指す。ラベルなしデバイスは、過去のキャプチャで検出された可能性があるものの、追加データがまだインポートされていないデバイスである。このパケットキャプチャ輸入事業者ウィンドウには、キャプチャファイルから抽出されたキャプチャ日時も表示されるが、何らかの理由でどちらか一方または両方が不正確だとユーザーが判断した場合、編集が可能である。左側のリストは未ラベル付けデバイスである。MUD-PD は MAC アドレスの先頭 24 ビットである組織固有識別子（OUI）に基づき製造事業者の照会を試み、の IP アドレス（v4 と v6 の両方が利用可能な場合は両方）も表示す。ユーザーはこのリスト内の任意のデバイスを選択し、追加の詳細をデータベースに輸入事業者して右側のラベル付きデバイスリストに移動させることができる。未ラベル付けリストの情報に加え、このウィンドウにはメインウィンドウのデバイスリスト（図 5）で利用可能な全情報が含まれる。デバイスの状態（ファームウェアバージョンなど）もここで更新可能である。このフィールドは MUD-PD の自動処理では使用されませんが、MySQL 経由で照会できる。

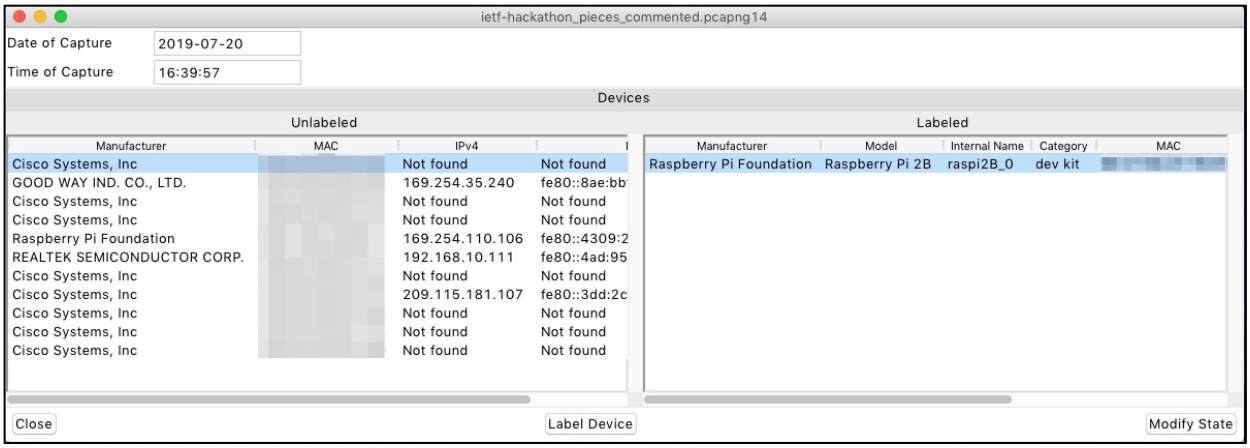

図 10. パケットキャプチャ輸入事業者処理中のラベル付き/未ラベルデバイス一覧ウィンドウ

図 10 の「デバイスにラベルを付ける」ボタンを選択すると、製造事業者、モデル、内部名、カテゴリー、備考、機能一覧を追加または変更するフィールドを備えたウィンドウが表示される（図 11）。製造事業者とモデルは必須フィールドである。内部名は必須であることに加え、一意である必要がある。デバイスのカテゴリーと備考は任意フィールドであるが、文書化や将来の分析に有用である。機能には MUD、WiFi、イーサネット、Bluetooth、ZigBee、ZWave、3G、4G、5G、その他が含まれる。MUD 以外の機能はすべてネットワークインターフェースであり、少なくとも 1 つを選択する必要がある。デバイスの MAC アドレスも表示されるが、キャプチャ自体から決定され識別子として使用されるため、変更できません。さらに、Fingerbank API との統合機能が含まれており、動的ホスト構成プロトコル（DHCP）フィンガープリントと MAC アドレスに基づいてデバイスモデルを識別することでユーザーを支援する。この機能を有効にするには、図 6 に示すように妥当性確認済みの Fingerbank API キーを取得して入力する必要がある。

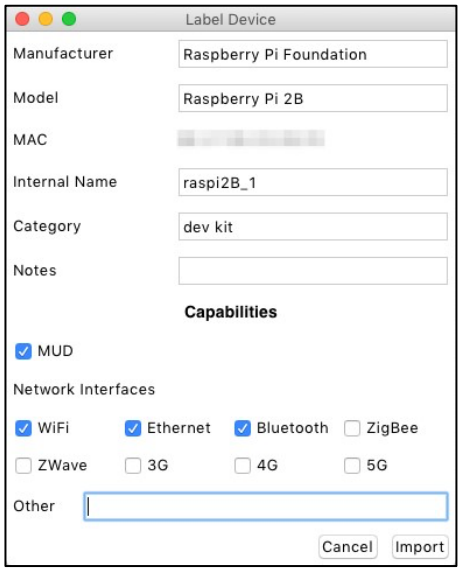

図 11. デバイス輸入事業者のウィンドウプロンプト

メタデータ入力後、図 11 の「インポート」ボタンを選択すると、デバイスのファームウェアバージョンを任意で入力するプロンプトが表示される（図 12）。現在 MUD-PD ではファームウェア情報は使用されませんが、将来の研究や分析のために記録しておく価値があるかもしれない。

図 12. データベースに記録されたファームウェアバージョンを更新するプロンプト

ファームウェアの更新情報を受信した場合、またはファームウェアバージョンを変更する必要がある場合、図 10 でラベル付きデバイスを選択し、「状態変更」ボタンをクリックすることで、このウィンドウに到達することもできる。

C.3. デバイスレポートの生成 (生成的)

デバイスレポートの生成プロセスは単純明快である（図 13）。ユーザーは任意のデバイスの組み合わせ、または単一デバイスに対してレポートを生成できる。レポート対象デバイスを選択すると、パケットキャプチャー一覧は当該デバイスが送信または受信したパケットのみに更新される。ユーザーはレポート対象として全パケットまたは任意の組み合わせを選択可能である。

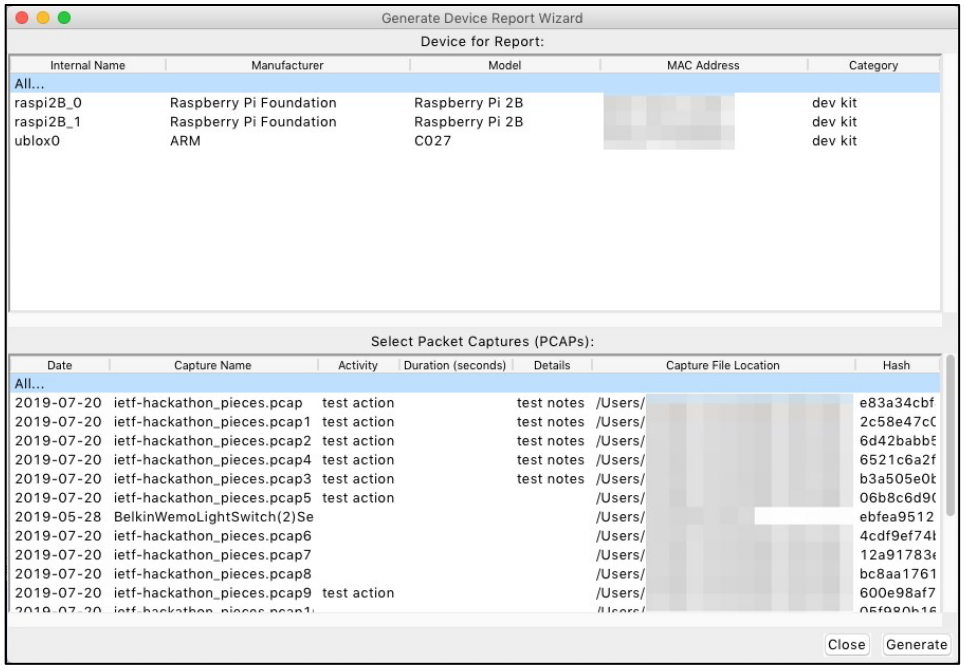

図 13. 人間が読めるデバイスレポート生成の生成的プロンプト

生成されたレポートには、デバイスが検出されたパケットキャプチャがファイルのハッシュ値と共に一覧表示される。図 14 に示すサンプルレポートには 1 つのファイル（example_file.pcap）のみが含まれているが、一般的なレポートには複数のファイルが含まれる場合がある。各ファイルのキャプチャメタデータも記載される。さらに、各キャプチャファイルの下には、キャプチャ中にネットワーク上で検出された他のローカルデバイスがリストされる。デバイスにラベルが付けられている場合は内部名も表示される。将来のバージョンでは、デバイスの MUD ファイル（存在する場合）に記載される情報と同様に、デバイスとのコミュニケーションに関するより詳細な情報がレポートに含まれる可能性がある。

```
This document serves to indicate the devices and operations captured in addition
to any specific procedures or environmental details of the captures.

Device: raspi0
MAC:    b8:27:eb:01:23:45

Capture File:  example_file.pcap
SHA256 Hash:   e83a34cbf4eab7bd8726bb9f4fce1db89b3928625c27a300d3c557ea7056466f
Device Phase:  Normal Operation
Environmental Variables:
    Internet enabled       True
    Human Interaction      False
    Preferred DNS Enabled  True
    Device Isolated        False
Action-based Capture:   False
Duration-based Capture: True
    Intended Duration:  600
    Actual Duration:    754
Start Time:     2020-02-02 12:34:56
End Time:       2020-02-02 12:47:30
Other Devices:
    Name:  router0
     MAC:  01:23:45:67:89:ab
    Name:  controller0
     MAC:  fe:dc:ba:98:76:54
    Name:  raspi1
     MAC:  d8:27:eb:67:89:ab
Notes:
        Example capture with made-up devices
```

図 14. 単一パケットキャプチャの詳細を示すデバイスレポート例

C.4. MUD ファイルの生成 (生成的)

MUD ファイルの生成には、デバイスレポートの生成よりも多くのユーザー入力と手順が必要である。ユーザーが「MUD ウィザード」ボタン（図 5 の D）をクリックすると、まず MUD ファイルを生成するデバイスを選択するよう促される（図 15）。

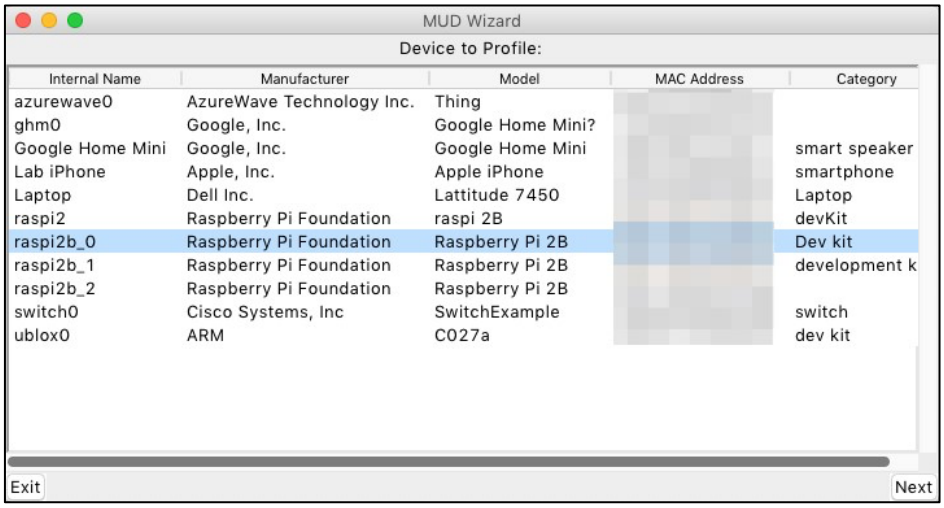

図 15. MUD ファイルを生成する対象デバイスの選択プロンプト

次のステップでは、デバイス詳細を入力する（図 16）。ここではサポート URL（MUD URL）、ドキュメント URL、デバイス説明を入力できる。さらに、MUD ファイルで定義するコミュニケーションタイプを選択可能である：インターネット、ローカル、同一製造事業者、指定製造事業者、ネットワーク定義コントローラ（my-controller）、コントローラ。該当デバイスが北南方向（N/S）または東西方向（E/W）のトラフィックを識別されている場合、インターネット／ローカルコミュニケーションが自動選択されることがある。

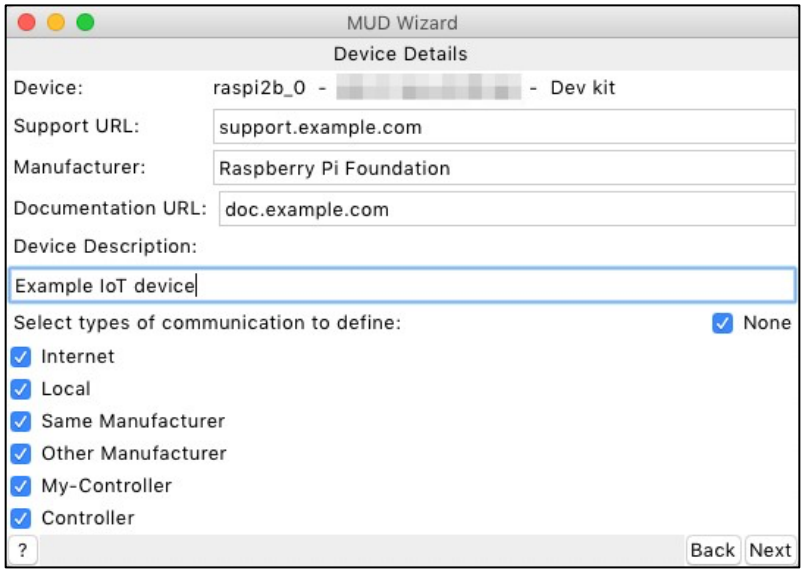

図 16. サポート URL およびドキュメント URL を含むデバイス詳細入力プロンプト

次のページに進むと、指定されたコミュニケーションタイプに対するルール一覧を定義できる（図 17）。このウィンドウのレイアウトは、6 つのコミュニケーションタイプすべてで共通である。インターネット通信およびローカル通信の場合、データベースに保存されたパケットキャプチャファイルのいずれかで、デバイスと通信したことが確認されたホストのリストが表示される。すべてのインターネットホストに対して DNS 解決が試行されるが、ローカルホストの IP アドレスは MUD 仕様[2]では使用されないため、ユーザーの参照用としてのみ表示される。

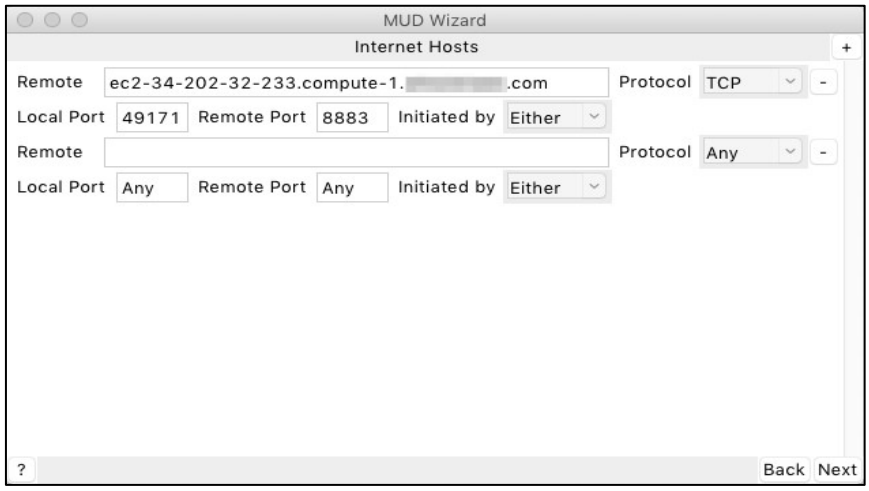

図 17. インターネットコミュニケーションルールのプロバイダ入力プロンプト

ローカルコミュニケーションルール設定ウィンドウ（図 18）は、インターネットコミュニケーションルール設定ウィンドウとは若干異なる機能を有する。監視対象のローカルトラフィックは、一般的なローカルトラフィックよりも細分化された定義が可能であるため（例：同一製造事業者やコントローラーなど他のコミュニケーションタイプに該当する場合）、これらのローカル ルールは、定義対象として選択された他の非インターネットコミュニケーションタイプへコピーまたは移動できる。残りのコミュニケーションタイプごとの各ウィンドウでも、ルールのコピーや移動が可能である。

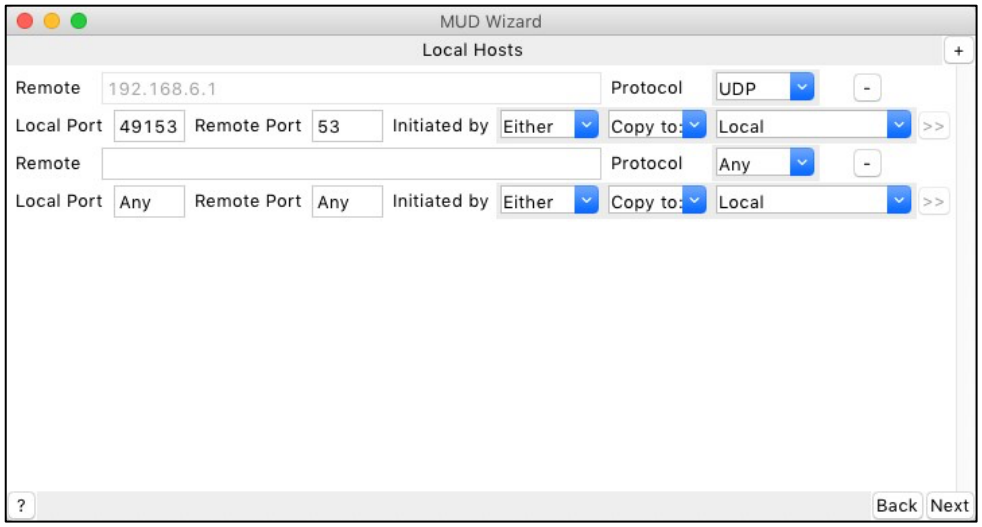

図 18. ローカルコミュニケーションルールのプロバイダプロンプト

ターゲットデバイスと通信していることが確認された各リモートホストについて、コミュニケーションプロトコル（TCP または UDP）は観測されたコミュニケーションに基づいて自動的に選択される。ローカルポートとリモートポートも観測されたコミュニケーションに基づいて自動的に入力される。ポートが空白または「any」と表示されている場合、すべてのポートが許可される。コミュニケーション開始方向は手動で指定可能（「開始元」ドロップダウンで「thing」または「remote」を選択）であり、それぞれ IoT デバイスまたはホストがコミュニケーションを開始する側となるかを示す。デフォルトでは「either」が選択され、いずれの側もコミュニケーションを開始できることを示す。

MUD 仕様[2]では、プロトコル、ポート、開始方向に関する複数の条件を定義している。「任意」プロトコルが許可されている場合（つまり TCP と UDP の両方）、ポートと開始方向は無視される。TCP が選択されている場合、ポートと開始方向を指定できる。UDP が選択されている場合、開始方向は無視され、どちらの側からもコミュニケーションを開始できる。

すべてのコミュニケーションタイプ（ローカル、同一製造事業者、ネットワーク定義コントローラ）が通信ルールでホストを定義するわけではない。これらのコミュニケーションタイプでは、リモートホストを定義する意図の入力はすべて無視される。ローカルコミュニケーションルールでは任意のローカルデバイスが指定ルールに従い、同一製造事業者では 2 ページ目に指定された製造事業者ホスト名を使用し、ネットワーク定義コントローラはローカルネットワーク管理者によって定義される。

必要なコミュニケーションタイプ規則の定義が完了すると、プロバイダは結果の MUD ファイルのプレビューを表示す（図 19）。ウィンドウ下部の「詳細モード」トグルにより、上級ユーザーはリスクを負って出力された MUD ファイルを手動修正できる。修正後の出力ファイルが正しくフォーマットまたは定義される保証はない。

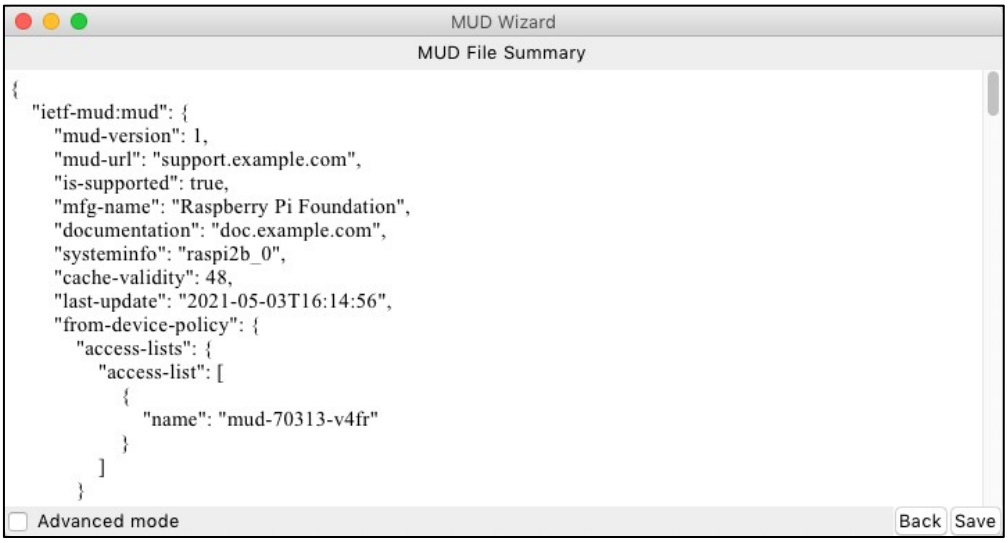

図 19. 生成予定の MUD ファイルのプレビュー