2-3-2　Low Layer セキュリティに関する研究活動の紹介

2-3-2　Overview of Our Research on Low-Layer Security

伊沢 亮一　竹久 達也　牧田 大介　三村 聡志　藤原 吉唯
ISAWA Ryoichi, TAKEHISA Tatsuya, MAKITA Daisuke, MIMURA Satoshi, and FUJIWARA Yoshitada

無線 LAN ルータやスマートウォッチなどの IoT（Internet of Things）機器は機能的ないくつかのレイヤー（階層）で構成されている。例えば、アプリケーションレイヤーやカーネルレイヤー、ハードウェアレイヤーがある。我々は階層が低いほうのレイヤーを「Low Layer（ローレイヤー）」と呼び、ローレイヤーも含めたセキュリティの研究開発することで IoT 機器全体のセキュリティ向上に資することを目的にしている。本稿では我々が実施している研究活動の中から「PPG（PhotoPlethysmoGram）センサのセキュリティ検証」と「IoT 機器に関する通信を収集するハニーポット」、「ファームウェアから SBOM（Software Bill Of Material: ソフトウェア部品表）情報を検出する研究」について紹介する。

IoT (Internet of Things) devices such as Wifi routers and smartwatches can be considered to consist of some functional abstraction layers. Examples of such layers include an application layer, a kernel layer, and a hardware layer. We research on each layer to make IoT devices more secure against cyber threats. In particular, we define low-layers as the layers lower than the application layer, and we mainly focus on low-layers as research topics. This paper briefly introduces three research topics of ours, which are security analysis for PPG (PhotoPlethysmoGram) sensors, a honeypot system that monitors malicious traffic related to IoT devices on the Internet, and a method for extracting SBOM (Software Bill Of Material) information from firmware of IoT devices, where SBOM can be used to identify vulnerabilities of IoT devices.

1　はじめに

無線 LAN ルータやスマートウォッチ、デジタルビデオレコーダーなどの多種多様な IoT（Internet of Things）機器が急速に普及し、より便利な情報化社会への発展に寄与している。一方、IoT 機器に脆弱性があると IoT 機器を利用しているユーザや IoT 機器を用いたサービスを提供している企業などがサイバー攻撃の被害にあう懸念がある。例えば、無線 LAN ルータの脆弱性を突くことで攻撃者が無線 LAN ルータを遠隔操作でき、そのネットワークからの情報の漏洩や別の機器への感染伝播活動が行われる。このようなサイバー攻撃を未然に防ぐ若しくは、被害を軽減するために IoT 機器のセキュリティ検証技術が希求されている。

我々は IoT 機器の内部を機能的な複数のレイヤー（階層）に分けて、それぞれのレイヤーの観点からセキュリティに関する検証技術を研究開発している。図 1 は我々が考える IoT 機器内部のレイヤーである。IoT 機器内部をファームウェアとハードウェア*1 のレイヤーに分け、さらにファームウェアはユーザランド（IoT 機器を管理するための Web 画面のソフトウェアなど）とカーネルランド*2（IoT 機器のオペレーティングシステム部分）で分けている。それぞれのレイヤーにおける検証技術を研究開発することで、ある固有の種類の IoT 機器だけでなく、例えば無線 LAN ルータと Web カメラのファームウェア両方を検証するために使うといった、IoT 機器の種類に極力依存しない横断的に活用できる検証技術が開発できる。その上で各レイヤーのセキュリティを担保することで IoT 機器全

*1　ハードウェアは CPU（Central Processing Unit）や I/O（入力装置）である RAM（物理メモリ）やストレージ（SSD: Solid State Drive）、NIC（ネットワークインタフェースカード）、センサーなどで構成される。

*2　カーネルランドはシステムコールインタフェースとカーネル、ドライバ、HAL（Hardware Abstraction Layer）で分けて考える。カーネルがオペレーティングシステムの中心となる役割を果たしており、ユーザランドのアプリケーションが許可されている動作のみを行うように管理するなどの役割がある。システムコールインタフェースはアプリケーションとカーネルがやりとりするために存在し、ドライバや HAL はハードウェアとの情報をやりとりするために存在する。

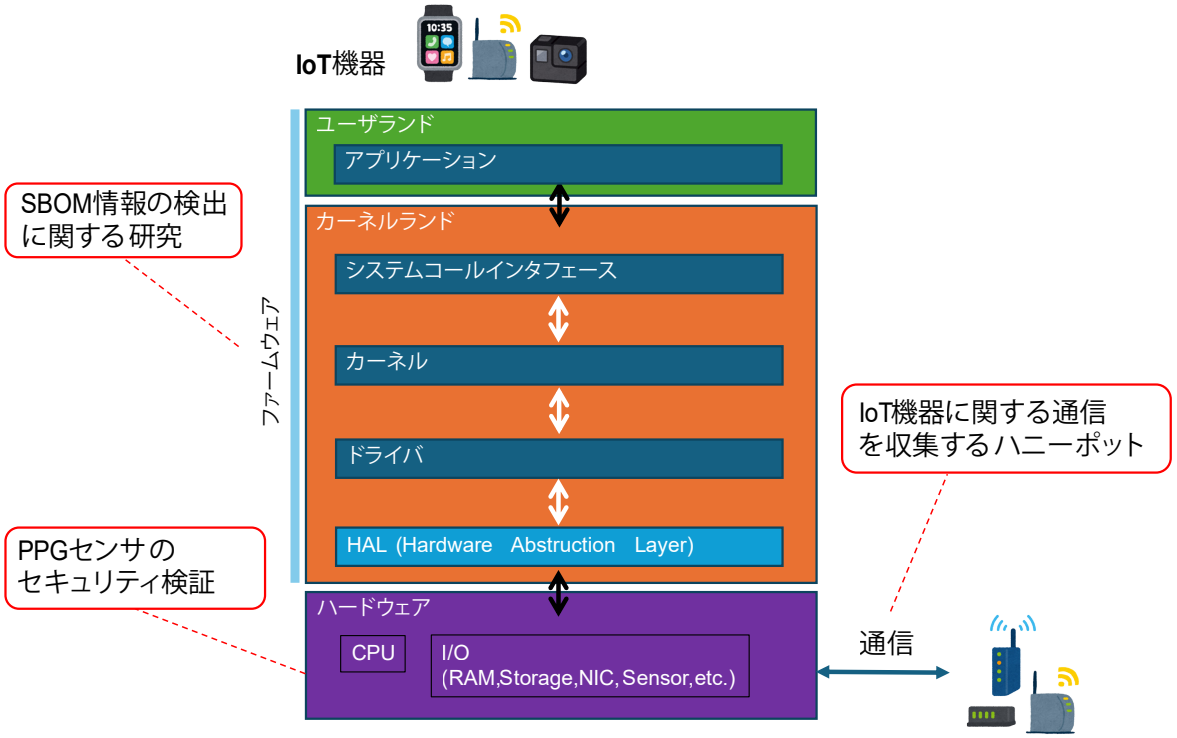

図 1　IoT 機器内部の機能的なレイヤー（赤枠は本稿で紹介する研究活動）

体のセキュアであることを担保する。特にハードウェアやカーネルランドといったユーザランドよりも低いほうのレイヤーを「ローレイヤー」と呼び、ユーザから見えるユーザランドだけでなくローレイヤーセキュリティに注力した研究開発を実施している。さらに、IoT 機器が行う通信にも脆弱性を検証するための情報が含まれており、通信内容に関しても研究の対象としている。

本稿では我々の研究活動のうち、図 1 の赤枠で示している 3 つを紹介する。1 つ目はハードウェアのレイヤーに属する PPG センサ*3 のセキュリティ検証である。PPG センサは人の心拍数を測定するためのセンサであり、スマートウォッチなどに搭載されている。心拍数を利用した健康に関するサービスが検討されているが、攻撃者が偽の任意の心拍数を PPG センサに誤認識させられる場合、サービスが成り立たない懸念がある。そこで心拍数を誤認識させる攻撃が可能であるかを検証し、その対策も含めて提案する研究活動である。2 つ目に紹介する研究テーマは通信に関するもので、IoT 機器に関する通信を収集するハニーポット*4 である。IoT 機器へのサイバー攻撃を対策するためには、現状どういった攻撃が行われているかを把握する必要がある。そこで世界中の IoT 機器がどのような不正な通信をしているのかを観測・収集することで攻撃の傾向を把握する。攻撃の対象となっている IoT 機器を把握し、セキュリティ検証を実施することで、いち早く脆弱性を発見、対策の提案をすることを目指している。

最後に紹介する研究テーマはファームウェアから SBOM*5 情報を自動で検出するシステムを取り上げる。SBOM とはソフトウェアを構成する部品表であり、本システムは SBOM 情報としてファームウェアが使用しているライブラリ*6 を特定することを目指す。もし、脆弱性を含むライブラリをファームウェアが使用していた場合、その IoT 機器が脆弱性を有する可能性があることが分かる。IoT 機器の開発者がライブラリをアップデートするなどの対応をとることでその脆弱性をなくし、IoT 機器をよりセキュアにすることができる。

本稿の構成は次のとおりである。**2** では PPG センサのセキュリティ検証を紹介し、**3** で IoT 機器に関する通信を収集するハニーポット、**4** でファームウェアの SBOM 情報検出に関する研究を紹介する。これらの章では各研究の要旨及び研究背景から説明し、検証技術・システムなどの紹介・評価について述べる。その後、**5** でまとめる。

＊3　PPG (PhotoPlethysmoGram) とは人の心拍数を計るための仕組みのことである。
＊4　ハニーポットとは不正な通信を収集するためのシステムのことである。
＊5　SBOM: Software Bill Of Material。ソフトウェアを構成する部品表のこと。部品とはライブラリなどのことを指す。
＊6　ライブラリとは多くのソフトウェアが共通して使用する機能（関数）をひとまとめにしたソフトウェアの部品のこと。ソフトウェアはライブラリから使用したい機能を呼び出すだけでその機能が実現でき、開発が効率化できる。

2　PPG センサのセキュリティ検証：PPG センサに対する反射型偽容積脈波提示攻撃の提案

2.1　要旨

我々は光電式容積脈波記録法（PPG: photoplethysmography）センサに対して、偽の脈波信号を提示する攻撃を研究している。先行研究では液晶ディスプレイを用いて PPG センサに対して脈波を能動的に提示する方法を提案している [1][2]。本稿では、液晶シャッターと鏡の利用により PPG センサが出力する光を減衰・反射させることで偽の脈波信号を提示する攻撃方法を提案し、市販されている 15 種のスマートウォッチに対して評価した結果について報告する。

2.2　研究背景

生体情報を取得可能なセンサ技術が発展し続けている。スマートウォッチなどのウェアラブルデバイスに搭載されている生体センサが低廉化したことで、医療従事者に限らず、幅広いユーザがバイタルデータを活用したサービスを利用できるようになっている [3]。

スマートウォッチの利用者の多くは、スマートウォッチに搭載された各種センサで取得した歩数や心拍数、血中酸素飽和度、血圧、心電図などの情報を、スマートウォッチや連携するスマートフォン、クラウドサービスなどにより得ることで、自身の健康管理に活かしている [4]。さらに、得られた複数の情報から、スマートフォンなどを通じて、利用者が同意の下に提供したバイタルデータを活用したサービスへの応用が始まっている [5]–[8]。すでに、バイタルデータを提出することで、自身の健康状態に基づきポイントが貯まるものや、従業員の労働安全のために活用するサービスなどが提供されている。

しかしながら、バイタルデータを取得するセンサから得られた情報が、人間由来のデータである保証はなく、攻撃者によって偽りのバイタルデータが提示された場合、関連するサービスの信頼性・完全性が損なわれる可能性がある。例えば、バイタルデータに基づいて得られるポイントの不正取得、健康状態を偽ることで休みを不正に取得することや、逆に健康ではないのに健康と偽ることで不正に就労する、AI などを利用した健康管理サービスの元となる学習データの汚染などのような脅威が考えられる。昨今のウェアラブルデバイスの普及に合わせて、「センサからは人間由来のデータのみを取得している」という仮定を見直し、偽りのバイタルデータ提示への対策を行うことが重要である。

我々は各種バイタルデータを取得するセンサのうち、脈波信号を計測するために利用される PPG センサに着目し、偽の脈波を与える攻撃及びその対策を研究している。我々は Active 型の攻撃 [2] や Reflective 型の攻撃 [9] を研究しているが、本稿では Reflective 型の攻撃とその対策を紹介する。本攻撃は PPG センサから発せられる光を液晶シャッターで調整しながら PPG センサに向けて光を反射させることで偽の脈波を誤認識させる。Active 型の攻撃は PPG センサを搭載している機器にあわせて種々の調整が必要であるが、Reflective 型ではそういった制限が少ない。

本章の構成は次のとおりである。**2.3** にて PPG センサでの脈波計測方法を説明する。**2.4** では提案手法と関連研究について述べる。**2.5** にて提案手法の評価結果を示し、**2.6** でまとめを行う。

2.3　基礎知識：PPG センサでの脈波計測

本節では PPG センサによる脈波計測方法を述べる。PPG では血液の成分であるヘモグロビンに光を与えると、光の波長ごとに吸光する特性を利用している。

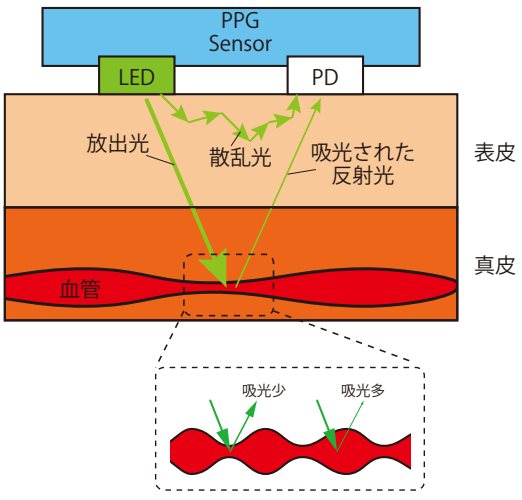

図 2　PPG センシング模式図

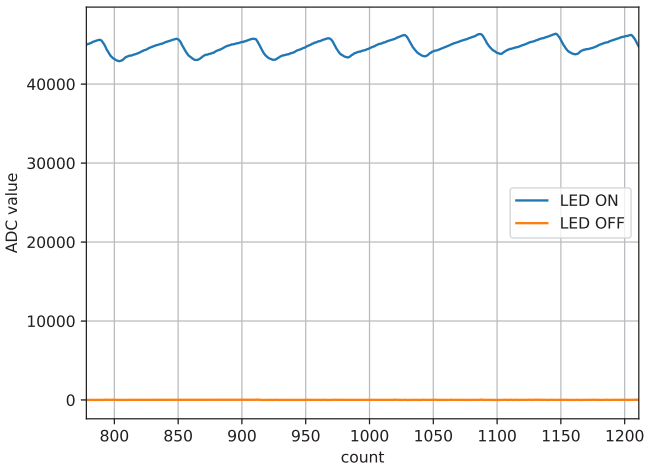

図 3　ROHM 社製 PPG センサ（BH1792GLC-EVK-001）[12] で取得した生体波形

心臓が収縮すると、押し出された血液が大動脈を通じて末梢方向へ波動として伝わる。PPG センサでは、この波動で生じる血管の容積変化に伴う血液量の変化を光学的に捉えるため、LED などを光源とした放出光を皮膚に照射し、フォトダイオード（PD: Photo-Diode）などの受光素子にて、吸光された反射光を電気信号に変換することで脈波を検出しており（図 2）、血液量が多い瞬間にセンシングすると吸光される量が多くなるため、PD にて取得できる反射光は小さくなる。反対に、血液量が少ない瞬間にセンシングすると反射光は大きくなる。したがって、PPG センサでは図 3 で示すような波形を得ることができる。

また、日光や照明、LED などの光源から放出した光を皮膚などの生体へ入射した場合、皮膚表面にて生体内で散乱した散乱光と反射光が観測される（図 2）。PPG では、光源から放出した光が皮膚内で散乱することで観測できる散乱光は DC オフセット成分として観測される。心拍数と血中酸素飽和度を同時に計測するようなデバイスでは、酸化ヘモグロビンと脱酸化ヘモグロビンの吸光係数に差がある赤色や近赤外の光源を用いて計測するが、心拍数のみを取得する腕時計型などのスマートウォッチでは、装着する部位が血流量の少ない手首と限定されるため、多くのデバイスでは、吸光係数の高い 550 nm 付近の緑色の光を用いて計測している [11]。

2.4　提案手法

2.4.1　提案手法（Reflective 型の攻撃）

提案手法の方針を以下に示す。

方針 A）攻撃者が自らのデバイスに対して、脈波を人工的に与えるデバイスにて偽りの心拍数を与える。

方針 B）PPG センサが計測対象とする生体内での反応を模倣する。

方針 C）生体由来の脈波や似せた脈波ではなく、単純な信号を与えることで攻撃を成功させる。

本提案手法では方針 A を実施するために、脈波を人工的に与えるキーとなるデバイスとして、液晶シャッターを採用した。液晶シャッターは外部から印可される電圧によって透過する光量を変化させることができるデバイスである。加えて、PPG センサとは反対となる側にフィルム状の鏡を液晶シャッターへ取り付けることで、液晶シャッターへ照射された光が減光され、鏡に反射し、さらに減光され反射光として放出される構成とし、図 4 で示す反射型脈波提示用デバイスとした。また、本研究で採用している液晶シャッターを図 5 に示す。

このように、Reflective 型の構成にすることによって、PPG センサから放出された放出光を、吸光された反射光として動的に PPG センサへ与えることができる。さらに、スマートウォッチの心拍数計測機能をだますためには、最初に機能を Activate する必要がある。心拍数計測機能の Activate は DC オフセット成分となる散乱光を与えることで実現可能であり [2]、シリコンシートを用いて散乱光を与えることとした（図 6）。これにより、生体内での反応に近い現象を模倣することができ、方針 B を満たす。

方針 C を実施するため、先行研究と同様に、対象となる PPG センサに対し、脈波を与える方法として簡単な正弦波を与える。これは攻撃対象となるデバイスに搭載されている PPG センサの脈波計測手法や、心拍数計測アルゴリズムを簡単には解析できないため、まず単純な正弦波による擬似的な脈波を与えることで、心拍数として計測されるかを理解するためである。

本稿では式（1）により擬似的な脈波波形を得る。α は

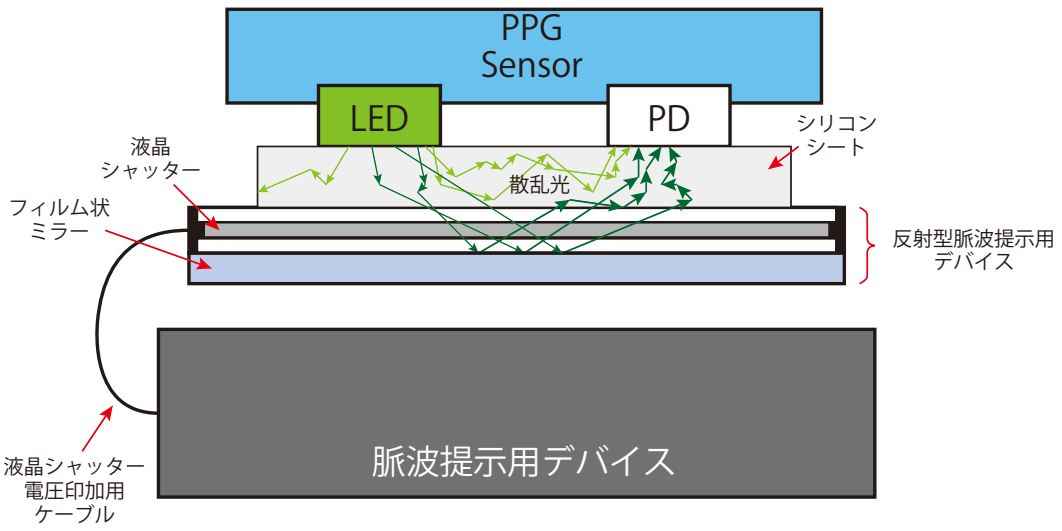

図 4　攻撃模式：Reflective 型

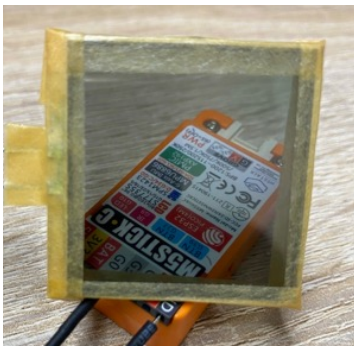

図５　液晶シャッター

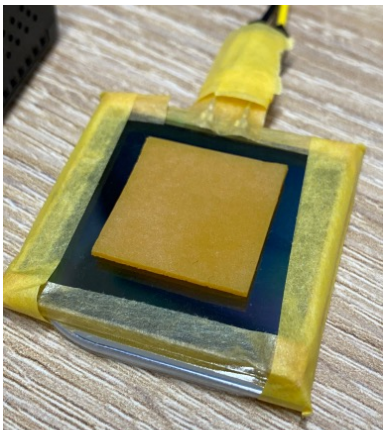

図６　Reflective 型脈波提示用デバイス

図７　評価のためのセットアップ
（左：M5Stack、右：図６の上に攻撃対象を載せた様子）

振幅の高さであり、β は振幅のオフセットである。ω は目的とする心拍数（BPM: Beat per Minute）に合わせ調整する。式（1）で得た値を、反射型脈波提示用デバイスの液晶シャッターへ与える電圧としている。

$$LCDvoltage = \alpha \sin(\omega t) + \beta \quad (1)$$

このようにして、反射型脈波提示用デバイスの液晶シャッター透過率を変化させることで、PPG センサの受光素子に対して動的に反射光を与えることができる。このことは Active 型とは違い、PPG センサが発する放出光をそのまま利用することになり、PPG センサの特性を解析することなく攻撃できるため Active 型より強力となる。

2.4.2　先行研究

先行研究として藤井らの研究 [1] があり、偽りの脈波を与えるために PPG センサを直接液晶ディスプレイに接触させ脈波を提示している。特定のスマートウォッチでの評価ではあるが、少ない誤差範囲内で目的の心拍数を与えることに成功している。本手法は能動的に液晶ディスプレイから放出する光を PD などの受光素子に入射させることで攻撃を行う。Reflective 型は PPG センサから発せられる光を反射する点でアプローチが異なる。

2.5　評価

2.5.1　評価方法

本評価では Reflective 型脈波提示用デバイスを制御するためのデバイスとして、近年電子工作や IoT 機器の試作などで多く使われている M5Stack 社の M5Stack Basic [13]（図７の左側）を用いる。M5Stack Basic は数千円と安価であり、多くの販売チャネルから購入できるため入手性も高い。また、液晶シャッターは Pimoroni 社の LCD shutter を利用した [14]。

2.5.2　評価結果

評価のため、複数の企業が市販しているスマートウォッチを入手し、それらのうち 15 種のデバイスにて攻撃の評価を行った結果、評価した 15 種のスマートウォッチ全てに対して目的の心拍数を与えることができた。さらに、先行研究（Active 型）[2] では、デバイスごとにパラメータを変更して攻撃を行う必要があったが、本提案手法（Reflective 型）ではデバイスごとにパラメータを調整する必要なく目的の心拍数を与えることに成功した。以上の検証によって、本提案手法の有効性及び先行研究と比較した優位性を示すことができた。

2.6　本章のまとめ

スマートウォッチなどが心拍数を計測するために利用している PPG センサへの攻撃方法を提案し、市販されているスマートウォッチにて評価した結果を示した。

このような攻撃により、PPG センサそのものの信頼性・完全性が損なわれるわけではないが、PPG センサを応用したデバイスにて計測した心拍波形や心拍数を、クラウドサービスなどで二次利用する際、生体から得られたものなのかをきちんと確認する必要があると考えられる。また、PPG センサから取得した波形を AI などで活用するための教師データとして蓄積する際も同様であり、確認せずに蓄積すると、データセットが人工物由来の波形により汚染される可能性がある。

今後は心拍数と血中酸素飽和度を同時に計測するようなデバイスへの攻撃方法の検討や皮膚表面温度を模倣する仕組みの検討、生体由来の波形を模した波形を与えることでの評価、各種攻撃への対策の提案を実施していく。さらにより高度な攻撃方法の検討を行う予定である。

3 IoT 機器に関する通信を収集する研究：全ポート待受型の簡易 TCP/TLS ハニーポットによるサイバー攻撃観測

3.1　本章の要旨

多種多様な IoT 機器がインターネットに接続されるようになった結果、これらの機器を標的としたサイバー攻撃が増加している。これらの機器では様々なサービスが様々なポートで動作しているため、ダークネットやハニーポットによる従来の観測だけでは、これらの機器への攻撃の観測は困難である。そこで我々は全てのポートにおいて TCP や TLS のハンドシェイク処理を行い、攻撃通信を観測する簡易ハニーポット（Handshaker）を構築し、サイバー攻撃の観測及び分析を行っている。本章では Handshaker の概要及びその観測結果について報告する。

3.2　研究背景

多種多様な IoT（Internet of Things）機器がインターネットに接続されるようになった結果、これらの機器を標的としたサイバー攻撃が増加している。例えば、2016 年に流行したマルウェア「Mirai」は、ブロードバンドルータやネットワークカメラなど、IoT 機器を中心に感染拡大し、大規模な DDoS（Distributed Denial-of-Service）攻撃を実行したことで知られている [15]。これらの IoT 機器では、様々なサービスが様々なポート番号で動作しているため、ダークネット*7 やハニーポット*8 による従来の観測だけでは、これらの機器への攻撃の観測は困難である。あらゆるポート番号への攻撃を網羅的に観測することができれば、攻撃者に狙われている脆弱性や IoT 機器、関連するマルウェアなどの実態把握に貢献できる。

そこで我々は全てのポートにおいて TCP（Transmission Control Protocol）や TLS（Transport Layer Security）のハンドシェイク処理を行い、攻撃通信を観測する簡易ハニーポット（以降、Handshaker と呼ぶ）を構築し、サイバー攻撃の観測及び分析を行っている。本章では Handshaker の構成と実装及び 2023 年の観測結果について報告する。

本章の構成は次のとおりである。まず、**3.3** で Handshaker の構成と実装を説明する。次に、**3.4** で 2023 年の観測結果の概要を述べ、**3.5** でまとめと今後の課題を述べる。

3.3　Handshaker の構成と実装

3.3.1　先行研究

全ての TCP ポートでサイバー攻撃の観測を試みる先行研究としては文献 [16][17] が挙げられる。大平らはポートのアクセス状況に応じて、実際の攻撃に使用される可能性のあるポートのみを待ち受ける汎用性の高いハニーポットシステムを構築した [16]。また、鈴木らは全 TCP ポートでの観測を実現するため、TCP の SYN パケットに対して SYN+ACK パケットを返す擬似応答センサを構築し [17]、初期ペイロードの分析を行った [18]。本稿の Handshaker も同様の挙動を示す観測システムであるが、Handshaker は TCP だけでなく、より広く TLS ハンドシェイク後の通信も観測できるようにシステムを構築した。

3.3.2　構成

Handshaker の構成を図 8 に示す。Handshaker は、TCP サーバ・TLS サーバ（以降、これらの 2 つのサーバプログラムを「TCP/TLS サーバ」とする）、アクセス

*7　インターネットから到達可能な未使用の IP アドレス空間のこと。サイバー攻撃関連のパケットを分析することができる。
*8　脆弱なサービスを模擬するおとりシステムのこと。

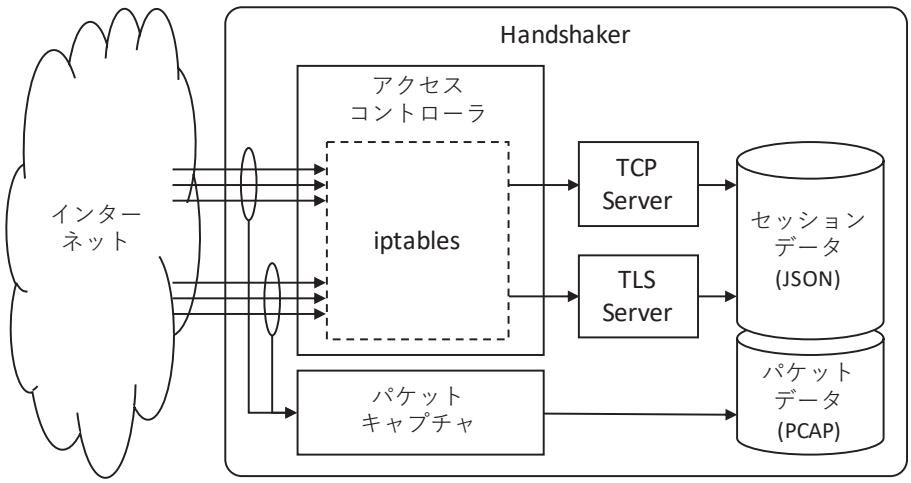

図 8　Handshaker の構成

コントローラ、パケットキャプチャの 4 つのコンポーネントから成る。TCP/TLS サーバはそれぞれ一つのポートで接続を待ち受け、それぞれ TCP/TLS のハンドシェイクを行い（TLS サーバは、TCP の 3-way ハンドシェイクで接続を確立したあとに TLS のハンドシェイクを行う）、接続元から送られてくるペイロードを受信し、これをセッション情報とともに記録する。アクセスコントローラは、インターネットからの全通信を TCP/TLS サーバが待ち受けるポートに変換し、これらの通信を制御する。ただし、TCP/TLS サーバはアクセスコントローラによって変換された IP アドレス・ポート番号を変換テーブルから読み取り、もとの接続情報をセッションデータとして記録する。パケットヘッダ情報など、より詳細な情報を分析できるようにするため、セッションデータとは別にパケットキャプチャを用意してパケットデータを記録する。なお、UDP（User Datagram Protocol）通信については、パケット単位でペイロードを受信し、セッション情報として記録する。

3.3.3　実装

TCP/TLS のハンドシェイクを行うサーバプログラムは Go 言語で実装した。現状の実装では、簡単のため、TCP の 3-way ハンドシェイク後に送られてくるペイロードで TLS のハンドシェイクを行うか否かを判断する実装にはせず、TCP のハンドシェイクを行うモード（TCP モード）と TLS のハンドシェイクを行うモード（TLS モード）がそれぞれ別の IP アドレスで動作することを前提としてプログラムを実装した。なお、Go 言語の標準ライブラリの都合上、TLS サーバは TLS 1.2 にのみ対応した。また、運用においては、TLS の証明書として自己署名証明書を使用した。アクセスコントローラには iptables を使用し、iptables の TPROXY 機能で全ポート宛の通信を TCP/TLS サーバが待ち受けるポートへ変換した。パケットキャプチャも Go 言語を用いて実装した。また、出力された JSON 形式のデータは、別のサーバ上で動作する PostgreSQL データベースに随時登録し、準リアルタイムでデータを分析できるようにした。

3.4　観測結果

本節では Handshaker が 2023 年の 1 年間（2023 年 1 月 1 日から 12 月 31 日）に観測した結果の概要をまとめる。TCP モード・TLS モードの Handshaker が観測する IP アドレスは、それぞれ同じネットワークに属する日本国内の 31 IP アドレスである。

3.4.1　観測結果の概要

2023 年の観測結果の概要を表 1 に示す。有効セッション数とは、Handshaker が受信したセッションのうち、接続元から 1 byte 以上のペイロードを受信したセッションの数である。すなわち、TCP モードでは TCP の 3-way ハンドシェイクが成立して 1 byte 以上のデータが送られてきたセッション数を、TLS モード

表 1　観測結果の概要（2023 年 1 月 1 日から 12 月 31 日）

	観測 IP アドレス数	有効セッション数	送信元 IP アドレス数	宛先ポート数
TCP モード	31	181,739,404	621,136	65,536
TLS モード	31	46,760,753	130,276	65,535
UDP	62	29,015,728	311,491	65,438

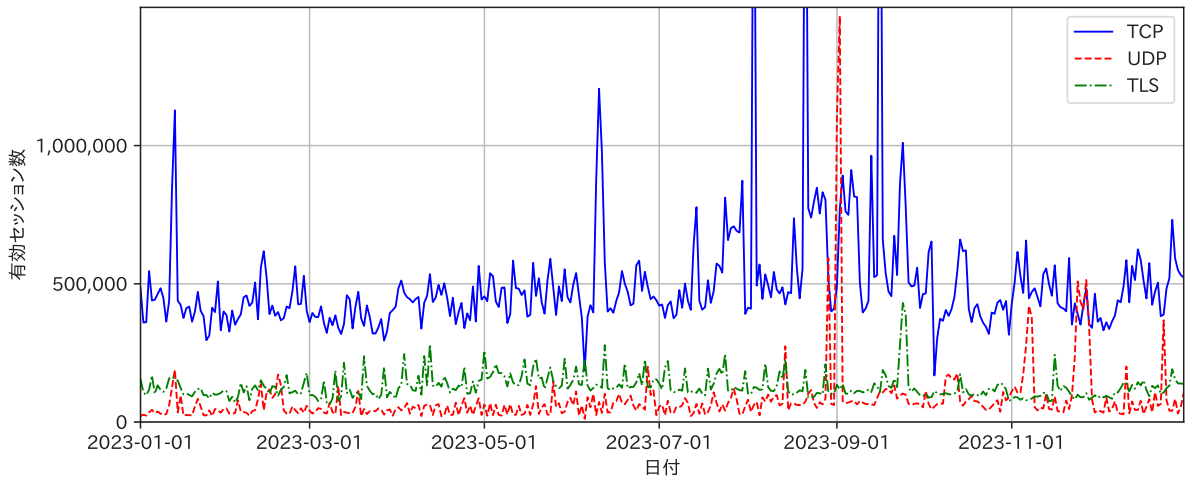

図 9　Handshaker が観測した日ごとの有効セッション数の推移

では TLS のハンドシェイクが成立して 1 byte 以上のデータが送られてきたセッション数を表す。送信元 IP アドレス数は、有効セッションの送信元 IP アドレスから重複を除いた IP アドレスの数であり、宛先ポート数はセッションの宛先ポート番号から重複を除いたポート番号の数である。

2023 年の 1 年間に、TCP モードでは約 62 万の IP アドレスから約 1.8 億件の通信が観測され、TLS モードでは約 13 万の IP アドレスから約 4676 万件の通信が観測された。また、UDP については、約 31 万の IP アドレスから約 2901 万のパケットが観測された。TCP・TLS・UDP いずれのプロトコルでも、広範なポートへの通信が確認された。

3.4.2　有効セッション数の推移

Handshaker が観測した日ごとの有効セッション数の推移を図 9 に示す。TCP モードの Handshaker では、1 日平均 47 万件（1 IP アドレスあたり約 1.6 万件）のセッションが観測され、TLS モードの Handshaker では、1 日平均 13 万件（1 IP アドレスあたり約 4 千件）のセッションが観測された。また、UDP については、1 日平均 8 万パケット（1 IP アドレスあたり約 1.3 千パケット）が観測された。

3.4.3　観測事例

Handshaker が観測する攻撃通信の種類は多岐にわたるが、通常のサービスでは使用されないようなポートで動作する IoT 機器の脆弱性を狙った攻撃通信も多く含まれる。例えば、国内で販売されていたブロードバンドルータの 52869/TCP で動作する UPnP 関連サービスには OS コマンドインジェクションの脆弱性が存在し、この脆弱性を狙った攻撃通信が 2017 年以降継続して観測されている [19]。また、特定の IoT 機器の脆弱性を狙った同様の攻撃は近年多数観測されており [20]、そのほかにも、TLS モードの Handshaker においては、SSL-VPN 機器の脆弱性を狙った攻撃が、通常の HTTPS のポート番号（443/TCP）と異なる高いポートで多数観測されている。

3.5　本章のまとめ

本稿では全てのポートにおいて TCP や TLS のハンドシェイクを行う簡易なハニーポット（Handshaker）の構成・実装について説明し、2023 年の観測結果を紹介した。今後の予定として、Handshaker で得られた知見を基に、より詳細な攻撃を観測する独自のハニーポットの研究開発を進めるとともに、ダークネット・Handshaker・ハニーポットを活用し、サイバー攻撃をより網羅的に観測・分析できる環境の構築を進める。

4　SBOM 情報の検出に関する研究：ライブラリ固有のリソース情報に着目したアーキテクチャに左右されない SBOM 検出手法

4.1　本章の要旨

ソフトウェアにおけるコンポーネントの管理手法のひとつとして、ソフトウェア部品表（SBOM）を用いた管理が着目されている。本来であればソフトウェア開発者により作成・管理・提供されるものであるが、提供されない場合や意図しないコンポーネントの混入有無を調べる場合に向けバイナリデータを解析して SBOM 情報を作成するツールが存在する。世の中に存在する多くのツールではライブラリファイルに対するハッシュ値やコード断片をはじめとした変化に弱い情報を用いるためバージョンやコンパイラ、アーキテクチャなどの影響を受けやすい問題がある。本研究ではアーキテクチャに極力依存しない情報を用いてコンポーネントを推定し SBOM 情報の出力を試みた。

4.2　研究背景

4.2.1　SBOM 及び SPDX について

SBOM（Software Bill of Materials）とは、ソフトウェア部品表とも呼ばれるソフトウェア構成を記した情報のことである。IoT 機器のファームウェアを含むソフトウェアサプライチェーンリスクへの対策としても注目されており、SBOM を用いることで OSS（Open Source Software）をはじめとした外部ライブラリの脆弱性や不具合が確認された際に、ソフトウェアに対する影響の有無を迅速に判断することが可能となる。

SBOM を文書化するための様々な出力フォーマットがあるが、代表的な出力フォーマットのひとつに SPDX（System Package Data Exchange）[21] がある。このフォーマットでは各コンポーネントの「コンポーネント名称」と「バージョン情報」、「開発者名」、「依存関係」を記録・管理する。本研究においても SBOM の出力フォーマットとして SPDX を採用している。

4.2.2　研究目的

SBOM は本来ソフトウェア開発者により作成・管理・提供されることが望ましい。しかしながら、製造ベンダから提供されない場合や実行バイナリ内における不要なライブラリの調査をする場合などにおいて、独自に SBOM を確認しなければならないことがある。そのような状況に向けて一般にバイナリデータを解析することで SBOM 情報を出力するツールが存在する [22]–[24]。

SBOM 情報検出の方法として、ソフトウェアを構成するファイルのハッシュ値やファイル内における既知のバイナリデータの有無により SBOM を検出する方法があげられる。この方法の SBOM 検出精度は高い

が、一方でソフトウェアが作成されるときのコンパイラオプションやソフトウェアが動作するアーキテクチャ、ソフトウェアのバージョンが変わるとハッシュ値やバイナリデータが変化するため、このような作成環境の差異に弱いという問題がある。そこで我々はアーキテクチャに依存しない情報を用いた SBOM 情報の検出を目標とし、IoT 機器に代表されるような様々なアーキテクチャや多少のコード修正にも対応可能な SBOM 情報検出技術の研究開発をしている。

4.2.3　研究方針

我々が研究開発している SBOM 情報検出技術ではファームウェアのパッケージファイルに対してファームウェアの展開と分析を実施し、SBOM 情報の生成及び SPDX 形式での出力を行う。ファームウェアは多種多様なファイルで構成されるが、現在の分析対象は .so の拡張子をもつ共有ライブラリに限定し、SBOM 情報として推定結果のライブラリ名、バージョン情報、類似度を検出する。

共有ライブラリに限定した理由は、ライブラリごとにファイル単位で分けられていること、汎用的に利用可能なライブラリであることからある特定の機種だけでなく、他機種においても利用されている可能性があり、研究におけるサンプルを用意する観点で有利であるためである。

本研究においては、アーキテクチャに依存しない情報としてパッケージファイルの rodata セクションのデータを使用する。このセクションには定数やデバッグ情報などが含まれており、各ライブラリ固有の特徴として使用することでライブラリ名など検出に活用できると考えたためである。

4.3　研究の方法と評価

4.3.1　ライブラリ名の判定方法

ライブラリ名を判定する方法の全体の流れを図 10 に示しており、手順は以下のとおりである。

1. ファームウェアバイナリを展開して共有ライブラリ（so ファイル）を抽出
2. 抽出した共有ライブラリから rodata セクションをテキスト形式で取得
3. テキスト形式で取得した rodata セクションに対して、分散表現 “doc2vec” を使い、モデルデータを作成
4. 作成したモデルを用いて、判定対象の未知のライブラリの類似度を計算（モデルに含まれるライブラリのうち、最も類似度が高いライブラリ名をその未知のライブラリの名称とする）

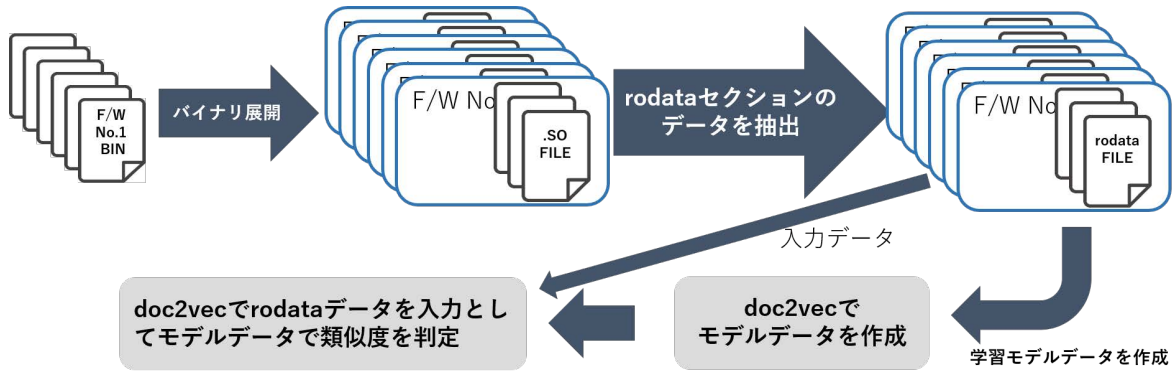

図 10　ライブラリ名の特定方法

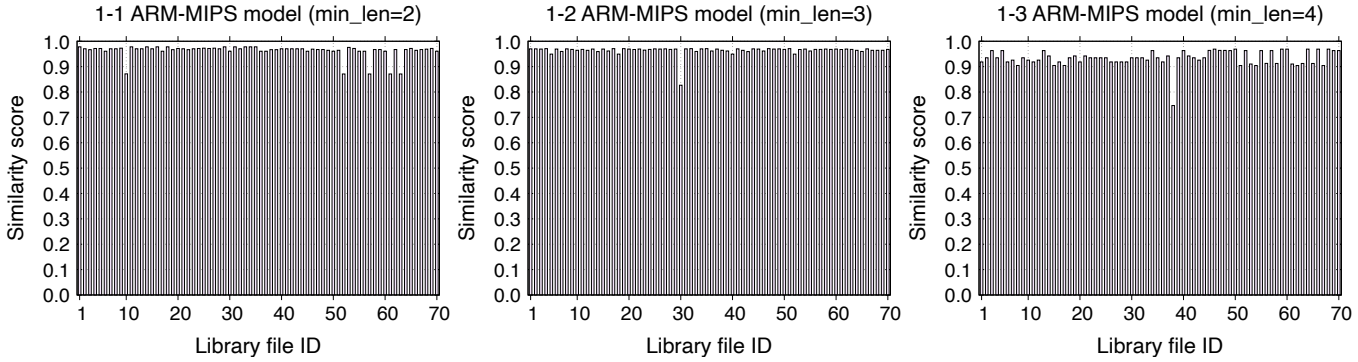

図 11　ARM-MIPS 混合モデルで求めた libz.so の類似度 (min_len: 最小単語サイズ)

4.3.2　フィージビリティの評価

4.3.2.1　単一のライブラリを用いた検証

ライブラリ名の判定方法のフィージビリティを評価するにあたり、まずは単一のライブラリのみを用いて実験する。本実験のデータセットとしてARMのlibz.soを54個（うち、libz.soのバージョンは6種類）とMIPSのlibz.soを36個（うち、libz.soのバージョンは5種類）を用意した。doc2vecのモデル作成用にARMとMIPSそれぞれから無作為に10個選び、最小単語サイズ*9を2、3、4に変えながらモデル（以下、ARM-MIPS混合モデルと呼ぶ）を3つ作成した。モデルの作成に使用しなかった44個のARMのlibz.soと26個のMIPSのlibz.soを判定対象の未知のライブラリとして使用する。本実験において、ARM-MIPS混合モデルを用いて、判定対象のlibz.soを検証し、高い類似度が得られれば実験成功とする。

図11にARM-MIPS混合モデルと判定対象のライブラリの類似度のグラフを示す。グラフ1-1～1-3はそれぞれARM-MIPS混合モデルの最小単語サイズ（min_len）2～4の結果に対応している。グラフの横軸は判定対象のライブラリに対応する番号（Library file ID）を示しており、縦軸にモデルとの類似度（Similarity score）を示している。1～44番はARMのlibz.soであり、45～70番はMIPSのlibz.soである。

グラフ1-1～1-3はおおむね0.9以上という高い類似度が得られ、未知のライブラリがlibz.soであることを判定できるフィージビリティを示唆している。グラフ1-1を見ると5個のライブラリの類似度が0.9を下回っている。これは文字数が少ない単語を用いてモデルを作成すると、ノイズになりやすく、類似度の低下を引き起こしているものと思われる。実用的には、より適した最小単語サイズを設定する必要がある。

4.3.2.2　単一アーキテクチャのモデルによる検証

アーキテクチャに極力依存せずにライブラリ名判定が可能かを調べるため、単一アーキテクチャのlibz.soでモデルを作成し、同一アーキテクチャ及び異なるアーキテクチャのlibz.soに対して高い類似度が得られるかを実験する。

ARM-MIPS混合モデルの作成に用いた10個のARMのlibz.soによりARMモデルを作成し、10個のMIPSのlibz.soによりMIPSモデルを作成する。判定対象の未知のライブラリとして前節と同様に1～44番のARMのlibz.soと45～70番はMIPSのlibz.soを使用する。

図12のグラフ2-1～2-3がそれぞれARMモデル（最小単語サイズ2）～ARMモデル（最小単語サイズ4）で求めた結果である。グラフ2-3からARMモデルによりARMのlibz.so（1～44番）に対して0.9以上の

*9　最小単語サイズとは文章から単語に分割したデータを作成する際に、採用する単語の文字数のことである。最小単語サイズ以下の短い単語は削除され、データが生成される。本研究の実装ではpythonパッケージgensimのsimple_preprocessメソッドでrodataのテキスト表現から単語を抽出している。最小単語サイズはsimple_preprocessに入力するパラメータmin_lenに対応している。

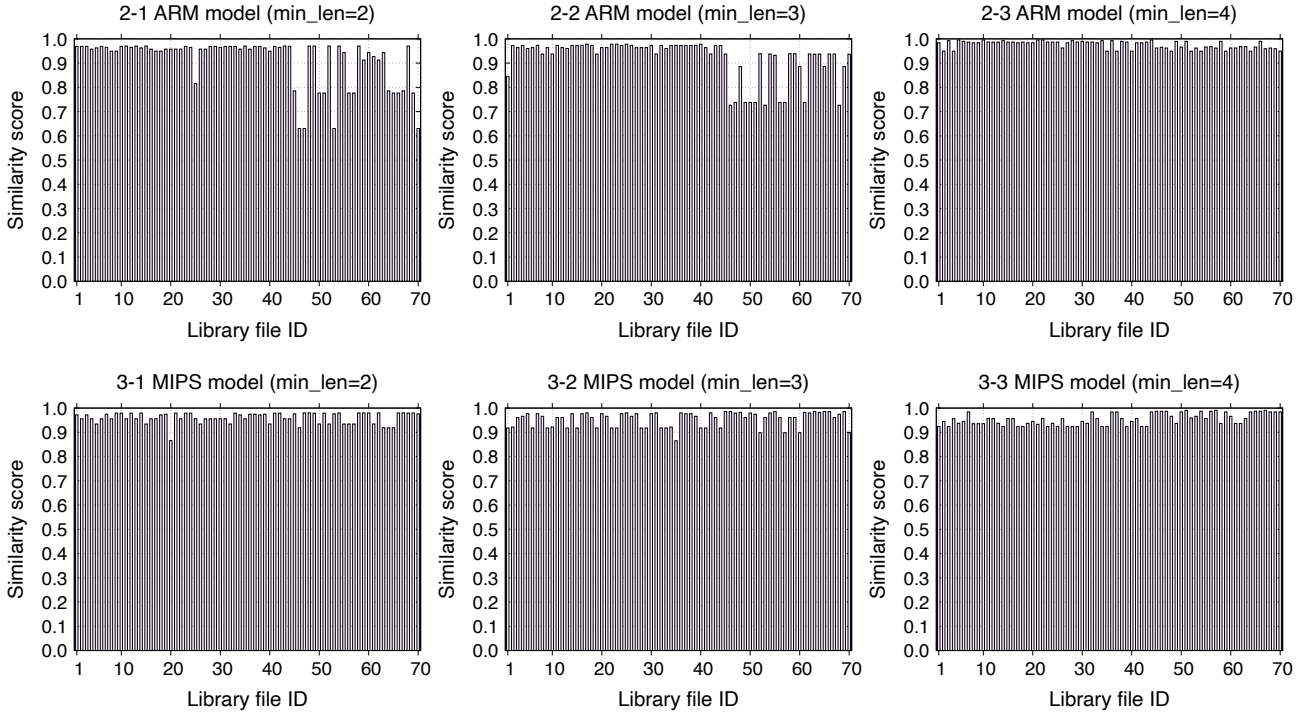

図12　単一アーキテクチャのモデルで求めたlibz.soの類似度

類似度が得られており、また異なるアーキテクチャである MIPS の libz.so（45 ～ 70 番）に対しても 0.9 以上の類似度が得られている。グラフ 3-1 ～ 3-3 が MIPS モデル（最小単語サイズ 2）～ MIPS モデル（最小単語サイズ 4）の結果であり、グラフ 3-3 から MIPS モデルにおいても ARM 及び MIPS の libz.so に対して 0.9 以上の類似度が得られている。rodata のテキスト表現で作成したモデルがアーキテクチャに大きく依存していないことが確認できた。なお、最小単語サイズが 2 と 3 の ARM モデル及び MIPS モデルでは 0.9 未満といった類似度が相対的に低くなっている libz.so があり、これは前節でも述べたように短い単語がノイズとなっていると考えられる。

4.3.2.3　複数種類のライブラリを用いた検証

複数種類のライブラリ名を判定できるかを評価するために 6 種類のライブラリにより実験した。表 2 に ARM-MIPS 混合モデル作成用のライブラリ及び判定対象のライブラリの種類とそれぞれの個数を示す。判定対象のライブラリは ARM と MIPS を合わせるといずれも 10 個である。ARM-MIPS 混合モデルは最小単語サイズを 2、3、4 と変えながら 3 つ作成した。

表 3 にライブラリ名判定の正解率（正しく判定できたライブラリ数／総数（10 個）× 100）を示す。最小単語サイズが 3 と 4 のモデルの平均正解率がそれぞれ 92 % と 93 % という良い結果が得られた。これは前節でも述べたように rodata セクションに含まれる単語のうち、文字数が少ない単語を除くことでノイズを除去できたためと考えられる。また、最小単語サイズが 4 のモデルの結果において、ライブラリ名を誤判定した 4 個の libpthread.so は全て MIPS のライブラリで、いずれも libcrypt.so に誤判定された。この誤判定が MIPS と ARM の差異によるものなのか、ライブラリの元のソースコードが ARM と MIPS で違っていたのかなどの調査は今後の課題とする。

4.4　本章のまとめ

我々が研究開発しているライブラリ名判定手法ではライブラリの rodata セクションのテキスト表現をもとに doc2vec のモデルを用いる。2 種類の評価実験を実施し、本手法によりライブラリ名を判定するためのフィージビリティを示した。特に、ARM 及び MIPS といった異なるアーキテクチャのライブラリを判定するモデルが作成できており、アーキテクチャに大きく依存せずに判定できる可能性を示した。今後も本研究を推進し、オープンで使い勝手がよく、脆弱性管理に活用できる実用的なシステムの開発を目指す。

表 2　ARM-MIPS 混合モデルの作成に使用したライブラリ

ライブラリ名	モデル作成に使用したライブラリ数		判定対象のライブラリ数	
	ARM	MIPS	ARM	MIPS
libz.so	5	1	5	5
librt.so	5	4	4	6
libpthread.so	8	7	6	4
libdl.so	8	9	7	3
libcrypt.so	47	30	5	5
ld-uClibc.so	5	9	4	6

表 3　複数種類のライブラリを混合した類似度検証結果

判定対象のライブラリ名	正解率（%）		
	最小単語サイズ		
	2	3	4
libz	100	100	100
librt	40	100	100
libpthread	60	60	60
libdl	100	100	100
libcrypt	70	90	100
ld-uClibc	100	100	100
平均正解率	78	92	93

5　おわりに

本稿では我々が実施しているローレイヤーセキュリティの研究活動のうち、PPG センサのセキュリティ検証と IoT 機器に関する通信を収集するハニーポット、ファームウェアから SBOM 情報を検出する研究について紹介した。これらの研究の目標は各レイヤーの個々のセキュリティ検証を実施することで IoT 機器全体のセキュリティを担保することである。また、IoT 機器のレイヤーの概念は多くの IoT 機器に共通するものであり、各レイヤーに対して我々が開発した技術はある特定の種類の IoT 機器だけでなく、多くの IoT 機器を横断的に検証できるように研究活動を実施している。今後も研究活動を継続し、論文や OSS（オープンソースソフトウェア）などを公開することで安心・安全な情報社会に貢献していく。

【参考文献】

1　藤井 敦寛, 村尾 和哉, “ディスプレイを用いて光電脈波センサに任意の脈波を計測させる手法の提案,” マルチメディア, 分散協調とモバイルシンポジウム 2021 論文集 2021 (1), pp.702–708, 2021-06-23.
2　竹久 達也, 丑丸 逸人, 牧田 大佑, 有末 大, 三村 聡志, 末田 卓巳, 飯島 涼, 伊沢 亮一, 井上 大介, “PPG センサを用いたウェアラブルデバイスに対する偽容積脈波提示攻撃に関する一考察,” コンピュータセキュリティシンポジウム 2021 論文集, 2021/10, pp.871–878.
3　総務省 : 令和 3 年版情報通信白書, https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/pdf/01honpen.pdf
4　塚本 昌彦, “ウェアラブルの最新事情と行動変容,” 日本健康教育学会誌 30 (1), pp.79–85, 2022-02-28, https://cir.nii.ac.jp/crid/1390854882637549184
5　田村 俊世, “ウェアラブルセンサ, 非接触センサの医療応用,” 医療機器学, 2020, 90 巻, 1 号, pp.11–23, 公開日 2020/04/11, Online ISSN1884-054X, Print ISSN 1882-4978, https://doi.org/10.4286/jjmi.90.11
6　板生 清, 駒澤 真人, “ウェアラブルデバイスの応用と近未来の展開,” エレクトロニクス実装学会誌, 2015, 18 巻, 6 号, pp.384–389, 公開日 2015/12/01, Online ISSN 1884-121X, Print ISSN 1343-9677
7　天笠 志保, 荒神 裕之, 鎌田 真光, 福岡 豊, 井上 茂, “医療・健康分野におけるスマートフォンおよびウェアラブルデバイスを用いた身体活動の評価：現状と今後の展望,” 日本公衆衛生雑誌, 論文 ID 20-143, [早期公開] 公開日 2021/06/11, Online ISSN 2187-8986, Print ISSN 0546-1766, https://doi.org/10.11236/jph.20-143
8　小糸 直基, “健康情報プラットフォームに求められる思想と実装,” 計測と制御, 2020, 59 巻, 4 号, pp.274-277, 公開日 2020/04/21, Online ISSN 1883-8170, Print ISSN 0453-4662, https://doi.org/10.11499/sicejl.59.274
9　竹久 達也, 丑丸 逸人, 牧田 大佑, 有末 大, 三村 聡志, 飯島 涼, 伊沢 亮一, 井上 大介, “PPG センサに対する反射型偽容積脈波提示攻撃の提案,” 信学技報 (BioX2022-59). 2022. 122. 197. pp.20–25.
10　S. Prahl, “Tabulated Molar Extinction Coefficient for Hemoglobin in Water,” https://omlc.org/spectra/hemoglobin/summary.html
11　高木 健太郎, “プレチスモグラフィ,” 医用電子と生体工学, 1965, 3 巻, 1 号, pp.3–14, 公開日 2011/07/05, Online ISSN 2185-5498, Print ISSN 0021-3292, https://doi.org/10.11239/jsmbe1963.3.3
12　ROHM: BH1792GLC-EVK-001, https://www.rohm.co.jp/sensor-shield-support/pulse-wave-sensor
13　M5Stack: M5Stack Core: ESP32 Basic Core IoT Development Kit, https://shop.m5stack.com/collections/m5-core/products/basic-core-iot-development-kit
14　Pimoroni, LCD shutter(small), https://shop.pimoroni.com/products/lcd-shutter
15　Brian Krebs, “Hacked Cameras, DVRs Powered Today’s Massive Internet Outage,” 2016. https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/
16　大平 健司, 宋 中錫, 高倉 弘喜, 岡部 寿男, “様々なアプリケーションへの攻撃活動を察知する汎用性の高いハニーポットシステムの構築と運用,” 電子情報通信学会 論文誌 D, vol.93, no.7, pp.1125–1134, 2010.
17　鈴木 悠太, 後藤 洋一, 中村 康弘, “簡易ハニーポットによるダークネット観測とペイロード分類手法の提案,” 第 77 回全国大会講演論文集, vol.2015, no.1, pp.483–484, 2015.
18　中村 康弘, “初期ペイロードに着目したネットワーク走査活動の分析,” 第 79 回全国大会講演論文集, vol.2017, no.1, pp.523–524, 2017.
19　NICTER Blog, “日本国内の Mirai に感染する機器の観測状況,” https://blog.nicter.jp/2021/05/jp_mirai_and_infected_logitec_routers/
20　情報通信研究機構 サイバーセキュリティネクサス, “NICTER 観測レポート 2023,” https://csl.nict.go.jp/report/NICTER_report_2023.pdf .
21　The Linux Foundation, “System Package Data Exchange (SPDX),” https://spdx.dev/
22　Lawrence Livermore National Laboratory (LLNL), “ROSE,” http://rosecompiler.org/ROSE_HTML_Reference/index.html
23　nexB Inc., “ScanCode toolkit,” https://github.com/nexB/scancode-toolkit
24　Tern-tools, “TERN,” https://github.com/tern-tools/tern

伊沢 亮一　(いさわ りょういち)

サイバーセキュリティ研究所
サイバーセキュリティ研究室
主任研究員
博士 (工学)
サイバーセキュリティ
【受賞歴】
2023 年　CSEC 優秀研究賞 (CSEC 研究会)
2020 年　Best Paper Award (IEEE TrustCom 2020)
2018 年　Best Paper Award (AsiaJCIS 2018)

竹久 達也　(たけひさ たつや)

サイバーセキュリティ研究所
サイバーセキュリティ研究室
招へい専門員
サイバーセキュリティ、ローレイヤー、IoT デバイス
【受賞歴】
2022 年　コンピュータセキュリティシンポジウム 2022　優秀論文賞
2018 年　コンピュータセキュリティシンポジウム 2018　最優秀論文賞

牧田 大佑　(まきた だいすけ)

サイバーセキュリティ研究所
サイバーセキュリティ研究室
主任研究技術員
博士 (工学)
サイバーセキュリティ
【受賞歴】
2016 年　情報処理学会論文賞
2015 年　情報処理学会論文誌ジャーナル / JIP 特選論文
2014 年　SCIS 論文賞

三村 聡志　(みむら さとし)

サイバーセキュリティ研究所
サイバーセキュリティ研究室
専門研究技術員
サイバーセキュリティ

藤原 吉唯　(ふじわら よしただ)

サイバーセキュリティ研究所
サイバーセキュリティ研究室
研究技術員
サイバーセキュリティ
【受賞歴】
2023 年　DA シンポジウム 2022 優秀発表賞
2020 年　CSEC 優秀研究賞