東京大学大学院
情報理工学系研究科 電子情報学専攻
修士論文

ARP リクエスト監視に基づく LAN 内異常検知

Anomaly Detection in LAN with ARP Request Monitoring

松藤 央
Kai Matsufuji

指導教員　落合 秀也 准教授

2020 年 1 月

概要

情報化社会の発展に伴い、我々の生活を豊かにする多様な情報技術とそれを利用したアプリケーションが社会産業活動の基盤として普及・浸透してきている。しかしそのようなデジタル化と情報化の進展に伴い、マルウェア感染などのサイバーセキュリティに関するリスク管理の問題が新たに注目されるようになった。ランサムウェアを始めとするさまざまな悪意ある攻撃が我々のデジタル化された社会産業システムを脅かす中、既存のセキュリティ技術であるファイアウォールによるシステムへのサイバー攻撃に対する防御はもはや万全とは言い難い状況へとなりつつある。その結果、現在の LAN を保護するための新しいセキュリティ技術の研究開発が求められている。IoT デバイスを始め生活のあらゆるところにデジタル機器が接続されている現代において、全てのデバイス一つ一つにウイルス対策ソフトや新しいセキュリティ技術を導入し、機能アップデートを継続的に実施することは事実上現実的ではない。

そこで本研究においては、既存のネットワーク構成は変えずに、かつ容易に導入可能な異常検知技術を提案した。提案手法では、LAN 内でブロードキャストされる ARP リクエストを異常の検知に用いる。まず、ARP リクエストを異常検知に利用するため、ARP リクエストパケットの特徴量抽出について複数の手法を検討した。その上で過去の特徴量の傾向に基づく予測ベースのフィッティングモデルを用いた異常検知を提案し、その性能評価を実システムを用いて行った。

さらに LAN 内拡散マルウェアに感染した疑いの高いネットワークに対して 2 ヶ月間の ARP 監視を行ったところ、3 つの特徴量を併用することで合計 2399 の MAC アドレスのうちのべ 2256 のアドレスで 1 回以上の異常を発見した。その異常をさらに詳細に分析するために、サービスを提供していない機器への通信を Malicious なものであるとみなし、それを抽出したところ、提案手法で見つかった異常のうち合計 11 についてはトラフィック解析からも明らかに Malicious なものであることを確認することができた。これにより本研究で提示した手法を用いて、実際に悪意のある挙動をしている端末を異常であると判定することができ、本研究の有用性を示すことができた。

目次

図目次

表目次

第 1 章

序論

1.1 背景

情報化社会の発展に伴い我々の生活は日々豊かになり、その技術は身の回りのあらゆる場面で見られるようになった。その代表例としてメールや SNS などを用いて顔も知らない相手と密に交流ができるようになったことや、24 時間無料で誰でも自由に接続できる公衆 Wi-Fi が普及し始めていることが挙げられる。さらに最近では家庭の電化製品に IoT 技術を導入し、製品自身が自動でインターネットに接続して情報交換を行う試みも行われている [1, 2, 3]。しかしそのような諸技術の普及には、利便性の裏に様々なリスクが隠れていることも忘れてはならない [4]。

例えばメール技術の浸透の裏には、フィッシングメールなどで今なお容易にマルウェアが端末に侵入してしまうという現実がある。さらに公衆 Wi-Fi を無防備に使用することは悪意のある攻撃者と同じ LAN に接続されていることを意味し、そこから通信の内容を傍受されたり新たなウイルスに感染するリスクを負うことになる。そして IoT デバイスにおいてはそのずさんな管理が問題となっている。デフォルトパスワードのまま放置されたデバイスは攻撃者からの格好の餌食となり、中でも Mirai を始めとしたマルウェアの被害は後を絶たない [5, 6]。そのような実情から、デバイスを監視しマルウェアをいち早く検出するための LAN 内セキュリティ技術が以前にも増して必要とされている。既存の LAN 内セキュリティ対策として代表的なものにファイアウォールあるが、マルウェアはあらゆる手段を使っていとも簡単に端末に侵入するため、もはや既存の技術だけでは万全とは言えないのが実情である。しかしそれらの問題を解決するため既存のウイルス対策ソフトや新しいセキュリティ技術を現存する端末全てに一つ一つ導入することは現実上不可能である。

そのため現在

- LAN 内への導入が容易である (一つで多くの端末を監視できる)
- 今あるネットワーク構成を変えない

を満たすような新しい LAN 監視手法が求められている。

そのような監視手法として、本研究では ARP を用いた解析に着目した。

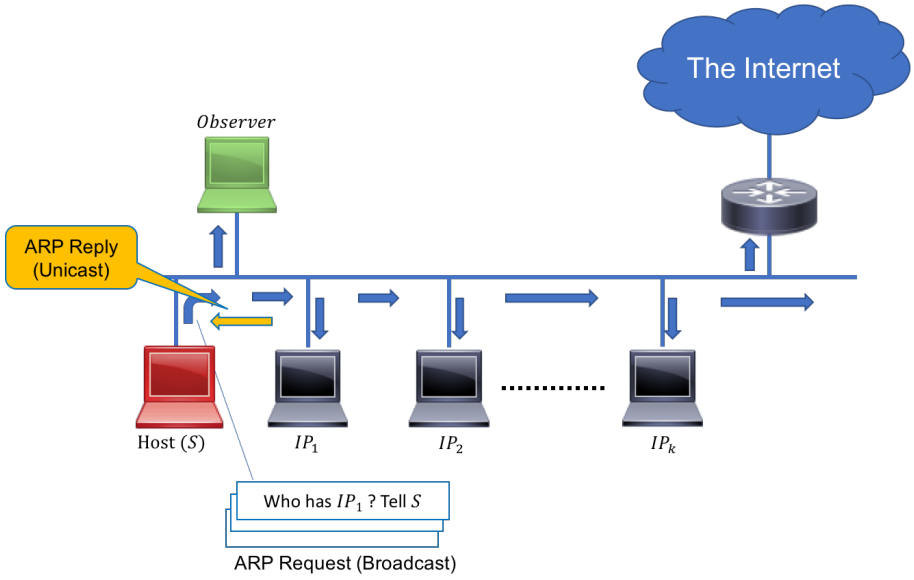


図 1.1. ARP の挙動

1.2　ARP について

同一の LAN に接続された端末同士が TCP/IP を用いて通信するためには、両者が相手の IP アドレスと MAC アドレスの両方を把握している必要がある。ARP(Address Resolution Protocol) はそのような IP アドレスから MAC アドレスを知るためのプロトコルであり、入手した IP アドレスと MAC アドレスとの対応関係を記録することで LAN 内の通信を可能としている。

ARP パケットは ARP リクエストと ARP リプライの二種類に大別される。図 1.1 において、Host が IP_1 との通信を試みているとする。まず Host は LAN 全体に ARP リクエストをブロードキャストする。ARP リクエストパケットには目標となるデバイスの IP アドレス情報が内蔵されており、ARP リクエストを受け取ったデバイスは全員自分のアドレスとパケットの IP アドレス情報を比較する。両者が一致した IP_1 は自分の MAC アドレス情報を載せた ARP リプライを Host 宛にユニキャストし、お互いが IP アドレスと MAC アドレスとの対応関係を ARP テーブルに保存して初めて以降の通信が可能となる。

このように、ARP は LAN 内の通信状況を探るための貴重な手がかりであり、ARP パケットの解析は LAN セキュリティを考える上で重要な役割を果たすことが期待される。

1.3　LAN の監視技術

さて、ここまで ARP の仕組みと LAN 解析における ARP パケットの有用性について述べてきたが、LAN 内の通信を監視する方法としては ARP に限らず全てのトラフィックを取得

しそれを解析するという手法も考えられる。しかしこの手法は現実的に様々な問題を孕んでいる。

まず第一に LAN 内の全トラフィックを監視することにはプライバシーの問題が発生する。ARP だけを見る場合とは異なり LAN 内のトラフィックには第三者に見られてはならない情報を含むものも当然存在し、セキュリティ向上のためであってもそれらを他人が監視することについては大きな議論の余地がある。

第二に、例えば大規模ネットワークでは全てのトラフィックを解析することはそもそも不可能である。そのためパケットの一部をサンプリングして解析する手法を採用する必要があるが、当然その場合異常なパケットを見落とす可能性が発生する。さらに仮にサンプリングした場合であっても継続的な監視には十分な計算リソースとストレージが必要であり、本研究で用いる手法と比較して大量のリソースを要求してしまう。

最後に、これらのトラフィックを取得するためにはそもそもネットワーク構成を変える必要があるという手間もある。

以上の問題を鑑みても、ネットワーク構成を変えず容易に導入できる本研究は現実的な解析手法であると言える。

1.4　目的

本研究では ARP リクエストを解析することによる新しい LAN 内異常検知手法を提案する。ARP による検知は、既存のネットワークの構成を変える必要がなく、さらに監視デバイスの導入が容易であるという利点がある。この ARP を LAN 内の異常検知として採用するため、本研究では LAN 内に流れる ARP の特徴量を抽出し、それを基に日々の傾向を分析して翌日の値を予測するフィッティングモデルを作成する。このフィッティングモデルは日々の傾向を学習するため動的なモデルである必要があり、さらに誤検知を抑えるため一度経験した大きな値の変化を記憶しておく必要がある。

そして外部期間で実際に運用されている LAN のデータを収集し、そのデータについて提案手法を適応する。それにより本手法によって実際にセキュリティリスクとなるような Malicious な通信を検知できることを確認し、本研究の有用性を示す。

1.5　貢献

ARP を用いた LAN 内異常検知モデルの作成のために、本研究ではまず ARP リクエストの送信先や送信間隔などの特徴量に着目する。その上で本研究においてはそれらの特徴量に対して移動平均をとり、時間変化に対応した動的なモデルを作成する。

そしてそのモデルのパラメータフィッティングのため研究室内のネットワークにおける ARP リクエストのデータを分析し、我々の目指すフィッティングモデルとして最も適当であると考えられるパラメータを決定する。

最後に調整されたモデルで実際に拡散マルウェアへの感染が疑われる LAN に対して異常検

知を行い、マルウェアに汚染している可能性の高いデバイスを適切に検知できている例を確認する。

第 2 章

関連研究

2.1　LAN 内のセキュリティについて

序論にて LAN 内のセキュリティが現在問題となっており、新しい異常検知技術が必要となっていることについて述べた。この項ではその裏付けとなるような研究について述べる。

昨今の情報化社会の発展に大きく寄与し、かつあまりに急速な浸透のためずさんな管理が問題となっている技術として IoT が挙げられる。各研究者は攻撃者が IoT デバイスを狙う理由の一つとして、IoT デバイスをボットネットの一部として取り込み [7, 8]、別の真の攻撃先に対して DDoS 攻撃をはじめとしたサイバー攻撃を仕掛けるための糧とするためであると考察している [9, 10]。そのようなマルウェアとしてもっとも危惧すべきものの一つとして、2016 年 8 月に MalWareMustDie によって発見された「Mirai」があり [11]、Antonakakis らをはじめとして様々な研究者が調査や対策法を提案した [12]。また Kolias らはこういったマルウェアが IoT デバイスを標的の対象として選んだ理由として 5 つの要素を挙げ、IoT デバイスへのセキュリティ管理の甘い現代社会へ向けて警鐘を鳴らした [13]。

このようなマルウェアが現代社会において大きな脅威をもたらしているという現状が、今日 LAN 内のセキュリティがますます注目を集めている理由の一つである。

また Kiravuo ら [14] は LAN 内技術の中でも特に Ethernet 技術に着目し、そのシンプルさと設定の容易さから常に危険を孕んだ技術であることを指摘している。彼らは Ethernet 関連の脆弱性に関して、

- Ethernet の不安定さを受け入れ、ファイアウォールでそれら全体を覆うこと
- スイッチとエンドホストとを論理的に分断すること
- 認証式のアクセスコントロール

の三つの解決策を提示しているが、これらのどの手法についても Ethernet に適したシンプルさと強固なセキュリティとを同時に満たすものであるとは言えないと結論づけられており、LAN 内セキュリティ問題の難解さを示している。

McHugh ら [15] は現代の WLAN 技術のセキュリティに注目した。彼らは現在主要な Wi-Fi 規格であるところの IEEE802.11ac について焦点を当て、この規格に関して脆弱性検査を

行った。そしてパケットキャプチャによって入手したこの規格におけるトラフィックを既存の IEEE802.11 規格でのトラフィックと比較し、既存規格と比べてセキュリティに関する新たなプロトコルは導入されていないと結論づけた。従って IEEE802.11 における WEP-PSK や WEP にある脆弱性は依然として残ったまま世間で運用されていることとなり、LAN 内のセキュリティに関する迅速な対応の必要性を裏付けている。

Waliullah ら [16] もまた WLAN 技術の脆弱性について警鐘を鳴らしている。彼らは IEEE802.11 のセキュリティに関する未解決問題をリストアップし、それに基づいてオープンソフトウェアツールによる脆弱性検査を実際に行った。そして彼らはそれらの攻撃を分析した上で解決策を提示しようと試みているが、最終的に適切なソリューションを提案するには至らず、解決策を見つけるための機会を提供するに留まっている。彼らは将来 WLAN に対してより複雑な攻撃が行われ、Wi-Fi 環境がユーザやビジネス環境にとってさらに危険な存在になりつつあると警鐘を鳴らしている。

2.2　ネットワークセキュリティに関する諸手法

ネットワークの脆弱性に関する問題が重要視されている昨今、そのセキュリティを強化するための手法を提案する研究が多く存在する。

Fiore ら [17]、Javaid ら [18]、He ら [19]、Taylor ら [20] はともにネットワークに対して機械学習を用いた異常検知手法を提示している。特に Fiore らはネットワークトラフィックに対して制限ボルツマンマシンを使用する手法を提案しているが、それに際した効果的なモデリングのために必要な学習データの生成について以下に示す二つの課題が存在するとしている。

- ネットワークトラフィックは非常に複雑で予測が困難であること
- ネットワークの異常は絶えず進化し、モデルは時間とともに変化しうること

これらの理由から彼らはネットワーク異常検出の効果的なモデルの望ましい特性は、変化に適応しその動作を複数の異なるネットワーク環境に一般化する能力 (自己学習システム) を有していることであると定義している。彼らの手法は機械学習的アプローチであるが、後に我々の提案するフィッティングモデルは上記のモデル定義における要件を正に満たしているものであり、我々のアプローチの正当性を裏付ける研究であると言える。

またセキュリティ問題の中でも我々と同様 LAN 内の異常検知に特化した研究も行われている。

Kolbitsch ら [21] はマルウェアのプログラムを学習し、マルウェアが侵入したかどうかを判断する検出アプローチを提案した。しかし彼らのアプローチはエンドノードで実行されるため、現在各地に設置されているデバイス一つ一つにこの手法を導入することは現実的でないと考えられる。

LAN セキュリティを題材とする研究の中には、我々の研究と同様に ARP に対して焦点を当てたものもある。

[22] で紹介されている機器の中には、ルータに接続するだけで内部ネットワークを監視し外

部からの攻撃を遮断するものがある。この機器は本研究と同じく ARP を利用したシステムにより動作しているが、この機器は ARP スプーフィングによって ARP テーブルを書き換えることで通信を監視し、不適切な通信を遮断している。しかし企業などで用いられているネットワーク機器には ARP スプーフィングを検知しそのパケットを遮断するものも存在するため、そもそもこの機器を採用できないというケースも存在する。一方本研究による手法はスプーフィングによるものではないため、そのようなバッティングのリスクなく各 LAN へ導入することができる。

Pandey[23] や Jinhua ら [24] 、Hou ら [25] もまた ARP を中心とした LAN セキュリティに関する研究を行っている。しかしながらこれらの研究は ARP スプーフィングに対する対策を講じることを重視しており、LAN 内で発生しうるその他の異常を検知するための手法ではない。従ってこれらの研究をマルウェアの検出を目的として採用することは困難であると言える。

2.3　トラフィック解析手法

LAN 内の異常を検知するための手法として、本研究ではトラフィック解析を用いている。本研究での提案は毎日のトラフィックを分析しその傾向を学習するフィッティングモデルを作成して異常検知を行うというものであるが、同様にトラフィックを別の手法で解析しマルウェアの検知として利用しようとする試みも多く見られる。

Wang ら [26] は、人工知能の観点からトラフィックを分類する新しい手法について提案を行っている。彼らはトラフィックデータを画像として取り込み、畳み込みニューラルネットワークを使用することでマルウェアの侵入したトラフィックを分類するという手法を提示した。しかし彼ら自身も指摘している通り、この手法は既知のマルウェアトラフィックにのみ対応した手法であって、未知のマルウェアトラフィックに対してそれが異常であると判断する機能を持ち合わせていないという欠点がある。

Stevanovic ら [27] はネットワークトラフィックを用いてボットネットを識別する手法について論じている。彼らはボットネットで発生する TCP、UDP、DNS の三つのトラフィックを想定し、教師あり学習によってそれらを通常のトラフィックと区別する手法を提案・実装した。結果としてその手法は高い検知率で通常のトラフィックとボットネットが生成したトラフィックを分別することに成功している。しかしこれらの手法は新しいデータによって定期的に学習結果を上書きする機能が実装されていないという難点がある。

第 3 章

提案手法

コンピュータネットワーク技術が日々発展・複雑化しつつある現代において、既存のネットワークに変更を加えることなく簡単に導入できる新しい LAN 内異常検知手法が求められていることを序論で述べた。本研究においてはそのような手法として ARP リクエストパケットに着目した異常検知手法を提案する。

この章ではまず ARP リクエストパケットの持つ固有な性質と、その性質を用いて本研究独自に定義した特徴量について述べる。次にそれらの特徴量を解析し、翌日の ARP リクエストの量や頻度を予測するフィッティングモデルを作成する。

3.1 ARP リクエストの性質と特徴量

監視デバイスから回収できる ARP リクエストパケットは、以下のような性質を持っている。

ブロードキャストパケットであること

あるデバイスから他のデバイス宛てに送られる ARP リクエストは、LAN 内の他のデバイスも受け取ることができる。これによりハニーポットと直接通信しないデバイスであっても、その通信の様子を解析することができる。

通信の指標となること

あるデバイスが LAN 内の他のデバイスと通信する際には、定期的に ARP を送り合いお互いの IP アドレスと MAC アドレスとを更新し続ける必要がある。したがって ARP リクエストはその端末が LAN 内でどのような通信を行っているのか、あるいはこれから行おうとしているのかの指標となる。

ARP スキャンに用いられること

ARP スキャンはその名の通り LAN 上の全ての IP アドレス宛てに ARP リクエストを送信し、どのホストから ARP リプライによる返答があったかを調べることで、LAN 内で起動しているホストの一覧を得る手法である。これは必ずしも悪意ある挙動であるとは言えないが、WannaCry をはじめとする多くのマルウェアはこのスキャンを利用して新たな標的を探すため、ARP リクエストの解析はそれらを検知するために重要な

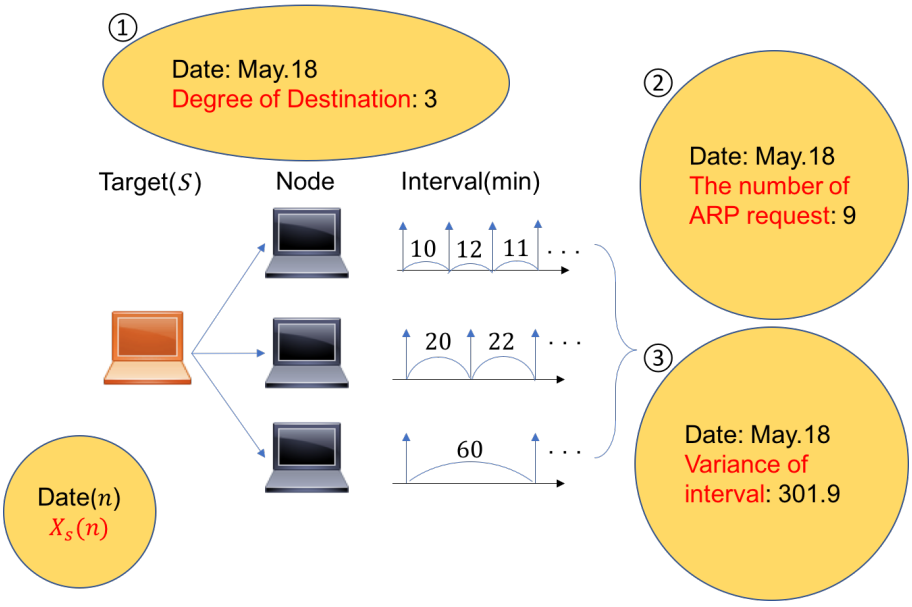


図 3.1. フィッティングモデルにおける特徴量

役割を果たすと考えられる。

以上のような ARP リクエストの性質を踏まえ、本研究においては端末異常検知のために幾つかの観点に着目した。本論文においては以降これらの観点のことを「特徴量」と定義する。本研究において着目した特徴量について図説したものが図 3.1 である。ここでは ARP リクエストを送信したデバイスを Target(S) とし、LAN 内の他のデバイス ($=S$ から ARP リクエストを受け取るデバイス) を Node としている。また S があるデバイスに対してどのくらいの間隔で ARP リクエストを送っていたかを Interval(送信間隔) として表している。例えば図における一番上の Node は、最初に ARP リクエストを受け取ってからそれぞれ 10 分、12 分、11 分の間隔を開けて次の ARP リクエストを受け取ったことがわかる。

本研究においては、特徴量について以下の 3 つを挙げ、それらの観点に基づき解析を進めていく。

通信相手数 (Degree)

Target$S$ が一日にいくつの Node に対して ARP リクエストを送った (通信を試みた) かを表した数である。IoT デバイスなど人の手を介さず自動で通信が行われるデバイスにおいては、通信を行いデータをやりとりする相手がほぼ固定されている傾向にある。従ってこの特徴量がある日を境に急激に増減している場合、マルウェアに感染した端末がスキャン攻撃を行ったり通信を遮断したりしている可能性がある。図においては、S は 1 日の間に 3 つの Node に ARP リクエストを送信したことがわかる。

ARP リクエストの総数 (Number)

S が一日に送った ARP リクエストの総数である。ある端末から一日に送信された ARP リクエストの総数が急激に増減した場合、通信相手数の項目で述べたケースと同様にスキャン攻撃や通信遮断が発生している可能性がある。しかし通信相手数のみを特徴量として考え異常検知を行った場合、ある一つの相手に対して非常に多くのリクエストを投げ続けているという異常ケースは検知できない。逆に ARP スキャンの検知に関しては、リクエストの総数よりも通信相手数を見たほうがより正確である。従ってこれら二つの特徴量を同時に考慮した分析を行うことで、そのようなケースも逃さず検知できるようにしている。図においては、S から各 Node に送信された矢印の総数がこれに該当し、当日の総数は 9 つであったことが示されている。

通信間隔の分散 (Variance)

これは観測日の Interval の分散をとったものである。人為的に新たに他のデバイスへ通信を試みない限り、端末同士が ARP パケットをやりとりする時間間隔はある程度決まっている。そのため (マルウェアに感染していない) デバイスの Interval に関して、その分散をとった値も日によらず安定した値を取ると考えられる。逆に分散の値が急激に変化すると、どこかへ送信しなければならないはずの ARP リクエストがうまく送信されていなかったり、もしくは何らかのノイズが入っていたりすることが想定される。図においては観測された Interval 全ての分散を取った 301.9 という値が分散となり、この値が日によって大きく変化しないことを期待する。

3.2　フィッティングモデル

前項では ARP リクエストの性質について触れ、それを踏まえて解析に際して着目すべき特徴量を挙げた。次にそれらの特徴量をどのように利用し、LAN 内の異常を検知するためのモデルを構成するかを考える必要がある。この課題を解決するため、本研究では「フィッティングモデル」という異常検知モデルを提案する。

フィッティングモデルはその名の通り検知したい当日の諸特徴量が平常時の値に「フィット」しているかどうかを表すモデルである。このモデルは前日までの特徴量を基に当日の値の範囲として適切であるような「幅」を提示し、その内部に当日の値がフィットしているかどうかを検査する。すなわちこれまでの傾向を基に学習された幅を平常時の運転を表すための指標とし、特徴量がその幅に収まっていればその日の通信は平常通りであると定める。逆に特徴量がこの幅から外れた値を取った場合、通信の異常が起きている可能性があるとしてアラートを鳴らす。図 3.2 ではフィッティングモデルの例として Degree を特徴量とし、それを元に異常検知を行う仕組みを示している。

まず予め得られた前日までの特徴量 ($X_s(n)$) を元に、フィッティングモデルは次の日の特徴量の値の範囲 (幅) を予測する。そしてその範囲と実際の当日の値が照合され、幅の内部に収

まっていれば正常、そうでなければ異常であるという結果を返す。そしてこれを様々な特徴量を用いて行い結果を比較照合することで、より強力な検知を目指す。

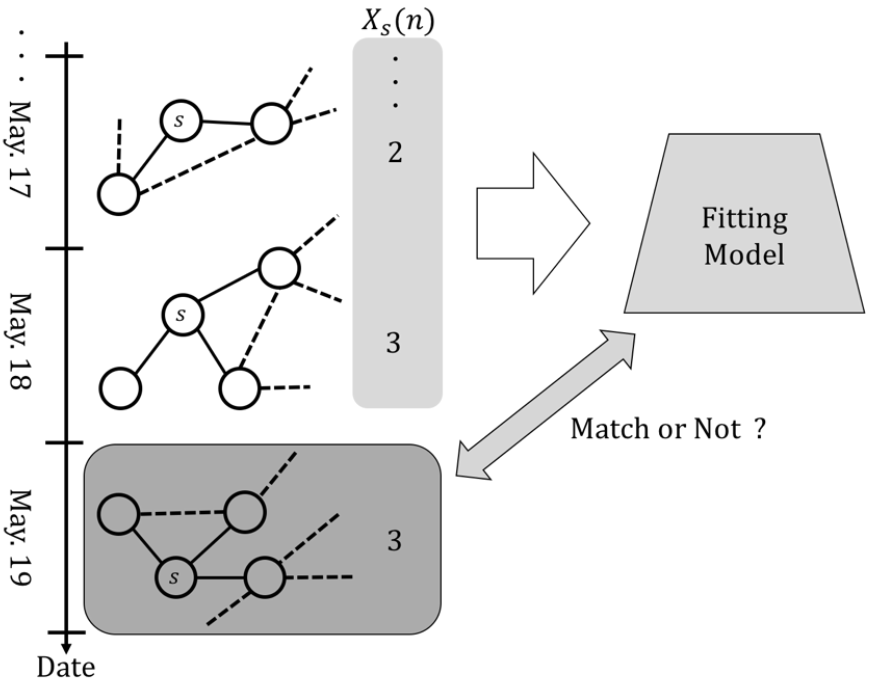


図 3.2. フィッティングモデル

3.3　フィッティングモデルを構成する要件

前項で述べたフィッティングモデルを構成する際には、異常を的確に検知するために考慮すべき点が複数存在する。本研究においてはその要件として以下に挙げる 3 つを挙げ、それらを全て満たすものを検知に用いるモデルとして妥当であるとする。

自動的に学習する機構を備えていること

本研究におけるフィッティングモデルは当日までの値から次々と翌日の幅を更新し続けなければならない。従ってモデルは適切に定式化され、自動でアップデートされ続ける必要がある。

時間変化に追随していること

フィッティングモデルでの検知に使用される正常値の幅は静的なものであってはならない。LAN 内のデバイスの構成は新しい機器の導入や古い機器の交換などが原因でしばしば変化するものであり、仮に予測値が時間変化を伴わない静的なものである場合、不審な通信であるとしてそのデバイスにアラートを鳴らし続ける恐れがある。従ってモデルが提示する予測値は、時間変化 (LAN 内の環境の変化) に対して適切に追随する必要がある。

一度経験した激しい値の変化を記憶していること

LAN 内のデバイスはその種類によって多種多様な ARP リクエストの特徴を示す。その中には毎日ほぼ一定間隔でどこかのデバイスにリクエストを送るものや数日おきに突然多くのデバイスと通信するもの、そもそも月に数回しか通信を行わず、ほとんど動いていないように見えるものなど様々な通信パターンが存在する。このような種々の通信パターンに対応したモデルを作成するためには、一度発生した激しい値の変化を記憶し、数日内に再び同じような値を観測した場合はそれを異常と判定しないような仕組みが必要である。

ここまでを元に、本研究における異常検知全体の流れを整理したものが図 3.3 である。まず検知の対象となる LAN 内に監視ノードを設置し、そこへ流れてくるブロードキャストパケットである ARP リクエストを回収する。それらのパケットデータはサーバに転送され、そこでフィッティングモデルによる異常検知が行われる。フィッティングモデルによる異常検知では、ARP リクエストパケットの特性を活かした 3 種類の特徴量を元にそれぞれ解析が行われる。モデルは時間変化に追随した動的なシステムであって、さらに通信パターンを学習して一つ一つのノードに焦点を当てて個別に検知する。そして解析の結果、ARP リクエストパターンが異常値を示していると判定されたノードに対してはアラートが返される。

なお本研究においては研究の都合上パケットデータを一度監視ノードからサーバへと転送してフィッティングモデルを作用させているが、実際の運用を考えた際にはその過程は必要なく、モデルを動かすプログラムは監視ノード上でも十分動かせることに留意する。

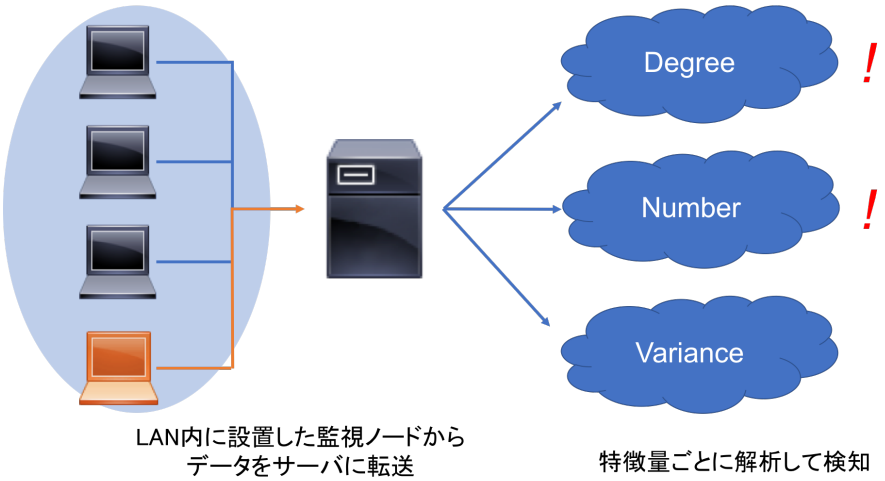


図 3.3. 本研究の流れ

3.4　フィッティングモデルの定式化

この節では、前節で挙げた要件を全て満たすようなフィッティングモデルの定式化を行う。図 3.4 は、本研究で我々が提案するフィッティングモデルを表している。まず $X_s(n)$ は特

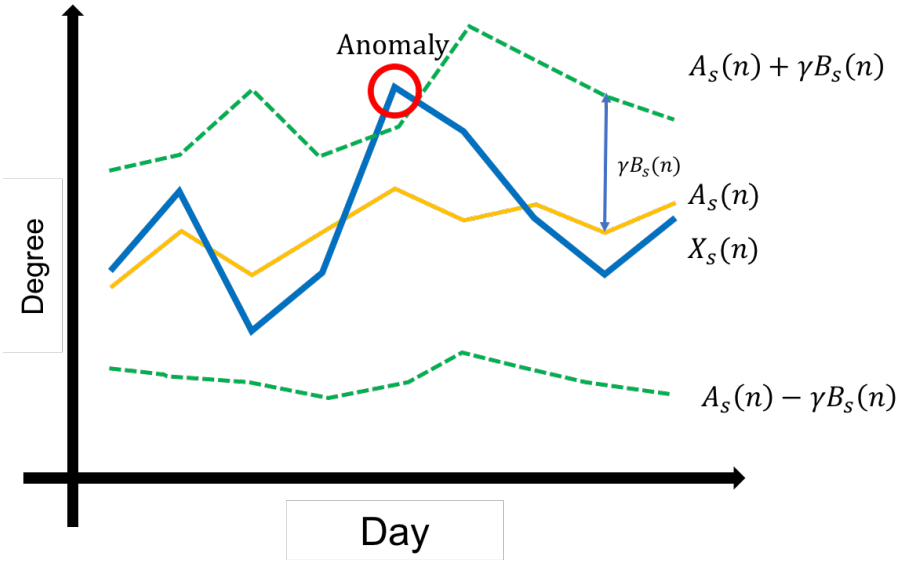


図 3.4. フィッティングモデルの定義

徴量の時間変化を表しており、観測した結果が毎日プロットされるようになっている。次に $A_s(n)$ は $X_s(n)$ の移動平均をとったものであり、フィッティングモデルにおける次の日の予測の核となる値である。ここで、$A_s(n+1)$ は $A_s(n)$ と $X_s(n)$ によって生成される。つまり、前日までの観測値の移動平均と当日の観測値をうまく組み合わせることで翌日の特徴量として適切であると考えられる数値を予測する。そして $\gamma B_s(n)$ はフィッティングモデルの「幅」であり、$X_s(n)$ は結局 $A_s(n) \pm \gamma B_s(n)$ の間にフィットしていることを期待される。$X_s(n)$ がこの二本の内側に収まっていればその日は正常な通信が行われていると判定し、そうでなければ何らかの異常が発生していると判定してその結果を返す。図 3.4 においては赤い丸の部分が二本の外側に出ているため、その日だけが異常な通信が発生した可能性があるとして提示される。

3.5　デバイスの分類

一つの LAN には非常に多くのデバイスが接続されており、デバイスによって用途が異なるため当然通信の頻度や通信先も異なる。図 3.5 は同一の LAN における二つのデバイスをフィッティングモデルにかけその結果を比較したものであり、両者の青いグラフは観測された特徴量をプロットしたものである。これにより左のデバイスは毎日 100 以上のデバイスとやりとりを行っているのに対し、右はごく稀に他のデバイスと通信し、ほとんどの期間はそもそも動いていないことがわかる。この二つのグラフを比較した際にもっとも意識すべき違いは、「デバイスが通信していない日を異常とみなすか否か」にある。左のデバイスに関して仮にどこかのタイミングで通信が途切れていれば当然異常とみなすべきであるが、右のデバイスがその日丸一日通信していなければそれはむしろ正常な挙動とみなせる。

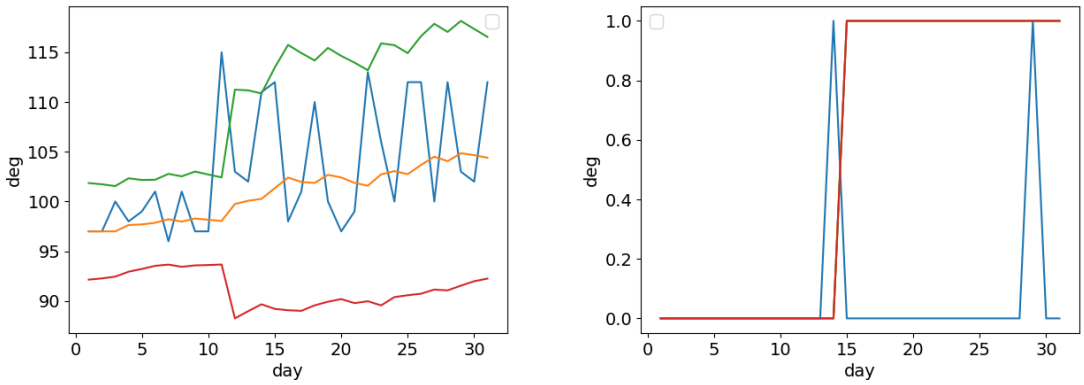


図 3.5. デバイスごとの挙動の差異

以上の理由により、本研究におけるフィッティングモデルはターゲットとなるデバイス群を通信の頻度によって

- Permanently-Connecting Device(PCD)
- Occasionaly-Connecting Device(OCD)

の二つの種類に分ける。

前者は IoT デバイスのように高頻度でどこかと通信しているデバイスであって、後者は滅多に起動しないデスクトップ PC のように月に数回程度しかパケットのやりとりを行わないデバイスである。PCD であると判定されたデバイスが丸一日通信を行っていない場合、フィッティングモデルはそれを異常値として検出する。一方 OCD の場合は通信が行われないという状態がデフォルトであるため、そのような日を正常な通信が行われていると判定する。

従ってフィッティングモデルの立式に際しては、先に述べた条件だけでなくこのようなデバイスによる異常検知の違いも考慮に入れる必要がある。

3.6　モデルの立式

前項までの要件を基に、本研究におけるフィッティングモデルを以下のように立式する。

$$A_s(n+1) = \alpha A_s(n) + (1-\alpha)X_s(n), A_s(0) = 0$$

$$Y_s(n)^2 = \{X_s(n) - A_s(n)\}^2$$

$$B_s(n+1)^2 = \beta B_s(n)^2 + (1-\beta)Y_s(n)^2, B_s(n) = 0$$

$$(0 < \alpha < 1, 0 < \beta < 1)$$

さらにフィッティングモデルはデバイスごとに「通信している期間としていない期間の比

率」を計算し、それを基にそのデバイスを PCD と OCD に振りわける。そしてそのデバイスが OCD の場合、上記の式に加えて「$X_s(n)$ が 0 の場合は各値を変更しない」という処理を加える。この処理を怠った場合 OCD において通信していない期間の $B_s(n)$(幅) がどんどん小さくなってしまい、結果 OCD がどこかと通信するたびに異常であると検知してしまう。また α と β は重み付けのための値であり、これを変化させることによってどの値を重視したモデルにするかを決めることができる。

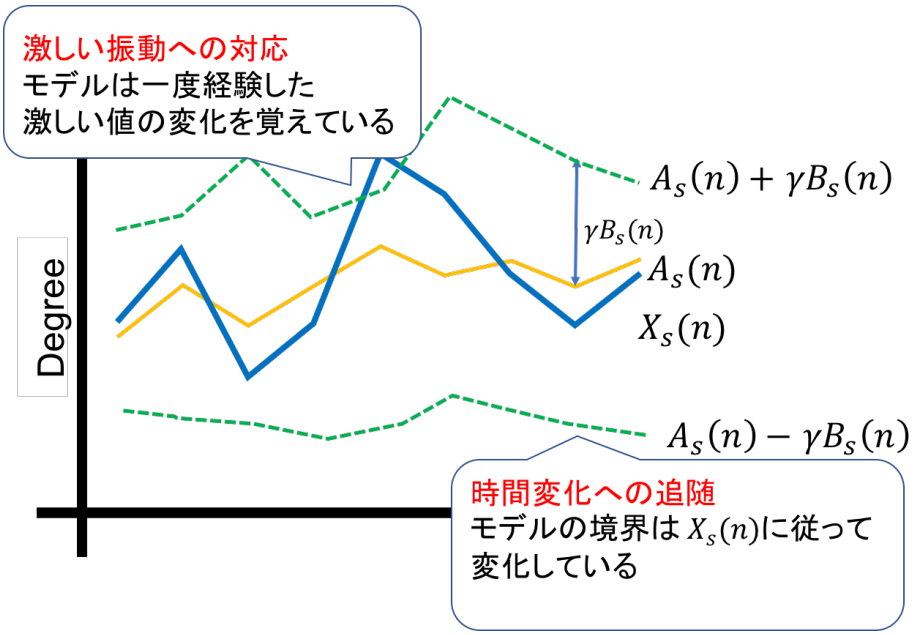


図 3.6. フィッティングモデルと諸条件との関係

最後にフィッティングモデルを上記の通り立式し、さらに諸パラメータを適切に設定できたと仮定した時、このモデルが先に示した諸条件を満たしているかどうかについて検討する。(図 3.6)

自動学習機構

これはフィッティングモデル全体が定式化され、前日までの結果から次の日の値の範囲を逐次更新し続けるモデルであることから明らかに満たされている。

時間変化の追随

フィッティングモデルにおける異常判別機構である $A_s(n) \pm \gamma B_s(n)$ は $X_s(n)$ の変化に従って時間変化している。これにより LAN の構成の変化などによって $X_s(n)$ に大きな影響があってもフィッティングモデルが機能しなくなるということはなくなる。

激しい振動への対応　フィッティングモデルがこの要件を満たしているかどうか確認するためには、前日のデータがフィッティングモデルにどのように反映されるかをみる必要がある。ここで $B_s(n)$ の定義式を再掲する。

$$B_s(n+1)^2 = \beta B_s(n)^2 + (1-\beta)(X_s(n) - A_s(n))^2$$

この式中の β は 0 以上 1 以下の実数であり、この値を適切に調整することで $B_s(n)$(以前までのデータ) と $X_s(n) - A_s(n)$(当日から得られるデータ) のどちらにウェイトを置くか定めることができる。そのため前日までのデータを強く参照するよう β の値を大きくとれば、数日前に起きた激しい値の増減をモデルに反映させることができる。

第 4 章

データセット

本研究におけるパラメータフィッティングで用いるデータセットとして、今回はラボ内ネットワークから ARP リクエストパケットのみを約二ヶ月間抽出したファイル (以下データセット 1 と呼ぶ) を準備した。ここでラボ内ネットワークと、その環境から得られる ARP リクエストの特徴について概説する。

今回用いるネットワークは、上流からルータを介して分離されたこのネットワークの下にさらに複数の NAT サーバが接続され、それらがまた各々のプライベートネットワークを構築している。そのため、単なる NAT 下のプライベートネットワークとは事情が異なる。ネットワークに割り当てられている使用可能な IP アドレスは計 126 であり、うち 80 個が固定 IP アドレス、残りが DHCP によって割り振られるアドレスである。80 個の固定アドレスの内訳は、ルーターが 1 台、メールサーバやウェブサーバなどのサービスサーバが 13 台、NAT サーバが 6 台、個人用が 56 台、残りが計算用などその他のサーバとなっている。

ここでデータセット 1 の ARP リクエストを送信者ごとに整理し、横軸をノード、縦軸を送信した数としてグラフ化したものが図 4.1 である。まずノード 1 はルーターであり、同ネットワークの他の機器が LAN 外と通信を行う際は必ずこの機器と通信を行う必要があるため、毎日非常に多くの ARP リクエストをやりとりしている。次にノード 2 や 13、40 は DNS やメールサーバなどのサービスサーバであって、これもまた毎日様々なノードと通信するため多くの ARP を送信していることがわかる。そして 32 や 65 は NAT サーバに該当し、これらのサーバも NAT 下のサーバとの通信の際のゲートとなるため比較的多くの ARP がやりとりされている。残りのほとんどの機器は個人用の機器であって、多くのリクエストを送信しているものからほとんど動いていないものまで不規律であることがわかる。

次にデータセット 1 中のある 1 時間をとり、横軸を時間、縦軸を送信ノードとして ARP リクエストが送られた時刻を散布図として表したものが図 4.2 である。これを見てもやはり 1 のルータや 2 の DNS は絶え間なく ARP リクエストを送り続けており、高頻度で他のデバイスと通信をしていることがわかる。また個人用デバイスにおいても、図 4.1 で確認したように高頻度で通信を行っているものから滅多に通信しないものまでまばらであることが見て取れる。ここでノードごとの送信間隔に着目してみる。例えば 65 の NAT サーバは、数分間隔で定期的にリクエストを送信していることがわかる。これは ARP テーブルの保持時間がある程度決

まっており、定期的にその更新のためにリクエストを送っているためであると考えられる。このように ARP リクエストの送信間隔も LAN におけるデバイスの特徴を決める重要なファクターであると言える。

以上のように、本研究において使用する環境は ARP リクエストという観点において多種多様な特徴を持つ機器の集まりであり、一部の研究用ネットワークのように通信が恣意的に統制されているものではない。そのような意味で本環境は現在社会において用いられる一般の LAN と同じような状況下にあり、パラメータフィッティングのためのデータセットとして用いるに遜色ないと判断した。

また後述するフィッティングモデルの評価で使用するデータセットについても同様にここで述べる。評価の際には外部機関で実際に運用されている LAN から監視ノードに流れていた 2 ヶ月分のトラフィックを回収したものをデータとして使用した。LAN の管理は外部機関に一任されておりその詳細な設置場所や仕様を把握することはできないが、入手したデータ (以下データセット 2 とする) を解析することで、その環境をある程度推定する。

まず、データ内で使用されている IP アドレスを集計することで、LAN 内で使用できる IP アドレスが高々 1000 程度であることがわかった。また同様に MAC アドレスを集計し、データ中少なくとも 1 回パケットを送信した MAC アドレスが計 2399 であることも判明した。さらにこのネットワークについても、研究室内のケースと同じように ARP リクエストを送信者ごとにグラフ化したものと ARP リクエストの送信時刻を散布図として表したものを作成した。その結果が図 4.3 と図 4.4 である。また図 4.3 に関して、このネットワークは不特定多数の端末が日替わりで確認されているため、端末 (の MAC アドレス) がデータセットに現れた順に番号を振りそれを横軸のノードとして採用している。そしてデータ中の MAC アドレスが非常に多いため図 4.4 では全端末のうち観測された順の早いものから 100 のノードを抽出して表示している。まず図 4.3 を見ることで、この LAN は図 4.1 と同じように様々な端末が稼働し ARP リクエストの送信数が非常に多いものからほとんど見られないものまで様々であることがわかる。次に図 4.4 の一部を見てみると、定期的な通信とそうでないものが混ざっていることがわかる。そして特定の時間帯のみ定期的な ARP 通信を行っている機器があることもわかり、ここから複数の機器がこのネットワークに出入りしていることが予想される。これとデータ中の MAC アドレスが IP アドレスと比較して多いことから、これは PC やスマホなどが含まれるユーザネットワークであり、その中でも特に不特定多数の端末が日替わりで接続する類のネットワーク (e.g. Wi-Fi) であることが推測される。ただし長期にわたり一定間隔で通信している機器もあるため、ユーザ機器ではない固定の端末も含まれている可能性が高いと見られる。

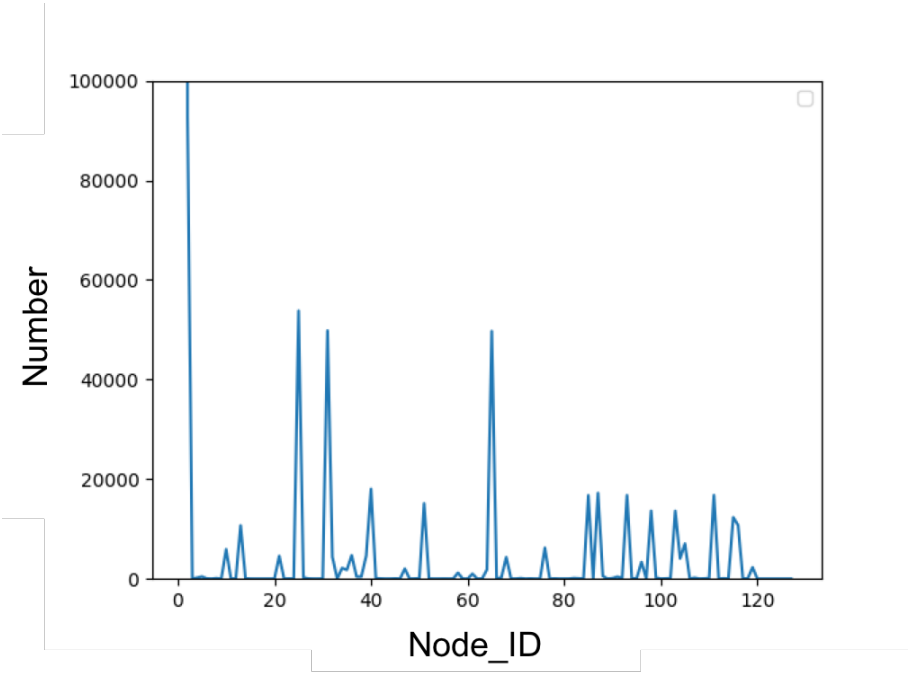


図 4.1. データセット 1 のノード別送信数

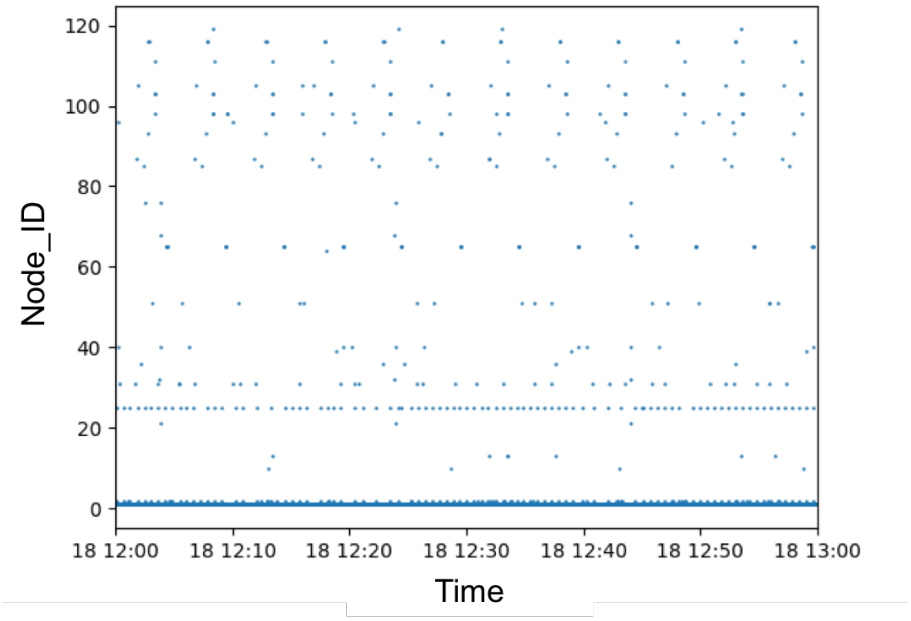


図 4.2. データセット 1 のリクエスト送信時刻の散布図

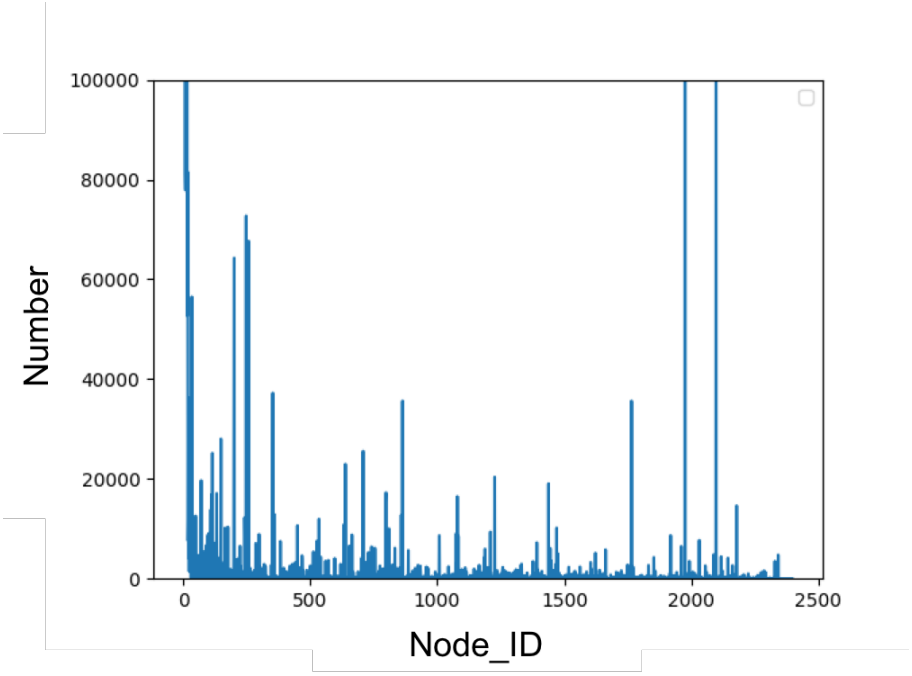


図 4.3. データセット 2 のノード別送信数

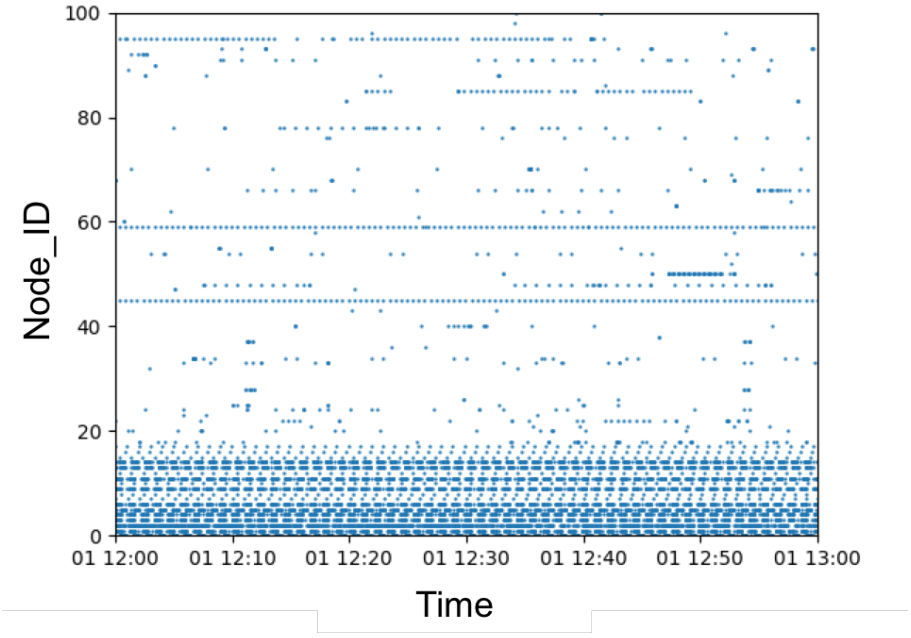


図 4.4. データセット 2 のリクエスト送信時刻の散布図

第 5 章

検証

5.1　実験目的と諸設定

先に定義したフィッティングモデルを用いて、3 種類の特徴量に基づいて実際の LAN 内を流れる ARP リクエスト群に対して異常検知を行いその結果を比較するという実験を行った。本実験ではデータセットとして前章に示したようにラボ内ネットワークの 2 ヶ月分の ARP リクエストパケット群を扱い、パラメータフィッティングによって異常検知モデルの挙動を制御することを考える。

5.2　パラメータチューニング

本節ではまず、フィッティングモデルの式中の諸パラメータについて議論する。まず PCD と OCD との境界を、「ARP リクエストを送った日数が月の 1/4 を超えているか」と定めた。つまり月の 1/4 以上デバイスが稼働していればそれは定常的に稼働している媒体であり、そうでなければ基本的に稼働していない媒体であるとした。次にモデルにフィットする「幅」の太さを決めるパラメータである γ を 2 と設定した。つまり $X_s(n)$ は $A_s(n) \pm 2B_s(n)$ の範囲にフィットしていれば正常な動作をしていることになる。

また今回のチューニングに先立って、観測開始から通信が行われた計 7 日間を学習期間と設定し、この期間を用いて $A_s(n)$ や $B_s(n)$ の値を安定させ検知するにふさわしいモデルとした。そして定義式の核となる α と β に関しては、さまざまな値について実際にフィッティングモデルを動かし、その精度を比較検討した。

モデルの式から、α と β はそれぞれ前日までの $A_s(n)$ や $B_s(n)$ の重み付けであることがわかる。従ってこの値が極端に小さい場合、以前までのデータはほとんど反映されず、ただ前日のデータを引き継ぐだけの性能として弱いモデルとなってしまう。そのため今回のチューニングにおいては (α,β) として (0.5,0.5),(0.9,0.9),(0.99,0.99) の 3 つのパラメータを採用し、それぞれのパラメータ下のもとでフィッティングモデルがどのように機能するかを比較した。

図 5.1 は LAN 内の 3 つのデバイスに関してそれぞれのパラメータを採用したフィッティングモデルによって異常検知を行った結果である。ただし IP_A と IP_B は PCD、IP_C は OCD

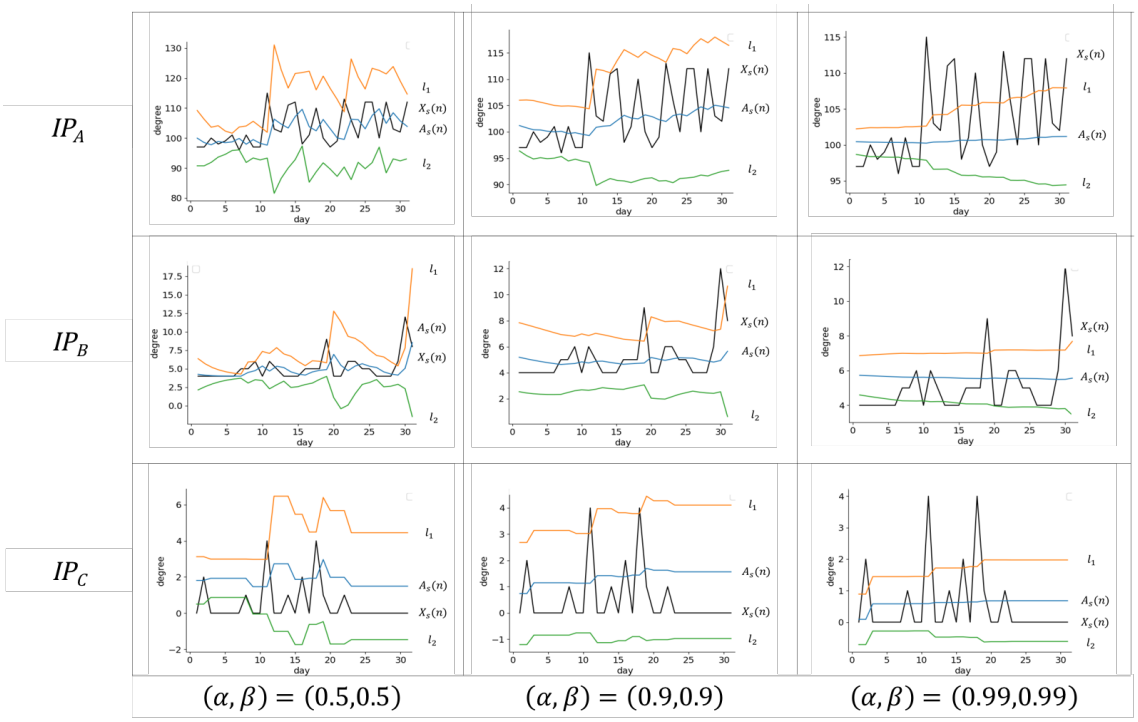


図 5.1. パラメータチューニング

であることがわかっている。

まず $(\alpha,\beta)=(0.99,0.99)$ の場合、$X_s(n)$ が大きな変化を取っている日であっても、どのデバイスにおいても $A_s(n)$ はほぼ一定の値を保っている。これは α の値が非常に大きいため、学習期間のうちに記憶された $A_s(n)$ の値が固定されているためであると考えられる。

一方 $(\alpha,\beta)=(0.5,0.5)$ の場合、フィッティングモデルは $X_s(n)$ の変化に非常に敏感に変化している。これは α と β の値が非常に小さいため前日の $X_s(n)$ や $A_s(n)$ の影響を強く受けすぎてしまい、結果以前のデータをほとんど記憶していていないようなモデルとなっているためである。

最後に $(\alpha,\beta)=(0.9,0.9)$ の場合、このパラメータを用いて動かしたフィッティングモデルが我々の期待通りに動作しているかどうかを、先に挙げた 3 つの要件に照らし合わせながら検討する。

自動的に学習する機構を備えているかどうか

これに関しては先に確認した通り、定式化されている時点で満たしている。

時間変化に追随しているか

該当する各グラフをみると、フィッティングモデルは滑らかにかつ $X_s(n)$ の激しい変化に追随するように動いていることがわかる。これにより他のパラメータにおける挙動よりもこの要件を満たした適切な動きをしていると言える。

一度経験した激しい値の変化を記憶しているか

この要件について議論するためには、図 5.1 における激しい変化に着目して議論する必要がある。3 つのデバイス全てにおいて、$X_s(n)$ の激しい変化を最初に観測した際これらの観測値は異常であるとして検知されている。これはフィッティングモデルがこの変化を記憶してしていないためであり、これまで予期していなかった激しい変化を検知したという意味で期待通りの挙動であると言える。一方、これ以降再び同程度の値が観測された時、フィッティングモデルは一度経験したこれらの値を記憶している。従って、IP_A のような大きな値の変化が引き続き観測されるようなケースにおいては、2 回目以降の値の変化については正常な値であるとして処理される。逆に IP_B のように一度大きな値を経験しても後日さらに大きな値の変化を経験した場合に関しては、後者の観測値を異常値として正しく検知できている。

上記の議論により、このパラメータはフィッティングモデルとしての要件を最も満たすように動作していることがわかる。よって以降の実験においては、$(\alpha,\beta)=(0.9,0.9)$ を採用する。

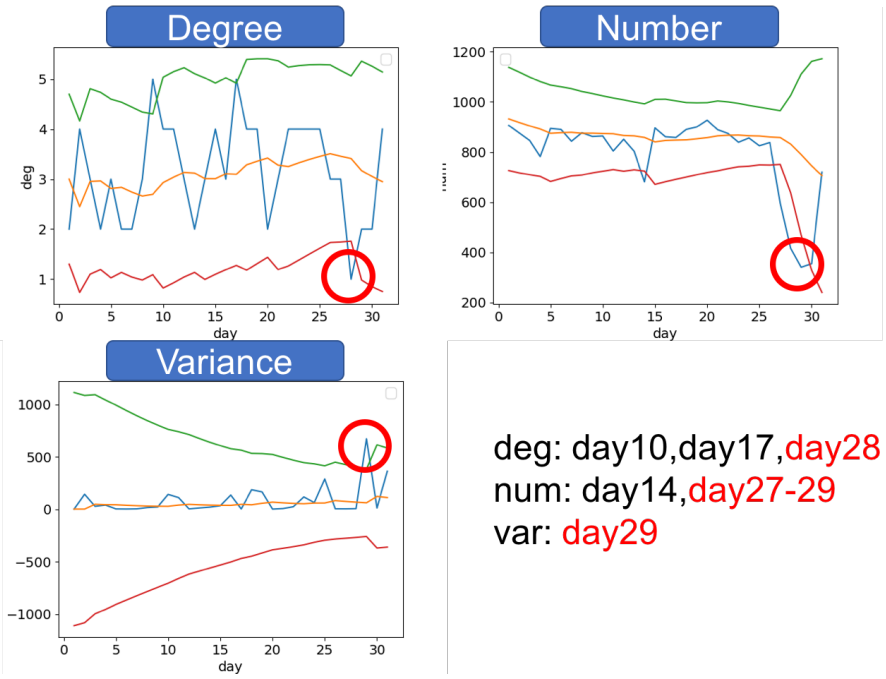


図 5.2. フィッティングモデルにおける異常検知例 1

5.3　実験結果

図 5.2、図 5.3 はある端末 2 つに関するフィッティングモデルの計測結果をグラフ化したものである。横軸は日付、縦軸は特徴量を表している。まず図 5.2 では、3 つのモデルがほぼ同タイミング (28〜29 日) で異常判定を行っていることが読み取れる。特に通信間隔の分散においては、普段の $X_s(n)$ がたかだか 200 程度の低い値を保っているのに対し、異常判定時は 700

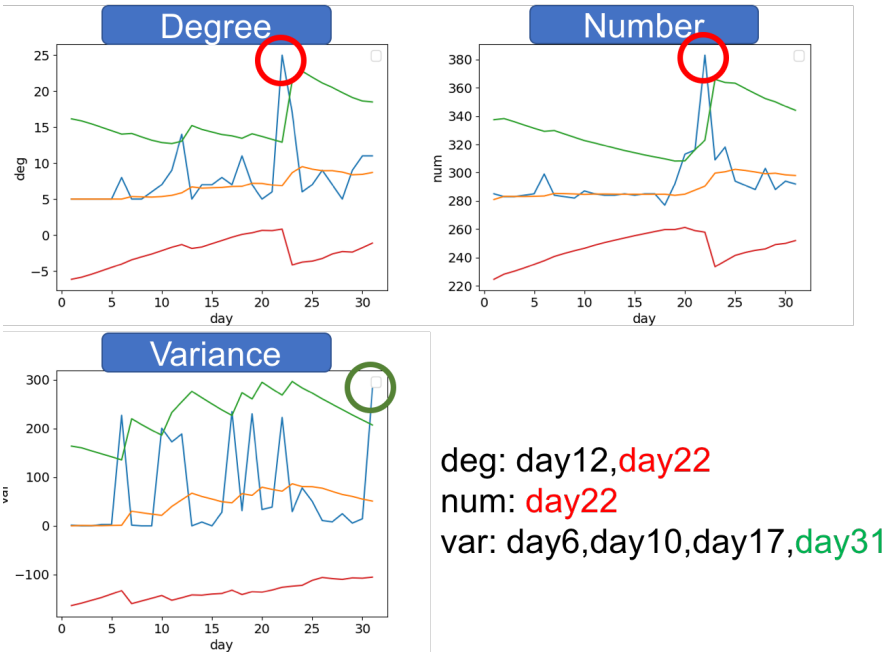


図 5.3. フィッティングモデルにおける異常検知例 2

前後の高い値を記録しており、普段とは異なる異常な通信が行われた可能性が高い。このように、複数の特徴量によりフィッティングモデルを構築しそれらを比較することで、より高い精度の検知が期待できる。そして図 5.3 は、それぞれの特徴量で異常であると判定した日付が異なる例である。3 つのフィッティングモデルのうち通信相手の数と ARP リクエストの送信数のモデルからは 22 日が同時に検出されており、この日付は高確率で何らかの異常が発生したことが推測される。一方通信間隔の分散のモデルを見ると、この端末では $X_s(n)$ が 50 以下の値と 200 前後の値とを数日おきに繰り返すような通信パターンをとっていることがわかる。従って数回検知されている 200 前後の値に関しては異常である可能性は低いと考えられるが、31 日については前例にないほど高い値をとっているため、異常が発生したと断定することはできないにせよ警戒すべきポイントであると考えられる。

5.4　考察

前述の実験結果において、フィッティングモデルは幾つかの日付に関して異常判定を返した。ここで、異常判定を返されたラボ内の端末の該当する日付が実際に何らかの異常を発生させていたのかについて検討していく。

まず図 5.2 であるが、これはラボ内のとあるユーザが管理している端末である。この日のリクエストを詳しく調査すると、この端末は普段は数分間隔で特定の 2 人の相手と必ず通信を行っており、その他は日によって 1〜3 の端末と一日に数回の通信を行う (そのため通信相手の数が 2〜5 に変化している) という通信パターンをとっていることがわかった。ここでまず

28 日を確認すると、この日は普段常時通信している端末のうち片方がダウンしており、それによって通信相手数とリクエストの数との減少が同時に起きていることがわかった。また 29 日は、この端末自身が 8 時から 9 時半までの間通信を全く行っていないことがわかった。これにより ARP リクエストの総数の減少と、端末が落ちている間の送信間隔のずれによる分散の増大が起きていることがわかった。

また図 5.3 は、研究室内の証明書発行や DHCP の配布を行っているサーバである。このサーバの 22 日における挙動をログを用いて調べたところ、この日に外部からの不正な ssh ログイン攻撃を大量に受けていることがわかった。また 31 日は、普段の通信の他に DHCP で割り当てられている IP アドレスと通信を行っていることがわかり、このサーバが新たに DHCP を配布したために起きたノイズで分散が増えているものと推察された。

第 6 章

評価

6.1 目的

前章において、本研究におけるフィッティングモデルが適切なパラメータフィッティングのもと異常を検知し、LAN 内異常検知モデルとして機能しうることを示した。

この章では LAN 内拡散マルウェアに感染した疑いの高いネットワークに対して提案手法を適応し、フィッティングモデルが実際にマルウェアの侵入したデバイスを検知できているという例を示すことで、本研究の有用性を示す。

6.2 事例データについて

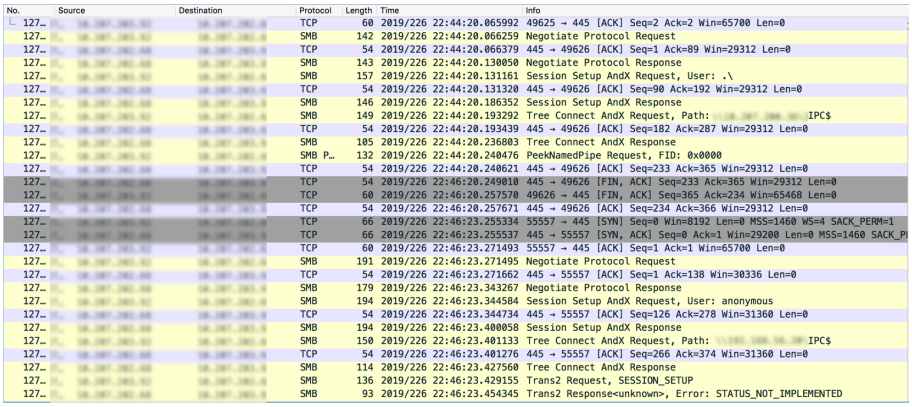
本研究で使用する事例データは、実際に運用されている LAN に監視ノードを設置しそこへ流れてきたパケット 2 ヶ月間集めたものである。そこで得たパケットを解析したところ、その中に LAN 内全体へ被害を拡散させるマルウェアが侵入している可能性が高いことがわかり、さらにそのマルウェアが WannaCry[28] によく似た悪意のある挙動を示していることがわかった。以下その類似性について概説する。

そもそも WannaCry とは、Microsoft Windows を標的としたワーム型のランサムウェアである。端末が WannaCry に感染すると、その内部のデータが暗号化されアクセスできなくなってしまう。そしてマルウェアはその暗号の解除と引き換えに端末の所有者へ金銭を要求しながら、周囲の端末へさらに感染を拡大させる。今回の事例ケースは、その感染拡大パターンに関して WannaCry との類似性が見られることがわかっている。

WannaCry の拡散手法は、Windows の Server Message Block(SMB) プロトコルの脆弱性である「EternalBlue」を突いたものであることが知られている [29]。攻撃のための SMB セッションは主にポート 445 を用いて行われ、一連のステップを経て他の端末へと感染する。ここで図 6.1 は、本ケースで扱う事例データにおいて発見された SMB セッションの一部を示したものである。ポート 445 によって行われるこの一連のやりとりはまさに WannaCry の拡散手法と一致しており、この LAN はマルウェアに感染している可能性が極めて高いと考えられる。

しかし逆にランサムウェアの侵入がわかったとしても、厳密な意味では現状このマルウェア

が WannaCry であると断定することは難しい。現実のウイルスと同様に元が一つのマルウェアであっても時間が経つにつれ様々な亜種が登場するという現象はしばしば起こりうるものであり、実際に (たとえば WannaCry に見られる警告表示であるような) 被害がでない限りはあくまで「LAN 内拡散マルウェア」という名称を取るしかない。しかしながらパケット解析によりそのマルウェアの特徴を捉え、既存のマルウェアとの類似性を探ることは、これからくるであろうさらなる攻撃を予測し感染拡大を防ぐための大きな手がかりとなる価値の高い探索であると言える。



No.	Source	Destination	Protocol	Length	Time	Info
127…	[redacted]	[redacted]	TCP	60	2019/226 22:44:20.065992	49625 → 445 [ACK] Seq=2 Ack=2 Win=65700 Len=0
127…	[redacted]	[redacted]	SMB	142	2019/226 22:44:20.066259	Negotiate Protocol Request
127…	[redacted]	[redacted]	TCP	54	2019/226 22:44:20.066379	445 → 49626 [ACK] Seq=1 Ack=89 Win=29312 Len=0
127…	[redacted]	[redacted]	SMB	143	2019/226 22:44:20.130050	Negotiate Protocol Response
127…	[redacted]	[redacted]	SMB	157	2019/226 22:44:20.131161	Session Setup AndX Request, User: .\
127…	[redacted]	[redacted]	TCP	54	2019/226 22:44:20.131320	445 → 49626 [ACK] Seq=90 Ack=192 Win=29312 Len=0
127…	[redacted]	[redacted]	SMB	146	2019/226 22:44:20.186352	Session Setup AndX Response
127…	[redacted]	[redacted]	SMB	149	2019/226 22:44:20.193292	Tree Connect AndX Request, Path: [redacted]IPC$
127…	[redacted]	[redacted]	TCP	54	2019/226 22:44:20.193439	445 → 49626 [ACK] Seq=182 Ack=287 Win=29312 Len=0
127…	[redacted]	[redacted]	SMB	105	2019/226 22:44:20.236803	Tree Connect AndX Response
127…	[redacted]	[redacted]	SMB P…	132	2019/226 22:44:20.240476	PeekNamedPipe Request, FID: 0x0000
127…	[redacted]	[redacted]	TCP	54	2019/226 22:44:20.240621	445 → 49626 [ACK] Seq=233 Ack=365 Win=29312 Len=0
127…	[redacted]	[redacted]	TCP	54	2019/226 22:46:20.249010	445 → 49626 [FIN, ACK] Seq=233 Ack=365 Win=29312 Len=0
127…	[redacted]	[redacted]	TCP	60	2019/226 22:46:20.257570	49626 → 445 [FIN, ACK] Seq=365 Ack=234 Win=65468 Len=0
127…	[redacted]	[redacted]	TCP	54	2019/226 22:46:20.257671	445 → 49626 [ACK] Seq=234 Ack=366 Win=29312 Len=0
127…	[redacted]	[redacted]	TCP	66	2019/226 22:46:23.255334	55557 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
127…	[redacted]	[redacted]	TCP	66	2019/226 22:46:23.255537	445 → 55557 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_P
127…	[redacted]	[redacted]	TCP	60	2019/226 22:46:23.271493	55557 → 445 [ACK] Seq=1 Ack=1 Win=65700 Len=0
127…	[redacted]	[redacted]	SMB	191	2019/226 22:46:23.271495	Negotiate Protocol Request
127…	[redacted]	[redacted]	TCP	54	2019/226 22:46:23.271662	445 → 55557 [ACK] Seq=1 Ack=138 Win=30336 Len=0
127…	[redacted]	[redacted]	SMB	179	2019/226 22:46:23.343267	Negotiate Protocol Response
127…	[redacted]	[redacted]	SMB	194	2019/226 22:46:23.344584	Session Setup AndX Request, User: anonymous
127…	[redacted]	[redacted]	TCP	54	2019/226 22:46:23.344734	445 → 55557 [ACK] Seq=126 Ack=278 Win=31360 Len=0
127…	[redacted]	[redacted]	SMB	194	2019/226 22:46:23.400058	Session Setup AndX Response
127…	[redacted]	[redacted]	SMB	150	2019/226 22:46:23.401133	Tree Connect AndX Request, Path: [redacted]IPC$
127…	[redacted]	[redacted]	TCP	54	2019/226 22:46:23.401276	445 → 55557 [ACK] Seq=266 Ack=374 Win=31360 Len=0
127…	[redacted]	[redacted]	SMB	114	2019/226 22:46:23.427560	Tree Connect AndX Response
127…	[redacted]	[redacted]	SMB	136	2019/226 22:46:23.429155	Trans2 Request, SESSION_SETUP
127…	[redacted]	[redacted]	SMB	93	2019/226 22:46:23.454345	Trans2 Response<unknown>, Error: STATUS_NOT_IMPLEMENTED

図 6.1. 事例データで発見された SMB セッションの例

6.3　事例データにおける Malicious な挙動

前述のネットワークに対して本研究における提案手法を適応したところ、検知中に確認された ARP パケットの送信元 MAC アドレスは合計 2399 であり、そのうちそれぞれの特徴量における検出数は以下の表 6.1 の通りとなった。

表 6.1. 特徴量ごとの異常検知数

特徴量	検知数
通信相手数	2178/2399
通信総数	2171/2399
通信間隔の分散	2063/2399
のべ検知数	2256/2399

この異常をさらに分析するため、本研究においてはさらにその異常とされたデバイスが「ポート 445 宛ての TCP パケット」が監視ノードに送っているかどうかを調査した。

前節でも触れた通り、ポート 445 は SMB セッションを張るために用いられるポートである。SMB セッションは普段はファイル共有やプリンタ共有などを行うためのプロトコルであるが、先に述べた脆弱性を持つため WannaCry を始めとするランサムウェアの拡散手法とし

ても用いられる。さらに監視ノードは本研究を行うために新たに導入されたデバイスであり、平常時においてパケット監視以外の特別な動作を行うことはないため他のデバイスと SMB プロトコルによる通信を行うことは考えられない。逆にこのようなパケットが監視ノードに対して送られてきた場合は、その送信元のデバイスは悪意のある挙動をしている可能性が高い (これらのパケットは ARP リクエストとは異なりブロードキャストではないため、LAN 内を流れる全ての TCP パケットを監視ノードが検知できているわけではないことに留意する)。よってこの特別な TCP パケットは、マルウェアの攻撃である可能性が極めて高い危険な兆候であると捉えてよい。

従って本研究においては、用いる事例データのうちこのようなパケットが見られたタイミングを「Malicious な挙動」であると定義し、フィッティングモデルが検知する Anomaly との相関を探る。また前章で使用したデータセットは研究室ネットワークであり、1 つの IP アドレスに対し 1 つの端末が割り当てられているため IP アドレスごとの検知を行っても問題なかったが、本章で使用するデータは必ずしも IP アドレスと端末が一対一対応であるとは限らないため、より正確に端末ごとの検知を行うため MAC アドレスごとの解析を行った。

異常であると判断されたデバイスのうち一度でもこのようなパケットを送信したことが確認されたアドレスは合計 11 であり、各々のアドレスにおける検知数は図 6.2 の通りであった。その中でも特に検知数の多い MAC_1 について、横軸を日付、縦軸を検知数としてグラフ化したものが図 6.3 である。このグラフから MAC_1 は数日おきに 4000〜8000 の Malicious なパケットを送り、他のデバイスへ感染を拡大させようと試みていることがわかる。またデータセットに含まれる MAC アドレスの中には、Malicous ノードに該当しながらどの特徴量を使用しても検知できていないデバイスはなかった。

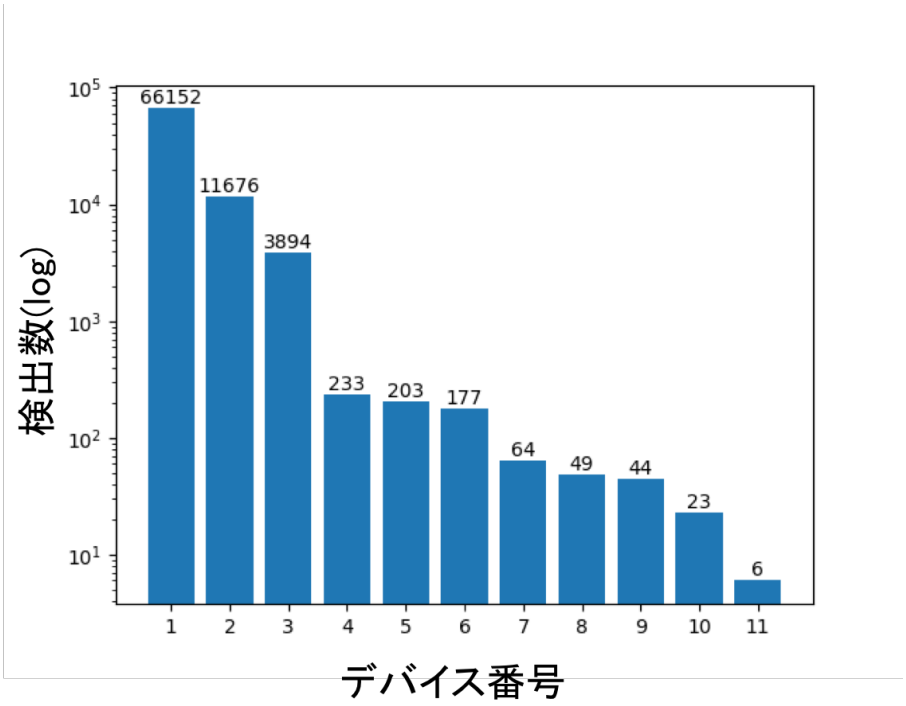


図 6.2. 事例データにおける Malicious な挙動の検知数

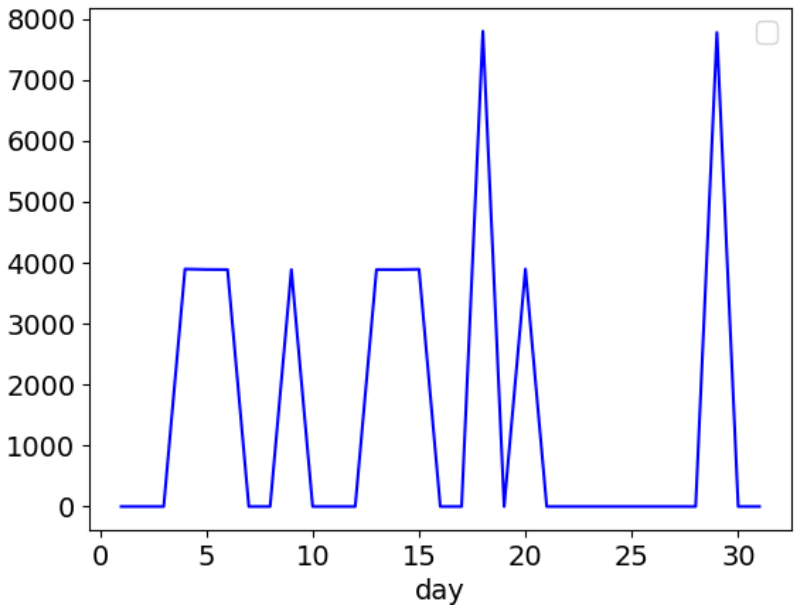


図 6.3. MAC_1 における Malicous パケット検知数の推移

6.4　モデルが検知する Anomaly と実際の Malicious な挙動との相関

Malicious なパケットが観測された 11 のアドレスに関して、それらのアドレスから送信される ARP リクエストをまとめ、さらにそれに対してフィッティングモデルを用いた異常検知を行った。図 6.4 は Malicious パケットの検知数とフィッティングモデルにおける Number を特徴量とした異常検知のグラフとを表示させたものである。左の図における赤色と緑色は順に ARP パケットの 1 日当たりの送信相手数とそれに対するフィッティングモデルの上界であり、異常検知された日は赤く塗られている (フィッティングモデルの下界に関しては検知対象日がないため省略した)。このグラフから、Malicious とされるパケットの送信パターンと ARP スキャンの実行パターンはかなりの類似性を持っていることがわかる。

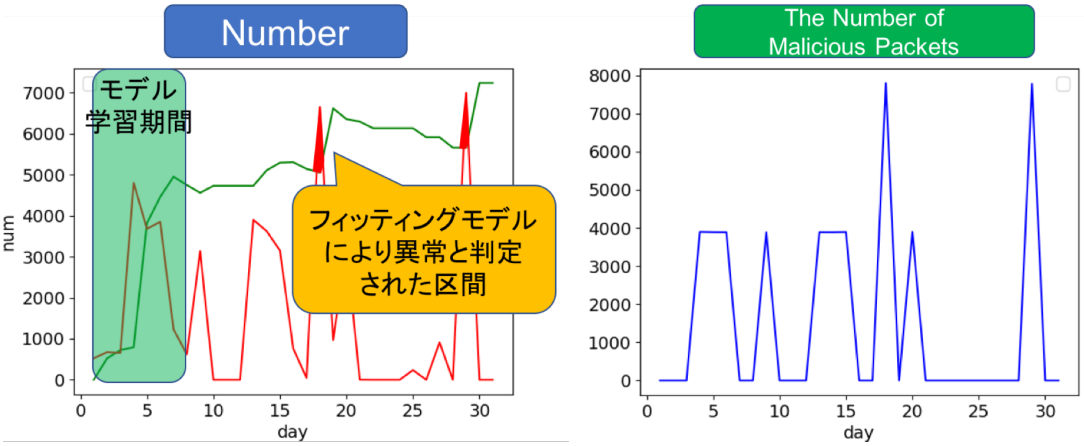


図 6.4. Malicous パケットとフィッティングモデルによる検知

6.5　考察

発見された Malicous ノードの中には、ある一つの特徴量だけでは異常であると検知できていないものも存在した。表 6.2 は、特徴量ごとの Malicous ノードの検知率を示したものである。まず送信相手数を特徴量とした際検知に失敗した二つのデバイス (MAC_1 及び MAC_6) の挙動及びそれらの端末から送信された Malicious なパケットの推移は図 6.5 のようになった。この二つを比較すると、これらはともに学習期間中に大きな値を経験してしまい、検知開始時点ですでにフィッティングモデルが過剰な値をとってしまったことが検知失敗の原因であることがわかる (学習期間中は異常検知を行わないため、この間に大きな異常が発生するとそれが検知できなくなってしまう)。また特に MAC_1 に関しては、特徴量の値が頭打ちになっていることも検知失敗に大きく関わっている。この LAN は高々 1000 程度の IP アドレスからなっているが、このデバイスは LAN 内のほぼ全てのアドレスに対して ARP リクエストを送

り続けている。そのため学習期間の間にこの特徴量が取りうる最大値を経験してしまい、後半の ARP スキャンを検知することができなかったと考えられる。このように送信相手数を特徴量とした検知のみでは、LAN 全体を対象とした ARP スキャン攻撃を検知することが難しいという弱点がある。

表 6.2. 特徴量ごとの Malicious とされるデバイスの検知率

特徴量	検知数	検知率	検知に失敗したアドレス
degree	9	0.82	MAC_1、MAC_6
number	10	0.91	MAC_{10}
variance	10	0.91	MAC_{10}

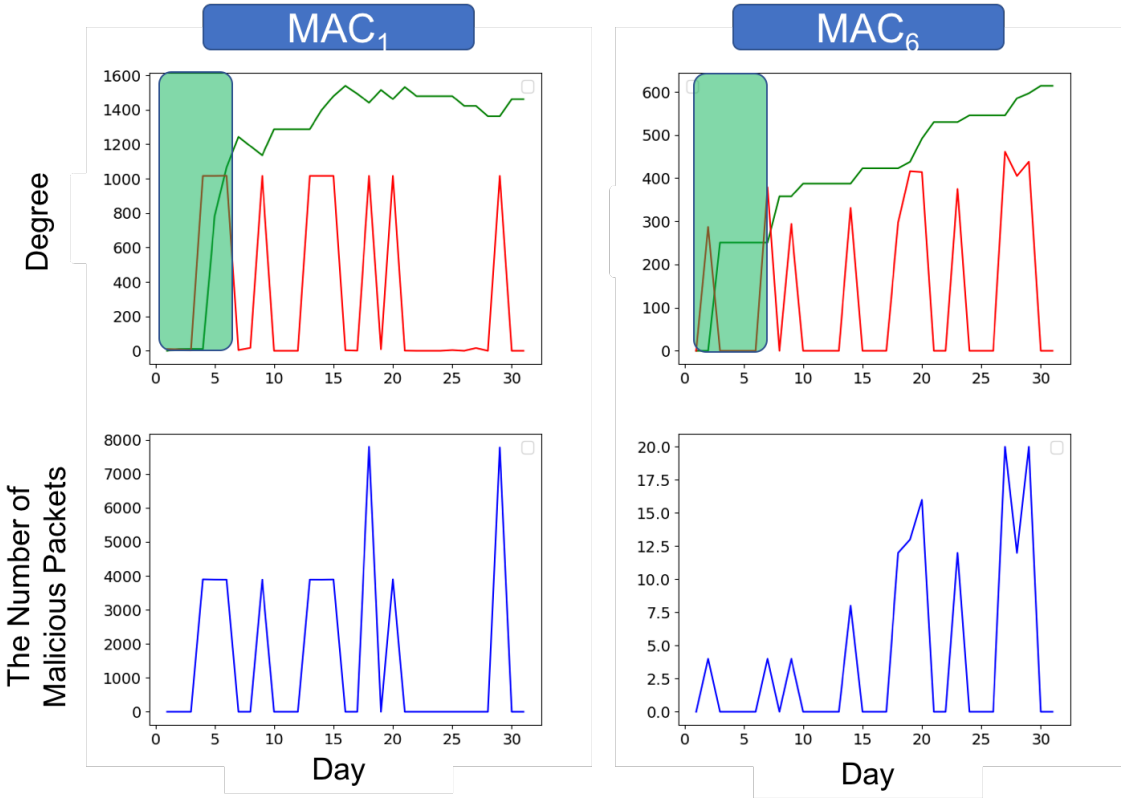


図 6.5. 送信相手数を特徴量にした時の失敗例

一方図 6.6 は MAC_1 と MAC_6 を送信リクエストの総数で見た場合の検知グラフである。このグラフにおいても学習期間中に大きな値を経験していることがわかるが、送信相手数を特徴量とした場合とは異なり別日の異常をきちんと検知している。特に MAC_1 に着目すると、一日に送信する ARP リクエストの総数には上限がなく、同じ LAN 全体に対する ARP スキャンであっても両者を区別できていることがわかる。このようにある一つの特徴量だけで

は発見できなかった異常も、他の特徴量による検知によって補完的に発見できることが確認できる。

逆に送信相手数を特徴量に据えることで初めて検出できた例もある。図 6.7 は MAC_{10} における各モデルの挙動を表しており、送信リクエストの総数や送信間隔を特徴量とした場合では検知しきれなかった。これらのグラフは先月の段階で学習期間を終えているが、このケースにおいてもまた学習期間中に大きな異常値が発生していることがわかっている。しかし今回のケースでは MAC_1 のケースとは異なり、学習期間中に起きた異常が LAN 内全体へのスキャンではなかった。そのため送信相手数を特徴量とした検知が正常に動作し続け、他の特徴量では発見できなかった異常を検知することができた。

このようにそれぞれの特徴量を組み合わせることにより今回 Malicious と定めた挙動をするデバイスを全て検知することができ、本研究で提案するモデルの有用性が確かめられた。

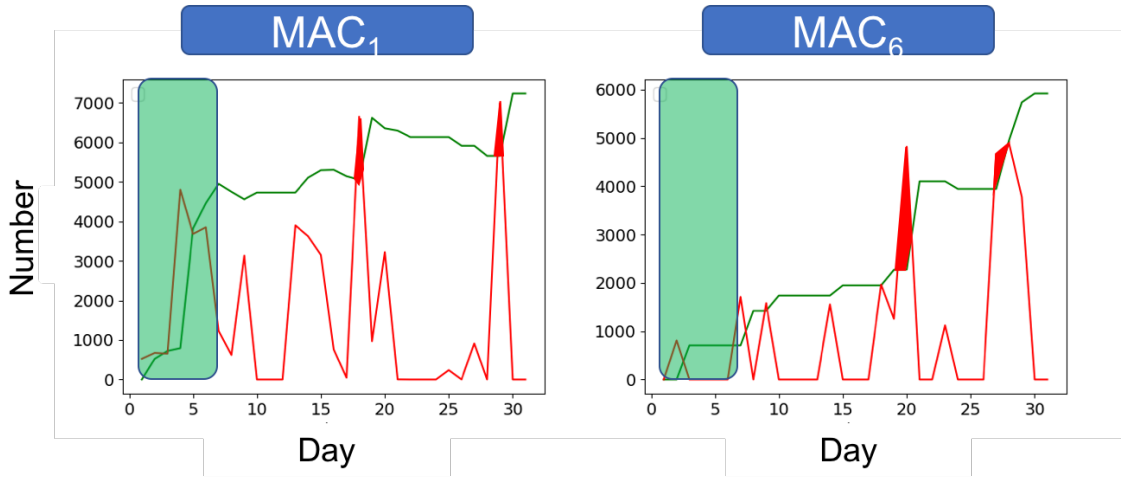


図 6.6. 送信リクエストの総数を特徴量としたときの MAC_1

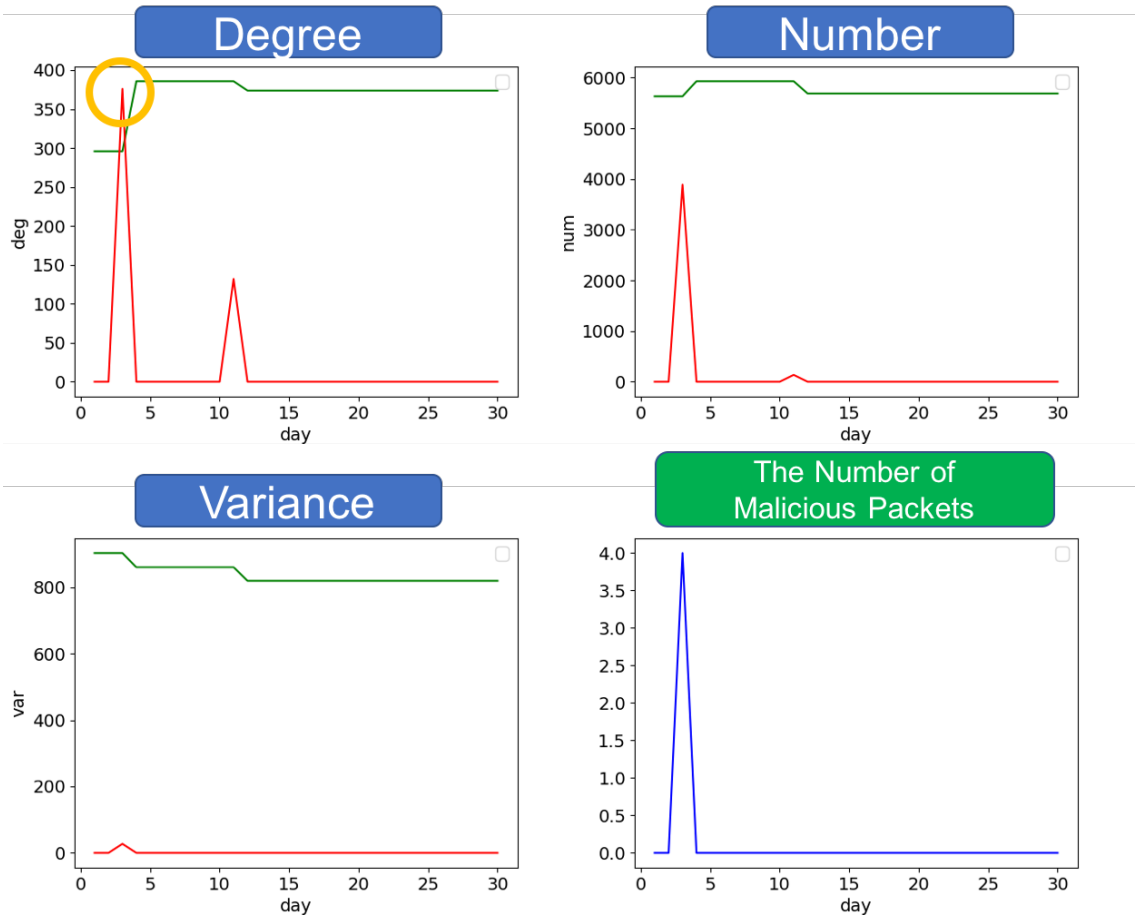


図 6.7. MAC_{10} における各特徴量の挙動

第 7 章

議論

7.1　公共 Wi-Fi への導入時の注意点

前章において、実際に拡散マルウェアの感染が疑われる LAN に対してフィティングモデルを用いた異常検知を行い、その有用性を確認した。しかし本研究で提案するモデルを不特定多数が接続する LAN(e.g. 公共 Wi-Fi) に対して導入する場合、特にその不特定な MAC アドレス (必ずしも毎日 LAN に接続されているわけではないアドレス) に関して注意すべきである。図 7.1 は前章での検知期間中一日だけ LAN での通信が行われたデバイスである。このデバイスは前章で定義した Malicious なパケットは送信しておらず、検知期間の終盤に突如として ARP リクエストの送信が確認された。データセットの章でも述べたが、本実験で用いたパケットデータには検知中全 IP アドレスの倍近くの MAC アドレスが確認されているため公共 Wi-Fi など不特定多数のアドレスが接続されるタイプの LAN であると考えられる。そのような公共の LAN において図 7.1 のような通信パターンが発生した場合、それが固定アドレスか DHCP によって割り当てられたアドレスかでパターンの意味合いが変わってくる。前者の場合は普段から接続されてかつ通信のなかったデバイスがある日突然どこかへ通信したというパターンであり、不審な挙動を見せている可能性が考えられる。一方後者の場合、このデバイスは通信が発生する日までそもそも LAN に接続されていなかった可能性が高い。つまり異常が検知された当日に初めて LAN に接続したため、学習期間において本来の通信パターンを学習できていないという問題が起こる。ここで今回の解析に際して、監視ノードでの ARP の情報に加えて DHCP によって割り当てられる IP アドレス帯についての情報も既知であるとする。この時仮に図 7.1 のアドレスが DHCP によって割り振られたものであるとわかれば、固定されたアドレスと DHCP によって割り振られたアドレスを区別して別々に検知することができる。例えば DHCP によって割り振られているアドレスは、検知開始の一週間ではなく最初に接続が確認された一週間を学習期間として設定するよう変更するといった改良を加えることで、より正確な異常検知が可能となると期待される。

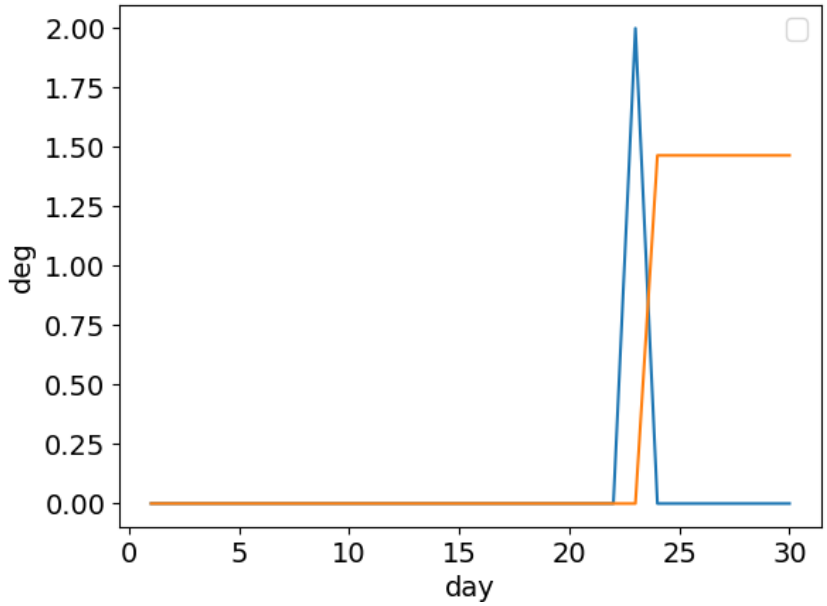


図 7.1. 検知期間中一日だけ通信が確認されたデバイス

7.2　ARP 以外のパケットに関しての検討

本研究では ARP リクエストパケットのブロードキャストという特性に着目し、それを使って LAN 内の通信を監視するという手法を採用した。ここで ARP リクエストの他に、本研究と同じ手法の対象となりうるパケットに関して検討する。この節では大きく、

- IPv6 における MAC アドレス解決プロトコルである Neighbor Discovery
- DHCP クライアントが送信する DISCOVER メッセージ

の二つに関して議論する。

まず Neighbor Discovery(ND) は、IPv6 における ARP のような役割を果たす機能である。本研究のターゲットである IPv4 においては ARP のブロードキャストを用いて MAC アドレス解決を行ったが、IPv6 ではブロードキャストアドレスではなく要請ノードマルチキャストアドレスが用いられる。ここで要請ノードマルチキャストでは送信先 IPv6 アドレスの下位 6 ビットを当てはめたメッセージを作成するため、ほとんどのケースで (下位 6 ビットが一致している場合を除いて) 特定の相手にしか送信されない。そのため全端末の通信を把握するのは厳しく、これを本研究の解析手法の対象に採用することは難しい。

また DCHP の DISCOVER パケットは、デバイスが DHCP から IP アドレスを振ってもらう設定になっているとき、DHCP のシーケンス開始のためブロードキャストされるものである。この際デバイスにはまだ IP アドレスが割り振られていないため、このパケットの送信

元 IP アドレスは「0.0.0.0」となっている。そのためこのパケットの送信元 IP アドレスから LAN 内の挙動を解析することはできない。一方送信元 MAC アドレスはデバイス本体のものが正しく表記されているため、監視デバイスからどの MAC アドレスがパケットを送信したかを知ることは可能である。しかしこのパケットによる解析では ARP とは異なり DHCP でアドレスを要求したという情報しか得られないため、通信頻度やその異常の解析として採用するには不適切であり、そもそも DHCP によって IP アドレスが割り振られない固定アドレス帯のデバイスに関しては何の情報も得られないという欠点がある。

ただしこのパケットを監視することで DHCP によって割り振られる IP アドレスに関する情報を得られるというメリットには注目すべきである。これにより LAN に接続されていながら単に通信してなかった機器とそもそも LAN に繋がってなかった機器を区別して扱うことができるようになる。そのため ARP を用いた解析を主軸としたまま本研究の精度を上げるための補助材料としてこのパケットを採用することは十分考えられる。

第 8 章

結論

本論文では、既存のネットワーク構成を変えずかつ簡単に導入できる LAN 内異常検知技術を提案した。本研究においては ARP リクエストを分析する手法に着目し、まずこのパケットの

- 通信相手数
- ARP リクエストの総数
- 通信間隔の分散

という三つの特徴量に着目した。その上で当日までの特徴量を基に翌日の傾向を動的に予測し、それを元に通信の異常を検知するフィッティングモデルを用いた異常検知を行った。

本研究において提案するフィッティングモデルは、

- 自動学習機能
- 時間変化への追随
- 一度経験した激しい値の変化の記憶

の三つの機能を持つ必要があるとみなし、それらの機能がもっとも効果的に表れるようパラメータをチューニングした。

そして外部期間で実際に運用されている LAN のデータを収集し、そのデータについて提案手法を適応した。その結果、3 つの特徴量を併用することで合計 2399 の MAC アドレスのうちのべのアドレスで 1 回以上の異常を発見した。さらに本研究においては LAN 内拡散マルウェアに感染した端末が送信する「ポート 445 宛ての TCP パケット」を Malicious なパケットであると定義し、Malicious なパケットを検知期間中一度でも送信した端末を Malicous ノードとして着目した。その結果提案手法で見つかった異常のうち合計 11 についてはトラフィック解析からも Malicious なものであることが確認できた。これにより本研究で提示した手法を用いて悪意のある挙動を見せるノードを指摘することができ、本研究の有用性が示された。

今後の課題として、公共 Wi-Fi など不特定多数の端末が不定期に接続してくる LAN に対して本手法を導入する場合に注意が必要であることが挙げられる。これは ARP パケットの観測のみでは端末が不定期に接続してくるものか固定で接続されているものかの判別ができないた

めであり、今まで通信を行っていなかった端末がある日突然活発に動き出した場合、それが単なる不定期な接続によるものであってもマルウェアによるスキャンであるかのようにモデルが判断し、異常であると判定してしまうためである。

この課題は、DHCP によって割り振られる IP アドレス帯についての情報を DISCOVER メッセージの解析などの手段を用いて入手することで改善することが見込まれる。固定されたアドレスと DHCP によって割り振られたアドレスを区別して本手法を適応することで、より正確な異常検知が可能となると期待される。

発表文献と研究活動

(1) Kai Matsufuji, Satoru Kobayashi, Hiroshi Esaki, and Hideya Ochiai. ARP Request Trend Fitting for Detecting Malicious Activity in LAN. International Conference on Ubiquitous Information Management and Communication. pp.89–96, 2019

(2) 松藤央， 落合秀也，江崎浩． 無線端末による ARP を用いたセグメント内の通信妨害攻撃とその対策． マルチメディア，分散，協調とモバイル (DICOMO2018) シンポジウム,2018.07.04.

参考文献

[1] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, Vol. 10, No. 4, pp. 2233–2243, 2014.

[2] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, Vol. 29, No. 7, pp. 1645–1660, 2013.

[3] In Lee and Kyoochun Lee. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, Vol. 58, No. 4, pp. 431–440, 2015.

[4] Ulf Lindqvist and Peter G Neumann. The future of the internet of things. *Communications of the ACM*, Vol. 60, No. 2, pp. 26–30, 2017.

[5] Hamdija Sinanovic and Sasa Mrdovic. Analysis of mirai malicious software. In *Proceedings of the International Conference on Software, Telecommunications and Computer Networks*, pp. 1–5, 2017.

[6] Georgios Kambourakis, Constantinos Kolias, and Angelos Stavrou. The mirai botnet and the iot zombie armies. In *Military Communications Conference (MILCOM), MILCOM 2017-2017 IEEE*, pp. 267–272. IEEE, 2017.

[7] Kishore Angrishi. Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets. *arXiv preprint arXiv:1702.03681*, 2017.

[8] Marios Anagnostopoulos, Georgios Kambourakis, and Stefanos Gritzalis. New facets of mobile botnet: architecture and evaluation. *International Journal of Information Security*, Vol. 15, No. 5, pp. 455–473, 2016.

[9] Krushang Sonar and Hardik Upadhyay. A survey: Ddos attack on internet of things. *International Journal of Engineering Research and Development*, Vol. 10, No. 11, pp. 58–63, 2014.

[10] Dragan Peraković, Marko Periša, and Ivan Cvitić. Analysis of the iot impact on volume of ddos attacks. In *33rd Symposium on New Technologies in Postal and Telecommunication Traffic (PosTel 2015)*, pp. 295–304, 2015.

[11] Mmd-0055-2016-linux/pnscan;elf worm that still circles around. http://blog.malwaremustdie.org/2016/08/mmd-0054-2016-pnscan-elf-worm-that.html, August 2016.

[12] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. Understanding the mirai botnet. In *USENIX Security Symposium*, pp. 1092–1110, 2017.

[13] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. Ddos in the iot: Mirai and other botnets. *Computer*, Vol. 50, No. 7, pp. 80–84, 2017.

[14] Timo Kiravuo, Mikko Sarela, and Jukka Manner. A survey of ethernet lan security. *IEEE Communications Surveys & Tutorials*, Vol. 15, No. 3, pp. 1477–1491, 2013.

[15] Kyle McHugh, Walter Akpedeye, and Thaier Hayajneh. Next generation wireless-lan: Security issues and performance analysis. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1–7. IEEE, 2017.

[16] Md Waliullah, ABM Moniruzzaman, Md Sadekur Rahman, et al. An experimental study analysis of security attacks at ieee 802. 11 wireless local area network. *International Journal of Future Generation Communication and Networking*, Vol. 8, No. 1, pp. 9–18, 2015.

[17] Ugo Fiore, Francesco Palmieri, Aniello Castiglione, and Alfredo De Santis. Network anomaly detection with the restricted boltzmann machine. *Neurocomputing*, Vol. 122, pp. 13–23, 2013.

[18] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pp. 21–26. ICST (Institute for Computer Sciences, Social-Informatics and ⋯, 2016.

[19] Zecheng He, Tianwei Zhang, and Ruby B Lee. Machine learning based ddos attack detection from source side in cloud. In *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 114–120. IEEE, 2017.

[20] Curtis R Taylor and Julian P Lanson. Network-based classification of authentication attempts using machine learning. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 669–673. IEEE, 2019.

[21] Clemens Kolbitsch, Paolo Milani Comparetti, Christopher Kruegel, Engin Kirda, Xiao-yong Zhou, and XiaoFeng Wang. Effective and efficient malware detection at the end host. In *USENIX security symposium*, Vol. 4, pp. 351–366, 2009.

[22] 本当は危ない AI・IoT・仮想通貨 最新サイバーリスク 2019. 本当は危ない AI・IoT・仮想通貨 最新サイバーリスク 2019. 日経 BP 社, 2018.

[23] Poonam Pandey. Prevention of arp spoofing: A probe packet based technique. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, pp. 147–153. IEEE, 2013.

[24] Gao Jinhua and Xia Kejian. Arp spoofing detection algorithm using icmp protocol. In *Computer Communication and Informatics (ICCCI), 2013 International Conference on*, pp. 1–6. IEEE, 2013.

[25] Xiangning Hou, Zhiping Jiang, and Xinli Tian. The detection and prevention for arp spoofing based on snort. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, Vol. 5, pp. V5–137. IEEE, 2010.

[26] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye, and Yiqiang Sheng. Malware traffic classification using convolutional neural network for representation learning. In *2017 International Conference on Information Networking (ICOIN)*, pp. 712–717. IEEE, 2017.

[27] Matija Stevanovic and Jens Myrup Pedersen. An analysis of network traffic classification for botnet detection. In *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, pp. 1–8. IEEE, 2015.

[28] Wannacry malware profile — fireeye inc. `https://www.fireeye.com/blog/threat-research/2017/05/wannacry-malware-profile.html`. (Accessed on 01/14/2020).

[29] Qian Chen and Robert A Bridges. Automated behavioral analysis of malware: A case study of wannacry ransomware. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 454–460. IEEE, 2017.

謝辞

本論文の執筆にあたり、非常に多くの方にご指導・ご協力をいただきましたことを心より感謝申し上げます。

指導教員である落合 秀也准教授は私が本研究へと至るための第一歩を提示してくださった方であり、LAN Security Monitoring Project へのお誘いに始まり研究の軌道修正から日常生活に至るまで非常に多くのご指導をいただきました。特に IMCOM2019 における海外発表を成し遂げることができたのは先生の手厚いご指導によるところが大きく、私の研究生活において非常に大きなモチベーションを与えてくださいました。私の二年間の修士過程は先生のご指導あってのものだと確信しています。ここに深く感謝申し上げます。

研究室全体を率いる存在である江崎 浩教授には、学部過程に始まり様々なご指導をいただきました。ご多忙の中研究に関する的確なご指導ご鞭撻をいただき、また折に触れて様々な興味深いお話を伺うことができました。先生からいただいた数多くの知識は、これから社会の中で生きていく私にとって必ずや有益なものになるであろうと考えております。深く感謝申し上げます。

国立情報学研究所の小林 諭氏には、研究内容から日々の生活に至るまで多岐にわたるご指導をいただきました。お忙しい中ご自身の経験からなる的確なアドバイスをいただき、研究に関する詳細な軌道修正や日々の心構えなど、長期にわたる研究室生活を最も身近に支えてくださった方だと考えています。深く感謝申し上げます。

夏合宿に始まり折に触れて研究に関するご指導ご鞭撻をいただいた塚田 学准教授、山本 成一助教授に深く感謝申し上げます。

長期にわたる学生生活をともに励ましあい、実りあるものにしてくださった新津 雄大氏、大井 貴晴氏、長嶋 秀幸氏、山城 裕陽氏をはじめとする研究室の全ての方々に感謝申し上げます。自分のこれまでの人生を様々な面でサポートしてくださった家族や友人に感謝申し上げます。

最後に、ここまでお世話になりました全ての方々に感謝申し上げます。