SMS スミッシング難読化対策の検討と検出アルゴリズム DSmishSMS-A の改善

Study of SMS smishing obfuscation countermeasures and improvement of detection algorithm DSmishSMS-A

張 文佳* 奥原 雅之 [1]
Zhang Wenjia* Masayuki Okuhara[1]

[1] 東京都立産業技術大学院大学 Advanced Institute of Industrial Technology
*Corresponding author: Zhang Wenjia, c192@foxmail.com

Abstract Smishing is a type of social engineering attack carried out by sending fake mobile text messages. The phone numbers used in the attacks and the URLs in the smishing messages are typically changed for each attack, making it difficult to block them completely. Smishing is becoming one of the growing security threats in our daily lives, and there is an urgent need to understand and address its serious consequences. In this study, we analyzed the content of smishing messages observed in reality and made the following findings. First, there is widespread abuse of dynamic domain name provision services and shortened URL services. Second, while subdomain names used by attackers change constantly, IP addresses change less, so countermeasures such as blacklist construction are more effective when based on IP addresses rather than URLs. Third, the same Unicode character may be represented in different encodings, making it difficult for inspection systems to consistently detect them. Fourth, the permissions required by APKs downloaded through redirection are different from those of APKs for official applications. Finally, based on these findings, we recommend improvements to the existing anti-smishing method, the DSishSMS-A algorithm.

Keywords phishing; smitshing; information security; cyber sercurity

1 はじめに

スミッシングとは、偽のモバイルテキストメッセージを送付することにより、利用者を欺いてマルウェアをダウンロードさせたり、機密情報を共有させたり、攻撃者あてに送金させたりすることを意図したソーシャルエンジニアリング攻撃の一種である[1]。「スミッシング」という用語は、テキストメッセージを支える技術である「SMS」または「ショートメッセージサービス」と「フィッシング」を組み合わせた造語である。

本論文の執筆の背景として、筆者および身近な人々が頻繁にスミッシング SMS を受信しているという現実がある。これは非常に迷惑であるだけでなく、セキュリティ上も深刻な問題である。送信元の電話番号やスミッシングメッセージに記載されている URL は毎回異なるため、これらの情報に基づいてスミッシングメッセージを完全にブロックすることは難しい。スミッシングは、我々の日常生活において増加するセキュリティ上の脅威の一つとなりつつあり、その深刻な影響に対処するための理解と対策が喫緊の課題となっている。

この現象に直面した経験から、筆者はなぜスミッシング SMS が増加しているのか、どのような手法が使用されているのか、そしてこれによって生じる潜在的な被害やリスクは何なのか、といった疑問を抱いた。この論文は、私たちが日々直面するスミッシング SMS の問題について徹底的に調査し、その原因と対策について知見を提供することを目的としている。

フィッシング攻撃の動向は時間とともに急速に変化しており、かつ件数も急増している。特に SMS を用いた攻撃が増加し、フィッシングサイトの存続時間は短くなっている。このため、フィッシング関連情報の収集・共有に新しい取り組みが必要とされており、フィッシング URL の迅速な共有についても新しい方法が求められている。FIDO に代表される新たなユーザー認証技術が期待されている一方、既存のフィッシング対策技術の適切な使用や、IT サービス事業者による認証の推奨事項やベストプラクティスへの理解も重要である。

フィッシング対策は技術的な対策のみでは難しく、フィッシングの実行前段階での対策も検討されるべきである。ドメインの管理が重要になり、ドメイン名の廃止時の注意やドロップキャッチの事例、再考ドメイン名など、ドメイン関連の問題も考察されるべきである。

フィッシング対策協議会によるフィッシングレポート 2022[2]は、WG のメンバーが多角的な視点から寄稿し、フィッシングの現状や課題について幅広く記述された内容となっている。

本稿では、現実に観測されているスミッシングメッセージの内容を分析し、これに基づき既存の対策手法である Mishra らの DSishSMS-A アルゴリズム[3]の改善を提言する。

2 先行研究

XLoader と FakeSpy は、日本を狙う主要なスフィッシングボットネットで、従来の攻撃手法から攻撃戦略を大きく変更している。Saeki らは、ボットネットと Twitter データを観察して、戦略や活動パターンを分析し、それをマルウェアや悪意のあるドメイン検出に応用した[4]。提案された方法は、偽陽性と偽陰性率が低いことに加え、少ないコンピュータリソースの使用により、ユーザーデバイスで実行できる。また、Frida を使って TCP/IP トラフィックの上位レイヤーをデコードし、コマンドレベルのトラフィック分析を行い、攻撃の特徴を特定した。悪意のあるドメイン検出では、攻撃者がバッチでドメインを作成する傾向を利用し、SMS メッセージの到着率に焦点を当てた。これらの方法は誤り率が低く、パフォーマンスが高いと期待されている。

フリーウェブサイトビルダー（FWBs）は、技術知識やコーディングスキルがなくてもウェブサイトを作成できるコスト効率の良い方法を提供するツールである。しかし、この種のツールが悪用されフィッシングサイトのホスティングに利用されることも少なくない。Roy らは、FWBs で作られたフィッシングサイトを継続的に特定するフレームワーク「FreePhish」を提案した[5]。FreePhish を用いて、Twitter や Facebook で共有された 17 の FWBs サービスを使用して作成された 31,400 以上のフィッシング URL を検出し、特徴を分析した。FWBs は攻撃

Received: 2024-10-30; Published online: 2025-02-28

者にフィッシングサイトの作成・維持を容易にし、アンチフィッシング対策を回避する機能を提供している。アンチフィッシングのブロックリストやブラウザ保護ツールは、FWBs ベースのフィッシング攻撃に対するカバー率が低いと検出が遅れることが示された。FreePhish を Chromium ウェブ拡張として提供し、エンドユーザーが FWBs ベースのフィッシング攻撃にアクセスすることを防ぐことができる。

Sakurai は、本物と同様に見せかけたフィッシングサイトを検出することを目指し、phishtank の 100 件のフィッシングサイトのドメイン情報と HTTP レスポンスヘッダを集めて分析し、自動検出の方法を提案した[6]。フィッシングサイトのドメインとヘッダ情報は、正規なサイトとは異なる特徴がある。フィッシングサイトを作るキットの特徴を見つけることで、検出の精度を上げている。

Sakurai は、TLS 証明書のフットプリントによるフィッシングサイトの検出方法について提言している[7]。最近のウェブ上での HTTPS の普及に伴い、攻撃者はフィッシングウェブサイトを「HTTPS 化」し始めた。HTTPS 化されたフィッシングウェブサイトは、あたかも正当なサイトに見えるようになり、攻撃者にとっては従来の URL やネットワーク内のウェブコンテンツを利用する検出方法を回避できる利点がある。一方、HTTPS の採用は、ウェブサイトの準備時に発行される公開鍵証明書が必要になり、内的な痕跡を生成し、検出の機会を提供する可能性がある。

証明書に基づく検出の潜在的な利点は以下の通りである。

1. 攻撃者がダイナミック DNS（DDNS）やホスティングサービスを利用した場合でも、証明書を発行直後にすぐに使用して、すべての HTTPS ウェブサイトを包括的に監視できる。
2. インターネット上に公開される前に対応するフィッシングウェブサイトを検出する。

そこで、この研究では「TLS 証明書の痕跡をいかに利用してフィッシング攻撃から守るべきか」をリサーチクエスチョンとして設定し、証明書の透明性（CT）ログからフィッシングウェブサイトに対応する TLS 証明書の大規模なセットを収集し、これらの TLS 証明書を広範に分析した。証明書のクラスタリング分析を通じて得られた共通名のテンプレートは、以下のような応用に使われることが期待できることを示した。

1. 低い偽陽性を実現しながら、以前に知られていなかったフィッシングウェブサイトの発見。
2. フィッシングウェブサイトを生成するために使用されるインフラストラクチャの理解。

また、無料の証明書認証局（CA）の悪用に関して見つかった事実を適示し、このような悪用に対する可能な解決策について議論し、CA に対する推奨事項を提言している。

Roy らは、セキュリティ意識の高いユーザーが攻撃情報を Twitter で報告し共有する取り組みにおいて、フィッシング攻撃を特定する新しい取り組みを紹介している[8]。2021 年 6 月から 8 月の期間に 701 個の Twitter アカウントが投稿した 16,400 以上の報告を評価し、そのうち 11,100 のユニークな URL を分析した。その結果、これらの報告は多くの正しいフィッシング URL を共有し、PhishTank や OpenPhish の 2 つの人気のあるオープンソースフィッシングフィードと比較して、フィッシングウェブサイトに関する詳細情報も多く含まれていることが示された。しかし、これらの報告には Twitter 上の他のユーザーからの相互作用が非常に少なく、報告された URL がターゲットとされたドメインや組織からの相互作用も乏しく、報告されてから 1 週間以内に 31%の URL がまだアクティブで、多くのアンチフィッシングツールによって検出されていないことがわかった。この研究は、Twitter で共有されるフィッシング報告の有用性と、それらをオープンソースの知識ベースとして使用して新しいフィッシングウェブサイトを特定する利点を示唆している 。

スマートフォンの普及によるウェブ接続の増加は、フィッシングやスミッシング攻撃のリスクを高めている。スミッシングは悪意のある SMS を送信する攻撃で、フィッシングは悪意のあるメールを送る行為である。過去数年、研究者はスミッシングの検出方法を提案してきたが、偽陽性を減らす方法は見つかっていない。そこで、Mishra らは「Smishing Detector」モデルを提案した[9]。このモデルは 4 つのモジュールからなり、SMS の内容を分析し、悪意のあるコンテンツやキーワード、URL、ウェブサイトのソースコード、ダウンロードされる APK を検出する。このシステムは SMS データセットを使って検証され、正確性が 96.29%に達した。他のモデルと比べると、セキュリティの側面をより多くカバーしていることが示された。

3 アプローチ

ISO/IEC/IEEE 15288: System life cycle processes では、プロセスを「インプットをプロセスアクティビティにて処理し、アウトプットに変換する活動」と定義している。本研究では、提案手法が実際のフィッシングキャンペーンに普遍的に適用可能なプロセスであることを確認し、フィッシング詐欺を体系的に理解し、各活動にて結果を引き起こす因子を明らかにすることで、脅威を予測することが容易になることを示す。

DSmishSMS-A System to Detect Smishing SMS ではシンプルで効率的なスミッシング検出のためのアルゴリズム（図 1）が提案されており、本研究ではこのアルゴリズムの改善を目指す。

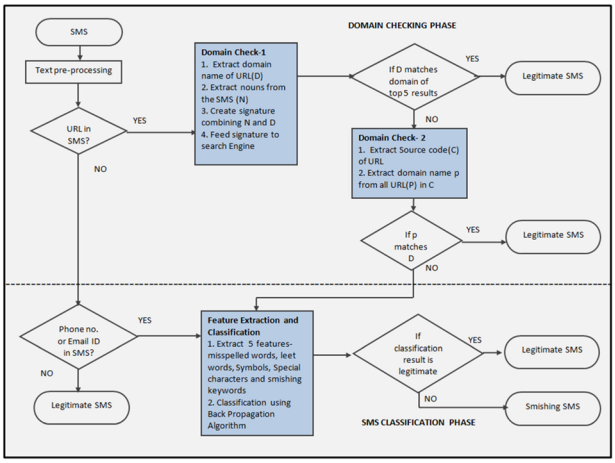

図 1　DSmishSMS-A System to Detect Smishing SMS のアルゴリズム

4　サンプルについての分析

アルゴリズムの検討に先立ち、筆者の環境で得られたスミッシングメッセージを素材とし、その分析を試みる。収集した 2020 年から 2023 年の期間のサンプルは図 2 の通りである。

No	日付	SMS内容	URL抽出
1	2020/06/30	お荷物のお届けにあがりましたが不在の為持ち帰りました。ご確認ください。http://tajfvantmf.d	http://tajfvantmf.duckdns.org
2	2020/10/20	ご本人様不在の為お荷物を持ち帰りました。ご確認ください。http://gwzuayjzwb.duckdns.org	http://gwzuayjzwb.duckdns.org
3	2020/10/20	ご本人様不在の為お荷物を持ち帰りました。ご確認ください。http://gxqmjcfhgk.duckdns.org	http://gxqmjcfhgk.duckdns.org
4	2021/03/16	ご本人様不在の為お荷物を持ち帰りました。ご確認ください。http://jnvfnpweot.duckdns.org	http://jnvfnpweot.duckdns.org
5	2021/03/16	ご本人様不在の為お荷物を持ち帰りました。ご確認ください。http://kdwckijqlj.duckdns.org	http://kdwckijqlj.duckdns.org
6	2022/03/23	お荷物の住所が不明でお預かりしております。確認してください。http://zwiqx.euzaw.com	http://zwiqx.euzaw.com
7	2022/12/19	お客様が不在の為お荷物を持ち帰りました。こちらにてご確認ください kg.***mhnpv***.com??ximgl	kg.***mhnpv***.com??ximgl
8	2023/01/01	住所が不足していますので、お荷物を再度確認してください。https://t.co/axnJ6WKrGG	https://t.co/axnJ6WKrGG
9	2023/02/04	お客様が不在の為お荷物を持ち帰りました。こちらにてご確認ください kg.***mhnpv***.com??ximgl	kg.***mhnpv***.com??ximgl
10	2023/03/23	お客様が不在の為お荷物@を持ち帰りました。こちらにてご確認ください tinyurl.com/2rue7eah	tinyurl.com/2rue7eah
11	2023/05/16	お客*様が不在の為お荷物を持ち帰りました。こちらにて%ご確*認ください z-xl7.***stm***mh.com?bth	z-xl7.***stm***mh.com?bth
12	2023/08/24	お客*様が不在の為お荷物を持ち帰り&ました。こちらにてご確*認ください z-4t.iu***dop***.com??	z-4t.iu***dop***.com??
13	2023/11/12	お届け先にお伺いしたところ、ご不在でした。再配達を行います。https://t.co/OOXd6y0tVq	https://t.co/OOXd6y0tVq
14	2023/12/30	お客様不在の際に荷物をお届けしましたが、持ち帰りました。こちらからご確認ください。https://t	https://t.co/xJorLInyFv

図 2　14 件のスミッシングサンプル

URL のエンコーディング方式についての分析

14 個のサンプルから、URL のエンコーディングの複雑化が進んでいることがわかる。例えば通常の ASCII 文字だけでなく、特殊な Unicode 文字 ("***mh***n***pv***") が URL に含まれている。この工夫により、視覚的に類似した文字を使用することで、攻撃者は被害者が本物の URL と見分けにくくなるようにしている。

Unicode 文字は複数のエンコーディング方式で表現でき、UTF-8 などが一般的に使用される。このため、同じ文字が異なるエンコーディングで表現される可能性があり、検査システムがこれを一貫して検知するのが難しくなる。この複雑なエンコーディングは、攻撃者がセキュリティ検査を回避しやすくする一因となっている。

URL のドメインネームについての分析

14 のサンプルの中で、5 つは DuckDNS（Duck Domain Name System）を利用している。DuckDNS は、動的 IP アドレスを持つユーザーに対して無料でドメインネームを提供するサービスである。

攻撃者は DuckDNS を利用し、自身の動的 IP アドレスに関連づけたドメインを取得している。そして、そのドメインを利用してフィッシングサイトや偽のウェブサイトに誘導し、悪意を持って利用している。

短縮 URL サービスの悪用についての分析

14 のサンプルの中には短縮 URL サービスを利用しているものがある（図 3）。サンプルのうち 1 例は TinyURL（タイニーユーアールエル）を使用している。TinyURL は、長い URL を短縮する（URL 短縮化する）サービスである。短縮 URL は元の URL を非常に短い形式に変換するため、そのままではリンク先が分からないことがある。URL がブラックリストに登録された場合でも、 その URL が短縮 URL に変化されれば、検知されない可能性がある。従ってブラックリストに登録された場合でも、攻撃者が新しい URL を生成して回避することが容易になる。

No	日付	URL抽出
8	2023/01/01	https://t.co/axnJ6WKrGG
10	2023/03/23	tinyurl.com/2rue7eah
13	2023/11/12	https://t.co/OOXd6y0tVq
14	2023/12/30	https://t.co/xJorLInyFv

図 3　短縮 URL の抽出

また、3 例では「t.co」を利用している。「t.co」は、Twitter が提供する短縮 URL サービスのドメインである。

Twitter は著名なサービス提供者であることから、攻撃者が「t.co」を利用することで、利用者はその信頼性を過信してしまう可能性がある。また、一部のセキュリティツールやメールプロバイダは、短縮 URL を展開して元の URL を確認することがあるが、「t.co」のような一般的なサービスはこの処理から除外されることがある。これにより、悪意のある URL が検知されにくくなる。

IP アドレスについての分析

サンプル中には、サブドメイン名は異なるが、IP アドレスは同一のものを使っている例がある（図 4）。 そのため、URL をブラックリストに入れるより、IP アドレスをブラックリストに入れる方がより効果的である。

No	日付	URL抽出	IPアドレス抽出
3	2020/10/20	http://gxqmjcfhgk.duckdns.org	3.97.179.200:80
4	2021/03/16	http://jnvfnpweot.duckdns.org	3.97.179.200:80
8	2023/01/01	https://t.co/axnJ6WKrGG	104.244.42.133
10	2023/03/23	tinyurl.com/2rue7eah	103.80.134.41:80
13	2023/11/12	https://t.co/OOXd6y0tVq	104.244.42.5:443
14	2023/12/30	https://t.co/xJorLInyFv	104.244.42.69:443

図 4　IP アドレスの抽出

APK についての分析

3 つのリダイレクト可能な URL を分析したところ、どのブラウザを使用しているかに関係なく、Android デバイスの利用者

に対して「セキュリティ向上のため、最新バージョンの Chrome にアップデートしてください。」と表示され、偽造された APK（Android Package Kit）ブラウザーのダウンロードへ誘導し、これを通じてユーザーの機密情報を盗み取ることを試みるものであった。

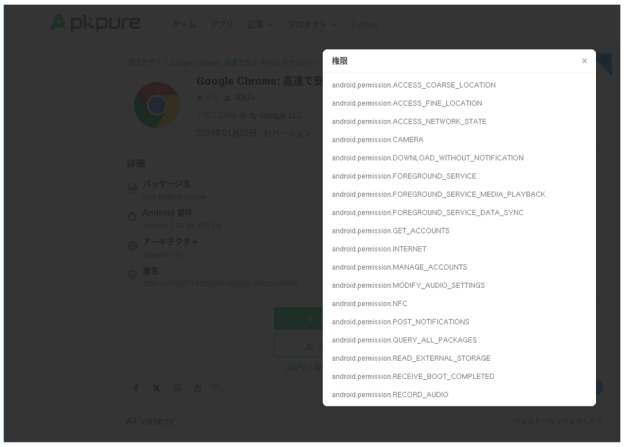

図 5　公式サイトの APK 権限リスト

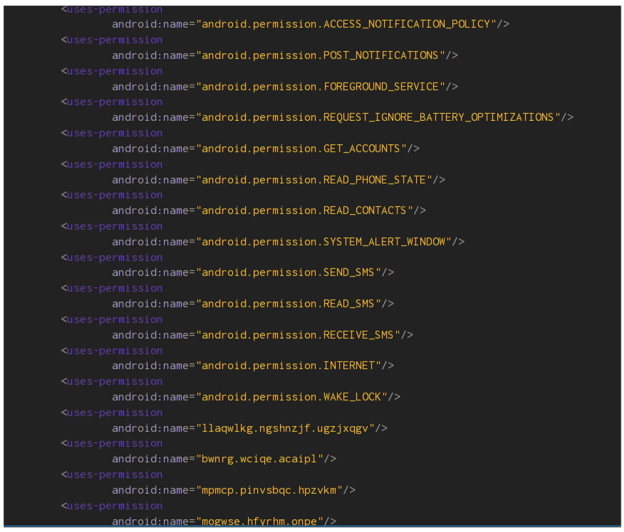

図 6　誘導先の権限リスト

URL から APK をダウンロードし、公式サイトの AndroidManifest.xml ファイルと誘導先のファイルを比較すると、本来必要としていない権限を要求していることがわかる（図 5 および図 6）。これらの APK は本来必要としない SEND_SMS、READ_SMS、RECEIVE_SMS の権限を要求しており、それらの権限を許可した端末は、攻撃者によってボットとしてさらなる攻撃に悪用される可能性があると考えられる。

5　DSmishSMS-A アルゴリズムの改善の提案

SMS コンテンツ内の URL 存在チェック

DSmishSMS-A で提案されたアルゴリズムは、SMS に特定の URL が含まれているか否かに基づいて判定を行っており、URL の存在有無に関する判断方法について触れていない（図 7）。

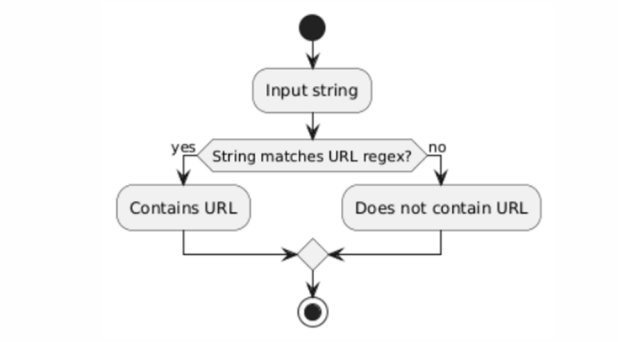

図 7　URL 存在チェックのアルゴリズム

DSmishSMS-A アルゴリズムでは「URL in sms?」のみの判断であり、「***z***-4t.iu***d***o***p***.com?7」のような通常の URL ではない場合は、非該当と判断する可能性がある。そのため、URL チェックアルゴリズムとして、まずエンコーディング方式が混在していることを避けるため、前処理を追加し、すべての内容を半角、小文字に変換する。URL の判断方式を通常の URL パターンから、以下のようなピリオド「.」とアルファベット文字列の配置パターンによる判断方法とする（図 8）。

①前処理を追加する
・文字列を半角に変換
・文字列を小文字に変換
②URL 判断方式を変更する
「.」があるかつ、「.」の右と左にアルファベット文字列がある

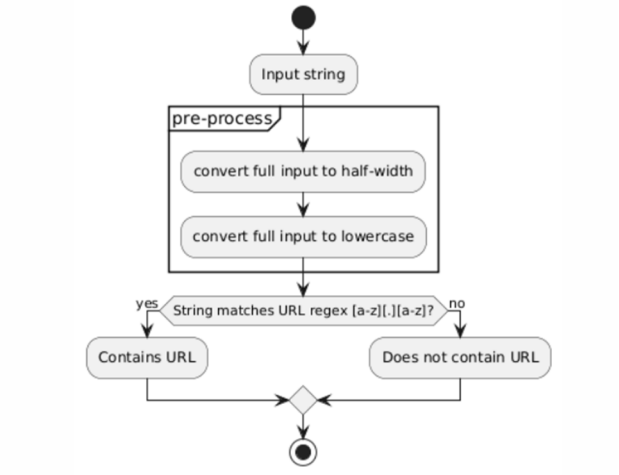

図 8　改善後の URL 存在チェックアリゴリズム

IP アドレスブロックリストの追加

ドメイン名を頻繁に変更し、送信元が特定されないような攻撃においても、同じ IP アドレスを使用しているケースがある

ことが判明したため、これに対応するために「URL in sms?」によってスミッシング SMS として判断された場合、IP アドレスを抽出し、DB に登録する。

「URL in sms?」の直後に IP アドレスの判定アルゴリズムを入れ、IP アドレスが一致した場合は、スミッシングと判断する。この場合、その後の判定はすべて省略できる（図 9）。

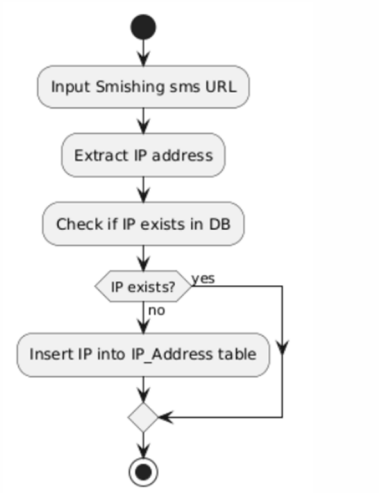

図 9　IP アドレスブロックアリゴリズム

APK 権限を比較するアルゴリズムの追加

APK が存在する場合は、ダウンロードしようとしている APK は公式の APK と一致しているかを確認する（図 10）。この APK の比較により、必要としない権限が存在しているかを判断することができる。

① domain チェック 2 でリダイレクトの URL を抽出する。
② URL に APK ダウンロードがあればそれをダウンロードし、AndroidManifest.xml ファイルを公式サイトのものと比較する。一致であれば、legal、不一致であれば、illegal と判断し、DB に登録する。

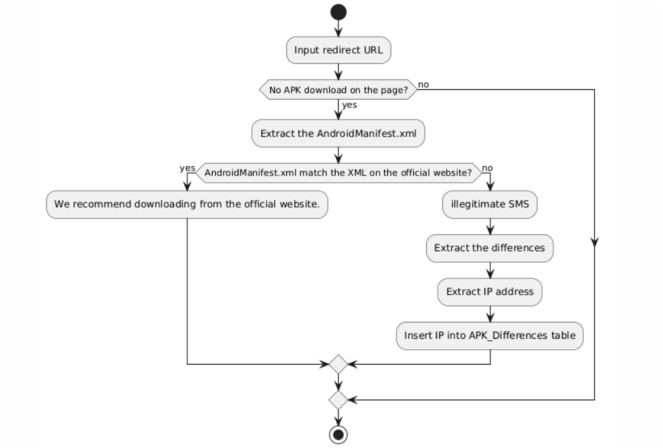

図 10　APK 権限を比較するアルゴリズム

6　おわりに

2020 年から 2023 年までの 14 通の詐欺 SMS を分析したことにより、絶えず進化している詐欺手法の実態が判明した。まず、宅配業者や郵便事業者を装ったものが大半を占めている。ダイナミックなドメインネーム提供サービスや短縮 URL サービスの悪用が広く見られ、スミッシングの URL は恒久的なものではなく、「賞味期限」がある。また、ネットワークフィッシング対策ツールでも悪意ある SMS を完全に検知できないことがわかる。詐欺の目的は主に個人のアカウント情報や銀行カード情報を盗むことだと推測できる。

攻撃者が利用するサブドメイン名は常に変化するが、IP アドレスは変化が少ないことから、ブラックリストを構築したり、脅威情報として共有する場合には、URL ではなく IP アドレスに基づく方がより効果的である。

Unicode 文字は複数のエンコーディング方式で表現でき、UTF-8 が一般的である。このため、同じ文字が異なるエンコーディングで表現される可能性があり、検査システムがこれを一貫して検知することは困難が伴う。しかしブラウザーがユーザーの便利性を考慮し、アクセスできる URL になるように自動的に不適切な文字を変換する。そしてユーザーは攻撃者のスミッシングサイトにアクセスしてしまい、通信業者の検査システムにも、ブラウザーが行っている URL の補正機能を加えて、検知率をあげることができると考えられる。

APK の権限を公式サイトと比較し、必要としていない権限があるかのチェックのアルゴリズムがあれば、より効率的にスミッシングを判断できる。

本研究が今後の研究に一石を投じる手助けとなることを期待している。

参考文献

1. 林憲明, 唐沢勇輔, 中村智史, 坂本美子, 柘植悠孝, 岡田雅之, 加藤雅彦, フィッシング詐欺のビジネスプロセス分類, フィッシング対策協議会, 2021. https://www.antiphishing.jp/news/collabo_20210316.pdf
2. フィッシング対策協議会 技術・制度検討ワーキンググループ, フィッシングレポート 2022. https://www.antiphishing.jp/report/phishing_report_2022.pdf
3. Sandhya Mishra, Devpriya Soni. DSmishSMS-A System to Detect Smishing SMS. Machine Learning Applications for Security, Vol. 35,pp.4975–4992, 2023.
4. Ryu Saeki; Leo Kitayama; Jun Koga; Makoto Shimizu; Kazumasa Oida. Smishing Strategy Dynamics and Evolving Botnet Activities in Japan. IEEE Access vol.10, pp.114869-114884. DOI: 10.1109/ACCESS.2022.3217795
5. Sayak Saha Roy, Unique Karanjit, Shirin Nilizadeh. Phishing in the Free Waters: A Study of Phishing Attacks Created using Free Website Building Services. IMC '23: Proceedings of the 2023 ACM on Internet Measurement Conference, pp.268-281. https://dl.acm.org/doi/abs/10.1145/3618257.3624812
6. 桜井啓多, ドメイン情報と HTTP レスポンスヘッダに基づくフィッシングサイトの識別と評価, 明治大学総合数理学部 先端メディアサイエンス学科 菊池研究室, 2018. https://www.kikn.fms.meiji.ac.jp/paper/2018/bachelor/sakurai/thesis-resume.pdf
7. Yuji Sakurai; Takuya Watanabe; Tetsuya Okuda; Mitsuaki Akiyama; Tatsuya Mori. Discovering HTTPSified Phishing Websites Using the TLS Certificates Footprints. 2020 IEEE

European Symposium on Security and Privacy Workshops (EuroS&PW). DOI: 10.1109/EuroSPW51379.2020.00077.

8. Sayak Saha Roy; Unique Karanjit; Shirin Nilizadeh. Evaluating the Effectiveness of Phishing Reports on Twitter. 2021 APWG Symposium on Electronic Crime Research (eCrime). DOI: 10.1109/eCrime54498.2021.9738786.
9. Sandhya Mishra, Devpriya Soni. Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis. Future Generation Computer Systems, Volume 108, July 2020, pp.803-815. https://doi.org/10.1016/j.future.2020.03.021.