国際化ドメイン名の自動リンク処理等における セキュリティリスクの検討

白倉 大河[1]　長谷川 皓一[2]　山口 由紀子[2]　嶋田 創[2]

概要：日本語ドメイン名を始めとする国際化ドメイン名は普及しつつあり、多くのアプリケーションが対応している。一方でその処理は複雑で、プレーンテキストから URL を抽出する処理や、その URL の自動リンクを作成するといった処理での国際化ドメイン名の扱いはアプリケーションの実装に依存している。本研究では国際化ドメイン名の処理に際して、正規化処理を悪用したセキュリティ機構のバイパスや予期しないドメイン名と解釈されることで生じる問題について考察し、セキュリティ上の脅威について検討を行った。正規化や複数コードポイントで表現する絵文字等、複雑な処理や変換を十分に想定していない実装が存在することが確認された。

キーワード：Unicode, 自動リンク処理, Punycode

Security Risks in Automated Internationalized Domain Names Processing including Automatic Links

Shirakura Taiga[1]　Hasegawa Hirokazu[2]　Yamaguchi Yukiko[2]　Shimada Hajime[2]

Abstract: Internationalized Domain Names (IDN) that includes Japanese domain names are becomming common and supported by many applications. On the other hand, IDN processing such as extracting URLs from plain text and creating automatic links to the URLs are complex and depend on implementations of applications. In this study, we examine security threats in the processing of IDN such as bypassing security mechanisms by abuse of normalization processing and unexpected domain name extraction by wrong interpretations. We confirmed that some implementations are not sufficiently designed for complex processing and conversions, such as normalization and complex character (e.g. emoji that expressed with multiple code points).

Keywords: Unicode, Automatic Links, Punycode

1. はじめに

インターネットは日常生活に必要不可欠なものとなっており、Web ページへのアクセスが便利になるように、クリックすれば目的のページにアクセスできるリンクを自動的に作成するアプリケーションも多い。

国際化ドメイン名は従来 ASCII 文字の一部に限られていたドメイン名を拡張し、日本語を始めとする様々な言語を使用可能とする仕組みである。視覚的なわかりやすさから広く活用されており、Web ブラウザやメールソフトなど多くのアプリケーションで利用可能となっている。しかしながら、国際化ドメイン名の処理は複雑で実装がアプリケーション依存となっており、挙動が統一されていない。そこで、本研究では国際化ドメイン名の判別や処理に際して予期しない解釈が起こり、セキュリティ機構のバイパスや危険な処理が行われる可能性について調査し、セキュリティ上の脅威について検討を行った。

[1] 名古屋大学 Nagoya University
shirakura@net.itc.nagoya-u.ac.jp
[2] 名古屋大学 Nagoya University

2. 背景

2.1 Unicode の複雑性

Unicode はかつて国や環境ごとに異なっていた世界中の文字を統一されたコードで表すことを目指した規格である。Unicode は世界中の文字を収録することを目指しており、日常的に使われる文字以外にも、絵文字や記号、日本の旧字体、ヒエログリフのような古代文字等も含まれる。現在でも拡張が続いており新たな文字や記号が追加されている。

Unicode はかつては 16bit ですべての文字を表現できると考えられていたが CJK 統合漢字等の工夫を行ってもなお 16bit で収めることはできず、異体字セレクタや 2 文字分を使って一つの文字を表す UTF-16 のサロゲートペアが必要となった。また、絵文字の色を選択する制御文字や、複雑な文字を表現するための結合文字も近年の仕様に含まれており、一つの文字を複数のコードポイントで表現することも珍しくない。

このような状況においては、Shift-JIS や EUC-JP の判別失敗などに起因する不具合や文字化けは減少したものの、Unicode には UTF-8/UTF-16/UTF-32 といった複数の符号化形式と、エンディアンやバイトオーダーマークといったオプションの組み合わせが多数存在し、完全な解決には至っていない。また、Unicode の処理が不完全な状況なことに起因する問題もしばしば発生している。文字の結合の規則といった該当言語の使用者以外には理解が難しい事情も絡み、開発者が詳しくない言語を取り扱うと一般的な文字列でも問題が生じるよう処理が見落とされる可能性がある。

2.2 国際化ドメイン名

国際化ドメイン名 (IDN: Internationalized Domain Name) ではドメイン名に Unicode が利用できるように拡張を行っている。単純に使える文字を増やしただけではなく、利用者の利便性確保等を確保するためにいくつかの変換処理が行われる。一例として以下のような変換が行われる。

- か (U+304B) + ゛ (U+3099) → が (U+304C) のような結合
- 。 (U+3002) → . (U+002E) のような記号の置き換え
- ⓔ (U+24D4) → e (U+0065) のような装飾の除去

よって、「http://ⓔxample。ｃｏｍ」という URL へのリンクをクリックした場合、ブラウザは `http://example.com` に接続することとなる。これらの処理は IDNA(Internationalizing Domain Names in Applications)2003[1]、IDNA2008[2]、並びに PRECIS(Preparation, Enforcement, and Comparison of Internationalized Strings) Framework[3] のような RFC の規定を参考に実装されていることが多い。ただし、地域や Unicode のバージョンによって異なる変換が行われる可能性が存在する。

また、正規化とは別に、DNS のような既存の仕組みとの互換性を保つために従来の ASCII コードの範囲で表されるドメイン名と相互変換する Punycode と呼ばれる仕組みが存在する。Punycode によって Unicode 文字による表記 (U ラベル) と、従来の ASCII コードの範囲の表記 (A ラベル) は 1 対 1 に対応する。以下、U ラベルでの表記と A ラベルでの表記の例を挙げる。

U ラベルでの表記　日本語.jp

A ラベルでの表記　xn–wgv71a119e.jp

2.3 ドキュメントの複雑性

国際化ドメイン名の詳細を理解するためにはドメイン名や各種プロトコルの RFC のみならず、Unicode の技術文書、Web ブラウザのドキュメントといった複数のドキュメントを参照する必要がある。特に UTS46(Unicode Technical tandard #46) [*1] には国際化ドメイン名に関連する Unicode の技術情報がまとめられており、この文書を採用するかによって処理が異なってくる。また、技術文書に記述されている内容でも、現実の実装が遵守していない場合や古い Unicode に基づいている場合もある。

2.4 URL の抽出と自動リンク

プレーンテキストをやり取りするアプリケーションやサービスでは文章中に含まれる URL を自動的に判別し、リンク先のメタ情報を取得したり、クリック可能なハイパーリンクとして表示するものが数多く存在する。チャットクライアントやメールクライアント、スパム対策ソフト、SNS などのサービスが代表例である。国際化ドメイン名への対応は実装次第だが、多くのアプリケーションやサービスが対応している。しかし、URL を判定・抽出するアルゴリズムや URL の処理方法は単純ではなく、アプリケーションごとに異なる。結果として、クライアントやサービスごとに抽出される URL や作成されるハイパーリンクの先が異なるといった現象が発生する。

3. 想定される不具合

URL を処理する際に想定される、予期しない状況や不具合について検討する。メールやチャットアプリケーションが URL を含むプレーンテキストを解釈する状況を例に、処理の段階を以下の 4 つに分類して起こりうる問題をまとめた。

- 入力データの処理する際

[*1] `https://unicode.org/reports/tr46/`

- URL を解釈する際
- URL をプレーンテキスト中から抜き出す際
- URL を表示する際

3.1 入力データの処理する際

メールやチャットアプリケーションは URL に関連する処理を行う前に、HTML やメールファイル等をパースし、本文を取り出す必要がある。この際にも予期しない状況が発生しうる。

3.1.1 文字コードの判別や変換

Unicode が幅広く使われるようになった現在においても、アプリケーションによっては内部の処理や出力データの文字コードとして Unicode 以外の文字コードを使用している場合がある。例えば、日本語の電子メールにおいては UTF-8 以外に ISO-2022-JP がよく使用される。文字コードの変換は文字集合の違いや正規化の都合により、必ずしも同一の文字や元に戻せる形で変換できるわけではない。文字コードを変換した結果、異なる URL に解釈される文字列なってしまう可能性がある。

また、データを不適切な文字コードとして解釈させることが脆弱性につながった例も報告されている [4]。

3.1.2 エスケープや制御文字の除去

アプリケーションの処理によっては URL の処理を行う前に入力データに対してエスケープの解除や制御文字の除去等が行われることがある。例えば、改行で途切れている URL が次の行と接続され、後続の処理でより長い URL として処理される恐れがある。

ブラウザがセキュリティを向上させるために導入した XSS フィルタが、逆に XSS 脆弱性の原因となった例も報告されている [5]。

3.2 URL をプレーンテキスト中から抜き出す際

HTML やブラウザのアドレスバー等では URL として解釈すべき文字列は自明なものとして与えられる。しかし、プレーンテキストでやり取りを行う電子メールやチャットのクライアントアプリケーションの多くは、URL を自動判別してクリック可能なハイパーリンクとして表示する機能を有している。例えば、「詳細は http://example.com をご覧ください」という文章から「http://example.com」の部分がハイパーリンクとして表示される。

この処理は国際化ドメイン名への対応を行わない場合には http://や https://から始まり、スペースや URL に使わない記号までを抜き出すといった方法で実現できる。しかし、国際化ドメイン名は自由度が高く、URL を適切に抽出することが難しい。上記のテキストの場合は「`http://example.com` をご覧ください」という文字列も URL として有効であり、実際にそのように抜き出す実装も多数

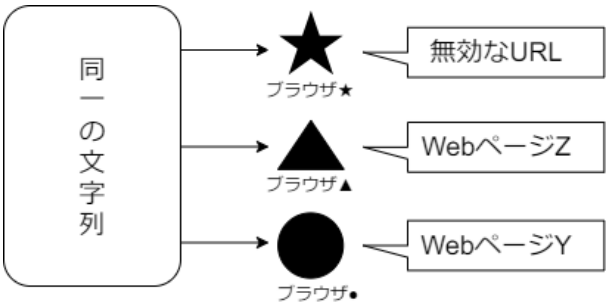

図 1 アプリケーションによって URL の解釈が異なる可能性

存在する。`http://`がなくても URL と認識したり、トップレベルドメインのリストを判定の方法に加えていたりする場合もあり、実装はアプリケーションごとに実装が大きく異なる。結果として、同じプレーンテキストでもアプリケーションによって違う URL が抽出されたり、URL として認識されないという事態が発生する。

3.3 URL を解釈する際

抽出した URL を解釈する際にも国際化ドメイン名では通常よりも複雑な処理が必要となる。解釈処理に違いがあった場合、それを開くブラウザや URL スキャンツールによって開かれる URL が異なるという事象が発生する (図 1)。

検証でも「http://■.example.com」(■ は U+1F468 U+200D U+1F9B0 の絵文字) という URL がブラウザによって異なるドメイン名に接続されたり、エラーになるなど挙動が異なることが確認された。

3.3.1 大文字小文字の正規化

ドメイン名はインターネット創生期より大文字小文字を区別せず、`example.com` と `EXAMPLE.COM` を同一ドメインとみなしてきた。この原則は国際化ドメイン名でも引き継がれているが、Unicode における大文字小文字は ASCII 範囲と比べてより複雑である。例えば、i の大文字は英語では I であるが、トルコ語では İ(U+0130) である。İ を小文字に変換する際も、国際化ドメイン名の処理で言及されている Unicode Inc. の SpecialCasing.txt を採用した処理では U+0069 U+0307 となり、採用しない処理では U+0069 となる。Web サーバ等の URL の処理系では前者の処理を行うのが原則だが、プログラミング言語やライブラリによっては後者に変換するものが存在する。データベースへの登録時やホワイトリストとの照合を小文字に統一した上で行うと、処理の内容や順番、地域の設定等によっては本来一致しないはずの URL が一致したり、その逆の事象が起こる可能性がある。

3.3.2 正規化と検証

IDNA では以下のような規則で変換が行われる。

1. IDNA Mapping Table に基づいて変換と無効な文字の検出等を行う。この際に句点 (U+3002)、半角句点

(U+FF61)、全角ピリオド (U+FF0E) を半角ピリオド (U+002E) に変換すると言った処理が行われる。

2. NFKC(Normalization Form Compatibility Composition) に基づいて正規化
3. FULLSTOP(U+002E) でラベルに分割
4. ラベルの Punycode の処理や検証

しかし、これらの処理や検証が不適切な場合、予期しない文字が生成される可能性がある。例えば、正規化の結果に「.」や「/」が含まれる文字が Unicode には存在する。本来は上記のマッピング時に無効な URL としてマークされるはずだが、そのプロセスを怠ると別のドメインにつながる URL が生成される恐れがある。実際にこの問題に起因する脆弱性が HostSplit として 2019 年に報告されている [6]。

また、正規化による文字の変換や検証結果は Chromium と Firefox のような著名な処理系でも異なることが確認されている。絵文字中のゼロ幅接合子 (U+200D) が削除されるかどうかの違いや、Chrome では無効な URL として処理される文字列が、Firefox では無効な文字を REPLACEMENT CHARACTER (U+FFFD) に置き換えられた状態で処理を続行する状況が存在する。

3.3.3 国際化ドメイン名の形式変更

Punycode では仕様上 U ラベルに変換すると不適切な文字や、正規化されていない文字も表現することができる。また、Punycode としてデコードできなくても有効な A ラベルとして解釈ができる。UTS46 では A ラベルと U ラベルが正常に変換を行えるかの検証方法を示しているが、実装ごとに検証が不完全だったり、検証処理を実装していないアプリケーションやライブラリが多く存在するのが実情である。

3.3.4 Unicode バージョンによる差異

Unicode は年々機能の拡張が続いており、バージョンによって新たに文字の定義が追加されるだけでなく、既存の文字の分類が変更になったこともある。例えば、U+30FB KATAKANA MIDDLE DOT (・) は Unicode 4.1 のリリース時に Gerneral Category が Pc(Punctuation_Connector) から Po(Punctuation_Other) に修正され、Java や C#のコンパイラがそれに追従したことで従来のソースコードがコンパイルエラーになる現象が発生した[*2]。つまり現在は安全な処理でも、ライブラリや OS の更新によって破壊的な変更が発生し、不具合や処理漏れが発生する可能性がある。

3.3.5 複雑な文字の処理

Unicode は様々な需要や問題に対応するため、従来の文字コードと比べて複雑な仕組みが存在する。例えば、UTF-16 において通常の 2 文字分を使用して表現するサロゲートペア、複数コードポイントにまたがって一文字を表示する結合文字、異体字セレクタやゼロ幅接続子等の複雑な制御文字といった仕組みは Shift-JIS には無い機能であったり、Unicode の途中のバージョンで追加されたものである。アプリケーションによってその対応具合や処理の内容はまちまちである。現在全てに対応しているアプリケーションであっても、Unicode の仕様の追加や変更に追従し続けなければ不完全な処理となるかもしれない。

[*2] http://www.unicode.org/reports/tr44/tr44-4.html#Change_History

図 2 画像に置き換えられた絵文字

3.4 URL を表示する際

3.4.1 絵文字の表示

国際化ドメイン名では仕様上絵文字を使うことができる。.jp では絵文字の登録を認めていないが、.com を始めとするいくつかのレジストリでは実際に登録可能となっている。

絵文字の表示は文字列以上に環境やフォントごとに見た目が異なったり、正常に表示できない場合がある。クライアントや Web ページの中には絵文字を統一感を保って適切に表示するといった目的で img タグのような画像に置き換える処理を行うものが存在する。絵文字が URL 中に含まれる状況の考慮が不十分な場合などに表示が崩れたり、URL が適切に解釈されない可能性がある。

3.4.2 予期しない文字列の生成

アプリケーションの中には、A ラベルで与えられたドメイン名を U ラベルに変換したり、正規化を行って表示するものが存在する。この際に画面の表示と実際に接続される URL が異なってしまったり、"や>といったアプリケーションやスクリプトで特殊な意味を持つ文字が出現する可能性がある。正規化やエスケープのタイミングを誤ると、表示が崩れたり、セキュリティ上の問題につながる可能性がある。

3.5 複数のアプリケーション間の連携

URL の解釈や処理は単独でも複雑だが、現実には複数のアプリケーションを経由することでより複雑となる。例えば、URL を含むメールを送信し、受信者がブラウザで開いた場合、以下のようなソフトウェアが URL に触れる事となる。

- 送信者のメールクライアント
- スパムフィルタ
- 受信者のメールクライアント
- メールをチェックするアンチウイルスソフト
- ウェブブラウザ
- DNS

- OS

これらの全てがプレーンテキストに含まれた URL を同一のものとして処理できれば問題は発生しない。しかし、一つでも URL の解釈が異なったり、固有のエスケープ等をした状態で他の処理に引き渡すなどした場合、予期しない結果を招く可能性がある。

4. 想定される攻撃

4.1 スパムフィルタ等のバイパス

メールや SNS には貼り付けられた URL の検証を行い、スパムやマルウェアなどの危険があれば警告をする機能を有するものがある。しかし、サーバとクライアントで解釈が異なる場合、スパムフィルタはバイパスされてしまう。例えば、「example.com。ｅｖｉｌ。ｃｏｍ」はスパムフィルタでは「example.com」、クライアント側では「example.com.evil.com」として解釈される危険がある。

4.2 予期しない URL の参照

認証が必要なサービスの中には、識別情報や機密情報を指定された URL に送信する事がある。この際にアプリケーションごとに URL の解釈が異なると、別のドメインに対して機密情報を送信してしまう恐れがある。たとえば、「https://evil.c[a/c].office.com」([a/c] は U+2100 の組み文字) という URL に対し、「office.com のサブドメイン」と解釈するサービスやアプリケーションと「https://evil.ca/c.office.com」と認識して evil.ca にデータを送信するサービスやアプリケーションが混在した場合、機密情報の流出等の問題につながる [6]。

また、「192.168.0.1。example.com」という URL を「example.com のサブドメイン」と解釈するサービスと「192.168.0.1」と解釈するサービスが混在したした場合、プライベートアドレスのデータを取得して体部に送信する、サーバサイドリクエストフォージェリの脆弱性につながる可能性がある。

4.3 XSS や SQL インジェクション

絵文字を含む URL や、正規化によって”や<が出現するドメイン名を与えることで、それを想定していないアプリケーションが不適切な HTML や SQL クエリを生成する可能性がある。処理の内容によっては、XSS や SQL インジェクション、DoS 攻撃などにつながる可能性がある。

5. 現実のアプリケーションやサービスでの検証

5.1 国際化ドメイン名処理での不具合

これまでに、Web ブラウザや URL を認識するアプリケーションの間で URL の解釈に差があり、スパムやセキュリティ上の検証をバイパスされる危険性について指摘されている [7]。本研究では国際化ドメイン名の場合も同様の問題が生じる可能性を検討した。

5.1.1 スパムフィルタバイパスの検討

メールや SNS ではしばしばフィッシング詐欺やマルウェアの URL が送りつけられることがある。メールクライアントではハイパーリンクとして表示されるため、クリックするだけで簡単に危険な URL に接続されてしまう。スパムメールフィルタやサーバによるメッセージの監視では、そのような文章中の URL を悪意のある URL のリストと照合し遮断すると言った方法でユーザを保護している。しかし、このセキュリティ機構が国際化ドメイン名の解釈によって想定したとおりに働かない場合を検証した。

ここでは example.com という URL がスパムメールであると判断される環境を想定する。国際化ドメイン名の処理によって以下のような攻撃の可能性がある。

1. 攻撃者は example.com の一部を別の文字に書き換えて送信する。たとえば、「http://example。com」とピリオドを句点に変換する。
2. 一部のスパムフィルタのようなスキャンプログラムは書き換えられた URL を URL と認識できないか、example.com とは別の URL と認識してフィルタを通過させる。
3. 国際化ドメイン名を解釈するメールクライアントや SNS は URL を example.com へのハイパーリンクとして表示する。

5.1.2 メールクライアント

文書中に半角英数字以外の文字列を含むドメインがメールクライアントでどのように表示されるかを確認した。確認した結果を表 1 に示す。各メールクライアント名は匿名化してあり、アプリ版と Web 版があるサービスに関してはカッコ内に種類を示した。処理結果は、全体がリンクになった (○)、自動リンクが行われなかった (×)、及び部分的にリンクが作成された結果のリンク先を示してある。

今回試したメールクライアントはすべて自動リンクの機能を有しており、国際化ドメイン名に対応しているものも多かった。しかし、その処理はアプリケーションごとに異なり、同一の開発元の場合ですらアプリ版と Web 版で作成されるハイパーリンクの先が異なる場合があることが確認された。

メールサービス F の Web 版クライアントでは絵文字を画像に置き換える処理が存在するが、URL 中の絵文字でもその置き換えが行われる。その結果、図 3 のようにに UR の前に HTML タグが付加された崩れた表示となった。さらに、ソースを確認したところ、図 4 のようにタグ中の URL の表示に失敗していることがわかった。この不具合についてバグレポートとして報告し、現時点では先方からの返信は得られていない。

表 1　各メールクライアントにおける国際化ドメイン名の処理の違い

メールクライアント	http://example. 例え。テスト	http://exampl⓮.com	http://アカ🈂タ.example.com
A	○	○	○
B	http://example. 例え	○	http://アカと http://タ.example.com
C(iOS/Android)	○	×	×
C(Web)	○	○	○
D	http://example.	http://exampl	http://.example.com
E	○	○	○
F(Web)	○	○	(壊れた HTML が生成され、表示が崩れた)
F(iOS/Android)	○	○	○
G	http://example	http://exampl	http://
H	http://example. 例え	○	http://アカサタ.example.com
I	×	×	×

図 3　メールサービス F における表示崩れ

図 4　メールサービス F でブラウザが解釈したソースコード

5.1.3　スパムフィルタの動作検証

オープンソースのスパムフィルタリングツール X を用いて検証した。X が動作するメールサーバを用意し、PhishTank で有害と登録された URL を含むパターン 1、パターン 2 のメールについてスパムと判断されるか確認した。

(パターン 1) PhishTank の URL をそのまま本文に書く

(パターン 2) . (U+002E) を。(U+3002) に置き換えて書く

10 件の URL で検証を行ったところパターン 1 では 5 つがスパムと判定された。パターン 2 では一つもスパムと判断されなかった。通信を確認すると、パターン 1 ではブラックリストへの問い合わせを行っているが、パターン 2 では行っていないことが確認された。

スパムフィルタリングツール X は PhishTank などの情報を元にスパムメールをブロックする機能を有するが、一部を別の文字に変えることで URL として認識しなくなり、フィルタをすり抜けてしまう状況があると言える。表 1 に示すように、多くのメールクライアントは。を. に変換して自動リンクするため、スパムフィルタとしての機能を妨害することに成功したといえる。仮にアルゴリズムを変更して特定のメールクライアントと同じ方式で URL を解釈することにした場合でも解釈の異なるメールクライアントの方式ですり抜けてしまう危険を完全に防ぐことはできない。この挙動について開発元に報告し、現時点では先方からの返信は得られていない。

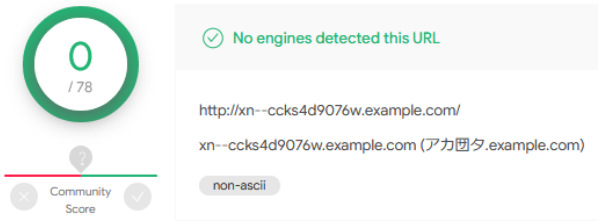

図 5　URL スキャンサービスの結果

5.1.4　URL スキャナの検証

複数の URL スキャンサービスについても、国際化ドメイン名を含む URL(http://アカ🈂タ.example.com) を手作業で入力し検証を行った。以下の 3 つのパターンに別れた。

1. エラーや正常に検査が行えない、不正な文字として除去される等、国際化ドメイン名を正常に扱えない
2. ブラウザと同様の xn--ccks4a2b.example.com の検証結果を表示した
3. U ラベルとして正規化が行われていない xn--ccks4d9076w.example.com の検証結果を表示した (図 5)

アカサタ (xn--ccks4a2b) とアカ🈂タ (xn--ccks4d9076w) の両方のサブドメインを受け付けるサーバを用意し、3 の結果を示したサービスでスキャンを行った。その結果、xn--ccks4d9076w のドメインのみにアクセスがあったことから、このサービスは xn--ccks4a2b.example.com ではなく xn--ccks4d9076w.example.com の調査のみを行っていることが確認された。

5.2　正規化されていない Punycode での不具合

Punycode では U ラベルにしたときに正規化されていない文字も表現することができる。また、そのような Punycode を含む A ラベルも URL として有効である。U ラベルに戻す処理が挟まる場合、正常に処理できない可能性がある (図 6)。

そこで、以下の考えのもと、🈂という絵文字を含むラベルに付いて実験を行った。

1. xn–677h という A ラベルは Punyocde の規則に基づい

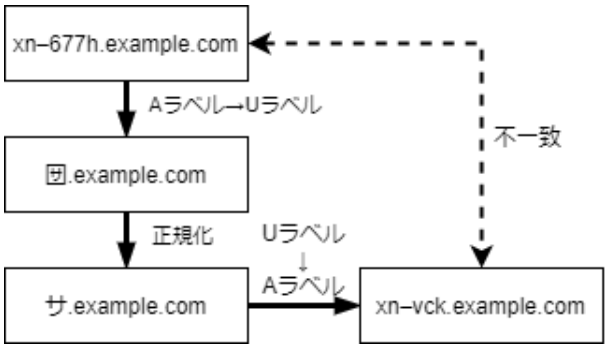

図 6　正規化されていない A ラベルの処理

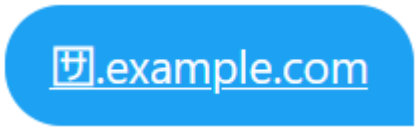

図 7　アプリケーション⑦の表示

て U ラベルに変換すると㋮という絵文字となる

2. この絵文字を国際化ドメイン名の規則に基づいて正規化すると、カタカナの「サ」となる
3. 「サ」という U ラベルをは Punycode の規則に基づいて A ラベルに変換すると xn–vck となる。

そこで、`http://xn--677h.example.com` という URL を用意し、複数の SNS やアプリケーションでどのような自動リンクが生成されるかを確認した。

アプリケーション①, ②, ③, ④
`http://xn--677h.example.com` へのリンクになる。

アプリケーション⑤ `http://xn--677h.example.com` がハイパーリンクになるが、クリックしても反応しない。

アプリケーション⑥ `http://xn--677h.example.com` がハイパーリンクになるが、クリックするとエラーが表示される。

アプリケーション⑦ `http://xn--677h.example.com` へジャンプする短縮 URL が表示される。画面の表示は U ラベルに直した㋮`.example.com` となる (図 7)。表示された URL をブラウザのアドレスバーに入力した場合、ブラウザは㋮を正規化してサ`.example.com`(`xn--vck.example.com`) に接続するため、クリック時と異なる Web ページに接続されるという現象が発生する。

5.3　レジストラの問題

ドメイン名を取得する際に、レジストラは使用可能な文字列に制限を加えている。これはホモグラフ攻撃によるフィッシングを予防したり、混乱を招くドメイン名の取得を阻止するのが目的である。例えば、.jp ドメインでは日本語を含む日本語.jp やテスト A.jp といったドメインを取得できるが、テスト β のようなギリシャ文字やキリル文字を含むドメイン名は取得できない。

しかし、サブドメインの管轄は各ドメイン名の管理者に委ねられており、著名なホスティングやクラウドサービスでもサブドメイン部分に任意の文字列を使用可能なものが存在する。Unicode や xn–から始まるラベルの仕様を許可している場合、レジストラの規制よりも緩いドメイン名を作成することが可能である。

6.　まとめ

本研究では国際化ドメイン名の抽出や自動リンクの挙動について複数のアプリケーションの動作を調査した。Unicode や国際化ドメイン名の処理は IDNA2003 から IDNA2008 の移行、UTS46 の採用、Unicode の仕組みの理解など複雑で考慮すべきことが多岐にわたる。そのため、いくつかのクライアントで、クリックしても反応がなかったり、エラーが表示されるハイパーリンクが作成されることが確認された。また、利用者の見た目を優先するためか、A ラベルを U ラベルに戻す処理を行ったことで表示とリンク先で整合性が取れないという問題が生じるサービスも存在した。

自動リンクの結果が統一されていないことからわかるように、ドメイン名が絡む処理を実装する機会は多いにも関わらず、利用者の使い勝手を損なわず、かつ不具合やセキュリティ上の問題を起こさないようにする方針は十分に整備されているとは言い難い。今後、国際化ドメイン名の処理の差異がより広範囲に影響を与える状況を精査し、統一的な基準を用意できないかを検討していく予定である。

謝辞

本研究の一部は JSPS 科研費 JP19H04108, JP19K11961 の助成を受けたものである.

参考文献

[1] P. Faltstrom, P. Hoffman, and A. Costello. Internationalizing Domain Names in Applications (IDNA). RFC 3490, IETF, March 2003.

[2] J. Klensin. Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework. RFC 5890, IETF, August 2010.

[3] P. Saint-Andre and M. Blanchet. PRECIS Framework: Preparation, Enforcement, and Comparison of Internationalized Strings in Application Protocols. RFC 8264, IETF, October 2017.

[4] Daniel Bates, Adam Barth, and Collin Jackson. Regular expressions considered harmful in client-side XSS filters. pp. 91–100, January 2010.

[5] Masato Kinugawa. MKSB(en): X-XSS-Nightmare: XSS Attacks Exploiting XSS Filter. `https://mksben.l0.cm/2015/12/xxn.html`, 2018. [Online;

accessed August 20, 2020].

[6] Jonathan Birch. Host/split: Exploitable antipatterns in unicode normalization, August 2019. Black Hat USA 2019.

[7] Qilang Yang, Dimitrios Damopoulos, and Georgios Portokalidis. WYSISNWIV: What You Scan Is Not What I Visit. Vol. 9404, pp. 317–338, November 2015.