Computer Security Symposium 2019
21 - 24 October 2019

サーバ証明書解析によるフィッシングサイトの発見手法

櫻井 悠次[1,a)]　渡邉 卓弥[2]　奥田 哲矢[2]　秋山 満昭[2]　森 達哉[1,3]

概要：Web サイトの急速な「HTTPS 化」に伴い，フィッシングサイトの HTTPS 化も一般化しつつある．HTTPS を採用することは攻撃者にとって次の利点が存在する．1) アドレスバーに鍵マークが表示されることで作成した Web サイトを本物に似せることができる．2) 暗号化されたトラフィックによってネットワーク上における URL ベースの検知から逃れることができる．一方で，Web サイトの HTTPS 化はフィッシングサイトを発見する上で有用な「痕跡」を残す可能性がある．なぜなら，HTTPS 化を行う際には正当な公開鍵証明書を PKI に登録する必要があるからである．以上のことを踏まえて，本研究では証明書のコモンネーム（CN）から抽出したテンプレートを用いた，フィッシングサイト発見手法を提案する．本手法により，未知のフィッシングサイトの発見およびフィッシングサイトを生成するエコシステムの詳細な理解が可能であることを示す．また，本研究で得られた知見を基に，無料の証明書発行サービスを行う認証局への提言を行う．

キーワード：フィッシング，HTTPS，公開鍵証明書，クラスタリング

Discovering the HTTPSified Phishing Websites with the TLS Certificates Analysis

Yuji Sakurai[1,a)]　Takuya Watanabe[2]　Tetsuya Okuda[2]　Mitsuaki Akiyama[2]　Tatsuya Mori[1,3]

Abstract: With the recent rise of HTTPS adoption on the web, attackers have also started "HTTPSifying" phishing websites. HTTPSifying a phishing website has the following benefits for an attacker. First, it may somehow contribute toward making the site appear legitimate. Second, it will disable the URL-based detection in the network because TLS encryption will wipe out the information of the full URL path. On the other hand, adopting HTTPS could also contribute to generating intrinsic *footprints*, which can in turn be used to systematically detect such phishing sites. This is because they need to register a public key certificates in advance. Given this background in mind, we conduct an extensive analysis of public key certificates collected from a large set of HTTPSified phishing websites. We demonstrate that a *template* of common names, which are equivalent to the FQDN of the subjects of the certificates, obtained through a clustering analysis of the certificates can be used for the following promising applications: discovering previously *unknown* phishing websites, and understanding the infrastructure used to generate the phishing websites. We use our findings on the abuse of free Internet services (such as free certificate authority) to provide recommendations to associated stakeholders.

1. はじめに

Web サイトの HTTPS 普及率がこの数年で急速に増加している [8,9]．Google の報告 [9] によると，アメリカやドイツ，フランスなどの国々では 90%以上の Web トラフィックが「HTTPS 化」している．その傾向は他の国々でもみられるようになってきており，ブラジルや日本，インドでは 70%以上の Web トラフィックが HTTPS 化している．この急速な増加にはブラウザにおける検索ランキングの変更 [11] やセキュリティ指標の変更 [10,15]，容易に Web サイトの HTTPS 化を行えるツールの登場 [13,22] などが重要な役割を果たしている．Naylor [16] らは HTTPS 化にか

[1] 早稲田大学 (Waseda University)
[2] NTT セキュアプラットフォーム研究所 (NTT Secure Platform Laboratories)
[3] 情報通信研究機構 (NICT)
[a)] s5i@nsl.cs.waseda.ac.jp

かるコストが近年極めて低くなっていることを指摘した．こうした周辺状況の変化も HTTPS 化した Web サイトの増加に貢献している．

急激な Web サイトの HTTPS 化に伴い，フィッシングサイトの HTTPS 化も普及し始めている．HTTPS 化することで Web サイトのアドレスバーに「保護されていない通信」と表示されることがなくなるため，攻撃者は自身が作成したフィッシングサイトの信憑性を高めることができる．また，HTTPS 化された Web サイトのサーバとクライアント間の通信は TLS により暗号化されるため，ネットワーク上の URL や Web コンテンツをベースとした検知を回避することができる．さらに，Let's Encrypt [13] や cPanel [5] などの証明書を無料で提供する認証局の登場により，攻撃者は容易に HTTPS 化を行うことが可能となった．APWG が発表した 2019 年第一四半期の Phishing Activity Trends Report [1] によると，2015 年では HTTPS 化したフィッシングサイトは全体の 2%程度だったが，その割合は 2016 年から急激に増加し始めていき 2019 年には 58%にまで到達している．

フィッシングサイトの HTTPS 化には上記で示したように様々な利点が存在する一方で，新たなフィッシングサイトの発見につながる「痕跡」を残す可能性がある．攻撃者は HTTPS 化の際に正当な公開鍵証明書をいずれかの認証局から発行してもらう必要があり，その証明書には Web サイトの FQDN を表すコモンネーム（以降，CN と呼ぶ）[*1]や，発行を行った認証局を表す発行者名，発行日などの攻撃者の特徴を表す情報が記載されているためである．さらに，証明書発行の段階でフィッシングサイトの発見を試みることは．近年増加している DDNS やホスティングサービスを利用する攻撃に対して非常に有効である [20, 21]．それらのサービスでは既存のドメインを使用してホストネームを作成するため，証明書情報を用いた探索が最も早い段階での発見につながるからである．以上の背景を踏まえ，本研究では次の Research Question (**RQ**) を掲げる．

RQ: フィッシング攻撃を防ぐために，公開鍵証明書の情報をどのように活用できるか？

上記の **RQ** に取り組むため，本研究では HTTPS 化されたフィッシングサイトを収集し，その証明書を抽出する．収集したフィッシングサイトに使用された証明書の大規模な分析により，証明書の CN をクラスタリングすることにより得られたテンプレートが，以下の有用な応用に役立つことを明らかにした．

- 未知のフィッシングサイトの発見
- フィッシングサイトを生成するインフラの理解

さらに，本調査により前例のない，多くの充実した機能をもつフィッシングキットの存在が明らかとなった．それらの機能には自分でカスタマイズして大量のスパムメールを送信することができる大量スパムメール送信機能，奪取したアカウントの売買が可能なマーケット機能，通知機能，ロギング，アナリティクスなどが含まれている．

本論文の構成は以下の通りである．2 章ではフィッシングサイト検知と公開鍵証明書の背景知識について，3 章では本研究で提案するフィッシングサイト発見手法について，4 章では使用したデータについての説明を行う．5 章では提案した手法による分析・探索結果およびフィッシングサイトを生成するエコシステムの一例を示す．6 章では本研究の制約および今後の課題について述べ，無料の証明書発行サービスを行う認証局への提言を行う．7 章では関連研究について報告し，8 章にて本論文のまとめを行う．

2. 背景

フィッシングは最も一般的なサイバー攻撃の一つであり，広く世間に認知されている比較的単純な攻撃であるが，現在でも非常に大くの被害をもたらしている．IC3（Internet Crime Complaint Center）の報告によれば，2018 年において web phishing，vishing（voice），smishing（SMS）による被害件数は 26,379，被害総額は 4,820 万米ドルに及んだ [12]．それらの攻撃は本物に類似した偽サイトやメールアドレスを使用して，オンラインバンキングで用いられるユーザの重要な個人情報などの窃取を目的としている．典型的なフィッシングでは，標的のユーザが送られたメール内の悪性 URL をクリックしてしまうように攻撃者は特定のブランドやサービスになりすます．

フィッシング攻撃への対策として，これまでにドメイン名 [24] や URL [2]，コンテンツ [29]，メール [27] を基にフィッシング攻撃の特徴分析や検知を行う手法が提案されてきた．しかし，これらの手法にはいくつかの制限が存在する．ドメイン名の登録を監視することにより，将来悪性目的に使用され得るドメイン名をプロアクティブに検知することが可能となるが，既存のドメイン名に紐づけられた DDNS や Web ホスティングサービスを利用した Web サイトによる攻撃は見逃してしまう．URL やコンテンツ，メールの特徴を用いて検知を行う手法の中には非常に高い精度で検知を行えるものが存在するが [17]，それらの手法は本質的にリアクティブであり，事前の検知はできない．

上記で示した既存手法とは異なり，公開鍵証明書を使用する手法では CT（Certificate Transparency）ログを活用する．CT は不正な証明書発行を防ぐ仕組みとして導入され，Chronium プロジェクトでは 2018 年 4 月より新たに発行される証明書は CT ログサーバに登録されていることを

[*1] 近年，ブラウザでは CN の代わりにサブジェクト代替名を確認することにより証明書の妥当性を検証することがある．サブジェクト代替名とは複数のホストネームを格納できる証明書のフィールドである．多くの証明書では，サブジェクト代替名に格納されているホストネームは CN に記載されていることが経験的に明らかとなっているため，CN を使用した調査とサブジェクト代替名を使用した調査に分析上の差は小さいと考えられる．

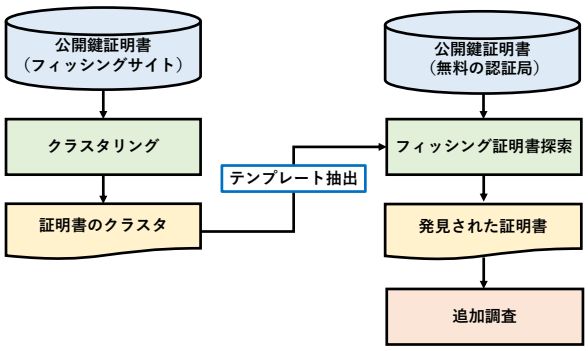

図 1 フィッシングサイト発見手法および調査の処理フロー

必須としており [25]，現在のインターネット上のほぼ全ての証明書は CT ログサーバに登録されている．Censys [3] により調査を行ってみると，2018 年 4 月以降に発行された証明書は 635.7 M 存在し，その内の 99.3%は認証局により CT ログに登録されていた．その認証局には無料で証明書発行を行うことが可能な Let's Encrypt [13] や cPanel [5] なども含まれているため，そのいずれかの認証局に発行を依頼した顧客の証明書は自動的に CT ログへ登録される．つまり，攻撃者が既存のドメインを利用してホストネームを生成する DDNS やホスティングサービスを利用していたとしても，発行された証明書からその攻撃をプロアクティブに検知することができる．

3. 調査手法

本章では初めに本研究で提案するフィッシングサイト発見手法を含む調査の処理の流れを示す．次に，フィッシングサイトに使用された証明書に対するクラスタリングの手法について述べる．クラスタリングにより，同様の攻撃インフラから生成されたと考えられる証明書のクラスタを抽出できる．最後にクラスタからフィッシングサイトの発見に利用するテンプレートの抽出手法について述べる．

3.1 調査概要

図 1 に本研究で提案するフィッシングサイト発見手法および調査の概要を示す．初めに，フィッシングサイトに使用された URL より対応する証明書を収集し，3.2 節に示すクラスタリングを適用してクラスタを抽出する．次に 3.3 節に示す手法によりクラスタから抽出したテンプレートを用いて，フィッシングサイトに使用され得る証明書を探索する．そのために，本研究では無料の証明書発行に対応している認証局から発行された証明書を探索対象のデータとして利用する．5.3 節では本研究の提案手法により発見した潜在的なフィッシングサイトの統計を示す．5.4 節では発見した証明書の解析によりフィッシングサイトを生成するエコシステムが明らかとなった例について述べる．

3.2 クラスタリング

本研究では互いに関連性があると考えられるフィッシングサイトのグループ（クラスタ）を特定する．各クラスタ内において共通するパターンを抽出することにより，未知のフィッシングサイトの発見につながる特徴の特定を狙う．この目的を達成するため，本研究ではフィッシングに使用された証明書の CN に対してクラスタリングを適用する．

クラスタリングを行う前に，本研究では次のデータ処理を行う．初めに，`www` のホスト名は多くの CN に使用されている普遍的な文字列であるため，CN から削除する．次に，文字列長の短い CN は類似性を決定する際の曖昧さを回避するために削除する．例として `appse`（`.com`）や `appla`（`.com`）, `apps`（`.com`）という CN がデータに含まれていた場合，これらは非常に類似しているためにクラスタリング処理後に同一のクラスタに取り込まれる．しかし，明らかにその類似性に意味はなく，それぞれ独立したドメインである．本研究では経験的に調査対象とする文字列長の閾値を 10 と決定し，それよりも短い CN は対象データから取り除く．

本研究ではクラスタリングアルゴリズムとして DBSCAN，2 つの文字列（CN）の類似度を求める距離関数として *Ratcliff-Obershelp* アルゴリズム [23] を採用する．*Ratcliff-Obershelp* アルゴリズムは再帰的に最長共通部分文字列（LCS）を求める処理によって類似度を計算する．2 つの文字列 (x, y) における類似度は $d(x, y) = 2M/T$ によって導出される．この時，M は再帰処理によって得られた LCS の長さの合計であり，T は $T = |x| + |y|$（$|s|$ は文字列 s の文字列長を表す）と定義される．つまり，類似度は $[0, 1]$ の範囲で表され，1 に近づくほど類似度は高い．

最後に，その構造や文字列長が異なる CN に対して分析を行うため，本論文では CN に含まれるドット数を表す新たな変数 m を導入する．m は CN に `www` が含まれていた場合にはそれを削除した後のドット数を表す．例えば，`ieee.org` という CN は $m = 1$ であり，`www.cs.example.edu` は $m = 2$ である．クラスタリングは同一の m の値（$m = 1,\ 2,\ 3,\ 4,\ \geq 5$）をもつ CN 同士で行う．これは，同一の攻撃者が生成した CN は同一のドメイン構造を保っている，つまり，CN に含まれるドット数は同じであるという経験的に推測された仮定に基づいている．また，CN の TLD は普遍的に多く使用されるものが多いことに加え，TLD のみを変更して多くの CN を生成する攻撃者も存在することから，クラスタリングの際には比較対象としない．

図 2 に $m = 2$ である場合のクラスタリング処理の例を示す．いくつかの証明書の CN が類似していた場合，それらをクラスタとしてグループ化し，どのクラスタとも類似していない CN の証明書は外れ値として削除する．DBSCAN では 2 つのパラメータ（ϵ, $minPts$）を設定する必要があ

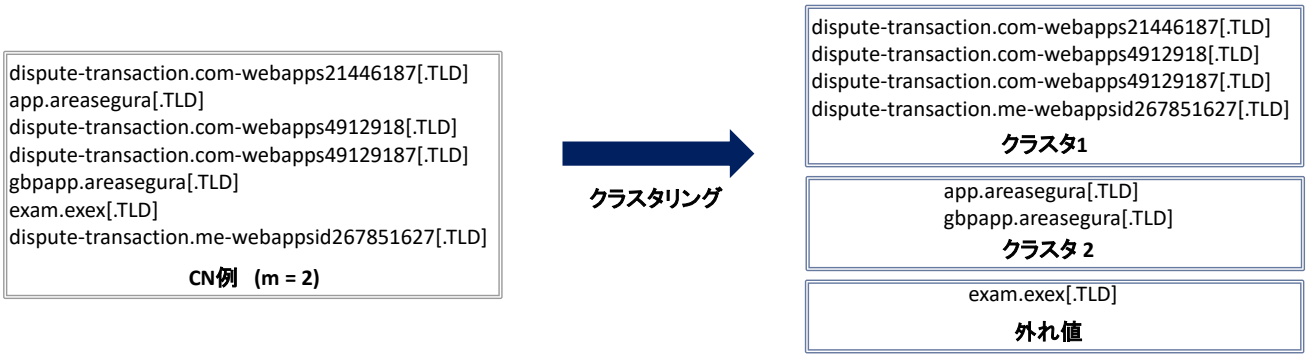

図 2　$m = 2$ におけるクラスタリング処理の例

表 1　$m = 2$ における，ϵ の値が 0.25, 0.30 および 0.35 である場合のクラスタリング結果

	クラスタ
$\epsilon = 0.25$	login.portaleprivatimps[.TLD] secure.portaleprivatimps[.TLD] accesso.portaleprivatimps[.TLD]
$\epsilon = 0.30$	login.portaleprivatimps[.TLD] secure.portaleprivatimps[.TLD] accesso.portaleprivatimps[.TLD] secure.mpsprivati[.]com
$\epsilon = 0.35$	login.portaleprivatimps[.TLD] secure.portaleprivatimps[.TLD] accesso.portaleprivatimps[.TLD] secure.mpsprivati[.TLD] certificazione.areaprivatimps[.TLD] certificazione.portalemps[.TLD] certificazione.mpsprivati[.TLD]

るが，本研究では ϵ のみを調節してクラスタ内の CN の類似度を調整し，クラスタ内の最小の証明書数を決定する指標である $minPts$ は 2 で固定する．

表 1 に ϵ の値がそれぞれ 0.25, 0.30, 0.35 である場合のクラスタリング処理の結果例を示す．表に示す通り，ϵ の値が 0.25 であるときのクラスタ内の CN は全て類似しているが，それ以外のクラスタ内には類似していない CN が混ざっている．したがって，$m = 2$ である場合の ϵ の値は 0.25 と決定する．その他の m においても同様の処理を行い，$m = 1$ の時の ϵ は 0.24，$m = 3$ では 0.3，$m = 4$ では 0.33，$m \geq 5$ では 0.35 とする．

3.3　テンプレート抽出

3.2 節にて取得したクラスタからテンプレートを抽出する処理フローを図 3 に示す．まず，クラスタ内の全ての CN に共通する部分文字列を抽出する．この時，共通部分文字列の抽出は各ドットで分割されたセグメントごとに行い，3 文字以上の文字列に限定する．また，クラスタリングの際と同様に www および TLD は除いて処理を行う．次に，クラスタ内における各 CN の共通部分文字列以外の文字列を正規表現化し，それぞれの正規表現をマージする．マージの際には共通部分文字列を残し，正規表現化された部分における文字列長の最小値と最大値を抽出して統合する．例えば [a-z]{3} と [a-z]{5} をマージさせると正規表現は [a-z]{3,5} となる．アルファベットと数字の両方が含まれていた場合には [a-z0-9] としてどちらのデータタイプも表現する．

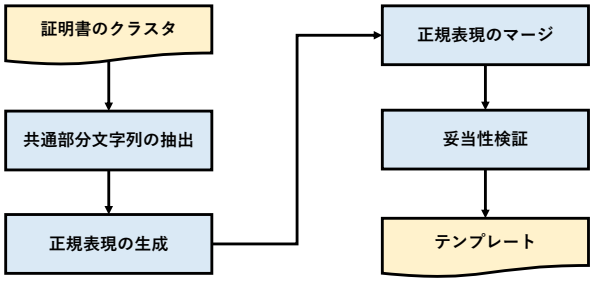

図 3　テンプレート抽出の処理フロー

最後に，得られた正規表現の「汎用性」を評価する．正規表現があまりに汎用的である場合，その表現を用いた検出には多くの誤検知が含まれる可能性がある．以下では正規表現の汎用性を評価する尺度として，Xie [28] らが提案したエントロピー削減量 (entropy reduction) を用いる．ある正規表現を e とする．$B_e(u)$ を正規表現 e を用いて文字列 u を表現した際に，その表現を 2 進数で符号化するために必要となるビット数の平均値（情報エントロピー）とする．同様に $B(u)$ は正規表現を使用せずに文字列 u を表現する際の情報エントロピーとする．ランダムな文字列に対する情報エントロピーは $L \log_2 N$ のように計算できる．ここに L は u を構成する文字数であり，N は使用可能な文字数を表す．このように定義される情報エントロピーはパスワードの強度を測る目的にも用いられる．

元の文字列の情報エントロピー $B(u)$ およびテンプレート適用後の文字列の情報エントロピー $B_e(u)$ は以下のように計算される．

$$B_e(u) = \overline{L_a}\log_2 A + \overline{L_d}\log_2 D + \overline{L_{ad}}\log_2(A+D) \quad (1)$$

$$B(u) = L\log_2(A+D) \quad (2)$$

ここに $\overline{L_a}$, $\overline{L_d}$, $\overline{L_{ad}}$ はそれぞれ正規表現 `[a-z]` (alphabet), `[0-9]` (digits) および `[a-z0-9]` (alphabet + digits) で表される文字数の平均値を表す．A, D はそれぞれ `[a-z]`, `[0-9]` で表現可能な文字の数であり，具体的な数値は $A = 26$, $D = 10$ である．

テンプレート化によるエントロピー削減量は $d(e) = B(u) - B_e(u)$ によって計算できる．$d(e)$ が小さい正規表現はより汎用的な表現であるため，誤分類が生じる．すなわちテンプレートとしては適さない．本研究では経験的に定めた閾値を用い，$d(e) \geq 55$ となった正規表現をテンプレートとして抽出し，そのパターンに一致する CN をもつ証明書をフィッシングサイトに使用され得るものとして検出する．なお，エントロピーを計算する際に，DDNS やホスティングサービスのドメイン名が含まれていた場合にはその部分を削除して計算を行う．それらのサービスが提供するドメイン名は必ずしもフィッシングサイトに限定したものではなく，汎用的であるためである．

上記のテンプレート抽出処理の例を次に示す．あるクラスタに 2 つの CN，`apple-accountverify123[.TLD]`[*2] と `payment-accountverify55[.TLD]` が存在したものとする．2 つの文字列の共通部分文字列は `-accountverify` である．それぞれの CN の正規表現は，
`[a-z]{5}-accountverify[0-9]{3}[.TLD]` および
`[a-z]{7}-accountverify[0-9]{2}[.TLD]` となり，これらをマージすると
`[a-z]{5,7}-accountverify[0-9]{2,3}[.TLD]` を得る．次に，この正規表現のエントロピー削減量を求める．文字列 `-accountverify` を構成する文字数は 14，`[a-z]` と `[0-9]` の正規表現部分では文字数の平均値 L_a と L_d がそれぞれ 6 と 2.5 であるため，全体の文字列長 L はそれらの和である 22.5 となる．したがって式 1，2 より，$B_e(u) = 36.5$, $B(u) = 116.3$, $d(e) = 79.8$ を得る．エントロピー削減量が閾値 55 を上回っているため，この正規表現をテンプレートとして採用する．このテンプレートを用いた場合，`google-accountverify37[.TLD]` という CN はテンプレートの正規表現とマッチするためフィッシングとして検知される．一方で，`security-accountverify9[.TLD]` は同じ共通部分文字列を含んでいるが，正規表現の文字数がマッチしていないため検知されない．以上のようにテンプレート抽出およびフィッシングサイトの探索を行う．

4. データ

フィッシングサイトに使用された証明書を収集するため，本研究では OpenPhish [19] というデータベースを利用して 2018 年 10 月から 2019 年 1 月の期間に発見されたフィッシングサイトの URL を収集した．それぞれの収集した URL に対応する，同期間に発行された証明書を Censys [3] にて探索し，2,638 のユニークな証明書を取得した．その後 3 章で述べたデータ処理を加えた結果，合計で 1,634 のフィッシングサイトに使用された証明書を抽出した．

表 2 に抽出した証明書の数を示す．表に示す通り，多くの証明書の CN に含まれるドット数は少なく，ドット数 m が 2 以下の証明書が全体の 87%を占める．一方で，ドット数が多い CN をもつ証明書も一定数存在しており，5%以上の証明書はドット数 m が 4 以上である．すなわち，類似した CN 同士をもつ証明書を発見する際には，各ドット数 m において慎重に閾値設定を行う必要がある．

取得した証明書の発行者を調査した結果，非常に多くの証明書は 2 つの無料で利用可能な認証局から発行されていた．1,634 の収集した証明書の内，852（50.9%）は Let's Encrypt，714（42.7%）は Comodo [4] によって運営される認証局である cPanel から発行されていた．加えて，自己署名証明書が占める割合は 1%にも満たなかった．したがって，これら 2 つの認証局から発行された証明書を CT ログから探索することにより，効率的にフィッシングサイトを発見することができる．以降の章では，これら 2 つの認証局から発行された証明書を収集・分析する．表 3 に収集した証明書の統計を示す，2 つの認証局から合計約 3,870 万の証明書を収集した．

表 2 各 m の値におけるフィッシングサイトに使用された証明書数

$m=1$	$m=2$	$m=3$	$m=4$	$m\geq 5$	合計
956	468	110	70	30	1,634

表 3 収集した，2 つの無料発行を行う認証局から発行された証明書数

Let's Encrypt	cPanel
2,1230,624 (54.9%)	1,7438,554 (45.1%)

5. 証明書の分析結果

本章では，初めに証明書の CN に対するクラスタリング結果およびテンプレート抽出結果を示す．次に抽出したテンプレートを用いて未知のフィッシングサイトを探索した結果について述べる．最後に，HTTPS 化したフィッシングサイトを生成するエコシステムの詳細が判明した例をケーススタディとして示す．

5.1 クラスタリング

DBSCAN アルゴリズムによるクラスタリングをフィッシングサイトに使用された CN に適用した結果，106 のクラスタを取得した．これらのクラスタには本研究での調査対象である 1,634 の証明書の内，341（20.8%）の証明書が含まれていた．次に，各 m の値におけるクラスタリング

[*2] 本論文では悪性サイトの URL を伏せるため，TLD の部分を [.TLD] と置換する．

表 4　クラスタリング結果

	$m=1$	$m=2$	$m=3$	$m=4$	$m\geq 5$	合計
クラスタ	47	39	8	7	5	106
証明書	124	122	49	33	13	341

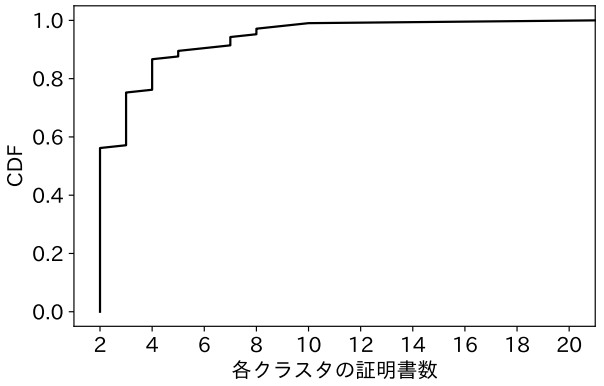

図 4　各クラスタの証明書数を表す累積分布 (CDF)

結果を表 4 に示す．表に示す通り，多くのクラスタおよび証明書は 2 以下の少ないドット数に偏っている．これは，4 章で述べたように対象データには少ないドット数の証明書が多く存在するためである．図 4 に各クラスタに含まれる証明書数の累積分布を示す．全体として，クラスタに含まれる証明書の数は少ないがいくつかのクラスタには比較的多くの証明書が含まれていることが分かる．

5.2　テンプレート抽出

3 章で述べた手法により本研究で設定した閾値を満たすテンプレートを 106 のクラスタの内 69（65.1%）抽出した．表 5 に，前節で取得したクラスタおよび抽出したテンプレートの例を示す．表中のフィッシングサイトに使用された CN のドメインには `serveirc[.TLD]` や `hoster-test[.TLD]` といった DDNS やホスティングサービスが提供するドメインが含まれているため，これらのフィッシングサイトを生成した攻撃者はそれらのサービスを利用していたことが分かる．このようなドメインが含まれるテンプレートは，抽出したテンプレートの中に 10（14.5%）含まれていた．次節ではこのテンプレートを用いて未知のフィッシングサイトの探索を行った結果を述べる．

5.3　フィッシングサイト探索結果

前節にて抽出したテンプレートを使用し，その正規表現とマッチする CN をもつ証明書を探索する．Let's Encrypt および cPanel から発行された 3,870 万の証明書を対象データとして探索を行った結果，1,650 のフィッシングに使用される可能性がある証明書を新たに発見した．また，抽出した 69 のテンプレートの内 1 つ以上の証明書を発見したものは 46（66.7%）存在した．これより，多くのテンプレートは新たなフィッシング攻撃を事前に捉えることに有用であるといえる．図 5 に各テンプレートで発見した証明書の数を示す累積補分布を示す．図より，多くのテンプレートではマッチする証明書の数は少ないが，無視できない数のテンプレートでは非常に多くの証明書を発見していることが分かる．特に，上位 2 つのテンプレートでは 924，395 の証明書を発見していた．そのような一つのテンプレートにマッチする CN をもつ大量の証明書は，攻撃者がフィッシングサイトを作成する際に自動で生成されている可能性が高い．

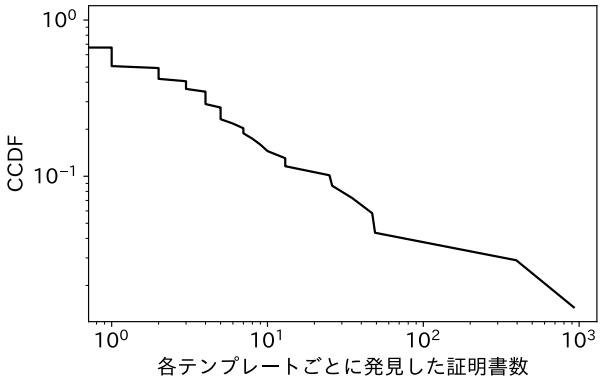

図 5　テンプレートで検知した証明書数の累積補分布 (CCDF)

395 の証明書を捉えたテンプレートのドメインはあるホスティングサービスが提供するドメイン（`zap-webspace[.TLD]`）であった．さらに，他の DDNS およびホスティングサービスが提供するドメインを含むテンプレートによって合計で 15 の証明書が発見された．したがって，本手法は DDNS またはホスティングを利用した攻撃であっても発見可能であることが分かる．次節では最も多い 924 の証明書を捉えたテンプレートに焦点を当てる．

5.4　ケーススタディ

本節ではケーススタディとして，証明書の解析によりフィッシングサイトのエコシステムの詳細が明らかになった事例を示す．最も多くの証明書を発見したテンプレートは `[a-z]{6,8}.runescape.com-[a-z]{1,8}[.TLD]` であったが，取得した CN は `secure.runescape.com-[a-z]{1,8}[.TLD]` または `services.runescape.com-[a-z]{1,8}[.TLD]` のいずれかのパターンであった．これらは世界的に人気が高いオンライン RPG である *Runescape* のユーザを標的としている．表 6 に *Runescape* を標的としたクラスタにおける証明書の CN を例示する．このテンプレートが発見した 924 の証明書の 93.8%は同一の認証局，Let's Encrypt から発行されていることが分かった．さらに，これらの CN の多くはセカンドレベルドメインと TLD の組み合わせが異なっていた．発見した証明書を詳細に調査すると，これらの証明書はあるアンダーグラウンド企業が販売するフィッシン

表 5　抽出したクラスタの CN およびテンプレートの例

	クラスタ（$m = 2$）	クラスタ（$m = 4$）
CN	verify-webapps25476.serveirc[.TLD] verify-webapps72647.serveirc[.TLD] verify-webscrid2678.serveirc[.TLD]	onedrive.liveviewuserauthaspx209hr28jh.srv156794.hoster-test[.TLD] onedrive.liveviewuserauthaspx209hr28jh.srv156816.hoster-test[.TLD] onedrive.liveviewuserauthaspx209hr28jh.srv156797.hoster-test[.TLD] onedrive.liveviewuserauthaspx209hr28jh.srv156796.hoster-test[.TLD]
テンプレート	verify-web[a-z0-9]{4,5}.serveirc[.TLD]	onedrive.liveviewuserauthaspx209hr28jh.srv156[0-9]{3,3}.hoster-test[.TLD]

表 6　*Runescape* を標的としたクラスタの CN 例

secure.runescape.com-mq[.TLD]
services.runescape.com-an[.TLD]
secure.runescape.com-g[.TLD]
secure.runescape.com-l[.TLD]

表 7　*Runescape* を標的としたフィッシングサイト生成サービスが使用していた証明書の CN 例

services.runescape.com-rv[.TLD]
services.runescape.com-vo[.TLD]
services.runescape.com-rs[.TLD]
secure.runescape.com-rs[.TLD]
secure.runescape.com-ao[.TLD]
secure.runescape.com-vo[.TLD]

グサイト生成サービスにより発行された可能性が非常に高いことが分かった．Web 上で検索することでその企業名は判明するが，研究倫理の観点からその実名は本論文では伏せる．その企業が提供するサービスの詳細な調査により，表 7 に示すような CN の証明書を生成していることが確認された．また，これらの CN は本研究で生成したテンプレートにマッチする．

当該フィッシングサイト生成サービスには，攻撃者が効率的にフィッシングを行うための多くの拡張機能— 自身でカスタマイズして大量のスパムメールを送信できる大量スパムメール送信機能，通知機能，ロギング，アナリティクス，奪取したアカウントの売買を行えるマーケット機能など—が備わっていた．フィッシングサイト生成サービスの存在を指摘する先行研究 [18, 20] では，そうしたサービスのエコシステムに対する詳細な分析は行っていない．以上の結果は，本研究の提案手法がフィッシング攻撃のエコシステム解明に有用であることを示している．

6. 議論

6.1 本研究の制約事項

本研究ではテンプレート抽出によりフィッシングサイトと紐づいていると考えられる証明書を発見したが，フィッシングサイトの生存期間は非常に短いため，その証明書を使用する Web サイトへのアクセスを行うことができなかった．そのため，*Runescape* を標的とした一部のフィッシングサイトを除いて，発見した Web サイトが本当にフィッシング行為を行っていたかは検証できていない．本研究で提案した手法を用いてフィッシングサイトをリアルタイムに検知し，実際にフィッシングサイトであるかを確認・遮断するシステムの開発は今後の課題である．

攻撃者がワイルドカード証明書を使用していた場合には，提案手法による検知は行えない．しかしながら，ワイルドカード証明書を用いる場合，攻撃者が使用するホストの内 1 つでも検知されると，他のホストも芋づる式に検知される可能性が高まる．つまり，ワイルドカード証明書を利用することは攻撃者にとって全体としての検知率を高めるリスクが存在する．攻撃検知回避の観点では，ワイルドカード証明書を使わず，各ドメイン名に証明書を発行するのが有利な戦略である．実際，本研究で発見したフィッシングキットはそのような運用を実施していた．

6.2 認証局に対する提言

無料で利用できる証明書を用いたフィッシングサイトの増加に伴い，認証局による対策を行う重要性が高まっている．cPanel の運営を行っている Comodo 社は証明書がフィッシングなどの非合法な目的に使用されたことが発覚した場合，その証明書を 7 日以内に失効すると Certification Practice Statement (CPS) に明記している．一方，Let's Encrypt は証明書を発行する際に Google Safe Browsing によって発行先の Web サイトが悪性であるかの確認を行っていたが，その取り組みを 2019 年 1 月に終了させた [14]．その理由として Let's Encrypt は，ドメイン認証の証明書はクライアント・サーバ間の通信を安全に保つことを目的としており，Web サイトの安全性を保証するものではないからであると主張している．しかし，5 章で述べたように *Runescape* を標的としたフィッシングキットが生成していたとされる証明書の 93.8%は Let's Encrypt から発行されていた事実を踏まえると，認証局は明らかに悪性な活動に利用される Web サイトの証明書を特定し，悪用される前に失効させる，もしくは登録者に対して何らかのペナルティやコストを課すことが望ましい．そのような悪性サイトを検出する上で，本研究の提案手法は有効である．

7. 関連研究

HTTPS 化したフィッシングサイトの増加と共に，証明書を用いてその検知を試みる研究が行われ始めている [6, 7, 26]．Torroledo [26] および Dong [6] らは証明書のフィールドの特徴を基に悪性サイト検知を行う手法を提案した．しかし，Drury らが 2019 年に発表した研究 [7] で

は，同一の認証局から発行された全ての証明書ではいくつかのフィールドは全く同じ，または非常に類似していることを指摘している．そのため，同一の認証局から発行されていた場合，その証明書の情報のみを用いて対象の Web サイトが正規サイトかフィッシングサイトであるかを判断することは非常に困難であると結論付けた．本研究は先行研究のアプローチと異なり，クラスタ分析を適用することで攻撃者が残した「痕跡」，すなわちテンプレートを抽出し，未知のフィッシングサイトの発見およびフィッシングサイトを生成するエコシステムの詳細な理解を実現した．

8. まとめ

公開鍵証明書の情報を基に HTTPS 化したフィッシングサイトを発見する手法を提案した．フィッシングサイトが使用した証明書に対してクラスタリング分析を適用し，クラスタから抽出したテンプレートを用いることで，未知のフィッシングサイトを発見できることを示した．また，発見した証明書に対して詳細な調査を行うことにより，フィッシングサイトを生成するエコシステムを詳細に理解する手がかりを得ることができる．公開鍵証明書からフィッシングサイトを発見するアプローチが有効であるのは，DDNS やホスティングサービス，無料の公開鍵証明書発行サービスを利用する攻撃者が存在する事実に起因する．本研究の成果が，公開鍵証明書を活用したフィッシング対策に貢献することを期待する．

参考文献

[1] APWG: Phishing Activity Trends Report 1th Quarter 2019, https://docs.apwg.org/reports/apwg_trends_report_q1_2019.pdf.

[2] Blum, A., Wardman, B., Solorio, T. and Warner, G.: Lexical Feature Based Phishing URL Detection Using Online Learning, *Proc. of ACM AISec*, pp. 54–60 (2010).

[3] Censys: https://censys.io/.

[4] COMODO: https://ssl.comodo.com/.

[5] cPanel: https://cpanel.net/.

[6] Dong, Z., Kapadia, A., Blythe, J. and Camp, L. J.: Beyond the Lock Icon: Real-time Detection of Phishing Websites Using Public Key Certificates, *Proc. of APWG Symposium eCrime* (2015).

[7] Drury, V. and Meyer, U.: Certified Phishing: Taking a Look at Public Key Certificates of Phishing Websites, *Proc. of USENIX SOUPS* (2019).

[8] Felt, A. P., Barnes, R., King, A., Palmer, C., Bentzel, C. and Tabriz, P.: Measuring HTTPS Adoption on the Web, *USENIX Security*, pp. 1323–1338 (2017).

[9] Google: HTTPS encryption on the web, https://transparencyreport.google.com/https/overview?hl=en.

[10] Google: A secure web is here to stay, https://security.googleblog.com/2018/02/a-secure-web-is-here-to-stay.html.

[11] Google: Webmaster Central Blog, HTTPS as a ranking signal, https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html.

[12] Internet Crime Complaint Center: 2018 Internet Crime Report, https://pdf.ic3.gov/2018_IC3Report.pdf.

[13] Let's Encrypt: https://letsencrypt.org/.

[14] Let' s Encrypt : Let' s Encrypt No Longer Checking Google Safe Browsing, https://community.letsencrypt.org/t/let-s-encrypt-no-longer-checking-google-safe-browsing/82168.

[15] Mozilla: Communicating the Dangers of Non-Secure HTTP, https://blog.mozilla.org/security/2017/01/20/communicating-the-dangers-of-non-secure-http/.

[16] Naylor, D., Finamore, A., Leontiadis, I., Grunenberger, Y., Mellia, M., Munafò, M., Papagiannaki, K. and Steenkiste, P.: The Cost of the "S" in HTTPS, *Proc. ACM*, CoNEXT, pp. 133–140 (2014).

[17] Nguyen, L. A. T., To, B. L., Nguyen, H. K. and Nguyen, M. H.: A novel approach for phishing detection using URL-based heuristic, *Proc. of IEEE ComManTel*, pp. 298–303 (2014).

[18] Oest, A., Safei, Y., Doup, A., Ahn, G., Wardman, B. and Warner, G.: Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis, *APWG Symposium eCrime*, pp. 1–12 (2018).

[19] OpenPhish: OpenPhish FAQ, https://openphish.com/faq.html.

[20] Peng, P., Xu, C., Quinn, L., Hu, H., Viswanath, B. and Wang, G.: What Happens After You Leak Your Password: Understanding Credential Sharing on Phishing Sites, *Proc. of ACM*, Asia CCS, pp. 181–192 (2019).

[21] PHISHLABS: 2019 PHISHING TRENDS AND INTELLIGENCE REPORT The Growing Social Engineering Threat, https://info.phishlabs.com/hubfs/2019PTIReport/2019PhishingTrendsandIntelligenceReport.pdf.

[22] Qualys: SSL Server Test, https://www.ssllabs.com/ssltest/.

[23] Ratcliff, J. W. and Metzener, D. E.: Pattern-matching-the gestalt approach, *Dr Dobbs Journal*, Vol. 13, No. 7, p. 46 (1988).

[24] Shirazi, H., Bezawada, B. and Ray, I.: "Kn0W Thy Doma1N Name": Unbiased Phishing Detection Using Domain Name Based Features, *Proc. of ACM SACMAT*, pp. 69–75 (2018).

[25] The Chromium Project: Chromium Certificate Transparency Policy, https://github.com/chromium/ct-policy.

[26] Torroledo, I., Camacho, L. D. and Bahnsen, A. C.: Hunting Malicious TLS Certificates with Deep Neural Networks, *Proc. of ACM AIsec* (2018).

[27] van der Heijden, A. and Allodi, L.: Cognitive Triaging of Phishing Attacks, *USENIX Security*, pp. 1309–1326 (2019).

[28] Xie, Y., Yu, F., Achan, K., Panigrahy, R., Hulten, G. and Osipkov, I.: Spamming botnets: Signatures and characteristics, *ACM SIGCOMM CCR*, pp. 171–182 (2008).

[29] Zhang, Y., Hong, J. I. and Cranor, L. F.: Cantina: A Content-based Approach to Detecting Phishing Web Sites, *Proc. of the 16th International Conference WWW* (2007).