IIR

Internet
Infrastructure
Review

Jun.2018
Vol.39

定期観測レポート

メッセージングテクノロジーなりすましメール対策としてのDMARCの普及

フォーカス・リサーチ(1)

RSAアルゴリズム鍵生成モジュールの実装問題(ROCA)

フォーカス・リサーチ(2)

大規模メールシステムにおけるメールの不正送信対策について

IIJ
Internet Initiative Japan

Internet Infrastructure Review

June 2018 Vol.39

《 読者アンケート 協力のお願い 》

今号をお読みいただいたご意見・ご感想をお聞かせください。今後の参考にさせていただきます。
回答いただいた方の中から抽選で30名様にインターネット便利帳付きIIJオリジナルリングノートをプレゼントいたします。
※なお、当選のお知らせは、プレゼントの発送をもってかえさせていただきます。

【 アンケート受付期間 】
～2018年8月31日(金)まで

【 回答方法 】
IIJのWebサイト内、以下のページまたは右側のQRコード(https://www.iij.ad.jp/dev/report/iir/039.html)から
ご回答ください。

エグゼクティブサマリ

我が国では、通信の秘密は憲法で保障されており、IIJのような電気通信事業者、ならびに、それに従事する者は、電気通信事業法において、取り扱う通信の秘密を侵してはならないとされています。一方、電気通信事業を営むうえでは、課金のために顧客の通信履歴を参照したり、パケットを宛先に届けるためにヘッダ情報を参照するなど、通信の秘密を侵して事業を行っています。これらに加えて、迷惑メール対策のフィルタリングやOP25B、児童ポルノブロッキングなど、インターネットを安心してご利用いただくために、私たちISPが常日頃から行っている行為のなかには、通信の秘密に抵触するものもありますが、それらはお客様の同意をいただいていたり、刑法における違法性阻却の枠組みに沿って慎重に整理がなされています。

そのようななか、4月に政府の知的財産戦略本部が決定した「インターネット上の海賊版サイトに対する緊急対策」において、法制度整備が行われるまでの臨時的かつ緊急的な措置として、特に悪質な海賊版サイトをISPがブロッキングするのは適当であると提言されたことは、私たちISPにとって大きな驚きであり、法律家や消費者団体などからは強い懸念が表明されました。今後、タスクフォースを立ち上げて議論されることになっていますので、注視していきたいと思います。

IIRは、IIJで研究・開発している技術の幅広い紹介を目指しており、日々のサービス運用から得られる各種データをまとめた「定期観測レポート」と、特定テーマの掘り下げた「フォーカス・リサーチ」から構成されています。今回は、1章の定期観測レポート、2章のフォーカス・リサーチ(1)ではROCAと呼ばれるRSA暗号アルゴリズムの鍵生成モジュール、3章のフォーカス・リサーチ(2)では電子メールを取り上げています。

1章の前半では、IIJのメールサービスで検知した迷惑メールの受信メールの推移を、2008年から2017年までの500週分について報告すると共に、2017年の特徴的な動きを解説しています。後半では、迷惑メール対策に有効な送信ドメイン認証技術DMARCの普及状況や標準化の動向、更には日本で導入する場合に重要となる法的な取り扱いに関して解説しています。

2章のフォーカス・リサーチ(1)においては、ROCAと呼ばれるRSA暗号アルゴリズムの鍵生成モジュールの実装不備に絡み、プログラムもしくは設計の不具合により、暗号が想定していた安全性を確保できていない事例や、そのような脆弱性を引き起こす要因の分析、更には研究結果が及ぼす今後の影響について報告しています。

3章では、IIJが提供している数百万規模のサービスプロバイダ向け大規模メールシステムの構築や、その運用で蓄積した不正利用対策について紹介しています。メールシステム管理者の業務の多くは、メールシステムの不正利用に起因した対応やそれに付随する障害対応です。そのため、メールシステムに不正利用対策を確実に実装することは、運用管理の負荷軽減や、エンドユーザへの安定的なサービス提供に大きく寄与します。

IIJでは、このような活動を通じて、インターネットの安定性を維持しながら、日々改善・発展させていく努力を続けております。今後も、皆様の企業活動のインフラとして最大限にご活用いただけるよう、様々なサービス、ソリューションを提供してまいります。

島上 純一（しまがみ じゅんいち）
IIJ 取締役 CTO。インターネットに魅かれて、1996年9月にIIJ入社。IIJが主導したアジア域内ネットワークA-BoneやIIJのバックボーンネットワークの設計、構築に従事した後、IIJのネットワークサービスを統括。2015年よりCTOとしてネットワーク、クラウド、セキュリティなど技術全般を統括。2017年4月にテレコムサービス協会MVNO委員会の委員長に就任。

メッセージングテクノロジー
なりすましメール対策としてのDMARCの普及

1.1 はじめに

ここでは、迷惑メールを中心としたメールの動向と、迷惑メール対策に関係する技術動向について報告します。

2008年の創刊以来、迷惑メール量の変化を示す指標として、IIJのメールサービスで検知した迷惑メールの受信メールに対する割合の推移を報告してきましたが、本年度からのメールシステムのリニューアルに伴い、これまでの形式での報告は今回が最後となります。今後は、大きな変化や動きなどがあった場合に、別の形で報告します。

技術動向は、引き続き送信ドメイン認証技術の解説や普及状況を報告します。また、送信ドメイン認証技術DMARCの導入に関して、昨年、法的な整理がなされましたので、その概要についても報告します。

1.2 迷惑メールの動向

迷惑メールの動向を示す指標として、ここではIIJのメールサービスが提供する迷惑メールフィルタで検知した迷惑メールの割合の推移を報告します。今回は、これまでの調査結果全体について、2008年の第23週（2008年6月2日からの1週間）から2017年の第52週（2017年12月25日からの1週間）までの期間、ちょうど500週分の推移となります（図-1）。

2017年の迷惑メールの割合の平均は30.5%でした。2016年の平均が39.9%でしたので、9.4%減少したことになりますが、2015年は平均24.7%でしたので、単純に減っているわけではなさそうです。実際に迷惑メールの中には、引き続き大手企業を詐称したフィッシングメールが現在も多く存在しますし、最近ではランサムウェアの実行に導くような悪質な迷惑メールも増えているようです。

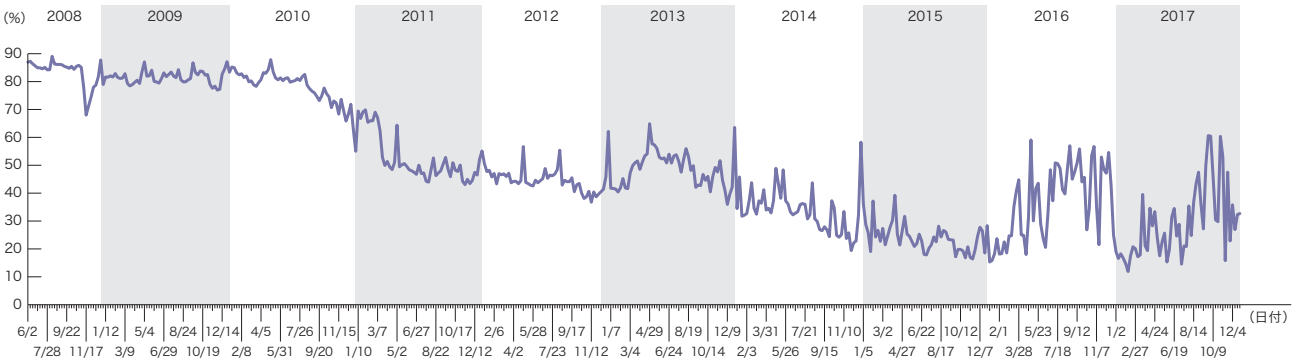


図-1 迷惑メール割合の推移

1.2.1 フィッシングメールの認証結果

前回(Vol.35)は、マイクロソフト社を詐称する迷惑メールについて、その送信ドメインの認証結果を報告しました。その後も大手企業を詐称するメールが続いています。

フィッシング対策協議会では、フィッシングメールの情報を公開[*1]していますので、届いたメールに示されたURLやHTMLファイルを開く前に、最近流通しているフィッシングメールかどうかを確認することが大切です。しかし、送信ドメイン認証技術を利用することで、もっと簡単にフィッシングメールを見破る方法があります。大手企業の多くは、既にDMARCを導入していますので、DMARC認証をすることで送信者情報が詐称されているかを判断できます。

Apple社を例に取ると、iCloudへのログインに関するメールに対しては、SPF、DKIM、DMARCすべてに対応したメールを送信します。最近送信されているフィッシングメール(図-2)では、メールヘッダ上の送信者情報(RFC5322.From)に本物と同じドメイン名email.apple.comを利用します。当然のことながら、送信元が違いますのでSPFは失敗(softfail)しますし、DKIMの署名はない(none)ので、DMARCの認証は失敗(fail)することになります。しかも、email.apple.comのDMARCレコードは、ポリシーがreject(p=reject)ですので、DMARCの仕様に沿った受信判断をする場合、こうしたメールは届かないことになります。

「楽天市場」や「楽天カード」を詐称したメールも依然として数多く送信されています。同様にこれらもヘッダ上の送信者情報にrakuten.co.jpやmail.rakuten-card.co.jpドメイン名を利用していますが、いずれのドメイン名もDMARCレコードを設定しています。そのため、すべてDMARC認証が失敗していますので、簡単に詐称メールと見破ることができます。送信ドメインのDMARC導入が進めば、こうした不要なメールを排除できるようになります。また、詐称されやすいドメイン名を管理している場合は、詐称対策としてのDMARCレコードの設定が望まれます。


お客様のApple ID が、ウェブブラウザからiCloudへのサインインに使用されました。

日付と時間：2018年3月8日 22:08 JST

のブラウザ： Chrome

オペレーティングシステム： Windows

IP：[redacted]

上記が問題でない場合は、このメールを無視してください。

最近iCloudへサインインを行ったことがなく、他者が違法にお客様のアカウントを使用していると考えられる場合は、Apple IDでパスワードをリセットしてください。

今後ともよろしくお願い致します。

Apple サポートセンター

Apple ID | サポート | プライバシーポリシー
Copyright 2018
Apple Distribution International, Hollyhill Industrial Estate, Hollyhill, Cork, Ireland.
すべての権利を保有しております。

図-2 Apple社を詐称するメール

＊1　フィッシング協議会：フィッシングに関するニュース(http://www.antiphishing.jp/news/alert/)。

1.2.2 メールの新たな脅威

米国FBIのIC3（Internet Crime Complaint Center）は、2017年のInternet Crime Reportを発表[*2]しました。その中の2017年のトピックスとして挙げられているものに、BEC[*3]とランサムウェア（Ransomware）があります。

BECは、巧妙な手口でメール受信者を騙して不正送金させる詐欺行為の1つです。IC3では、2017年に15,690件の苦情を受け、6.75億ドル以上の損失が発生したと報告しています。

日本でも大手航空会社が2017年9月に取引先を装った送金先変更のメールに騙され、3億円以上の被害が発生したことを同年の12月に公表し、大きなニュースとなりました。もちろん、多くのメール利用者は、こうした詐欺メールに騙されるわけがないと考えているかもしれません。しかしながら、実際には多くの被害が発生していますし、報道などによればかなり巧妙な手口を使って用意周到に準備した上で実行しているようです。被害に遭わないためにも、まずは技術的な対策をしっかり導入することが必要でしょう。

日本でも2017年5月に大きなニュースとなったWannaCryはランサムウェアであり、広義には不正プログラム（マルウェア）の一種となります。ランサムウェアに感染すると、重要なファイルが暗号化され、解読する鍵を得るために仮想通貨などでの支払いを要求されます。感染経路は様々ですが、そのひとつに標的型攻撃も含まれますので、メールの対策も重要となります。これまでのマルウェアのビジネスモデル（機密情報などを入手し別途闇市場などで金銭を得る）と異なり、被害者から直接金銭を得る手法であること、送金手法として仮想通貨を要求することで受け取り手の実態を掴めなくする、といった手法が新しいと言えます。

IC3では、2017年にランサムウェアと認識できた苦情が1,783件、被害額が230万ドル以上と報告しています。今年に入ってからも、3月に米国アトランタ市でランサムウェアの攻撃を受け、大きな被害が発生しているようです[*4]。

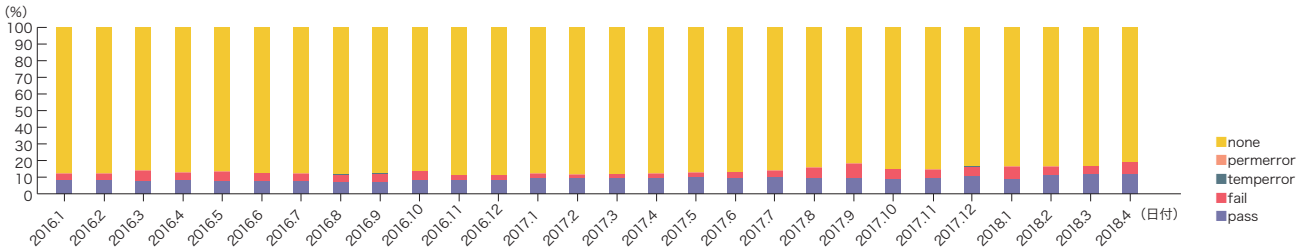


図-3 DMARC認証結果の推移

＊2　FBI、"Latest Internet Crime Report Released"（https://www.fbi.gov/news/stories/2017-internet-crime-report-released-050718）。
＊3　BEC：Business Email Compromise。
＊4　The City of Atlanta、"Ransomware Cyberattack Information"（https://www.atlantaga.gov/government/ransomware-cyberattack-information）。

1.3 メールの技術動向

ここでは、送信ドメイン認証技術DMARCの普及状況や関連技術も併せた標準化動向、日本で導入する場合に重要となる法的な取り扱いについて報告します。

1.3.1 DMARCの普及状況

IIJのメールサービスでは、受信したメールに対してDMARC認証を実施しています。DMARCの認証結果の2018年4月までの毎月の平均の推移を図-3に示します。

最新の調査結果である2018年4月の受信メールでは、DMARC認証できたメールの割合が、これまでの調査で最も高い割合の19.3%となりました。認証結果がpassであった割合も最も高く、12.2%という結果でした。まだDMARCが普及しているとは言えないレベルですが、少しずつDMARCを導入するドメイン名が増えています。

次に、SPF、DKIMを含めた送信ドメイン認証技術の認証結果のうち、2018年4月の組み合わせを図-4に示します。図-4の"DMARC+SPF+DKIM"の項目(8.8%)は、DMARCとSPFとDKIMすべての認証がpassした割合を示しています。つまり、DMARC認証できたドメイン名で最も多い組み合わせは、SPFもDKIMも導入しているドメインであることが分かりました。これは前回の調査結果(Vol.35)と同じ組み合わせでした。認証の組み合わせで最も割合が高いのは、"SPF"単体(35.4%)でした。他との組み合わせを含めて"SPF"でpassした割合の合計は69.3%となり、やはり導入の容易さが普及の要因であることが推測できます。総務省の最新の取りまとめデータ[*5]の2018年3月では、passの割合が90%を超えていました。

逆に"!(...)"の項目は、passした認証技術が1つもなく、括弧内に示された認証技術の組み合わせが失敗した割合を示しています。図-4からは、SPF単体で認証に失敗した割合"!(SPF)"が最も多く、6.5%であったことが分かります。送信ドメイン名を詐称している可能性もありますが、SPFが正しく認証できない利用例であるメール転送されて受信した割合も少なからず含まれているのではと推測しています。

次に、DMARC認証できたドメイン名のTLD(Top LevelDomain)別の割合を図-5に示します。もっとも数が多かったTLDは、com

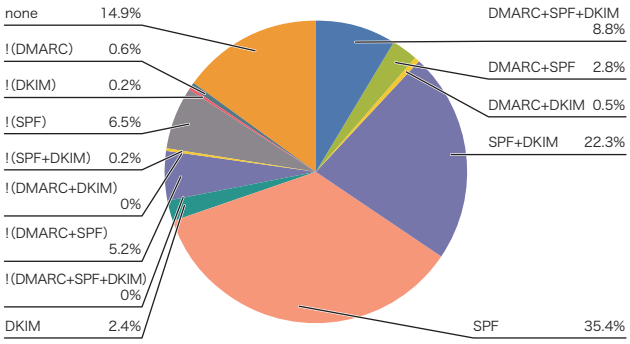


図-4 送信ドメイン認証結果の組み合わせ(2018年4月)

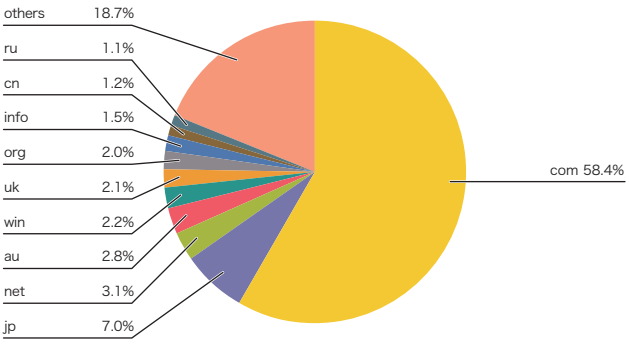


図-5 DMARC認証できたドメイン名のTLD別の割合

*5 総務省:統計データ(http://www.soumu.go.jp/main_sosiki/joho_tsusin/d_syohi/m_mail.html#toukei)。

ドメイン(58.4%)でした。次がjpドメイン名(7.0%)でしたが、comドメインとの差はかなり大きい結果となりました。国の行政機関レベルで導入を働きかけている豪州[*6](au、4位、2.8%)や英国[*7](uk、6位、2.1%)も、日本での受信メールのことを考えれば、かなり高い割合であると言えます。

流量ベースでは、comが53.6%、jpが43.4%となり、これら2つのTLDで、DMARCドメイン名の大部分を占める結果となりました。

豪州と英国の行政機関へDMARC導入を働きかけていると述べましたが、米国も国土安全保障省(DHS)が連邦機関に対して、電子メールとウェブのセキュリティ強化を決定しました(BOD 18-01)[*8]。この決定では、90日以内にDMARCレコードを少なくとも"p=none"のポリシーで設定することを求めています。更に1年以内に"p=reject"と設定しなければなりません。既に報告しているとおり、"p=reject"とDMARCレコードのポリシーを設定している場合、DMARC認証が失敗したときにメールの受信を拒否される可能性が高くなります。つまり、"p=reject"と宣言するためには、SPF、DKIMの設定を含めて、正規のメールがDMARC認証で失敗しないようにメールシステムをきちんと管理していく必要があります。その意味でも米国DHSは大変重要な決定をしたと言えます。日本の政府や自治体なども検討していただくことを希望しています。

1.3.2 jpドメインの導入状況

2005年4月から2012年5月まで、WIDEプロジェクトはjpドメイン名を管理する日本レジストリサービス(JPRS)と共同研究契約を結び、jpドメイン名でのSPFなどの普及率を計測してきました[*9]。この期間は、ちょうどSPFの普及時期と重なり、その変化の度合いやその効果を類推する上で、貴重なデータとなりました。

今回、DMARC普及を進めて行くにあたり、総務省はこの調査をDMARCも含めて改めて開始することにしました[*10]。具体的な方法としては、総務省の業務委託先である(一財)日本データ通信協会がJPRSと共同研究契約を結ぶことになりました。この調査に関しては、日本データ通信協会の客員研究員の立場で筆者も参加しています。

総務省が発表した2018年1月時点での調査結果[*5](表-1)では、メールに利用するドメイン名のSPF設定割合は全体で56.9%でした。WIDEプロジェクトで調査していたSPFの普

属性	ドメイン数	MX設定数	SPF設定数	SPF設定率(%)	DMARC設定数	DMARC設定率(%)
ad	252	212	140	66.0	6	2.8
ac	3596	3367	2086	62.0	10	0.3
co	403955	380239	252961	66.5	1089	0.3
go	582	428	395	92.3	1	0.2
or	35146	33012	21043	63.7	71	0.2
ne	13044	10617	5590	52.7	99	0.9
gr	6112	5438	2884	53.0	27	0.5
ed	5230	4852	2854	58.8	21	0.4
lg	1652	1216	921	75.7	2	0.2
地域・都道府県型	13414	7530	3959	52.6	28	0.4
汎用	988365	756800	391728	51.8	5565	0.7
合計	1471349	1203711	684561	56.9	6919	0.6

表-1 jpドメインの送信ドメイン認証技術設定調査結果

＊6 DMARC、"Australian Government Agency Recommends DMARC, DKIM, and SPF"(https://dmarc.org/2016/08/australian-government-agency-recommends-dmarc-dkim-and-spf/)。
＊7 DMARC、"DMARC Required For UK Government Services By October 1st"(https://dmarc.org/2016/06/dmarc-required-for-uk-government-services-by-october-1st/)。
＊8 DHS、"Binding Operational Directive 18-01"(https://cyber.dhs.gov/bod/18-01/)。
＊9 WIDEプロジェクト、「ドメイン認証の普及率に対する測定結果」(http://member.wide.ad.jp/wg/antispam/stats/index.html.ja)。
＊10 総務省、「JPドメイン名における送信ドメイン認証技術の設定状況の調査」(http://www.soumu.go.jp./menu_news/s-news/01kiban18_01000035.html)。

及率は、MXレコードを設定したドメイン数に対する、全体でのSPFレコード設定数の割合でしたので、上記の普及率とは少し計算方法が異なります。同じ条件では、2018年1月時点で58.1%となります。WIDEプロジェクトでの2012年5月時点での普及率は43.89%でしたので、同じ条件では約14.2%増加したことになります。

jpドメイン名に対するDMARCレコードの設定調査は、今回が初めての試みとなります。IIJのメールサービスにおける受信メール比(流量比)での普及率は19.3%でしたが、実際のjpドメインでの設定割合は、残念ながら全体の平均でわずか0.6%でした。WIDEプロジェクトによる最初のSPFの調査結果が0.1%でしたので、今後の伸びに期待したいところです。また、英国や豪州、米国の政府機関での取り組みを紹介しましたが、日本政府機関での普及も期待したいところです。実際、SPFについては、政府機関が利用するgo.jpドメインで92.3%と高い普及率となっています。地方自治体でよく利用されるlg.jpドメインについても75.7%と、属性型別で2番目に高い普及率となっています。

DMARCレコードの設定は、メールサーバの出口を確認しなければならないSPFレコードの設定より更に簡単ですので、既にSPFレコードを設定できているのであれば、まず"p=none"ポリシーのDMARCレコードから設定するべきと考えています。

1.3.3 関連技術も含めた標準化動向

DMARCなどのインターネット上のいわゆる技術標準は、IETF(Internet Engineering Task Force)[*11]でRFC(Requestfor Comments)として文書で公開されます。IETFでは、議論の対象分野ごとにWG(Working Group)が作られ、WG内で技術仕様などを議論し、最終的にRFCが発行されます。今回、2018年3月に開催されたIETF 101 meetingに参加したので、最近のIETFやメール関連の状況について報告します。

IETF meetingは年3回開催されます。概ね欧州や北米、アジア地域が開催場所として選ばれます。IETF 101 meetingは欧州のロンドンで開催されました。次回のIETF 102 meetingはカナダのモントリオールで7月に開催予定となっています。参加者数はIETF 101 meetingで1,189名と報告されました。WGごとに会議室と時間帯があらかじめ設定され、複数のWG会合が同時に開催されます。通常、1つのWGは1度会合が設定されますが、参加者が多くより議論が必要と判断されるWGについては複数回の会合が催されます。また、WGによっては会合

*11 IETF(https://www.ietf.org)。

自体が開催されないこともありますので、IETF meetingに参加する場合には、あらかじめ会合のスケジュールを確認する必要があります。最近では、会場に直接参加せず、オンラインで参加する人も増えています。しかし、意見などを言いたい場合にはやはり会場に直接参加する方が良いでしょう。

IETFのdmarc WGではDMARCの仕様をRFC7489として発行しましたが、現在はARC（Authenticated Received Chain）の仕様検討を行っています。また、Informational RFCとして発行されたDMARCについても、標準化を目指すStandard Trackとするための改善の検討（主にメール再配送問題への対処など）や、DMARCレポートに含まれる情報についての検討なども行っています。

他にメールに関連するWGとしては、DKIMに対しての暗号アルゴリズムや鍵長の追加を検討するdcrup WG、JSON形式のデータを利用してIMAPやSMTPに代わる新しいアクセスプロトコルであるJMAPを検討するjmap WGなどが現在も活動中です。

IETFでの議論には原則として誰でも参加できますので、広い意見を集めることができる反面、なかなか技術仕様が固まらず、長い時間がかかってしまう、という課題もあるようです。メール関連技術に関しては、M^{3}AAWG[*12]内で検討や相互疎通テストなどが行われますので、比較的迅速にRFC化されているようです。

1.3.4 法的な整理

SPFやDKIM、DMARCを受信側で導入するためには、認証のためにメール配送上の情報やメール本文を参照する必要がありますので、原則としてメール利用者から参照することについての同意を得なければなりません。このうち、SPFとDKIMについては、送信ドメイン認証によって大量に送信される詐称メールを判断できるようになることから、認証結果のラベリングについては、一定の条件のもとで、事前に同意を得ていなくても正当業務行為として違法性が阻却できると判断されました[*13]。

一方、新しい送信ドメイン認証技術であるDMARCでは、送信側のドメイン管理者が、DMARCレコードに設定するポリシーの値で、認証が失敗したメールの処理方法を指定することができます。これにより、例えばポリシーを"p=reject"と設定すれば、多くの詐称メールの受け取りを拒否し、メール受信者に不要なメールを届ける必要もなくなります。しかしながら、これまでのSPF、DKIMなどの送信ドメイン認証技術の法的整理では、認証結果のラベリングまで整理されており、DMARCのように受信時の受け取りを拒否する処理まで整理されていませんでした。

＊12 M^{3}AAWG（https://www.m3aawg.org）。
＊13 総務省、「送信ドメイン認証技術等の導入に関する法的解釈について」（http://www.soumu.go.jp/main_sosiki/joho_tsusin/d_syohi/m_mail/legal.html）。

こうした背景から、迷惑メール対策推進協議会などを中心として検討を重ね、最終的に新たに受信側の処理方法を含め、DMARCを導入する上での課題について整理されました。その内容については、総務省から公開されています[*13]。この整理については、DMARCのポリシーに基づく受信側の処理方法とDMARCレポートについて述べられています。DMARCレポートには、集約レポート(aggregate report)と失敗レポート(failure report)の2種類があります。このうち、集約レポートは包括同意により個別の同意を必要とせず、受信メール側が送信ドメイン側に送信することができます。失敗レポートは、エラーメールと同様に元の送信メールの内容が含まれます。そのため、メールの通信当事者とは限らないドメイン管理者への送信は慎重に考えるべきだ、との判断がなされました。よって、包括同意によって失敗レポートを送信する場合には、元の送信されたメールの本文や件名(Subject:ヘッダの内容)を含まないこと、という条件が付きました。失敗レポートの目的の1つは、本当に送信したメールが認証失敗したのか、詐称されたメールなのかを判断することです。元の送信メールが完全な形で含まれていなくても、失敗レポートに含まれる他のヘッダ情報などから、ある程度正規のメールかどうかを判断できるはずですので、現在の制約でも十分有益な情報であると考えています。

これらの整理により、SPF、DKIM、DMARCが受信側も含めて導入が進むことを期待しています。

1.4 おわりに

迷惑メール対策の関係者が参加する「迷惑メール対策推進協議会」が、今年(2018年)で設立10年となります[*14]。その間の2014年には、迷惑メール対策に関する国際的な行政機関の会合であるLAP(London Action Plan、現在はUCEnet)の10年目の会合としてLAP 10 Tokyoが日本で開催されました。更に同じ2014年には、筆者が創設から継続して参加してきたM^{3}AAWGの10周年記念会合が米国ボストンで開催されました。

10年といえば以前は「ひと昔」でしたが、ドッグイヤーのIT業界では、区切りというよりかなりの時間が経過してしまったと言えます。にもかかわらず、迷惑メールの問題がなかなか良くなったように見えない現状には、長い間関わってきた者の1人として、少なからず責任を感じます。しかし、最悪の状況、メールが使われなくなる可能性もあったことを考えれば、それと同時に、多少の貢献ができたのでは、とも思います。実際、モバイルデバイスなどを中心に、チャット系のアプリケーションが複数利用されている状況をみれば、今後もコミュニケーションツールの利用形態は十分に変化する可能性があるとも考えています。本来、こうした新しい仕組みを考える立場でもありますので、現状のメールの課題に取り組みつつ、新しいコミュニケーションの仕組みや、そこで同じような問題が発生しないような検討を続けていきたいと考えています。

執筆者:
櫻庭 秀次(さくらば しゅうじ)
IIJ ネットワーク本部 アプリケーションサービス部 担当部長。
コミュニケーションシステムに関する研究開発に従事。特に快適なメッセージング環境実現のため、社外関連組織と協調した各種活動を行う。M^{3}AAWGの設立時からのメンバー。迷惑メール対策推進協議会 座長代理、幹事会 構成員、技術WG 主査。一般財団法人インターネット協会 迷惑メール対策委員会 委員長。Email Security Conference プログラム委員。一般財団法人日本データ通信協会 客員研究員。

*14 迷惑メール相談センター(https://www.dekyo.or.jp/soudan/contents/anti_spam/index.html)。

RSAアルゴリズム鍵生成モジュールの実装問題(ROCA)

2.1 はじめに

2017年10月、ACM CCS 2017[*1]の論文発表予定をトリガーとして、暗号技術に関連する大きな報道がありました。1つはKRACKsと呼ばれるWPA/WPA2の仕様上の問題に起因する攻撃の報道です[*2]。プロトコルそのものの問題であったことから、大きな問題になると予想され、SNSを通して過剰な反応が見られましたが、比較的軽微な修正で問題をフィックスできたため、少々肩透かしを食らう事例でした。この一件はJNSAセキュリティ十大ニュース[*3]の第4位にランクインするなど、不確かな情報流通や報道の在り方について一石を投じるものとなりました。

もう1つはROCA(The Return of Coppersmith's Attack)と呼ばれるRSA暗号アルゴリズムにおける鍵生成モジュールの実装不備の問題[*4]です。ROCAはKRACKs攻撃とほぼ同じ時期に報道されたにもかかわらず、国内では大きく取り上げられていません。一方で、より現実的な時間とコストでRSA公開鍵の素因数分解が可能になる攻撃であったことから、速やかに対策が行われました。本来持つべき暗号機能が十分に発揮されないまま機能低下を引き起こす攻撃であると認識されたため出荷ベンダーにより修正パッチが展開され、かつ脆弱な鍵生成モジュールで作成されたRSA鍵ペアの更新が促されています。プログラムのバグもしくは設計の不具合により、本来持つべきもしくは持っていると考えられていた暗号機能の実装がはるかに破られやすくなっている事例[*5]はこれまでにも露呈しており、ROCAもその1つとしてリストされたことになります。本フォーカスリサーチでは、過去の失敗事例を紹介しながら、ROCAと同様に鍵や各種パラメータの空間を狭めることで生じる暗号実装の脆弱性について、その要因を整理・俯瞰していきます。また、ROCAを含む一連の研究結果が今後どのような影響を及ぼすかについても触れます。

2.2 ROCAの概要

SSL/TLSなどのセキュリティプロトコルを利用する際には暗号技術が使われています。鍵マークが表示されていることで安全であることを一般ユーザが確認できるブラウザなどのアプリケーションにおいて、暗号化(機密性の確保)とデジタル署名(完全性の確保)を目的に公開鍵暗号方式が使われています。その1つであるRSA暗号方式は素因数分解の難しさを安全性の根拠に置いた方式であり、サーバ証明書の多くはRSA公開鍵が格納されており、例えばSSL/TLSにおいてサーバの確からしさを保証する仕組みとして広く流通しています。最近ではPerfect Forward Secrecy[*6]の観点からサーバ証明書に格納されている公開鍵を機密性確保の用途に用いずに、その都度DHやECDHアルゴリズムでEphemeral鍵(一時鍵)を生成して暗号化を行う方法が推奨されています。そのためブラウザもしくはユーザがサーバの確からしさを確認する際には、署名専用のアルゴリズムも利用可能になっています。その代表例としては楕円曲線暗号をベースとしたECDSA署名[*7]があります。実際RSA公開鍵ではなくECDSA鍵を含むように発行されたサーバ証明書の割合が増えており、主要ブラウザでもサポートされています。一方で、現在でもRSAベースのサーバ証明書が広く利用されており、ROCAの影響を受けることとなります。

2017年10月、チェコのMasaryk大学の研究チームによってInfineon Technologies AG製のRSA鍵生成モジュールの脆弱性とその影響が報告されました。RSAアルゴリズムは鍵生成時には2つの素数を生成してそれらを秘密鍵とし、2つの素数をかけ合わせた合成数を公開鍵とする暗号方式です。公開鍵である合成数を素因数分解するために要する計算量が膨大で解読が現実的でないことで安全性を担保しています。今回RSA鍵生成モジュールにおいて実装上の脆弱性が発見され、生成される鍵に偏りがあることを利用すれば、想定されるよりもはるかに短い

＊1 ACM Conference on Computer and Communications Security 2017(https://ccs2017.sigsac.org)。CCS 2017 - Accepted Papers(https://acmccs.github.io/papers/)。
＊2 Key Reinstallation Attack(https://www. krackattacks.com)。
＊3 JNSA、「セキュリティ十大ニュース」(http://www.jnsa.org/active/news10/)。
＊4 Centre for Research on Cryptography and Security(CRoCS), Masaryk University、"ROCA:Vulnerable RSA generation"(CVE-2017-15361)(https://crocs.fi.muni.cz/public/papers/rsa_ccs17)。
＊5 Internet Infrastructure Review vol.17「1.4.1 SSL/TLS、SSHで利用されている公開鍵の多くが他のサイトと秘密鍵を共有している問題」(https://www.iij.ad.jp/dev/report/iir/017.html)。
＊6 Internet Infrastructure Review vol.22「1.4.2 Forward Secrecy」(https://www.iij.ad.jp/dev/report/iir/022/01_04.html)。
＊7 NIST、"FIPS 186-4 Digital Signature Standard(DSS)"(https://csrc.nist.gov/publications/detail/fips/186/4/final)。6章にてECDSAが規定されている。同文書は4章にDSA、5章にRSAによる署名方式も記載がある。

時間で素因数分解が可能になることが報告されています。本来よりも狭い空間から鍵やパラメータを導出するために脆弱だと認識された事例がこれまでにもいくつか報告されていますが、その多くが疑似乱数生成モジュールの不具合によるものでした。今回も研究結果だけを見ると同様の問題としてカテゴライズされかねませんが、問題の本質は擬似乱数生成部にはなく、実際には素数生成に関わる実装部分の不具合にありました。

2.3 鍵のライフサイクルと過去の失敗事例

公開鍵暗号方式では、デジタル署名や復号時に使用される秘密鍵とサーバ証明書や鍵リポジトリを通じて公にされる公開鍵の2種類の鍵が利用されます。そのため利用者はまず初めにそれぞれの暗号方式で規定された鍵ペアを生成しますが、その際には擬似乱数生成モジュールから安全に利用できる乱数列をソースとして鍵ペアが生成されます。この疑似乱数生成モジュールは鍵生成時だけではなく、鍵利用時(例えば署名時)に必要に応じて参照され乱数列を得ることができるように配備されています。公開鍵を用いた暗号化処理や秘密鍵を用いた署名処理、その対としての署名検証などが行われる鍵利用の際には公開鍵に有効期限が設定されることが多く、期限切れのあとは廃棄され新たな鍵を生成するという一連のライフサイクルが存在します。この鍵管理フローの中には、呼応する秘密鍵が漏えいまたはその可能性が生じた時点で、有効期限内であっても鍵を強制的に無効にするといったパスも用意されます。SSL/TLSにおけるサーバ証明書では有効期限が設けられており、その前に証明書を廃棄する仕組みを考えると、これらの一連のライフサイクルを理解することができるかと思います。

次に、前述の鍵管理ライフサイクルにおいて、どの段階で問題が生じたかを理解するためにいくつかの失敗事例を見ていきます。今回と同様にRSA鍵生成時に問題が生じていた事例の1つとして、2008年に露呈したDebian OpenSSLにおける鍵生成問題が挙げられます[*8]。Debianの特定バージョンにおけるOpenSSLを使って鍵生成を行った場合、極端に少ない鍵空間からしか秘密鍵を導出していないというバグが生じていたため、脆弱な鍵生成モジュールから生成しうる公開鍵リストが公開され、チェックできるような体制が取られました。

同様に鍵生成の問題としては、台湾市民カードの不具合[*9]が2013年に指摘されています。FIPS140-2と呼ばれる暗号モジュールに対する認定基準をクリアしたICカードでしたが、生成される素数に大きな偏りが見られ、素因数分解が可能な公開鍵が生成されている例が報告されています。これはROCAとは異なり疑似乱数生成モジュールの不具合としてカテゴライズされています。ここで報告された脆弱な鍵の中には2012年に報告された、意図せず秘密鍵を共有してしまう問題[*10]の一例として複数含まれており、はるかに少ない計算量で素因数分解が可能となっていました。意図せず秘密鍵を共有してしまう事例は、生成される鍵空間が少ないことに起因しており、あるICカードではたった36通りのRSA公開鍵しか生成できない実装があるなど、非常にインパクトの大きい報告がなされていました[*11][*12]。

また、鍵生成時ではなく鍵利用時における同様の失敗事例もありました。ビットコインウォレット機能を持つAndroidアプリにおいてECDSA署名に用いられるパラメータを再利用したために同一エンティティによる2つの署名から秘密鍵が同定されてしまう事故です[*13]。署名アルゴリズムとしてパラメータを使い回ししてはいけないという制約条件を無視した実装で、具体的にはAndroidアプリが利用する擬似乱数生成モジュールのエントロピーが低いために当該パラメータが偶然重なってしまったことが原因でした。このように様々なフェーズにおいてROCAと同様の問題が起こっていることが分かります。

---

*8 JVNVU#925211、「DebianおよびUbuntuのOpenSSLパッケージに予測可能な乱数が生成される脆弱性」(http://jvn.jp/vu/JVNVU925211/)。
*9 Daniel J. Bernstein et.al, Factoring RSA keys from certifed smart cards:Coppersmith in the wild、"Cryptology ePrint Archive:Report 2013/599" (https://eprint.iacr.org/2013/599)。
*10 PKIDay2012 須賀、「公開鍵の多くが意図せず他のサイトと秘密鍵を共有している問題」、PKI Day 2012講演資料(http://www.jnsa.org/seminar/pki-day/2012/data/PM02_suga.pdf)。
*11 Arjen K. Lenstra et al., Ron was wrong、"Whit is right"(https://eprint.iacr.org/2012/064)。
*12 Nadia Heninger et al., Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices、"Proceedings of the 21st USENIX Security Symposium"(https://factorable.net/paper.html)。
*13 Bitcoin Project、"Android Security Vulnerability"(https://bitcoin.org/en/alert/2013-08-11-android)。

2.4 ROCAにおける問題の本質

Infineon Technologies AG製の暗号ライブラリで見つかったこの問題は、生成される鍵空間の狭さが原因という観点からいえば、Debian OpenSSLなどと同じ要因であるとも言えます。しかし特筆すべきは、これが疑似乱数生成モジュールから吐き出されるランダムデータに偏りがあることで起こるのではなく、鍵生成アルゴリズムに問題があるという点にあります。

報告者らによると、脆弱性の見つかった鍵生成モジュールはソースコードのレビューやリバースエンジニアリングによる発見ではないと主張されています。鍵生成モジュールをブラックボックスとして扱い、複数の鍵ペアを生成させて大量の鍵を取得した後、鍵データの偏り（バイアス）を観測することで取りうる鍵空間が狭いことが明らかになりました。RSA の公開鍵は2つの素数の積で得られます。一般的な鍵生成モジュールの処理はランダムデータから素数候補となる奇数を生成した後、その数が素数であるかどうかを判定する素数判定部を有しています。一般に素数判定は時間を要することから、演算速度もメモリ空間も制約されたICカードなどの環境下で鍵生成を行った場合にはボトルネックとなる可能性があります。今回見つかった脆弱な鍵生成モジュールで発生される素数は特徴的であり、すべての素数空間と比較するとはるかに小さい空間からしかピックアップしていないことが分かっています。発見者も論文中で指摘しているとおり、この特徴的な素数の形式に制約があるのは素数生成を高速化する意図があったと見られます。設計者や実装者はこのような高速化を施すことが結果的に脆弱になってしまうことに気がついていなかったと考えられます。レスポンス時間に対する要求事項のレベルが高く、処理目標よりもはるかに性能が低かったために、本来行うべき処理を省略してしまったとも想像できます。

同じような問題としてはAndroidアプリにおいてバックグラウンドでSSL/TLS通信を行う際に証明書検証を省いているという脆弱性報告が毎年数十件のレベルで報告され続けているという事実もあります。一般的なブラウザであればSSL/TLSサーバと通信する際の証明書検証結果をURL入力エリアやセキュリティインディケータを通じてユーザに知らせていますが、Androidアプリにてバックグラウンドで行われているSSL/TLS通信においてはそのような表示やユーザアクションを必ずしも必要としていないため、証明書検証モジュールを省略して高速化を図るという選択をしていると考えられます。

ROCAで指摘された脆弱性を持つモジュールで生成される素数pは以下のような特徴を持つことが判明しています。

$$p = k*M + (65537^{a} \bmod M)$$

ここでMは2から連続するn個の素数の積であり、生成したい素数の長さによって定められます（256ビットの場合にはn=39、512ビットの場合にはn=71など）。nが固定されるとMが自動的に固定されるためパラメータkとaを適当に動かすことで素数候補を選択していくことになります。例えばRSA-512公開鍵を生成するためには256ビットの素数を2つ掛け合わせることで実現されますが、どのくらいの密度で素数が存在するかを示す素数定理によると$2^{248.5}$個程度の候補があることが知られており、非常に大きな素数空間からたった1つの素数をランダムに選択することができることが分かります。一方で今回の脆弱な素数生成モジュールではn=39つまりM = 2*3*5*...*167は約219ビット長であるためパラメータkとしては37ビット分しか可動せず、パラメータaも62ビット分しか選択できないため結果的に99ビット分しかエントロピーを持っていない、つまり本来よりもかなり狭い鍵空間からしか素数生成していないことが分かります。

RSAは素因数分解の困難性を安全の根拠とするアルゴリズムであり、NISTによりどのくらいのRSA鍵長を利用することが共通鍵暗号での何ビット鍵による暗号化に匹敵するかどうかが見積もられています。例えばRSA2048は112ビット安全性を保有するなどの対応表が記載されており、先に挙げたECDSAなど楕円曲線暗号では鍵長の丁度半分のビットセキュリティを確保するとされています[*14]。2010年に768ビットRSA鍵が素因数分解されるなど、現在は2048ビット長以上の

*14 NIST、SP 800-57 Part 1 Rev. 4、"Recommendation for Key Management, Part 1：General"（https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-4/final）。

RSA鍵を利用することが推奨されており、PKIでのRSA署名利用の際には1024ビット鍵や昨年コリジョンが見つかり脆弱なアルゴリズムとして認識されるようになったハッシュ関数SHA-1[*15]を利用しないよう移行指針がCA Browser Forumなどで規定され、証明書発行の現場では実際にそのような運用が実施されています。CRYPTRECでは毎年公開される報告書[*16]にスーパーコンピュータの持つ能力との比較が記載されており、現時点では2048ビットRSAが十分に安全であることが裏付けられています。

2.5 ROCAの影響

前節で述べたように、素数に制約があることで鍵空間が大幅に狭まっているために素因数分解を行う探索空間が狭まることは容易に理解できます。同じように秘密鍵の制約条件を用いることで素因数分解を容易にする研究があり、ROCAのタイトルにもなったCoppersmithによる方式[*17]がよく知られています。Coppersmithアルゴリズムは RSAで用いられる2素数p、qの積N(=p*q)と片方の素数pの下半分データからpを効率的に復元し結果的に素因数分解を成功させることができます。OpenSSLにおけるHeartBleedバグの脅威の度合いを推し量るために行われたコンペティションで、効率的にSSL/TLSサーバの秘密鍵を導出した手法の1つでもあります[*18]。

表-1 素因数分解に必要なクラウドリソース利用時のコスト

RSA鍵長	素因数分解に必要なCPUリソース	クラウド利用による素因数分解コスト
512 bit	1.93 CPU hours	$0.06
1024 bit	97.1 CPU days	$40〜$80
2048 bit	140.8 CPU years	$20,000〜$40,000

Coppersmithによる方式はそれをROCA向けに拡張したことにより、素因数分解に必要なクラウドリソース利用時のコストを表-1のように見積もっています。これによると現在広く利用されている2048ビットRSA鍵でも現実的なコストで解読可能であることが分かります。この結果の発表を受けてわずか1週間後に5-25%程度効率よく素因数分解ができることが示唆されています。これは原論文の見積りよりも容易にかつ安価に素因数分解が可能であることを意味しています[*19]。

今回問題となった暗号モジュールで生成された鍵であるかどうかをチェックするツールが著者らによって公開されています。その中にはオフラインでチェックできるPythonコード[*20]のほか、公開鍵をブラウザからポストすることでチェックできるオンライン版[*21]やS/MIME署名をメール送信することで結果を得るなど、様々な検証手段が用意されています。ROCA脆弱性チェックの仕組みは非常に単純で簡潔に記載されています[*22]。著者らによると、誤検知はなく2^{-154}の確率で偶然ROCA脆弱な鍵が生成できてしまうと見積もられており、これは無視できるほど小さいものです。

エストニア政府によって発行されたe-Residency IDカード[*23]で利用される証明書はROCAの影響を受けることがアナウンスされており[*24]利用者に対してファームウェアのアップデートと鍵の再生成を促しています[*25][*26]。原論文によれば、調査対象としたe-Residency IDカードは4400程度ありましたが、そのすべてで影響を受けることが示されています。更にROCAは既に2012年からエンバグしており、過去に遡って調査すると現在は有効期限切れではあるものの、当時から素因数分解可能で

＊15 CRYPTREC暗号技術ガイドライン(SHA-1)改定版(https://www.cryptrec.go.jp/topics/cryptrec_20180427_eval_gl_2001_2013r1.html)。Security Diary、SHAttered attack(SHA-1コリジョン発見)(https://sect.iij.ad.jp/d/2017/02/271993.html)。
＊16 CRYPTREC報告書(https://www.cryptrec.go.jp/report.html)。
＊17 Don Coppersmith, "Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known", EUROCRYPT96, 178–189.
＊18 IIJ-SECT Security Diary、「Heartbleed bugによる秘密鍵漏洩の現実性について」(https://sect.iij.ad.jp/d/2014/04/159520.html)。
＊19 The cr.yp.to blog、"2017.11.05:Reconstructing ROCA"(https://blog.cr.yp.to/20171105-infineon.html)。
＊20 ROCA:Infineon RSA key vulnerability(https://github.com/crocs-muni/roca/)。
＊21 ROCA Vulnerability Test Suite(https://keychest.net/roca)、Test your RSA Keys(https://keytester.cryptosense.com)。
＊22 Key fingerprinting(https://github.com/crocs-muni/roca/blob/master/roca/detect.py)。
＊23 Republic of Estonia、"e-Residency"(https://e-resident.gov.ee/become-an-e-resident/)。
＊24 Politsei.ja Piirivalveamet、"Possible Security Vulnerability Detected in the Estonian ID-card Chip"(https://www2.politsei.ee/en/uudised/uudis.dot?id=785151)。
＊25 Politsei.ja Piirivalveamet、"For the user of ID-card and mobile ID"(https://www2.politsei.ee/en/nouanded/isikut-toendavad-dokumendid/id-kaardi-ja-mobiil-id-kasutajale.dot)。
＊26 Politsei.ja Piirivalveamet、"Renewal of document certificates-frequently asked questions"(https://www2.politsei.ee/en/teenused/isikut-toendavad-dokumendid/sertifikaatide-uuendamine/)。

あったにもかかわらず、その脆弱性を知らずに5年間以上使い続けられていた鍵が存在していたと考えられます。日本のベンダーも含め、TPM（Trusted Platform Module）が搭載されている製品群について各社が詳細な情報を提供しています[*27][*28]。推奨されるアクションはファームウェアのアップデートと共に、脆弱なTPMチップで生成されたRSA鍵ペアを廃棄し、新たに生成し直す作業が求められました。TPMは耐タンパー性を持つチップで、秘密鍵への物理的な攻撃を防いでおり、メモリなど記憶装置に鍵データが格納されるよりも安全であるとされてきました。しかし今回のROCAの事例が発覚したことで、鍵を使うモジュールをブラックボックス化することによるデメリットが新たに発生しうることが露呈しました。管理が大変な部分をアウトソースして管理下から追いやる一方で、コントールできないところで新たな脅威が生まれる可能性があるとも言えます。

2.6 ROCA発見に致るまでの経緯

2.4節で見たようにROCAは特殊な形式を吐き出す素数生成モジュールに着目できたことで脆弱性の発見に至りました。リバースエンジニアリングも行わずソースコードにもアクセスせずに大量の素数を観測することで偏りを発見したそのプロセスは過去の同チームによる研究結果がベースとなっています。2016年8月に開催されたUSENIX Security 2016にて38種類の暗号ソフトウエアやICカードで生成させたRSA鍵に関する考察がそれにあたります[*29]。素数p、qの最上位2から8ビット目までの7ビット分の出現状況を示すp、q 2軸のヒートマップから各暗号ライブラリごとに特徴があることが露呈しました。更に素数の最上位2から7ビット目、素数の最下位2ビット目、素数mod 3、公開鍵N mod2という全部で9ビットの最も特徴的と考えられる部分のみを抽出して傾向を調べることで38種類からのモジュールを13カテゴリに分類しています。ROCAの標的となったInfineon製のモジュールはクラス12に分類されており、他のカテゴリに属する暗号ライブラリと比べ突出した特徴を持つことがこの時点で指摘されていました。

更にACM CCSのあとに開催されたACSAC2017でも同じ研究チームから関連する研究結果が発表されています[*30]。これまでの暗号ライブラリに関する知見を生かして公開鍵情報からのみでその公開鍵が生成された暗号ライブラリを特定するという試みです。USENIX Security 2016でも同様のアプローチがありましたが公開鍵Nのmod3、mod4の剰余を計算して偏りがないかを検出する方式を取ることで暗号モジュールを同定しています。研究者らによると誤検知は1パーセント以下であると主張しており、例えばTorで用いられる公開鍵情報を観測することで同じユーザや地域のノードであることが露呈するなどのプライバシの影響も指摘されています。

2.7 ROCAが他の暗号アルゴリズムに影響する可能性

ROCAそのものについてはRSA暗号方式における鍵生成モジュールの脆弱性、つまり素数生成ロジックの問題であるため、素数を生成・利用はするものの秘密裏に保持する必要のないRSA以外の暗号アルゴリズムに影響を及ぼす可能性は低いと思われます。RSA暗号方式では秘密鍵の候補として、例えばRSA2048ビット公開鍵を生成する際には1024ビット長の素数が2つ必要となります。一方で、ビットコインで利用されている前述したECDSA署名では、署名に用いられる秘密鍵は整数（256ビット長）を導出するのみで生成できるため複雑なロジックを必要としません。つまり擬似乱数生成モジュールの確からしさのみが鍵生成モジュールの安全性に影響を及ぼすこととなり、ROCAと同じような脆弱性が横展開される可能性は低いと言えます。

鍵のライフサイクルにおいて鍵生成ではなく鍵利用フェーズを考えます。このとき先に挙げたようなAndroidにおける擬似乱数生成モジュール実装の問題で見たように、本来ならば毎回異なるパラメータを生成して署名する必要がありますが、ここにも素数生成などに見られるような特殊なロジックは必要とされ

＊27 Infineon Technologies AG、"Information on TPM firmware update for Microsoft Windows systems as announced on Microsoft's patchday on October 10th 2017"（https://www.infineon.com/cms/en/product/promopages/tpm-update/?redirId=59160）。
＊28 CERT、"Vulnerability Note VU#307015, Infineon RSA library does not properly generate RSA key pairs"（https://www.kb.cert.org/vuls/id/307015）。
＊29 Centre for Research on Cryptography and Security（CRoCS）, Masaryk University、"The Million-Key Question - Investigating the Origins of RSA Public Keys [Usenix Sec 2016, Best Paper Award]"（https://crocs.fi.muni.cz/public/papers/usenix2016）。
＊30 Centre for Research on Cryptography and Security（CRoCS）, Masaryk University、"Measuring Popularity of Cryptographic Libraries in Internet-Wide Scans [ACSAC 2017]"（https://crocs.fi.muni.cz/public/papers/acsac2017）。

ていません。ただし秘密鍵にせよ各種パラメータにせよ、必要とされているだけのランダムデータを導出せずに先頭がゼロで埋め尽くされているケースや、短いビット列を繰り返し埋めることでデータを確保するケースなど、ECDSA署名アルゴリズムにおいても秘密鍵空間の狭さからくる脆弱性の存在は無視できません。このとき、ビットコインの公開鍵情報は過去のブロックチェーンを参照することによって同じ鍵ペアを導出していないかチェックすることができます。つまり、ほかの第3者と秘密鍵を共有してしまっているか判断することができます。このことからも、あたかも正しい手順で鍵生成を行っているように見えますが、実際には鍵空間が狭められているというバックドアが仕掛けられている実装が存在しうることが想像できます。

ROCAと同じように意図せず鍵生成モジュールが脆弱であることが露呈した場合には、鍵生成を何度も繰り返し行うことで想定されるよりも高い確率で偶然他の誰かが所有するものと同じ鍵ペアを導出する攻撃が考えられます。この場合、コールドウォレットと呼ばれる署名鍵や署名モジュールをネットワークに繋がないことで安全を確保する技術に対しては効果がありません。このようにビットコインに限らず仮想通貨における鍵生成フェーズはとても重要であることが分かります。前述した台湾市民カードの例で見たように、FIPS140-2などによってお墨付きを得た暗号ライブラリやHSM(Hardware Security Module)であっても、脆弱な製品が存在しうることを想定した鍵管理の運用が求められます。ビットコインでは所有者IDとして用いられるビットコインアドレスが存在します。ビットコインアドレスはSecp256k1で識別される楕円曲線上のECDSA署名方式において鍵ペア生成を行い、公開鍵データから2つのハッシュ関数を用いてダイジェストを算出した後でバージョン情報やチェックサム用データを付与し、Base58符号化を用いることで最終的に26〜35文字の可読文字に変換するという手順で生成されます。この手順において、ビットコインアドレスにユーザの指定する文字が出現するように「お好みのアドレス」を導出する方法が知られています。ここで用いられるハッシュ関数はSHA-2とRIPEMD160であり、その出力データを意図したように導出することは困難であることから、異なる鍵ペアを何度も試行錯誤しながら導出することになります。この方式において相当な回数の鍵ペアを生成することから、ここでも安全なランダムデータが必要となります。このような生成ツールが多く出回っていますが、これらを信用する手立てはなく、特にソースコードではなくバイナリで配布されているケースもあるため、利用する際には注意が必要です。

今回RSAアルゴリズムが実装された暗号モジュールの脆弱性と今後起こりうるプライバシ問題を取り上げました。素数生成、素数判定という少々難解なロジックを要するために実装上のバグが入り込みやすい暗号方式であることが再認識されましたが、RSAアルゴリズム自体にはほとんど傷がついておらずこれまでの実装に関する知見が共有されていれば安全に利用することができます。量子コンピュータが出現することで素因数分解が容易となり今後利用できなくなるという予想もあり耐量子暗号と呼ばれる次世代暗号も開発されるようになりました[*31]。NISTでは現在標準化のためのコンペティションが行われており今後3-5年程度の暗号解析と検討ののち2年程度で標準化ドラフトが共有されるスケジュールで進められる見込みです[*32]。次世代の暗号技術を実装する際には、設計者・実装者にとって更に複雑と考えられる仕組みや理解のために膨大な知識が必要な状況が考えられます。RSAなどの現代暗号で起こったROCAのような問題の本質を理解することで次の世代への教訓として受け継いでいくことが期待されます。

執筆者:
須賀 祐治(すが ゆうじ)
IIJ セキュリティ本部 セキュリティ統括室 シニアエンジニア。
2008年7月より現職。暗号と情報セキュリティ全般に関わる調査・研究活動に従事。CRYPTREC暗号技術活用委員会 委員。
暗号プロトコル評価技術コンソーシアム 幹事。電子情報通信学会 ISEC研究会 幹事補佐。IWSEC2018 Organizing committee member。
ECC2018 Organizing committee member。Virtual Currency Governance Task Force(VCGTF) Security WG member。

*31 Internet Infrastructure Review vol.31「1.4.3耐量子暗号の動向」(https://www.iij.ad.jp/dev/report/iir/031/01_04.html)。
*32 Dustin Moody、"THE SHIP HAS SAILED The NIST Post-Quantum Crypto "Competition""(https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/asiacrypt-2017-moody-pqc.pdf)。

大規模メールシステムにおける メールの不正送信対策について

3.1 はじめに

IIJでは長年、数十万ユーザから数百万ユーザ規模のサービスプロバイダ向け大規模メールシステムの構築サービスや大規模メールASPサービスを提供しています。大規模メールシステムの運用には多くのノウハウが必要とされますが、中でも効果的な不正利用対策の組み込みには多くの知見と経験が必要とされます。メールシステム管理者の業務の大半は、メールシステムの不正利用に起因した対応やそれに付随する障害対応ですから、この対策をシステム的にしっかり行うことは、メールシステム管理者の負担を軽減し、またエンドユーザに対する安定したサービス提供にも大きく寄与します。

不正利用の傾向は日々変わるため、メールシステム管理者が都度対処することはいたちごっこであり、対処を続けても不正利用をなかなか減らすことができず、体力を大きく消耗する不毛な戦いになりがちです。一方で、ポイントを押さえてシステム的な運用対応が取れるような仕組みを準備することで、不正利用を相当数減らすことも可能です。

今回は、いろいろなメールシステムでの導入や運用の経験に基づき、効果が見られたメールの不正利用対策をいくつか取り上げ、解説を加えます。なお、ここで述べる対策は、サービスプロバイダが提供するメールサービス約款の定義やユーザ同意が必要なものを含むため、適用に関しては、それぞれのサービスプロバイダにおいて検討が必要となる点に注意が必要です。

3.2 メールの不正利用対策の重要性

エンドユーザがサービスプロバイダのメールサーバを利用してメールを送信する場合は、一般的にMUAからSMTP認証や、Webメールを利用します。SMTP認証及びWebメールの利用には認証IDとパスワードが必要ですが、容易に推測できるパスワードが設定されている場合や、PC自体がウイルスに感染することで認証IDとパスワードが漏えいしやすくなります。この漏えいした認証IDとパスワードを用いて正規のエンドユーザになりすまし、迷惑メールが送信されているケースが多く存在します。

迷惑メール送信は、総じて機械的に繰り返し大量のメールを送信することが多く、その結果、メールシステムに大量のメールが流入し、最終的にインターネットに送信されます。大量の迷惑メールがインターネットに送信されると、インターネット側では、メールシステムの送信出口のIPアドレスを迷惑メールの送信元として認識し、ブラックリストに登録します。ブラックリストに登録されると、主に以下のような影響があります(図-1)。

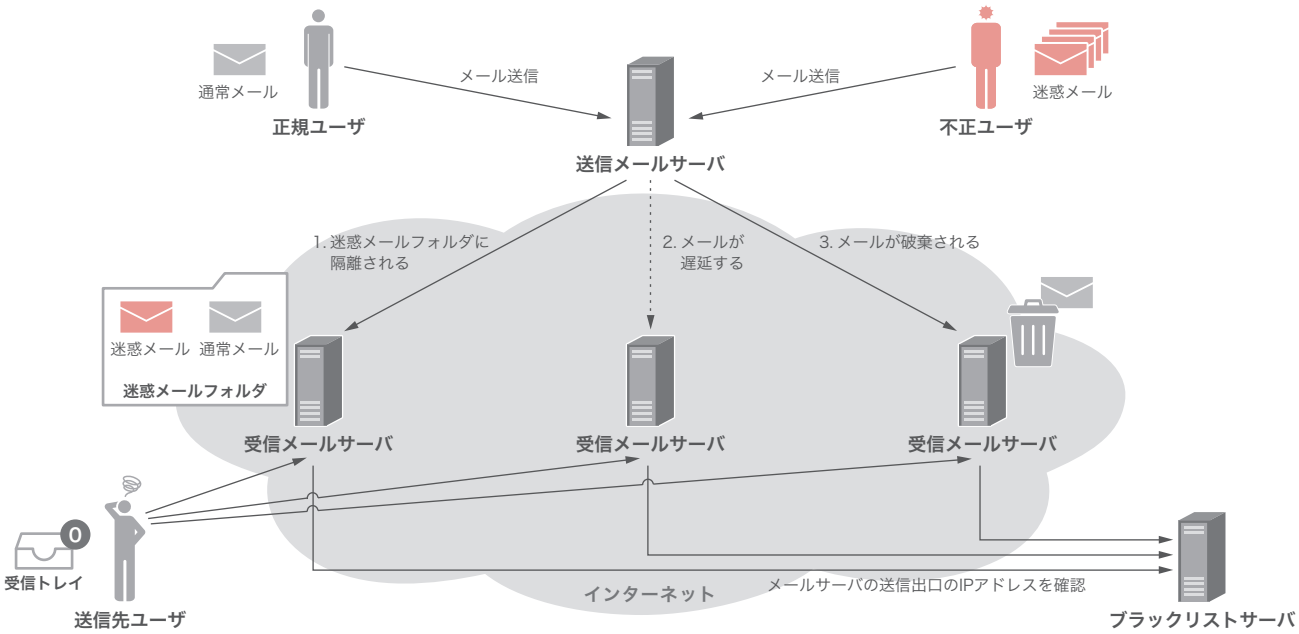


図-1 メールの不正送信における影響

1. 通常のメールまで迷惑メールと判定される(gmail・hotmail・yahoo.com・キャリアメールのケースが多い)
2. 今まで届いていたメールが遅延する
3. メールが破棄されて、まったく届かなくなる

ブラックリストから解除されるためには、原因を取り除いて正常化する必要があります。しかし、原因調査や対策検討には手探りの部分が多く、システム管理者が大きく疲弊するところでもあるため、効果的なメールの不正利用対策はシステム運用上非常に重要なポイントになります。

3.3 メール不正送信の傾向

メールの不正送信の形態は時々刻々と変わってきています。今のトレンドは国外からのメールの不正送信が大半であり、以下のような事例が挙げられます。

1. 国外から単一の認証IDを利用した大量メール送信
2. 国外から複数の認証IDを利用した同時多発的なメール送信
3. 国外からWebメールを利用した大量メール送信

基本的に国外からの通信が関係しているため、メールの送信元IPアドレスを元に送信元の国を判断する仕組みがあれば、効果的なメールの不正送信対策を取ることができます。

国判定の仕組みには、送信元IPアドレスに基づいて国を判別するデータベース及びそれらを容易に利用可能なシステム作りが必要です。国データベースはMaxMind GeoIP2をはじめ数種類存在しますが、有償・無償、サポート有・無、国以外の情報有・無(地域レベルやgmailのレンジ判別など)・更新頻度などの違いがあるため、必要に応じて選択する必要があります。

また国データベースの利用形態には、APIでの利用とダウンロードして整形する方法があります。サービスプロバイダレベルの大規模メールシステムでは、大量のメールを受け取る都合上、国データベースの参照回数が多くなる傾向があり、かつ国データの更新頻度は経験上それほど重要ではないので、ダウンロードして整形する運用が適していると考えています。ダウンロードした国データベースは後述するログ解析やメールサーバでのリアルタイム国判定に利用します。図-2に構成例を記載します。

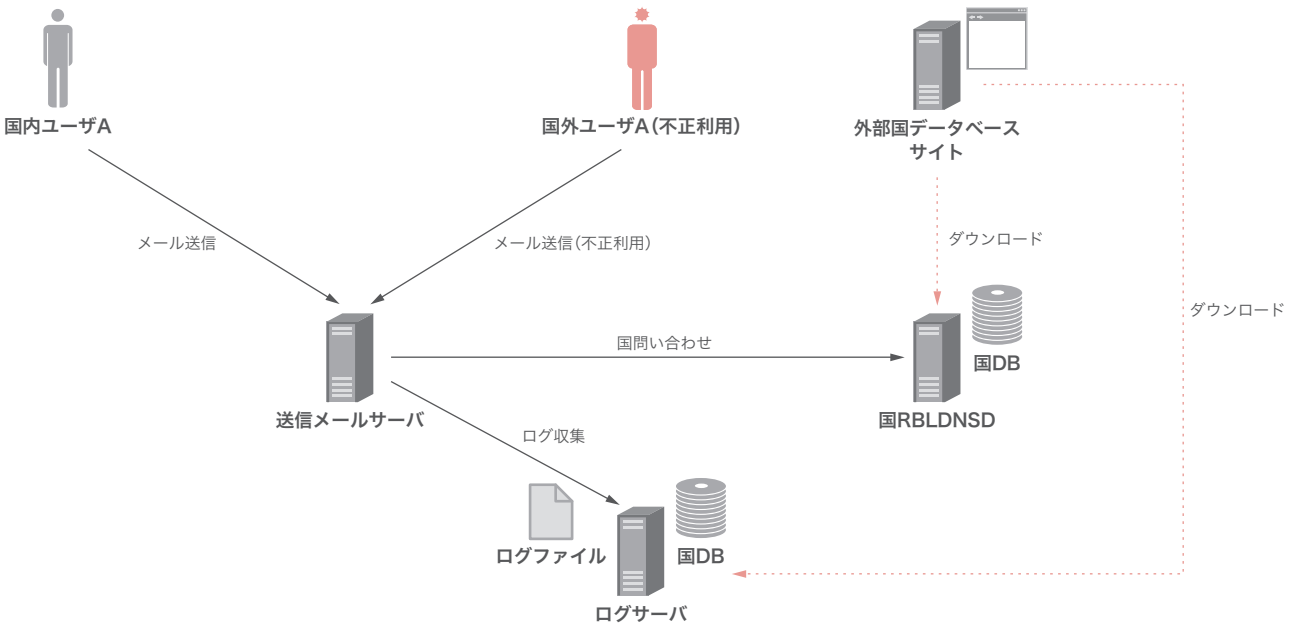
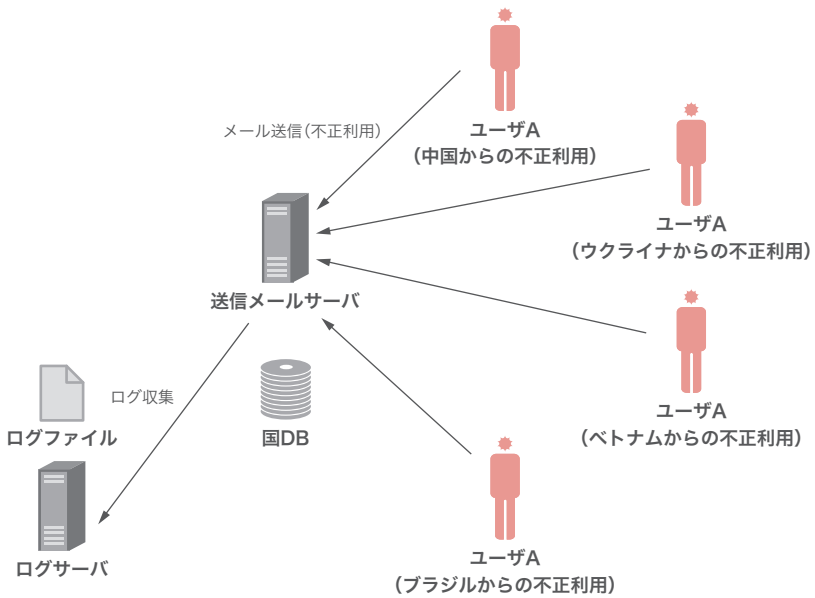


図-2 国データ判定システム概要

3.4 メールログを利用した国外判定の実装

一般的にメールサーバのログにはSMTP認証で利用された認証ID及び送信元IPアドレスが記載されます。前述した国データベースを用いたメールサーバのログを解析するプログラムを作成することで、認証IDごとの国別送信数、全送信数などを解析できるようになり、メールの不正利用を容易に特定できるようになります（図-3）。

- **数時間以内に複数国からメールを送信された場合**
  例えば、中国、ウクライナ、ベトナム、ブラジルからほぼ同時刻に、同じ認証IDを使ってメールを送信された場合は、ほぼ同時刻に同じユーザが世界各国を移動できるはずもないため、認証IDが不正利用されていると考え、当該認証IDからのメール送信を停止するアクションを取りやすくなります。
- **単一国外から大量にメールを送信された場合**
  単一国外からメールを大量送信されるケースも往々にして発生します。この場合は先ほどと異なり明示的に不正利用されているかどうかの判断がつきにくいことがあるのですが、一定時間中に常識的なメール送信数を大きく超過した場合は、一時的にメール送信を停止する対処が考えられます。なお、国外の事業所から正規のメールを大量に送信するケースも存在するため、特定の認証IDをホワイトリストに登録して運用することも考慮に入れる必要があります。



●中国からのメール送信ログ（中国のIPアドレス: 1.0.*.***から認証ID: example@example.comで接続している例）
May 25 03:34:09 server11 smmta[16316]: AUTH=server, relay=from.example.com [1.0.*.***] (may be forged), authid=example@example.com, mech=PLAIN, bits=0
May 25 03:34:09 server11 smmta[16316]: w4OIY9On016316: from=from@example.com, size=0, class=0, nrcpts=1, proto=ESMTP, daemon=MSA, tls_verify=NONE, auth=PLAIN, relay=from.example.com [1.0.*.***] (may be forged)

●ブラジルからのメール送信ログ（ブラジルのIPアドレス: 23.97.**.***から認証ID: example@example.comで接続している例）
May 25 03:35:23 server11 smmta[16319]: AUTH=server, relay=from.example.com [23.97.**.***] (may be forged), authid=example@example.com, mech=PLAIN, bits=0
May 25 03:35:23 server11 smmta[16319]: w4P5Y60n028961: from=from@example.com, size=0, class=0, nrcpts=1, proto=ESMTP, daemon=MSA, tls_verify=NONE, auth=PLAIN, relay=from.example.com [23.97.**.***] (may be forged)

図-3 ログ解析を用いた不正利用判定

このように、ログに基づいた対処は、大半のメールシステムにおいて適用できる点では汎用的かつ効果の高い有用な手法です。しかし、ログ処理はバッチ方式を用いられることが多く、リアルタイム性が落ちることで、メールの不正送信を止めるまでに1万通レベルでのメールを打ち込まれてしまい、結果的にブラックリストに登録されてしまったというケースも散見されます。このため、大規模メールシステムにおいては、メールログ解析だけでは効果を上げきれない場合があります。

3.5 メールサーバでのリアルタイム国外判定の実装

メールサーバが国データベース(独自構築した国RBLDNSDなど)に送信者の送信元IPアドレスを問い合わせることで、送信元の国をリアルタイムに判別する方法があります(図-4)。本方法はメールサーバレベルでリアルタイムに判定できるため、同一時間帯に複数の国から送信された場合は不正利用と見なして即座に停止する対策をとることができます。これにより、メールログ解析で問題となっていた停止までの時間差をゼロにすることができるため、気がついたときには大量のメールが送信された後だったという事態を防ぐことができ、高い導入効果が得られます。

また実装を工夫することで、メール送信全体ではなく国外からのメール送信のみを停止する実装も可能なため、国内利用が大半であるエンドユーザへの影響を小さくすることが可能です。ユーザサポートの観点でも優れた対応と言えます。

一方でメールサーバにリアルタイム検知の機能を実装するにはMilterプログラムの導入、PostfixやsendmailなどのOSSメールサーバの改造、あるいはプログラミング機能を持つ商用メールサーバ(Cloudmark Security Platform for EmailやVade Secureなど)の導入が必要であり、導入効果が高い反面、技術的な難度が高い点が懸念事項として挙げられます。実際、過去の導入時は相当数時間をかけて入念なテストを繰り返して品質を担保することが必要でした。

3.6 Webメール経由でのメールの不正送信の対策

前述のメールの不正送信対策はSMTP認証を利用した送信を対象としていましたが、現在はWebメール経由でのメールの不正送信も増えています。日本のメール業界における代表的なWebメールソフトウェアは数種類あり、ログイン方法やメール送信方法はソフトウェアごとにまったく異なるにもかかわらず、それぞれのソフトウェアに対応した形で大量のメールが送信されたという事実もあります。また、Webメール送信の痕跡が残らないように、送信後は送信済みのメールボックスを空にしてから行儀よくログアウトするという話もあり、相当高度化している印象があります。

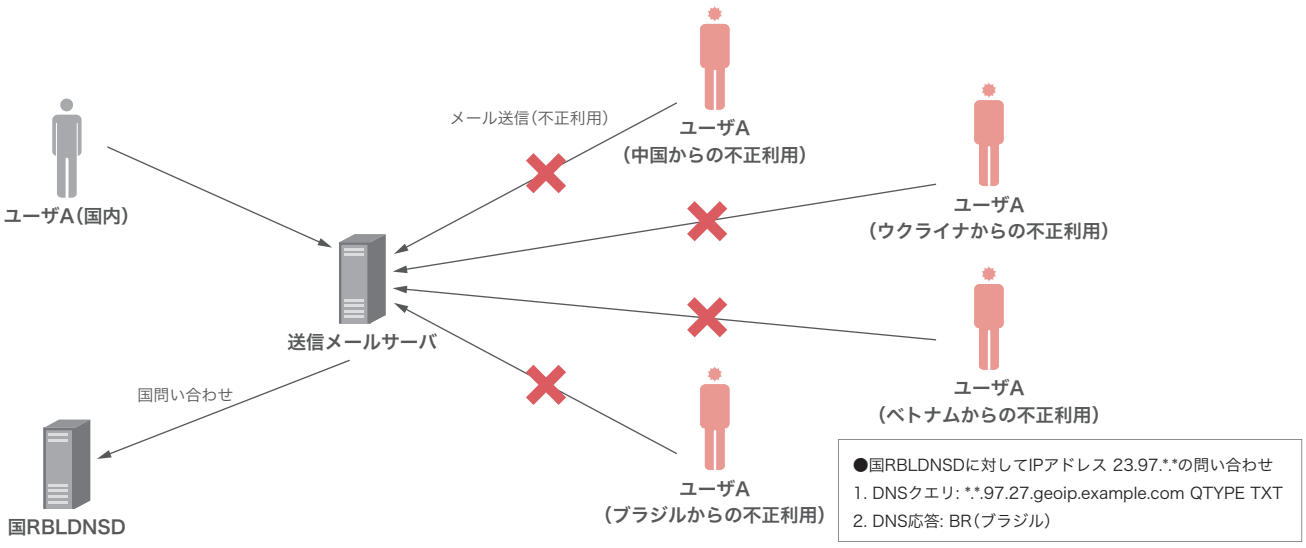


図-4 不正利用のリアルタイム判定

一般的に、Webメールから送信されたメールの送信元IPアドレスは、受信側のメールサーバでは分からないため、前述した国判定が行いにくいのが実情であり、結果的に不正利用の判別が難しく、メールの不正送信の第二の手法として攻撃者に狙われていると考えられます。対策としては、送信元IPアドレスをメールヘッダに埋め込むことが可能なWebメールソフトウェアを採用し、更にメールサーバ側でヘッダ情報を利用した制御を実装するようにしました(図-5)。本方法をとることで、国判定をメールサーバ側でリアルタイムに行うことが可能になり、不正利用の早期発見や対策ができるようになりました。また不正利用アカウントについては、Webメールの利用停止、もしくは国外からのWebメールの利用に限って停止できるようにWebメールを改造したこともあります。

3.7 ユーザ選択による国外からのメール利用制御

国外からのメールアクセスの制御をユーザ選択に委ねるという方法もあります(図-6)。具体的にはSMTP認証を用いたメール送信・Webメール送信・POP/IMAPなどのメール受信を国外から許可する・しないをユーザに選択してもらい、メールサーバサイドでユーザごとのアクセス制御をかける手法になります。

本手法を利用することで、通常時は国外からのメール送信やメール受信、Webメールログインを制限しつつ、国外への出張や旅行の際には管理画面から許可するような運用が可能になります。

エンドユーザに選択権を与えることで利用者の同意を取りやすくし、国外からのアクセスが不正利用されやすい点についての啓蒙活動も兼ねることができる点では有用な手法と言えます。一方で国外からのアクセスをデフォルトで制限するにはユーザの同意を得る必要があり、メールの不正利用を大きく低減する手法とは言いきれない一面もあります。

3.8 SMTP接続DoS攻撃の対応

話題の方向性が少し変わりますが、メールの不正送信の一例として、複数の認証IDを利用して同時多発的にメールサーバにSMTP接続し、その状態を長時間維持させることで、メールサーバのコネクションをわざと枯渇させるケースも見られます。これらは当該メールサーバのタイムアウトの仕組みを把握・利用した巧妙なDoS攻撃であり、新規のSMTP接続がまったくできないケースにも発展し、サービス提供に大きな影響を与える悪質な攻撃です(図-7)。

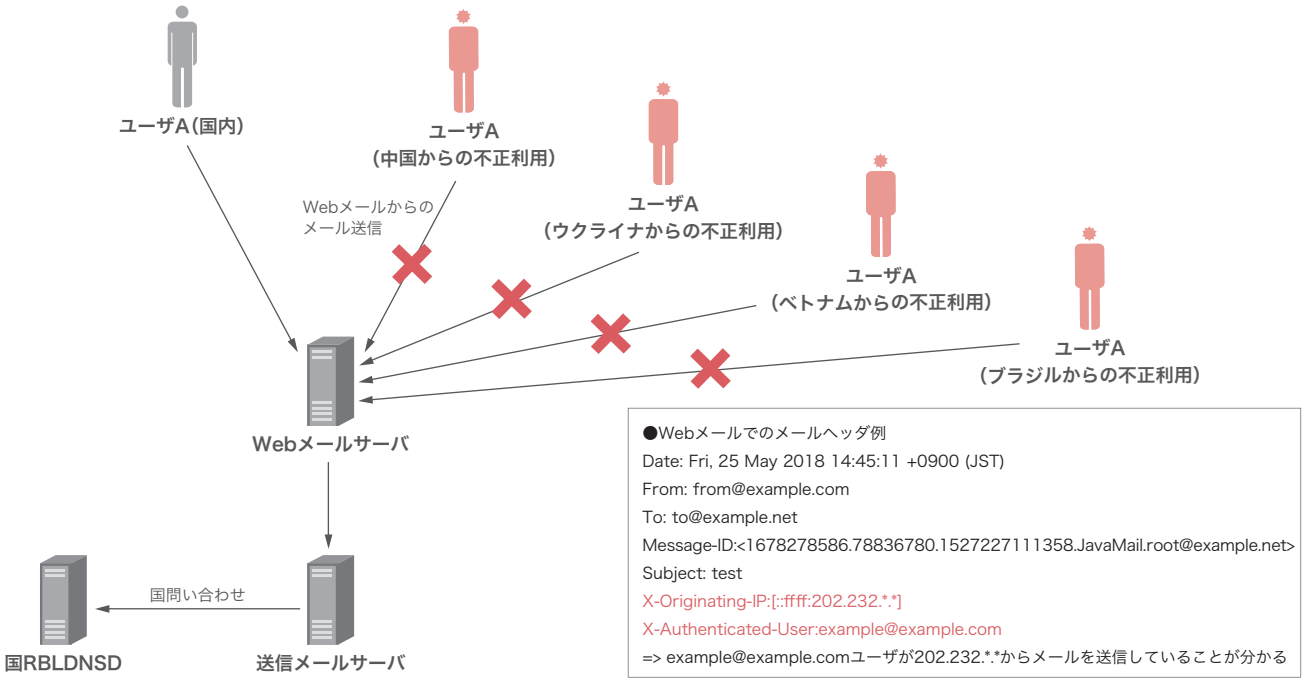


図-5 Webメールを用いた不正送信判定

SMTP接続の長時間維持の対応としては、一般的にメールサーバのタイムアウトを短縮する方法があります。メールサーバ標準のタイムアウトが長いケースも散見されるため、導入時にはタイムアウトを適切にチューニングする必要があります。一方でRFC5321(Simple Mail Transfer Protocol)4.5.3.2. TimeoutsでSMTPタイムアウトの推奨値が定義されており、不用意な設定を行うと正規のメール送信に影響が出ることもあるため、短縮できないことも多く、タイムアウトの調整だけでは限界もあります。

他の手段としては、定期的にSMTP接続数を確認し、システム上限の接続数に近づいてきた場合は、アイドルタイムアウトが長いセッションを判別し、サーバサイドから当該コネクションを切断する方法があります(tcpkillコマンドなど)。同時に、不正利用アカウントに対するSMTPセッションの切断も併せて実施することで、SMTPセッションも蓄積しにくくなり、SMTP接続DoS攻撃のリスクを下げることができます。

3.9 おわりに

大規模メールシステムにおけるメールの不正送信対策には多様な手法が存在し、複数の仕組みを効果的に組み合わせる必要があります。導入にはコストや技術的な難度を伴いますが、メールサービスの維持やシステム管理者の運用負担の大きな低減及びモチベーション向上を考えると、何かしらのメールの不正送信対策の導入は必要不可欠と考えています。今回の記事が安定したメールシステム運営の一助になれば幸いです。

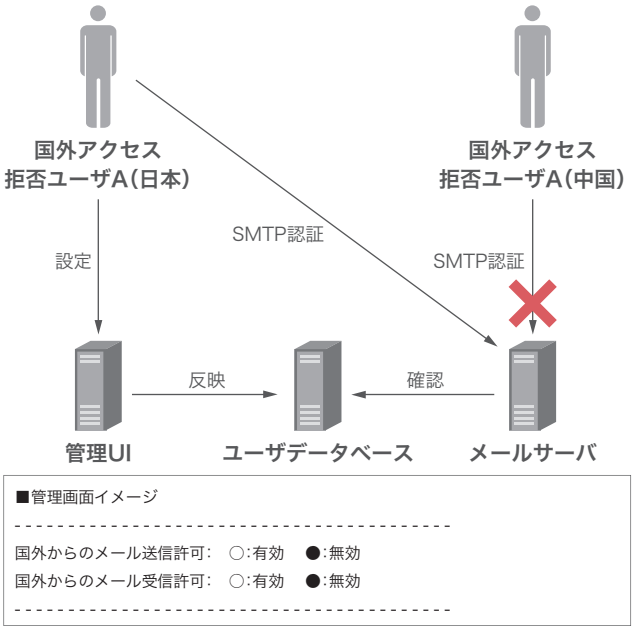


図-6 ユーザ選択による国外利用制御

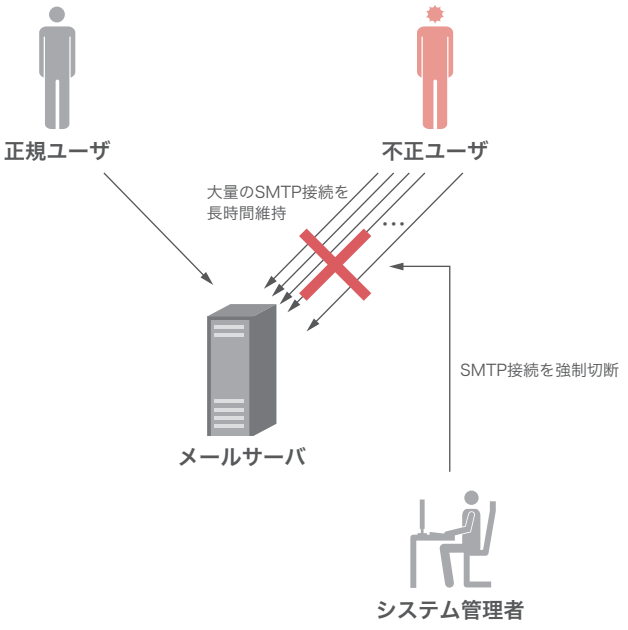


図-7 SMTP接続DoS攻撃

執筆者:
衣笠 茂浩(きぬがさ しげひろ)
IIJ クラウド本部 エンタープライズソリューション部 メールソリューション課 課長。

June 2018 Vol.39

IIJ
Internet Initiative Japan

株式会社インターネットイニシアティブ(IIJ)について

IIJは、1992年、インターネットの研究開発活動に関わっていた技術者が中心となり、日本でインターネットを本格的に普及させようという構想を持って設立されました。
現在は、国内最大級のインターネットバックボーンを運用し、インターネットの基盤を担うと共に、官公庁や金融機関をはじめとしたハイエンドのビジネスユーザに、インターネット接続やシステムインテグレーション、アウトソーシングサービスなど、高品質なシステム環境をトータルに提供しています。
また、サービス開発やインターネットバックボーンの運用を通して蓄積した知見を積極的に発信し、社会基盤としてのインターネットの発展に尽力しています。

本冊子の情報は2018年6月時点のものです。

IIJ-MKTG019-0039

株式会社インターネットイニシアティブ
〒102−0071 東京都千代田区富士見2-10-2　飯田橋グラン・ブルーム
E-mail：info@iij.ad.jp　URL：https://www.iij.ad.jp