Vol.2012-CSEC-59 No.7
2012/12/14

OpenID Connectを利用したメール送信者制御手法の検討

城間 政司[1,a)]　長田 智和[1,b)]　谷口 祐治[1]　名嘉村 盛和[1]　玉城 史朗[1]

概要：本稿では、受信したメールが正当なメールであることを判別できるようにするため、OAuth 2.0 および OpenID Connect を利用し、メール送信者の属性情報を用いたメール制御手法を検討する。本手法は、メールアドレスの所有者が認可したメール送信者に対してアクセストークンを発行し、アクセストークンをメールに添付して送信することで、当該メールアドレスへのメール送信を認可されたメール送信者が送信した正当なメールであることを判別可能にする。

1. はじめに

インターネットの普及により電子メールが広く利用されており、日常生活や業務で必要不可欠なものとなっている。一方、メールアドレス所有者の意図しない多量のスパムメールの存在が問題となっている。マイクロソフト社のセキュリティインテリジェンスレポート [1] によると、当社の Forefront Online Protection for Exchange サービスでチェックしたメールのうち約 8 割がスパムメールとして配送前にブロックされている。このため、広告、フィッシング目的のメールやマルウェアを含んだメールなど、多量のスパムメールの中に重要なメールが埋もれてしまうケースがあり、さまざまなスパムメール対策が利用されている。

一方、Facebook や Twitter に代表されるコミュニティサイトや電話帳を元にしてコンタクトリストを構築する LINE など、SNS(Social Networking Service) の利用者数が近年増加している。実社会における交友関係をホワイトリストにしてメッセージ送信者を制限できるため、SMTP を利用したメールシステムと比較してスパムメッセージの割合が少ない。ところで、このような SNS では、Web API をベースとしたサービス連携機能を提供しており、ユーザの情報を管理している SNS サイト以外のサービスでも交友関係を利用できるようになっている。

本研究では、SNS のメッセージ送信システムと SMTP のメール送信システムの違いとして、SNS における交友関係と、任意のメールアドレスに誰でも送信可能である SMTP の特徴に着目した。本稿では、SNS における交友関係やユーザの属性情報をメール送信者制御に活用する手法を提案する。本手法は、OAuth 2.0[2] および OpenID Connect[3] を利用しており、メールアドレスの所有者が当該メールアドレスへのメール送信を認可したユーザに対してアクセストークンを発行し、そのアクセストークンが添付されたメールが正当なメールであることを判別する。

[1] 琉球大学
1 Senbaru, Nishihara, Okinawa 903-0213, Japan
[a)] joma@ns.ie.u-ryukyu.ac.jp
[b)] nagayan@ie.u-ryukyu.ac.jp

2. スパムメール対策手法

スパムメール対策はユーザレベルからシステム、メールサーバレベルまでさまざまな対策手法がある。

正当なメールであることを証明する手法として、S/MIME[4]、DKIM[5] がある。これらの手法は公開鍵暗号方式を用いており、送信者がメールに電子署名を付与し、受信者が電子署名を検証することでメールの送信者や本文が正当なものであることを確認できる。S/MIME は認証局が発行した証明書によってメールの検証が可能であり、対応するアプリケーションも数多く存在する。ただし、証明書発行のコストが高いという欠点がある。また、DKIM は、メールのヘッダや本文を対象とした署名を送信側メールサーバがメールに付与し、受信側メールサーバは DNS を介して送信側メールサーバの公開鍵を取得して受信したメールの署名を検証する。メールサーバ側で署名の付与と検証をするため、ユーザは新たなアプリケーションやプラグインなどを導入する必要がない。

一方、スパムメールを配送・中継するメールサーバを判定する手法として DNSBL[6] がある。DNSBL では、スパムメールを配送・中継するメールサーバをブラックリストとして公開し、ブラックリスト上のメールサーバから配送されたメールをブロックする。DNSBL にはメールサーバ情報の誤登録やブロック対象メールの取り扱いなどの課題がある。

また、Web 上に公開したメールアドレスを収集するボッ

ト (以降、アドレス収集ボットと称する) への対策として address-munging がある。address-munging は、Web サイトなどに掲載するメールアドレスをアドレス収集ボットに対して難読化する手法である。例えば、taro@example.com というメールアドレスを掲載する場合、“taro AT example.com (AT の部分は@に置換してください)” というように記載し、アドレス収集ボットにメールアドレスが収集されることを防ぐ。ただし、正当なユーザが当該アドレスにメールを送信するためには正しいメールアドレスに修正しなければならず、メール送信に手間がかかったり誤送信するケースが起こりうる。

また、コンタクトフォームを Web サイト上にフォームを設置し、CAPTCHA などと組み合わせて送信者を制限する手法がある。この手法ではメールアドレスを公開する必要がないため、アドレス収集ボットにメールアドレスを収集されることがない。ただし、メールの送信方法が従来の方法と異なるためメール送信者の手間が増えたり、コンタクトフォームから送信したメールをメールクライアントに保存できないという問題がある。

以上のような対策の他にもメール本文の内容を解析してスパムメール判定を行う対策などもあるが、スパムメール作成者とのいたちごっことなっていたり、メール送信の手続きが従来の送信方法と異なったりと、単一の手法だけでは利便性や安全性を保ったままスパムメールを防ぐことは困難である。このため、新たな対策手法や複数の手法の組み合わせの検討が必要である。

3. ID 連携技術を利用したメール送信者制御手法

本節ではメールアドレスの所有者が認可した相手からのメールであることを確認してスパムメールと区別するため、OAuth 2.0 および OpenID Connect を利用したメール送信者制御手法を提案する。本手法はメールアドレスの所有者が設定したポリシーや認証方法によってメール送信者を認証し、認証が成功したメール送信者に対してアクセストークンを発行する。メール送信者はアクセストークンを添えたメールを送信することで、宛先アドレスへのメール送信を認可されたことを証明する。

本手法のメール送信手順は図 1 および以下の通りである。この手順は OpenID Connect の Implicit フローに沿った手順である。

手順 1. メール送信者はユーザエージェントを用い、宛先のアドレスに対して OpenID Connect ディスカバリを実行し、認可リクエストに必要な情報を取得する。

手順 2. メール送信者はユーザエージェントを用いて認可サーバに認可リクエストを送信する。このとき、認可リクエスト内の scope パラメータには mailto スキーム形式の宛先アドレスを含める。

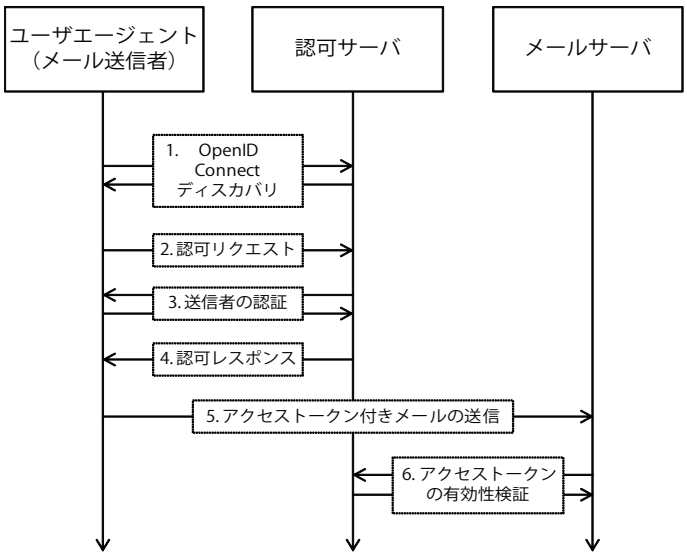

図 1 提案手法におけるメール送信手順
Fig. 1 Overview of E-Mail Sending Process on the Proposal

```
GET /.well-known/simple-web-discovery?service=http
    ://openid.net/specs/connect/1.0/issuer\&
    principal=taro@example.com HTTP/1.1
Host: example.com

HTTP/1.1 200 OK
Content-Type: application/json

{
 "locations":["https://authz.example.com"]
}
```

図 2 OpenID Connect ディスカバリの例
Fig. 2 An example of OpenID Connect Discovery

手順 3. 認可サーバはメールアドレス所有者が事前に設定した認証手順でメール送信者を認証する。

手順 4. 認可サーバは認可レスポンスをメール送信者のユーザエージェントへ送信する。

手順 5. メール送信者は認可レスポンスに含まれるアクセストークンを添付したメールを作成し、宛先アドレスへメールを送信する。

手順 6. メールサーバまたはメールアドレス所有者は、受信したメールに含まれるアクセストークンの有効性を認可サーバに問い合わせて検証する。

まず手順 1 では、メール送信者は、宛先のアドレスに対して OpenID Connect ディスカバリ [7] を実行し、当該アドレスに関する認可サーバの情報を収集する。OpenID Connect ディスカバリは Simple Web Discovery[8] をベースとしたディスカバリプロトコルであり、メールアドレスや URI を元にして認可サーバの URI やサポートする認証ポリシーなどのコンテキスト情報を取得できる。例

えば、taro@example.com というメールアドレスに関する OpenID Connect 対応サーバを探索する場合、図 2 のように、principal パラメータに taro@example.com、service パラメータに http://openid.net/specs/connect/1.0/issuer を指定し、example.com に対してディスカバリを実行する。

手順 2 では、メール送信者が認可サーバに認可リクエストを送信する。このとき、OAuth 2.0 および OpenID Connect の認可リクエストでは、要求するアクセス範囲を scope パラメータで指定することができる。そこで本手法では、mailto スキームで宛先アドレスを指定して認可リクエストを送信する (例：scope=mailto:taro@example.com)。

手順 3 では、メールアドレス所有者が事前に設定した認証方法を用いて認可サーバがメール送信者を認証する。認証方法には、CAPTCHA によるテストや合言葉の入力、Facebook や Twitter とサービス連携してメールアドレス所有者とメール送信者が交友関係にあることを認証する方法などが考えられる。

手順 4 では、手順 3 の認証が成功した場合に認可サーバはアクセストークンを発行し、認可レスポンスにアクセストークンを含めて送信する。

手順 5 では、メール送信者は認可レスポンスに含まれるアクセストークンを添付したメールを作成し、宛先アドレスにメールを送信する。メールにアクセストークンを添付する方法として、メールヘッダ部分に OAuth 2.0 の Authorization ヘッダを追加する方法や taro+ACCESS_TOKEN@example.com のような拡張アドレスを用いる方法などが考えられる。

最後に手順 6 では、アクセストークン付きのメールを受信したメールサーバまたはメールアドレス所有者は認可サーバにアクセストークンの有効性を問い合わせてその有効性を検証し、受信したメールが認可したメール送信者から送信されたものであることを判別する。

以上の手順により、宛先アドレスの所有者が認可したメール送信者によるメールであることを確認して正当なメールとスパムメールの判別が可能となる。

4. 考察

本節では、本稿で提案した手法における宛先アドレスやアクセストークンに関連する内容について検討および考察する。

4.1 ディスカバリ対象となる宛先アドレス公開の検討

3 節の手順 1 でメール送信者は宛先となるアドレスをパラメータに指定してディスカバリを実行し、認可リクエストに必要な情報を収集する。このとき、メール送信者は宛先アドレスを事前に知る必要があるが、Web などで宛先を公開するとアドレス収集ボットに収集される可能性がある。

OpenID Connect ディスカバリではメールアドレスの他に URI を対象とするディスカバリも実行可能である。ここで、宛先となるメールアドレスに紐付く URI をディスカバリ用の URI として公開する方法が考えられる。例えば、taro@example.com というメールアドレスに対して http://example.com/user/123 というディスカバリ用 URI を発行する。3 節の手順 1 ではディスカバリ用 URI を対象として OpenID Connect ディスカバリを実行して認可リクエストに必要な情報を収集し、手順 2 の scope では scope=mailto:http://example.com/user/taro のように scope パラメータを指定する。このとき、認可レスポンス内に宛先となるメールアドレスを含めれば、メール送信に必要な宛先アドレスとアクセストークンをメール送信者にのみ開示できる。

以上のように、メールアドレス所有者はディスカバリ用の URI を公開することで、本来の宛先となるメールアドレスを不特定多数に公開せずにメール送信者にのみ開示できる。

4.2 アクセストークンの盗聴対策

SMTP を用いて送信するメールは基本的に平文のデータであるため、送信経路の途中でアクセストークンが盗聴される可能性がある。さらに、アクセストークンが不正利用され、スパムメールや不正なメールを認可されたメールとして送信されるリスクが生じる。

OAuth 2.0 ではアクセストークンと共にリフレッシュトークンを発行できる。通常アクセストークンは盗聴や不正利用の対策のため有効期限が短く設定されており、リフレッシュトークンを用いて新しいアクセストークンを取得できるようになっている。前述のケースにおいても、アクセストークン付きのメールを検証後にアクセストークンを無効化し、メール送信者はリフレッシュトークンを用いてアクセストークンを再度取得することで盗聴や不正利用に対策可能である。

4.3 アクセストークン添付方法の検討

本手法では、アクセストークンを添えたメールを送信する。このとき、受信したメールのどの情報がアクセストークンであるかを判別しなければならない。

OAuth 2.0 の従来の方法では、HTTP ヘッダ内の Authorization ヘッダにアクセストークンを指定する方法が規定されている。SMTP におけるメールにもヘッダ部分があるため、メールのヘッダに Authorization ヘッダとアクセストークンを追加する方法が考えられる。ただし、メールのヘッダを追加するにはメールクライアント側の本手法への対応が必要となるため、この方法では任意のメール送信者のメールクライアントを用いたメール送信に対応できない。

また、拡張アドレスを用いる方法が考えられる。拡張アドレスとは、メールアドレスのユーザ名の部分を+記号のデリミタと任意の文字列で拡張することで、同一ユーザが複数のメールアドレスを持つことを可能にする (例：taro+jiro@example.com)。3 節の手順 5 で述べたように、アクセストークンを拡張部分に指定してメールを送信することでアクセストークンを添えたメールを送信可能である。ただし、拡張アドレスはアクセストークンの添付以外の用途でも利用される他、メールアドレスが複雑な表記になってしまう。

拡張アドレス方式ではアクセストークンが宛先のメールアドレスに含まれ煩雑な表記となるが、メールアドレス所有者側のメールシステムに本手法を適用するだけでよいため、プロトタイプや個人用途に適していると考えられる。一方、Authorization ヘッダ方式はアクセストークンをユーザに隠匿したメール送信が可能であるが、メール送信者のメールクライアントを本手法に対応するための具体的な仕様の検討が必要となる。

5. まとめ

本稿では、スパムメールと正当なメールを判別するための手法として、メールにアクセストークンを添付する手法を提案した。本手法はメールアドレスの所有者が認可したメール送信者にアクセストークンを発行し、アクセストークン付きのメールが正当なメールであることを判別可能にする。また、本手法は OAuth 2.0 および OpenID Connect をベースとしている。このユーザ認証手続きでは、ID とパスワードによるユーザ認証以外にも CAPTCHA テストや SNS における交友関係を用いた認証方法を利用することで柔軟な認可ポリシーの設定が可能である。

今後の展望として、本手法を実装する上で必要となる詳細なプロトコルの検討や、プロトタイプによる検証を行う予定である。

参考文献

[1] Corporation, M.: Microsoft Security Intelligence Report - Volume 13, Microsoft Corporation (online), available from ⟨http://www.microsoft.com/security/sir/default.aspx⟩ (accessed 2012-11-11).
[2] Hardt, D.: The OAuth 2.0 Authorization Framework, IETF (online), available from ⟨http://tools.ietf.org/html/rfc6749⟩ (accessed 2012-11-11).
[3] Foundation, O.: OpenID Connect Specs, OpenID Foundation (online), available from ⟨http://openid.net/connect⟩ (accessed 2012-11-11).
[4] Ramsdell, B.: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification, IETF (online), available from ⟨http://tools.ietf.org/html/rfc3851⟩ (accessed 2012-11-11).
[5] Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J. and Thomas, M.: DomainKeys Identified Mail (DKIM) Signatures, IETF (online), available from ⟨http://tools.ietf.org/html/rfc4871⟩ (accessed 2012-11-11).
[6] Levine, J.: DNS Blacklists and Whitelists, IETF (online), available from ⟨http://tools.ietf.org/html/rfc5782⟩ (accessed 2012-11-11).
[7] Sakimura, N., Bradley, J., Jones, M. B. and Jay, E.: OpenID Connect Discovery 1.0, OpenID Foundation (online), available from $\langle http://openid.net/specs/openid-connect-discovery-1_0.html \rangle$ (accessed 2012-11-11).
[8] Jones, M. B. and Goland, Y. Y.: Simple Web Discovery (SWD), IETF (online), available from ⟨http://tools.ietf.org/html/draft-jones-simple-web-discovery⟩ (accessed 2012-11-11).