HTTPS プロトコルに対する中間者攻撃に関する研究

松尾和人研究室　　瀬戸崎 喬

1　はじめに

インターネット上のウェブサーバとクライアント間の HTTP プロトコルを利用した機密情報のやり取りは、第三者による改ざんや盗聴、なりすましなどの危険を伴う。これらの危険性は暗号技術を利用した秘匿通信や認証によって防ぐことができる。ウェブ通信では、暗号プロトコル SSL/TLS を HTTP に適用した HTTPS が利用されている。しかし、安全とされている HTTPS による秘匿通信を破る中間者攻撃が存在する。中間者攻撃は盗聴方法の一種であり、攻撃者が通信を行う二者間に割って入り両者になりすますことで、気づかれることなく通信内容の改ざんや盗聴を行う手法である。HTTPS に対する中間者攻撃は、「通信経路の変更」と「通信データの盗聴や改ざん」の 2 つのフェーズに分けられる。通信経路の変更手法には、Address Resolution Protocol (ARP) スプーフィングと Border Gateway Protocol (BGP) に対する意図的な通信経路の変更手法 [1] の 2 種類が知られており、ユーザと攻撃者の位置関係によって使い分けられる。また、通信データの盗聴や改ざん手法は複数知られている [2, 3, 4, 5]。特に、Marlinspike によって提案された sslstrip 攻撃 [5] と呼ばれる攻撃法は、HTTPS の安全性を脅かす脅威となっている。しかし、sslstrip 攻撃を防ぐ対策として、HTTP Strict Transport Security (HSTS) [6] が知られている。

本研究では、インターネットで標準的に利用されている通信経路の制御手法の 1 つである BGP に対する意図的な通信経路の変更手法 [1] の実験を行い、成功条件を考察する。また、この手法に対する現状の対策方法についても考察する。さらに、本研究では、HSTS による対策を回避可能にする sslstrip 攻撃 [5] の拡張も提案する。提案攻撃手法により、sslstrip 攻撃を適用できなかった状況においても中間者攻撃を行うことが可能となる。また、提案攻撃手法の適用範囲を検討し、提案手法の対策も考察する。

2　HTTPS プロトコル

本節では、HTTPS と HTTPS を構成する HTTP と SSL/TLS について説明する。HTTPS (Hypertext Transfer Protocol Secure) は秘匿通信プロトコルである。HTTPS では、平文で通信を行う HTTP 通信を安全にするために、SSL/TLS による暗号化されたセキュアな通信路上で HTTP 通信を行う。HTTP 通信では第三者に盗聴される危険性のあったユーザが入力した ID やパスワードなどを、HTTPS 通信を利用することで安全にサーバに送信することができる。

HTTP (HyperText Transfer Protocol) はウェブの基本となる転送プロトコルである。HTTP が登場した当初は主にハイパーテキストを転送するためのプロトコルであったが、現在ではハイパーテキストだけではなく、静止画や動画などコンピュータで扱えるデータであれば転送可能である。

SSL/TLS とは、Secure Sockets Layer と Transport Layer Security という 2 つのプロトコルの略称である。

SSL/TLS は、2 台のコンピュータ間の通信を暗号化し、第三者にデータの盗聴などをされないようにするプロトコルである。このプロトコルは、公開鍵証明書を利用し、通信データの保護や通信相手が正しいかを確認するための認証も行う。

SSL/TLS による通信は Handshake とデータ転送の 2 つのフェーズからなる。Handshake ではサーバの認証や使用する暗号化アルゴリズムの選択などが行われ、これが完了するとそれらの情報をもとにデータを暗号化し、データ転送が開始される。

3　HTTPS に対する中間者攻撃

HTTPS に対する中間者攻撃は、「通信経路の変更」、「通信データの盗聴や改ざん」の 2 つのフェーズに分けられる。

まず、ユーザとサーバ間の通信経路の変更が HTTPS に対する中間者攻撃の第一フェーズとなる。中間者攻撃を行うためには、攻撃者がユーザとサーバ間に攻撃者が割り込む必要があるため、攻撃者を介さない通信を攻撃者を介するように変更しなければならない。そのため、ユーザとサーバ間の通信経路の変更を攻撃の最初に行う必要がある。

次に、通信データの盗聴や改ざんが HTTPS に対する中間者攻撃の第二フェーズとなる。中間者攻撃では、攻撃者がユーザの機密情報などの盗聴や、サーバの提供するコンテンツの改ざんすることが目的である。HTTP などの暗号化されていない通信に対する中間者攻撃であれば第一フェーズを行うだけで通信データの盗聴や改ざんができるが、HTTPS による通信は暗号化された通信であるため、攻撃者はユーザとサーバ間の通信経路を変更しただけでは盗聴などを行えない。そのため、その暗号化された通信データや暗号化される前の通信データを改ざん等することで、通信データを盗聴や改ざんが可能な状態にする必要がある。

3.1　通信経路の変更

通信経路の変更手法は、ユーザと攻撃者の位置関係によって異なる。現実的な環境として考えられるのは、ユーザと攻撃者が異なる LAN 内に存在する環境である。本節では、2008 年に Alex Pilosov, Tony Kapela が発表した [1]、BGP に対する意図的な通信経路の変更によって、ユーザとサーバ間の通信を攻撃者を介したものに変更する方法を説明する。ここでは、攻撃者はある AS（以降は攻撃者 AS とする）の BGP ルータを掌握しているものとする。

まず、攻撃者は各 AS がどのように接続されているかを把握し、同一の AS を 2 回通過しない攻撃者 AS を介するユーザの属する AS（以降はユーザ AS とする）からサーバの属する AS（以降はサーバ AS とする）までの通信経路の設計をする。次に、攻撃者はサーバ AS が広告しているネットワークよりプレフィックス値が高いネットワークを各 AS に広告する。これは、プレフィックス値が異なる 2 つの経路情報に当てはまる IP アドレスへデータを送信する場合、プレフィックス値が大きい経路情報が最適な経路情報として選択されるためである。このとき、BGP の機能の一つである “prepend” を利用し、攻撃者が設計した攻撃者 AS からサーバ AS の経路上に存在する攻撃者 AS 以外の AS 番号を広告する経路情報に付加する。これにより、ユーザからサー

バに送信される通信データがすべて攻撃者 AS を通過する。同様に、攻撃者がユーザ AS が広告しているネットワークよりプレフィックス値が高いネットワークを “prepend” を利用し、各 AS に広告すると、サーバからユーザに送信されるデータも攻撃者 AS を通過するようになる。

以上のように、攻撃者が偽の経路情報を広告し、最適経路を操作することで、ユーザとサーバ間の通信データがすべて攻撃者を通過することになる。

3.2 通信データの盗聴や改ざん

HTTPS に対する中間者攻撃を行うためには、3.1 節で説明した通信経路の変更を行った後、攻撃者が中継する HTTPS で暗号化されたデータや暗号化される前のデータを操作し、通信データを盗聴や改ざん可能な状態にする必要がある。このような方法には、偽の証明書を利用する攻撃 [2]、Null Prefix 攻撃 [3]、再ネゴシエーション攻撃 [4]、sslstrip 攻撃 [5] などが知られている。本研究では、sslstrip 攻撃について取り上げる。

3.2.1 Sslstrip 攻撃

Sslstrip 攻撃 [5] は、2009 年に Marlinspike によって提案された HTTPS に対する中間者攻撃である。ログイン機能などを実装している一般的なウェブサイトは、HTTP 接続によって取得させるページ内にログインページなどの HTTPS 接続させるリンクを配置していることが多い。Sslstrip 攻撃では HTTPS 接続するように設定されているリンク（`https://`で始まるリンク）を HTTP 接続するように設定されているリンク（`http://`で始まるリンク）に書き換えて攻撃を行う。ユーザと攻撃者間の HTTPS 接続を HTTP 接続にすり替えることで、ユーザと攻撃者間の通信が暗号化されず、攻撃者は通信内容を盗聴することが可能となる。sslstrip 攻撃を図 1 に示す。攻撃者はあらかじめ中間者としてユーザとサーバの間に存在するものとする。

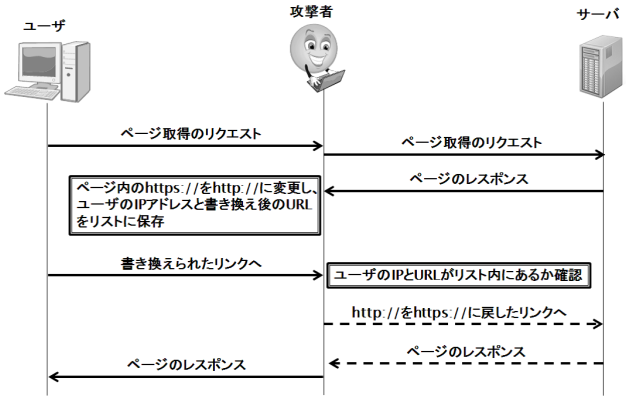

図 1: Sslstrip 攻撃の流れ（実線矢印：HTTP 接続, 破線矢印：HTTPS 接続）

この攻撃の対策として、ブラウザが登録ドメインに対して HTTPS 接続を強制する HTTP Strict Transport Security (HSTS) の利用が知られている。HSTS [6] は、ForceHTTPS [7] を基に Hodges らによって提案された、予め登録された特定のドメインへの接続を強制的に HTTPS 接続にする機能である。サーバが HSTS を利用している場合、攻撃者が HTTPS 接続するように設定されているリンクを HTTP 接続するように書き換えたとしても、ブラウザが強制的に HTTPS 接続に変更するため sslstrip 攻撃が失敗する。

4 BGP に対する意図的な通信経路の変更に関する考察

本節では、ユーザと攻撃者が別の LAN 内に存在する現実的な環境で行う BGP に対する意図的な通信経路の変更の実験を行う。そして、その実験結果から成功条件と対策方法について考察する。

4.1 実験

まず、BGP に対する意図的な通信経路の変更攻撃を実験するために、仮想環境内に仮想マシンを 15 台用意し、3 台を 1 組にして AS100〜500 の 5 つ構成する。各 AS のマシン 1 台を BGP ルータとして動作させる。また、攻撃者は AS300 を掌握し、AS300 の BGP ルータを自由に操作できるものとする。この環境で、ユーザの属する AS（ユーザ AS）、サーバの属する AS（サーバ AS）、攻撃者の属する AS を様々な組み合わせで繋ぐことで、ユーザと攻撃者間の通信路がどのような条件下で変更できるか調査した。実験では、各 AS の接続完了後、通常通りに BGP ルータによる経路情報の広告を行い、さらに攻撃者によって偽の経路情報を広告する。そして、ユーザの端末からサーバに対して ping コマンドを使用し、その経路が意図した経路に変更されているか確認する。また、サーバの端末からも同様の操作を行い、意図した経路に変更されているか確認する。

4.2 BGP に対する意図的な通信経路の変更の成功条件

4.1 節の環境で実験を行った結果、「攻撃者がユーザとサーバの IP アドレスを把握」、「攻撃者が同一の AS を 2 回通過しない通信経路を設計可能」、「ユーザ AS が広告するユーザの IP アドレスが含まれる経路情報のプレフィックス値を超えるプレフィックス値を攻撃者が設定可能」、「サーバ AS が広告するサーバの IP アドレスが含まれる経路情報のプレフィックス値を超えるプレフィックス値を攻撃者が設定可能」の 4 つが BGP に対する意図的な通信経路の変更の成功条件であることがわかった。

以上の条件より、インターネットサービスプロバイダなどの BGP ルータを自由に操作できる人物が攻撃者となれば、この攻撃は高い確率で成功すると考えられる。

4.3 BGP に対する意図的な通信経路の変更の対策

BGP に対する意図的な通信経路の変更への対策として、経路ハイジャック検知システム、BGPsec [9] の 2 つが存在する。経路ハイジャック検知システムとは、システムに登録された経路情報とシステムが BGP 経由で取得した経路情報を比較し、異なる場合は該当する経路情報の登録を行っている AS 等に警告を送信するシステムである。経路ハイジャック検知システムは、国内外で複数のシステムが存在し、運用されている。BGPsec [9] とは、BGP にセキュリティ機能を追加したものである。BGPsec では、受信した経路情報の生成元 AS とその AS に属するネットワークの組み合わせの検証、さらにその経路情報の AS-PATH の検証を行う。BGPsec（特に Path Validation に関する部分）は、現在 IETF で標準化が進められている。これらの対策を利用することで、BGP に対する意図的な通信経路の変更を防ぐ

ことが可能となる。

5 HSTS による対策を回避可能な sslstrip 攻撃の拡張

Sslstrip 攻撃において HSTS による対策を回避するためには、ユーザがアクセスするドメインがユーザのブラウザの HSTS リストに存在しないことが必要である。そこで、本節で提案する sslstrip の拡張手法では、これまでと同様の書き換えに加えて、サーバから受け取った html ファイル内の “https://” から始まるリンクのドメイン名を直前に HTTP 接続していたドメイン名に書き換えることで、書き換えられた html ファイル内のリンクを HSTS で保護されていないドメインに設定し、HSTS による対策を回避して攻撃を可能とする。図 2 に HSTS による対策を行っているサーバに対する提案手法による攻撃手順を示す。

まず、(1) ユーザが http://example.com/index.html にアクセスするためにリクエストをサーバに送信する。次に、(2) 攻撃者はユーザからのリクエストを中継し、その際にリクエスト先のドメイン名 “example.com” を保存する。そして、(3), (4) 攻撃者とサーバが HTTP 通信を行う。(5) サーバからのレスポンスとして index.html を受け取った攻撃者はこのファイル内の “https://login.example.com/login.html” を、(2) で保存したドメイン上の架空のディレクトリ /login/1/を示す “http://example.com/login/1/” に書き換える。ディレクトリ名に含まれる “1” は書き換え順を保存するためのインデックスである。同時に、書き換え前の URL “login.example.com/login.html” をインデックス “1” とともに事前に準備した URL リストに保存する。さらに、(6) 攻撃者は、書き換えた index.html をユーザに送信する。続けて、(7) ユーザが書き換えられた index.html のリンクを経由して書き換えられた URL“http://example.com/login/1/” に対するリクエストを送信する。このとき、(1) で示したようにドメイン example.com は HTTP 接続が可能であり HSTS による保護対象外であることから、ブラウザは http://example.com/login/1/ へのリクエストを HTTP 通信のまま送信する。(8) 攻撃者がこのリクエストを受け取ると、ディレクトリ名 “/login/1/” を基に、URL リスト内のインデックスに対応する URL を確認し、本来の URL “https://login.example.com/login.html” を取り出す。そして、(9), (10) 攻撃者はユーザのリクエストを本来の URL に送信し、それに対するレスポンスを中継する。(11) 攻撃者は中継するレスポンス login.html 内の “https://login.example.com/user. php” を “http://example.com/login/2/” に書き換える。このとき、(5) と同様に、書き換え前の URL とそのインデックス “2” を URL リストに保存する。そして、(12) 攻撃者は書き換えた login.html をユーザに送信する。(13) ユーザがそのページ内に ID とパスワードを入力すると、HTTP 接続でサーバへのリクエストが送信される。(14) このリクエストを中継する攻撃者はユーザの入力した ID とパスワードを入手し、URL に新たに追加した “/login/2/” をもとに URL リスト内のインデックスに対応する URL を確認し、本来の URL を取り出す。(15), (16), (17) では、攻撃者はユーザと HTTP 通信、サーバと HTTPS 通信を行う。以降もこの流れを繰り返すことで、気づかれることなく通信を盗聴し続けることができる。

このように、sslstrip 攻撃におけるリンクの書き換えに加えてドメイン名を書き換えることで、HSTS による対策をされた場合にも、ユーザと攻撃者間を HTTP 接続、攻撃者とサーバ間を HTTPS 接続にしつつ攻撃を行うことができる。

5.1 提案手法の適用条件

3.2.1 節で示した sslstrip 攻撃は、HTTP 接続で取得したページ内のリンクからページ遷移して HSTS を利用していないサーバと HTTPS 接続を行うユーザに対して適用可能である。提案手法も sslstrip 攻撃が適用可能な状況に対しては全て適用可能である。また、サーバが HSTS 対策を行っていても、HTTP 接続用と HTTPS 接続用のドメインが異なり、HTTP 接続で取得したページ内のリンクからページ遷移して HTTPS 接続を行うユーザに対しては提案手法を適用可能である。したがって、サーバが HTTP 接続と HTTPS 接続を混用している多くの状況で提案手法を適用可能であると考えられる。HSTS を利用した上で HTTP 接続と HTTPS 接続を同一ドメイン上で混用している場合は提案手法を適用できないが、HSTS の設定はドメイン単位で行われるため、HSTS を設定した上で HTTP 接続と HTTPS 接続を混用する、このような運用は通常行われない。一方で、ユーザとサーバ間の全ての通信が HTTPS 接続になっている場合には提案手法を適用できないことが明らかであるが、この場合以外に提案手法を適用できない現実的な状況は考えにくい。

また、攻撃対象サーバとユーザ間の全ての通信が HTTPS 接続のみになっている場合でも、HTTP 接続で取得可能なページ内に攻撃対象サーバへのリンクを設置している別のサーバが存在する場合は提案手法を適用可能である。ユーザとの通信を HTTPS 接続によって行っているサーバが HTTP 接続を許可しているポータルサイト等のサーバからリンクされていることは珍しくなく、提案手法の適用範囲は広範囲にわたると考えられる。

5.2 提案攻撃手法の対策

前節まで議論を踏まえ、本節では提案攻撃法の対策を検討する。

5.1 節で示した通りユーザとサーバ間の全ての通信が HTTPS 接続で行われる場合は提案手法を適用することができない。したがって、通常はサーバが HTTPS 接続のみを許可するとことでユーザから HTTPS 接続でリクエストがサーバに送信されると考えられ提案手法による攻撃を防ぐことができる。

次に、HTTP 接続ドメインと HTTPS 接続ドメインが混在する環境に対する HSTS を利用した対策を検討する。ユーザが HTTP 接続したページ内のリンクからページ遷移して HTTPS 接続を行う場合に、HTTP 接続するドメインと HTTPS 接続するドメインが異なると、提案手法により HSTS を回避して攻撃することが可能である。そこで、ユーザが HTTP 接続するドメインと HTTPS 接続するドメインを同一にした上で HSTS による対策を施すことで提案手法を防ぐことができる。

上記の方法以外に HSTS による対策を施した上でページ内のリンクの書き方を工夫する方法が考えられる。

次に、複数組織間にわたる攻撃の対策について検討する。攻撃対象サーバの通信が全て HTTPS 接続になっていても別のサーバの HTTP 接続可能なページからリンクされていると、そのリンクを利用して提案手法に

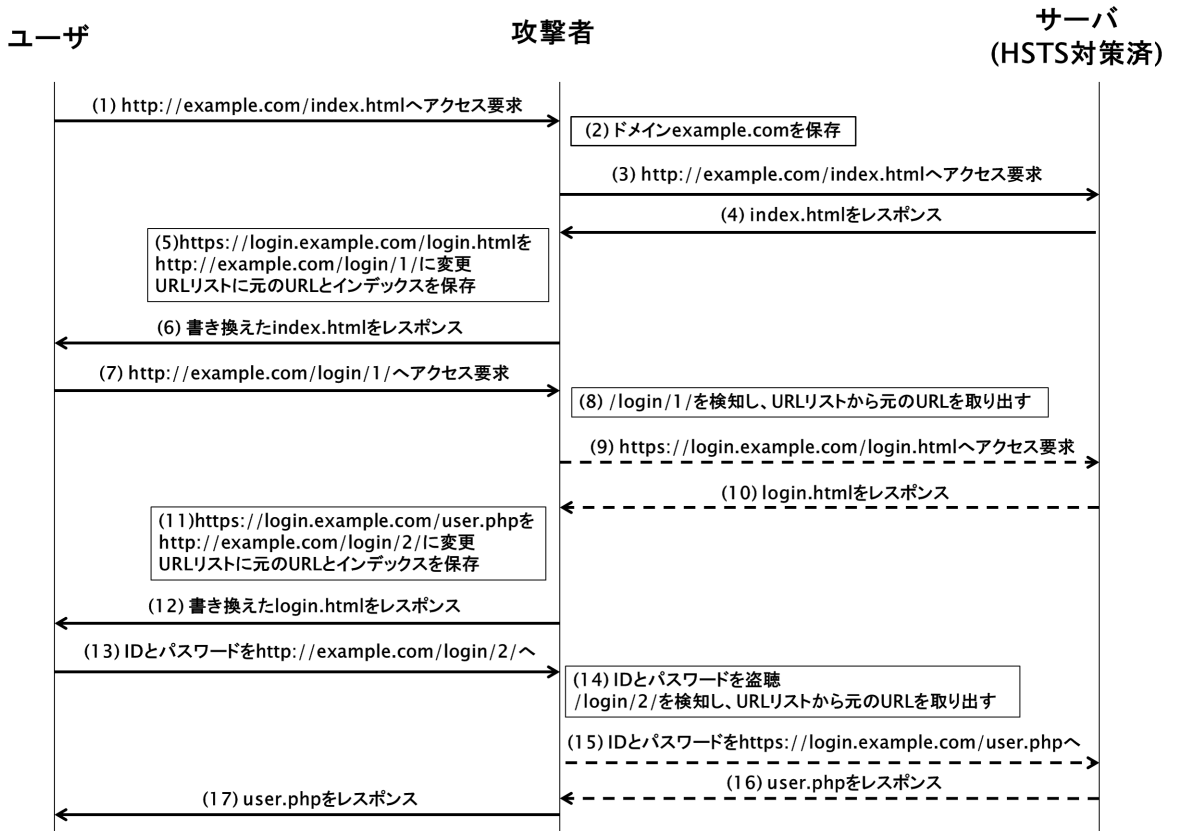

図 2: 提案攻撃手法の流れ（実線矢印：HTTP 接続, 破線矢印：HTTPS 接続）

よる攻撃が可能となる。そのため、単一の組織内で対策を施しても攻撃を防ぐことは困難であり、インターネット上の全てのウェブ通信に対する対策が必要となる。具体的には、インターネット上の全ての HTTP 通信を HTTPS 通信に置き換えることが必要であると考えられる。

6 まとめ

本研究では、HTTPS プロトコルに対する中間者攻撃について考察などを行った。

まず、BGP に対する意図的な通信経路の変更に関する考察を行った。この手法には、4 つの成功条件があり、インターネットサービスプロバイダなどの BGP ルータを自由に操作できる人物が攻撃者となれば、高い確率で攻撃が成功すると考えられる。

次に、HSTS による対策を回避可能な sslstrip 攻撃の拡張を提案した。提案攻撃手法により、サーバが HSTS を利用して sslstrip 攻撃の対策をしていても、HTTP 接続するドメインと HTTPS 接続するドメインが異なり、ユーザが HTTP 接続したページ内のリンクからページ遷移して HTTPS 接続を行う場合には攻撃が可能となる。さらに、提案攻撃手法の適用範囲の拡張と提案手法の対策についても議論した。

今後は様々な状況で実験を行いながら、BGPsec の脆弱性や更なる提案手法の拡張の検討が必要である。また、本論文で考察した提案手法の対策よりも簡便かつ効果的な対策方法の発見も今後の課題である。

参考文献

[1] Alex Pilosov and Tony Kapela. Stealing The Internet An Internet-Scale Man In The Middle Attack. In *Defcon 16*, 2008. https://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-pilosov-kapela.pdf.

[2] F. Callegati, W. Cerroni, and M. Ramilli. Man-in-the-Middle Attack to the HTTPS Protocol. *IEEE Security and Privacy Magazine*, Vol. 7, pp. 78–81, 2009.

[3] M. Marlinspike. NULL Prefix Attacks Against SSL/TLS Certificates. In *Blackhat 09 and Defcon 17*, 2009. https://moxie.org/papers/null-prefix-attacks.pdf.

[4] T. Zoller. TLS/SSLv3 renegotiation vulnerability explained, 2011. http://www.g-sec.lu/practicaltls.pdf.

[5] M. Marlinspike. New Tricks For Defeating SSL In Practice. In *Black Hat DC 2009*, 2009. https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf.

[6] J. Hodges, C. Jackson, and A. Barth. HTTP Strict Transport Security (HSTS). RFC 6797, RFC Editor, November 2012. http://www.rfc-editor.org/rfc/rfc6797.txt.

[7] C. Jackson and A. Barth. ForceHTTPS: Protecting High-Security Web Sites from Network Attacks. In *Proc. of the 17th International World Wide Web Conference (WWW2008)*, 2008. https://crypto.stanford.edu/forcehttps/forcehttps.pdf.

[8] Apache Software Foundation. Apache HTTP Server. http://httpd.apache.org/.

[9] Matthew Lepinski and Sean Turner. An Overview of BGPsec. Internet-Draft draft-ietf-sidr-bgpsec-overview-08, IETF Secretariat, June 2016. http://www.ietf.org/internet-drafts/draft-ietf-sidr-bgpsec-overview-08.txt.

[10] 瀬戸崎喬, 松尾和人. HSTS による対策を回避可能な sslstrip 攻撃. コンピュータセキュリティシンポジウム 2016 論文集, pp. 733–740, 2016.