sslstrip攻撃の脅威に関する検討

王 晶栄[1,a)]　矢内 直人[1]　大久保 隆夫[2]　岡村 真吾[3]

概要：sslstrip 攻撃はユーザと通信先のサーバ双方に気づかれることなく、HTTPS から HTTP にダウングレードさせ通信内容を傍受する中間者攻撃である。HSTS が提案されて以降、sslstrip 攻撃は対策されたという認識が広まっているが、実際に HSTS を施しているサーバは多くないのが現状である。本稿では、まず JP ドメイン上位 500 の web サイトを対象に、既存の sslstrip 攻撃の影響度を HSTS 対応済みのサーバの割合など実態調査を通じて明らかにする。また、さらなる潜在的脅威の検討として、新たな sslstrip 攻撃を示す。既存攻撃が 14%の JP ドメインに有効であることに対し、本稿の攻撃はほぼ 100%のドメインに有効である。関連して、これらの攻撃に関する潜在的脅威および対策についても考察する。

キーワード：sslstrip, HSTS, HTTPS, 中間者攻撃, 実態調査

A study on sslstrip

Shouei Ou[1,a)]　Naoto Yanai[1]　Takao Okubo[2]　Shingo Okamura[3]

Abstract: Sslstrip attack is a man-in-the-middle attack, whereby an adversary downgrades HTTPS communication to HTTP and intercepts contents of the communication without both the user and the server being aware. It has been widely recognized that the sslstrip attack can be prevented by HSTS. However, only a few servers have deployed HSTS. In this paper, we first conducted an empirical study about threats by ssltrip attacks through evaluation of percentage of severs with HSTS, targeting the top 500 JP domains. In addition, we show new sslstrip attacks. Whereas the existing ssltstrip attacks are useful for only 14% domains, our new attacks are useful for almost 100% domains. Furthermore, we discuss potential threats and their countermeasures with respect to the sslstrip attacks.

Keywords: sslstrip, HSTS, HTTPS, Man-in-the-middle Attack, Empirical Study

1. はじめに

HTTPS は HTTP において暗号化プロトコルである SSL/TLS を利用することで通信のセキュリティを保証する、現在のウェブサービスにおいて欠かせない技術である。HTTPS を利用する際は、サーバは証明書を取得した上でクライアントのブラウザと通信を行う。近年では Let's Encrypt [*1] など無料の証明書発行サービスもあり、HTTPS の普及は急速に進んでいる。

この HTTPS に対する攻撃として sslstrip 攻撃 [1] が知られている。この 攻撃は攻撃対象となるクライアントとサーバ間の通信において、攻撃者が中間者として動作することで通信を HTTPS から HTTP に切り替える攻撃である。この攻撃は攻撃者観点では HTTP としての平文通信を盗聴できる一方、サーバとの通信は SSL/TLS により依然暗号化されているため検知は容易ではない。

sslstrip 攻撃の対策として HTTP Strict Transport Security (HSTS) [2] が知られている。直観として HSTS は特定のサーバへの接続が HTTP で行われた際に、次回以降そのサーバへの接続を全て HTTPS にする技術である。これにより、sslstrip 攻撃における盗聴を防ぐことが可能となる。しかしながら、HSTS の実装は不備が多く、普及

[1] 大阪大学. Osaka University
[2] 情報セキュリティ大学院大学. Institute of Information Security
[3] 奈良工業高等専門学校. National Institute of Technology, Nara College
[a)] sho-ou@ist.osaka-u.ac.jp
[*1] https://letsencrypt.org/

も進んでいないことが知られている [3]。また、近年では sslstrip 攻撃をさらに発展させた攻撃 [4], [5] として、HSTS を部分的に無効化できるものもある。このため、対策としての HSTS の実用性含めて、sslstrip 攻撃がどの程度の脅威となっているか明らかになっていない。

本稿ではまず JP ドメイン上位 500 個の Web サイトを対象に、HSTS の実装状況を調査する。これにより、クラシックな sslstrip 攻撃が我が国においてどの程度の影響を与えるか明らかにする。次に HSTS を回避可能な攻撃として瀬戸崎-松尾攻撃 [5] が知られているが、この攻撃をさらに発展させた攻撃手法を示す。本稿の攻撃では、まずユーザが踏み台となるサイトとして、ssl が施されていない任意のサイトに HTTP 接続した際を攻撃のきっかけとする。このとき、踏み台となるサイトからのレスポンスの中に、真の攻撃対象となるサイトのドメインをすり替えた情報を含める。これにより、瀬戸崎-松尾攻撃では対策がされてしまう状況においてすら、sslstrip 攻撃が可能になる。また、上述した調査結果を踏まえて JP ドメインにおける影響度も確認する。とくに、瀬戸崎-松尾攻撃が可能な JP ドメインの割合は従来の sslstrip 攻撃に加えて 14%程であったが、本稿の攻撃は後に述べる制約があるもののほぼ全てのドメインに対して有効である。また、関連して sslstrip 攻撃の潜在的な脅威とその対策についても明らかにする。

本稿の貢献を要約すると、以下の通りである。

- JP ドメインにおける従来の sslstrip 攻撃の影響度調査として、瀬戸崎-松尾攻撃 [5] が 14%の JP ドメインに有効であることを示す。
- 新しい sslstrip 攻撃を示すことで、ほぼ 100%の JP ドメインに有効な、HSTS による対策をより緩和する潜在的な脅威を示す。
- 本稿で示した sslstrip 攻撃の影響度と対策を検討する。

2. sslstrip

本節では、sslstrip 攻撃 [1] にの概要と HSTS による対策について述べる。

2.1 攻撃概要

sslstrip 攻撃 [1] は，HTTPS に対する中間者攻撃である。ログイン機能を実装しているウェブサイトは，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 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を HTTP 接続するように書き換えて攻撃を行う。この攻撃によってユーザとサーバ間の通信が暗号化されず，中間者である攻撃者は通信内容を盗聴できるようになる。この攻撃例を図 1 に示す。通信先のサーバはホームページに `target.com`, ログインページに `login.target.com` というドメインを用意しており、それぞれ HTTP 接続、HTTPS 接続を要求するものとする。また、ログインページでは ID とパスワードを要求し、HTML のフォームを使用しているものとする。さらに、攻撃者の要件として、あらかじめユーザとターゲットとなるサーバとの通信経路上に存在する中間者とし、ssl で保護されていない通信に関しては通信内容を読み取ることができるものとする。

まず、ユーザは通信先のサーバのホームページ `target.com` に HTTP でアクセスする。このとき攻撃者はサーバからのレスポンス内のログインページへのリンク `https://login.target.com/login` を、non-ssl なリンク `http://login.target.com/login` に書き換える。この改ざんしたファイルをレスポンスとすることで、ユーザがログインを行う際に、HTTP で `login.target.com` にアクセスすることになる。その後に攻撃者は HTTPS 接続でサーバにリクエストを送信し、このリクエストに対するレスポンス内のフォームのアクション `https://login.target.com/account` を `http://login.target.com/account` に書き換えて、HTTP 接続でユーザにレスポンスを返す。これにより、ユーザが入力した ID 及びパスワードは HTTP 接続で送信されることになる。

以上のように、攻撃者はユーザからのリクエストを全て HTTP 接続させるようにして、攻撃者自身とサーバ間の通信を HTTPS 接続とすることで、ユーザとサーバ双方に気づかれることなく、攻撃が可能となる。

2.2 HSTS による対策例

HSTS [6] は、Web ブラウザに特定のドメインに対して HTTPS 接続を強制するポリシーである。サーバが Strict-TransportSecurity(STS) ヘッダを含めたレスポンスを送ることで、ブラウザは HSTS リストにサーバのドメインを追加する。以降、ユーザが HSTS リストに含まれているドメインに HTTP リクエストを送る場合、ブラウザが自動的に HTTPS 接続としてサーバと通信を行う。この仕組みを用いることで、前述の sslstrip 攻撃の対策が可能となる。

以下に、HSTS による sslstrip 攻撃の対策例を述べる。なお、以下の説明ではサーバの構成は図 1 に示した sslstrip 攻撃の例と同じものとし、ログインページのドメイン login.target.com は HSTS リストに含まれているものとする。まず 1. の処理でユーザが target.com に HTTP でアクセスし、3. の処理で攻撃者はレスポンス内のリンクを書き換える。このとき、4. のユーザがログインを行うためにリンクをクリックした際に HSTS が作動し、HTTP ではなく HTTPS 接続でサーバにリクエストを行う。この場合、攻撃者はサーバからのレスポンスの中身を確認できないためリンクの書き換えが出来なくなる。また、攻撃者は login.target.com のサーバ証明書を保持していないため、ユーザはサーバ認証を行うことができない。以上のように、HSTS の普及によって sslstrip 攻撃は対策されてし

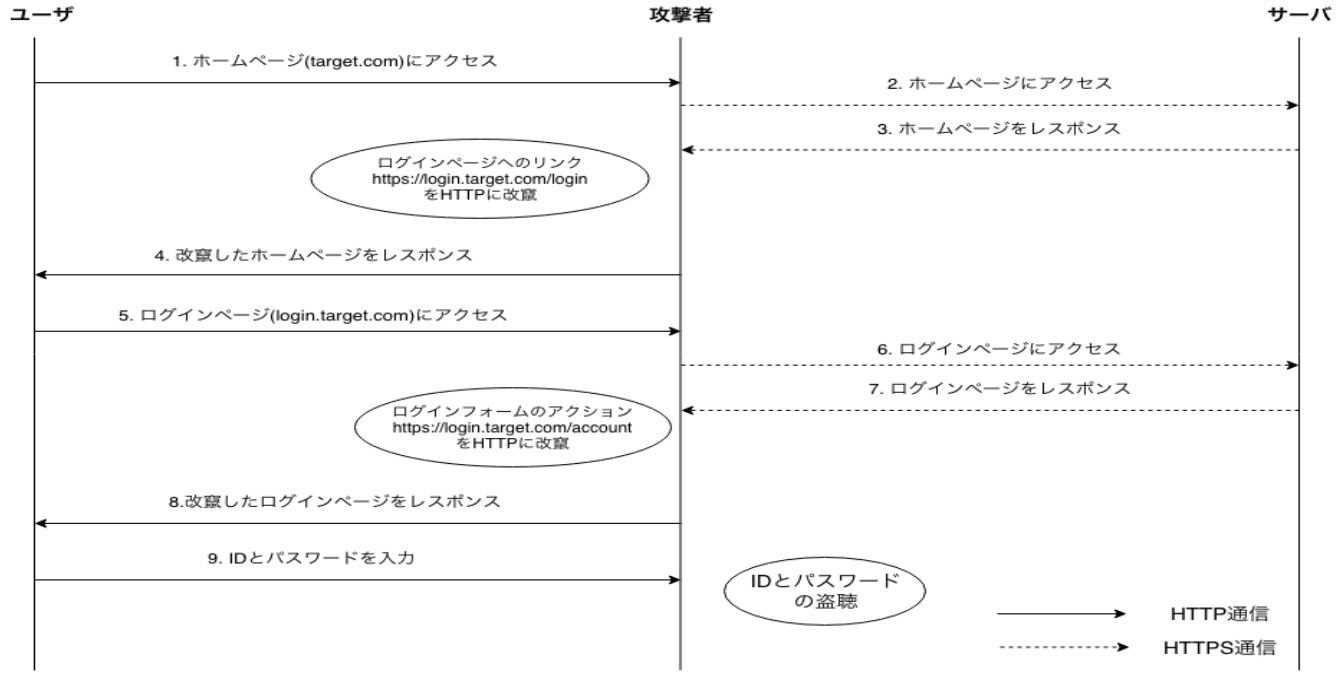

図 1 初期の sslstrip 攻撃の攻撃例

まい、HTTPS 接続への脅威は失われたと思われた。

3. 実態調査

本節では、世の中の Web サイトに対する ssltrip 攻撃の脅威として、著者らの知る限り最も強力な攻撃である瀬戸崎-松尾攻撃 [5] の影響度を調査する。まず瀬戸崎-松尾攻撃について述べ、調査方法および調査結果について述べる。

3.1 瀬戸崎-松尾攻撃

HSTS による対策を回避する方法として提案された手法が、瀬戸崎-松尾攻撃 [5] である。この攻撃手法は従来の sslstrip 攻撃を基に拡張する形で作られており、ユーザが HSTS リストにないドメインへのリンクに書き換えることで、HSTS を作動させずにユーザに HTTP 接続をさせる。具体的な攻撃例を図 2 に示す。サーバの構成は 2.1 節の sslstrip 攻撃の例と同じものとし、ログインページのドメイン `login.target.com` は HSTS リストに含まれているものとする。従来の sslstrip 攻撃と同様、ユーザははじめにサーバのホームページ `target.com` に HTTP 接続でアクセスする。攻撃者はサーバからのレスポンス内のログインページへのリンク `https://login.target.com/login` を `http://target.com/login` に書き換える。ユーザがログインページへのリンクをクリックした場合、`target.com` は HSTS リストに含まれていないため、そのまま HTTP 接続でリクエストが送られる。攻撃者はそのリクエストを受けて、接続先を `login.target.com` に変え、HTTPS 接続でサーバにリクエストを送る。そして、サーバからのレスポンス内のログインフォームのアクション `https://login.target.com/account` を `http://target.com/account` に書き換えてユーザにレスポンスを返す。`target.com` には HSTS が作動しないため、ユーザが入力した ID 及びパスワードは HTTP 接続で送信されることになる。この攻撃手法により、sslstrip 攻撃は再び HTTPS 接続の安全性に対する脅威となった。

3.2 調査内容

日本の主要な Web サイトの構成を調べ、従来の sslstrip 攻撃及び瀬戸崎-松尾攻撃が通用する割合を調査する。調査範囲としては、tranco-list [7](2019 年 6 月 9 日時点) より JP ドメイン TOP500 個を対象とする。

調査方法としては、sslstrip 攻撃が成立する環境を把握するため、まず攻撃者相当のプロキシを構築し、ユーザがプロキシを通って通信するように設定した。このとき、各 Web サイトのトップページ、ログインページの構成について、(1) トップページが ssl で保護されているか否か、(2) トップページが HSTS で保護されているか否かという二つの観点で調査している。これらを後述する表 1 に記載のカテゴリーに分類し、各カテゴリーに属するいくつかの Web サイトについて sslstrip 攻撃を試行した。その結果として、カテゴリー毎に通用する攻撃手法を確認している。

3.3 調査結果

調査の結果を表 1 に示す。なお、ログインページはログインページ以外にも問い合わせページなど機密情報と考えられるページへのリンクが存在する場合、ログインページとみなして調査している。これらに相当するページが存在

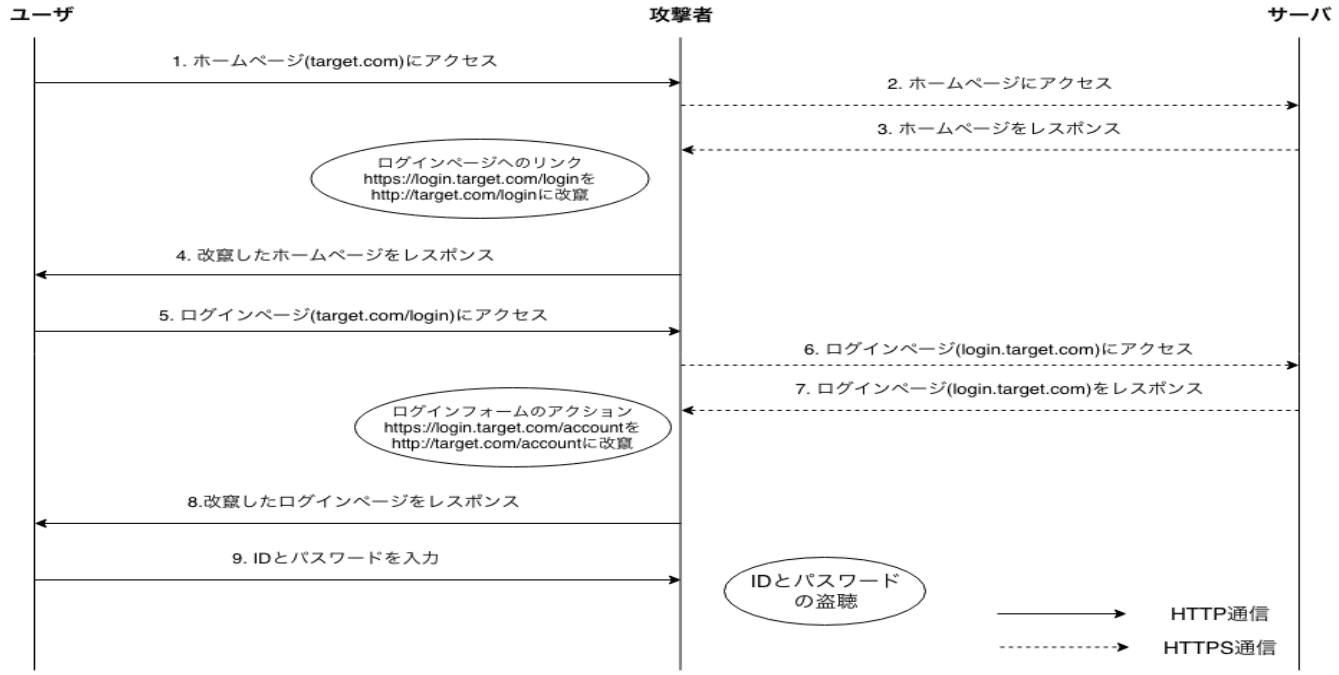

図 2 瀬戸崎-松尾攻撃

しないものについてはカテゴリー分けしていない。項目は左から順にトップページの接続方式が HTTPS と HTTP のいずれか、HTTPS であった場合は HSTS が有効になっているか、ログインページの接続方式が HTTPS か HTTP か、HTTPS であった場合は HSTS が有効になっているかについて表している。Percentage の行では各カテゴリーの全体に対する割合（小数点以下四捨五入）及び成立する攻撃手法を表している。

トップページとログインページ双方に ssl が施されていないドメインの割合は全体の 4%であり、この場合は sslstrip 攻撃する必要なく攻撃者は通信内容を入手することが出来るため非常に危険である。次に、トップページに ssl が施されておらずログインページには ssl が施されているが HSTS は施されていないドメインは従来の sslstrip 攻撃が成立する対象となるが、このカテゴリーは全体のおよそ 9%ほどであった。また、トップページに ssl が施されておらずログインページには ssl が施されていて HSTS も施されているドメインは瀬戸崎-松尾攻撃が成立する対象となるが、このカテゴリーのドメインの割合は今回の調査範囲では全体のおよそ 1%ほどしか存在しなかった。このため、従来の sslstrip 攻撃と比べて影響度が格段に増えたわけではないことが分かる。

4. sslstrip 攻撃の拡張

本節では、前節の表 1 において 1 節で述べた従来型の sslstrip 攻撃と瀬戸崎-松尾攻撃 [5] が成立する対象とならなかった 86%のドメインに対して重点を置き、sslstrip 攻撃を拡張することで、これらのドメインに対して有効な sslstrip 攻撃手法を新たに示す。

4.1 自己署名証明書を持たせる攻撃

トップページが HTTPS 接続を要求するドメインに対してユーザに HTTP 接続させる方法として、攻撃者自身が自己署名証明書を保持してユーザにサーバ認証をさせることが挙げられる。この攻撃手法の攻撃例を図 3 に示す。

ユーザは HTTPS 接続でサーバのホームページ `target.com` にリクエストを送り、攻撃者はユーザと ssl/tls ハンドシェイクを開始する。このとき、攻撃者は自身が発行した自己署名証明書を `target.com` を発行先にしてユーザに送る。ブラウザはルート証明書を確認できないため、セキュリティ証明書のエラー警告ページがでる。ユーザがこの警告を無視して接続を続けた場合、HTTP 接続で通信を行うことになり、攻撃者はユーザの送信する内容を盗聴することが可能となる。ただし、この攻撃はユーザが警告を無視して接続を続けるという必要があるため、機密情報である ID やパスワードの入力を簡単には行わないことが予想される。また、ドメインに HSTS が施されている場合はブラウザは強制的に HTTPS で接続を試みるため、サーバにリクエストを送信することはできず攻撃は成功しない。警告を出さないための方法としては、攻撃者が発行したルート証明書をユーザ側にインストールさせる必要があり、特定のページにアクセスした場合に攻撃者のルート証明書を自動ダウンロードさせるなどが考えられる。しかし、ダウンロードした後にユーザ自身が証明書のインポートを行う必要があるため、やはり簡単ではないといえる。整理すると、この攻撃はトップページ、ログインページともに ssl が

TopPage	TopPage HSTS	LoginPage	LoginPage HSTS	Percentage	攻撃手法
HTTPS	enabled	HTTPS	enabled	14%	-
HTTPS	enabled	HTTPS	disabled	9%	-
HTTPS	disabled	HTTPS	enabled	6%	-
HTTPS	disabled	HTTPS	disabled	56%	-
HTTP	-	HTTPS	enabled	1%	瀬戸崎-松尾攻撃
HTTP	-	HTTPS	disabled	9%	従来の sslstrip 攻撃
HTTP	-	HTTP	-	4%	攻撃手法必要なし

表 1　調査結果

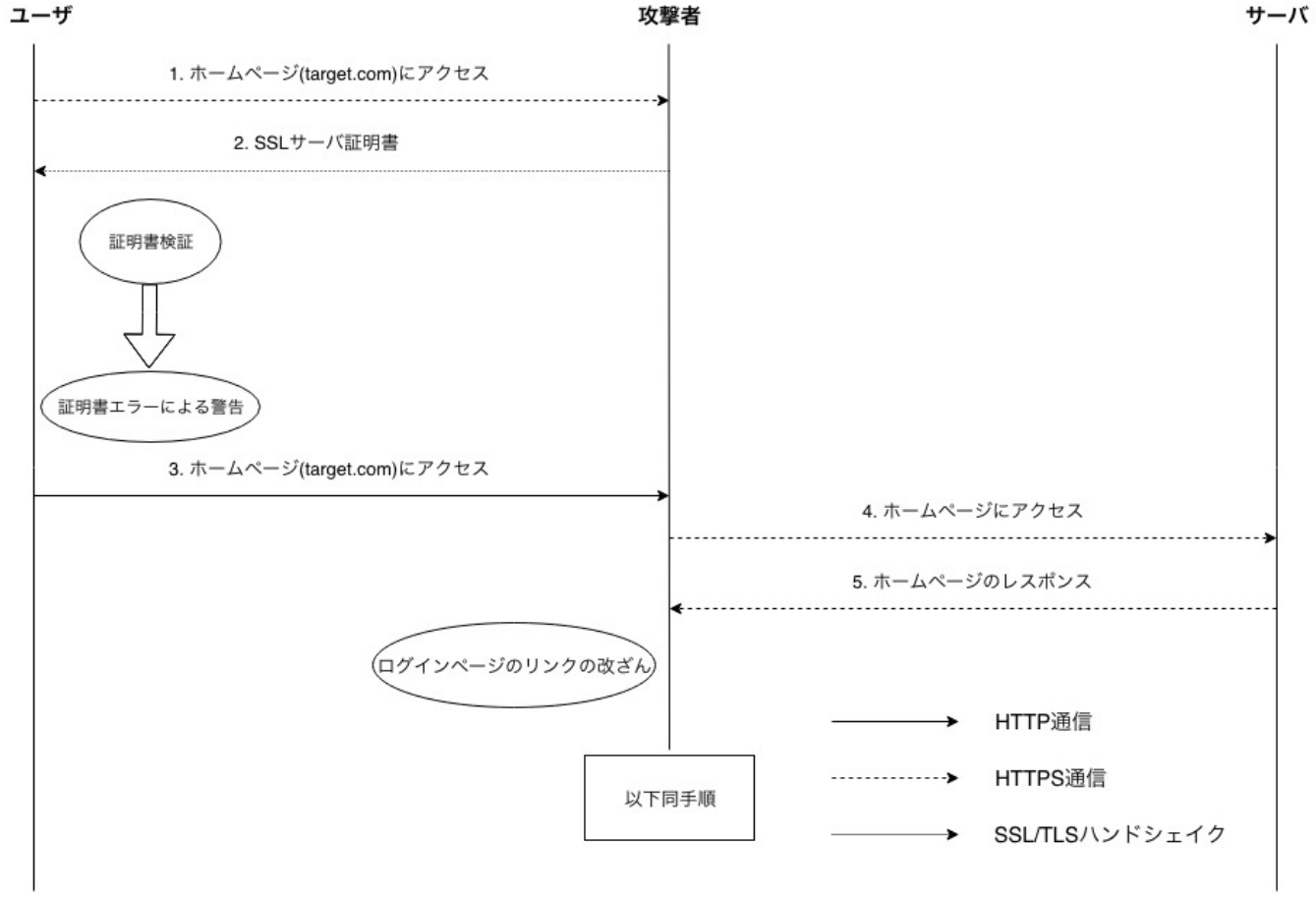

図 3　自己署名証明書を持たせる攻撃

施されているが HSTS は施されていない場合に成立する。表 1 によると 56%のドメインが対象となるが、ユーザに気づかれる可能性も高いといえる。

4.2　HTTP サイトを経由する攻撃

次に、ユーザが他の HTTP サイトを経由してターゲットの Web サイトに遷移する攻撃を考える。具体的には、ユーザが HTTP 接続で検索可能な検索サイトからターゲットサイトにアクセスしたり、攻撃者がフィッシングなどの方法で他の HTTP サイトにユーザーを誘導した上でユーザがターゲットサイトにアクセスする状況を想定している。ターゲットサイトの前にアクセスするドメインを (anyother.com) とする。

まずユーザが HTTP 接続で anyother.com へアクセスする。攻撃者は、そのレスポンス内にあるターゲットサーバへのリンク https://target.com を http://target1.com と書き換える。このとき、リンクの遷移先のドメインは target1.com であるため、target.com に HSTS が施されていたとしてもブラウザ側で HSTS が作動することなく、ユーザがリンクをクリックした際に HTTP 接続で target1.com にアクセスされる。攻撃者は接続先を target.com に変更して HTTPS 接続でサーバと通信し、レスポンス内のログインページへのリンクを https://login.target.com/login を http://target1.com/login に変更する。そして、ユー

ザがそのリンクをクリックした場合、攻撃者は接続先を login.target.com/login に変更して HTTPS 接続でサーバと通信し、レスポンス内のログインフォームのアクション https://login.target.com/account を http://target1.com/account に変更する。ユーザが ID とパスワードを送信した場合、攻撃者はその内容を盗聴することが可能となる。この一連の攻撃の流れを図 4 に示す。

この攻撃ではユーザに一旦他の Web サイトを経由させたうえで、ターゲットサイトへのリンクのクリックを待つという制約がある。しかしながら、実際の攻撃では標的となるサイトを一つに絞らず複数のリンクを http に書き換えることで、攻撃の成功率を上げることが出来ると考えられる。また、攻撃者がすり替えたドメインのサーバ証明書を予め取得することで、ユーザと攻撃者間の通信を HTTP ではなく HTTPS 接続にすることも可能である。この場合、HTTP 接続する場合と比べ、ユーザがより攻撃に気づきにくくなると考えられる。この攻撃では、ログインページへのリンクが存在する全てのドメインが攻撃対象になる。すなわち、ほぼ 100%のサイトが攻撃対象となり、ユーザがドメインのすり替えに気づかない限りは攻撃に成功する。

5. 対策と考察

本節では前節で述べた二つの拡張版 sslstrip 攻撃の考察として、潜在的な脅威、対策および制約について述べる。

5.1 潜在的な脅威シナリオ

本稿で示した拡張版攻撃は HSTS に対しても動作することから、ドメインがすり替えられていることにユーザが気が付かない限り、攻撃が成功する。このため、影響度の直観として、ドメインのすり替えに気が付くかどうかが重要な争点となる。一方、ユーザがドメインのすり替えに気が付かない可能性であるが、極めて高いことが予想される。

その根拠として、Thompson ら [8] による EV-SSL 証明書の影響調査が挙げられる。Thompson らによると、ドメインそのものの正しさを保証する EV-SSL 証明書の表示とドメインの実在性を保証しない DV-SSL 証明書の利用において、ユーザのふるまいは変わらないことが示されている。すなわち、大多数のユーザはドメインのすり替えを意識していない。この事実を受け、Google Chrome および Firefox では EV-SSL 証明書の表示と DV-SSL 証明書が同様に扱われるようになることが公表されている[*2]。このため、ドメインのすり替えに気が付かない可能性は今後さらに高まることもあり得る。

また、個々のウェブサイトの構成に応じても、攻撃発生の事実を見落とすことは起こりえる。例えば入力フォームが複数のスタイルシートに分割されているような構成では、攻撃者はそのすべてのシートに対しドメインのすり替え含めた攻撃を行う必要がある。すなわち、一枚のスタイルシートに集約されているような状況では、攻撃者にとってドメインのすり替えが相対的に容易になる。

5.2 対策

5.2.1 ユーザ側の対策

ユーザ側の対策としてはブラウザの機能に寄るところが大きい。本稿の拡張版含め sslstrip 攻撃はユーザからの HTTP でのアクセスを前提にしているため、ブラウザの拡張機能として全ての接続を HTTPS に切り替えるプラグインの導入が挙げられる。具体的には、HTTPS Everywhere[*3]や ForceTLS[*4]がある。また、ユーザ自身の行動により、sslstrip 攻撃を行う余地を削減することも可能である。具体的には、頻繁にアクセスするウェブサイトであればブラウザのブックマークに登録することである。あらかじめブラウザに格納された URL を直接呼び出すことで、攻撃者によるすり替えの機会を削減することが可能となる。

5.2.2 サーバ側の対策

期待できる対策の一つは、HPKP [9] の利用である。HPKP はユーザ側のブラウザ内において、証明書をドメインごとに予め関連付け（ピン止め）ておくことで、ドメインのすり替えを検出するプロトコルである。HPKP を導入する際は、ドメイン側が自らのピン止めポリシーに基づき、ブラウザ側に証明書の情報を送る必要がある。HPKP は、HSTS と同様 sslstrip 攻撃に対して有効であることが Li ら [4] により指摘されている。本稿では HSTS に対する攻撃が示されたが、HPKP は対策として依然有効である。これはブラウザ側で予めドメインと証明書の紐づけによるところが大きい。なお、Santos ら [10] によると、HPKP の実装は HSTS と比べて進展が大幅に遅れており、2019 年 3 月時点では Chrome、Internet Explore、Safari などシェアの大きなブラウザにおいても HPKP は実装がされていない [11]。HPKP の実装普及は社会的な課題といえる。

サーバ側での導入が期待されるもう一つの対策は maTLS [12] である。maTLS はプロキシなど通信経路上に中間者が存在する状況において、ユーザと中間者間、中間者とサーバ間がそれぞれ TLS セッションを張りながら、ユーザとサーバ間の互いの透明性も確保する技術である。プロキシを sslstrip における中間者として見立てたとき、maTLS で解決している本質的な課題は sslstrip 攻撃の対策としても有効といえる。特に maTLS では中間者攻撃に対する安全性も形式手法で示していることから、sslstrip 攻撃対策としても高い効果が期待できる。

[*2] Extended Validation Certificates are (Really, Really) Dead. https://www.troyhunt.com/extended-validation-certificates-are-really-really-dead/

[*3] HTTPS Everywhere. https://www.eff.org/https-everywhere

[*4] ForceTLS.http://sidstamm.com/forcetls/

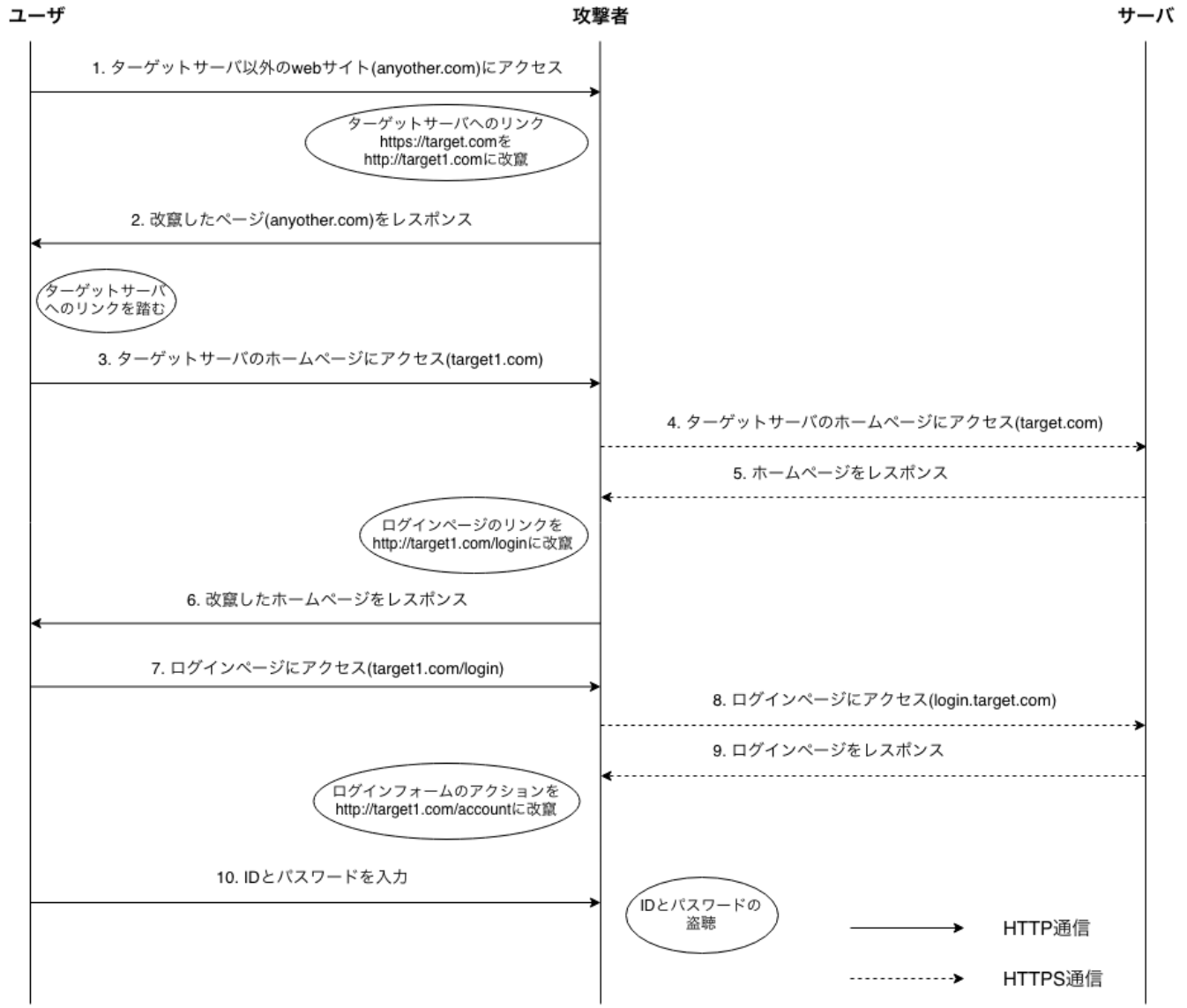

図 4　HTTP サイトを経由する攻撃

5.3　制約

拡張版 sslstrip 攻撃において改善が必要な点を、本稿における制約として以下に記載する。まず、攻撃時にてドメインをすり替えた際、広告などウェブサイト上の情報として読み込めないパーツがあることに起因して、実際のドメイン上のものと比較して表示が崩れるサイトが何点か見受けられた。このような状況では、攻撃の事実についてユーザに気づかれる可能性が高い。ただし、これは広告をブロックしている場合に緩和できる。次に、入力フォームが一枚のスタイルシートで閉じているようなウェブサイトは少数であった。このため攻撃者の観点では、攻撃対象が持つ入力フォームをすべて揃えた状態で行う必要がある。これにより、現状としては、攻撃に関する難易度が相対的に高いといえる。各ドメインのさらなる実態調査含め、これらの制約に関する詳細な検討は今後の課題である。

6.　関連研究

sslstrip 攻撃の現状：Marlinspike ら [1] に sslstrip 攻撃が示されたのち、多くの拡張攻撃 [4], [5], [13] が示されている。著者らの知る限り、もっとも強力な攻撃が瀬戸崎-松尾攻撃である。瀬戸崎-松尾攻撃が 14%のドメインに有効であることに対し、本稿の攻撃は調査範囲におけるすべてのドメインで攻撃が成立する。

HTTPS のセキュリティ：HTTPS のセキュリティは SSL/TLS の実装に依存するが、クライアントの実装には屡々不備がありセキュリティ上の問題が指摘されている [14], [15], [16]。また、鍵管理や証明書の有効性調査も行われている [17], [18]。近年では HTTPS のセキュリティを改善する Everest Project [19] として、実装の正しさを検証したライブラリ開発 [20], [21], [22] も盛んであり、Internet Explorer などのブラウザにも導入が進んでいる。

その他、SplitTLS：sslstrip に類似した研究として、SplitTLS [23] と呼ばれる、プロキシのような中間者の存在を前提とした TLS の研究がある。SplitTLS の研究においては、攻撃者が中間者を不正に利用することで HTTPS のセキュリティに影響を与えられること [18] や、実際にどの程度不正な証明書が発生しているか調査 [24] がされている。この SplitTLS において安全性を理論的に保証可能な技術が maTLS [12] である。

7. まとめ

本稿では sslstrip 攻撃の脅威調査として、JP ドメインで実装されている Web サイトのうちトップページ内にログインページへのリンクが存在するドメインを対象に、ssl と HSTS がそれぞれ施されているかという観点でカテゴリー分けを行った。また調査結果を踏まえ、既存の sslstrip 攻撃が成立しないカテゴリーでも成功する攻撃を二つ提案した。いずれもユーザに気づかれない必要があるという制約がつくが、実際にすべての JP ドメインに対し提案手法による攻撃が成立することを確認した。今後は、ユーザがどれほど攻撃に気付けるかという観点で実験を行い、攻撃の成功率を確かめる予定である。また、対策方法の実装および検討も今後の課題である。

謝辞 本研究の一部は JSPS 科研費 18K18049 およびセコム財団挑戦的研究助成の助成を受けたものです。

参考文献

[1] Moxie Marlinspike. More tricks for defeating ssl in practice. *Black Hat USA 2009*, 2009.

[2] Jeff Hodges, Collin Jackson, and Adam Barth. Http strict transport security (hsts). *http://tools.ietf.org/html/rfc6797*, 2010.

[3] Michael Kranch and Joseph Bonneau. Upgrading https in midair: Hsts and key pinning in practice. In *Proc. of NDSS 2015*. Internet Society, 2015.

[4] Xurong Li, Chunming Wu, Shouling Ji, Qinchen Gu, editor="Lin Xiaodong Beyah, Raheem", Ali Ghorbani, Kui Ren, Sencun Zhu, and Aiqing Zhang. Hsts measurement and an enhanced stripping attack against https. In *Proc. of SecureComm 2017*, Vol. 238 of *LNICST*, pp. 489–509. Springer, 2017.

[5] 瀬戸崎喬, 松尾和人. Hsts による対策を回避可能な sslstrip 攻撃. CSS2016 論文集, 第 2016 巻, pp. 733–740, oct 2016.

[6] Jeff Hodges, Collin Jackson, and Adam Barth. HTTP Strict Transport Security (HSTS). RFC 6797, November 2012.

[7] Victor Le Pochat, Tom van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Proc. of NDSS 2019*, 2019.

[8] Christopher Thompson, Martin Shelton, Emily Stark, Maximilian Walker, Emily Schechter, and Adrienne Porter Felt. The web's identity crisis: Understanding the effectiveness of website identity indicators. In *Proc. of USENIX Security 2019*, pp. 1715–1732. USENIX Association, 2019.

[9] C Evans, C Palmer, and R Sleevi. Public key pinning extension for http (hpkp). *https://tools.ietf.org/html/rfc7469*, 2015.

[10] Sergio De los Santos and José Torres. Analysing hsts and hpkp implementation in both browsers and servers. *IET Information Security*, Vol. 12, No. 4, pp. 275–284, 2017.

[11] MDN web docs. Http public key pinning (hpkp). *https://developer.mozilla.org/ja/docs/Web/Security/Public_Key_Pinning*, 2019.

[12] Hyunwoo Lee, Zach Smith, Junghwan Lim, Gyeongjae Choi, Selin Chun, Taejoong Chung, and Ted Taekyoung Kwon. matls: How to make tls middlebox-aware? In *Proc. of NDSS*. Internet Society, 2019.

[13] Sendong Zhao, Wu Yang, Ding Wang, and Wenzhen Qiu. A new scheme with secure cookie against sslstrip attack. In *Proc. of WISM 2012*, Vol. 7529 of *LNCS*, pp. 214–221. Springer, 2012.

[14] Shuo Chen, Ziqing Mao, Yi-Min Wang, and Ming Zhang. Pretty-bad-proxy: An overlooked adversary in browsers' https deployments. In *Proc. of IEEE S&P 2009*, pp. 347–359. IEEE, 2009.

[15] X de Carné de Carnavalet and Mohammad Mannan. Killed by proxy: Analyzing client-end tls interception software. In *Proc. of NDSS 2016*. Internet Society, 2016.

[16] Martin Georgiev, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, and Vitaly Shmatikov. The most dangerous code in the world: validating ssl certificates in non-browser software. In *Proc. of CCS 2012*, pp. 38–49. ACM, 2012.

[17] Pawel Szalachowski, Stephanos Matsumoto, and Adrian Perrig. Policert: Secure and flexible tls certificate management. In *Proc. of CCS 2014*, pp. 406–417. ACM, 2014.

[18] Frank Cangialosi, Taejoong Chung, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, and Christo Wilson. Measurement and analysis of private key sharing in the https ecosystem. In *Proc. of CCS 2016*, pp. 628–640. ACM, 2016.

[19] Karthikeyan Bhargavan, Barry Bond, Antoine Delignat-Lavaud, Cédric Fournet, Chris Hawblitzel, Catalin Hritcu, Samin Ishtiaq, Markulf Kohlweiss, Rustan Leino, Jay Lorch, et al. Everest: Towards a verified, drop-in replacement of https. In *Proc. of SNAPL 2017*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[20] Jean-Karim Zinzindohoué, Karthikeyan Bhargavan, Jonathan Protzenko, and Benjamin Beurdouche. Hacl*: A verified modern cryptographic library. In *Proc. of CCS 2017*, pp. 1789–1806. ACM, 2017.

[21] Barry Bond, Chris Hawblitzel, Manos Kapritsos, K Rustan M Leino, Jacob R Lorch, Bryan Parno, Ashay Rane, Srinath Setty, and Laure Thompson. Vale: Verifying high-performance cryptographic assembly code. In *Proc. of USENIX Security 2017*, pp. 917–934. USENIX Association, 2017.

[22] Tahina Ramananandro, Antoine Delignat-Lavaud, Cédric Fournet, Nikhil Swamy, Tej Chajed, Nadim Kobeissi, and Jonathan Protzenko. Everparse: Verified secure zero-copy parsers for authenticated message formats. In *Proc. of USENIX Security 2019*, pp. 1465–1482. USENIX Association, 2019.

[23] Jeff Jarmoc and DSCT Unit. Ssl/tls interception proxies and transitive trust. *Black Hat Europe*, 2012.

[24] Lin Shung Huang, Alex Rice, Erling Ellingsen, and Collin Jackson. Analyzing forged ssl certificates in the wild. In *Proc. of IEEE S&P 2014*, pp. 83–97. IEEE, 2014.