(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5640752号
(P5640752)
(45)発行日　平成26年12月17日(2014.12.17)　(24)登録日　平成26年11月7日(2014.11.7)

(51)Int.Cl.　　F I
G06F 21/57 (2013.01)　　G06F 21/00 157D
G06F 11/28 (2006.01)　　G06F 11/28 340A

請求項の数 6　(全 16 頁)

(21)出願番号　特願2011-3477(P2011-3477)
(22)出願日　平成23年1月11日(2011.1.11)
(65)公開番号　特開2012-146100(P2012-146100A)
(43)公開日　平成24年8月2日(2012.8.2)
審査請求日　平成25年10月7日(2013.10.7)

(73)特許権者　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100087480
弁理士　片山　修平
(72)発明者　森川　郁也
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

審査官　岸野　徹

最終頁に続く

(54)【発明の名称】攻撃模倣テスト方法、攻撃模倣テスト装置及び攻撃模倣テストプログラム

(57)【特許請求の範囲】
【請求項１】
サーバへの接続が可能な第１のブラウザが前記サーバに対するアクセスを行った際に当該第１のブラウザが被害を被る可能性のある状態にするための情報である要素データを、前記サーバへの接続が可能な第２のブラウザが表示中のＨＴＭＬページから取得する取得工程と、
前記取得工程で前記第２のブラウザが取得した前記要素データを、前記第２のブラウザが前記第１のブラウザに送信する送信工程と、
前記送信工程で送信された要素データを前記第１のブラウザが受信する受信工程と、
前記第１のブラウザが、前記第１のブラウザの状態に、前記受信工程で受信された前記要素データを反映させて、当該第１のブラウザの状態を、前記サーバにアクセスした場合に被害を被る可能性のある状態に設定する設定工程と、
を含む攻撃模倣テスト方法。
【請求項２】
前記要素データは、フォームであり、
前記第１のブラウザが被る被害は、クロスサイトリクエストフォージェリー攻撃による被害であることを特徴とする請求項１に記載の攻撃模倣テスト方法。
【請求項３】
前記要素データは、クッキーであり、
前記第１のブラウザが被る被害は、セッションフィクセーション攻撃による被害である

ことを特徴とする請求項１に記載の攻撃模倣テスト方法。
【請求項４】
前記送信工程及び前記受信工程では、前記要素データを、ＩＰネットワークを用いた受け渡し、クリップボードを用いた受け渡し、ファイルを用いた受け渡し、のいずれかにより、前記第２のブラウザと前記第１のブラウザとの間の送受信を行うことを特徴とする請求項１～３のいずれか一項に記載の攻撃模倣テスト方法。
【請求項５】
サーバへの接続が可能な第１のブラウザと、
前記第１のブラウザが前記サーバに対するアクセスを行った際に、当該第１のブラウザが被害を被る可能性のある状態にするための情報である要素データを表示中のＨＴＭＬページから取得する、前記サーバへの接続が可能な第２のブラウザと、を備え、
前記第２のブラウザは、取得した前記要素データを前記第１のブラウザに送信する送信部を有し、
前記第１のブラウザは、前記要素データを受信する受信部と、当該第１のブラウザの状態に前記受信部で受信した前記要素データを反映させて、前記第１のブラウザの状態を前記サーバにアクセスした場合に被害を被る可能性のある状態に設定する設定部と、を有することを特徴とする攻撃模倣テスト装置。
【請求項６】
サーバへの接続が可能な第１のブラウザが前記サーバに対するアクセスを行った際に、当該第１のブラウザが被害を被る可能性のある状態にするための情報である要素データを、前記サーバへの接続が可能な第２のブラウザが表示中のＨＴＭＬページから取得し、
前記取得する処理で前記第２のブラウザが取得した前記要素データを前記第２のブラウザから前記第１のブラウザに送信し、
前記送信する処理で送信された前記要素データを前記第１のブラウザで受信し、
前記第１のブラウザの状態に、前記受信する処理で受信された前記要素データを反映させて、当該第１のブラウザの状態を、前記サーバにアクセスした場合に被害を被る可能性のある状態に設定する処理を、コンピュータに実行させることを特徴とする攻撃模倣テストプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本件は、攻撃模倣テスト方法、攻撃模倣テスト装置及び攻撃模倣テストプログラムに関する。
【背景技術】
【０００２】
最近、Ｗｅｂブラウザの制御機能を狙った攻撃として、「クロスサイトリクエスト偽造」（cross-site request forgery：以下、「ＣＳＲＦ」と呼ぶ)や、セッションフィクセーション（session fixation）といった攻撃がある（例えば、特許文献１参照）。これらの攻撃は、被害者側ブラウザが、攻撃者にとって都合のよいリクエストをサーバへ送信し、サーバがそれを受け入れてしまうことで実現される攻撃である。したがって、このような攻撃を防ぐためには、例えば、サーバが攻撃者にとって都合のよいリクエストを受け取っても、それを受け入れないようにしておくことが必要となる。
【０００３】
例えば、ＣＳＲＦ攻撃を防ぐ方法には、攻撃者が被害者側ブラウザへ働きかけることでは被害者側ブラウザからサーバに送信され難い値をリクエストに含めておく方法がある。この場合、サーバは、リクエストの正誤によって異常なリクエストを検知する。なお、被害者側ブラウザからサーバに送信され難い値とは、フォーム内のhidden型フィールドやReferererヘッダー値などである。
【０００４】
また、セッションフィクセーション攻撃を防ぐ方法には、サーバがログインなどの重大

なリクエストに際して常にセッションを作り直すようにする方法がある。この場合、セッションクッキーが書き換えられた状態で重大なリクエストが送信されたときには、攻撃者のセッションに権限が付与されないようにすることができる。

【０００５】

このように、攻撃ごとに防御する方法が異なることから、サーバには、様々な攻撃を防ぐための機能を設けておくことが望ましい。しかるに、たとえば、サーバにおいて、対策していない箇所があったり、対策方法が間違っていたりする可能性もあるため、Ｗｅｂアプリケーション開発者は、正しい対策が行われているかについてテストを実行する必要がある。

【０００６】

テストの代表的な方法には、実際に攻撃が行われるのと同様の方法を模倣（シミュレート）して、攻撃が失敗したら正しく対策されているとする方法がある。このようなテストはブラックボックステスト方式と呼ばれ、Ｗｅｂアプリケーションの外部からＨＴＴＰプロトコルによって通信する方式で行われるのが一般的である。また、一般的なブラックボックステストについては、自動化が進んでいる一方、ＣＳＲＦ攻撃やセッションフィクセーション攻撃に関するテストを自動的に行うことは難しい。これらの攻撃を模倣してテストするには、アプリケーションに依存した適切な処理が必要であり、アプリケーションに関する知識・仕様情報をもたない装置やプログラムでは実現が難しいからである。

【０００７】

このため、ＣＳＲＦ攻撃やセッションフィクセーション攻撃への対策に関するテストは、人間が適切な状態・方法や攻撃の成否を判断しながら対話的に行うのが有力と考えられる。例えば、それぞれが攻撃者側と被害者側の役割を持つ二つ（又はそれ以上）のクライアント（一般的にはＷｅｂブラウザが仮想的にクライアントとして動作する）を実行し、その両方を操作しながら行う方法である。

【先行技術文献】

【特許文献】

【０００８】

【特許文献１】特開２００９－３０１３２７号公報

【発明の概要】

【発明が解決しようとする課題】

【０００９】

しかしながら、従来のＷｅｂクライアント、Ｗｅｂブラウザには対話的にテストを行うための機能が用意されていない。また、仮に全て人手で対話的な操作を行おうとすると、大変手間がかかる。このことは、同じテストを何度も繰り返す場合に、特に問題となる。

【００１０】

そこで本件は上記の課題に鑑みてなされたものであり、攻撃に対するテストを簡易に行うことが可能な攻撃模倣テスト方法、攻撃模倣テスト装置及び攻撃模倣テストプログラムを提供することを目的とする。

【課題を解決するための手段】

【００１１】

本明細書に記載の攻撃模倣テスト方法は、サーバへの接続が可能な第１のブラウザが前記サーバに対するアクセスを行った際に当該第１のブラウザが被害を被る可能性のある状態にするための情報である要素データを、前記サーバへの接続が可能な第２のブラウザが表示中のＨＴＭＬページから取得する取得工程と、前記取得工程で前記第２のブラウザが取得した前記要素データを、前記第２のブラウザが前記第１のブラウザに送信する送信工程と、前記送信工程で送信された要素データを前記第１のブラウザが受信する受信工程と、前記第１のブラウザが、前記第１のブラウザの状態に、前記受信工程で受信された前記要素データを反映させて、当該第１のブラウザの状態を、前記サーバにアクセスした場合に被害を被る可能性のある状態に設定する設定工程と、を含んでいる。

【００１２】

本明細書に記載の攻撃模倣テスト装置は、サーバへの接続が可能な第１のブラウザと、前記第１のブラウザが前記サーバに対するアクセスを行った際に、当該第１のブラウザが被害を被る可能性のある状態にするための情報である要素データを表示中のＨＴＭＬページから取得する、前記サーバへの接続が可能な第２のブラウザと、を備え、前記第２のブラウザは、取得した前記要素データを前記第１のブラウザに送信する送信部を有し、前記第１のブラウザは、前記要素データを受信する受信部と、当該第１のブラウザの状態に前記受信部で受信した前記要素データを反映させて、前記第１のブラウザの状態を前記サーバにアクセスした場合に被害を被る可能性のある状態に設定する設定部と、を有する。

【００１３】

本明細書に記載の攻撃模倣テストプログラムは、サーバへの接続が可能な第１のブラウザが前記サーバに対するアクセスを行った際に、当該第１のブラウザが被害を被る可能性のある状態にするための情報である要素データを、前記サーバへの接続が可能な第２のブラウザが表示中のＨＴＭＬページから取得し、前記取得する処理で前記第２のブラウザが取得した前記要素データを前記第２のブラウザから前記第１のブラウザに送信し、前記送信する処理で送信された前記要素データを前記第１のブラウザで受信し、前記第１のブラウザの状態に、前記受信する処理で受信された前記要素データを反映させて、当該第１のブラウザの状態を、前記サーバにアクセスした場合に被害を被る可能性のある状態に設定する処理を、コンピュータに実行させる。

【発明の効果】

【００１４】

本明細書に記載の攻撃模倣テスト方法、攻撃模倣テスト装置及び攻撃模倣テストプログラムは、攻撃に対するテストを簡易に行うことができるという効果を奏する。

【図面の簡単な説明】

【００１５】

【図１】一実施形態に係る攻撃模倣テスト装置を含むシステムの構成を概略的に示す図である。

【図２】図１の情報処理部のハードウェア構成を示す図である。

【図３】図１の情報処理部の機能ブロック図である。

【図４】図４（ａ）は、攻撃者側ブラウザＡの処理を示すフローチャートであり、図４（ｂ）は、被害者側ブラウザＢの処理を示すフローチャートである。

【図５】要素データ（フォーム）を示す図である。

【図６】図６（ａ）は、要素データがフォームである場合における、図４（ａ）のステップＳ１２の処理を示すフローチャートであり、図６（ｂ）は、要素データがフォームである場合における、図４（ｂ）のステップＳ２４の処理を示すフローチャートである。

【図７】要素データ（クッキー）を示す図である。

【図８】図８（ａ）は、要素データがクッキーである場合における、図４（ａ）のステップＳ１２の処理を示すフローチャートであり、図８（ｂ）は、要素データがクッキーである場合における、図４（ｂ）のステップＳ２４の処理を示すフローチャートである。

【図９】変形例を示す図である。

【発明を実施するための形態】

【００１６】

以下、一実施形態について、図１～図８に基づいて詳細に説明する。図１には、一実施形態にかかる攻撃模倣テスト方法を実現可能な攻撃模倣テスト装置１０を含むシステムの構成が概略的に示されている。図１に示すように、攻撃模倣テスト装置１０は、インターネット（ＩＰネットワーク）などのネットワーク１２に接続されている。また、ネットワーク１２には、テスト対象のサーバ１４（Ｗｅｂサイトとも呼ばれる）も接続されている。

【００１７】

攻撃模倣テスト装置１０は、汎用的なＰＣ（Personal Computer）などの端末であり、情報処理部２０と、ユーザインタフェース２２（キーボードやマウス、表示装置など）と、を備える。図２には、情報処理部２０のハードウェア構成が示されている。図２に示すように、情報処理部２０は、ＣＰＵ９０、ＲＯＭ９２、ＲＡＭ９４、記憶部（ここではＨＤＤ（Hard Disk Drive））９６、及び可搬型記憶媒体用ドライブ９９等を備えており、情報処理部２０の構成各部は、バス９８に接続されている。情報処理部２０では、ＲＯＭ９２あるいはＨＤＤ９６に格納されているプログラム（攻撃模倣テストプログラム）、或いは可搬型記憶媒体用ドライブ９９が可搬型記憶媒体９１から読み取ったプログラム（攻撃模倣テストプログラム）をＣＰＵ９０が実行することにより、図３の各部の機能が実現される。

【００１８】

図３には、情報処理部２０の機能ブロック図が示されている。この図３に示すように、情報処理部２０では、二つ以上（図３では２つ）のＷｅｂクライアント（ブラウザ（クライアント）Ａ及びブラウザ（クライアント）Ｂ）が動作する。なお、各ブラウザＡ，Ｂはそれぞれが汎用的なオペレーティングシステム（ＯＳ）上で動作するアプリケーションプログラムであり、仮想的にクライアントとして動作するものとする。また、本実施形態では、各Ｗｅｂクライアントソフトウェアが攻撃者側と被害者側の両方の機能を兼ね備えているものとする。なお、各ブラウザは、それぞれ攻撃者側又は被害者側として動作することで、役割分担を行っている。

【００１９】

ブラウザＡは、図３に示すように、通常処理部３２と、要素データ抽出部３４と、送信部としての要素データ送信部３６と、受信部としての要素データ受信部３８と、設定部としての要求可能状態設定部４０と、しての機能を有する。これらのうち、要素データ抽出部３４及び要素データ送信部３６は、ブラウザＡが、攻撃者側（第２のブラウザ）となる場合に動作する。一方、要素データ受信部３８及び要求可能状態設定部４０は、ブラウザＡが、被害者側（第１のブラウザ）となる場合に動作する。通常処理部３２は、一般的なブラウザの処理を実行する。

ブラウザＢも、ブラウザＡと同様、通常処理部４２と、要素データ抽出部４４と、送信部としての要素データ送信部４６と、受信部としての要素データ受信部４８と、設定部としての要求可能状態設定部５０と、しての機能を有する。これらのうち、要素データ抽出部４４及び要素データ送信部４６は、ブラウザＢが、攻撃者側（第２のブラウザ）となる場合に動作する。一方、要素データ受信部４８及び要求可能状態設定部５０は、ブラウザＢが被害者側（第１のブラウザ）となる場合に動作する。

【００２０】

要素データ抽出部３４は、他方のブラウザ（ブラウザＢ）がサーバ１４に対するアクセスを行った際に当該ブラウザが被害を被る可能性のある状態にするための情報である要素データを抽出（取得）する。同様に、要素データ抽出部４４は、他方のブラウザ（ブラウザＡ）がサーバ１４に対するアクセスを行った際に当該ブラウザが被害を被る可能性のある状態にするための情報である要素データを抽出（取得）する。要素データとしては、例えば、フォームや、クッキーが挙げられる。

【００２１】

要素データ送信部３６，４６は、要素データ抽出部３４，４４が取得した要素データを他方のブラウザ（要素データ抽出部３４の場合ブラウザＢ、要素データ抽出部４４の場合ブラウザＡ）に対して送信する。要素データ受信部３８，４８は、要素データ送信部４６，３６から送信されてきた要素データを受信する。

【００２２】

要求可能状態設定部４０，５０は、要素データ受信部３８，４８が受信した要素データを、ブラウザの状態に反映させて、当該ブラウザの状態をサーバ１４にアクセスした場合に被害を被る可能性のある状態に設定する。

【００２３】

ここで、ＣＳＲＦ攻撃と、セッションフィクセーション攻撃について、具体的に説明する。

【００２４】

最近のｗｅｂ（World Wide Web）では、掲示板やウェブログ（ブログ）のような発信情報の更新処理、ネットワークに接続された機器の管理、商品やサービスの購入といった商取引など、多くのアプリケーションが実現されている。このような高度なＷｅｂアプリケーションでは、特定のユーザのみに情報を提供したり操作を許可したりすることが求められる。このため、ＳＳＬ（Secure Sockets Layer）やＴＬＳ（Transport Layer Security）による保護を行うＨＴＴＰＳプロトコルや、ＨＴＴＰ認証といったセキュリティ機能が出現してきている。またＨＴＴＰプロトコルはステートレスであり、プロトコル自体は複数のリクエストを関連付ける機能を持たないため、クッキーと呼ばれる技術も出現してきている。これはサーバがレスポンスに付与した値（クッキー）を、ブラウザが暗黙的に記憶し、ブラウザは、それ以降リクエストのたびにその値を付けてサーバに送信するというものである。

【００２５】

一方で、Ｗｅｂブラウザには、その振舞いをサーバ側の指定で制御できる機能が設けられるようになった。その一つであるクライアント側スクリプトはＨＴＭＬページ内やそこから参照されるファイル内に記述できる簡易なプログラム言語であり、Ｗｅｂブラウザの振舞いをある程度制御することができる（例えば、Java（登録商標）Script）。ただし、これらのＷｅｂブラウザ制御機能は、ユーザや開発者に機能性や利便性をもたらすのみならず、セキュリティ侵害を起こそうとする者にとっても好都合である。

【００２６】

すなわち、攻撃者は、細工されたＨＴＭＬページを任意のサイトに用意し、ユーザにそのページを開かせることにより、そのページに用意されたJava（登録商標）Scriptやその他の要素によってＷｅｂブラウザを操作するようにする。このような手段で、被害者側のブラウザに、被害者が意図しないリクエストを送信させることにより引き起こされる攻撃がＣＳＲＦ攻撃である。このＣＳＲＦ攻撃は、特に、リクエストが重要な処理、たとえば掲示板などへのメッセージの書き込みや削除、機器への設定変更処理、商品購入やサービスの申し込みなどである場合に、問題となる。ＣＳＲＦ攻撃ではリクエストをサーバへ送信するのは被害者側ブラウザである。このため、ブラウザが記憶しているクッキーによって依頼者を識別している場合や、ブラウザが稼動しているコンピュータの識別子（たとえばＩＰアドレス）によってアクセス可否を判断している場合などが多い。この場合、本来、攻撃者自身が送信するリクエストでは許可されない処理が、サーバで受け付けられるため、深刻な被害となる。

【００２７】

また、リンクやフォームのような単純なハイパーテキスト部品であっても、ユーザはその送信先や送信内容を必ずしも確認しないので、ユーザが意図しない送信先へ意図しない内容のリクエストを送らせる行為は容易に実現できる。このような手段で、被害者のブラウザが記憶しているセッションクッキー情報を強制的に書き換えておくことで、被害者のリクエストを攻撃者にとって都合の良いセッションに属するものとサーバに識別させる攻撃手法が、セッションフィクセーション攻撃である。この攻撃では、クッキーが書き換えられた状態に気づかずに被害者がリクエストを送信すると、攻撃者にとってアクセス可能なセッションに被害者の送信内容やログイン権限などが付与される。これにより、攻撃者は被害者の送信した情報を盗み見たり、被害者になりすまして別の処理を要求したりすることが可能になる。クッキー情報を書き換える手段としては、ブラウザのバグを悪用したり、適用ドメインを広く指定したクッキーを与えたり、通信路上で改ざんしたりする、などの方法が知られている。

【００２８】

本実施形態では、ＣＳＲＦ攻撃や、セッションフィクセーション攻撃に対して、サーバ１４上での対策が有効か否かを検査するためのテスト（ブラックボックステスト）を、攻

撃模倣テスト装置１０を用いて実行する。このテストは、ＣＳＲＦ攻撃やセッションフィクセーション攻撃を模倣して（同一の攻撃を被害者側ブラウザに仕掛けて）、その結果を見るテストである。

【００２９】

以下、図４（ａ）、図４（ｂ）のフローチャートに沿って、その他図面を適宜参照しつつ、攻撃模倣テスト装置１０のブラウザＡとブラウザＢを用いた攻撃模倣テストについて、詳細に説明する。

【００３０】

（ＣＳＲＦ攻撃に関するテスト）

本実施形態では、ブラウザＡが、第２のブラウザとしての攻撃者側ブラウザであるものとし、ブラウザＢが、第１のブラウザとしての被害者側ブラウザであるものとする。すなわち、本処理では、図３の要素データ受信部３８、要求可能状態設定部４０、要素データ抽出部４４、及び要素データ送信部４６は、機能しないものとする。

【００３１】

図４（ａ）のフローチャートは、攻撃者側ブラウザＡの処理を示し、図４（ｂ）のフローチャートは、被害者側ブラウザＢの処理を示している。これらの処理は同時並行的に実行される。まず、図４（ａ）のステップＳ１０では、通常処理部３２が、攻撃者としての準備処理を行う。より具体的には、ユーザ（テスト実施者）の指示の下、通常処理部３２が、ログインや目的のページへの遷移といった準備作業を実行する。また、図４（ｂ）のステップＳ２０では、通常処理部４２が、被害者としての準備処理（ログインや目的のページへの遷移など）を行う。

【００３２】

次いで、図４（ａ）のステップＳ１２では、要素データ抽出部３４が、要素データを抽出する。なお、ステップＳ１２の処理は、ユーザ（テスト実施者）からの指示を受けたタイミングで行われる。ここでは、ＣＳＲＦ攻撃のテストを行うので、要素データ抽出部３４は、要素データとしてフォームを抽出することになる。要素データ（フォーム）の構成は、一例として、図５に示すような構成となっている。図５に示すように、要素データは、項目「method」、「action」、「パラメータ群」から成る。「method」はＨＴＴＰメソッド名であり、formタグのmethod属性から得られる。「action」はフォームの送信先ＵＲＬであり、formタグのaction属性（および同属性が相対ＵＲＬの場合にはフォームを含むページを得たリクエストＵＲＬ）から得られる。「パラメータ群」は一般に複数の下位項目から成り、各下位項目はformタグに含まれるコントロールと呼ばれる要素を表すタグ（inputタグ、selectタグなど）から得ることができ、それぞれが「種類」、「名前」、「値」から成る。

【００３３】

ここで、「種類」はコントロールの種類を指し、例えば、text、password、hidden、textarea、radio、checkbox、selectがある。「名前」はコントロールの名前を指し、name属性から得ることができる。「値」はコントロールに設定されている初期値を指し、value属性から得ることができる。ただし、複数の選択肢を持つコントロール（radio、selectなど）の場合、値は複数であってもよい。

【００３４】

図４（ａ）のステップＳ１２では、具体的に、図６（ａ）のフローチャートに沿った処理を実行する。まず、図６（ａ）のステップＳ３０では、要素データ抽出部３４が、表示中のＨＴＭＬページ内のformタグを選択する。次いで、ステップＳ３２では、要素データ抽出部３４が、ステップＳ３０で選択されたformタグのmethod属性とaction属性を取得する。次いで、ステップＳ３４では、formタグ内の各コントロールについて、種類、名前、値を取り出す。以上のようにして、図５のような要素データが抽出されると、図４（ａ）のステップＳ１４に移行する。

【００３５】

図４（ａ）のステップＳ１４では、要素データ送信部３６が、要素データの送信を行う

。また、これに伴って、図４（ｂ）のステップＳ２２では、要素データ受信部４８が、要素データの受信を行う。なお、本実施形態では、攻撃者側ブラウザＡと被害者側ブラウザＢとが、同一の装置上に存在しているので、要素データの送受信には、一般的なプロセス間通信の手段を用いることができる。ただし、これに限らず、例えば、攻撃者側ブラウザＡと被害者側ブラウザＢとが異なる装置上に存在している場合には、通信機能（例えばＩＰネットワーク）を用いて要素データを送受信することができる。たとえば、ＩＰネットワークを用いる場合であれば、被害者側ブラウザが特定のポート番号で待ち受けし、攻撃者側ブラウザがそのポート番号に接続して、要素データを適当な方式で符号化して送受信することで実現できる。なお、接続するクライアントの識別子、例えば名前やポート番号などは、ユーザ（テスト実施者）が事前に登録しておくか、あるいは、必要になったときにユーザ（テスト実施者）に問い合わせるなどの一般的な方法で定めるものとする。なお、上述した構造の要素データを受け渡す際の中間形式としては、たとえばＸＭＬ文書にするなどの方法がある。

【００３６】

次いで、図４（ｂ）のステップＳ２４では、要求可能状態設定部５０が、要求可能状態設定処理を実行する。すなわち、要求可能状態設定部５０は、被害者側ブラウザＢにおいて、受け取ったフォーム情報をformタグとして反映させたＨＴＭＬページを表示させる。この場合、たとえばJava（登録商標）Scriptを用いて動的にＨＴＭＬページを作成することが可能である。なお、この場合の「Java（登録商標）Script」には、その他の亜種や類似技術であるECMAScript、JScript、VBScriptなども含まれる。図４（ｂ）のステップＳ２４においては、具体的には、図６（ｂ）のフローチャートに沿った処理が実行される。

【００３７】

図６（ｂ）の処理では、まず、ステップＳ４０において、要求可能状態設定部５０が、表示中のＨＴＭＬページを選択する。なお、表示中のＨＴＭＬページが存在しない場合には、要求可能状態設定部５０は、新たな空白のＨＴＭＬページを作成する。

【００３８】

次いで、ステップＳ４２では、要求可能状態設定部５０が、ステップＳ２２で受信した要素データに含まれるmethod属性とaction属性をもつformタグを生成する。次いで、ステップＳ４４では、要求可能状態設定部５０が、各パラメータについてformタグ内に対応する名前、値を持ったコントロールを生成する。なお、フォーム情報に含まれる各パラメータはformタグ内のコントロールとして表現されるが、攻撃者側と異なる種類のコントロールにしても構わない。たとえば、種類がhiddenのパラメータをtext型のコントロールとして表現することで、そのパラメータの値を、送信前にブラウザ上で編集可能になる。したがって、攻撃者側ブラウザＡあるいは被害者側ブラウザＢに、編集可能なパラメータの値を、ユーザ（テスト実施者）の意図に合わせて編集するための機能を設けてもよい。これにより、テスト結果をより簡単に判別できるような内容や、より実際の攻撃に近い内容に書き換えることが可能となる。

【００３９】

以上の処理により、要求可能状態設定部５０は、被害者ブラウザＢの状態を要求可能状態とすることができる。すなわち、被害者ブラウザＢの状態が攻撃者によって細工されたリクエストをサーバへ送信することが可能な状態に設定されることになる（攻撃模倣）。その後は、図４（ｂ）のステップＳ２６に移行する。

【００４０】

ステップＳ２６に移行すると、通常処理部４２は、細工されたリクエストの送信処理を実行する。なお、このステップＳ２６の処理は、ユーザ（テスト実施者）の操作に応じて、すなわち、ユーザがフォーム送信ボタンなどを押したタイミングで、行われる。

【００４１】

そして、次のステップＳ２８では、通常処理部４２が、送信したリクエストに対してサーバ１４から送信されてきたレスポンスを出力（表示）する。この出力をユーザが確認することで、攻撃模倣による結果を確認することができ、これにより、攻撃模倣テストの目

的が達成されることとなる。なお、ユーザは出力されたレスポンスを確認する場合のみならず、例えば、リクエスト送信後のサーバの動作を観察することでも、攻撃模倣による結果を確認することが可能な場合がある。

【００４２】

（セッションフィクセーション攻撃に関するテスト）

次に、セッションフィクセーション攻撃に関するテスト方法について説明する。このセッションフィクセーション攻撃に関するテストの場合、前述したＣＳＲＦ攻撃の場合と異なり、要素データはクッキーの情報の少なくとも一部となる。また、図４（ａ）のステップＳ１２と、図４（ｂ）のステップＳ２４の具体的処理内容が、ＣＳＲＦ攻撃の場合と異なる。

【００４３】

要素データ（クッキー）は、具体的には、図７のような構成を有する。すなわち、図７に示すように、要素データ（クッキー）は、「名前」、「値」、「対象ドメイン名」の項目を有する。「名前」と「値」はそれぞれクッキーの名前と値であり、「対象ドメイン名」はクッキーを送信するドメイン名を指す。なお、要素データ（クッキー）の項目としては、上記に加えて、図７に示すように、「対象パス」、「有効期限」、「Secure属性」などの項目を採用してもよいが、必須の項目ではない。

【００４４】

本テストでは、図４（ａ）のステップＳ１２の処理（要素データ抽出処理）において、図８（ａ）のフローチャートに沿った処理を実行する。図８（ａ）の処理では、まず、ステップＳ５０において、要素データ抽出部３４が、表示中のＨＴＭＬページに属するdocument.cookie（セッションクッキー情報）を取得する。この場合、要素データ抽出部３４は、例えば、ＨＴＭＬページを開いている状態でJava（登録商標）Scriptを実行することで取得することができる。あるいは、要素データ抽出部３４は、ブラウザＡのクッキー格納領域（通常処理部３２が有しているものとする）を直接操作して読み出してもよい。一般にブラウザは複数のクッキーを記憶できる。したがって、ステップＳ５０では、複数のクッキーが取得される場合がある。ステップＳ５２では、ステップＳ５０において複数のクッキーが取得されたか否かを判断する。ここでの判断が否定された場合には、ステップＳ５６に移行するが、ステップＳ５２の判断が肯定された場合には、ステップＳ５４に移行する。

【００４５】

ステップＳ５４では、要素データ抽出部３４が、取得された複数のクッキーから適当なクッキーを選択する。この場合、選択される適当なクッキーを、クッキーの名前などにより事前に決めておくなどすることができる。あるいは、ユーザ（テスト実施者）に対してクッキーを表示し、ユーザに選択を促すようにしてもよい。

【００４６】

そして、ステップＳ５６では、要素データ抽出部３４が、取得（選択）したクッキーの名前、値、対象、ドメイン名を取得する。

【００４７】

これに対し、図４（ｂ）のステップＳ２４の処理（要求可能状態設定処理）では、図８（ｂ）のステップＳ６０において、要求可能状態設定部５０が、ブラウザＢに対象ドメイン名のページを表示しているか否かを判断する。ここでの判断が否定された場合には、ステップＳ６４に移行するが、判断が肯定された場合には、ステップＳ６２に移行する。

【００４８】

ステップＳ６２に移行した場合、要求可能状態設定部５０は、新たに対象ドメイン名のページを表示する。この場合、要求可能状態設定部５０は、たとえばJava（登録商標）Scriptを用いて対象ドメイン名のページを開くことができる。そして、ステップＳ６４に移行すると、要求可能状態設定部５０は、対象ドメイン名のページの状態はそのままで、ブラウザＡの要素データ送信部３６から送信され、ブラウザＢの要素データ受信部４８で受信したクッキーをブラウザＢのクッキー記憶領域（通常処理部４２が有しているものとす

る）に追加または上書きする。この場合、要求可能状態設定部５０は、たとえば、対象ドメイン名のページのcookie属性（document.cookie）にクッキーの名前と値を追加する。

【００４９】

なお、上記の処理以外は、ＣＳＲＦ攻撃に対する攻撃模倣テストと同様の処理（図４（ａ）、図４（ｂ）の処理）を行うことになる。このようにすることで、セッションフィクセーション攻撃に対する攻撃模倣テストを行うことが可能となる。

【００５０】

以上、詳細に説明したように、本実施形態によると、サーバ１４への接続が可能な被害者側ブラウザＢがサーバ１４に対するアクセスを行った際に、被害者側ブラウザＢが被害を被る可能性のある状態（要求可能状態）にするための情報である要素データを、サーバ１４への接続が可能な加害者側ブラウザＡが取得し（ステップＳ１２）、加害者側ブラウザＡが取得した要素データを加害者側ブラウザＡから被害者側ブラウザＢに送信し（ステップＳ１４）、被害者側ブラウザＢの状態に要素データを反映させて、被害者側ブラウザＢの状態を、サーバ１４にアクセスした場合に被害を被る可能性のある状態（要求可能状態）に設定する（ステップＳ２４）。このように、要素データの取得、送受信、要求可能状態への設定を自動化することで、攻撃模倣テスト（ブラックボックステスト）を何度も繰り返すような場合であっても、簡易にテストを行うことが可能となる。

【００５１】

また、本実施形態によると、要素データとしてフォームやクッキーを用いることで、クロスサイトリクエストフォージェリー（ＣＳＲＦ攻撃）や、セッションフィクセーション攻撃に関するテストを簡易に行うことが可能となる。

【００５２】

なお、上記実施形態では、攻撃者側ブラウザＡと被害者側ブラウザＢとが、同一の装置上に存在しており、各ブラウザ間では、一般的なプロセス間通信の手段を用いて、要素データを送受信する場合について説明した。しかしながら、これに限られるものではなく、要素データの受け渡しをクリップボード経由で行ってもよい。クリップボードとは、情報処理部２０のＯＳやアプリケーションから容易に読み書きできる一時的なデータの仮想的な格納場所であり、一般的に切り取り（カット）、コピー、貼り付け（ペースト）などの機能を実現するために用いられる。この場合、攻撃者側の要素データ送信部３６が、要素データ（フォーム情報やクッキー情報）を、クリップボードへ書き込み、被害者側の要素データ受信部４８がクリップボードから当該情報を読み出すことで、受け渡しが完了する。なお、クリップボード上に複数のデータを格納して選択的に取り出せるようにしてもよいし、あるいは、クリップボード上で、データの種類（フォーム情報とクッキー情報の区別）をメタデータとして保持するようにしてもよい。

【００５３】

また、要素データの受け渡しをファイル経由で行うこととしてもよい。攻撃者側ブラウザＡの要素データ送信部３６は、要素データをファイルへ書き出し、被害者側ブラウザＢの要素データ受信部４８は、そのファイルから要素データを読み出すようにする。この場合、被害者側ブラウザＢは、特定のファイルの更新を監視したり、攻撃者側ブラウザＡからネットワークなどを介してファイル更新イベント（およびそのファイル名）を受け取ったりしたときを、受信（読み出し）の契機とすることができる。

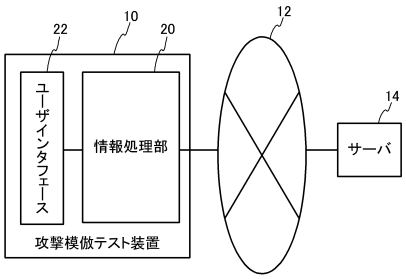
【００５４】

なお、上記実施形態では、ＣＳＲＦ攻撃に関する攻撃模倣テストを行う場合、被害者側ブラウザＢにおいて、受け取ったフォーム情報を含むＨＴＭＬページを生成することとしていたが、これに限られるものではない。例えば、当該ＨＴＭＬページの生成を攻撃者側ブラウザＡで行うこととしてもよい。例えば、攻撃者側において上記実施形態の被害者側と同様にＨＴＭＬページを生成し、それをファイルに保存する（図９の（１ａ）参照）。この場合、攻撃者側から被害者側へ渡すデータは、保存したＨＴＭＬファイルのＵＲＬのみとすることができる（図９の（２）参照）。このようにすることで、被害者側では受け取ったＵＲＬへアクセスすることで要求可能状態に設定することが可能である（図９の（

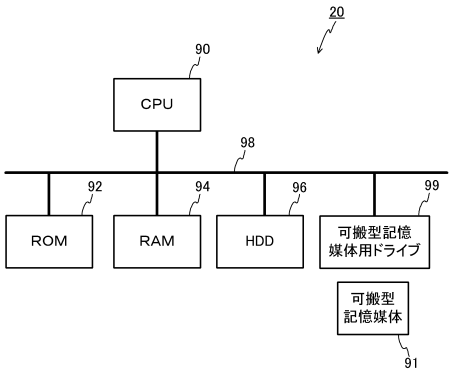
３ａ）参照）。
【００５５】
なお、図９の（１ｂ）、（３ｂ）に示すように、攻撃者側ブラウザＡが特定のポート番号で待ち受けて一種のＷｅｂサーバとして振る舞うようにしてもよい。この場合、攻撃側ブラウザＡにアクセスしてきた被害者側ブラウザＢに対して、生成したＨＴＭＬページを提供するようにしてもよい。この場合、図９の（２）では、攻撃者側ブラウザＡは、待ち受けているポート番号を指すＵＲＬを被害者側ブラウザＢに送信するのみでよい。
【００５６】
また、図９の（１ｃ）、（３ｃ）に示すように、生成したＨＴＭＬページをテスト対象とは別のＷｅｂサーバへアップロードすることとしてもよい。
【００５７】
なお、上記の処理機能は、コンピュータによって実現することができる。その場合、処理装置が有すべき機能の処理内容を記述したプログラムが提供される。そのプログラムをコンピュータで実行することにより、上記処理機能がコンピュータ上で実現される。処理内容を記述したプログラムは、コンピュータで読み取り可能な記録媒体に記録しておくことができる。
【００５８】
プログラムを流通させる場合には、例えば、そのプログラムが記録されたＤＶＤ（Digital Versatile Disc）、ＣＤ－ＲＯＭ（Compact Disc Read Only Memory）などの可搬型記録媒体の形態で販売される。また、プログラムをサーバコンピュータの記憶装置に格納しておき、ネットワークを介して、サーバコンピュータから他のコンピュータにそのプログラムを転送することもできる。
【００５９】
プログラムを実行するコンピュータは、例えば、可搬型記録媒体に記録されたプログラムもしくはサーバコンピュータから転送されたプログラムを、自己の記憶装置に格納する。そして、コンピュータは、自己の記憶装置からプログラムを読み取り、プログラムに従った処理を実行する。なお、コンピュータは、可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することもできる。また、コンピュータは、サーバコンピュータからプログラムが転送されるごとに、逐次、受け取ったプログラムに従った処理を実行することもできる。
【００６０】
上述した実施形態は本発明の好適な実施の例である。但し、これに限定されるものではなく、本発明の要旨を逸脱しない範囲内において種々変形実施可能である。
【００６１】
なお、以上の説明に関して更に以下の付記を開示する。
（付記１）　サーバへの接続が可能な第１のブラウザが前記サーバに対するアクセスを行った際に当該第１のブラウザが被害を被る可能性のある状態にするための情報である要素データを、前記サーバへの接続が可能な第２のブラウザが取得する取得工程と、前記取得工程で前記第２のブラウザが取得した前記要素データを前記第２のブラウザから前記第１のブラウザに送信する送信工程と、前記送信工程で送信された要素データを前記第１のブラウザで受信する受信工程と、前記第１のブラウザの状態に、前記受信工程で受信された前記要素データを反映させて、当該第１のブラウザの状態を、前記サーバにアクセスした場合に被害を被る可能性のある状態に設定する設定工程と、を含む攻撃模倣テスト方法。
（付記２）　前記要素データは、フォームであり、前記攻撃は、クロスサイトリクエストフォージェリーであることを特徴とする付記１に記載の攻撃模倣テスト方法。
（付記３）　前記要素データは、クッキーであり、前記攻撃は、セッションフィクセーションであることを特徴とする付記１に記載の攻撃模倣テスト方法。
（付記４）　前記送信工程及び前記受信工程では、前記要素データを、ＩＰネットワークを用いた受け渡し、クリップボードを用いた受け渡し、ファイルを用いた受け渡し、のいずれかにより、前記第２のブラウザと前記第１のブラウザとの間の送受信を行うことを特

徴とする付記１～３のいずれか一項に記載の攻撃模倣テスト方法。
（付記５） サーバへの接続が可能な第１のブラウザと、前記第１のブラウザが前記サーバに対するアクセスを行った際に、当該第１のブラウザが被害を被る可能性のある状態にするための情報である要素データを取得する、前記サーバへの接続が可能な第２のブラウザと、を備え、前記第２のブラウザは、取得した前記要素データを前記第１のブラウザに送信する送信部を有し、前記第１のブラウザは、前記要素データを受信する受信部と、当該第１のブラウザの状態に前記受信部で受信した前記要素データを反映させて、前記第１のブラウザの状態を前記サーバにアクセスした場合に被害を被る可能性のある状態に設定する設定部と、を有することを特徴とする攻撃模倣テスト装置。
（付記６） 前記要素データは、フォームであり、前記攻撃は、クロスサイトリクエストフォージェリーであることを特徴とする付記５に記載の攻撃模倣テスト装置。
（付記７） 前記要素データは、クッキーであり、前記攻撃は、セッションフィクセーションであることを特徴とする付記５に記載の攻撃模倣テスト装置。
（付記８） 前記送信部及び前記受信部では、前記要素データを、ＩＰネットワークを用いた受け渡し、クリップボードを用いた受け渡し、ファイルを用いた受け渡し、のいずれかにより、前記第２のブラウザと前記第１のブラウザとの間の送受信を行うことを特徴とする付記５～７のいずれか一項に記載の攻撃模倣テスト装置。
（付記９） サーバへの接続が可能な第１のブラウザが前記サーバに対するアクセスを行った際に、当該第１のブラウザが被害を被る可能性のある状態にするための情報である要素データを、前記サーバへの接続が可能な第２のブラウザが取得し、前記取得する処理で前記第２のブラウザが取得した前記要素データを前記第２のブラウザから前記第１のブラウザに送信し、前記送信する処理で送信された前記要素データを前記第１のブラウザで受信し、前記第１のブラウザの状態に、前記受信する処理で受信された前記要素データを反映させて、当該第１のブラウザの状態を、前記サーバにアクセスした場合に被害を被る可能性のある状態に設定する処理を、コンピュータに実行させることを特徴とする攻撃模倣テストプログラム。
（付記１０） 前記要素データは、フォームであり、前記攻撃は、クロスサイトリクエストフォージェリーであることを特徴とする付記９に記載の攻撃模倣テストプログラム。
（付記１１） 前記要素データは、クッキーであり、前記攻撃は、セッションフィクセーションであることを特徴とする付記９に記載の攻撃模倣テストプログラム。
（付記１２） 前記送信する処理では、前記要素データを、ＩＰネットワークを用いた受け渡し、クリップボードを用いた受け渡し、ファイルを用いた受け渡し、のいずれかにより、前記第２のブラウザから前記第１のブラウザへ送信することを特徴とする付記９～１１のいずれか一項に記載の攻撃模倣テストプログラム。
【符号の説明】
【００６２】
１０ 攻撃模倣テスト装置
１２ ネットワーク（ＩＰネットワーク）
１４ サーバ
３６、４６ 要素データ送信部（送信部）
３８，４８ 要素データ受信部（受信部）
４０，５０ 要求可能状態設定部（設定部）

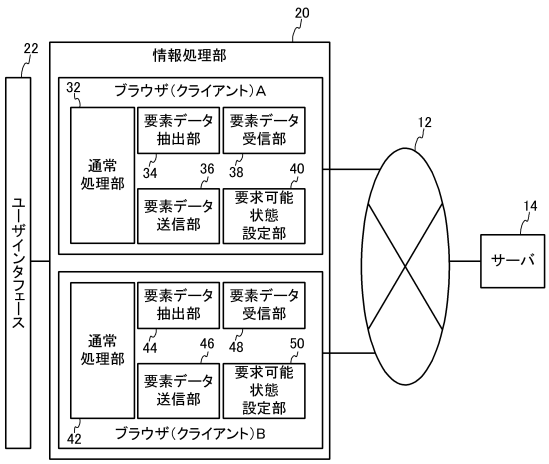
【図１】


22
10
20
12
14
ユーザインタフェース
情報処理部
サーバ
攻撃模倣テスト装置

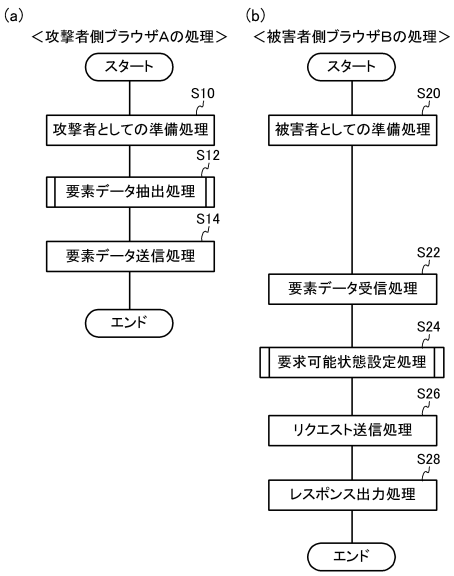
【図２】


20
90
CPU
98
92
94
96
99
ROM
RAM
HDD
可搬型記憶媒体用ドライブ
可搬型記憶媒体
91

【図３】


20
22
情報処理部
32
ブラウザ（クライアント）A
通常処理部
要素データ抽出部
要素データ受信部
34
36
38
40
要素データ送信部
要求可能状態設定部
12
14
サーバ
ユーザインタフェース
44
46
48
50
42
ブラウザ（クライアント）B

【図４】

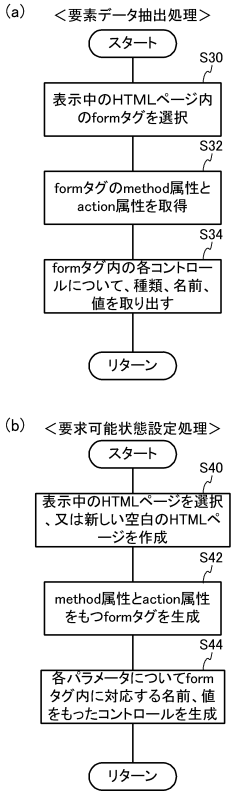

(a)
＜攻撃者側ブラウザAの処理＞
スタート
S10
攻撃者としての準備処理
S12
要素データ抽出処理
S14
要素データ送信処理
エンド
(b)
＜被害者側ブラウザBの処理＞
スタート
S20
被害者としての準備処理
S22
要素データ受信処理
S24
要求可能状態設定処理
S26
リクエスト送信処理
S28
レスポンス出力処理
エンド

【図５】

＜要素データ(フォーム)＞

項目	値		
method	POST		
action	http://www.example.com/submit		
パラメータ群	種類	名前	値
	text	title	Re:product
	testarea	message	Hi.I would like…
	hidden	token	0a1b2c3d

【図６】

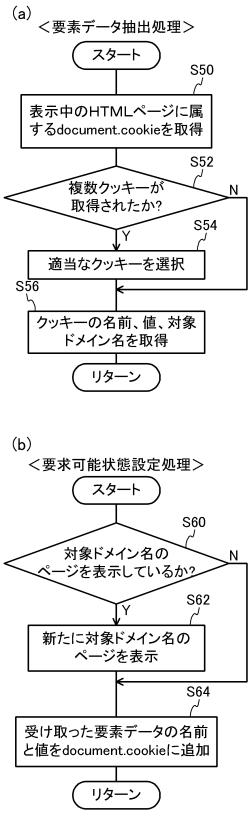


【図７】

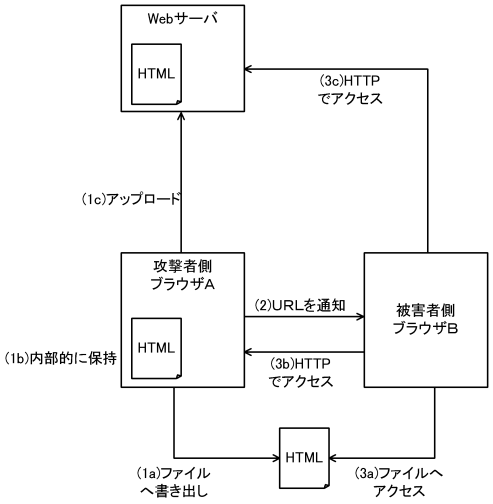
＜要素データ(クッキー)＞

項目	値
名前	JSESSIONID
値	0a1b2c3d
対象ドメイン名	www.example.com
対象パス	/
有効期限	なし(ブラウザが動作する間のみ)
Secure属性	False

【図８】



【図９】


Webサーバ
HTML
(3c)HTTP
でアクセス
(1c)アップロード
攻撃者側
ブラウザA
(2)URLを通知
被害者側
ブラウザB
HTML
(1b)内部的に保持
(3b)HTTP
でアクセス
HTML
(1a)ファイル
へ書き出し
(3a)ファイルへ
アクセス

フロントページの続き

(56)参考文献 特開２００４－１６４６１７（ＪＰ，Ａ）
特表２０１０－５３３９０８（ＪＰ，Ａ）
特開２０１０－２５０５８３（ＪＰ，Ａ）
特開２００８－２６２３１１（ＪＰ，Ａ）
小菅祐史，効果的な攻撃テストによるＷｅｂアプリケーションの脆弱性検出手法，情報処理学会研究報告　平成２１年度▲１▼　［ＣＤ－ＲＯＭ］，日本，社団法人情報処理学会，２００９年６月１５日，１～８頁

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／５７
Ｇ０６Ｆ　１１／２８