情報処理学会論文誌 Vol.53 No.9 2107–2116 (Sep. 2012)

DNSとWebブラウザを協調させたWebアクセス制御方式 Request Policy Frameworkの提案と評価

植村 崇史[1,a)] 小須田 優介[2] 佐々木 良一[1]

受付日 2011年11月30日, 採録日 2012年6月1日

概要：近年，Web サイト改ざんと Web ブラウザを通してマルウェアをダウンロードさせる Drive-by download を用いた攻撃手法である Gumblar が Web の脅威となっている．既存の対策としては Web サイト改ざんに頻繁に使用される悪意ある JavaScript コードの検知手法等があるが，危険なスクリプトを発見するのは容易ではなく万全な対策とはいい難い．そこで，著者らは，改ざんによって生じる正規の Web サイトによる不正な誘導を制御し，精度の高いホワイトリストによって Web サイト管理者の意図するリクエストのみを行うための仕組みである Request Policy Framework を提案する．本論文では，Request Policy Framework の実現方式，試作開発結果，機能，性能の評価結果ならび考察結果等の報告を行う．

キーワード：アクセス制御・認証，Web セキュリティ，Gumblar，DNS

Proposal and Evaluation of Web Access Control System Request Policy Framework for Cooperation of DNS and a Web Browser

Takashi Uemura[1,a)] Yusuke Kosuda[2] Ryoichi Sasaki[1]

Received: November 30, 2011, Accepted: June 1, 2012

Abstract: The drive by download attack technique such as Gumblar, which compromise websites by infecting them with a virus and direct operations, has been increasing rapidly in recent years. Existing measures to detect dangerous scripts using a Web browser cannot protect against all of the attacks, because it is difficult for everyone to find dangerous scripts. Therefore, the authors devised a mechanism named RPF (Request Policy Framework) that uses a highly accuracy whitelist obtained by using a Web browser cooperating with the DNS server. This paper reports the detailed mechanism of RPF, the prototype program, evaluation results of its function and performance, and the consideration of coverage by RPF.

Keywords: access control and authentication, Web security, Gumblar, DNS

1. はじめに

近年，Web サイト改ざんと Web 感染型マルウェアを組み合わせた攻撃手法である Gumblar が Web の新たな脅威となっている．Gumblar は，Web サイトからユーザの同意を得ずにマルウェアをダウンロードさせる Drive-by download 攻撃と，その攻撃 Web サイトへ誘導するために正規 Web サイトを改ざんするという攻撃を組み合わせており，ユーザは正規 Web サイトを訪れるだけで攻撃を受けるため，マルウェアに感染しやすいという特徴を持つ．

Gumblar のような Web サイトの改ざんを用いた Drive-by download 攻撃（以降，Gumblar 攻撃と記載する）の手順としては，**図 1** に示す流れが一般的である．(1-a) 攻撃者は事前にマルウェア配布サイトを用意し，(1-b) マルウェア配布サイトへ誘導させるためのスクリプトを正規の Web サイトに埋め込む等の改ざんを行う．(2) 一般ユーザが改ざんサイトを閲覧することで，(3-a) 一般ユーザの Web ブラウザでスクリプトが動作しマルウェア配布サイトへ誘導

[1] 東京電機大学
Tokyo Denki University, Adachi, Tokyo 120–8551, Japan
[2] NEC ソフト株式会社
NEC Soft Ltd, Koto, Tokyo 136–0082, Japan
[a)] uemura@isl.im.dendai.ac.jp

© 2012 Information Processing Society of Japan

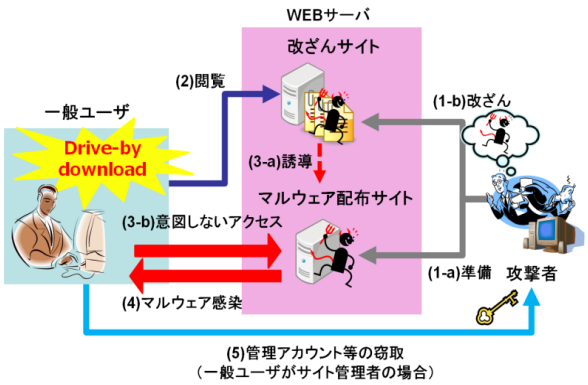

図 1 Gumblar 攻撃の流れ
Fig. 1 Flow of Gumblar attack.

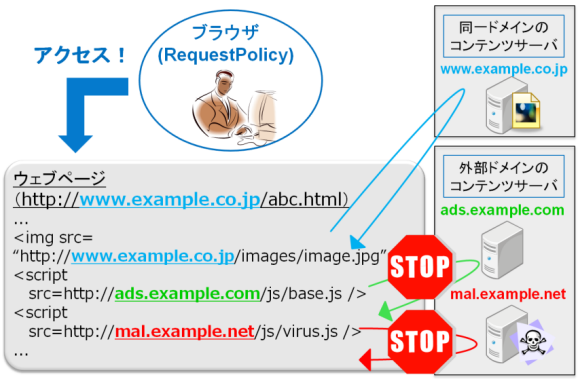

図 2 RequestPolicy の動作例
Fig. 2 Example of operation based on RequestPolicy.

され，(3-b) 一般ユーザは意図せずマルウェア配布サイトへアクセスする．そして，(4) 一般ユーザの PC の脆弱性を攻撃して自動的にマルウェアを感染させる．また，(5) マルウェア感染した一般ユーザが Web サイト管理者であった場合，管理アカウントを盗まれ，その Web サイトが新たに改ざんされることで，同様の攻撃と被害が拡大していく．

実際に改ざんされた Web サイトを一般ユーザが閲覧して被害に遭うようなことがあれば，その Web サイトは信用を失ってしまう．そのため，Gumblar 攻撃に対する効果的な対策が必要とされている．一般的な対策であるパターンマッチング方式のアンチウィルスソフトはパターンファイルが未対応の脅威には対応が難しい．また，攻撃者の IP アドレスをファイアウォールにブラックリストとして登録し，Web サイトへの不正アクセスを防止する対策は，未定義の IP アドレスからのアクセスは防止することができない．

さらに，既存の対策や関連研究で提案されている手法には，誤判定（false positive，false negative）や最新の脅威へ対応するまでのタイムラグ等の問題点があり，万全な対策とはいい難い．

本研究では正規の Web サイトからの不正な誘導を制御し，精度の高いホワイトリストによって Web サイト管理者の意図するリクエストのみを行うための仕組みである Request Policy Framework を提案する．本論文では，2 章で考察による既存の対策手法に対する問題点の洗い出しを行い，3 章において，提案システムの概要を述べる．さらに，4 章では，提案手法の実装に関する説明を行い，5 章で実装に対して機能，速度の評価を行う．6 章において，提案システムの運用について考察を行う．

2. 既存の対策・関連研究

2.1 既存の対策

本節では既存の対策方法に関して述べる．既存の対策には以下に示すように Web ブラウザを用いたスクリプトの無効化・制御やクロスドメインリクエストの制御，Web サイト側で対策する改ざんチェックサービス等がある．

(1) Web ブラウザによるスクリプトの無効化・制御

悪性のスクリプトコードによるリダイレクトを防止するための対策として，Web ブラウザの設定でスクリプト動作を無効にするという対策が存在する．しかし，この対策を導入すると JavaScript をはじめとする Web サイトに用いられるスクリプトは，現在の Web に広く普及しているため，Web サイトの正常なサービスを受けられない可能性が高い．そのため，Web ブラウザのスクリプト動作を無効にするという対策は現実的であるとはいい難い．

Web ブラウザのスクリプト動作を無効にせず，悪意のあるスクリプトを防ぐ手段として，NoScript [1] という Firefox（Web ブラウザ）の拡張機能（アドオン）があるが，デフォルトですべてのスクリプトがブロックされるため，許可設定を手動で入れるまで，多くの Web サイトが正常に動作しない．一般ユーザが Web ページの動作に必要なスクリプトを適切に判断して許可するのは技術的なスキルが必要とされるため難しく，Web サイトの更新に追随して設定をメンテナンスすることは負担が大きく困難である．また，.htaccess を改ざんしたリダイレクト攻撃をする Gumblar の亜種を防げない可能性があるため，対策として不十分である．

(2) Web ブラウザによるクロスドメインリクエストの制御

Web ブラウザのスクリプト動作を無効にせずに，別ドメインの Web サイトへのリダイレクトを防ぐ手段として，Firefox の拡張機能に RequestPolicy [2] というものがある．RequestPolicy は，閲覧中の Web ページの ⟨img⟩，⟨script⟩，⟨iframe⟩ タグ等から発生するクロスドメインリクエストをホワイトリストによって制御する機能を有している．図 2 の例では，アクセスした www.example.co.jp ドメインの Web ページから，Web ブラウザの自動読み込みで発生するクロスドメインリクエスト（ads.example.com, mal.example.net へのリクエスト）のみが RequestPolicy のホワイトリストで許可されていない場合は制御の対象と

© 2012 Information Processing Society of Japan

なる．

しかしながら，RequestPolicy にも問題点はある．Web ページと同一ドメインへのリクエストは自動的に許可されるため，前述の NoScript よりも多くのサイトが正常に動作するが，それでも，数多く存在する埋め込み動画やブログパーツ，他の Web サービスが提供する API を利用した Web サイトでは正常に動作しない．これらに対しては一般ユーザが手動で許可設定をする必要があるが，前述の NoScript と同様に，一般ユーザが Web ページの動作に必要なコンテンツを適切に判断して許可するのは技術的なスキルが必要とされるため難しく，Web サイトの更新に追随して設定をメンテナンスすることは負担が大きく困難である．

(3) Web サイトの改ざんチェックサービス

Web サイトの改ざん対策として，Web サーバのコンテンツを定期的に監視して改ざんのチェックを行い，改ざん発見時に自動対応してくれるサービスを導入する方法がある．代表的なサービスとしては，gred セキュリティサービス [3] や Gumblar Watch [4] がある．これらのサービスを Web サイトに導入する利点は，セキュリティの知識や技術を持たない Web サイト管理者であっても，改ざん攻撃の対策が容易にできることである．

しかし，Web サイトの改ざん対策の導入や維持管理が簡略化される一方で，問題点も存在する．それは，サービスの導入・維持のためにコストがかかるため，個人で Web サイトを運営している管理者にとっては採用が難しいことである．また，これらのサービスは Web サイト内のファイルを定期解析することにより，改ざんを検知するが，Web サイトの改ざんから検知までのタイムラグを 0 にできないため，その間の訪問者はマルウェア感染の危険に晒されてしまう．

2.2 関連研究

関連研究として，Web ページの例外振舞いに焦点を当てた検知手法やリンクの深さや広がりに焦点を当てた異常な通信の検知手法，インジェクション攻撃に対する痕跡検知手法，リダイレクトの特定サイトへの集中と改ざんサイトの更新頻度に焦点を当てた相補的な検知手法等があげられる．本節ではそれらの関連研究に関して述べる．

(1) Web ページの例外振舞い分析

悪性の Web ページは通常の Web ページには見られない特徴があるとして，それらの特徴を例外的な振舞いとして分析・点数化することで，悪性の Web ページを検知する Web Page Inspection（WPI）[5] という手法が提案されている．WPI はデータセットを用いた実験において，false negative の割合が 0%であり悪性の Web ページはすべて検知するという高い検知率を示している．しかし，正規の Web ページを悪性と判断する false positive の割合が 3.53%あり，それらは通常のオンライン広告であったとされている．また，攻撃方法が変化した場合の追随性に疑問が残る．

(2) リンクの深さと広がりによる異常な通信の検知手法

改ざんサイトの Web ページのリンクの深さや広がりに焦点を当てた，マルウェアの感染活動に関わる異常な通信の検知手法 [6] が提案されている．文献内では，Web ブラウザの拡張機能としてプロトタイプが作成され，Web ブラウザによる自動通信をリンクの深さと広がりの指標から判定することで通信を遮断するシステムを実装していた．プロトタイプを用いた実験では改ざんされた正規の Web サイトの件数は少なかったものの，マルウェア感染をすべて阻止することができたことを示していた．しかし，正規の Web サイトにアクセスした際に一部の広告画像が表示されない等の誤検知が 3.33%発生したことも述べられている．また，(1) と同様に，攻撃方法が変化した場合の追随性に疑問が残る．

(3) インジェクション攻撃に対する痕跡検知手法

文献 [7] では，誘導元 Web サイトに埋め込まれるリダイレクト命令文，マルウェア配布サイトの動作，ドメイン名のトップレベルドメインや存続期間，検索エンジンにおける Web サイトのランクの視点から改ざんサイトを検知する手法が提案されている．

文献内では，⟨script⟩ タグや ⟨iframe⟩ タグを利用したインジェクション攻撃の検知は有効性が示されていたが，パターンマッチ手法を用いているため，未定義の難読化スクリプトのインジェクション攻撃に対して，検知できない可能性が懸念される．

(4) 特定サイトへのリダイレクトの集中と改ざんサイトの更新頻度による相補的な検知手法

文献 [8] では，複数の Web サイトからリダイレクトが集中する Web サイトをマルウェア配布サイトと仮定し，リダイレクト元サイトを改ざんサイトとして検知する探査と攻撃者がリダイレクトの集中を回避するために高頻度で変更を加える Web サイトを改ざんサイトとして検知する探査を組み合わせた手法が提案されている．

この手法では，特定の Web サイトにリダイレクトが集中した場合に，リダイレクト元サイトを改ざんサイトと判断しているが，著名な Web サイトは Gumblar とは関わりがなくてもリダイレクトが集中する可能性が考えられるため，誤検知が懸念される．

2.3 問題点のまとめ

既存の対策・関連研究で提案されている手法についての問題点を以下にまとめる．

(1) 脅威へ対応するまでにタイムラグがある

パターンマッチングによる検知手法は，新たな攻撃・脆弱性が発覚するたびに対策を必要とするため，最新の脅威

© 2012 Information Processing Society of Japan

へ対応するまでにタイムラグが生じる可能性があり，その間にユーザが脅威に晒されてしまう．

改ざんチェックサービスは，定期実行により改ざんを検知するため，検知までにタイムラグが生じ，その間はユーザが脅威に晒されてしまう．

(2) 悪性な Web ページの検知手法には誤判定がある

関連研究に代表されるような Web ページの振舞いの分析から悪性の Web ページを検知する手法には誤判定があり，脅威を防げない可能性がある．また，正常であるにもかかわらずオンライン広告の表示を妨げてしまう等，通常のサービスが受けられない状態が発生する可能性がある．このことから，Web ページのコンテンツの善悪を第三者が完璧に判断することは難しいと考えられる．

(3) Web ブラウザによる制御は適切な設定が困難

Web ブラウザの設定，あるいは，NoScript で単純にスクリプトをブロックするだけでは，Gumblar の亜種に対応できない可能性があるが，スクリプトに限らずクロスドメインリクエストを制御する RequestPolicy を利用すれば対応が可能である．しかしながら，RequestPolicy のホワイトリストを手動でメンテナンスし続けることは，技術的にも労力的にも一般ユーザへの負担が大きく困難である．

(4) 企業向けの対策は個人の Web サイトへの適用が難しい

改ざんチェックサービスのような企業向けの対策は導入・維持にコストがかかるため，個人で Web サイトを運営している管理者にとっては採用が難しい．

著者らは，以上の問題点を解決するための方式を考案した．次章で提案方式について説明を行う．

3. 対策手法の提案

3.1 必要とされるアプローチ

2 章において提起した既存の対策手法の問題点を考慮した結果，正規の Web サイトによる不正な誘導を防ぐためには，RequestPolicy のようなドメインレベルのホワイトリスト制御をベースとした手法に，以下のような要件を加えた総合的なシステムが効果的な対策になると考えた．

① ホワイトリストは Web サイト管理者が作成して提供する．
② ホワイトリストは Web サービスと分けて管理する．
③ 一般ユーザが Web サイトにアクセスした際にホワイトリストが自動適用される．
④ 既存のネットワークサーバを活用する．

Web サイト作成者以外が Web サイトに必要なコンテンツを判断することが難しいことを示唆したが，Web サイトを作成した管理者本人ならばそれが可能であると考えた（①）．次にそのホワイトリストをどこへ保管し，提供するかだが，Gumblar 攻撃の手法では Web サイトを管理するためのアカウントを窃取されてしまうことを考慮すると，ホワイトリストは Web サイト管理者が管理できる範囲の外に置くことで，安全性を確保する．これは，別サーバであることが好ましいが，同一サーバ内でも Web とは異なるサービスの管理下に置くことで，攻撃の難度は上がるため，一定の安全性が確保されると考えられる（②）．そして，一般ユーザが Web サイトにアクセスした際にホワイトリストを自動取得し，Web ブラウザに自動適用することで，改ざんコンテンツ対応までのタイムラグを 0 にできる（③）．以上の機能を備えた手法により，Gumblar 等の不正誘導の被害を未然に防ぐ仕組みを実現できると考えた．さらに既存のネットワークサーバを活用することで追加の機器購入をせずに導入できることを目指した（④）．次節で提案手法の具体的な仕組みについての説明を行う．

3.2 提案手法

著者らは，前述の RequestPolicy のホワイトリスト機能と DNS を利用した既存の技術である Sender Policy Framework（SPF）に着目した．SPF は電子メールの送信ドメイン認証の 1 つで，RFC4408 [9] に規定されており，日本では携帯電話事業者を中心に普及している．

DNS はドメイン名から IP アドレスを解決する分散型のデータベースという認識が一般的であるが，ドメイン名から IP アドレスの解決に用いられる A レコードは検索データの型であるリソースレコード（RR）の 1 つであり，そのほかにもいくつか種類が存在する．SPF は TXT レコード[*1]という RR を利用してホワイトリストとして許可する IP アドレスを指定している．この TXT レコードが任意の文字列を書き込むことが可能である点，ドメインと情報の紐付けが容易である点，Web サービスとは異なる管理下にある点をふまえ，前節における②の管理場所として最適であると考えた．なお，Web サービスと DNS が同一マシン上に存在している場合においても，各サービスへのアクセス権限が異なっていれば，Web サーバのコンテンツと DNS の設定ファイルの両方を 1 度に改ざんすることは難しいと考えられる．

Web サイト管理者が自身の Web サイトの構造をもとにホワイトリスト（RequestPolicy で許可させたいドメイン）を作成し，DNS の TXT レコードにそれを保管することで配信可能な状態を確立する．そして，Web ページへの Web アクセスが発生した際に，DNS へホワイトリストの配信を要求し，取得したホワイトリストを RequestPolicy に自動適用し，Web ページに含まれるコンテンツへのリクエストをドメインレベルで制御することで，安全な Web アクセスの仕組みを実現できる．

[*1] DNS サーバが対応している場合，SPF レコードという RR に定義されることもある．

© 2012 Information Processing Society of Japan

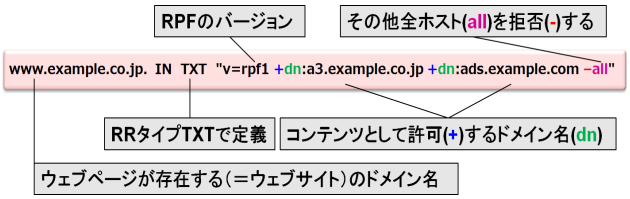

図 3 RPF 情報の基本フォーマット
Fig. 3 Basic format of RPF information.

3.3 Request Policy Framework の概要

提案手法は，RequestPolicy と Sender Policy Framework を応用した技術であるため，Request Policy Framework（RPF）と名付けた．RPF のシステムを実現するためには，一般ユーザ，Web サイト管理者，DNS 管理者が事前準備を行う必要がある．各々必要とされる事前準備の項目を以下に示す．

① 一般ユーザ

Web ブラウザに RPF 対応機能を導入する（著者らは，Firefox の拡張機能として実装した）．

② Web サイト管理者

Web ページに含まれる外部コンテンツのドメインを定義したホワイトリストを自身の Web サイトのドメインを管理している DNS 管理者に伝達する．

③ DNS 管理者

②のホワイトリストを，Web サイトのドメインの TXT レコードに登録する．

この①から③の事前準備が整った環境下で，当該の Web サイトにアクセスすることで，Web サイト管理者が作成した信頼性の高いホワイトリストが，一般ユーザの Web ブラウザに自動適用される．これにより，Gumblar 等の不正誘導を防ぎ，安全な Web アクセスが実現可能となる．以上が RPF の概要である．

本研究では，DNS の TXT レコードに許可するドメイン情報を登録したホワイトリストを，SPF の記述方式を模倣し，**図 3** のような形態を基本フォーマットとする．また，RPF 環境下における DNS の TXT レコードに登録したホワイトリストは RPF 情報と呼ぶ．RPF 情報では，DNS の TXT レコードに +dn:の形で Web サイト管理者が許可するドメインを定義する．ここで許可したドメイン以外は-all の箇所に該当するため，拒否される．

3.4 Request Policy Framework の仕組み

3.3 節で示した事前準備の完了をもって，初めて RPF の Web アクセスが有効となる．さらに，本節では，RPF における Web アクセスの仕組み（**図 4**）を解説する．

① Web アクセス

RPF を導入した Web ブラウザを使用して Web サーバ（図 4 の例では www.example.co.jp）にアクセスし，Web ページを取得する．

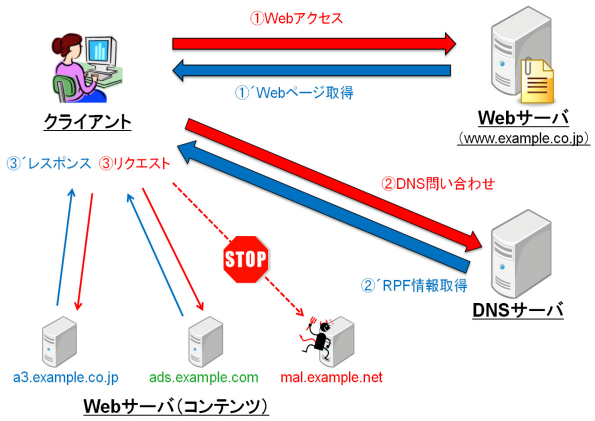

図 4 RPF 環境下における Web アクセスの仕組み
Fig. 4 Mechanism for Web access under RPF environment.

② ホワイトリストの取得

取得した Web ページを Web ブラウザ側で解析し，クロスドメインリクエストが発生した場合，Web アクセスした Web サーバのドメイン情報が登録されている DNS へ問い合わせ，RPF 情報を取得する．

③ ホワイトリストに基づいたリクエスト制御

取得した RPF 情報をホワイトリストとして Web ブラウザへ格納し，許可されたドメインの Web サーバへのみリクエストし，コンテンツを取得する．図 4 の例では，図 3 で定義した RPF 情報に基づいたリクエストを想定していて，+dn:で許可されている a3.example.co.jp, ads.example.com へリクエストし，コンテンツを取得している．仮に Web ページが mal.example.net の Web サーバへリクエストするように改ざんされていたとしても，RPF 情報で許可されていない（-all に該当する）ため，リクエストは遮断される．

上記のプロセスを経て，RPF 環境下における Web アクセス時の高精度のホワイトリストによるクロスドメインリクエストの制御が実現する．ホワイトリストによる制御は，Web ページ上にあるリンクやフォームボタンをクリックするようなユーザの操作をともなう場合は対象外であり，それ以外の外部ドメインの Web サイトへのリクエストが対象となる．具体的には，⟨img⟩，⟨script⟩，⟨iframe⟩ タグ等から自動発生するリクエストが制御の対象であり，Gumblar の亜種である.htaccess ファイルを改ざんしたリダイレクトにも対応可能である．

3.5 Request Policy Framework の効果

RPF の導入には，Web サイト管理者がホワイトリストを作成する作業が生じるが，それ以上にメリットが存在する．以下に，RPF の導入メリットを一般ユーザと Web サイト管理者の両者の観点から示す．

一般ユーザ

- 自動化により Web サイトごとにホワイトリストの設

© 2012 Information Processing Society of Japan

定をする手間がかからない．
- 信頼性の高いホワイトリストが利用可能．

Web サイト管理者
- 提供したい広告用のドメインが除かれることがない．
- ホワイトリスト配布用の新たな（ネットワークサーバ等の）機器を準備する必要がない．

以上が，RPF 環境を導入することによって得られるメリットであり，将来的に RPF の普及を期待する．しかし，Web サイト管理者がホワイトリストの更新を依頼する手間がかかることや DNS 管理者が更新の負担を負うデメリットも考えられる．著者らは今後，これらのデメリットに対して，更新依頼の手間と登録の負担を軽減するための自動化システムを検討していきたいと考えている．

3.6 提案手法のまとめ

Web サイトがどの外部ドメインと関連しているかの判断は，その Web サイトを作った本人ならば正しく行える．そのため，ホワイトリストを Web サイト管理者自身が作成し，提供することで，誤検知発生の可能性を 0 にすることができると考える．

提案システムをまとめると，RPF の導入により精度の高いホワイトリストが一般ユーザの Web ブラウザに自動適用され，Web アクセスよって発生する外部ドメインへの不正なリクエストの制御が可能となる．それはセキュリティに詳しくない一般ユーザにとって扱いやすく，Web サイト管理者にとってもサービスをとりこぼすことなく提供でき，不正改ざんに対する脅威への対策につながる．将来的に RPF が普及することで，Gumblar 攻撃による被害の減少を期待できる．

4. 実装

4.1 実装方法の概要

一般ユーザ側の Web ブラウザに導入する拡張機能としては，オープンソースソフトウェアである RequestPolicy に RPF で必要となる機能を新たに追加し，拡張プログラム名を「RPF 対応版 RequestPolicy」として実装した．実装の概要と環境を以下に示す．

Web ブラウザ
- Mozilla Firefox version 8.0.1

開発言語
- JavaScript，CSS，XPCOM，XUL

追加ステップ数
- 約 400 ステップ

追加機能
- DNS から RPF 情報を取得し，ホワイトリストとして自動適用する機能
- RPF 問い合わせ用 DNS 指定（特定の DNS に RPF 情報を定義して試用するため）
- ホワイトリスト作成支援機能（次節で詳しく記述する）

4.2 ホワイトリスト作成支援機能の開発

RPF の事前準備におけるホワイトリスト作成には，Web サービスの規模によっては，Web サイト管理者の負担が増大する可能性がある．

また，ブログパーツや動画サイトによって提供されているスクリプト，埋め込み動画等の Flash コンテンツは動作中にクロスドメインリクエストが発生するため，埋め込んだスクリプトや動画自体のドメインを許可するだけでは不十分である．

これらを考慮すると，Web サイト管理者がすべてのクロスドメインリクエストを把握し，適切なホワイトリストを作成することは難しいと考えられる．そこで，Web サイト管理者のホワイトリスト作成の負担軽減のため，ホワイトリスト作成の支援機能が必要であると考えた．

RequestPolicy では，一般ユーザが Web サイトにアクセスした際に，クロスドメインリクエストが発生した場合，Web ブラウザのステータスバー右下のアイコンをクリックすることで，許可・拒否されたクロスドメインの一覧を確認することができる（**図 5**）．この機能を応用し，Web サイト管理者のホワイトリスト作成の負担軽減のための，ホワイトリスト作成の支援機能を追加実装した．

追加実装したホワイトリスト作成支援機能の使用手順は以下①から③のとおりである．なお，ホワイトリスト作成支援機能として追加実装した機能は，③の箇所である．

① RPF 対応版 RequestPolicy を導入した Web ブラウザ（Firefox）で，自身が管理している Web ページを閲覧する．

② ホワイトリストとして許可したいドメインに対して，「禁止されている送信先」（クロスドメインリクエスト）を「許可されている送信先」に変更する（**図 6**）．

③ 「許可されている送信先をクリップボードへコピーする」をクリックすることで，許可したいドメインの一覧をコピー（**図 7**）し，テキストエディタ等に貼り付けることができるようになる．また，RPF 情報形式でのコピーは「許可されている送信先をクリップボード

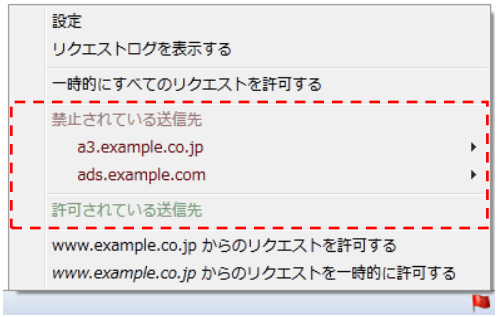

図 5 クロスドメインリクエスト一覧
Fig. 5 List of cross-domain requests.

© 2012 Information Processing Society of Japan

図 6 送信先ドメインの許可
Fig. 6 Method for permission of destination domain.

図 7 送信先ドメインのコピー
Fig. 7 Method for copy of destination domain.

へコピーする（RPF 形式）」をクリックする．

以上の追加機能により，Web サイト管理者のホワイトリスト作成の負担が軽減されることが期待できる．また，Flash 等の動作後に複数のクロスドメインリクエストが発生するコンテンツを Web ページへ埋め込む場合は，コンテンツの提供者（埋め込み動画であれば動画サイトの運営者）が必要となるドメインのリストを提供する形態を確立する[*2]ことによって，Web サイト管理者の負担は将来的に緩和可能であると考えている．

5. 実験・評価

5.1 動作確認実験

RPF の動作確認のため，Web サーバと DNS サーバを用意し，RPF の実験用 Web サイトを作成して実験を行った．Web サイトの構成としては，⟨img⟩ タグによる画像，ブログパーツ，YouTube の埋め込み動画，実験用 Web サイトとはドメインの異なる Web サイトへのリダイレクト命令の各々異なるクロスドメインコンテンツを 4 種類設置し，DNS の TXT レコードにはリダイレクト命令以外の 3 種類のコンテンツが動作するように設定した．**図 8** に実験用 Web サイトのドメインの TXT レコードに設定した RPF 情報を示す．そして，RPF 対応版 RequestPolicy を用いて実験用 Web サイトにアクセスした結果，**図 9** に示

[*2] SPF の include に相当する機能の追加を想定．

```
;; ANSWER SECTION:
rpf.isl.im.dendai.ac.jp. 3600   IN      TXT     "v=rpf1 +dn:bijo-linux.com +dn:s
fx-images.mozilla.org +dn:www.youtube.com +dn:s.ytimg.com +dn:i4.ytimg.com -all"
```

図 8 実験用 RPF 情報
Fig. 8 RPF information for experiments.

図 9 動作確認実験
Fig. 9 Items for operation check experiments.

すようにホワイトリストが適用されて「許可されている送信先」としてドメインが登録されている．また，ホワイトリストで許可されていないリダイレクト命令の外部ドメインが「禁止されている送信先」に表示され，リダイレクトを防いだことを示している．

今回の実験では，Web サイトに設置したコンテンツに対して，DNS から取得した RPF 情報のホワイトリストに基づいたリクエストが正常に行われたことを確認した．また，Web サイトに設置した YouTube の埋め込み動画に関しては，動画の再生によって新たにリクエストするドメインが増加し，YouTube 側の負荷分散処理により，それらのドメインが不定期に変化するため，現状の仕様では頻繁に RPF 情報を更新しなければならないことが分かった．許可するドメイン名をワイルドカードで指定できるようにする，または，ネットワークアドレス単位で許可する機能（SPF の ip4/ip6 に相当する機能を想定）等を追加することで，将来的にこの問題は解決可能であると考えている．

実際の改ざんサイトに使われた環境の入手・構築が難しいため，今回はこのような疑似環境で動作確認を行った．実際の攻撃に関しても，クロスドメインリクエストによって悪意のある Web サイトと通信を行うため，RPF 環境を導入することで安全な Web アクセスを実現できると考えられる．

5.2 ホワイトリスト作成支援機能の評価実験

4.2 節で述べたホワイトリストの作成支援機能の評価実験を行った．実験における Web サイトとしては，アクセス数の多い Web サイト [10] から上位 10 位以内の比較的コ

© 2012 Information Processing Society of Japan

表 1 支援機能の実験結果
Table 1 Experimental result for support function.

Web サイト名	URL	+dn の数	許可されたリクエスト数
Yahoo!JAPAN	http://www.yahoo.co.jp/	12	12
アマゾン	http://www.amazon.co.jp/	9	9
楽天	http://www.rakuten.co.jp/	15	15

ンテンツ数の多い Yahoo! JAPAN と楽天，アマゾンの 3 サイトを選定した．そして，それらの Web サイトのホワイトリストを作成し，そのホワイトリストが対象 Web サイトで正常に動作するかの確認をした．なお，実際の環境の DNS の代わりに，実験用問合せ DNS にホワイトリストを設定し，実験を行った．**表 1** に実験結果を示す．

今回の実験では作成直後に Web アクセスを行い，上記の 3 サイトでホワイトリストに基づくコンテンツへのリクエストを確認した．しかし，5.1 節で述べたように，ページ内のコンテンツが動的に変化する場合や動的に Web ページを生成するサービスの場合には，後々発生するリクエストをその時点で把握することは難しいが，将来的にこの問題は改善可能であると考えている．

また，アマゾンと楽天で閲覧ページから呼び出されている外部ドメインからさらに他のドメインへのリクエストが発生したため，それらのリクエストは遮断された．提案システムでは Web ページ上に別の Web ページを読み込む ⟨iframe⟩ タグのようなコンテンツを動作させるためには，その読み込んだ Web サイトにも RPF 環境（ホワイトリストの設定）が必要である．そのため，今回の実験では，⟨iframe⟩ のようなコンテンツは動作しなかったが，それらの外部ドメインに対してもホワイトリストが設定されていれば，新たにホワイトリストを取得するので，RPF 環境が外部ドメインを含めて普及すれば改善可能である．

5.3 速度評価実験

RPF では通常のブラウジングに加え，RPF 情報取得の DNS 問合せ，コンテンツへのアクセス制御処理が追加されるため，RPF 情報を定義した Web サイトにアクセスした際に，Web ブラウザの処理速度にどの程度の影響が出るのか実験を行った．実験は 5.2 節で使用した環境を用い，通常の Firefox と RPF 対応版 RequestPolicy をインストールした Firefox の両方で Web ページの表示時間を各サイト 10 回測定し，その平均値を結果とした．実験結果を**表 2** に示す．

結果，RPF 対応版 RequestPolicy の Web アクセスでは，通常の Web アクセスと比較して約 9%の処理時間が増加するという結果が得られた．この程度オーバヘッドが増えても実用の範囲にあると考えることができる．

表 2 速度測定結果
Table 2 Result of speed measurement.

Web ブラウザ	Yahoo!JAPAN	アマゾン	楽天
①Firefox	2.5084(s)	4.4109(s)	13.5744(s)
②Firefox (RPF 対応版 RequestPolicy)	2.7292(s)	4.881(s)	14.7169(s)
②−①（RPF 処理時間）	0.2208(s)	0.4701(s)	1.1425(s)
増加率	8.41(%)	10.7(%)	8.41(%)
	平均 9.17(%)		

表 3 ドメイン対 Web サイトの関係
Table 3 Relationships between domains and Web sites.

形態	RPF 適用	形態の説明
1 ドメイン 1 サイト	○	1 つのドメインに対して Web サイトの管理者が一人
多ドメイン 1 サイト	○	複数のドメインを 1 つの Web サイトで運用
多ドメイン 多サイト	○	ユーザ名などがドメインに組込まれている Web サイト
1 ドメイン 多サイト	△	同一のドメインで複数のユーザが各々のサイトを運用

○：適用可
△：適用には制約が必要

5.4 提案手法における適用範囲の評価

ドメイン単位でアクセス制御する RPF の適用可能な Web サイトを明確にするため，ドメインと Web サイトの関係を以下の 4 種類の形態に定義した（**表 3**）．

1 ドメイン多サイトは，同一のドメイン内で複数の Web サイト管理者が各々の Web サイトを管理している形態で，ホスティングサービスやブログサービス等が該当する．RPF の方式では，ドメイン単位で Web サイトにホワイトリストが適用されるため，この形態においては，各々が管理する Web サイトに対する RPF 情報の適切な適用が難しい（各サイトの許可を全体のドメインに紐づけるため，非常に緩いルールを共有することになってしまう）．しかし，ドメイン配下サイト全体に共通して，⟨img⟩ タグは特定サーバ（RPF で許可）へアップロードして埋め込み，⟨script⟩，⟨iframe⟩ タグ等の使用は禁止する等の制限を設けた場合，RPF 情報の作成はサービスの提供者のみで済むため，適用可能となる．

© 2012 Information Processing Society of Japan

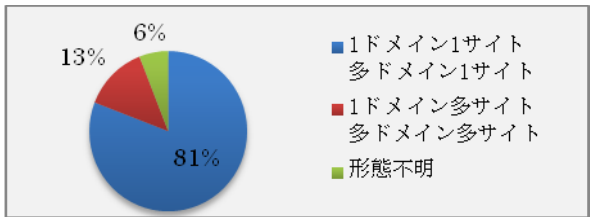

図 10 形態別の被害割合
Fig. 10 Ratio on type of damages.

次に，Gumblar の被害に遭った Web サイト [11] の形態を分類したところ，**図 10** の結果を得た．ただし，被害に遭った Web サイトの形態と，世の中に公開されている Web サイトの形態の割合はイコールとはいえず，この数字がそのまま世の中のサイトへの RPF 適用可否の割合とはならない点に注意が必要である．

結果，1 ドメイン多サイトに適用できないと見た場合，Gumblar の被害に遭った 8 割以上の Web サイトの形態には適用可能だと推測する．この割合が，十分な結果であるとはいい難いが，1 ドメイン多サイトのような形態は，SSL サーバ証明書や Same Origin Policy [12] といった既存のドメイン単位のセキュリティ技術とも相性が悪い [13]．そのため，ドメインと Web サイトのあるべき関係は，セキュリティの観点からドメイン単位での Web サイトの区別ができること理想的なため，RPF の対象は妥当であると考えられる．

6. 提案システムの運用についての考察

6.1 DNS キャッシュポイズニングの脅威について

ネームサーバの問合せ処理は，処理の過程で得たドメインの情報をキャッシュし，キャッシュの生存期間中は他のネームサーバへ問い合わせずに，キャッシュされている情報を返答する．この仕組みによって，DNS の負荷の軽減，問合せ時間の短縮効果が得られる一方で，この仕組みを突く DNS キャッシュポイズニングと呼ばれる攻撃がある．DNS キャッシュポイズニングは，偽の情報をネームサーバにキャッシュさせることで，名前解決を要求してきたクライアントに対して，偽の情報を返答させる攻撃である．

本提案システムの RPF では，DNS に登録した RPF 情報を真とするため，DNS キャッシュポイズニングによって，RPF 情報が改ざんされた場合，Web サイトの不正改ざんによる悪意のある Web サイトとの通信を防げなくなることや，正規コンテンツへアクセスできなくなることが予想される．提案システムでは，この攻撃に対する対抗手段は検討していないが，DNS の既存の対策技術として Source Port Randomization や DNSSEC 等の技術 [14] が存在しており，それらの技術と併用することで RPF 情報の整合性は十分に担保できると考えられる．

6.2 ホワイトリストの適切な更新方法の考察

Web サイト管理者が DNS 管理者に RPF 情報の更新を依頼し，設定した後も DNS キャッシュの生存期間中は更新が反映されない可能性がある．DNS キャッシュの生存期間は DNS のゾーンファイルに設定されている TTL（Time To Live）の値に依存している．そのため，Web サイトの更新にあたって，RPF 情報の更新が必要な場合は，Web コンテンツを更新する TTL 時間以上前に RPF 情報を更新する必要がある．たとえば，RPF 情報の TTL 設定を 3,600 秒（1 時間）としていた場合，Web コンテンツを更新する少なくとも 1 時間以上前に RPF 情報を更新すればよい．

また，RPF 情報の TTL が 86,400 秒（1 日）のように長い場合は，RPF 情報の設定を更新する前に，事前に TTL 値を小さく設定し，最初の TTL 値の時間（86,400 秒）の経過後に TXT レコードの RPF 情報を更新することで，意図した時間に反映させることが可能になる．新しいホワイトリストの浸透がクライアントから確認でき次第，TTL 値を最初の値に戻すことで適切な運用が可能になると考えられる．

6.3 Web サイト管理者と DNS 管理者が異なる場合のホワイトリストの運用

RPF では DNS へホワイトリストの情報を置くが，Web サイト管理者と DNS 管理者が異なる場合（個人の Web サイトの DNS がプロバイダ側にある場合等）RPF 情報を受け渡す運用が必要になる．RPF 情報は Web コンテンツの変化により，更新される可能性があるため，通常 DNS に登録されている Web サーバの正引き，逆引きエントリよりも更新頻度が高くなると推測される．また，Web サイト管理者になりすまして RPF 情報の申請をされてしまうと，改ざんコンテンツを許可するホワイトリストを登録されてしまう可能性がある．これらの条件を満たしつつ，RPF を運用するために，レンタルサーバ事業者が提供する Web ベースの DNS 管理機能 [15], [16] を参考にしたシステムが有効になると考え，検討中である．

7. おわりに

本論文は，近年多く発生している Gumblar に代表される改ざんサイトを用いた Drive-by download 攻撃に対する既存の対策手法の問題点を指摘し，Web サイトに存在するクロスドメインコンテンツに対する Web ブラウザと DNS の連携によるアクセス制御の仕組み，Request Policy Framework を提案した．さらに，RPF の導入により，Gumblar のような改ざんサイトによる不正誘導に用いられるクロスドメインリクエストの制御が可能であることを実験により示した．また，考察では RPF 環境下で懸念される問題事項を検討し，RPF 環境下における運用形態確立の必要性を明らかにした．

© 2012 Information Processing Society of Japan

今後，RPF が Web アクセス方式の標準になることを目指し，ソフトウェアの機能を追加し，使いやすさを向上させる．具体的には RPF 環境の運用形態確立と DNS に対する RPF 情報登録の支援システムの開発を行っていく．そのうえで，プログラムを一般ユーザに配布し利用者を増やすとともに，RPF 対応サイトの拡大を図る．

謝辞 本研究の初期の段階において検討に参加いただいた元東京電機大学学生髙木翔太氏に深謝申し上げる．また，研究の発展のために貴重なご意見をいただいたコンピュータ疫学研究会のメンバ等多くの方々に感謝申し上げる．

参考文献

[1] Mozilla Corporation: NoScript, Add-ons for Firefox, available from ⟨https://addons.mozilla.org/ja/firefox/addon/noscript/⟩ (accessed 2012-03-26).
[2] Samuel, J. and Zhang, B.: RequestPolicy: Increasing Web Browsing Privacy through Control of Cross-Site Requests, *Proc. Privacy Enhancing Technologies Symposium*, Vol.5672, pp.128–142 (2009).
[3] 株式会社セキュアブレイン：gred セキュリティサービス，入手先 ⟨http://www.securebrain.co.jp/products/gred/index.html⟩ （参照 2011-11-25）.
[4] at+link 専用サーバサービス：Gumblar 亜種にも有効！サイト改竄チェックサービス Gumblar Watch，入手先 ⟨http://www.at-link.ad.jp/topics/news/news-20100225.html⟩ （参照 2011-11-25）.
[5] Chia-Mei, C., Wan-Yi, T. and Hsiao-Chung, L.: Anomaly Behavior Analysis for Web Page Inspection, *Proc. 2009 1st International Conference on Networks & Communications* (*NETCOM '09*), pp.358–363 (2009).
[6] 安藤槙悟，寺田真敏，菊池浩明，趙 晋輝：通信の遷移に着目した不正リダイレクトの検出による悪性 Web サイト検知システムの提案，研究報告コンピュータセキュリティ（CSEC），Vol.2011-CSEC-54, No.32, pp.1–6 (2011).
[7] 阪井哲晴，寺田真敏，土居範久：Web サイトに埋め込まれたインジェクション攻撃の痕跡検知システムの提案，研究報告コンピュータセキュリティ（CSEC），Vol.2010-CSEC-48, No.9, pp.1–7 (2010).
[8] 上松晴信，名坂康平，酒井崇裕，西垣正勝：相補的な Web 感染型マルウェア検知方式の提案，研究報告コンピュータセキュリティ（CSEC），Vol.2011-CSEC-52, No.53, pp.1–6 (2011).
[9] Internet Engineering Task Force (IETF), Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1, available from ⟨http://www.ietf.org/rfc/rfc4408.txt⟩ (accessed 2011-11-28).
[10] 日本の人気サイトランキング 500，入手先 ⟨http://akimoto.jp/japan/⟩ （参照 2012-03-17）.
[11] セキュアブレイン：セキュアブレイン gred セキュリティレポート Vol.7【2010 年 1 月分統計】図表 4-1 Gumblar 被害サイト内訳（2010 年 1 月），入手先 ⟨http://www.securebrain.co.jp/about/news/2010/02/gred-report7.html⟩ （参照 2012-03-08）.
[12] 株式会社技術評論社：Same-Origin ポリシーと迂回技術，入手先 ⟨http://gihyo.jp/dev/serial/01/web20sec/0002⟩ （参照 2012-03-08）.
[13] 高木浩光：共用 SSL サーバの危険性が理解されていない，高木浩光@自宅の日記，入手先 ⟨http://takagi-hiromitsu.jp/diary/20100501.html⟩ （参照 2012-03-08）.
[14] 藤原和典：DNSSEC の現状，オープンポリシーフォーラム，入手先 ⟨http://venus.gr.jp/opf-jp/opm15/jpopm15-08.pdf⟩ （参照 2011-11-11）.
[15] さくらインターネット：さくらで取得したドメインのゾーン編集，入手先 ⟨http://support.sakura.ad.jp/manual/dom/zone.html⟩ （参照 2012-03-21）.
[16] レンタルサーバー・ホスティングの【WebARENA（ウェブアリーナ）】：DNS アウトソーシング，入手先 ⟨http://web.arena.ne.jp/suitex/spec/domain/dns.html⟩ （参照 2012-03-21）.

植村 崇史

2011 年東京電機大学未来科学部情報メディア学科卒業．同年 4 月東京電機大学大学院未来科学研究科情報メディア学専攻博士前期課程入学．現在，Web セキュリティに関する研究に従事．

小須田 優介 （正会員）

2008 年東京電機大学工学部第二部情報通信工学科卒業．同年 NEC ソフト株式会社入社．大規模ミッションクリティカルシステムの構築に従事．平成 20 年度情報処理学会山下記念研究賞受賞．

佐々木 良一 （フェロー）

1971 年 3 月東京大学卒業．同年 4 月日立製作所入社．システム開発研究所にてシステム高信頼化技術，セキュリティ技術，ネットワーク管理システム等の研究開発に従事．2001 年 4 月より東京電機大学工学部教授，2007 年 4 月より未来科学部教授．工学博士（東京大学）．1998 年電気学会著作賞受賞．2002 年情報処理学会論文賞受賞．2007 年総務大臣表彰等．著書に，「IT リスクの考え方」岩波新書 2008 年等．日本セキュリティ・マネジメント学会会長，内閣官房情報セキュリティセンター情報セキュリティ補佐官．

© 2012 Information Processing Society of Japan