Ajax アプリケーション診断方式の提案と評価

祢宜 知孝 †　　河内 清人 †　　北澤 繁樹 †　　藤井 誠司 †

† 三菱電機株式会社　情報技術総合研究所
247-8501 神奈川県鎌倉市大船 5-1-1

あらまし　セキュアな Web アプリケーション開発のためには，Web アプリケーションセキュリティ診断ツールを用い，安全性を確認する必要がある．しかし，Ajax 技術の普及により，非標準フォーマットでパラメータ送信が行われるため，従来のツールでは診断ができない．また，フレームワークを利用した開発が広まり，診断不要なフレームワーク内で処理されるパラメータまでもが送信され，診断対象となっている．本稿では，使用されているフレームワークを特定し，フレームワークに関する情報を利用した．フレームワークに関する情報を用いて非標準フォーマットへの対応と診断対象パラメータの絞込みを行うことにより，Ajax アプリケーションに対する診断を可能とした．

A method for security scanning of Ajax application

Tomonori Negi†　　Kiyoto Kawauchi†　　Shigeki Kitazawa†　　Seiji Fujii†

†Information Technology R&D Center, Mitsubishi Electric Corporation
5-1-1 Ofuna Kamakura-Shi Kanagawa 247-8501 Japan

Abstract　When we develop a secure Web application, we use a Web application vulnerability scanner to verify safety. But as an Ajax technology became widely used, conventional scanners don't work well. It's because Ajax applications transmit parameters in informal format. In addition, it became popular to use Web application framework. Frameworks send additional parameters which are for frameworks, and conventional scanners target them. In this paper, we show we're able to scan Ajax applications, through identifying the used framework, supporting informal format parameters and deriving targets from many parameters with information of framework.

1　はじめに

近年，リッチで使い易い UI(User Interface) や高速な応答性から Ajax 技術を用いた Web アプリケーションが普及してきている．Ajax は，非同期通信によって Web ページの一部を更新することで，効率の良いページ更新と使い易い UI を実現した．

一方，セキュリティの観点では，Web アプリケーションに関する脆弱性の件数が増加している [2]．そこで，Web アプリケーションの開発者は，網羅的に脆弱性の有無を確認するために Web アプリケーションセキュリティ診断ツールを用いる．現行のツールには，IBM の Rational AppScan [1] や HP の WebInspect [6] などがある．

また，アプリケーションの開発を効率化するために，Ajax Web アプリケーションフレームワークを用いた開発が主流となってきている．これは，フレームワークを用いることにより，セッション管理やページ遷移制御といった共通機能はフレームワークが処理するため，開発者は本来行いたい処理の実装に専念できるからである．

そこで本稿では，従来の Web アプリケーションセキュリティ診断ツールが抱える課題とフレームワークの使用によって生じる課題を提示し，解決方法を提案する．本稿で提案する方式は，Web アプリケーションセキュリティ診断時に，使用されているフレームワークを特定し，フレームワーク情報を用いるこ

とで Ajax アプリケーションの診断を可能とする．

2　課題

本節では，Ajax アプリケーションに対するセキュリティ診断の課題を述べる．まず，従来の診断ツールは，一般的に Ajax を多用したアプリケーションの自動解析をうまく行えないという課題がある [8]．Ajax による通信では，独自フォーマットでパラメータを送信することが可能であることが理由の一つである．

また，フレームワークを用いた Web アプリケーションに対する診断では，フレームワーク内で処理されるパラメータに対する無駄な診断が行われてしまう．フレームワークを用いた Web アプリケーションでは，HTTP リクエストに，本来アプリケーションが必要とするパラメータの他に，フレームワークが必要とするパラメータが多く付加される．そして，診断ツールは，フレームワークが付与したパラメータまでをも診断対象とし，擬似攻撃を加えて診断を行う．ところが，提供元のベンダやセキュリティコミュニティによってフレームワークの安全性は十分検証されており，フレームワークが付与するパラメータに対して改めて診断を行う必要はない．従って，フレームワークが付与するパラメータに対する診断は，無駄な診断処理となる．

3　提案方式

筆者らは，第 2 節で挙げた課題を解決する方法を提案する．提案する方式は，診断対象から取得される Web ページから Web アプリケーションで使用されているフレームワークを特定し，そのフレームワークに関する情報を利用することで，独自フォーマットへの対応や，診断不要なパラメータの絞り込みを行う．本節では，提案方式について述べる．

3.1　診断手順

本節では，診断ツールによる診断手順を述べる．診断手順は，事前解析，診断，検証の 3 フェーズで構成される．

3.1.1　事前解析フェーズ

(1)．診断対象である Web ページに移動する．この時，ツールは，Web サーバとの通信を記録しておき，第 3.2 節に述べる方法で使用されているフレームワークを特定する．

(2)．ユーザにブラウザを通じて Web アプリケーションを操作をさせ，ブラウザによって生成される HTTP リクエストを記録する．記録された HTTP リクエストは，擬似攻撃 HTTP リクエストを生成するための雛形となる．

(3)．雛形として記録した HTTP リクエストからパラメータ群を抽出する．この時，ボディ部のデータが標準的なフォーマットに合致しない場合には，ツールは，フレームワークの情報を基に第 3.3 節に述べる方法でパラメータを抽出する．

3.1.2　診断フェーズ

(1)．抽出されたパラメータ群からパラメータ (P) を一つ取り出す．そして，P が診断対象かどうか，第 3.4 節に述べる方法で判断する．

(2)．(3)〜(6) の手順に従い P に対して擬似攻撃データを挿入した診断用 HTTP リクエストを生成する．

(3)．P が診断対象と判定された場合，既存の技術を用いて擬似攻撃データを生成する．

(4)．擬似攻撃データが適当であるか否かをを第 3.4 節で述べる処理によって記録されるパラメータ値書式指定文字列にマッチするか否かで判断する．

(5)．マッチする場合，P のパラメータ値を第 3.4 節で述べる処理によって記録されるオフセット情報を用いて攻撃データを設定する．

(6)．パラメータ群を再び HTTP リクエスト内に格納する．雛形の HTTP リクエストが標準なフォーマットであった場合には，標準フォーマットに従い抽出されたパラメータを格納する．一方，独自のフォーマットであった場合には，第 3.3 節で述べる処理によって付加情報として記録された抽出順序に従って HTTP リクエストボディ部に格納する．この時，擬似攻撃データを含まないパラメータに対しては，付加情報として記録されたマッチした全文字列を格納する．そして，擬似攻撃データを含む場合には，付加情報として記録されたオフセット情報を用いて攻撃データを格納する．

(7)．擬似攻撃データを挿入した診断用 HTTP リクエストを送信する．

3.1.3　検証フェーズ

(1)．診断時に生成された HTTP リクエストに対する応答として，HTTP レスポンスを受信する．

(2)．受信した HTTP レスポンスの解析することで，セキュリティホールの有無を検査する．

提案方法では，フレームワーク特定技術，独自フォーマットへの対応技術，パラメータ絞込み技術が

表 1: フレームワーク識別情報例

フレームワーク名	照合箇所		パターン文字列
	ファイルタイプ	詳細指定	
DWR	JavaScript	1,50	dwr¥.engine
DOJO	JavaScript	1,20	dojotoolkit
WICKET	HTML	//*[string-length(@href) > 0]	wicket

表 2: フォーマット定義情報例

フレームワーク名	パラメータフォーマットパターン文字列	パラメータ名書式指定パターン文字列	パラメータ値書式指定文字列
DWR	^{{NAME}}={{VALUE}}	[^¥t¥n¥r¥f¥v]+	[^¥t¥n¥r¥f¥v]*

要である．第 3.2 節〜第 3.4 節では，これらの技術に関して説明する．

3.2 フレームワーク特定

フレームワークの特定は，HTML ファイルや同 HTML ファイルからインポートされている JavaScript フィル内の文字列と，フレームワーク識別情報 (表 1) であるシグニチャとを照合することで行う．フレームワーク特定が行われるフローを図 1 に示す．

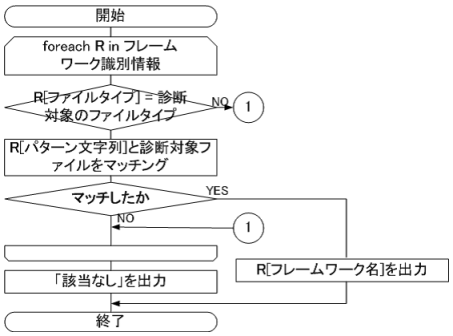

図 1: フレームワーク特定処理フロー

表 1 に示すとおり，一つのシグニチャは,「フレームワーク名」,「照合箇所」,「パターン文字列」で構成される．さらに，照合箇所は,「ファイルタイプ」と「詳細指定」で構成される．ファイルタイプには HTML，もしくは JavaScript が指定される．ファイルタイプが HTML である場合には，詳細指定に，XPath[5] による照合対象ノード条件と照合箇所が指定される．また，ファイルタイプが JavaScript である場合には，詳細指定に,「探索開始行」,「終了行」が指定される．そして，フレームワーク特定処理は，指定されたファイルタイプと同一タイプのファイルに対して，詳細指定で指定された箇所に，正規表現で表されたパターン文字列に該当する文字列が存在するか否かを確認することで行われる．

3.3 独自フォーマットへの対応

独自フォーマットへの対応は，フレームワーク情報を基に，HTTP リクエストボディ部に独自フォーマットで格納されたパラメータを抽出すると同時に，診断用 HTTP リクエストのボディ部を生成する為に必要な情報を記憶することで実現する．パラメータの抽出が行われるフローを図 2 に示す．

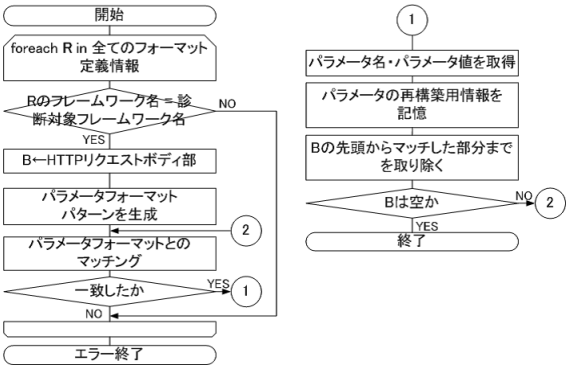

図 2: パラメータ抽出機能処理フロー

フォーマット定義情報は,「フレームワーク名」,「パラメータフォーマットパターン文字列」,「パラメータ名書式指定パターン文字列」,「パラメータ値書式指定文字列」で構成される（表 2 参照）．パラメータフォーマットパターン文字列は,「パラメータ名」と「パラメータ値」のプレースホルダを含んだ正規表現文字列である．パラメータ名とパラメータ値のプレースホルダは，それぞれ {{NAME}}，{{VALUE}} とする．さらに，パラメータ名書式指定文字列は，パラメータ名の書式を表した正規表現であり，パラメータ値書式指定文字列はパラメータ値の書式を表した正規表現である．

以降では，処理の流れを説明する．始めに，ツールは，特定されたフレームワークに対応するフォーマット定義情報を取り出す．そして，フォーマットパターン文字列中の {{VALUE}}，{{NAME}} をパラメータ名書式指定パターン文字列，パラメータ値書式指定パターン文字列で置換し，パラメータフォーマットパターンを生成する．次に，ツールは，生成したパラメータフォーマットパターンと HTTP リクエストボディ部とをマッチングする．マッチした場合，ツールは，パラメータ名，パラメータ値と共

表 3: 診断対象パラメータ判定条件例

フレームワーク名	パラメータ格納場所	パラメータ名パターン文字列	パラメータ値パターン文字列	パラメータ値書式指定文字列	対象/対象外
DWR	BODY	c¥d+-.+¥d+	.+:{{VALUE}}	[^¥t¥n¥r¥f¥v]*	対象
DWR	BODY				対象外

に，付加情報として「抽出順序」,「マッチした全文字列」,「パラメータ名・パラメータ値としてマッチした部分の文字列先頭からのオフセット」を記憶する．

付加情報は，前述のとおり独自フォーマットのボディ部を含んだ診断用 HTTP リクエストの生成時に利用される．抽出順序は，パラメータの格納順序として，マッチした全文字列は，擬似攻撃データを含まないパラメータとして，オフセット情報は，パラメータ値を擬似攻撃データで置換するための位置情報として使用される．

3.4 パラメータの絞込み

パラメータの絞込みは，特定されたフレーム枠に対応する診断対象パラメータ判定条件を取得し，診断対象の当否を判断することで実現する．パラメータの絞込みが行われるフローを図 3 に示す．

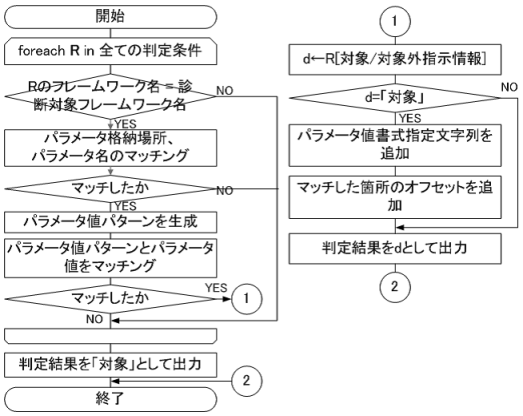

図 3: パラメータ絞込み機能処理フロー

診断対象パラメータ判定条件は,「フレームワーク名」,「パラメータ格納場所」,「パラメータ名パターン文字列」,「パラメータ値パターン文字列」,「パラメータ値書式指定文字列」，及び「対象/対象外指示情報」で構成される（表 3 参照．パラメータ格納場所には，URL，Cookie ヘッダ，またはボディ部が指定される．パラメータ名パターン文字列には，正規表現によってパラメータ名に対する条件が指定される．パラメータ値パターン文字列には，擬似攻撃データ格納箇所を示すプレースホルダ {{VALUE}} を含んだ正規表現によって，パラメータ値に対する条件が指定される．パラメータ値書式指定文字列には，正規表現によって，前述のプレースホルダに格納される文字列の書式が指定される．そして，対象/対象が指示情報には，条件に合致したパラメータの診断対象の当否が指定される．

以降では，処理の流れを説明する．始めに，ツールは，フレームワークに対応する診断対象パラメータ判定条件を取得する．そして，ツールは，判定条件内のパラメータ格納場所情報とパラメータ名パターン文字列が，パラメータ格納場所，パラメータ名に該当するかを判断する．該当する場合には，ツールは，パラメータ値パターン文字列内のプレースホルダをパラメータ値書式指定文字列で置換したパラメータ値パターンを生成し，パラメータ値にマッチするかを確認する．マッチした場合，ツールは，対象対象外指示情報を確認し，診断対象であれば，擬似攻撃データ設定のために，付加情報として「パラメータ値書式指定文字列」と「同文字列にマッチしたオフセット情報」を記憶する．

パラメータ値書式指定文字列は，生成された擬似攻撃データが書式に従っているか否か確認するために使用される．また，オフセット情報は，パラメータ値として擬似攻撃データを設定するための位置情報として使用される．

4 評価

筆者らは，第 3 節で提案した方式を適用した診断ツールを試作し，評価を行った．本節では，試作したツールを紹介したうえで，本提案方式の評価を行う．評価を行う項目は，以下の 3 点である．

1. フレームワークの特定を行えるか
2. 独自フォーマットのパラメータを認識できるか
3. パラメータ絞込みによる診断対象パラメータ数の削減効果

4.1 試作ツール

本提案方式を適用したツールを Mozilla Firefox 3 上のアドオンとして試作した．本ツールは，Firefox のツールメニューから起動される（次頁図 4）．

4.2 評価環境

筆者らは，提案方式による改善効果を確認するため，フレームワークを用いた Web サイトを構築した．評価に用いたフレームワークと Web アプリケーションを，表 4 に示す．ただし，Dojo は，Web アプ

表 5: 評価結果

使用されている フレームワーク名	特定されたフレームワーク名	総パラメータ数	要診断パラメータ数	絞込み結果	診断漏れ数	削減率 (%)
DWR	DWR	259	46	46	0	100
Wicket	Wicket	46	22	22	0	100
Dojo	Dojo	12	6	6	0	100

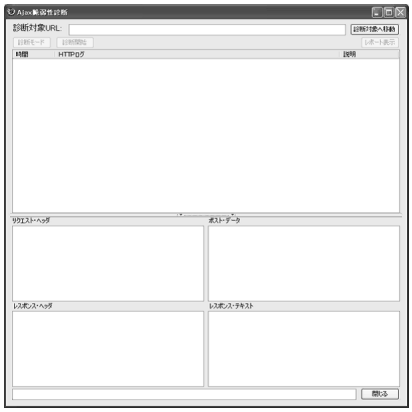

図 4: 診断ツール起動画面

リケーションフレームワークである struts 2.1.6 から呼び出す．評価に用いた設定は，それぞれ表 1〜表 3 の設定例に基づいて行った．

表 4: 評価に用いた Ajax フレームワーク

フレームワーク名	Web アプリケーション名
DWR	TuduLists-DWR v.2.3
Wicket	Wicket-1.3.6 Examples
Dojo	Struts 2.1.6 Showcase (Ajax Examples)

4.3 評価結果

4.3.1 フレームワーク特定

試作ツールを用いたフレームワーク特定結果を表 5 に示した．同結果より，正しくフレームワークを特定できていることが判る．

4.3.2 独自フォーマットへの対応

Wicket と Dojo では，標準フォーマットでパラメータが送信される．そのため，DWR を対象に評価を実施した．DWR 上で Web アプリケーションを構築した場合，図 5 のような DWR 固有のフォーマットで HTTP リクエストが交信される．従来の診断ツールは、このような独自フォーマットで送信されるパラメータを抽出できず，診断を行えなかった．本試作ツールでは，フレームワークが DWR であることを認識し，さらに送信されるパラメータ全てを表 6 のとおり抽出することができた．この内，“ c0-param0=string: ··· ”から“ c0-param5=string:admin ”までが，Web アプリケーションで使用されるパラメータである．

```
POST /tudu-dwr/secure/dwr/call/plaincall/
Multiple.2.dwr HTTP/1.1
···(略)···
Cache-Control: no-cache

  callCount=2
  ···(略)···
  c0-id=0
  c0-param0=string:8aca835d22e258770122e2588···
  c0-param1=string:test
  ···(略)···
  c0-param5=string:admin
  ···(略)···
```

図 5: HTTP リクエスト例

表 6: パラメータ抽出結果

パラメータ	抽出結果	パラメータ値
callCount	○	2
page	○	/tudo-dwr/secure/show···
httpSessionID	○	E2FEFDED6D3606FA···
scriptSessionId	○	D52BC1155067585AC1···
c0-scriptName	○	todos
c0-methodName	○	addTodo
c0-id	○	0
c0-param0	○	string:8aca835d22e258···
c0-param1	○	string:test
c0-param2	○	string:1
c0-param3	○	string:08%2F04%2F2009
c0-param4	○	string:Test%20Notes
c0-param5	○	string:admin
c1-scriptName	○	todos
c1-methodName	○	forceGetCurrentTodoLists
c1-id	○	1
batchId	○	3

4.3.3 パラメータの絞込み

フレームワークを用いた Web アプリケーションにおいて診断を要するパラメータ数と提案方式によって絞り込まれたパラメータ数とを表 5 に示す．本提案方式によって，診断漏れを起こすことなく，診断対象パラメータを，診断を要するパラメータに限定できていることが判る．本評価では，DWR，Wicket，Dojo のいずれにおいても不要なパラメータを正確かつ確実に削減している．

5 考察

本稿で提案した方法は，Ajax アプリケーションの診断に有効である．提案方法により，フレームワー

ク独自のデータフォーマットの解析は無論、診断漏れを起こすことなく，診断対象パラメータを診断を要するパラメータに限定することが可能である．従って，フレームワークによって付与されるパラメータが多いほど，パラメータの絞込み効果により，無駄な診断処理を省略することが可能である．

本稿ではフレームワークの特定やパラメータ送信フォーマットの把握，診断対象パラメータの絞込みのための設定を事前に行っている．フレームワークの特定はシグニチャマッチングで行っているため，新たなフレームワークが登場した時にシグニチャファイルをアップデートすることになる．だが，誤検知のないシグニチャの開発には，十分な検証が必要であるという課題が残る．

6 関連研究

Web アプリケーションに対するセキュリティ診断に関する研究は従来から数多く行われている [3] [4] [7][9]．例えば，[9] は，ホワイトボックステストをベースとした診断技術であり，ブラックボックステストをベースとする筆者らの研究とは根本的に異なる．[3] は，HTML や JavaScript ファイルなどの解析を行い，自動全探索を行うことを可能とした診断技術である．本技術は，Ajax アプリケーション上で発生しうる全ての HTTP リクエストを特定できないという課題を考慮しておらず，筆者らの研究とは異なる．また，[4] は，診断者が診断シナリオを作成してブラックボックステストを行う技術であり，Ajax アプリケーション上で発生しうる全ての通信をシナリオに記述する必要がある点で，筆者らの研究とは異なる．[7] は，Web サーバにエージェントを組み込んだうえで，ブラックボックステストを行う技術であり，エージェントレスである筆者らの研究とは異なる．このように、様々な研究がなされているが，筆者らのような研究は行われていない．

また，AppScan や WebInspect なども Ajax への対応を行っている．しかし，商用ソフトウェアであり，Ajax への対応方法が明らかではない．そのため，筆者らの研究と比較を行うことが不可能である．

7 おわりに

本稿では，Web アプリケーションセキュリティ診断時に，フレームワークを特定し，フレームワーク情報を用いることで Ajax アプリケーションの診断を可能とする方法を提案した．そして，サンプルプログラムやフリーのプログラムに対して評価を行い，提案方式により Ajax アプリケーションの診断が可能になることはもちろん，診断対象パラメータを正しく絞り込めることを示した．提案方式は，予めフレームワークを識別するためのルールと、フレームワーク毎のパラメータフォーマット定義情報、診断対象パラメータの判定条件を設定しておく．そして，診断時にこれらの情報を元にフレームワークの特定，及びパラメータの抽出 · 絞込み · 診断用 HTTP リクエストの生成を行う．

今後は，Web アクセスによって取得される HTTP や JavaScript ファイルからではなく、診断対象サイトの内部のファイル構成などからフレームワークを特定する技術の開発に取り組む．これは，内部のファイル構成などからフレームワークを特定し，その上でデータフォーマットや診断対象パラメータの決定に本提案方式を適用することで，設定項目の削減が可能と考えているためである．

参考文献

[1] http://www-01.ibm.com/software/awdtools/appscan/.

[2] IBM Co. IBM Internet Security Systems X-Force 2007 Trend & Risk Report. Technical report, IBM Global Services, January 2009.

[3] Yao-Wen Huang, Chung-Hung Tsai, Tsung-Po Lin, Shih-Kun Huang, D. T. Lee, and Sy-Yen Kuo. A testing framework for Web application security assessment. *Comput. Netw.*, Vol. 48, No. 5, pp. 739–761, 2005.

[4] Xiaoping Jia and Hongming Liu. Rigorous and Automatic Testing of Web Applications. In *In 6th IASTED International Conference on Software Engineering and Applications (SEA 2002)*, pp. 280–285, 2002.

[5] XML Path Language (XPath) Version 1.0 W3C Recommendation 16 November 1999, November 1999. http://www.w3.org/TR/xpath.

[6] https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-201-200^9570_4000_100__.

[7] 坂田匡通, 中山弘二郎, 西木健哉, 石崎健史, 富坂稔. Web システムにおけるセキュリティ検査手法の検討 (セキュリティ). 情報処理学会研究報告. マルチメディア通信と分散処理研究会報告, Vol. 2005, No. 111, pp. 43–48, November 2005.

[8] 吉濱佐知子, 石田愛, 浦本直彦. Web2.0 アプリケーションにおける代表的な攻撃手法とその対策. 情報処理学会誌, Vol. 50, No. 1, pp. 44–54, January 2009.

[9] 小黒博昭, 市原尚久, 道坂修. Web アプリケーション脆弱性発見のためのソースコード診断ツールの開発. 情報処理学会研究報告. CSEC, [コンピュータセキュリティ], Vol. 2008, No. 45, pp. 97–102, May 2008.