6S-04

URL 埋め込み型クロスサイトスクリプティング攻撃の特徴検出

海寶 貴人[*1]　松田 健[*2]　園田 道夫[*3]　趙 晋輝[*4]

中央大学 理工学部 情報工学科 [*1] [*4]
静岡理工科大学 総合情報学部 コンピュータシステム学科 [*2]
サイバー大学 IT 総合学部 [*3]

要約　クロスサイトスクリプティング（以下, XSS）攻撃はサイバー攻撃の一種であり, フィッシング詐欺や Cookie の盗難によるなりすましなどの被害を招いており, 社会に悪影響を与えている. 本研究では, XSS 攻撃に含まれる記号の出現位置に基づいて攻撃の特徴を抽出する既存のアルゴリズムに改良を加え, XSS 攻撃を誘導するような URL と正常な URL のみにデータの対象を絞ることで攻撃検知する手法を提案した.

キーワード　クロスサイトスクリプティング, 特徴検出, 機械学習

1　序論

近年, ウェブページの開発技術が向上したことで, ユーザの入力に応じて出力を行う動的なウェブページが容易に開発できるようになった. これに伴い, 電子商取引や SNS, インターネットバンキングなどのサービスが普及し, ユーザがウェブブラウザ上で重要な個人情報（名前, 住所, クレジットカード番号など）を入力する機会が増加している. 個人情報の入力を促す動的なウェブページなどに含まれる脆弱性を利用し, 機密情報などを盗み出すサイバー攻撃が存在する. その一つとして, サイト間を横断して悪意のあるスクリプトを埋め込むクロスサイトスクリプティング (以下, XSS) 攻撃が挙げられる. XSS 攻撃対策のため, 主要なブラウザにはアドレス欄に入力された URL が XSS 攻撃である場合に通信を遮断する機能や, サイトごとに XSS の脆弱性を判定する機能が含まれている [1]. ただし, これらの機能を保護していないブラウザやアプリケーションでは XSS 攻撃の被害を受ける危険性がある. また, XSS の検出については, URL のパラメタに攻撃が埋め込まれているタイプの XSS に対象を絞り込む場合でさえも, 正常な URL のデータとの差異をデータに含まれる記号や文字の出現頻度の情報だけで区別することは容易でないデータが存在する. そこで, 本研究では, 攻撃データと正常データに含まれる記号の出現位置の情報に注目した特徴抽出法を用いて攻撃と正常それぞれの特徴を抽出し, 従来手法よりも容易に実装可能な XSS の自動検出方法について検討する.

Feature extraction of Embedded URL Cross-Site Scripting Attacks

[*1] Kaiho Takahito, Chuo University
[*2] Matsuda Takeshi, Shizuoka Institute of Science and Technology
[*3] Sonoda Michio, Cyber University
[*4] Chao Jinhui, Chuo University

[2] T.Matsuda, D.Koizumi, M.Sonoda: *Cross site scripting attacks detection algorithm based on the appearance position of characters*, MIC-CCA, pp.65-70, 2012.

2　XSS

XSS 攻撃は, 文字入力を求めるウェブ上のサイトや, アプリケーションに含まれるセキュリティ上の不備（脆弱性）を利用し, 不正な動作を誘因するスクリプトを埋め込むことで実現するサイバー攻撃である. 攻撃方法は, 主に以下の 3 つの種類の攻撃に分類される. いずれの攻撃においても, ユーザが悪質なリンクにアクセスしない限り被害は発生しない. 本研究では, 反射型 XSS にのみ攻撃文の対象を制限している.

2.1　反射型 XSS

不正なリンクを添付したメールなどをユーザに送信し, ユーザがリンクにアクセスするように仕向ける攻撃は反射型 XSS とよばれる. リンクにアクセスしたユーザは悪意のあるスクリプトを実行し, 被害が発生する.

2.2　蓄積型 XSS

SNS や掲示板, ブログなど, 不特定多数のユーザが閲覧するサイトに悪意のあるスクリプトを埋め込み, そのサイトにアクセスしたユーザのブラウザでスクリプトを実行させる攻撃は蓄積型 XSS とよばれる.

2.3　DOM Based XSS

ウェブブラウザが構築する HTML の DOM 機能を用いて, サーバを介さずにスクリプトを実行させる攻撃は DOM Based XSS とよばれる. DOM Based XSS では, HTML に直接スクリプトを埋め込まずクライアントサイドでスクリプトの処理を行う. そのため, 2.1, 2.2 で述べた攻撃手法と異なり, サーバサイドで攻撃文を検知することができない.

3　XSS 攻撃に対する既存の対策手法

3.1　特定記号のエスケープ

XSS 攻撃が発生する要因として, HTML タグや JavaScript などで記された文字列をサーバ, クライアントがそのまま実行してしまうことが挙げられる. そこで, データが入力された際, または HTML が生成された際に特徴的な記号のエスケープを行うことで, これらの文字列を置き換えスクリプトが実行されないようにする. エスケープを行うことで XSS 攻撃を無効化することが可能であるが, アプリケーション内に機能を実装する過程で脆弱性が生じる可能性がある.

3.2　単語の出現位置に基づく攻撃検知アルゴリズム

Web 検索に用いられる関連単語抽出アルゴリズムを基にした, XSS 攻撃についての特徴量をあるデータについて算出するアルゴリズムが存在する [2]. このアルゴリズムは従来の対策手法とは異なり, 大規模な計算を

行わない. 入力, および出力は以下のとおりである.

- 入力
  - データ集合
    $L = \{l_i\}(i = 1, 2, ..., I)$
  - XSS 攻撃に多く含まれる特殊な記号の集合
    $S = \{s_j\}(j = 1, 2, ..., J + 1)$
- 出力
  - 各記号ごとの評価値
    $a_j(j = 1, 2, ..., J + 1)$
  - 各データごとの評価値
    $x_i(i = 1, 2, ..., I)$

最適なしきい値 α を定め, $x_i > \alpha$ ならば, データ l_i は攻撃文であるとする.

具体的なアルゴリズムは以下のとおりである.

1. データ l_i の i_0 文字目の記号を l_{i,i_0} とする. データ l_i 内に s_j が含まれてる場合, 出現順に $s_j^{(1)}, s_j^{(2)}, ..., s_j^{(a)}, ..., s_j^{(A_j)}$ として, 以下の式を計算する.
   $E(s_j^{(a)}, l_{i,i_0}) = \frac{|l_i|-|k-i_0|}{2^{\frac{1}{|l_i|}\{|l_i|(|l_i|+2i_0-1)-2i_0(i_0-1)\}}}$
2. 前過程でもとめた値を用いて, データ l_i における s_j の評価値を計算する.
   $E(s_j, l_i) = A_j \frac{1}{|l_i|} \sum_{a=1}^{A_j} \sum_{i_0=1}^{|l_i|} E(s_j^{(a)}, l_{i,i_0})$
3. 各データごとの記号の重要度の平均を計算し, 各記号ごとの評価値を計算する.
   $a_j = \frac{1}{N} \sum_{i=1}^{N} E(s_j, l_i)$
4. l_i に含まれる s_j の数を $|s_j|$ として, 各データごとの評価値を計算する.
   $x_i = \sum_{j=1}^{J+1} |s_j| \hat{a_j}$
   $$\hat{a_j} = \begin{cases} a_j & (if\ l_i\ includes\ s_j) \\ 0 & (otherwise) \end{cases}$$

また, 最適なしきい値 α^* は以下の式を最大にする値である. しきい値 α を任意に定めることにより得られる攻撃文検知率 P_A, 正常文検知率 P_N を用いる.

$$\int_0^1 \{\beta \cdot P_A + (1 - \beta) \cdot P_N\} d\beta$$

4 提案手法

本研究では, 3.2 で述べたアルゴリズムの拡張を行う.

4.1 検出範囲の拡張

既存の研究 [2] では, 検出範囲をパラメタのみに限定していたが, 実際の攻撃文である URL には複数のパラメタが含まれるため, パラメタの特徴量を URL 単位の特徴量に換算するための計算が必要となる. そこで, URL 単位の特徴量算出を考える.

4.2 正常文の特徴を加味した特徴抽出

既存のアルゴリズム [2] で用いられている特徴記号は以下の 32 文字である.

”	>	/	<	(space)	=	’	:
.	(	)	-	;	\	&	{
}	#	+	!	,	@	?	[
]	_	~	*	|	^	%	$

以上の特徴記号群を用いたまま URL 単位の XSS 攻撃の検知を行うことは難しい. なぜならば, 「=」(イコール), 「?」(クエスチョンマーク), 「&」(アンド) などの記号は, 通常の URL にもよく含まれるからである. そこで, 攻撃文に多く含まれ正常文には通常含まれない以下の 4 文字を特徴文字として扱う.

”	’	<	>

4.3 交差確認

既存のアルゴリズムを用いてもとめた結果は, 訓練データに強く依存するため, 交差確認を行う.

交差確認とは, すべてのデータを n 個の集合に分割し, うち 1 つの集合を訓練データ, その他の集合をテストデータとして扱う操作を, 訓練データとする集合を変えながら何回も繰り返すことでデータ, また結果の正当性を確かめるものである.

5 結果

攻撃文検知率 P_A, 正常文検知率 P_N は以下のとおりである. 攻撃 URL 数は 200[3], 正常 URL 数は 150 である. データの分割数は 5 とした.

表 1 グループごとの P_A, P_N, およびそれぞれの平均

訓練データ	P_A	P_N
グループ 1	97.5%	97.5%
グループ 2	98.8%	97.5%
グループ 3	98.8%	98.3%
グループ 4	96.2%	98.3%
グループ 5	97.5%	98.3%
平均	97.8%	98%

6 結論

本研究では, 既存の XSS 攻撃検知アルゴリズムで用いられている特徴文字を削減することで, URL 単位の XSS 攻撃検出を高い水準で行えることがわかった. 発展的課題として 3 つの課題が挙げられる. 1 つ目は, 本研究で提案した手法を実用化する際の実装をどのように行うかである. 2 つ目は, XSS 攻撃に多く含まれる文字の自動学習である. 3 つ目は, 蓄積型 XSS, DOM Based XSS などの攻撃に対応するための拡張である.

謝辞

今回の研究を進めるにあたり数々のご助言をいただいた趙 晋輝先生, サイバー大学の園田 道夫先生, 静岡理工科大学の松田 健先生, そして趙研究室の皆様に深く御礼申し上げます.

参考文献

[1] ” クロスサイト スクリプト フィルター”. Microsoft Windows. http://windows.microsoft.com/ja-jp/internet-explorer/products/ie-9/features/cross-site-scripting-filter, (参照日 2015-01-02)

[2] T.Matsuda, D.Koizumi, M.Sonoda: *Cross site scripting attacks detection algorithm based on the appearance position of characters*, MIC-CCA, pp.65-70, 2012.

[3] J.Grossman, et al: *XSS Attacks*, SYNGRESS, 2007.