R-1

コード解析と質問による IPS ルール自動生成

橋浦研究室　　1165129 宇南山 直紀　　1165131 榎本 大貴

1. はじめに

サイバー攻撃から情報システムを守るためには情報セキュリティが必要である．適切なセキュリティ対策にはセキュリティの技術と防護対象となるソフトウェアの理解が求められるため，セキュリティの知識が無い者は導入が困難である．さらに不正侵入防止システム（IPS）は WEB ページに合わせたルールの設定の自動化が確立されておらず，手作業で設定されている現状がある．

2. 研究目的

本研究は，IPS ルールを生成する知識を持たない者でも，IPS ルールを生成できることを目的とする．目的を達成することで IPS ルールを生成する敷居を下げ，WEB アプリケーション独自の IPS ルールの生成を行い，セキュリティの向上を目指す．本研究は IPS Snort のルール生成を対象に行う．

3. IPS ルールを生成する知識

研究目的を達成するにあたり，ツールの補う知識を図 1 に体系化した．IPS 導入のプロセスより手作業での IPS ルール作成には以下の知識が必要になる．IPS の導入対象である Web ページが存在した際に，まずは対象 Web ページの機能を把握する必要がある．図 1 では主に XSS, Option Bleed, SQL インジェクションを IPS の検知対象としているため使用される HTTP メソッドやデータベースの有無が Web ページ機能に挙げられる．被害条件では以上の機能より発生しうる攻撃の判断を行う．ここでは生じうる攻撃の対策および過剰な検知が発生しないという 2 つの効果がある．そこから抽象攻撃で攻撃種別を明らかにしたうえで，具体攻撃にて実際に行われるであろう通信を明確にする．最後に Web ページの仕様を阻害しないように具体攻撃を検知するルールを IPS へ適用する．

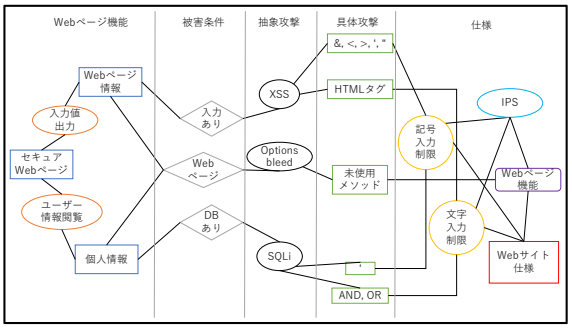

図 1 IPS ルール作成に伴う知識の体系化

4. 提案手法

検知ルールの自動生成については様々なものがある．一例として Sagala[1]はハニーポットのログ情報を利用したルール生成を行っている．しかしながら，このような方法は広範な攻撃に対する検知ルールを生成できるものの，特定の防護対象の仕様にあわせた適切なルールの生成は難しい．

目的を達成するために，IPS のルールを生成するための知識を補う必要がある．本研究では，IPS ルール生成に関する知識やソフトウェアの理解を補うためのツールを作成する．IPS ルール生成に関する知識は，体系化したプロセスをツールが行うことで補う．ソフトウェア理解のためにはソースコード解析を行う．また，ソースコード解析だけでは仕様を把握できない．そこでユーザに対し，質問を行うことでより詳細な仕様も把握する．

5. ツールの実装

ツールは PHP のプロジェクトの選択，PHP のソースコードを解析する機能，どのファイルにルールを適応するか選択できる機能，ツール使用者に対し WEB アプリケーションの仕様に対する質問を行う機能，ソースコードの解析結果と質問を基に IPS ルールの生成を行う機能を実装する．

図 2 ツールの UI

6. 実験

ツールの有効性を確認するために，日本工業大学工学部情報工学科に所属する学生 9 名を被験者とする実験を行った．

実験を行うにあたって RQ は以下の 2 点である．

1. ツールを使用せずに IPS ルールを作成した場合，攻撃の検知は困難なのか
2. ツールによって作成する IPS ルールの検知率，誤検知率に変化が起きるか

実験概要は，被験者を 2 つのグループに分け，1 つ目のグループは初めにツールを使用し，ツールを使用したのち，テキストエディタにルールを書いてもらい，実際にルールを生成してもらう．2 つ

目のグループは初めにテキストエディタにルールを書いてもらい，次にツールを使用してルールを生成してもらう．ツールを使用して作成したルールと使用せずに作成したルールを比較することで，ツールによって生じた差を調べる．

6.1. 作成ルール

実験では Options Bleed，XSS，SQL インジェクションの攻撃を防ぐルールを 1 つずつ作成してもらう

①alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Accessed by DELETE"; content:"delete"; http_method; nocase; sid:1000001;)

Options Bleed につながる使用していないメソッドを検知する．今回は DELETE メソッドを対象として検知する．

②alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"XSS - Typed script tag"; content:"GET"; nocase; http_method; pcre:"/.*?%3Cscript.*?%3E.*?%3C%2Fscript%3E.*/i"; sid:1000002;)

XSS で使用されるスクリプトタグの入力を検知する

③alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"SQL injection - Typed OR "; content:"GET"; nocase; http_method; uricontent:" SQLi.php" ; pcre:"/.*?%27.*?OR.*?%3D.*?/i"; sid:1000003;)

SQL インジェクションで使用されるクエリの OR を用いた入力を検知する．特定ページのみで検知する条件が付いている．

6.2. 実験方法

実験の流れは以下のとおりである．

① Web ページ仕様，ツール使用法，IPS ルール作成方法の資料を与える

② 資料を用いてツールまたはテキストエディタを使用して IPS ルールを作成する

③ IPS ルールを作成出来たら実験者が IPS ルールの内容に応じて修正依頼を出す

④ 修正依頼がある場合は②の作業を行う

IPS ルールに問題が無いまたは 2 回目のルールの修正が完了した時点で実験終了となる

7. 評価方法

実験の評価は WEB アプリケーションセキュリティスキャナ OWASP ZAP を用いて，作成した IPS ルールを適用した WEB サイトに攻撃を行う．そして攻撃を検知した割合を評価基準とする．今回は Options Bleed，XSS，SQL インジェクション，その他(OS コマンドインジェクション，パストラバーサルなど)の攻撃を行い，ルールが対象とする攻撃を検知した割合を検知率，ルールが対象としない攻撃を検知した割合を誤検知率とする．

8. 実験結果と考察

ツールまたはテキストエディタを用いて作成した IPS ルールに対する OWASP ZAP のスキャン結果を表 1 に示す．テキストエディタを用いた実験において作成した IPS ルールの構文にいくつか不備が見られた．本研究で対象としている IPS Snort は構文の不備があると起動することが出来ないため，不備があった箇所にはコメントアウトを行い不備の無いルールが稼働できる状況でスキャンを行った．2 つの実験を比較してツール使用時のほうが検知率の向上と誤検知率のわずかな上昇が生じた．

表 1　OWASP ZAP 使用による検知率，誤検知率

	検知率	誤検知率
ツールなし	0.2670225	0.0001385
ツール使用	0.5995283	0.0939437

RQ1 について考えると IPS ルール作成には 3 つの障壁があることが分かった．1 つ目は時間の壁だ．テキストエディタを用いた実験では実験時間の観点から 40 分の時間制限を用いた．被験者 9 人中 3 人で時間内にすべてのルールを書けないケースが見られた．これは作成ルールの③で生じた．複雑な IPS ルールは作成に時間がかかるようだ．2 つ目は構文の障壁だ．被験者 9 人中 7 人に IPS ルールの構文不備が発生した．原因としてはスペルミスやスペースの不足だった．今回はコメントアウトの処置を行ったが実際では構文不備が生じると Snort が機能しなくなるので特に留意が必要な項目である．3 つ目は通信検知の壁だ．3 つすべての IPS ルールを構文不備なく書けた被験者 2 人中 1 人がうまく攻撃を検知できない IPS ルールを記述した．構文に不備が無くとも IPS ルールオプションの記述順によって通信を検知できないケースが生じる．結果すべての IPS ルールを正しく書けたものは 9 人中 1 人であった．この結果から IPS ルールの作成は困難な作業であることが分かった．

RQ2 について考え，前述の障壁についてツールの効果を示す．時間について，ツール使用時に時間不足によって IPS ルールが書き終わらないケースは生じなかった．作成にかかった時間の平均を

表 2 に示す．ツール使用によって 9 分の短縮が見られた．

表 2　ツール有無による平均所要時間

	平均所要時間
ツールなし	0:39:22
ツール使用	0:30:16

また構文の不備に関してはツール使用により発生しなくなった．最終的な IPS の記述をツールが行うことでスペルミスといった人的なミスの発生を抑えている．最後に通信検知に関して，実験結果に検定をかけることで通信検知に変化が見られたかを確認する．ツール使用の有無による検知率に対して Welch の片側 t 検定を行った．検定表を表 3 に示す．帰無仮説 H_0 を「ツール未使用時とツール使用時での検知率は等しい」対立仮説 H_1 を「ツール未使用時の検知率はツール使用時の検知率より小さい」とし，優位水準 α は 0.05 とした．表 3 より $p<0.05$ を満たすため H_1 を採用し，ツールによる検知率の向上が確認できた．

表 3　検知率に対する
ツールの有無の片側 t 検定表

	ツールなし	ツール使用
平均	0.26702245	0.59952827
分散	0.055719	0.00709923
観測数	9	9
自由度	10	
t	-3.979950394	
P(T<=t) 片側	0.001300587	
t 境界値 片側	1.812461123	

次にツール使用の有無による誤検知率に対し Welch の両側 t 検定を行った．帰無仮説 H_0 を「ツール未使用時とツール使用時での検知率は等しい」とし，優位水準 α は 0.025 とした．表 4 より $p>0.025$ のため帰無仮説 H_0 を採用しツールの有無による差は見られなかった．

表 4　誤検知率に対する
ツールの有無の両側 t 検定表

	ツール使用	ツールなし
平均	0.09394373	0.00013854
分散	0.01241075	1.7275E-07
観測数	9	9
自由度	8	
t	2.526073719	
P(T<=t) 両側	0.035471652	
t 境界値 両側	2.306004135	

9.　まとめと今後の課題

本研究の目的はセキュリティ知識の無い者でも IPS ルールを作成できることであった．実験結果からセキュリティの知識を持たない者が IPS ルール作成が困難である 3 つの障壁が見つかった．そして，ツールはその障壁を軽減または取り除く効果があることを確認した．

今後の課題として，検定では差が無かったがツール使用によってわずかながら上昇した誤検知率について着目する．この誤検知はツールの質問に対する回答を誤ったことで，過剰防御が発生したことにより生じた．実験後に簡易的に行ったアンケートでもツールが使いづらいと感じた被験者が 2 名ほど見られた．ツールの使用性が検知率，誤検知率に直結するツールのため質問内容を明確にする，ツールを初めて触れる人が直感的にわかるような UI にするといった使用性の向上が重要になる．

文　献

[1] Albert Sagala,”Automatic SNORT IDS Rule Generation Based on Honeypot Log,”Proc. of International Conference Information Technology and Electrical Engineering(ICITEE),pp.576-580,2015.