Web アプリケーション・セキュリティの自動検証フレームワーク
—ハッカーの頭脳を超える高精度脆弱性検出—

1. 背景

Web アプリケーションを狙った攻撃が多く発生している。この背景には、セキュリティ検証のプログラムの記述が難しいことと、攻撃に対する動作チェックが十分に行われていないことが原因として挙げられる。これらの多くは開発者によって手作業で行われるため、間違いや見落としが発生しやすい。そのため Web アプリケーションの開発者は、脆弱性検出ツールを使用して Web アプリケーションの出荷前に脆弱性を根絶させようと試みる。しかし既存のツールでは、テストの実行回数が多い上、誤検知を多く発生するため、パワー不足である。

さらに近年では、Web2.0 と呼ばれる技術の登場によって、Web アプリケーションのメカニズムはさらに複雑化し、それに伴ってセキュリティ問題が高度化している。そのため、セキュリティの確保がより一層困難になっているのが現状である。

2. 目的

攻撃から Web アプリケーションを守るためにはセキュリティ確保を徹底しなくてはならない。そのためには、人手に頼らず、また、従来の脆弱性検出ツールより効率的かつ正確に検証を行うことができる脆弱性検出ツールが必要である。そこで本プロジェクトは、従来の脆弱性検出ツールと比較して、より短いテスト時間・回数で多くの脆弱性を検出することができるソフトウェアの開発を目的とする。この目的を達成するためには、従来の手法における脆弱性検出能力を超える必要があり、新規性のある脆弱性検出手法が必要である。そのため、対象とする脆弱性に対して、より多くの脆弱性をより効率的かつ正確に検出することが可能となる新たな手法の提案と、実際に有効性を示すことも目的とする。

3. 開発の内容

近年の巧妙な攻撃に対応した Web アプリケーションの脆弱性を自動検出するフレームワーク Amberate を開発した。Amberate は開発者が Web アプリケーションの出荷前に行う脆弱性検出作業をサポートするツールとして開発した。

手法としては、それぞれの Web アプリケーションに適した攻撃を自動生成し、攻撃テストを実行する。さらに、攻撃後の反応を自動検証することで、脆弱性や設定ミスなどを自動検出する。目指すべき目標は、既存のツールより、短いテスト時間・回数で多くの脆弱性を検出することである。

また、様々な攻撃手法が存在している現状から、より多くの攻撃に対してテストを行うことができるように、脆弱性検出用プラグインを読み込むことによって適宜必要なテストを行うことができるツールとして実装した（図 1）。

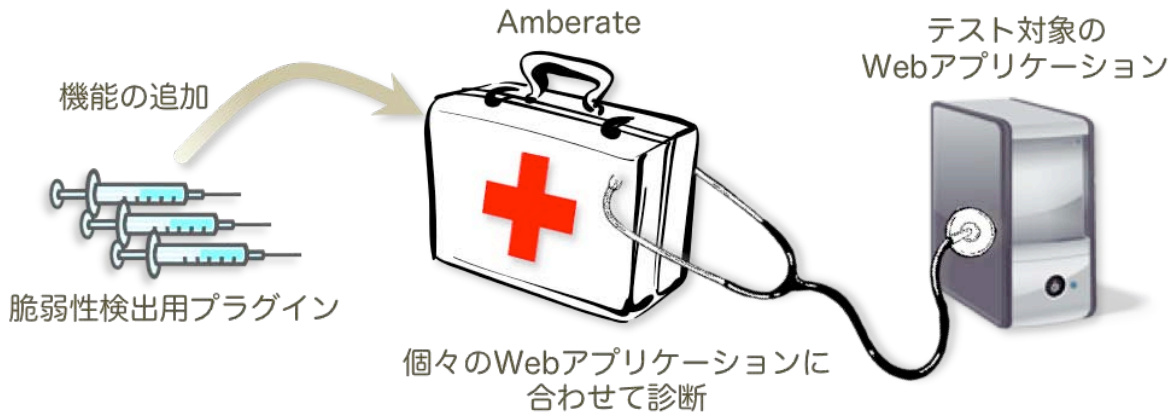

図 1 Amberate の概要図

本開発においては、以下の 3 つの攻撃に対する脆弱性検出用プラグインの開発も行った。

- SQL インジェクション攻撃
- クロスサイト・スクリプティング
- JavaScript Hijacking

各脆弱性検出用プラグインの攻撃生成や脆弱性検出手法は、既存の手法では検出できなかった脆弱性を効率よく検知することを目的とした、新規性のある検知手法を基に開発した。

4. 従来の技術(または機能)との相違

Amberate は既存の脆弱性検出ツールの技術と主に以下の 3 点で異なる。

- 攻撃が可能な箇所を正確に特定
- 各 Web アプリケーションに合わせて強力な攻撃を独自のアルゴリズムで自動生成
- 構文解析や文脈解析を用いた高精度の脆弱性検出

これらの 3 点に着目し、従来の技術との比較を表 1 に示した。Amberate ではこれらの処理を行うときに、より多くの情報を用い、多くの解析を行う。技術的詳細は割愛させていただくが、この作業により余計な攻撃テスト回数を減らし、より正確な脆弱性検出を行うことができる。

表 1 従来の手法との比較

	Amberate	従来の技術
攻撃箇所	任意のパラメータ	すべてのパラメータ
攻撃の生成方法	動的に生成	固定
脆弱性検出方法	構文・文脈解析	文字列マッチング

Amberate の手法の有効性を確認すべく、評価実験を行った。実験では 6 つの Web アプリケーションに対して攻撃テストを行った。これらの Web アプリケーションは、実際にフリーで公開されており、テストではそれらをダウンロードし、ローカル環境でテストを行った。また、これらのアプリケーションが実際に改変され、使用されていることを確認した。ただし、評価実験において未知の脆弱性を検出したため、公的機関に脆弱性の届け出を行っている。公表許可が未だ降りていないため、アプリケーション名の公表は控えさせていただく。

また、実験において Amberate と従来の手法の脆弱性検出能力の比較を行うため、Paros との比較を行った。比較に Paros を用いた理由は、Amberate と同じように未知の脆弱性を検出でき、脆弱性検出能力は高く、評価が高いからである。Paros は Insecure.Org の Web アプリケーション脆弱性スキャナランキングで 2 位であった。Paros は HTTP リクエストと HTTP レスポンスを取得・解析し、攻撃テストを行うツールである。この手法は、Paros 以外の他のツールでも用いている手法である。それに対し Amberate は、より多くの情報（例えば SQL インジェクション攻撃に対する脆弱性検出においては SQL クエリの取得・解析を行う）を用いる、といった技術的差分がある。

既存のツールとの比較実験の一例である、SQL インジェクション攻撃に対するテスト結果を表 2 に示した。 表には、Amberate と Paros のテスト結果を示した。Amberate と Paros における攻撃数とは、実際にテスト対象の Web アプリケーションに攻撃を送った回数である。正検知数とは、検出した脆弱性の数であり、テスト後に実際に攻撃が成功するか手作業で確認を行った。誤検知数とは、手作業での確認作業によって、攻撃が成功しないことを確認した数である。この結果からは、Amberate の方が Paros より、少ない攻撃数で、多くの脆弱性を検出し、誤検知が少ないことがわかった。

表 2 SQL インジェクション攻撃に対するテスト結果

テスト対象の Web アプリケーション	Amberate			Paros		
	攻撃数	正検知数	誤検知数	攻撃数	正検知数	誤検知数
A	214	21	0	362	5	2
B	706	26	0	4802	0	7
C	1080	44	5	5477	0	20
D	276	8	2	1698	0	20
E	498	16	0	1210	0	6
F	290	9	0	1924	0	11
合計	3064	124	7	15473	5	66

各 Web アプリケーションのすべてのページにテストを実行

他の攻撃に対しても同様の結果を得ることができ、Amberate は既存のツールでは検出できなかった脆弱性を効率的かつ正確に検出できることがわかった。

また、Web 技術の高度化によって、攻撃手法が巧妙化している。その結果、多種多様な脆弱性が報告されている。例えば、以下の特徴を持つ脆弱性が存在する。

・永続的データの不完全な安全化による存在する脆弱性
・複数のターゲットの相互作用により顕在化する脆弱性
・ブラウザや Web アプリケーションの環境依存の脆弱性

既存の脆弱性検出ツールでは、これらの脆弱性を検出することができない、あるいは誤検知を引き起こしやすいという特徴がある。一方、Amberate はこれらの脆弱性の検出を可能にする。詳細は割愛するが、前述の評価実験結果と同様に、Amberate の優位性を示すことができた。

5. 期待される効果

従来の脆弱性検出ツールは、テストの実行回数が多い上、誤検知が多かった。それに対し Amberate は効率的かつ正確な脆弱性検出を行うことができる。実際に産業界において Amberate を使用することによって、世の中の Web アプリケーションから脆弱性を減少させることができると考えている。

6. 普及(または活用)の見通し

開発当初は Amberate をフリーかつオープンソースで公開する予定だった。しかし、Amberate の攻撃の自動生成におけるメカニズムが攻撃者にとって極めて有効な情報となってしまう。そのため、予定を変更し、現在は事業化に向けて動いている。ただし、事業化に至るには以下に挙げる 2 点を実装する必要がある。

・より多くの攻撃への対応
・より多くの Web アプリケーションの仕様に対応

これらを数年以内に実装し、事業化を行う予定である。

7. 開発者名(所属)

小菅　祐史 (慶應義塾大学大学院　理工学研究科)

(参考)開発者URL

http://www.sslab.ics.keio.ac.jp/~yuji/