IPA

2008 年度上期未踏 IT 人材発掘・育成事業(未踏ユース)採択案件評価書

1. 担当PM

竹内　郁雄 PM（東京大学大学院　情報理工学系研究科　創造情報学専攻　教授）

2. 採択者氏名

チーフクリエータ：　小菅　祐史（慶應義塾大学大学院理工学研究科　修士課程）
コクリエータ：なし

3. プロジェクト管理組織

株式会社 ゴーガ

4. 委託金支払額

2,693,460 円

5. テーマ名

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

6. 関連Webサイト

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

7. テーマ概要

Web アプリケーションのセキュリティに関する設定やプログラムの多くは、開発者によって手作業で記述され、さらにそれらの記述の正しさの検証も手作業で行われている。そのため、間違いや見落としが発生しやすい。

さらに近年においては、Web2.0 と呼ばれる技術の登場によって、Web アプリケーショ

ンのメカニズムは複雑化し、手作業による安全性の確保はより一層難しくなっている。こうした現状から、セキュリティに関する処理がWebアプリケーションに正しく組み込まれ、実際に安全性が確保されているかを検証する必要がある。

本プロジェクトでは、Web2.0の技術を使用して動的にコンテンツを配信するようなWebアプリケーションに対し、セキュリティの自動検証を行うフレームワーク：Amberateの開発を行う。Amberateはそれぞれの Webアプリケーションに適した攻撃を動的に自動生成し、攻撃テストを実行するためシステムである。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ための環境を提供する。

また、本プロジェクトにおいては、Amberateに組み込み可能な脆弱性検出を行うプラグインの開発も行う。

今回、開発を行うのは、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に対する脆弱性を検出するプラグインである。それぞれの攻撃生成や脆弱性検出の手法は、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開発することを目指す。

8. 採択理由

世の中に、暇で悪い奴が一杯いるというのは「人を見たら善人だと思う」ほうの竹内としてはなんとも納得がいかないし、情けないのだが、それがソフトウェア創造のタネにもなるとは皮肉なものだ。小菅君のセキュリティ脆弱性検出法は、Web アプリケーションが自動生成する SQL クエリや HTTP レスポンスを構文解析や文脈解析することによって「下手な鉄砲、数打ちゃ当たる」検出法より、はるかに効率的だ。

小菅君は卒業研究のときに Sania という、かなり優れものの SQL インジェクション攻撃の脆弱性検出システムを開発したのだが、本提案ではそれを統合フレームワークへと抽象化し、いろいろな脆弱性検出アルゴリズムをプラグインできるようにする。これぞシステムの正常進化である。フレームワーク化と同時に、今回はまだ世の中に検出ツールのない Javascript Hijacking の脆弱性検出プラグインも開発する。まさに未踏であるが、実現性にも大きな期待がもてる。その期待を裏付けるプログラミング能力の高さは、オーディションでも実感できた。

このシステムがオープンソースとして公開されれば、Web2.0 サービスを提供しようとする人にとっては福音となるに違いない。また、ほかの人々が新たなプラグインをつくってくれる。

9. 開発目標

Web アプリケーションにおける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正解に、かつ簡単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。

10. 進捗概要

当初予定していた開発は、予定より早く終えることができた。そのため、Amberate 脆弱性検出能力の向上のための実装や、テストを容易に行うためのユーザ・インタフェースの追加を行った。また、テスト用 Web アプリケーションの作成を行った。

11. 成果

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）。

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。

12. プロジェクト評価

採択理由に「このシステムがオープンソースとして公開されれば、Web2.0 を提供しようとする人にとっては福音となるに違いない」と書いたが、予想外の大ドンデン返しが起こった。小菅君の成果のインパクト、少なくとも社会的インパクトは想像を超えてしまったのである。このシステム Amberate については、軽々にオープンソースなどとは言ってはいけないと思われる。これはプロジェクトの途中で、竹内が気づき、小菅君も認めざるを得なくなった。

Amberate は、小菅君の前作 Sania をアーキテクチャ的に一新し、さまざまな攻撃に対応する脆弱性検出モジュールをプラグインとして追加できるような仕組みにしたものである。これにより、拡張性・将来性が担保できた。従来あった SQL インジェクションに対する脆弱性検出機構をプラグイン化するとともに、大幅に機能を強化したうえ、さらにクロスサイト・スクリプティングと JavaScript Hijacking という新しい攻撃に対する脆弱性検出に対してもそれぞれプラグインを作成し、その道の専門家も舌を巻く素晴らしい検出結果を出した。実際、Amberate が SQL インジェクションの脆弱性を発見した多くのオープンソースの Web アプリケーションは IPA セキュリティセンターに届けられた。現時点では成果にその具体的事例を書けないのがちょっと残念である。このこ

とも含めて、プロジェクト期間中にはIPAの担当部署、IPAセキュリティセンターの方々には大変お世話になった。

各種の攻撃に対する脆弱性の検出をここまでのレベルで行なえるということは、それがそのまま武器にもなる。また攻撃者にはこの強力な脆弱性検出ツールの穴を突くヒントを与えてしまいかねない。もっとも、SQL インジェクションについては、Web アプリケーションからデータベースへの SQL クエリを Amberate から見えるようにしないといけないので、外側からの攻撃は容易ではないと思うが、必殺の攻撃生成ルールが組み込まれているので、ソース公開はやはり危険である。薬が毒にもなるというのはこのことだ。これは登大遊君が 2003 年に開発・公開して，世間に波紋を起こしたソフトイーサを思い出させる。

しかし、これだけのものができたからには何らかの形で世の中の役に立たせないといけない。オープンにすると危険とはいえ、Web アプリケーション開発現場という閉じた場で、脆弱性を検出するという目的には非常に強力で有用なシステムである。閉じた場で開発中の Web アプリケーションがボロボロにされても、原因がわかれば対策することは比較的容易だからである。SQL インジェクションの多くは、入力された文字列のうちの危ない文字をエスケープするというサニタイゼーション（消毒）を行なえば防げるのだが、やっぱりうっかりして抜かしてしまうものらしい。そういう抜けのチェックのために極めて有効である。

攻撃の道具として使われないようにしつつ、かつ閉じた開発現場という閉じた場で使用できるようにするというソフトウェアの使い方がどのようにしたら実現できるかの解は少ないと思う。小菅君は計画書の段階では脆弱性検出ツール開発の発展のために、オープンソースで出すと言っていたが、Amberate をビジネスのタネにして起業することを思い立った。竹内も全面的に賛成である。最終報告会で小菅君がプレゼンの絵で示したように、保健所の人よろしく、ネットワークケーブルと消毒マークのついたカバンを持参して、開発現場で検査をするというビジネスモデルは立派に成立すると思う。これに関連して、オープンソースとは正反対に、悪用を防ぐためにソースの難読化や、モジュールに対する強力な暗号化が必要になる。

小菅君はこの開発成果を携えて博士課程に進学する。博士の研究課題としても奥行きの深い題材である。起業とどう関係づけるのかちょっと予測がつきにくいが、しばらくは小菅君の活躍を見守り、かつ支援したいと思う。実際、，すでにいろいろな方面から声がかかっている。

プロジェクトは予想以上に快調に進んだ。12 月には最後のちょっとした評価を残すだけとなった。正直に言って、こんなに速く進むとは思っていなかった。書いたプログラムの行数は 6 万行にも及ぶ。プログラムの行数で成果を測るつもりは毛頭ないが、やはり半端な大きさではない。

13. 今後の課題

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。