WEB アプリケーションのセキュリティ:
自動スキャンと手動による侵入テストの比較

DANNY ALLAN、戦略的研究アナリスト

ウォッチファイア ホワイトペーパー

目次

Copyright © 2006 Watchfire Corporation. All Rights Reserved. Watchfire、WebXM、Bobby、AppScan、PowerTools、Bobby ロゴ、および Flame ロゴは Watchfire Corporation の商標または登録商標です。その他のすべての製品、企業名、およびロゴは、それぞれの所有者の商標または登録商標です。

Watchfire の書面による明示的合意がある場合を除き、Watchfire は本ホワイトペーパーで公開される情報の適合性および正確性についていかなる表明も行うものではありません。いかなる場合も、読者が特定の目的のために本ホワイトペーパーで公開される情報にアクセスまたは使用したことに起因して、読者または第三者に生じた、直接損害、間接損害、偶発的損害、特別損害、もしくは結果的損害、または利益、収益、データ、もしくは使用の損失による損害に関して、一切責任を負いません。
www.watchfire.com

自動スキャンと手動による侵入テストの比較

序文

90% 以上のWeb アプリケーションが何らかのセキュリティの脆弱性を持ち[1]、75% 以上の攻撃が HTTP / S プロトコル上で行われることから、組織が Web アプリケーションを保護するための強力な対策をとることが非常に重要です。ポート 80 および 443 からの攻撃の割合が非常に大きいように見えますが、これらのポートは組織の玄関口であり、通常オンライン コミュニティ全体にさらされていることを考えれば、それもうなずけます。

Web アプリケーションがますます複雑になるつれ、個人情報、医療情報、財務情報など、取り扱いに注意を要する膨大な量のデータがやり取りされ、保存されています。このような情報のセキュリティに対する消費者の姿勢は、期待のみならず要求ともいえるでしょう。

Web アプリケーションのセキュリティ確保は、手動でのアプリケーション テスト、あるいは自動化システムやツールを使用したアプリケーション テストだけで実現できるものではありません。このセキュリティ確保のための作業は概念構築の段階からすでに始まっており、アプリケーションによってもたらされるセキュリティ上のリスクと実装すべき対策のモデル化がその出発点となります。セキュリティは、すべてのアプリケーションにとって品質指標の 1 つとみなされるべきであり、アプリケーション ライフサイクルのあらゆる段階において分析され、評価される必要があります。Web アプリケーションの脆弱性は、さまざまな方法で発見することができます。

- 自動
  - スキャン ツール
  - 統計解析
- 手動
  - 侵入テスト
  - コードのチェック

このホワイトペーパーの目的は、これらの脆弱性の検知手法を検証することです。特に手動による侵入テストと自動スキャン ツールを対比、比較します。

歴史

手動によるセキュリティ侵入テストは、アプリケーションを保護するための最も古い手法です。開発者は、ソフトウェア開発というものが行われるようになった当初からずっと、開発段階でアプリケーションをテストして欠陥や問題を探してきました。やがて攻撃の頻度が高まり、アプリケーションの複雑さが増すにつれて、このようなセキュリティ問題を発見して利用することだけを目的とする専門家が現れました。このような人たちは「ペン テスタ」として知られています。

自動 Web アプリケーション テストが最初に使用された記録は 1999 年にあります。[2] Web は揺籃期を脱して発展段階に入り、Web ブラウザはやっと、動的アプリケーションの複雑さを徐々に扱えるようになってきていました。これらのツールの目的は、Web アプリケーションを発見して、脆弱性検出のための障害を挿入するというプロセスを自動化することでした。

[1] http://www.imperva.com/company/news/2004-feb-02.html
[2] http://patft.uspto.gov/

© Copyright 2006. Watchfire Corporation. All Rights Reserved.

脆弱性のタイプ

一般的に、多くの Web アプリケーションの脆弱性は、技術的脆弱性と論理的脆弱性の 2 つに分類することができます。技術的脆弱性には、クロスサイト スクリプティング（XSS）、インジェクション フロー、およびバッファ オーバーフローという、よく知られたテストが含まれます。論理的脆弱性は、明示的に分類することがずっと困難です。これらの脆弱性はアプリケーションのロジックを操作して、まったく意図していなかったことを行います。たとえば 2002 年初頭には、ある悪意のある人が、論理的脆弱性を使って Microsoft Hotmail アプリケーションで必要な個人情報検証をバイパスし、[3] 1 つのセキュリティ用の質問の答えを推測してパスワードをリセットすることが可能になったというできごとがありました。

技術的脆弱性

自動化システムやツールはどちらも、技術的脆弱性のテストに関しては系統的で包括的です。Microsoft Hotmail アプリケーションの登録アプリケーションについて考えてみましょう。[4] この 1 つのフォームには約 30 の固有要素があります。非表示のものもあり、表示されているものもあります。このフォームの各要素はクロスサイト スクリプティング、インジェクション フロー、バッファ オーバーフロー、またはエラーの不適切な扱いに対して、潜在的に脆弱です。

技術的脆弱性であるクロスサイト スクリプティングを利用するために、70 種類以上のテクニックが使用できることをご存知でしょうか。[5] これは、ある登録フォーム上のこの 1 つの弱点を徹底的にテストするためだけでも、このフォームに対して 2,000 以上ものテスト（30 要素 x 70 XSS テクニック）をしなければならないということを意味します。このクロスサイト スクリプティング という 1 つの問題だけをみても 80% 以上のアプリケーションが脆弱であるのも不思議ではありません。[6]

Web アプリケーションのクローリング、解析、およびテストを行う自動化システムやツールは、技術的脆弱性に対して手動の侵入テストよりずっと優れています。自動スキャンやテスト ツールは現在、すべての技術的脆弱性に 100% 対処できるわけではありませんが、近い将来そうならないと考える理由はありません。アプリケーションのスキャン ツールが次のような分野で問題を抱えているということが、初期の障害でした。

- クライアント側で生成される URL
- 必要とされる JavaScript 機能
- アプリケーションのログアウト
- 特定のユーザー パスが必要なトランザクション ベースのシステム
- フォーム の自動送信
- 使い捨てパスワード
- ランダムな URL ベースのセッション ID を持つ「無限」Web サイト

自動 Web アプリケーション セキュリティ ツールが成熟するにつれて、これらの障害はすべて解決されました。

[3] http://www.computeruser.com/news/02/02/13/news2.html
[4] https://accountservices.passport.net/reg.srf?roid=2&sl=1&vv=310&lc=1033
[5] http://ha.ckers.org/xss.html
[6] http://www.imperva.com/company/news/2004-feb-02.html

© Copyright 2006. Watchfire Corporation. All Rights Reserved.

自動評価における判定の不確定性（誤検知）および問題を見逃す可能性（検出漏れ）はどちらも、しだいに減少するでしょう。逆に、アプリケーションのサイズと利用範囲が増大するにつれて、技術的脆弱性のテストを手動で実行できる可能性は、困難から不可能へと変わります。多くの企業組織では、自社に存在する何千もの Web アプリケーションを評価するために必要な時間、労力、およびコストを費やすことがまったく不可能になります。また、数千から数百万という技術的脆弱性のテストを人的労力に頼ると、間違いが起きやすくなり、信頼性が著しく損なわれます。調査会社 IDC は次のように結論付けています。「問題は規模とコストです。手動によるレビューを行うことは時間とコストがかかります。本当に優秀な人々がいれば非常に安全ですが、1 日に確認できるコードの量には限りがあります。ソフトウェア スキャナがあれば、より速く、より低価格で作業することができ、はるかに多くの領域をカバーできます。」

論理的脆弱性

論理的脆弱性とは、アプリケーションがどのように動作するかを理解し、そのビジネス フローを回避することによって利用できるものです。自動スキャン ツールと熟練した侵入テスト実施者はどちらも Web アプリケーション全体を検索することができますが、アプリケーションが「どのように」動作するかということやワークフローの背後のロジックを理解できるのは、人間だけです。手動の侵入テスト実施者は、アプリケーションのロジックとフローを理解することによって、ビジネス ロジックを覆してセキュリティの脆弱性をあらわにすることができます。たとえば、アプリケーションがユーザーを、A 地点からセキュリティ検証地点である B 地点を経由して C 地点に導くとします。アプリケーションを手動で検査すると、B 地点でのセキュリティ検証を完全にバイパスして、A 地点から C 地点に直接移動することが可能であるとわかるかもしれません。

統計

100 にのぼる Web サイトを対象とした最近の解析に基づいて、[7] 次の統計が発見されました。

- 36% の Web サイトでは、手動テストによって、自動スキャンより多くの脆弱性は発見されませんでした。
- 17 % の Web サイトでは、自動スキャンではまったく脆弱性が発見されなかったのに手動テストですべての脆弱性が発見されました。
- 46 % の Web サイトでは、手動テスト実施者と自動スキャン ツールそれぞれで発見された脆弱性は互いに補完的でした。

統計の数字といえども誤解をまねく恐れがあり、80 対 20 の法則（成果や結果の 80 % は要素や要因の 20 % がもらすという法則）は必ずしも当てはまりません。80% の脆弱性を検出しても、1 つの重大な脆弱性を見逃してサーバー／アプリケーションが完全にセキュリティ侵害にさらされることがあっては十分ではありません。

結論

Web アプリケーションのセキュリティ脆弱性を発見するために使用される手法にはさまざまなものがあることは、序文に述べました。これらの手法はどれも、単独では完全なものではなく、各手法には固有の長所と短所があります。

[7] http://www.webappsec.org/lists/websecurity/archive/2005-06/msg00014.html

© Copyright 2006. Watchfire Corporation. All Rights Reserved.

自動スキャンと手動による侵入テストの比較

手動による侵入テストと自動ツールはどちらも、Web アプリケーションの重要なセキュリティ脆弱性を発見するために使用できます。自動ツールは手動による侵入テストに替わることを意図したものでは決してなく、手動による侵入テストを完全に自動ツールに置き換えることは決してすべきではありません。しかし自動ツールを正しく使用すれば、Web アプリケーションの広い範囲にわたる技術的セキュリティ脆弱性を発見することができ、その結果を補うために手動の侵入テストによって論理的脆弱性をチェックすれば、時間とコストを節約することができます。

賢明な組織は、自動スキャンと手動による侵入テストの適切な組み合わせを判断して、可能な範囲で最善の Web アプリケーション セキュリティを実現します。

© Copyright 2006. Watchfire Corporation. All Rights Reserved.

ウォッチファイアについて

ウォッチファイアは、オンライン リスクを管理するためのソフトウェアおよびサービスを提供し、Web サイトのセキュリティ確保とコンプライアンスの実現を支援します。AXA Financial、SunTrust、HSBC、Vodafone、Veteran's Affairs、Dell をはじめとする 500 を超える大手企業や政府機関のユーザーが、オンライン ビジネスに影響を与える問題を監視、報告するために、ウォッチファイアの製品を利用しています。ウォッチファイアは、HP/IAPP プライバシー革新賞、*InfoSecurity Product Guide* の 2006 年度注目のセキュリティ企業、*Computerworld* の革新的技術賞、*Computer Reseller News* による「推奨」の評価、2006 年度*SC Magazine* 賞の最終選考をはじめとする、業界でのいくつかの賞を受賞しています。ウォッチファイアは、IDC によって Web アプリケーションの脆弱性評価ソフトウェアにおける世界的市場シェアのリーダーであると認められました。ウォッチファイアのパートナには IBM グローバル サービスや PricewaterhouseCoopers、TRUSTe、Microsoft、Interwoven、EMC Documentum and Mercury などがあります。ウォッチファイアは、米国 Massachusetts 州 Waltham 市に本社を置いています。詳細は、www.watchfire.com をご覧ください。

© Copyright 2006. Watchfire Corporation. All Rights Reserved.