アーキテクチャ・リスク分析

**悪用される前に
設計上の弱点を
見つけて修正します**

システムの設計上の欠陥を特定し、セキュリティ態勢を改善

セキュリティの問題を引き起こすソフトウェアの不具合のうち、半数はシステムの設計上の欠陥です。コードに含まれるセキュリティ・バグを検出するソフトウェア・スキャンや、アプリケーションのペネトレーション・テストだけでは問題の半数を見過ごしてしまい、組織は攻撃に対して脆弱なままとなります。

SDLC の早期に問題を修正

ソフトウェア開発ライフ・サイクル（SDLC）の終盤でセキュリティの不具合が見つかると、その修正には大きなコストがかかります。これを防ぐには、SDLC の早期段階でセキュリティ対策を実施する必要があります。ここで特に重要なのは、コードの実装や QA テストの実施まで待たず、SDLC の早期にセキュリティ上の問題を見つけて修正した方が、コストも時間もかからず、開発フローに与える影響も少ないという点です。

リスクを明確に可視化

アーキテクチャ・リスク分析（ARA）では、ブラック・ダックのエキスパートがソフトウェアに見つかった技術上のリスクをリストアップし、これらを緩和するための手法、ツール、戦略を提言します。また、関連するビジネス・リスクへの理解を助けるとともに、これらのリスクを許容可能なレベルに軽減するための適切な緩和策も提言します。

設計に潜む弱点を特定

ARA では、アプリケーションの設計に対する詳細なレビューも実施し、攻撃を成功させる可能性のある弱点を見つけます。これら設計上の欠陥を見つけるために、システムの主要なソフトウェア・コンポーネント、信頼ゾーン、資産、セキュリティ対策、資産フロー、および脅威エージェントを分析します。ARA により、迂回される可能性のあるセキュリティ対策や、弱いセキュリティ対策、的外れなセキュリティ対策を見つけることができます。

リスクを完全に取り除くか、ビジネスにとって許容可能なレベルまで緩和するシステム・オプションの包括的なリストが最後に提供されます

アーキテクチャ・リスク分析の流れ

ARA は、基本的に 4 つのステップで実施します。

1. ビジネス・コンテキストを分析する

   インタビューを実施して情報を収集・分析し、システムのビジネス目標に影響するセキュリティ・リスクへの理解を深めます。

2. 脅威モデルを作成する

   システムに存在する主要なコンポーネント、資産、脅威エージェント、セキュリティ対策を特定し、これらエンティティとその相互関係を図式化します。

3. リスク分析を実施する

   ソフトウェア・ベースのリスクを特定し、ビジネスへの影響に基づいてこれらに優先順位を付けます（データへの不正アクセスやサービス可用性など）。この分析には、以下のアクティビティが含まれます。

   - **既知の攻撃分析：**既知の攻撃パターン・セットを考慮して、レビュー対象システム内のコンポーネントに関するサブシステムとアプリケーションの挙動をモデル化します。
   - **システム固有の攻撃分析：**十分に確立したセキュリティの原則に照らし合わせて、システム・アーキテクチャの基盤を評価します。また、個別にはほとんど影響がなくても、複数の組み合わせによって深刻な脆弱性を引き起こすような、不特定のソフトウェアの挙動も見つけます。
   - **依存関係分析：**プラットフォーム内にあるソフトウェアの階層を 1 つずつ分析し、各階層によって混入または緩和されるセキュリティ・リスクを理解することに重点を置きます。

4. 緩和策を提言する

   各診断の最後に、担当開発チームとリモート会議を開き、診断中に特定された各脆弱性をレビューするとともに、これら脆弱性に関する開発チームからの質問に答え、緩和・修正の戦略について話し合います。

ブラック・ダックについて

ブラック・ダックは、業界で最も包括的かつ強力で信頼できるアプリケーション・セキュリティ・ソリューション・ポートフォリオを提供します。ブラック・ダックには、世界中の組織がソフトウェアを迅速に保護し、開発環境にセキュリティを効率的に統合し、新しいテクノロジーで安全に革新できるよう支援してきた比類なき実績があります。ソフトウェア・セキュリティのリーダー、専門家、イノベーターとして認められているブラック・ダックは、ソフトウェアの信頼を築くために必要な要素をすべて備えています。詳しくは www.blackduck.com/jp をご覧ください。

ブラック・ダック・ソフトウェア合同会社
www.blackduck.com/jp