車両サイバーセキュリティの未来（4）

車両開発におけるセキュア設計と脆弱性分析

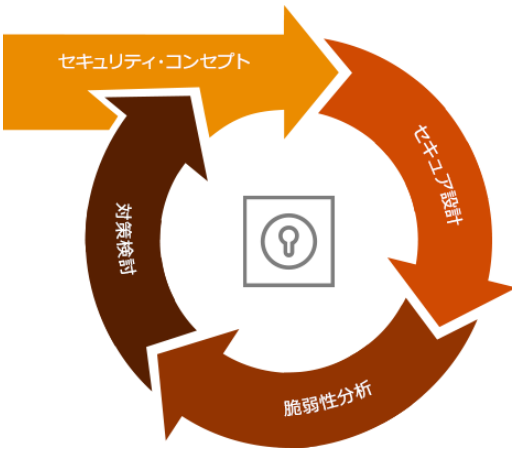
PwC コンサルティング合同会社　マネージャー　奥山 謙
PwC コンサルティング合同会社　シニアアソシエイト　澤 謙太

第3回はコンセプトフェーズにおける活動として、守るべき情報資産・機能資産に対して起こりうるセキュリティ脅威を特定する脅威分析と、特定された脅威に対して共通の評価基準からリスクレベルを算定するリスクアセスメントを紹介しました。第4回は、製品開発フェーズに議論を移して、システムや各コンポーネントの設計段階における活動について考察します。

設計段階でのセキュリティ活動概要

コンセプトフェーズでは、脅威の洗い出しやリスクアセスメントを行うことでサイバーセキュリティゴールが設定され、達成するための方針となるサイバーセキュリティ・コンセプトが定められます。製品開発フェーズでは、まずサイバーセキュリティ・コンセプトに従ってシステムとしての全体的な設計を行います。そして、設計のセキュリティ品質を向上するために脆弱性分析を行い、許容できない脅威の原因となる脆弱性が見つかった場合には対応方法を検討し、設計の改善を行っていきます。（図表1参照）

図表1：設計段階でのセキュリティ活動



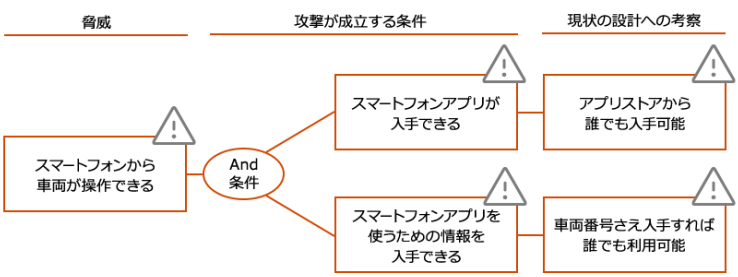
セキュア設計

はじめに、セキュリティ・コンセプトに基づいて、システム全体の設計を行います。設計段階でのセキュリティ品質は、製品開発全体におけるセキュリティ品質のベースとなるためとても重要です。例えば、使用する主要ハードウェアやOSといったシステムのベースとなる要素は、システム設計のような早い段階で決定することが多くあります。ソフトウェア設計のような後の段階でOSのようなシステムの根幹にかかわる部分に脆弱性が見つかり変更が必要となった場合、設計の変更が広範囲に及びます。また、実装やテストの段階のように、さらに後の工程で脆弱性が見つかった場合は、前の工程に戻ってやり直す「手戻り」も多くなります。このように、設計段階で組み込まれてしまった脆弱性は、製品開発全体のコストや期間に大きなインパクトを与えることになります。設計段階でのセキュリティ品質を向上させることは、後の工程で発見される脆弱性を減らし、手戻りが少ない効率的な開発につながります。こうした効率的な開発をするために、設計について脆弱性分析を行い、対策を設計に反映させるという図表1で示したサイクルを回すことでセキュリティ品質を向上させます。

脆弱性分析

脆弱性分析とは、攻撃のために悪用可能な脆弱性が設計上に存在しているかを確認する活動です。アタックツリーなどを用いて、脆弱性が存在しているか分析します（図表2参照）。

図表2：アタックツリーを用いた脅威脆弱性分析



まずコンセプトフェーズで洗い出した脅威に対して、どのような条件で攻撃が成立してしまうかを考えます。そして、現状の設計として攻撃が成立する可能性があるかを考察することで、脆弱性が存在しているかを判断します。スマートフォンやサーバーと通信するようなコネクテッドサービスの分析では、図表2に例示した車両そのものの分析だけではなく、関連するシステムも含めて分析する必要があります。

対応方法の検討と設計への反映

脆弱性分析によって、製品に脆弱性が存在すると判断された場合には、コンセプトフェーズで実施したリスクアセスメントの結果と照らし合わせ、許容できない脆弱性であるかを判断し、対応方法を検討します。対応方法には攻撃が成立しないように設計を変更する方法以外にも、システムを監視し、検知・対応および復旧ができる機能を追加するアプローチも含めることができます。このアプローチは、攻撃が発生したとしてもすぐに危険な状態にはつながらないような緊急度が低い脅威に対して適用することが可能です。このように複数ある対応方法から、リスクや開発コスト、日程などを総合的に判断し、対応方法を決定していきます。

このような脆弱性分析、対応方法の検討、設計への反映というサイクルを、脆弱性が存在しなくなる、もしくはリスクが許容できるレベルに低減できるまで繰り返し行います。その際には、設計の変更によって別の脆弱性が新たに発生するケースについても考慮する必要があります。また、システム全体の設計時だけではなく、ハードウェア設計やソフトウェア設計のように、より具体的な設計を行う際にも、これらの活動を行うことでセキュリティ品質を担保します。

設計段階で発生する脆弱性と実装段階で発生する脆弱性

セキュア設計だけでは全ての脆弱性について対応できないことにも注意する必要があります。図表3では設計段階と実装段階で発生する脆弱性の例を示しています。

図表3：設計段階と実装段階で発生する脆弱性の例

設計段階で発生する脆弱性	実装段階で発生する脆弱性
・認証機能の不備	・バッファオーバーフロー※2
・脆弱性のあるサードパーティー製ソフトの使用	・SQLインジェクション※3
・重要データの耐タンパ性※1が低いストレージへの保管	・ディレクトリトラバーサル※4

図表3で示したバッファオーバーフローやSQLインジェクションのような実装段階のコーディングに起因する脆弱性は設計段階で見つけることはできません。設計段階で発見ができない脆弱性については、実装段階での対策が必要となります。第5回は、実装段階において必要となるセキュリティ活動について考察します。

※1 耐タンパ性：外部から重要データを盗み出そうとする行為に対する耐性度合い

※2 バッファオーバーフロー：プログラムによって確保されたメモリ領域を超えるデータが送り込まれることにより誤動作が生じること

※3 SQLインジェクション：セキュリティ上の不備を利用して攻撃者が任意のSQL文（データベースへの命令文）を実行させてデータベースを不正に操作すること

※4 ディレクトリトラバーサル：ディレクトリをさかのぼることなどにより、本来はアクセスが禁止されているディレクトリやファイルに不正アクセスする手法

お問い合わせ

PwCコンサルティング合同会社
〒100-6921　東京都千代田区丸の内2-6-1 丸の内パークビルディング
Tel : 03-6250-1200（代表）　Mail : jp_cyber_inquiry@pwc.com