サイバーセキュリティを考慮した自動車ソフトウェアの開発

王 思淼・タミール ラハミーム　Argus Cyber Security Ltd.
japan@argus-sec.com

1. はじめに

2025 年には数億台のコネクテッドカーが路上で走ることになると予測されている。それに伴い，自動車の運転が人の判断から機械学習やアルゴリズムに変わろうとしている。車の機能が次々とコンピューター制御化され，今までと比べ物にならないほど運転や車内で過ごすことが快適になっている。しかしイノベーションには代償が伴う。ハッカーがコンピューター化された車載システムに侵入して乗るため攻撃手法を研究する時代がやってきた。車載システムに侵入して攻撃する手法は無限に考えられる。物理的に車への侵入を試みるだけではなく，スマートフォンを使いサーバー経由で車両を攻撃したり，サプライヤーチェーンやアフターマーケット部品を攻撃したりと，可能性は多岐にわたる。世界中の車両がリスクにさらされることもありうる。このように，自動車に対するサイバー脅威は広範囲かつ複雑になってきている。最重要事項である安全とプライバシーの確保のために苦闘する時代が始まりつつある。

この記事では，サイバーセキュリティを考慮した自動車ソフトウェアの開発に焦点を当て，自動車業界のソフトウェア開発について検討する。そしてアルガスサイバーセキュリティが提供する侵入テストの知見に基づいて，ハッカーに利用されやすい既知の脆弱性を例示し，その脆弱性がどの様に悪用されうるかを説明していく。最後には，これらのリスクを低減するためにサイバーセキュリティを考慮することの重要性について説明をする。

2. 自動車業界におけるソフトウェア開発 業界固有の事情とセキュリティ懸念

車載システムソフトウェア開発は，他業界のあらゆるシステム開発と共通する部分が多くある。車載システム開発においても，ソフトウェア開発ライフサイクル（SDLC）は計画から分析，設計，開発と実装，テストと保守といったサイクルとなる。しかしながら，自動車業界には固有の標準や複雑なサプライチェーンがあるため，技術者は業界固有の事情を考慮して開発を行う必要がある。

2.1 技術的観点では

安全性が重視される多くの ECU（エレクトリックコントロールユニット）は Classic AUTOSAR を使用し，CAN バス，FlexRay または MOST プロトコルといった自動車用途に最適化した通信を使用するケースが多くある（外部につながるコネクテッド ECU には，Linux，QNX または Android システムを使用するケースも多くあるがここでは対象外とする）。Linux や他のオープンソースのオペレーティングシステムと比較して，Classic AUTOSAR は利用台数が少ないため，多くの脆弱性が未発見のまま残されていることが多く，ハッカーは比較的容易に脆弱性を発見することができる。

ネットワークに目を向けると，自動車用通信プロトコルにも様々なセキュリティリスクがある。例えば CAN バスでは，メッセージが認証されることなくある ECU から他の ECU にコマンドが送信されるため，ハッカーが正規の ECU になりすまして他の ECU を制御することが可能となってしまう。

2.2　ビジネス観点では

OEM は多くのコネクテッドプラットフォームを車両に組み込もうとしている。Bluetooth や NFC，Wi-Fi を通じてスマートフォンと接続したり，専用プロトコルでサーバーや他の車ともつながるようになっている。その様な無線通信システムは，車とユーザーを今までに無かった新たな脅威にさらすことになる。「つながる」車は，大きなリスクを抱えることになるのである。例えば，テレマティックスシステムにアクセスすることが出来ると，テレマティックスセンターを介して同じサービスにつながる他の車が攻撃されるリスクある。更には Wi-Fi や Bluetooth などの標準ワイヤレス規格に接続にすることで，スマートフォンを利用した攻撃を受けるリスクもある。何かの新しいテクノロジーを採用する際には，ハッカーの悪用を防ぐための十分な検討と対策が必要となる。

2.3　影響と複雑性

自動車の場合はソフトウェア開発プロセスで見落とされた脆弱性が原因で，大事故につながったりドライバーの命を危険にさらしてしまう可能性がある。他の業界と比べリスクの影響範囲が大きいためミスが許されない。また車のソフトウェアを更新することは簡単ではないため，数年先に要求されるであろうセキュリティ基準を，開発段階で予め実装しておく必要がある。さらにサイバー業界の技術進歩のペースが非常に速いため，サイバー攻撃への調査・対策が比較的手薄な自動車業界にハッカーが目を付け，脆弱性を悪用するようになるのは時間の問題となってきている。

視点	ビジネス	テクノロジー	セキュリティ
背景	コネクテッドカーの急速な普及	車に特化した OS とネットワークプロトコル	製品が長期間利用されるが，ソフト更新頻度は低い
セキュリティ懸念	コネクテッドシステムのセキュリティ問題	セキュリティ調査経験が少ない規格故，攻撃が比較的容易	ハッカーが脆弱性を見つけるのに十分な時間がある

イノベーションには代償が伴う：ハッカーは車載システムに侵入するために新たなテクノロジーを追求

3.　共通脆弱性

車載ソフトウェアには脆弱性が残っている可能性があり，その脆弱性は設計上の脆弱性と実装上の脆弱性の 2 つのタイプに分類出来る。またソフトウェアの設計段階から検証テスト段階において検出された共通脆弱性は，きちんと対策を行っておくことは非常に重要である。

- 設計上の脆弱性はシステムロジックの欠陥のことを指す。通常動作ではシステムは意図した通りに動作するものの，想定していない極端なケースがあり処理ミスが起きると攻撃リスクにさらされてしまう。
- 実装上の脆弱性はシステムロジックの不適切な実装のことを指す。異常なデータ入力があった場合，データの表現や解釈の間違いによりプログラムが意図しない動作をするとリスクとなる。

アルガスサイバーセキュリティのリサーチチームは，車両や ECU を対象にした脅威分析とリスク評価，ECU や車載ネットワークの侵入テストなどのサービスを提供している。過去に行ったプロジェクトの経験から，ハッカーが利用する可能性が高い CWE（Common Weakness Enumeration）を例として挙げ，上述の 2 タイプの脆弱性を解説する。CVE とは異なり CWE とは，特定のアプリケーションに紐づかない脆弱性とことを指す。

- 設計時と実装上の CWE \\ 制限されたディレクトリへのパス名の不適切な制限（パス・トラバーサル）（“CWE-22”）－ 制限されたディレクトリへのアクセスをする際に，外部からの入力により相対パス名を指定することがある。ソフトウェアがパス名の特定要素を適切に無効化していない場合，入力されたパス名が制限されたディレクトリ外のリソースを指してしまう場合がある。この場合，攻撃者はシステム上の任意のファイルを読み取ることが出来るようになり，情報漏洩や，ファイルシステムに保存されている様々な機密情報が盗まれる。
- 実装上の CWE \\ 情報漏洩（“CWE-200”）は，情報へのアクセスが明示的に許可されていないユーザーへ情報を漏洩することである。自動車のセキュリティにおいて重要なものの一つはプライバシー情報である。ド

ライバー個人情報の漏えい，OEM 機密情報の漏えい，複数ユーザーのプライベート情報が入ったカーシェアのデータベースの盗用などが考えられる。

この CWE を実証する例としては，Linux デバイスの Bluetooth 脆弱性である CVE-2017-1000250 がある。BlueBorne と呼ばれるこの CVE は Bluetooth 接続時における攻撃ベクトルの一つである。この脆弱性を悪用すると，ほぼ全てのインフォテインメントシステムが危険にさらされる可能性がある。

- 実装上の CWE \\ 整数オーバーフローまたはラップアラウンド（“CWE-190”）– 計算ミスにより格納可能な値よりも大きくなる場合，整数オーバーフローまたはラップアラウンドが発生する可能性がある。整数オーバーフローは 3.2.1 より前の libarchive で発生（CVE-2016-6250）して，遠隔からの DoS（アプリケーションクラッシュ）やコード実行を許してしまう可能性がある。自動車のハッキングではこの脆弱性が利用され易い傾向にあるとアルガスのリサーチチームは考える。
- 設計時の CWE \\ 不適切な認証（“CWE-287”）ユーザーが特定の ID を持っていると主張した場合に，ソフトウェアにおいてその主張が正しいことを証明しない，或いは証明するが理由が不十分であるという問題。この CWE の例としては，アクセス管理が瓦解しデータ流出につながる共通脆弱性 CVE-2018-13908 がある。この脆弱性は車載システムの侵入テストで頻繁に利用されている。

これらが悪用された場合，損害リスクとして考えられるのは，データの流出やプライバシーの漏洩，OEM の評判失墜，金銭的損失，人命へのリスクなどがある。

コネクテッドカーが増えるに伴い，リスクも増大

4. 被害軽減のため実践すべきこと

ソフトウェア開発においてセキュリティのリスクを軽減するために有効なことは，そのソフトウェアよりも下のレイヤーの自動車アーキテクチャにまで目を向けることである。自動車業界は他の業界と比較して非常に複雑であるため，できるだけ多くの保護レイヤーを準備しておく必要がある。例えば，車載システムのコアとなる ECU や外部に接続する ECU のみならず，バックエンドサーバーや車へのコネクティビティサービス，製造ライン，サプライチェーン，アフターマーケット部品などをハードニングすることが必要となる。目指すところはハッカーのモチベーションを下げさせることであり，ハッカーは常に効率の良い経路を選ぶので，経済的•技術的•様々な観点から自動車業界は費用対効果が低いとハッカーに思わせることが出来れば良い。そのためには，車を保護するためのプロセスは以下の様にすべきと考える。

4.1 セキュリティを考慮した設計

基本的に他からの影響を受けないようにソフトウェア設計をすべきである。例としては以下のものがある。

- アーキテクチャのセグメント化：安全性が重視される ECU と，コネクティビティにより外部とつながる ECU の間を分離するために，ゲートウェイとドメインコントローラを設置
- 脅威分析とリスクアセスメント（TARA）：車両の設計段階で車載システムやアーキテクチャ，部品に内在する潜在的なセキュリティの問題と脆弱性を見出す。各々の脅威に対するリスクは，悪用のされやすさと，悪用された時の影響の大きさの二つの観点で精査し，対策の提案とセンサー設置場所の優先度を検討。
- 自動車業界の標準と法規に準拠した設計

4.2 サイバー的に安全な実装と防御の考え方

ネットワーク，コンポーネント，もしくは OEM サーバーで利用されるセキュリティプロトコルを標準化することで，セキュリティが全般的に向上するだけでなく，

それぞれが攻撃されにくくなる。例としては以下のものがある。

・セキュアコーディングについてのトレーニング：技術者のトレーニングは，開発段階ですぐに効果が期待でき，あらゆる実装に有効である
・コードレビュー：アプリケーションソフトウェアやデバイスのファームウェア，通信プロトコルに潜在するセキュリティの問題を特定し，問題解決のための対策を実装
・ファイアウォールの実装：パケットの内容まで精査して判断する DPI ファイアウォールの実装
・メモリ保護：制御フロー・インテグリティ（CFI）を使って，起動時だけでなく起動後もランタイムでソフトウェア整合性を検証
・ホスト型侵入検知および侵入防御システムの実装：システム内部で検出した攻撃性の異常を検知し，そのログを OEM へ送信する。ログにより，OEM のセキュリティアナリストが異常を調査し対策することが可能となる
・侵入テスト：侵入テストを実施することで，インターフェースやコネクティビティを通じて外部につながる ECU の脆弱性を識別
・ソフトウェア暗号化：暗号化アルゴリズムの実装は，悪意ある操作を困難にするために不可欠

4.3　ライフサイクル管理

車が工場から出荷されてから廃棄されるまでのライフサイクル全体でサイバーセキュリティを考え，対策が行われるべきである。例としては以下のものがある。

・検出した脅威に対する既販車への対応：車両および車両のサイバーセキュリティの対応の知見を持った，自動車用のセキュリティオペレーションセンターによるサポートが必要。
・既販車のログを監視：脅威検出エンジンを利用し，コネクテッドサービスへの脅威を検出
・脆弱性管理：調査ツールなどを利用して脅威を分析し，脅威への対応の意思決定を迅速化
・定期的なソフトウェア更新およびセキュリティポリシーの更新：定期的にソフトウェアを更新することで，脆弱性を見つけて悪用する時間をハッカーに与えない

ソフトウェアにおいては，より下位のプラットフォームに強固なセキュリティ層を実装すると，ソフトウェアが悪用される脆弱性やリスクを大幅に低下させることが出来る。もし脆弱性があったとしても，発見されることが困難になり，悪用されにくくなる。

5.　最後に

馬とロバのどちらが先に山の反対側に着くかというういう実験がある。馬は全力で山を登る一方，ロバは山の横を周り楽々と先に反対側に到着する。ハッカーはロバの様に，常に楽なリスクの少ない道を選ぶ人種である。

サイバーセキュリティの対策として重要なことは，出来るだけ多くの障害物を設置し，この車は他の車よりも「少し手がかかる」とハッカーに思わせて攻撃対象から外させることである。車を守るためには，1 つの ECU から世界中の車両までの幅広い分野のサイバーセキュリティを考慮しなければならない。サイバーセキュリティを考慮したソフトウェア開発は，車を守るための第一歩であることを忘れないで頂きたい。

アルガスではソフトウェアの脆弱性を発見・悪用させにくくする各種 OS に対応した ECU プロテクションソフトウェアを提供している。サイバーセキュリティを考慮したソフトウェア開発プロセスを導入すると同時に，このようなツールを利用して将来の新たな脆弱性をハッカーに発見・悪用させにくくすることが，製品サイクルの長い自動車ソフトウェアの防御には有効である。