Vol.2022-ARC-248 No.59
Vol.2022-SLDM-198 No.59
Vol.2022-EMB-59 No.59
2022/3/11

大規模組み込みシステムの分散開発における脅威および脆弱性に関する情報共有フレームワークに関する検討

左近　透[1,2]　中本 幸一[1]

概要：近年，高度な医療デバイス，コネクティッドカーや自動運転車両やロボット，製造ラインなどネットワーク化，高機能化，複雑化した組み込みシステムに対するサイバー攻撃が懸念となっている．通常，これら大規模組み込みシステムの開発は，複数の製造メーカでシステムの階層に分けられ分散して開発分担される．特に，担保すべきシステムの安全性は，製造メーカ間で安全機能に関する情報共有に基づき整合的に進められる．一方，サイバーセキュリティに関しては，開発時に製造メーカ間で共通に利用できる情報フレームワークは，従来の情報システムから派生したものが中心となっている．しかし，各製造メーカで保有する情報の相違や，知的財産などの隠蔽の必要性が存在すること，別々のメーカが作成した機能部品を経由したサイバー攻撃事例が想定されることなどから，階層的な分散開発分担で必要な脅威や脆弱性に対する記述能力が十分でないと考える．本研究では，大規模組み込み開発における階層的な分散開発において脅威，脆弱性，攻撃情報を共有し，分析し，設計や実装，対応要否の検討をおこなう情報の構造化を検討する．そのために，既存のセキュリティ関連情報の構造を分析する一方，セキュリティ関係のカンファレンスにおいて発表された事例を分析することにより，サイバー攻撃に関連する各種情報の構造化の検討を行った．

キーワード：CPS，分散開発，サイバーセキュリティ，脅威，脆弱性，攻撃事例分析，情報構造化

1. はじめに

今日，自動車や医療機器，産業用ロボットなどに代表される大規模，かつ高度に複雑化した Cyber Physical System（以下，CPS）は日常の生活，および生産活動において一般的な存在である．CPS はその性質上，故障や処理の異常に起因するシステムの異常な振る舞いが物理的な損害につながる可能性が高い．そのため，これら障害に起因する異常な振る舞いに関して適切な対応策をシステムに組み込む必要がある．

近年、安全に対する機能に付け加えて，近年，これら CPS に対するサイバーセキュリティのリスクも指摘されており，セキュリティ関連カンファレンスでの発表が行われ、実社会での攻撃事例も発生している．CPS に対するサイバー攻撃で攻撃者が利用するアタックサーフェスは，1)外部との通信，2)システム拡張や接続のための外部バスや拡張インタフェース，3)検査や保守，解析用インタフェース，4) 動作記録やソフトウエアそのものの保持，更新に利用される記憶装置，および 5) 電磁波や電力ノイズの注入や漏洩電波など物理インタフェースなど，多岐にわたる．CPS 開発時には，これらのアタックサーフェスからのサイバー攻撃を想定して，機能安全と両立するサイバーセキュリティの要件定義，設計，実装および検証を行わねばならない．

このような開発をプロセスレベルでサポートするため，たとえば，車のサイバーセキュリティ開発プロセスの標準規格である ISO/SAE 21434[1]では，機能安全側の規格である ISO 26262[2]と対になる，いわゆる V 字開発モデルで構成されている．

しかし，サイバーセキュリティの開発プロセスのアクティビティと安全機能開発のそれには相違点も存在する．特に，機能要件導出において，アタックサーフェスから実行されるサイバー攻撃の脅威とそのダメージのリスクを評価し，対策方針および対策機能を検討するサイバーセキュリティリスクアセスメント，および機能開発の各段階において行われるセキュリティ上の弱点，ひいては脆弱性を評価検証する脆弱性分析作業はサイバーセキュリティ開発固有のアクティビティである．開発プロセス上、サイバーセキュリティリスクアセスメントは，システムの機能要件導出に関連するアクティビティに，一方，脆弱性分析作業は，導出された機能要件の設計・実現に関連するアクティビティに含まれている．

車や生産システムなどの複雑かつさまざまな機能部品の組み合わせで構築される CPS は，複数の企業や組織が関連し，相互に設計情報などを交換しながら開発する分散開発手法が主流である．たとえば車両開発においては車両メーカーが，車両の機能要件を定義し，それらを車というシステムを構成するサブシステムに割り当て，サブシステムを実現する Supplier で当該サブシステムの開発を行う．さらに Supplier がさらに要件を分割詳細化し，それを下流の Supplier で開発を割り当てる場合もある．このような分散開発を想定する ISO/SAE 21434 などでは，リスクアセスメントは，サブシステム発注者である Consumer が，脆弱性分析は Supplier が担当する場合が想定されている．

Supplier が発見する脆弱性は，機能部品，すなわちサブシステムレベルでの脆弱性である．しかし，これがシステムレベルの脆弱性となるかは自明ではない．アタックサーフェスから当該脆弱性を直接利用可能な場合には，発見された脆弱性と，リスクを評価する Consumer との情報共有は単に当該脆弱性情報の共有で対応できると考えられる．

しかし，Black Hat[3]などのセキュリティカンファレンスで発表される攻撃，特に車両に対する攻撃は，複数の脆弱

1 兵庫県立大学情報科学研究科
School of Information Science, University of Hyogo
2 (株)オートネットワーク技術研究所
Autonetwork Technologies, ltd.

Vol.2022-ARC-248 No.59
Vol.2022-SLDM-198 No.59
Vol.2022-EMB-59 No.59
2022/3/11

性，製品仕様の前提条件，制約条件などを利用し，車両の制御や機密情報の漏洩を実現している．このような，複数の攻撃やシステムに関する情報分析を連携させておこなわれる複雑な攻撃を設計・開発側で発見する，もしくは発見されたあと，攻撃に関する知識として整理し，進行中もしくは将来の設計開発において利用するには，1) 連携する攻撃と攻撃に利用された弱点や制約の関係を取り扱うことこと，ならびに，2) それを複数の組織間で相互に理解可能な形で表現できることが必要と考える．

また，脆弱性は，設計や実装，さらには実装の対象となるターゲットシステムの制約により発生する場合も想定される．これらの情報はおもに Supplier グループが保有し，Consumer 側は必ずしも保有していない．また，企業の知財権などにより相互に共有できないものもある．そのため，3) 攻撃対象に関する情報は偏在している，という制約において，1)、2)の要件をみたすことが要求される．

本研究では，高度に複雑化した CPS の分散開発において脅威分析，脆弱性，攻撃情報を共有し，分析，再利用するための情報の構造化を検討する．本稿の構成は以下の通りである．まず，攻撃事例を分析検討から，CPS へのサイバー攻撃の構造化記述にたいする要件を検討する．次に，既存のサイバー攻撃の分析記法である共通脆弱性データベースや，代表的な脅威分析手法による記述による対応可能性を検討する．その上で，脅威，攻撃および脆弱性の構造化表記形式を提示する．当該表記形式を公開されているサイバー攻撃事例に適用した例を示すとともに，分散開発への適用について行った検討を示す．

2. CPS サイバーセキュリティ脅威事例分析

複雑な CPS に対するサイバーセキュリティ脅威が実現した事例として，そこで，Black Hat で既に発表済みの車両のハッキングに関する事例を検討する．また，同じく複雑度の高まりつつある System On Chip (SOC)への脅威として，サイドチャネル攻撃と Spector/Meltdown 攻撃を取り上げる．

2.1 事例 1

2015 年の Black Hat U.S.A.において，C.Miller と C.Valasec は，特定車種の車両を遠隔から改竄し，一部の車両機能について遠隔で操作可能な事例を示した[4]．この発表で彼らは車両に搭載された機器を携帯電話網または Wi-Fi ネットワークを経由した車外からの攻撃による機器の改ざんに成功した．さらに，彼らは改ざん機器を経由して車両内部のネットワークに接続された制御用コンピュータ群に不正なコマンドを含む通信を送り込むことにも成功した．彼らの用いた攻撃手順は，1)解放されていたポートの利用，ポートを開けていた機能にアクセスコントロールがなく，利用可能となっていた I/O Processor へのコマンド送信，ファイルシステムへのアクセス，システムのモード変更およびソフトウエアアップデートコマンド操作により，I/O Processor の改竄および，それを経由しての制御コマンドの送信を行なっている．それらの作業に必要な情報は，USB によるソフトウエアアップデートの解析や，入手したコードにリバースエンジニアリングに基づいた分析に基づいている．

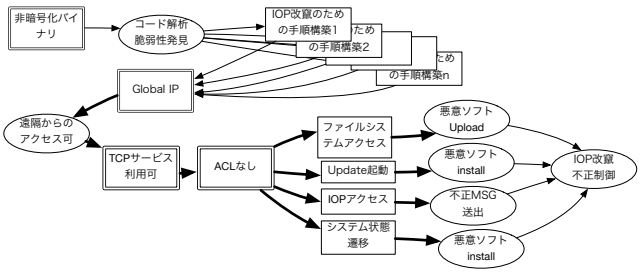

図 1　各種情報を統合した攻撃

図 1 の攻撃は，複数の情報源からの情報入手とさまざまな脆弱性を利用した分析によりシステムの弱点を結合して成立させたものである．

2.2 事例分析 2

2017 年，S.Nei らは Black Hat U.S.A.において外部からの車両 Hacking が可能であることを示した[5]．彼らは，車両に搭載されている HMI 装置から車載 LAN 内部に不正通信を行うことに成功した．彼らは，HMI 装置のブラウザの脆弱性を利用した．まず、JSCell.sort メソッドに対し，不正な SORT 比較関数を適用と既知脆弱性 CVE2011-3928 を用い必要なアドレス情報を取得した．次に、同じメソッドに別の不正 SORT 比較関数を適用し，アドレスに割り当てられたメモリ領域を解放し，解放直後のメモリ領域に不正な JavaScript を含む情報を挿入した．さらに，同様の手法で、JavaScript の処理系で使われるアドレスを取得、そのメモリ領域に Shell コードを書き込み，悪意ある JavaScript が実行された時に起動した．起動後、HMI 装置のカーネルの既知脆弱性 CVE 2013-6283 を利用してカーネルコードを改ざんによりルート権限を奪取した．車載ネットワークは、Gateway で保護されていたが、Gateway 管理用のバックドアを簡易 Proxys 操作 Linux コマンド (socat)で開けた．

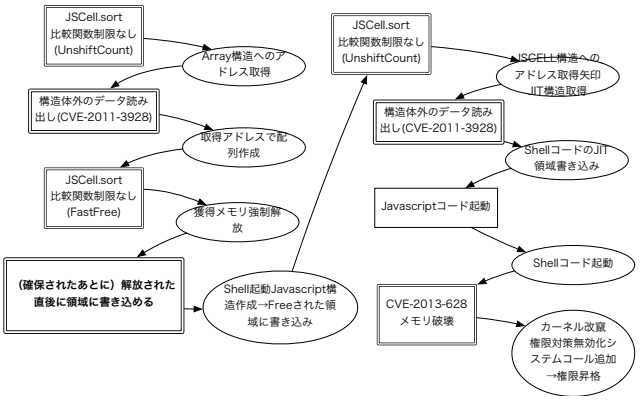

図 2　連鎖的な攻撃

図 2 は，リスト操作中に対象リストを変更する機能を指定することができるという弱点の使い方を変えながら利用し，攻撃者の目的達成のための機能を順に利用することで，初期の目的を達成している．

Vol.2022-ARC-248 No.59
Vol.2022-SLDM-198 No.59
Vol.2022-EMB-59 No.59
2022/3/11

2.3 事例分析 3 物理攻撃サイドチャネル攻撃

ハードウエア攻撃に関しては，特に非浸潤的に内部情報の取得に用いられる手段としてサイドチャネル攻撃がある[6]．これは，ハードウエア内部の処理機能が，処理の内容により電力消費や漏洩電波，処理完了のタイミングなどと関連している場合，それらの情報を収集し統計的手法などを用いて関連を分析して処理に関連する秘密情報を得るものである．サイドチャネル攻撃にはさまざまな手法が存在するが，共通する構造は図で表される構造になっている．

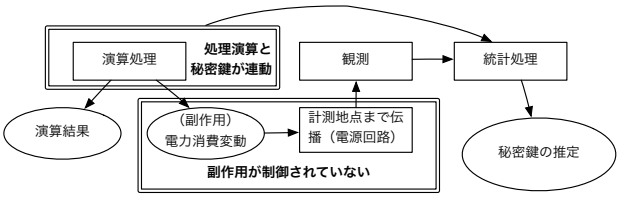

図 3 サイドチャネル攻撃の構成.

図 3 では，演算処理において，(電力消費型のサイドチャネル攻撃の場合)，電力消費の差分が外部から測定可能な領域まで伝播，並びに，処理ロジックのレベルで演算処理の詳細と電力消費に相関があるという弱点を利用され，十分な精度で消費量を取り出された場合，統計的処理などを施した上で処理詳細が分離されることを示している．

2.4 事例分析 4 Spector/Meltdown

サイドチャネルと別のハードウエアの弱点を突いた攻撃として Spector/Meltdown[7]などの事例がある．これらはプロセッサの高速処理のための機能である投機実行でのアクセス制御の弱点，投機実行実施確率操作の不正利用の弱点，キャッシュデータは実行コンテキストが変わっても残存する弱点を組み合わせて攻撃が実現されている．

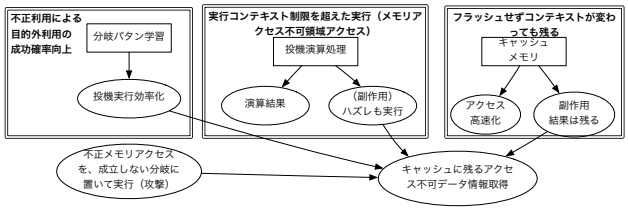

図 4 プロセッサの弱点攻撃の構成

図 4 では，プロセッサの持つ処理効率化機能の副作用，および副作用を有効に使うための不正な機能が組み合わされて，攻撃者の目的が達成されたことをしめしている．

2.5 脅威および脆弱性表記の必要要件

これらの特徴から，構造的な記述に要求される性質について，以下のことが判明する．

(1) 脆弱性の記述

先の４つの事例であるとおり，サイバーセキュリティ攻撃は複合攻撃であると同時に，利用される弱点やそれに起因する脆弱性もバグなどの欠陥に限定されていない．攻撃者が利用する脆弱性は３種類に大別できる．すなわち，バグを含む機能上・設定上の弱点による脆弱性，機能の作用の悪用を管理していない弱点による脆弱性，機能の作用の副作用を管理していない弱点による脆弱性である（図 5）．

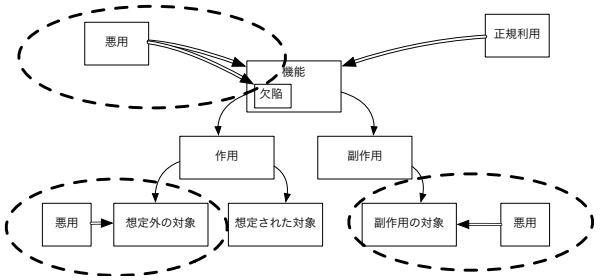

図 5 悪意ある攻撃と弱点の基本パターン

つまり，先の例で利用された弱点を表記するためには，存在する機能上の欠陥，想定外の利用，および副作用の三つであり，攻撃とは，それらに対する攻撃（利用）という形式が必要である．さらに，それらの弱点を利用した結果が関連づけられなければならない．

(2) 連鎖構造の記述

弱点を利用するためには，Attack Surface から弱点に到達し，なおかつ弱点を利用できる攻撃が可能でなければならない．事例(1)に示したように，攻撃は制御フローの上流の弱点を利用して、正常機能を介して行われる場合もある．また，機能そのものの管理や設定が不適切なためにその機能が利用される場合もある．そのため，単に攻撃と脆弱性を直接関連づけるだけでなく、途中に存在する機能や設定などを関連づけた経路を表記可能な形式が必要となる．

このような連鎖構造を記述するためには，(a)利用する弱点, (b)弱点を利用する方法, (c)弱点を利用するための条件, (d)弱点を利用した結果および(e)それにより可能となる現象の 5 つが必要となる．正常機能では、(a')機能の利用、(c')条件、(d')結果となる。これらはさらに攻撃の対象となる資源や機能と関連付けらなければならない．

(3) 分散開発に対応した階層情報構造の取り扱い

２章の例では，攻撃者は、システムを構成する各要素の情報や脆弱性を連携し、攻撃を関連させて実行している．一方，分散開発においては，モジュールやコンポーネントを担当組織で開発し，完成した成果物をさらに別組織で統合してサブシステムを構成，さらに別組織にサブシステムを統合して最終的なシステムを構築する．たとえば，車載システム開発では，プロセッサやソフトウエアコンポーネントを開発するベンダーや Tier-2 と呼ばれるサプライヤがおり，それらの開発物を統合してサプシステムを開発する Tier-1 サプライヤがあり，最終的に車メーカーでサブシステムを統合して車両を開発する．このような分散開発での階層構造では，上流と下流の階層間での情報の粒度が異なり、互いに持つ情報をそのまま開示した場合では、前提知識の相違から共通理解が困難であることが想定される。また、同じ階層での別組織間でも知的財産、著作権、企業機密や設計要求の根拠などを含む場合は，関連情報が情報として開示されない場合も想定される．

分散開発で，上流から下流への共有情報は、要件に関する

Vol.2022-ARC-248 No.59
Vol.2022-SLDM-198 No.59
Vol.2022-EMB-59 No.59
2022/3/11

ものであり、下流から上流への共有情報は成果物内部の脆弱性にかかるものである．前者では，上流の脅威分析において，その脅威が実際に実現しうるかに関する詳細な情報は必ずしも上流側が保有していない．実際に，脅威が現実化するために脆弱性が存在し，それらが利用可能性の判定には，詳細構造に含まれる情報を必要とする．つまり，脅威に関連して仮定した脆弱性や攻撃経路についてそれらを評価できる下流工程に対し情報を引き渡し，下流工程では保有する詳細情報に基づいて詳細化する必要がある(図 6).

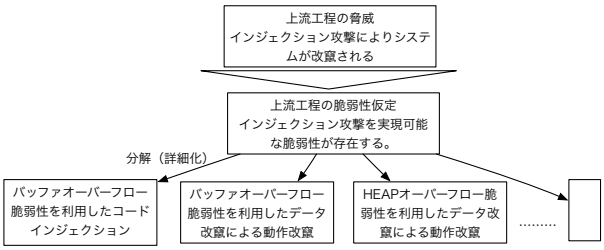

図 6　上流工程の想定脆弱性と下流工程における分解例

また逆に，下流工程で発見された、もしくは利用したコンポーネントに含まれる脆弱性や弱点について，それがより上位設計の粒度で表記されなければ，上流設計の設計粒度で対応が必要な場合に対応が難しくなる．そのため，個別の弱点，脆弱性をまとめて，上流の粒度に攻撃およびその条件と結果を提示することが必要となる．

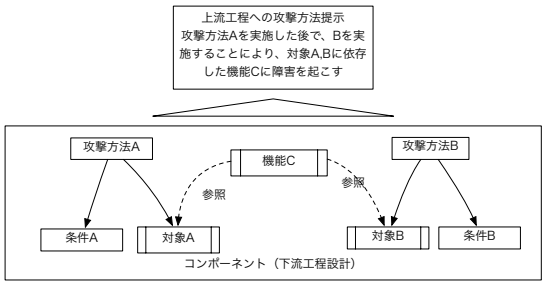

図 7　下流工程での脆弱性と上流工程への提示の例

図 7 では，下流工程は，単に，対象 A と対象 B の攻撃を示すのみならず，上流工程からは認知できない複合攻撃による機能 C の損失の情報について示す場合を示している．

3. 既存の表記方法

米国の MITER や 情報処理推進機構(IPA)などにより，情報システムの弱点やサイバー攻撃，および脆弱性は，それぞれの形式の情報をまとめられ一般に利用可能な形で公開されている．脆弱性に関する報告を取りまとめたデータベースとして代表的なものは Common Vulnerability Enumeration (CVE)[8]が，また Cyber-Attack に関しては Common Attack Pattern Enumeration and Classification (CAPEC) [9]，そしてシステムの弱点については Common Weakness Enumeration (CWE) [10]が存在する．また，情報および情報システムの脅威に関する付帯状況と連携させた標準化された記述法 Structured Threat Information eXpression (STIX)[11]がありサイバーセキュリティに関連する脅威や攻撃分析，特徴分析などについて共有するフレームワークが活用されている．

CVE,CAPEC,CWE の情報は構造化されており，XML データベースとしても利用可能な形で提供されている．XML 構文にはそれらの間の相互参照も定義されており，CAPEC に記載の攻撃が利用する弱点や脆弱性について，CWE や CVE の識別番号が記載されるなどしている．CVE, CAPEC, CWE の XML 構文で定義されている関係性を図 8 で示す．

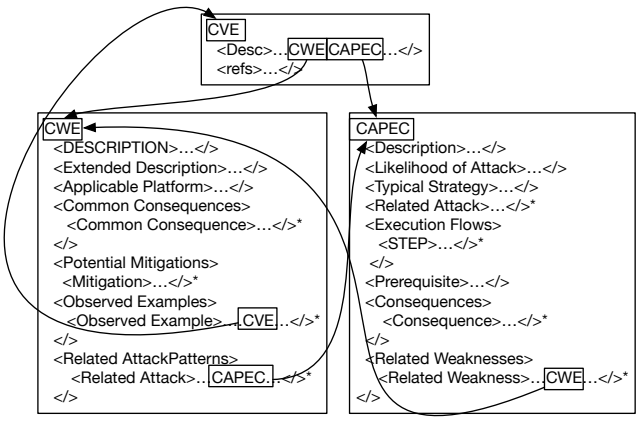

図 8　CVE-CAPEC-CWE 関係

CVE は脆弱性情報を記述する構造である．その記述は，Description の脆弱性に関する概要情報と，<refs>にある参考文献の記述によるところが大きく，構造化の度合いは低い．CAPEC は Execution Flow において手順が記載されている．Execution Flows の各項目からそれが利用する弱点や資源への記載規定がなく，あくまで単独の攻撃に関連する内容記述となっている．最後に CWE については，その性質上，さまざまな設計粒度に相当する弱点が記載可能なようにされている．しかし，弱点と，それに対する攻撃，さらに弱点を利用された場合の影響の対象の記載構造が要件を満たしていない．

STIX は具体的な脅威情報を共有化するために開発された構造化記述形式である．STIX の脅威記述は，脅威を構成する 18 の STIX Domain Object (SDO)，SDO の補助情報である STIC Cyber-observable Object (SCO)とそれらのオブジェクトの関連性を表す STIX Relationship Object (SRO)で構成されている．

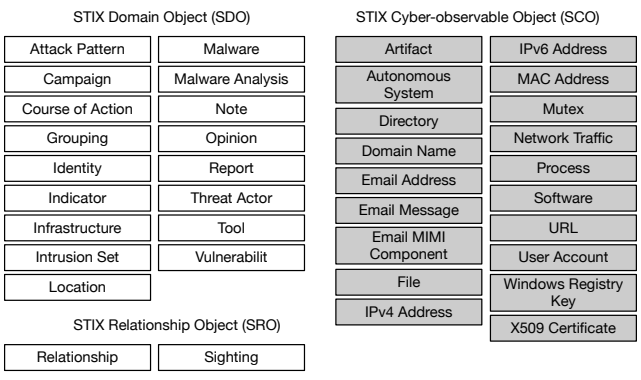

図 9　STIX Objects

Vol.2022-ARC-248 No.59
Vol.2022-SLDM-198 No.59
Vol.2022-EMB-59 No.59
2022/3/11

STIX の記述は SDO や SCO を SRO で結合して脅威を記述する．SRO には，関連を表記するためのさまざまな属性が設定可能である．また，内容表記のための記述文法も定められており，検索，類似事案の適用などの便を図っている．STIX は，攻撃のそれぞれの要素についてカテゴリ分けし，相互の関連性についての記述力も高い．また，記述対象であるオブジェクトの拡張や，記述方式のルール化なども定義されており，さまざまな脅威を攻撃手法や脆弱性，対照システムと連携させて記述するに十分なものである．しかし，開発における脆弱性の検証，評価においては，設計上や実装状の弱点が脆弱性となり，アタックサーフェスから利用可能であるかを示すことである．現状の STIX では，存在する脆弱性を連結して脅威を記述できるが，脆弱性そのものの構造の記述については，攻撃経路上の通常機能との関連など記述詳細度が不足している．また，STIX は設計の抽象度とリンクした階層構造や設計上の想定などを表すものは定義されていない. STIX を利用する場合でも拡張機能を用いた拡張が必要であると考える．

脅威モデリングに関して代表的な手法に STRIDE[12]などがある．キーワードを用いて脅威事象を導出し，その脅威事象が実現するための分析をアタックツリーなどの手法を用いて行う．アタックツリーは脅威が成立する要件を，分析レベルにおける設計情報などをもとに帰納的に導出する．STRIDE もアタックツリーもその条件導出には，脆弱性の存在は暗黙に想定している．また，その演繹的な性格から複数の攻撃を連携させた攻撃の発見（攻撃原因から）アタックツリーは脅威との連携は取れているが，脆弱性との関連性や先に述べた弱点との構造，階層の記述については詳細化の方向である．

システムのサイバーセキュリティの設計に用いられる脅威分析法のなかでは，PASTA[13]では脅威分析を行い，次に，既知脆弱性や弱いデザインパタンを抽出，その上で，脅威分析の際に抽出したアタックツリーのリーフ条件とそれらのマッピングを行う．そして，マッピングされたリーフ条件の攻撃モデリングをさらにアタックツリーなど用いて行う．このようにして，弱点や脆弱性，脅威，攻撃を統合的にあつかっている．しかし，情報システムを対象に手法が構築されており，CPS の分散開発への対応は明示的に言及されていない．

4. Cyber-Attack 情報構造と事例分析

ここでは先の説で述べた，問題点を解決するための，CPS に対する脅威，攻撃，弱点および脆弱性を含めた情報の記載に関する提案手法構造について記述する．

4.1 基本的な構成

まず，システムに対する攻撃の記述フレームワークを図 10 に示す． サイバー攻撃は，Attack Surface から攻撃結果までの連鎖で記述される．連鎖上には，攻撃ブロックと呼ぶ単位と，通常機能ブロックとよぶ単位で構成される．

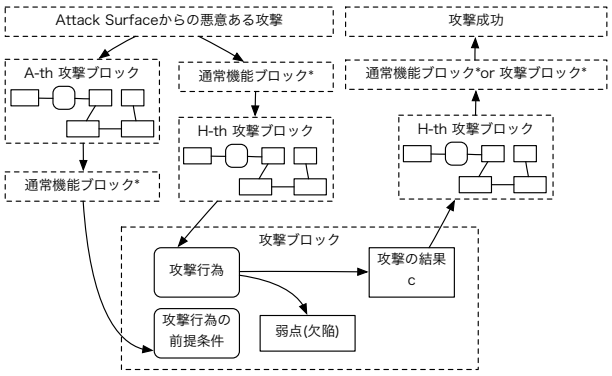

図 10　基本的な攻撃記述フレーム

脆弱性を利用した攻撃の記述は, 攻撃ブロックと呼ぶ単位を基本とする．攻撃ブロックは，攻撃内容の記述，その攻撃を実現するための条件，攻撃が利用する弱点，およびその弱点を利用した結果発生する結果を基本要素とする．攻撃行為と攻撃の結果、もしくは通常機能ブロックの出力が攻撃行為、もしくは前提条件への入力となり、その結果は、後続するブロックへの入力となる。

(1) 攻撃ブロック

攻撃ブロックの内容は図 11 で示される．

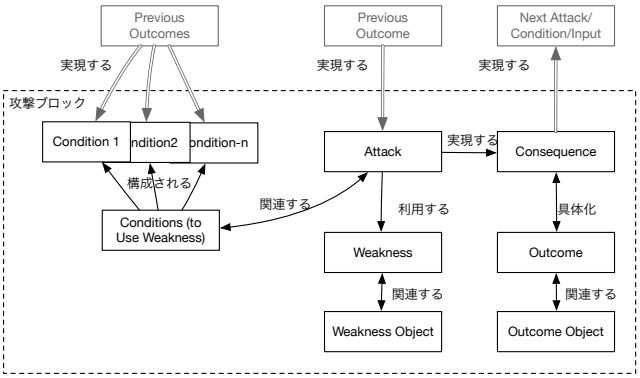

図 11 攻撃ブロック構造

記載項目は以下の通りとする．

- 攻撃内容：(Attack)

Weakness を利用する攻撃内容，すなわち処理の内容や操作方法，攻撃データを記載する．

- 攻撃実現条件：(Condition)

攻撃が対象とする弱点が攻撃可能となるための条件を記載する．この内容そのものが攻撃結果となる場合もある．

- 弱点：(Weakness)

攻撃の対象がもつ，攻撃されると好ましくない結果を産む弱点を記載する．弱点には欠陥，悪用および副作用の管理不足の３種類のパターンがある．

- 弱点対象(Weakness object)

弱点を含む対象．記載内容は記述粒度により異なる．

- 攻撃結果(consequence /outcome)：

攻撃の結果，可能となる結果を記載する．一般的な攻撃結果は consequence とし，それを攻撃結果対象に適用した

Vol.2022-ARC-248 No.59
Vol.2022-SLDM-198 No.59
Vol.2022-EMB-59 No.59
2022/3/11

場合の結果を outcome とする．result は CWE の common consequence に相当する．これは，次のブロックにリンクさせるときに Attack から参照，検索される．

- 攻撃結果対象：outcome object

攻撃が影響を及ぼした資産，機能などを記述する．出力結果だけならば，Outcome にそれを記載する．

なお、通常機能ブロックは、攻撃ブロックから弱点（Weakness）と弱点対象(Weakness Object)を省き、攻撃内容(Attack)を入力(Input)に置き換えたものである（図 12）。

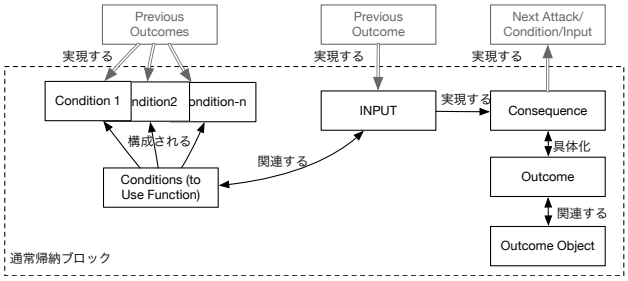

図 12 通常機能ブロック

(2) 仮想攻撃ブロック

攻撃連鎖の導出や，分散開発における開発者間の情報交換では，必ずしも，攻撃ブロックの存在が確定している場合のみではない．むしろ，攻撃や弱点の想定，前提条件などを記載して，それが実際に実現されるかどうかを検証する，もしくは，攻撃が可能かを下位の構成要素の担当者に判定してもらうための情報共有枠組みとして利用する場合には，情報の一部は確定しているとは限らない．攻撃ブロックの一部の情報が欠けたものを，仮想攻撃ブロックとよび，上に述べた用法にあてる．仮想攻撃ブロックには，欠けている情報部分に関する仮定要素を追加する．

- 仮定：(Assumption)

仮想ブロックで確定していない内容について，その内容が満たすべき条件や性質を記載する．攻撃の場合は，攻撃の満たすべき性質，例として引数指定が不正関数呼び出しで可能である，オーバーフローさせられるバッファの上限サイズなどを記載する．弱点については，想定されていが存在が確定していない弱点の性質，たとえば排他制御に欠陥があるなどを記載する．

(3) 攻撃ブロックの合成と分解

下流工程で見つかった弱点や脆弱性を上流工程の担当者と共有する場合，詳細な構造と関連する攻撃ブロックの内容を提示しても理解に必要な知識がない場合や，権利関係に抵触する場合などが想定される．そのような場合には，双方のインタフェース部分まで攻撃ブロックを連鎖させインタフェース部分から見える攻撃，攻撃条件，出力を提示する必要がある．このようにして，上流工程の粒度に合わせたり，内部の情報を隠蔽したりした攻撃ブロックを合成攻撃ブロックと称する（図 13）．逆に，上流工程で想定された仮想攻撃ブロックの内容を，設計情報や実装情報をもとに具体的な攻撃ブロックの組み合わせに分解してゆく操作を攻撃ブロックの分解と称する（図 14）．

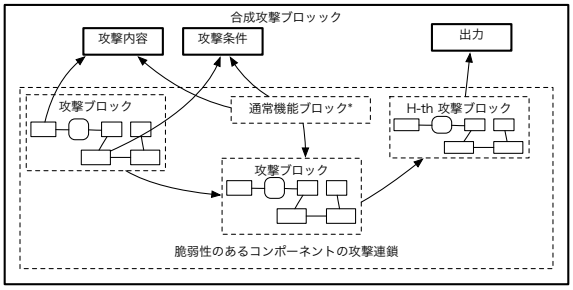

図 13 攻撃ブロック合成

上流工程に提示される部分が太わくで示されている。

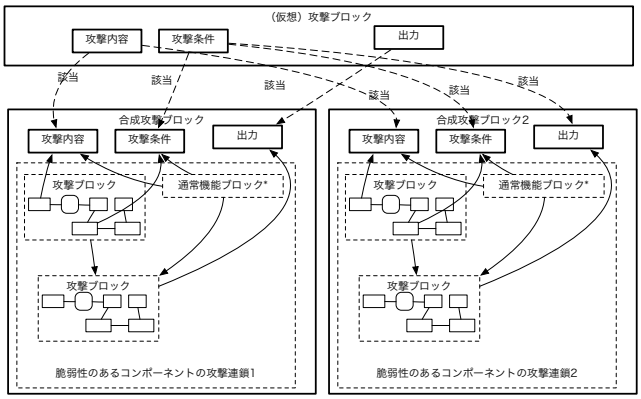

図 14 攻撃ブロック分解

分解に際しては、一つもしくは複数の合成攻撃ブロックを想定し､それの繰り返し詳細構造を利用して詳細化する。

システム，サブシステム，モジュールやコンポーネントなど，システムの粒度間でのサイバー攻撃の粒度を合わせることは，この合成／分解を適用して行う．

4.2 攻撃手段とそれに基づく攻撃手法の表記事例

攻撃の分析および,攻撃の開発を提案した記載で表記した例として BackHat 2019 USA における Wi-Fi モジュールの攻撃事例[14]を示す．攻撃対象を図 15 に示す．

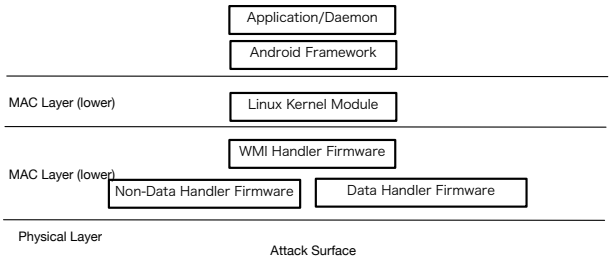

図 15　攻撃対象 WiFI システム

このシステムは，アーキテクチャレベルで GlobalMemory の Address Space Layout Randomization (ASLR)が設定されていないことおよび Flow Control Integrity が確保されていないという弱点があることが判明している．これらを利用した抽象的なレベルでの弱点の利用は，指定アドレスからの相対アドレスによるメモリアクセスおよび，アドレス指定による不正な関数呼び出しと想定される．前者の弱点利用を提案形式で表記すると図 16 となる．

Vol.2022-ARC-248 No.59
Vol.2022-SLDM-198 No.59
Vol.2022-EMB-59 No.59
2022/3/11

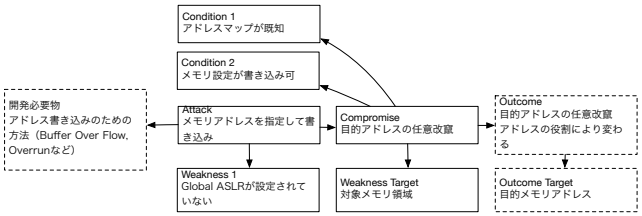

図 16　Global メモリの ASLR 対策不在による攻撃構造

前提条件として, ファームウエアは暗号化されておらず, バージョンアップとして配布されており，解析ツール類も入手可能ゆえ，Condition 1,2 は満たされている．したがって, 攻撃者はブロック内の点線で示された Assumption 要素内のアドレス書き込みのための方法，具体的には Memory Corruption の存在と，その Memory Corruption を目的とするメモリアクセス利用方法を探索することになる．

次に, Flow Control Integrity が保証されていないことの弱点利用の構造を図 17 示す.

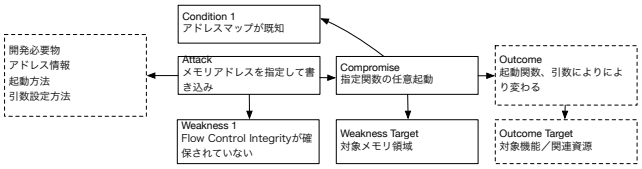

図 17　Flow Control Integrity 保証が無い場合の攻撃構造

これも，先と同じくファームウエア解析によりアドレスマップが既知であることから，Condition1 は満たされている．したがって，攻撃者は点線でしめされた Assumption 要素内の関数の起動方法および，初期の目的を達成するための起動関数とパラメータの設定方法を探索することになる.

攻撃者が実現を目指す脅威は，外部からシステムのシェルコードを起動して任意コマンドの実行であるとする.

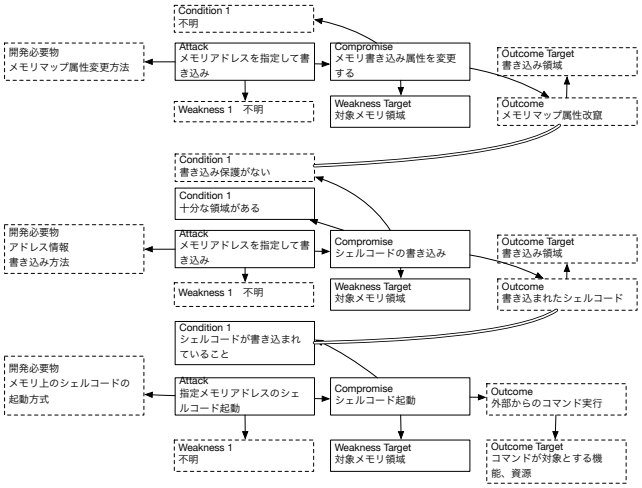

図 18 攻撃者の実現目標脅威の攻撃ブロック表記

この脅威の実現方法は，アタックサーフェスからの攻撃で図 18 の攻撃構造を実現することが目標となる. この脅威の構造に対して攻撃者が実現すべき項目は点線で囲まれた Assumption 部分である．実際の Black Hat の報告では，図，の攻撃構造を用い，Global Data 書き込みにより必要なパラメータを設定してメモリマッピング機能を，Flow Control Integrity の弱点を使って呼び出すことで書き込み禁止属性をはずすことで，攻撃を成立させた.

4.3 分散開発における脅威・脆弱性分析の共有

ここでは，大規模 CPS 開発でのサイバーセキュリティ分散開発における脅威，攻撃，弱点，脆弱性情報の共有を検討する．ここでは，機能要件導出・リスクアセスメントを実施する Consumer と機能要件実現および脆弱性評価を行う Supplier に分けてサイバーセキュリティ開発が行われるとする．前提として Consumer は機能実現を行うサブシステムに関する情報は持たず，Supplier は脅威分析やシステム全体および他サブシステムに関する情報は持たないとする．この分散開発において、Consumer の提示した要件や脅威は、分解を用いて Supplier に解釈される。逆に、Supplier の発見した脆弱性は、合成を用いて Consumer または、Supplier が開発した成果物の利用者に提示される。

(1) Consumer が懸念される脅威を発見した場合

Consumer は，脅威分析を実施，抽出された脅威事象に対しアタックツリーなどの手法で演繹的に脅威が実現するための条件を抽出するが、サブシステムの詳細情報を持たないため、この条件の成否が判定できない。よって、この条件を阻止すべきものとして Supplier に提示する。これを仮想攻撃ブロックとして Supplier は，攻撃ブロックの分解の手順に従い，合致する攻撃連鎖を持つ攻撃ブロック群を検索する．もし，ブロック群から作成された合成攻撃ブロックが成立した場合は、Consumer に提示する(図 19).

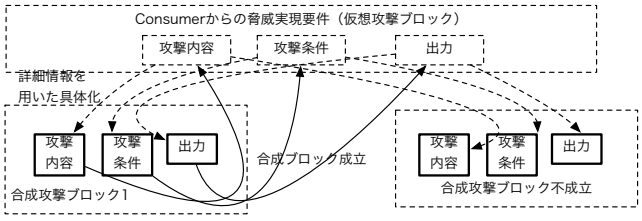

図 19 分解と合成をもちいた Consumer/Supplier 情報共有

相互に脅威分析の詳細や内部情報を提示することなく対応がなされる。

(2) Supplier がサブシステムレベルで弱点を発見した場合

サブシステムの設計，開発時に，セキュリティ上の弱点を発見する場合には，(a)実装や納期の制約上、弱点が解消できない，(b)構成要素が脆弱性の原因となる弱点を含む，(c)設計上，実装上の弱点のリリース後の発など考えられる.

いずれの場合も，Supplier は発見された弱点に対する可能な攻撃，条件および結果で攻撃ブロックを構築．それらを合成の手法で入力側と出力側に展開、サブシステムの境界に到達した時点で，合成攻撃ブロックが成立したとして，結果を Consumer に引き渡す．Supplier は内部情報の開示なくサブシステム境界の情報を Consumer に提示する（図 20）。

Vol.2022-ARC-248 No.59
Vol.2022-SLDM-198 No.59
Vol.2022-EMB-59 No.59
2022/3/11

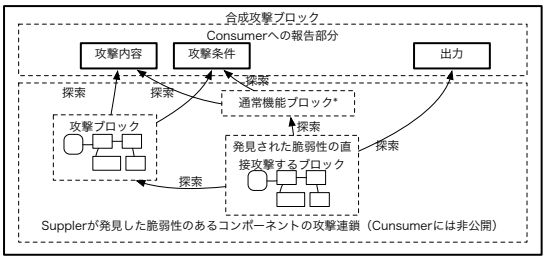

図 20 合成を用いた分散開発における Supplier 情報提示

5. 考察

脆弱性に関する情報は様々な組織により収集, 整理が進められている．攻撃手法も同様に，Exploit Database[15]や Kali Linux[16]に代表される収集や整理の作業が行われている．一方, 攻撃者が新たな脆弱性を見つけ出す過程については，セキュリティカンファレンスなどを見る限り系統的な整理状態にないと考えられる．ただし，文献[17]には攻撃者ごとに方法論があることが示唆されている．脅威や脆弱性と結びつけた攻撃の構造化は，攻撃者の手法を形式化することにつながる．本論文では 4 章で攻撃事例を攻撃者が実現させたい脅威事象と関連づけられることを示した．そこでは，攻撃者は実現脅威を構成し，それを仮想攻撃ブロックで表現し、探索すべき攻撃方法や条件実現を絞り込むとした．このように，攻撃者の手法を形式化する方法は，可能なセキュリティ検証方法として検討の余地があると考える．

分散開発で, サブシステム以下の開発で発見された脆弱性はシステムのリスク評価結果に影響を与える．脆弱性検証に相当する部分に ISO/SAE 21434 では Weakness を評価することを要求している．実際にリスク観点で行う Weakness の評価には，Weakness の利用が可能な経路存在検証と，その経路を用いた攻撃の可能性評価が必要となる．この可能性評価には， ISO/SAE 21434 では，攻撃可能性に対して CVSS[18]や Common Criteria[19]に基づくものなどの手法を提示している．しかし，手法間での価の整合性は保証されていない．本提案による脅威や脆弱性の記述方式は，複数の攻撃可能性評価が整合的である理由を与える可能性がある．提案記述方式では、システムへの攻撃は、ブロックのリンク構造で表されており、攻撃や条件設定の困難さや経路の複雑さに適切な重みづけが可能であれば攻撃可能性評価値の根拠とすることも可能と考える。また、攻撃は条件を揃えるための分析や条件設定、攻撃実行と分離されているため、実行に際しての条件や調査、設定の難しさと攻撃の難しさを分離して評価できる可能性がある. このような，上流設計のリスクに対する判断基準の数値を設計や実装側からのフィードバックで検証しつつ進めてゆく方式は検証の精度を高める意味から検討に値すると考える．

6. おわりに

Cyber-Attack に関する情報を構造的に表記，共有する仕組みは，分散開発時のみならず，製品のライフサイクルをとおして継続的に行われる収集・分析・対策においても基盤となると考える．また、サイバーセキュリティ情報を共有するため，近年，様々な企業が参加する Information Sharing and Analysis Center (ICAC)が業界分野ごとに設立されている．情報通信分野での IT-ISAC，車分野での Auto-ISAC，医療分野での Medical ISAC など設立され，日本でもそれらの対応組織が設立されており、企業を跨いだ情報共有の需要も高まっている．そのためには，脅威に関連する，攻撃や弱点，設計情報も含めた情報の整合的な整理が必要である．今後，さらなるサイバーセキュリティ攻撃の事例の分析検討を通じた記述性の検証と，本記述による設計作業との結合の整合性および，サイバー攻撃の事前検証可能性について事例をベースにした検討を進めたい．

参考文献

[1] ISO/SAE 21434 (2021), Road vehicles – Cybersecurity Engineering, International Standard Organization.
[2] ISO 26262 (2018), Road vehicles – Functional safety, International Standard Organization.
[3] Black Hat, https://www.blackhat.com, (参照 2022-2-1)
[4] Miller, C., Valasek, C. “Remote exploitation of an unaltered passenger vehicle,” Blackhat USA 2015, http://illmatics.com/Remote%20Car%20Hacking. pdf ,(参照 2022-2-1).
[5] Nie, S. Liu, L. and Du, Y. “Free-fall: hacking tesla from wireless to can bus,” Blackhat USA 2017, https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To- CAN-Bus-wp.pdf. (参照 2022-2-1).
[6] “平成 14 年度 耐タンパー性調査研究委員会報告書”,財団法人日本規格協会，平成 15 年 3 月
[7] “Meltdown and Spector”, https://meltdownattack.com. (参照 2022-2-1).
[8] https://cve.mitre.org/,(参照 2022-2-1).
[9] https://capec.mitre.org/,(参照 2022-2-1).
[10] https://cwe.mitre.org/,(参照 2022-2-1).
[11] https://oasis-open.github.io/cti-documentation/,(参照 2022-2-1).
[12] Adam Shostack, Threat Modeling designing for security, Willey 2014, ISBN 978-1-118-80999-0
[13] Tony UcedaVelez, Marco M. Morana, Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis, Wiley, May 2015, ISBN: 978-0-470-50096-5:
[14] Cong, X. Pi, P. “Exploiting Qualcomm WLAN and Modem Over the Air”, Black hat USA 2019, https://www.blackhat.com/us-19/briefings/schedule/#exploiting-qualcomm-wlan-and-modem-over-the-air-15481., (参照 2022-2-1)
[15] Exploit Database, https://www.exploit-db.com （参照 2022-2-1）
[16] Kali Linux, https://www.kali.org （参照 2022-2-1）
[17] Klein, T. “A Bug Hunter’s Diary: A Guided Tour Through the Wilds of Software Security (English Edition) 1st edition, No Starch Press, 2011, ASIN: B00652XO2I
[18] https://www.first.org/cvss/,(参照 2022-2-1).
[19] Common Criteria: Common Methodology for Information Technology Security Evaluation- Evaluation Methodology Version 3 Revision 5, Apr 2017.