CPS のサイバーセキュリティに求められる安全分析と STPA-Sec の有効性

Safety Analysis Required for CPS Cybersecurity and Effectiveness of STPA-Sec

福　島　祐　子

要　約　経済産業省は，CPS に対するサイバーセキュリティへの対応指針として，システム開発のライフサイクルにおいて，CPS の企画，設計，調達のような早期段階から，安全に関するリスク分析の結果を基にしたリスク対応を行うことの重要性を指摘している．

早期段階から適用できる安全分析手法として，STPA が米国で普及しており，これをセキュリティ向けに拡張した STPA-Sec という手法が考案されている．本稿では，STPA-Sec の適用方法を，分析例を示しながら解説し，システム開発のライフサイクルにおいて，早期段階でのセキュリティ要件・仕様への反映，セキュリティリスク分析における攻撃シナリオの導出に STPA-Sec が有効であることを示す．

Abstract　Regarding CPS, there are concerns that cyberattacks will increase in the future. As a CPS cyber security guideline, the Ministry of Economy, Trade and Industry has pointed out the importance of risk response from an early stage such as planning, designing, and procurement of CPS in the system development life cycle, based on the results of safety-related risk analysis.

STPA is widely used in the United States as a safety analysis method that can be applied from an early stage, and a method called STPA-Sec that extends STPA for security has been devised. In this paper, we explain how to apply STPA-Sec while showing the analysis results, and show that STPA-Sec is effective for reflection in security requirements and specifications at an early stage of development lifecycle as well as for deriving attack scenarios in security risk analysis.

1. は　じ　め　に

近年，IoT の進歩・普及に伴い，自動運転やスマートホームなど，実世界からさまざまな情報（他車状況，室温など）を収集・分析し，その結果を基に実世界の活動を支援（加減速，室温調整など）するサイバーフィジカルシステム（CPS）が広がりつつある．このようなシステムでは，機器，ソフトウェア，人といったさまざまなコンポーネントがつながることで，システム全体の構成が複雑化する．また，それらのコンポーネントが入れ替わるなど変化が生じると，コンポーネントが期待どおりに動作せず，システムをとりまく実世界にまで影響を及ぼし得る．そのような CPS が意図的なサイバー攻撃を受けると，物理的に甚大な影響を及ぼす可能性がある．従来の IT システムのセキュリティにおいては，情報の保護を重視してきたが，CPS のサイバーセキュリティにおいて守る対象には，情報だけではなく，実世界にある物理的なコンポーネント（人，機器，社会インフラなど）も含めなければならない．サイバー攻撃による物理的な被害の事例としては，イラン核燃料施設のウラン濃縮用遠心分離機破壊[1]，ウクライナの大規模停電[2], [3]，チェロキーのハッキング[4]など，数多く報告されている．

CPS に対するサイバーセキュリティへの対応指針として，2019 年に，経済産業省が「サイ

バー・フィジカル・セキュリティ対策フレームワーク」[5]を公開した．この中では，サイバー攻撃が被害をもたらす仕組みとして，脆弱性を突いてシステムを侵害するセキュリティ上の脅威が，危害の潜在的な源であるハザード要因となり，物理的な危害，機器の損壊等の安全上の問題につながる可能性を生じさせるとしている．そして，このような性質を持つサイバーセキュリティに対応するには，安全に関するリスク分析の結果を基に，セキュリティ・バイ・デザインの観点を踏まえ，システム開発のライフサイクルにおいて，CPS の企画，設計，調達のような早期段階からリスク対応を行うことが重要であると述べている[5]．

CPS 開発の早期段階から用いることができる安全分析手法として，STPA（System-Theoretic Process Analysis）[6], [7]が米国を中心に幅広い分野で普及している．この手法は，さまざまな環境における未知の（Unknown）事故原因，潜在的な原因の識別を目指す．そのため，システム思考をベースとし，環境まで含めシステム全体を捉え，抽象度の高いレベルから分析する．CPS のような今までに存在しないシステムの場合，未知のリスクが多いため，このような手法が有効と思われる．STPA を CPS のサイバーセキュリティに活用するために拡張した手法として，STPA-Sec が提案されている[8]．筆者は，本手法の有効性を確認するため，CPS のユースケースを用いて分析を試行した．

本稿では，2 章で安全分析手法 STPA と STPA-Sec の概要，3 章で STPA-Sec による分析方法と分析例について説明し，STPA-Sec の有効性に関する考察を述べる．

2. 安全分析手法 STPA と STPA-Sec の概要

本章では，CPS 開発の早期段階から用いることができる安全分析手法 STPA の概要と，STPA を CPS のサイバーセキュリティに用いるために拡張した手法である STPA-Sec の概要を説明する．

2. 1 安全分析手法 STPA の概要

STPA は，システム思考に基づく事故モデル STAMP（Systems Theoretic Accident Model and Processes）をベースとした安全分析手法である[6], [7]．米国では，航空・自動車・医療など様々な分野に広がりつつあり，国内では，JAXA が 10 年以上前から取り組んでいる[9]ほか，自動車業界[10]，鉄道業界[11]においても適用研究が進んでいる．業界標準への取り込みも進んでおり，航空では ASTM WK60748，SAE AIR6913 など，自動車では ISO/PAS 21448（SOTIF），SAE J3187 などから参照されている[12]．

STPA は，さまざまな環境における未知の事故原因，潜在的な原因を識別できるため，CPS のような新しいシステムのリスクを識別するのに役立つ安全分析手法である．安全分析には，従来から FTA（Fault Tree Analysis），FMEA（Failure Mode and Effect Analysis）などの手法がある．これらはシステムがまだ単純であった 1960 年代以前に作成されたものであり，事故は故障が原因で起きると想定している．それに対して STPA は，事故は，故障がなくてもコンポーネント間の相互作用から起き得ると想定し，コンポーネント間の不整合を分析する手法である．システム思考に基づく手法であり，環境も含めてシステム全体を捉え，抽象度の高いレベルから分析するため，さまざまなコンポーネントが複雑につながる CPS の安全分析に適している．

コンポーネント間の相互作用は，指示する側の上位のコンポーネントである“コントローラ”

から，指示を受ける側の下位のコンポーネントである“コントロール対象のプロセス”に対して“コントロールアクション”（Control Action．以降，CA と記載）を実行し，その結果を，コントロール対象のプロセスからの“フィードバック”あるいは“外部からの入力”を得ることにより行われる．コントローラは，フィードバックと外部からの入力を基に，次の CA を判断し実行する．たとえば，コントローラである「運転手」がコントロール対象のプロセスである「自動車」に対して，CA として「加速する」を実行すると，その結果として「先行車が近い」が運転手に入力され，これを基に，運転手は，CA として「ブレーキをかける」の実行を判断する．この CA が適切であれば事故は起きないが，もし「先行車が近いのに，ブレーキをかけない」という適切ではない CA を実行すると，「車が先行車に近づきすぎる」という危険な状態となり，事故につながる．前述したコントローラによる適切ではない CA のことを“非安全なコントロールアクション”（Unsafe Control Action．以降，UCA と記載），事故につながるシステムの危険な状態のことを“ハザード”と呼ぶ．そして，運転手が「先行車が近いのに，ブレーキをかけない」という UCA の実行を判断する原因として，実際のシステムの状態としては「先行車は近い」のに，運転手（コントローラ）は「先行車は近くない」と認識していることが考えられる．つまり，UCA の実行を判断する主な原因は，コントローラが認識しているシステムの状態とシステムの実際の状態が異なることにある．このコントローラが認識しているシステムの状態のことを“プロセスモデル”と呼ぶ．

STPA では，上記の考え方を基にハザード，ハザードにつながる UCA，最終的な事故原因として“損失のシナリオ”を識別する．UCA，損失のシナリオの識別においては，STPA が提供するガイドワード，ヒントワードを用いた発想により，未知の事故原因の識別を目指す．この未知の事故原因は，新たなリスクとしてリスク分析に取り込むことができ，そのリスク対応にも役立つ．並行して，ハザードを防ぐための“安全制約”と呼ばれる安全のためにシステムが守るべきルールを識別する．安全制約は，抽象度の高いシステムレベルの制約であるため，コンセプト策定，要求分析といったライフサイクルの早期段階で，システムの安全要件に反映できる．

2. 2　STPA-Sec の概要

STPA-Sec[8]は，STPA の分析結果を CPS のセキュリティ要件に反映し，セキュリティリスク分析に用いるために，STPA を拡張した手法である．STPA の分析結果を用いるため，ライフサイクルの早期段階で適用できる．STPA で識別した安全制約は，攻撃があっても守るべき安全上のルールであるため，STPA-Sec では，この安全制約をセキュリティ要件に反映させることにより，セキュリティに活用する考え方を示す．また，STPA では最終的に，損失につながる原因（プロセスモデルと実際のシステムの状態の不整合など）を損失のシナリオとして識別するが，STPA-Sec ではそのシナリオを故意に生じさせようとする攻撃者の視点から分析し，攻撃シナリオ（システムへの侵入など）を導出する．攻撃シナリオは，CPS のリスク分析の中で検討が必要とされているもの[5]であり，リスク対応に活用できる．このことから，CPS のサイバーセキュリティにおいて STPA-Sec が有効であると考える．

STPA-Sec の適用事例としては，考案者である Young 博士が所属する米国空軍の手法[13]や，航空関連の業界標準 RTCA DO 356A[12]がある．その他，MIT CAMS（Cybersecurity at MIT Sloan）[14]，クラウドセキュリティ事業を提供するアカマイ・テクノロジーズ[13]，航空機メーカー

であるエンブラエル[13]などで適用研究が進んでいる．

STPA-Sec では，STPA のステップ 1 からステップ 4 までをセキュリティの観点を加えて実施した後，STPA-Sec で攻撃シナリオを導出するために追加したステップ（本稿ではステップ 5 と記載）を実施する．なお，攻撃シナリオの導出については，2020 年に発表された具体的な方法[8]を取り入れた．その方法の詳細については，3.2 節の 5）で説明する．

STPA-Sec の分析ステップの概要を以下に示す．

1）ステップ 1：分析目的の定義

どのような損失の予防を，分析の目的とするのかを定義する．そのために，ステークホルダを交えて，受け入れがたい損失（事故など）を定義し，ハザード，システムの安全制約を識別する．この安全制約は，システム全体のセキュリティ要件に反映させることができる．

2）ステップ 2：コントロールストラクチャの作成

システム全体の環境も含めた機能的なコンポーネントを特定し，機能コントロールストラクチャ（Functional Control Structure．以降，FCS と記載）によりコンポーネント間のコントロール関係を明確にする．制約を守るためのコンポーネントの責任を明確にし，コントローラからコントロール対象のプロセスへの CA を識別する．

3）ステップ 3：UCA の識別

CA がハザードにつながる条件（タイミングなど）を導出し，UCA を識別する．その条件をさらに組み合わせることで，より精緻なハザード判定の条件を識別する．UCA から識別したコントローラの安全制約は，コンポーネント単位のセキュリティ要件・仕様に反映させることができる．

4）ステップ 4：損失のシナリオの識別

UCA につながる原因を識別することで，ハザードを引き起こし損失につながるシナリオを識別する．ここで識別したシナリオは，セキュリティがおびやかされることによっても起きる可能性があるため，セキュリティリスク分析において検討する攻撃シナリオの導出にも役立つ．

5）ステップ 5：攻撃シナリオの導出

サイバー攻撃を行う攻撃者の立場から，ステップ 4 で識別した損失のシナリオを実現するための攻撃の種類，目的，影響，攻撃タイミングを攻撃シナリオとして導出する．これにより，多くの攻撃方法が明らかとなり，そこから対策立案につなげることができる．

ステップにおけるアウトプットの関係を図 1 に示す．

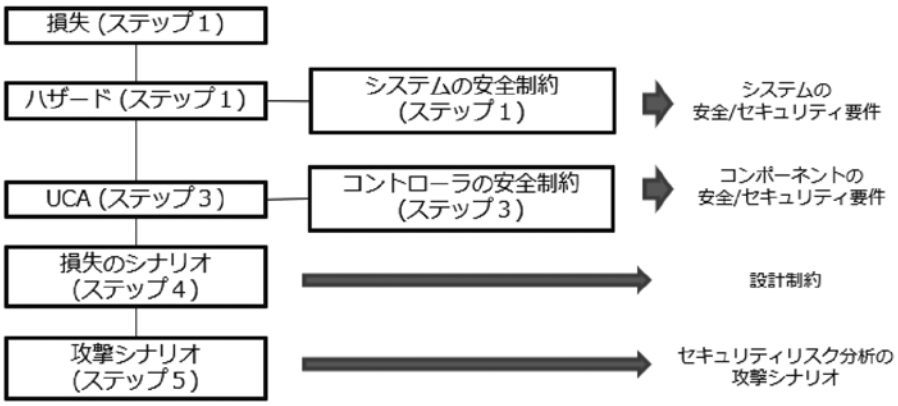

図 1　STPA-Sec のアウトプットの関係（STPA HANDBOOK[7] を参考に作成）

3.　STPA-Sec による分析の試行

本章では，STPA-Sec の適用方法を，分析例を示しながら解説する．まず，分析対象とする「自動運転車」のユースケースについて説明し（3.1 節），次に，「自動運転車」のユースケースへの STPA-Sec の適用を試行した結果を示しながら，STPA-Sec の適用方法について解説する（3.2 節）．その後，ⅰ）開発の早期段階へのセキュリティ要件・仕様の反映，ⅱ）攻撃シナリオの導出の観点から，STPA-Sec の有効性を示す（3.3 節）．

3.1　分析対象システム

分析対象とする「自動運転車」は，NIST（National Institute of Standards and Technology：アメリカ国立標準技術研究所）が公開している文書「SP800-160 Vol. 2: Developing Cyber Resilient Systems: A Systems Security Engineering Approach」[15] に掲載されたユースケースであり，周辺状況をセンシングした結果を基に自動走行する CPS のシステムである．このユースケースでは，記載を設計前のコンセプトレベルにとどめており，開発の早期段階であるコンセプト策定の成果物と同程度の抽象度であると想定できるため，開発の早期段階への適用をシミュレーションできる．また，ユースケースに攻撃シナリオが含まれていることから，STPA-Sec によって導出できる攻撃シナリオとの違いを確認することができる．NIST は，物理，化学などの幅広い分野の研究機関であるが，情報技術分野において，連邦政府機関および米国業界向けのサイバーセキュリティ標準の開発・普及を行っている．

「自動運転車」のユースケースの抜粋を以下に示す[15]．

> *自動運転車のミッションは，オペレーターが指定した場所への安全でタイムリーな輸送を提供することである．分析では，車両を自律システムとして捉えて分析対象とする．*
>
> *自動運転車のアーキテクチャには以下が含まれる．*
>
> - *パワートレイン（エンジン管理，ブレーキ）*
> - *安全機能（アダプティブクルーズコントロール（ACC））*
> - *テレマティクスとインフォテインメント（ラジオ，GPS）*
> - *統合されたコンポーネント（センサー，コントローラ，通信）*

上記を基に導出された攻撃シナリオは，以下のとおりである．

- *攻撃シナリオには，車両システムを乗っ取って衝突を引き起こしたり，車両を盗んだり，乗客を誘拐したりすることが含まれる．そのために，攻撃者は以下を行う．*
  - *インフォテインメントシステムにマルウェアのダウンロードを命令する*
  - *マルウェアは，コマンドを CAN バス[*1] に挿入，偵察，制御などの目標を達成する*
  - *マルウェアを利用して，以下を行う*
    - ➢ *テレマティクスで車両の位置を追跡する*
    - ➢ *サイバー影響（データ変更，サービス拒否など）を達成する*
    - ➢ *意図した物理的影響（衝突，盗難など）を達成する*

本分析では，「自動運転車」のアーキテクチャに含まれている安全機能である ACC に焦点を当てる．その理由は，ACC は自動運転を実現する自律的な制御を行う主要な機能の一つだからである．

3. 2　STPA-Sec の適用方法と分析例

STPA-Sec のステップごとに，適用方法と分析例を説明する．

1）ステップ 1：分析目的の定義

分析目的の定義では，システムの損失，ハザードと安全制約を以下のとおり識別した．

損失：

A1：車両内外の人の死傷

A2：車両および施設の損壊，盗難

ハザード：

H1：他車・障害物に近づきすぎる

安全制約：

SC1：他車・障害物との間に適切な距離を保たなければならない

2）ステップ 2：FCS の作成

ステップ 1 で識別した安全制約を守らせるための，システムの FCS を作成する．物理的なコンポーネントではなく，抽象度の高い機能レベルでコンポーネント間の関係を捉える．「自動運転車」のアーキテクチャの中の ACC に焦点を当てた分析例を，図 2 に示す．

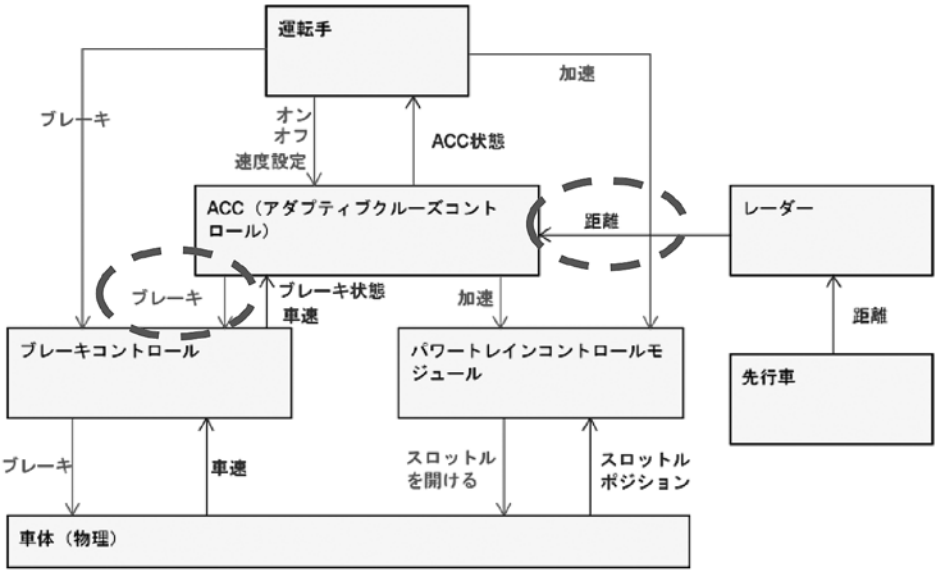

図 2 「自動運転車」の FCS（STPA HANDBOOK[7] を参考に作成）

まず，コントローラとコントロール対象のプロセスを識別する．図 2 では，コントローラ「運転手」とコントロール対象のプロセス「ACC」，「ACC」と「ブレーキコントロール」というように，コントローラとコントロール対象のプロセスを識別している．次に，安全制約を守らせるためのコントローラの責任を明確にし，そこから CA を識別する．コントローラ「ACC」の責任および CA を以下に示す．

「ACC」の責任：
　先行車との距離が近いときには，ブレーキを指示しなければならない
「ACC」から「ブレーキコントロール」への CA：
　ブレーキを指示する

ACC はブレーキを判断する際に，先行車との距離を知らなければならないので，「先行車との距離」をプロセスモデルとして導出する．次に，正しくプロセスモデルを捉えるために，先行車との距離の入力を識別する．

以降のステップでは，「ACC」の CA「ブレーキを指示する」を対象にして分析する．

3）ステップ 3：UCA の識別

CA ごとに，以下の四つのガイドワードを手掛かりにして，幅広く UCA を識別する．

A）どのような条件で，CA を指示しないとハザードにつながるか？
B）どのような条件で，CA を指示するとハザードにつながるか？
C）CA を誤ったタイミング（早すぎる，遅すぎる），あるいは誤った順序で指示するとハザードにつながるか？
D）CA をやめるのが早すぎる（アクションが短すぎる），あるいは遅すぎる（アクションが長すぎる）とハザードにつながるか？

CA「ACC は，ブレーキコントロールにブレーキを指示する」を対象とした UCA の識別例を表 1 に示す．

表 1　UCA の識別

ガイドワード / コントロールアクション	A）「指示しない」と ハザードにつながる	B）「指示する」と ハザードにつながる	C）「誤ったタイミング/順序」が ハザードにつながる	D）「やめるのが早すぎる/遅すぎる」と ハザードにつながる
ブレーキ	（UCA1）先行車が近いのに，ブレーキをかけない［H1］	（UCA2）先行車が近いときに，ブレーキを不十分にかける［H1］	（UCA3）先行車が近いのに，ブレーキをかけるのが遅すぎる［H1］	（UCA4）先行車が近いのに，ブレーキをかけるのが短すぎる［H1］

たとえば，「A）指示しないと ハザードにつながる」というガイドワードを適用し，ACC がどのような条件（「先行車が近いとき」など）で「ブレーキ」を指示しないとハザードにつながるかを考えることで，「（UCA1）先行車が近いのに，ブレーキをかけない」という UCA を識別できる．そして，この UCA を反転させることで，コントローラの制約を識別する．UCA1 を反転させると，ACC の制約として以下を導出できる．

（UCA1）先行車が近いのに，ブレーキをかけない
　—> ACC（コントローラ）の制約：
　　先行車が近いときには，ブレーキをかけなければならない

FCS は，機能構造を表しているので，物理的な制約を受けることはない．そのため，この段階で識別した制約をシステム開発に反映することにより，開発の早期段階で，どのような条件のときに CA を指示すべきか，あるいは指示すべきではないかというコントローラの要件を満たす構造を検討できる．さらに，この条件を幅広く見つけて組み合わせることにより，より精緻にハザードにつながる条件を捉えることができる．たとえば，ブレーキをかけないときの条件として，先行車との距離以外に天候や速度も考えられるため，これら条件の組み合わせによりハザード判定を行うことができる（表 2）．そして，ここから識別した制約を，開発の仕様にロジックとして反映させることができる．

表 2　条件の組み合わせによるハザード判定

コントロールアクション	先行車との距離	天候	速度	ハザード？
ブレーキをかけない	**近すぎる**	晴	規定内	**Yes**
	近くない	**雨**	規定内	**Yes**
	近くない	晴	**規定超**	**Yes**

4）ステップ 4：損失のシナリオの識別

ステップ 3 で識別した UCA の原因を特定し，さらにその原因について，複数のヒントワードを用いて漏れのないように深掘りすることで，損失のシナリオを識別する．

「(UCA1) 先行車が近いのに，ブレーキをかけない」の原因として，実際のシステムの状態である「先行車は近い」に対して，ヒントワード「プロセスモデルの不一致」を用い，プロセスモデルでは「先行車は遠い」と認識している不一致があるのではないかと考えることにより，「ACC は，先行車は遠いと認識しているため，先行車が近いのにブレーキをかけない」というシナリオを識別できる．さらに，ヒントワード「入力の遅延」を用いて深掘りすることにより，「先行車との距離の入力の遅延により，先行車は遠いと認識する」というように損失のシナリオを識別できる．

損失のシナリオ：
ACC は，先行車は遠いと認識しているため，先行車が近いのにブレーキをかけない
－＞　先行車との距離の入力の遅延

識別した損失のシナリオに対して対策を検討することで，開発の設計制約に反映させることができる．

5）ステップ 5：攻撃シナリオの導出

ステップ 4 で識別した損失のシナリオを基に，攻撃者の立場から，a) 攻撃の影響，b) 攻撃のターゲット，c) 攻撃の種類，d) 攻撃の目的，成功する条件を特定する．これにより，損失のシナリオから攻撃シナリオを導出できる．

a）攻撃の影響

分析中のコンポーネント ACC に対する攻撃が成功したときに，攻撃者が与えたい影響は何かを考える．それによりシナリオから，「先行車との距離の入力を遅延させる」という影響を導出することができる．

b）攻撃のターゲット

FCS を確認し，攻撃はどの要素に影響を与えたらよいか，つまり攻撃のターゲットを考える．たとえば，先行車との距離を測る「レーダー」をターゲットとして，「先行車との距離の入力を遅延させる」という影響を与えられると考える．さらに，「ACC」に対しても「先行車との距離の入力の処理を遅延させる」という影響を与えられる，というように分析を深めることができる（図 3）．

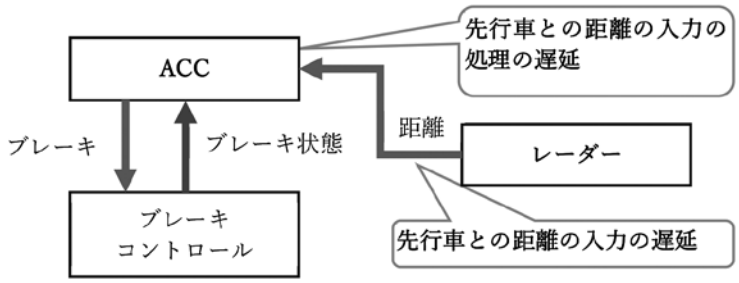

図 3　攻撃のターゲット

c）攻撃の種類

a）で特定した影響に対して，どのような攻撃を加えると影響を起こし得るか，つまり攻撃の種類を，STRIDE を使用して検討する（表 3）．STRIDE とは，6 種類の脅威（なりすまし，改ざん，否認，情報漏洩，DoS 攻撃，権限昇格）により，脅威を識別する手法である．レーダーに対して，DoS 攻撃，あるいは改ざんの攻撃を加えると，「先行車との距離の入力を遅延させる」影響を与えられるというように，攻撃とその影響を漏れのないようにあげることができる．

表 3　攻撃の種類

攻撃のターゲット	攻撃の種類	攻撃の影響
レーダー	DoS 攻撃	先行車との距離の入力の遅延
	改ざん	
ACC	DoS 攻撃	先行車との距離の入力の処理の遅延

d）攻撃の目的，成功する条件

損失のシナリオごとに，攻撃の目的，攻撃成功の条件は何かということを検討する．攻撃の目的としては，「先行車が近いのに，ブレーキをかけない」を導出することができ，さらに攻撃が成功する条件，つまりタイミングとしては，「先行車が近いとき」というように見つけることができる．

上記の a）～ d）の分析から，以下の攻撃シナリオを導出できる．

攻撃シナリオ：

X1）先行車が近いときに，レーダーに対して DoS 攻撃を行い，先行車との距離の入力を遅延させる．X2）これにより，先行車が近いときに ACC にブレーキをかけさせないようにし，衝突させる．

「X1）先行車が近いときに，レーダーに対して DoS 攻撃を行い，先行車との距離の入力を遅延させる」は，ユースケースの攻撃シナリオの中のサイバー影響を，いつ，何に対して，どのような攻撃をしかけるか，という観点で詳細化している．また，「X2）先行車が近いときに ACC にブレーキをかけさせないようにし，衝突させる」は，物理的影響を，いつ，どのように物理的に制御するか，という観点で詳細化したものと捉えられる．

この攻撃シナリオを基に，回避策を定義し FCS に反映し，さらにそれに対する攻撃シナリオを検討するというように，反復的に進める．

3. 3　STPA-Sec の有効性

3. 2 節で示した「自動運転車」のユースケースへの STPA-Sec の分析例を基に，ⅰ）開発の早期段階へのセキュリティ要件・仕様の反映，および ⅱ）攻撃シナリオの導出の観点から，STPA-Sec の有効性を示す．

ⅰ）開発の早期段階へのセキュリティ要件・仕様の反映

分析例では，ユースケースのミッション，アーキテクチャといった抽象度の高い入力を基に，分析を実施できている．ステップ 1 ではシステムのハイレベルな安全制約を識別できた（3.2 節 1）安全制約 SC1）．これは，攻撃のあるなしに関わらず，システムが守るべきセキュリティ要件であり，コンセプト策定，要求分析といった早期段階において開発のベースとすることができる．同様に，ステップ 3 では，ステップ 2 で識別した抽象度の高いコントローラの安全制約を識別した（3.2 節 3）ACC の制約）．この制約は，コンポーネントのセキュリティ要件に反映できる．

また，ステップ 3 において識別したハザードにつながる条件の組み合わせ（3.2 節 表 2）を用いることで，ハザードを回避するための判断ロジックを導出できる．この判断ロジックは，アーキテクチャ設計におけるセキュリティ仕様策定に役立つものである．

なお，STPA-Sec では，上記の安全制約およびハザードを回避するための判断ロジックを，未知のものを含めて導出できる．このことにより，たとえば，自動運転の分野における RSS（Responsibility-Sensitive Safety：責任感知型安全論）[16]のような事前に決定された安全のためのルールを検証する手法などにおいて，新たなルールを導出する際に活用できると思われる．

ⅱ）攻撃シナリオの導出

分析例では，ステップ 4 で識別した損失のシナリオを基に，ステップ 5 において，攻撃シナリオを導出できること，具体的には，「自動運転車」のユースケースの攻撃シナリオの中のサイバー影響，物理的影響の詳細化に効果があることを確認できた（3.2 節 5））．特に，攻撃が成功する条件（3.2 節 5）d））は，ステップ 3 においてガイドワードを用いて識別した UCA（3.2 節 3）UCA1）に含まれる条件を端緒として導出している．攻撃成功の条件を重視する理由は，攻撃者はシステムに潜伏しており，タイミングを見計らって攻撃することを想定しているからである．また，攻撃の影響（3.2 節 5）a））は，ステップ 4 においてヒントワードを用いて損失のシナリオ（3.2 節 4）損失のシナリオ）を深掘りすることにより導出されている．このように，ガイドワード，ヒントワードを用いて発想することで，できるだけ幅広く，漏れのないように捉えた損失のシナリオを基にして分析できる点が，STPA-Sec の大きな利点である．

一つの損失シナリオからは，攻撃のターゲット（3.2 節 5）b）），攻撃の種類（3.2 節 5）c））が複数見つかるため，さらに多くの攻撃シナリオを導出することができる．また，損失シナリオの中でも優先度の高いものから分析することにより，攻撃シナリオを効率的に導出することができる．

4. お わ り に

本稿では，サイバーセキュリティへの対応における安全分析の必要性について述べ，それに適した手法として STPA-Sec を紹介した．サイバーセキュリティの領域は，今後ますます重要性が増し，その影響範囲も広がることが予想される．STPA-Sec では，抽象度の高いレベルから，損失につながらないようにシステムをコントロールするというシステム思考を取り入れることにより，早期段階でセキュリティ要求・仕様および攻撃シナリオを導出することができ

る．したがって，サイバーセキュリティに望まれる，システムの上流工程からの方策の組み込みを目指すセキュリティ・バイ・デザインの実現に役立つものと考える．試行したい方は，ぜひ日本ユニシスにご相談いただきたい．

本稿で紹介した，STPA-Sec によって導出した攻撃シナリオは，セキュリティリスク分析の有効な手段である．しかし，これ以外にもミスユースケースや攻撃ツリーなど多くの手法があるため，STPA とそれらの手法をシームレスにつなげることができれば，多様なケースに対応した安全とセキュリティのための統合された分析手法が実現できると考えている．今後はさらに，さまざまなセキュリティリスク分析手法との連携，NIST のセキュリティ文献の調査などを進め，統合分析手法の確立を目指す．

最後に，本稿の執筆にあたり，多くの方々にご助言とご指導をいただいた．この場を借りて深く感謝申し上げる．

＊ 1　車載コンピュータ間で制御信号を送受信するバス．CAN は Controller Area Network の略．

参考文献

［1］ IPA セキュリティセンター，“制御システム関連のサイバーインシデント事例 4　～ Stuxnet：制御システムを標的とする初めてのマルウェア～”，IPA，2020 年 3 月，https://www.ipa.go.jp/files/000080701.pdf
［2］ IPA セキュリティセンター，“制御システム関連のサイバーインシデント事例 1　～ 2015 年ウクライナ 大規模停電～”，IPA，2019 年 7 月，https://www.ipa.go.jp/files/000076755.pdf
［3］ IPA セキュリティセンター，“制御システム関連のサイバーインシデント事例 2　～ 2016 年ウクライナ マルウェアによる停電～”，IPA，2019 年 7 月，https://www.ipa.go.jp/files/000076756.pdf
［4］ 中野学，“IoT 時代のセキュリティ”，IPA，2016 年 3 月，https://www.ipa.go.jp/sec/old/users/seminar/seminar_osaka_20160326-02.pdf
［5］ “サイバー・フィジカル・セキュリティ対策フレームワーク”，経済産業省 商務情報政策局 サイバーセキュリティ課，2019 年 4 月，P31 ～ 47 https://www.meti.go.jp/press/2019/04/20190418002/20190418002-2.pdf
［6］ Nancy G. Leveson, “Engineering a Safer World”, The MIT Press, 2011
［7］ Nancy G. Leveson and John P. Thomas, 白坂成功ほか訳, “STPA Handbook”, 2018.3, http://psas.scripts.mit.edu/home/get_file2.php?name=STPA_handbook_japanese.pdf
［8］ William Young, “BASIC INTRODUCTION TO STPA FOR SECURITY (STPA-SEC)”, 2020.7, MIT 2020 STAMP Workshop, http://psas.scripts.mit.edu/home/wp-content/uploads/2020/07/STPA-Sec-Tutorial.pdf
［9］ 片平真史，石濱直樹，“実利用段階に入った STAMP/STPA　～ Prologue to Autonomous System ～”，IPA 第 3 回 STAMP ワークショップ，2018 年 12 月，https://www.ipa.go.jp/files/000071001.pdf
［10］ 宮崎義弘ほか，“自動運転システムへの STPA 試行事例　～ JASPAR 機能安全 WG 活動成果紹介～”，IPA 第 3 回 STAMP ワークショップ，2018 年 12 月，https://www.ipa.go.jp/files/000070993.pdf
［11］ 北村知，“鉄道システムの安全性検証に STAMP を使ってみて”，IPA 第 3 回 STAMP ワークショップ，2018 年 12 月，https://www.ipa.go.jp/files/000071002.pdf
［12］ John P. Thomas, “STPA in Industry Standards”, MIT STAMP Workshop 2020.7, http://psas.scripts.mit.edu/home/wp-content/uploads/2020/07/JThomas-STPA-in-Industry-Standards.pdf
［13］ William Young, “Leveraging Secure Systems Analysis to Generate Testable Cybersecurity Requirements”, ITEA CYBERSECURITY WORKSHOP, 2018.3, https://www.itea.org/images/pdf/conferences/2018_Cybersecurity/Proceedings/Young%20ITEA%208%20Mar%20Final%20as%20delivered.pdf
［14］ Nancy W. Stauffer, “Protecting our energy infrastructure from cyberattack”, MIT

News, 2019.6,
http://news.mit.edu/2019/protecting-our-energy-infrastructure-from-cyberattack-0604
［15］ Ron Ross, et al., “SP 800-160 Vol. 2 Developing Cyber Resilient Systems: A Systems Security Engineering Approach”, NIST, 2019.11, P38, P174-178,
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v2.pdf
［16］ Shai Shalev-Shwartz, et al., “On a Formal Model of Safe and Scalable Self-driving Cars”, Mobileye, 2017, https://arxiv.org/pdf/1708.06374.pdf

※上記参考文献に挙げた URL は 2021 年 8 月 4 日時点での存在を確認.

執筆者紹介　福　島　祐　子（Yuko Fukushima）

1985 年日本ユニシス(株)入社．汎用機の基本ソフトウェア開発保守を担当後，大規模システム開発の標準化，プロセス策定・適用に従事．2015 年より総合技術研究所に所属．システムズエンジニアリング，MBSE，STAMP/STPA の適用研究に従事．(独)情報処理推進機構（IPA）ソフトウェア高信頼化センター「ソフトウェア高信頼化推進委員会」委員（2017，2018 年），AI/IoT システム安全性シンポジウムプログラム委員 (2019 年－現在),「CAST HANDBOOK」Nancy G. Leveson 著，共訳（2021 年），STAMP 関連の論文・講演多数．INCOSE 会員.