STAMP海外事例の紹介：STPA-SafeSec

仙台高等専門学校／IoTシステム安全性向上技術WG委員　岡本 圭史
国立大学法人信州大学／IoTシステム安全性向上技術WG委員　岡野 浩三

本稿では、STAMP海外事例としてSTPA-SafeSecを紹介する。STPA-SafeSecは、安全性と脆弱性を統合して分析するためのSTPA拡張である。また、とくに脆弱性分析をSTPAベースで実施する際に有用となる関連事項を併せて紹介する。

1　はじめに

STAMP（System Theoretic Accident Model and Processes）はシステム理論に基づく事故モデルであり、STPA（System Theoretic Process. Analysis）はSTAMPに基づくハザード分析（安全分析）手法である［1］［2］。STPAは既存の安全分析手法で分析が困難であった複雑な対象に対し有効であると言われている［1］。

走行中の自動車のエンターテインメント系から走行系への乗っ取り、コンピューター・ワームによる遠心分離機の破壊といった、セキュリティ侵害が安全性を脅かす事例がある。これらの事例は、STAMP/STPAの用語を用いれば、セキュリティにかかわるハザード誘発要因（Hazard Causal Factor、HCF）が最終的に安全制約を破るという事例である。このような事例を分析するためには、安全分析とセキュリティ分析を統合したSTPAの拡張が必要となる。

前述の事例は、STPAのstep0準備1において安全性にかかわるアクシデント、ハザード、安全制約を識別し、step2のHCF特定のヒントとしてセキュリティにかかわるヒントを導入するだけで、分析できそうに思える。しかし、セキュリティ分析にはシステムの詳細情報が必要なことが多く、STPAで使用するコントロールストラクチャー図（Control Structure Diagram、以下CSD）がセキュリティ分析に十分であるかといった検討は必要であろう。このような背景の下に、STPAの拡張として、STPA-Sec［3］やSTPA-SafeSec［4］が提唱されてきた。

本稿では、具体的な事例（マイクログリッドにおける広域電力網と局所電力網の接続（併入）をSTPA-SafeSecで分析）を用いてSTPA-SafeSecの手順が説明されている論文を紹介する。はじめに、STPA-SafeSecの特徴と手順を紹介し、次にSTPA-SafeSecの適用事例を紹介する。更に、とくに脆弱性分析をSTPAの拡張で分析する際に有用な事項を紹介する。

なお本稿では、STPA-SafeSecで用いられている用語を、標準的STAMP/STPAの用語に著者の解釈で置き換えている。

2　STPA-SafeSecの概説

本節では、文献［4］で提案されているSTPA-SafeSecを紹介する。STPAは安全性分析を目的としているが、STPA-SafeSecは、安全制約と脆弱性を統合して分析できるSTPA-SafeSecの拡張であり、文献［4］では、STPA-SafeSecの詳細な手順が提案されている。本節ではスペースの関係から、STPA-SafeSecの手順詳細を割愛し、標準的STPAの手順［2］に合わせて解説する。

2.1　STPA-SafeSecの特徴

STPA-SafeSecでは以下の上2つが貢献として挙げられている。また本稿では、1つ目の貢献からの派生効果であるが3つ目も貢献として挙げる：

1. 機能CSDと物理CSDを持つ
2. Step2で使用するHCFヒントのセキュリティ拡張
3. 安全性・セキュリティ対策の統合

これらの特徴について、それぞれ述べる。

機能レイヤ（Control Layer）のCSD（以下、機能CSD）内のコントロールループごとに構築する物理レイヤ（Component Layer）のCSD（以下、物理CSD）を用いることで、step2でセキュリティ侵害の経路が特定しやすくなる。例えば、上位の機能CSDの時刻同期機能は、下位の物理CSDではGPSに詳細化されたとする。この詳細化により、GPSの既存の脆弱性をHCFとして利用できる。また機能CSDと物理CSD間のコンポーネントを対応付けることで、機能CSDで特定したUCAから、物理CSDにおいて識別するそのUCAへ至るHCFとハザードシナリオとの対応が容易になる。更に、物理CSDを考えることで、既存の脆弱性分析を活用するには、抽象化した機能レベルの分析では限界があり、この事例のような物理CSDの導入が必須となる。

標準的STPAは安全性を分析するために、アクシデント、ハザード、安全制約を識別し、最終的にHCFとハザードシナリオを特定する。HCFを特定する際には、コントロールループ中で安全性にかかわるHCFヒントとして、コンポーネント故障、ヒューマンエラー、コミュニケーションエラー、ソフトウェア不具合、要求仕様不具合などを用いることが一般的である。STPA-SafeSecでは、これら従来のヒントにセキュリティにかかわるHCFヒントを追加し、HCFとしてなりすましなどのセキュリティにかかわる誘発要因を特定できるようにしている。

なお、STPA-SafeSecで採用されているセキュリティにかかわるHCFヒント以外では、例えば、セキュリティにかかわるヒントとしてSTRIDE［5］の利用が考えられる。

STPA-SafeSecのstep2では、抽象的ハザードシナリオから具体的ハザードシナリオを導出し、安全制約やセキュリティ制約を特定している。この導出方法により、ハザードシナリオ（安全・セキュリティ制約）たちは木構造となる。この木構造を分析することで、安全制約とセキュリティ制約間の関係が明らかになり、これらを統合できる。例えば、機能CSDであるフィードバックが間違っていることがUCAの指示につながり、ひいてはハザードを引き起こすこというシナリオ1が策定できたとする。更に、この機能CSDを詳細化した物理CSDにより、サイバー攻撃によりそのデータが改ざんされ、同じUCAへ至るというシナリオ1.1を特定できたとする。このとき、安全の観点から導入されたデータチェック機構は改ざん検出にも利用できるため、シナリオ1の安全対策がシナリオ1.1のセキュリティ対策を兼ねることになる。

2.2 STPA-SafeSecの手順

STPA-SafeSecは詳細な手順に分割されている（［4］図3）。本稿では、標準的STPAのプロセスである［2］で解説されている手順にまとめSTPA-SafeSecを解説する。

Step 0準備1（STPA-SafeSec II～IV）：対象とするシステムのロス（アクシデント）、ハザードを定義し、各ハザードに対する安全制約とセキュリティ制約を識別する。セキュリティ制約といったセキュリティにかかわる事項を対象とすること以外は、標準的step0準備1と同じである。

Step0準備2（STPA-SafeSec V）：上記制約の実現に必要な、機能コンポーネントとそれらの相互作用（コントロールアクションとフィードバック）を分析して機能CSDを構築する。機能CSDは標準的step0準備2で構築するCSDに対応する。

Step1（STPA-SafeSec VI～IX）：機能CSD内の各コントロールループに対し、トーマス博士が提案する拡張step1［6］により、UCA（原文Hazardous Control Action）を抽出する。拡張step1は、コントローラの入力の組み合わせに対し網羅的にUCAか否かを判定するため、自動化に適しているといった特徴を持つ。なおSTPA-SafeSecの他手順との関連から、STPA-SafeSec step1は標準的step1でも良いと考えられる。他方、STPA-SafeSec step1で用いるガイドワードは標準的STPAの4つのガイドワードと同じである。

Step2はSTPA-SafeSecの特徴的な概念・手順を多く含むため、［4］に従い、以下の3つの手順に分割して解説する。

Step2a（STPA-SafeSec X、XI）：機能CSDの各コントロールループに対し、物理CSDを構築する。物理CSDは機能CSDをアーキテクチャレベルへ詳細化した記述である。このとき、これらの構成要素間を対応付ける。また、抽象的ハザードシナリオ（原文Safety Related Flaws、System Flaws）を策定する。この抽象的ハザードシナリオは、標準的STPA［1］の安全にかかわるHCFヒントを参考に特定される。

抽象レベルでのシナリオとして、ハザードシナリオのみを扱うのは、標準的STPAが扱うシナリオに加え、セキュリティ侵害が安全性を脅かすシナリオを扱うことを目的としているためと考えられる。

Step2b（STPA-SafeSec XII）：step0準備1で識別済みのハザード及びリスト1、2のセキュリティ制約を基に機能CSDのコンポーネントへ抽象的安全・セキュリティ制約を課し、機能CSDと物理CSDの対応に基づき、抽象的安全・セキュリティ制約を物理CSDの要素に割り振る。

物理CSDのコンポーネントに課される安全制約は、step1準備1で識別した安全制約であり、標準的STPAの安全制約である。他方、セキュリティ制約はSTPA-SafeSec step2bで登場する。後の事例において、物理CSDのコンポーネントに既知の脆弱性としてスプーフィング（spoof）とジャミング（jam）が知られている場合に、このコンポーネントへセキュリティ制約（CSTR-A-1、CSTR-A-2）を課すという利用法からは、リスト1、2の内容はセキュリティにかかわるHCFヒントであるとも言える。

リスト1：完全性に対する汎用的脅威
- CSTR-I1 コマンド・インジェクション（Command injection）
- CSTR-I2 コマンド欠落（Command drop）
- CSTR-I3 コマンド操作（Command manipulation）
- CSTR-I4 コマンド遅延（Command delay）
- CSTR-I5 観測値インジェクション（Measurement injection）
- CSTR-I6 観測値欠落（Measurement drop）
- CSTR-I7 観測値操作（Measurement manipulation）
- CSTR-I8 観測値遅延（Measurement delay）

リスト2：可用性に対する汎用的脅威
- CSTR-A1 通信遅延（Communication delay）
- CSTR-A2 通信欠落（Communication dropped）
- CSTR-A3 ノード過負荷（遅延）（Node overloaded（delay））
- CSTR-A4 ノード過負荷（欠落）（Node overloaded（drop））

Step2c（STPA-SafeSec XIII）：step2aで識別した抽象的ハザードシナリオを機能CSDに対するトップレベルのハザードシナリオとし、それを物理CSDへ詳細化していく。このとき、詳細化関係があるため、ハザードシナリオたちは木構造となる。

このように、抽象的ハザードシナリオを具体的ハザードシナリオへ詳細化するアプローチは、［6］3.4 Identifying causal factor scenariosでも紹介されている。しかし［6］では、機能CSDのみを用いて詳細化しているのに対し、STPA-SafeSecでは2つのCSDを用いて詳細化している点が異なる。

3 STPA-SafeSecによる分析事例

本節では、[4]の4、5節にある事例を解説する。文献[4]は事例としてマイクログリッドを用いており、とくに広域電力網と局所電力網の接続(併入)におけるハザード分析を実施している。事例対象の簡単な解説は、本稿のstep0準備2とstep1にある。また詳細な解説は、[4]と[7]を参照いただきたい。

3.1 Step0準備1

STPA-SafeSec Step0準備1では、安全に関する事柄に加えセキュリティに関する事柄を考える以外は、標準的STPAと同じである。この事例では、次のロスを識別している：

- L1：人間への危害
- L2：電力機器の損傷
- L3：ユーザの電器機器の損傷
- L4：停電

続いて、次のハザードを識別している(カッコ内は関連するロスを表す)：

- H1：非同期での系統併入(L1、L2、L3、L4)
- H2：電力機器の運転制限外での運用(L1、L2、L3、L4)
- H3：電力品質指標の逸脱
  - ▶ H3.1 電圧(L1、L3)
  - ▶ H3.2 周波数(L3、L4)
- H4：同期制御の不調(L4)
- H5：地域の電力需要への対応不可(L4)

更に、システムに対する高抽象度の安全制約を、ハザードの否定形を取ることで識別している。このとき、制約は安全制約(CSTR-Sn)、可用性制約(CSTR-An)、完全性制約(CSTR-In)のようにどのような属性に対する制約かを分けて番号付けしている。なおこの事例では、安全制約CSTR-S1からCSTR-S5(H1からH5の否定形)しか登場しないが、一般には可用性制約と完全性制約も扱う。

3.2 Step0準備2

STPA-SafeSecのstep0準備2では機能CSDを構築する。この機能CSDが標準的STPAのCSDに相当する。機能CSDにおけるコンポーネント(原文Node)はNnで、接続はCnで番号付けされる。なおstep2aで、この機能CSDを詳細化した物理CSDを構築し、2つのCSDのコンポーネントを対応付ける。従って、対応が分かりやすい番号付けが望ましい。

この事例では、とくに速度制御器が制御するコントロールループ図に着目し、機能CSD(図1)を構築している。

図1について解説する。速度制御器(N1)、ローカルPMU(N4、ローカルマイクログリッドにある電圧位相計測装置(Phasor Measurement Unit))、ホストPMU(N5)、速度制御器とローカル・ホストPMU間の接続(C4とC5)により接続されている。各PMUからは、電圧(Xm)、周波数(ω)、位相(φ)が周期的に送られてくる。速度制御器は、同期が取れているかを確認し、サーキットブレーカ(N6)へ再開が安全か否かを送信する(この事例では、サーキットブレーカは自動ではなく、操作員が相当する操作を実行すると仮定しているため、開閉命令ではなく、安全か非安全かの情報が送信されている)。また速度制御器は、原動機制御器(N2)を経由して、ジェネレータ(N3)へ運転設定値(set point、C1)を設定する。

3.3 Step1

STPA-SafeSecのstep1では、機能CSDからトーマス博士が提案した拡張step1[6]を用いて、UCAを識別する。ここで、コントローラである速度制御器(N1)が参照する変数は、ΔXm(t)(電

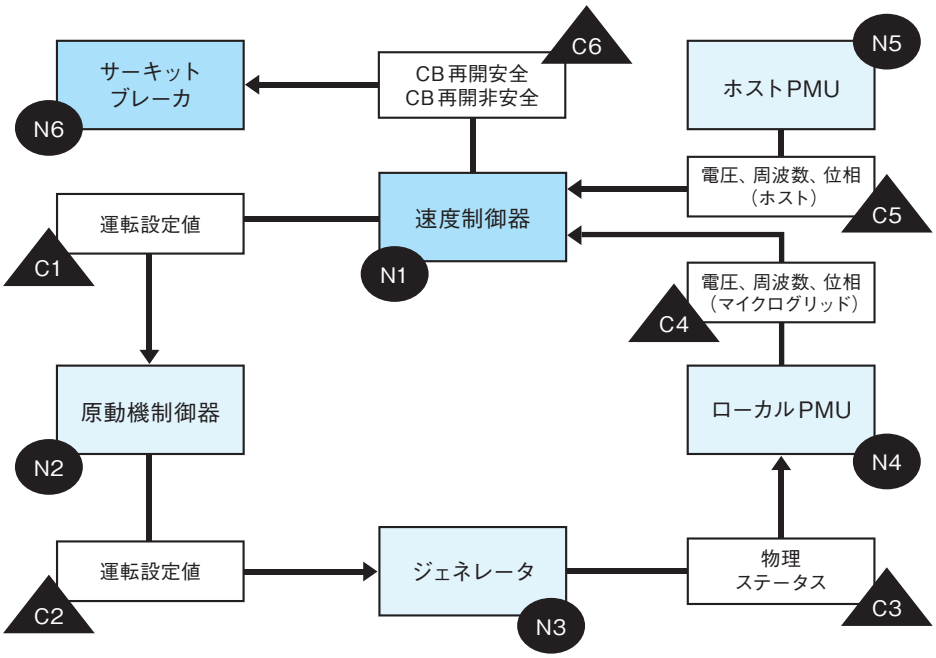

図1　機能コントロールストラクチャー図

圧差：ホストとローカルの電圧差、値は制限内、制限外）、Δω（t）（周波数差：ホストとローカルの周波数差、値は制限内、制限外）、Δφ（t）（位相差：ホストとローカルの位相差、値は制限内、制限外）、St_{cb}（サーキットブレーカの状態、値は開、閉）の４変数であり、速度制御器は、C_{sp}（原動機制御器（N2）への指示、値は運転範囲内、運転範囲外）、C_{cb}（サーキットブレーカ（N6）への連絡、値はCB再閉安全、CB再閉非安全）の２つのコントールアクションを指示する。

STPA-SafeSecは拡張step1を採用しているため、step1分析結果の記述形式が、標準的な記述形式［1］［2］と異なる。この事例のUCA1は、速度制御器のコントールアクションC_{cb}＝CB再開安全とハザードへ至る条件ΔXm（t）＝制限外の組み合わせに対し、ガイドワードProviding（Anytime）、Too early、Too lateのときにハザード（H1、H3）へ至ると記述されている（すなわちUCA1には、Too earlyとToo lateが一つのガイドワードであるとすれば、２つのUCAがまとめられている）。

速度制御器からのコントールアクションに対するstep1の結果は以下の通りである：

- UCA1：ブレーカが解放状態のとき、電圧差が制限外であるにもかかわらず、サーキットブレーカへCB再開安全をProviding,Too early,Too lateで指示（H1、H3）
- UCA2：ブレーカが解放状態のとき、周波数差が制限外であるにもかかわらず、サーキットブレーカへCB再開安全をProviding,Too early,Too lateで指示（H1、H3）
- UCA3：ブレーカが解放状態のとき、位相差が制限外であるにもかかわらず、サーキットブレーカへCB再開安全をProviding,Too early,Too lateで指示（H1、H3）
- UCA4：運転範囲外の設定値を原動機制御器に指示（H2）
- UCA5：ブレーカが解放状態のとき、運転範囲内の設定値を原動機制御器にToo late、Notで指示（つまり、設定値の更新が行われない）（H3、H4、H5）

3.4 Step2a：物理CSDの構築

STPA-SafeSec step2aでははじめに、機能CSDを物理CSDへ詳細化する。物理CSDは機能CSDをアーキテクチャレベルで実現した記述である。図２は図１の機能CSDを基に作成した物理CSDである。

元の事例では、物理CSDの要素（node）はNnの形で、接続はCnの形で表現される。また両CSDの要素間には対応が付けられる。本稿では、機能CSDと物理CSD間の対応の理解性向上のために、機能CSD内のNmと対応する物理CSD内のコンポーネントはNm-nと表記する。なお、機能CSDと物理CSDの要素は多対多対応のため、Nm-nとNm'-n'が同じ要素を表すことがある点には注意が必要である。

機能CSDと物理CSDの対応の一部を示す。N1（速度制御器）は、N1-1（速度制御器CPU）、N1-2（アナログ・デジタル変換器）とN1-3（N1-1とN1-2間のUSB接続）により構成される。またC5（ホスト電圧）は、C5-3（ホスト電圧）、C5-2（ファイアウォール）、C5-1（スイッチ）、C5-4（ホスト電圧、ローカル電圧）により構成される。

機能CSDと物理CSDの対応付け後に、続いて、安全にかかわる抽象的ハザードシナリオ（この事例ではSystem Flaw）を特定している（STPA-SafeSec X）。抽象的ハザードシナリオは、標準的STPA［1］の安全にかかわるHCFヒントを参考に、特定される。この事例では、以下の６つの抽象的ハザードシナリオを特定している。F1：速度制御器は電圧が制限内と誤認識、F2：速度制御器は周波数が制限内と誤認識、F3：速度制御器は位相角が制限内と誤認識、F4：原動機制御器は運転範囲外の設定値を受け

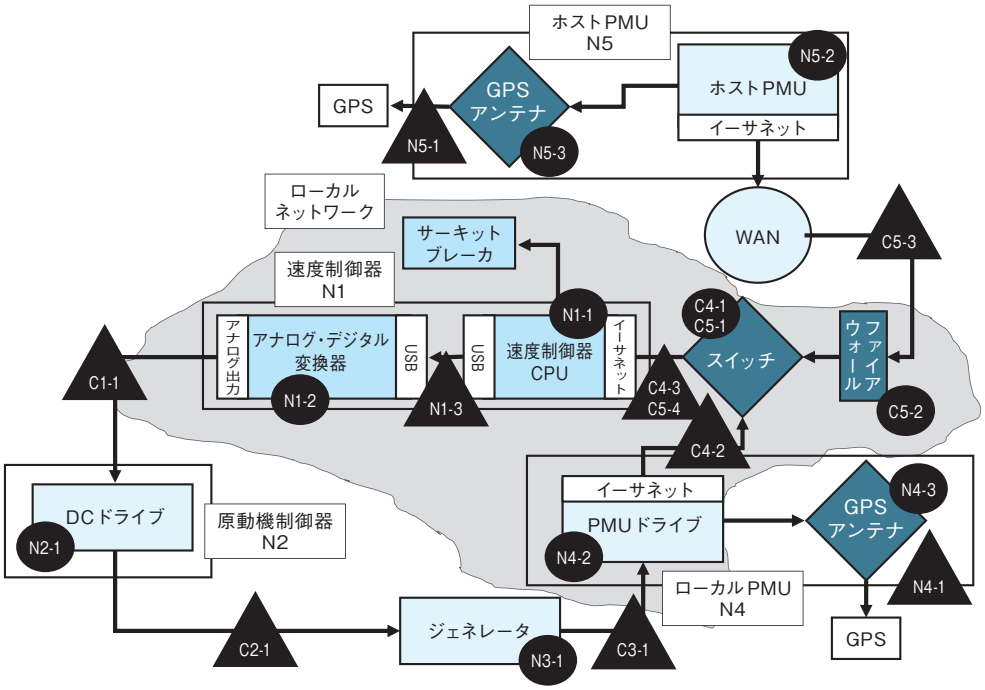

図２　物理コントロールストラクチャー図

取る、F5：速度制御器は設定値変更が要求されていないと誤認識、F6：遮断機制御器は「CB再開安全」という誤情報を受信。

3.5　Step2b：制約の詳細化

標準的STPAでは、この後にハザードシナリオを導出する。他方STPA-SafeSecでは、ハザードシナリオ導出の前に、機能CSDの要素へ安全・セキュリティ制約を課し、機能CSDと物理CSDの対応に基づき、安全・セキュリティ制約を物理CSDの要素に割り振っている。この安全・セキュリティ制約を破る要因がHCFとなる。

物理CSDの要素に対し、安全・セキュリティ制約を課す利点としては、例えば以下の利点が挙げられている。物理CSDのコンポーネントとしてGPSが使用されていることが決まれば、GPSの既知の脆弱性としてスプーフィング（spoof）とジャミング（jam）が知られているため、これらに対するセキュリティ制約（CSTR-A-1、CSTR-A-2）を要素N4-3、N4-1に課す必要があることが分かる。しかし、機能CSDではGPSが使用されるか否かは決定されていないため、これらの制約を課すべきか否かは決定できない。

STPA-SafeSec step2bでは、はじめに、安全・セキュリティ制約と機能CSDの要素を対応付ける。このとき、安全制約として識別済ハザードを用い、セキュリティ制約としてリスト1、2の制約を用いる。正確にはハザードやリスト中記述の否定形が制約である。

次にSTPA-SafeSec step2bでは、step2cで策定するハザードシナリオの理解容易性を高めるため、機能CSDの制約を物理CSDへ詳細化する。この事例では、幾つかのデバイスに対して制約の詳細化が示されているが、本稿では速度制御器（N1）に対する制約の詳細化のみを紹介する。

はじめに、ハザードより安全制約を導出し、リスト1、2よりセキュリティ制約を導出する。速度制御器に対しては、H1（非同期での系統併入）、H2（電力機器の運転制限外での運用）、H3（電力品質指標の逸脱）とH5（地域の電力需要への対応不可）が課される。次に、これらの安全制約とセキュリティ制約を、速度制御器の構成要素である、速度制御器CPU N1-1、アナログ・デジタル変換器N1-2とUSB接続N1-3に割り当てる。ここでは、H1、H3とH5は速度制御器CPUに、H2はアナログ・デジタル変換機に割り当てられる。

3.6　Step2c ハザードシナリオ策定

はじめに、トップレベルのハザードシナリオ（シナリオ1）を策定する。トップレベルのハザードシナリオは、step2aで識別した抽象的ハザードシナリオであり、それに関連するUCA（Hazardous Control Action）と機能CSDのコンポーネント及び物理CSDのコンポーネントから構成される。次に、トップレベルのハザードシナリオに含まれる各層の要素に着目し、ハザードシナリオを詳細化していく。このとき、詳細化されたハザードシナリオのHCFに対応する制約を合わせて記述する。下位シナリオ（シナリオ1.1以降）は、機能CSDのコンポーネント、物理CSDのコンポーネントに加え、安全制約、セキュリティ制約から構成される。

トップレベルのハザードシナリオから詳細化して策定されたハザードシナリオたちは、木構造を成す。木構造の上位ノードはより抽象的ハザードシナリオが対応する。すなわち、あるノードの子ノードには当該ノードに割り当てられたシナリオのサブシナリオが付される。このような木構造にすることで、あるハザードシナリオへの対応策は、木構造におけるそのノードの下位ノードの対応策になる。

本稿では、一部のハザードシナリオのみ紹介する。

シナリオ1：速度制御器は、（ローカル・ホスト間の）電圧差が制限値内と誤認識する。「ハザード：H1（非同期での系統併入）、H3（電力品質指標の逸脱）、抽象的ハザードシナリオ：F1（電圧差が制限内と誤認識）、UCA：UCA1（ブレーカが解放状態のとき、電圧差が制限外であるにもかかわらず、サーキットブレーカへCB再開安全をProviding,Too early,Too lateで指示（H1、H3））、機能CSD関連コンポーネント：N1、N4、C4、N5、C5、物理CSD関連コンポーネント：N1-1、N4-2、C4-2、C5-1、C5-2、N5-2、C5-3、C5-4）」

シナリオ1.1：速度制御器（N1）は、正しいフィードバックを間違って認識する。「機能CSD関連コンポーネント：N1（速度制御器）、物理CSD関連コンポーネント：N1-1（速度制御器CPU）、安全制約：デバイス（速度制御器CPU）とアルゴリズムの信頼性、アルゴリズムの正しさ、セキュリティ制約：CSTR-I5 Measurement injection（フィードバック（以下FB）信号へのインジェクション攻撃）、CSTR-I7 Measurement manipulation（FB信号の操作）」

シナリオ1.2：速度制御器は、ホストPMUからの間違った信号を受け取るが、それを正しいと認識する。「機能CSD関連コンポーネント：N1、N5、C5、物理CSD関連コンポーネント：N1-1、C5-1、C5-2、N5-2、C5-3、C5-4（注意：C4-3と同じ対象を指す）、安全制約：N5の信頼性、セキュリティ制約：CSTR-I5（FB信号へのインジェクション攻撃）、CSTR-I7（FB信号の不正操作）」

シナリオ1.2.1：ホストPMUが間違ったFB信号を送る。「機能CSD関連コンポーネント：N5、物理CSD関連コンポーネント：N5-2、安全制約：N5-2の信頼性、セキュリティ制約：CSTR-I5（FB信号へのインジェクション攻撃）、CSTR-I7（FB信号の不正操作）、N5-2への脆弱性攻撃成功（Successful exploit）」

シナリオ1.2.2：リモートPMUからの正しいFB信号が、ホスト電圧（C5）で改ざんされる、またはインジェクション攻撃される。N1-3の通信は正常であるとする。「機能CSD関連コンポーネント：C5、物理CSD関連コンポーネント：C5-3、N5-1、N5-2、安全制約：なし、セキュリティ制約：CSTR-I5（FB信号へのインジェクション攻撃）、CSTR-I7（FB信号の不正操作）」

シナリオ1.2.3：リモートPMUからの正しいFB信号が、ホスト電圧（C5）で改ざんされる、またはインジェクション攻撃される。N1-3の通信は異常だが受け入れられるとする。「機能CSD関連コンポーネント：N1、C5、物理CSD関連コンポーネント：N1-1、C5-3、N5-1、N5-2、安全制約：なし、セキュリティ制約：CSTR-I5（FB信号へのインジェクション攻撃）、CSTR-I7（FB信号の不正操作）」

4　今後の課題

本節では、STPAをベースに脆弱性分析を行う際の課題として、

step2で用いるHCF導出のヒントに関する課題を述べる。またSTPAでは分析時に妥当な仮定を置かずに分析を実施すると、分析対象が肥大化したり、分析者により分析結果が大きく異なったりといった状況に陥りがちである。そこでSTPA一般の課題として、分析時の仮定について述べる。

4.1 セキュリティにかかわるHCFヒントに関する課題

STPA-SafeSecでは、標準的STPA step2で利用されるHCFのヒントに加え、セキュリティにかかわるHCFを導出するために、リスト1、2にあるヒントを利用する。他方、STAMP WorkbenchやSafetyHATといったSTAMP/STPA支援ツールでは、HCFヒントを分析対象領域に依存して適切に変更でき、更に分析者が独自に編集できる。例えば、[8]にあるHCFヒントは機械のコントローラを想定しており、機械のコントローラに対しては適切なヒントであるが、人間のコントローラに対しては異なるヒントのほうがHCFを導出しやすいであろう。従って、セキュリティにかかわるHCFヒントも、適宜修正・変更することで、HCFを導出しやすくなると考えられる。

STPA-SafeSecで採用されているセキュリティにかかわるHCFヒント以外にも、例えば、セキュリティにかかわるヒントとしてSTRIDE[5]の利用が考えられる。STRIDEは、Spoofing(スプーフィング)、Tampering(改ざん)、Repudiation(否認)、Information Disclosure(情報漏えい)、Denial of Service(サービス拒否)、Elevation of Privilege(特権の昇格)の頭文字から成り、それぞれは代表的な脅威、すなわちセキュリティにかかわるHCFヒントを表す。

4.2 分析時の仮定に関する課題

STAMP/STPAで解析を行うときに一般に難しい点は、どの抽象度とどの仮定のもとでコントールストラクチャやHCFの設定を行うかであろう。モデル化を行う際にはある程度のドメイン知識を暗黙裏に仮定する。この仮定の妥当性は解析とモデル化を繰り返すことにより補強するのが現在の標準的な手順である。この際にknown-unknownsやunknown-knownsなどの仮定の境界上の事項[9]を意識することが強く望まれる。

STPA-SafeSecでは物理CSDの導入によりunknown-knownsの気づきに貢献している。例えば、「物理CSDのコンポーネントとしてGPSが使用されていることが決まれば、GPSの既知の脆弱性としてスプーフィング(spoof)とジャミング(jam)が知られている」というのは「GPSの既知の脆弱性としてスプーフィング(spoof)とジャミング(jam)が知られている」というドメイン知識を解析者のunknown-knownsからknown-knowsへの変換に寄与しているとみなすことができる。

一方、known-unknownsについては次のような対策が取れる。known-unknownsについては典型的には定性的要因が分かっているが、定量的な値が不明であるという特徴を持つことが多い。その場合は値に関する変数を不定値とみなしたり、あるいは統計的量として捉えることによりモデル化できることがある。その場合はそれぞれに適した数理モデルや解析手法の活用が可能となる。

5 まとめ

本稿では、STAMP海外事例として、STPA-SafeSecを紹介した。STPA-SafeSecは、安全性とセキュリティを統合して分析するためのSTPA拡張であり、機能CSDと物理CSDを持ち、STPA step2で使用するHCFヒントをセキュリティ拡張したという特徴を持つ。併せて紹介したSTPA-SafeSecの適用事例はSTPA-SafeSecの有用性を示している。しかし、セキュリティにかかわるHCFヒントはリスト1、2のヒント以外にもSTRIDEを活用するといったことも考えられる。また、既存のセキュリティ分析手法とSTPA-SafeSecを統合し、分析結果を充実させるといった点にも改善余地はあると考えられる。

【参考文献】

[1] LevesonG.Nancy. (2011). Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press.

[2] システム安全性解析WG. (2016). はじめてのSTAMP/STPA. 情報処理推進機構.

[3] YoungWilliam,LevesonNancy. (2013). Systems Thinking for Safety and Security. In Proceedings of the 29th Annual Computer.

[4] Ivo FriedbergMcLaughlin,Paul Smith,David Laverty,Sakir SezerKieran. (2017). STPA-SafeSec: Safety and security analysis for cyber-physical systems. Journal of Information Security and Applications.

[5] マイクロソフト. (日付不明). モノのインターネットのセキュリティ アーキテクチャ.
参照先 :https://docs.microsoft.com/ja-jp/azure/iot-hub/iot-hub-security-architecture

[6] ThomasJohn. (2013). EXTENDING AND AUTOMATING A SYSTEMS-THEORETIC HAZARD ANALYSIS FOR REQUIREMENTS GENERATION AND ANALYSIS. Ph.D Thesis,MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

[7] IvoFriedberg,DavidLaverty,KieranMacLaughlin,PaulSmith. (2015). A Cyber-Physical Security Analysis of Synchronous-Islanded Microgrid Operation. Proceedings of 3rd International Symposium for ICS & SCADA Cyber Security Research.

[8] LevesonG.Nancy, ThomasJohn. (2013). An STPA Primer.

[9] Sebastian ElbaumS. RosenblumDavid. (2014). Known unknowns: testing in the presence of uncertainty. Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering.