Vol.2018-DPS-174 No.6
Vol.2018-CSEC-80 No.6
2018/3/5

安全性解析手法 STAMP／STPA への脅威分析（＝STRIDE）の適用

金子朋子[†1] 早川拓郎[†3] 髙橋雄志[†3] 大久保隆夫[†2] 佐々木良一[†3]

概要： IoT 時代の複雑なシステムに対する新たな安全性解析手法として注目を集めている理論とその安全分析手法に STAMP （System Theoretic Accident Model and Processes）と STPA （System Theoretic Process Analysis） がある．STAMP/STPA はセーフティを中心に展開されてきたが，セキュリティ上のリスク分析にも適用可能である．STPA のセキュリティ対応手法である STPA-Sec の他に STPA-SafeSec も提案されている．本稿では STPA-Sec や STPA-SafeSec に示されていないサイバーセキュリティの視点からの脅威分析の必要性を提示する．具体的には STPA-SafeSec 論文のスマートグリッドの事例に対して STRIDE モデルを適用し，その効果を考察する．

キーワード：STAMP／STPA，脅威分析，STRIDE，セーフティ・セキュリティ，セキュリティ・バイ・デザイン

Proposing enhancement of a threat analysis from the security perspective for STAMP／STPA as a safety analysis

KANEKO TOMOKO[†1] TAKUO HAYAKAWA[†3] TAKAHASHI YUJI[†3] TAKAO OKUBO[†2] RYOICHI SASAKI [†3]

Abstract:
STAMP (System Theoretic Accident Model and Processes) and its safety analysis application, STPA (System Theoretic Process Analysis) have attracted much attention as a new safety analysis method for complex systems of IoT. Though, STAMP/STPA is disseminated as a safety analysis technique, they also can be applied in the security risk analysis, and the security response of STPA is proposed as a STPA-Sec, or STPA-SafeSec.
This paper presents the need for threat analysis from the viewpoint of cyber-security in stpa-sec and stpa-safesec. Specifically, the STRIDE model is applied to the case of the smart grid of the safesec paper, and the effect is examined.

Keywords: STAMP／STPA, Threat Analysis, STRIDE, Threat Tree, Safety, Security by Design

1. †はじめに

現代のシステムはネットワークを介して様々な機器やクラウドと連携しながら動作する新たなサービスが拡大している．このように異なる分野の製品や産業機械などがつながって新しいサービスを創造するモノのインターネット（IoT: Internet of Things）は新産業革命とまで言われ，大きな期待を集めている．IoT は家電，自動車，各種インフラ業者など新規プレーヤーの登場を産み，その取り込みは加速化している．しかし相互につながる際に最も懸念されるのは，IoT システムへのセキュリティ上の脅威である．IoT システムにおいても攻撃者はシステムの脆弱性を突いて攻撃を仕掛けてくるためである．

IoT 時代には相互につながるシステムへの脅威に対して，より安全な機器，システムを開発することが必要とされる．そのためには，従来の情報セキュリティ上の機密性と完全性と可用性に加え，安全性の視点が必要になる．そこで筆者らは安全の視点でリスク分析を行うために IoT 時代の複

1 情報処理推進機構 INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN(IPA)

†2 情報セキュリティ大学院大学 INSTITUTE of INFORMATION SECURITY

†3 東京電機大学 TOKYO DENKI UNIVERSITY

Vol.2018-DPS-174 No.6
Vol.2018-CSEC-80 No.6
2018/3/5

雑なシステムに対する安全解析手法 STAMP （System Theoretic Accident Model and Processes） [1][2]とそのハザード分析手法である STPA（System Theoretic Process Analysis）[3][4]に着目し，安全を考慮した新しいセキュリティ分析手法を提案する．STAMP／STPA は安全（セーフティ）を中心に展開されてきたが，セキュリティ上のリスク分析にも適用可能であり[4]，STPA のセキュリティ対応手法である STPA-Sec も提案されている[5][6]．さらに安全性と脆弱性を統合して分析できる STPA の拡張である STPA- SafeSec Sec も提案されている[7][8].

本論文は 2 章で STAMP と各種ハザード手法，セキュリティ要求分析に関連した技術と考え方を紹介する．続く 3 章で STPA のセキュリティ対応手法である STPA-Sec と STPA-SafeSec の説明とその課題を示す．4 章では STPA-SafeSec のスマートグリッドの事例に対して STRIDE 分析を適用した内容を述べる．5 章で提案方式に関する考察を行い，6 章でまとめと今後の方針について述べる．

2. 関連研究

2.1 STAMP と関連手法

STAMP とはシステム理論に基づく事故モデルであり，STPA は STAMP モデルにもとづく代表的な手法として，ハザード分析を行うものである．

前提として，システム事故の多くは，構成要素の故障ではなく，システムの中で安全のための制御を行う要素（制御要素と被制御要素）の相互作用が働かない事によって起きるとし，「要素（コンポーネント）」と「相互作用（コントロールアクション）」に着目してメカニズムを説明し，「アクションが働かない原因」が「コントロールアクションの不適切な作用」に等しいという視点を持つことで原因を有限化している．

STAMP に基づく分析の道具立てプロセスとして，仕様記述，安全性ガイド設計，設計原理などのシステム工学，リスク管理の運用，管理の原則／組織設計の規制を利用する（図 1）．

ツール（手法）には STAMP モデルに基づき，事故／イベント分析（CAST: Causal Analysis based on STAMP），ハザード分析（STPA），早期概念分析（STECA: Systems-Theoretic Early Concept Analysis），組織的／文化的リスク分析，先行指標識別，セキュリティ分析（STPA-Sec）が提示されている．事故／イベント分析（CAST）は事故が起きてからイベントとして分析する手法，STPA-SEC はそのセキュリティ版である（図 1）．セーフティとセキュリティを統合する手法としては STPA-SafeSec が提案されている[7][8].

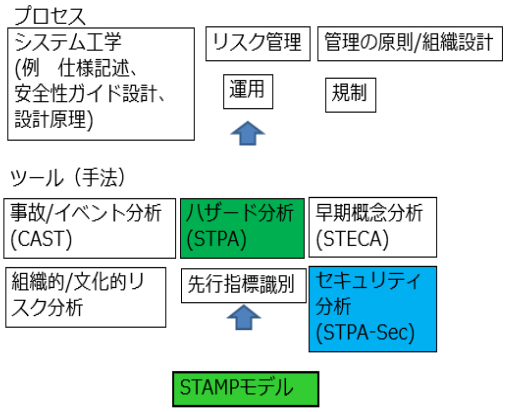


図 1.　STAMP に基づく分析の道具立てとプロセス

2.2 ハザード分析手法

FTA（Fault Tree Analysis），FMEA（Failure Mode and Effect Analysis）は，フォールトツリー図や影響分析表を用いてハザード要因を分析する伝統的なハザード分析手法である．システムの構成要素と故障モードが決まるアーキテクチャ設計の段階から適用できる．機器や組織の単一故障をハザード要因として識別する分岐条件を論理的に組むことで網羅的に分析できる特徴をもつが，深く分析できる反面，構成要素間の相互作用から発生するアクシデントといった全体的な視野を必要とする分析が難しい．

STPA は STMAP モデルに基づき，安全制約の実現に関係するコンポ―ネントとその相互作用を制御構造図にしたコントロールストラクチャーとコントロールループ図を用いてハザード要因を分析する安全解析手法である．システムの大まかな構成要素が決まる概念設計の段階から適用できる．複数の機器や組織（人間）が，相互作用を行う複雑なシステムにおいて，相互作用に潜むハザード要因を識別する特徴をもち，過去のアクシデント事例データに基づくガイドワードにより分析する．またシステム全体の振る舞いを確認しながら分析できる．

2.3 セキュリティ・バイ・デザイン

内閣府サイバーセキュリティセンター （NISC）によると「セキュリティ・バイ・デザイン」とは「情報セキュリティを企画・設計段階から確保するための方策」[9]であり，「安全な IoT システムのためのセキュリティに関する一般的枠組」[10]においては，目的としてまた基本原則として掲げられている重要な概念である．IoT 時代を迎え，セキュリティ上の脅威が多大な被害を及ぼす可能性がでてきているため，企画・要件定義工程や設計工程というより早い段階から事前にセキュリティを作りこむことが求められているのである（図 2）．

Vol.2018-DPS-174 No.6
Vol.2018-CSEC-80 No.6
2018/3/5

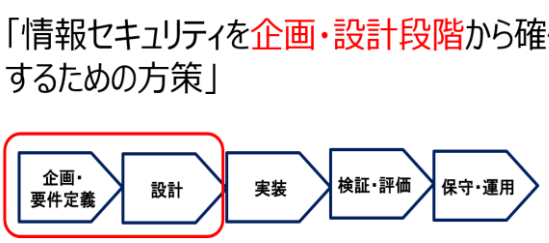


図 2.　セキュリティ・バイ・デザインの定義

2.4　セキュリティ分析手法

セキュリティ分析手法にはアタックツリー[11][12]，ミスユースケース[13]，マイクロソフトのセキュリティ開発ライフサイクル[14]，STRIDE 分析を含む脅威モデリング[15]などがある．セキュリティ開発ライフサイクル[14]はデータフロー図を詳細化し脅威の観点 STRIDE で脅威分析を実施する．設計による安全性確保を重視し設計段階でセキュリティ要求を抽出している．また STRIDE を元にした Threat Tree 分類[15]も示されている．ただし，安全解析における FTA，FMEA 等のように歴史があり，標準化もなされたセキュリティ分析手法は存在せず，開発現場でも普及した設計手法は定まっていないのが現状である．

3.　STPA のセキュリティ対応

3.1　STPA-Sec とその課題

STAMP／STPA は安全性解析手法であり，セーフティを中心に展開されてきたが，これらの特徴はセキュリティ上のリスク分析にも適用可能である．そこで 2016 年 3 月 STAMP workshop でのチュートリアル文献[6]に基づいて，サイバーセキュリティなどに特化した事故分析手法である STPA-Sec について説明する．

①　既存のアプローチと比較した特徴

STPA は全体を俯瞰してトップダウンに分析をする手法である．STPA-Sec も同様に全体俯瞰の上で，トップダウンに分析を実施する．STPA-Sec は従来のセキュリティエンジニアリングがフォーカスしてきた物理的な機能（図 4 の青の部分）に比べ，より広範囲で概念的な機能・目的にフォーカス（図 3 の緑の部分）している．

また，STPA は手段（How）に着目した手法ではなく何（What）に着目した手法である．STPA-Sec も同様に従来のセキュリティ要求分析手法であるアタックツリーやミスユースケースのようにどのような脅威があるのかを洗い出す How ではなく，コンセプト段階での問題 What であるかを明確にするアプローチである．

セキュリティ対策は現状，保守・運用段階での脆弱性対処が中心である．しかし，多様な機器・システムが複雑につながる場合には何がセキュリティを確保する際の問題となるのかを事前に対処できることがより重要になってきている．そのため STPA-Sec のアプローチは現在のセキュリティー開発の課題解決に役立つであろう．

②　STPA と STPA-Sec との違い

STPA と STPA-Sec の分析手順は基本的な部分は同じである．ただし，セキュリティ上の脅威抽出に必要な分析の視点が追加される．要因の特定に関して図 4 に示すオレンジの部分が STPA-Sec での追加事項となる．コントローラーなどに悪意ある，権限を持たない，部分的なインプットを要因として追加的に分析される．

③　分析手順

STPA-Sec の分析手順は安全ではない状態を考えるのと同様に，セキュアではない状態を考慮する．また，非安全なコントロールの原因の特定に際し，セキュアではないコントロールアクションを導くシナリオを識別し，影響度を踏まえ，よりクリティカルなコントロール戦略を選択することなどが STPA に対する主な追加事項である．

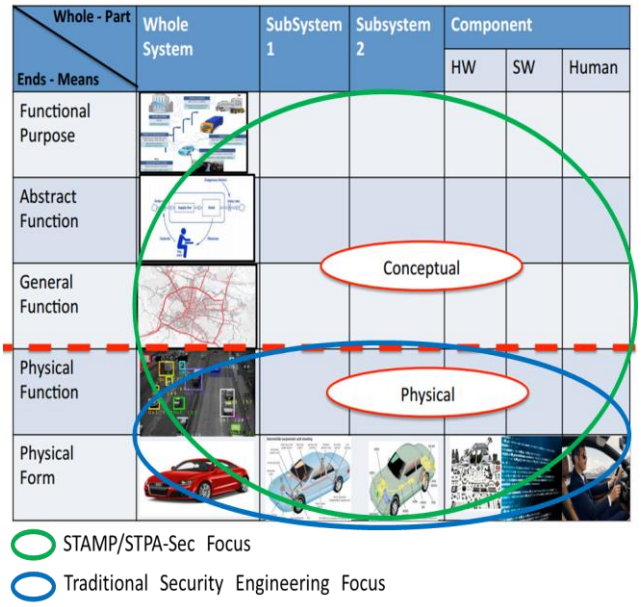


図 3．STPA-Sec のフォーカスエリア

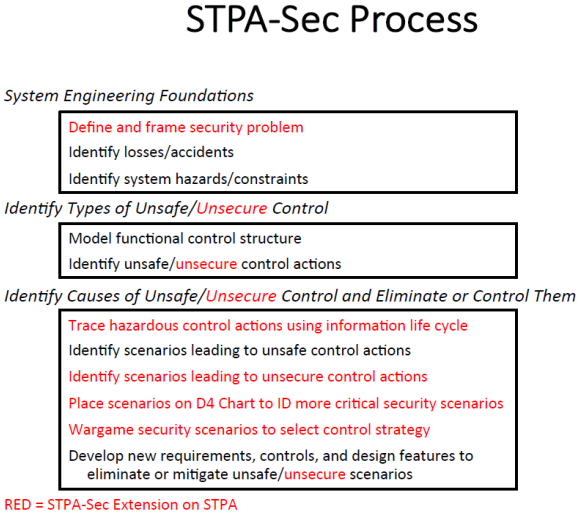


図 4．STPA-Sec の拡張部分

現在の STPA-Sec はコンセプチュアルなミッション・ビジネスレベルに重きをおき，従来のセキュリティエンジニアリングが重視してきた部分に対してはどのように実施するのかについて，不明確である．

現在の STPA-Sec はミッション・ビジネス運用とシステム脆弱性に焦点を当てて，ハザード分析を行うものが公開されているだけで（図 3），システマチックな脅威分析を実施する手順や事例は公開されていない．MIT の事例[6]をみても非安全なコントロールアクションに対して，セキュリティ制約をどうやって，導き出すかの手法が提示されていない．また，STPA 分析手順の Step2（図 4）に対応する段階では，要因のシナリオを導き出す際にセキュリティ要因の必要十分性を説明できないのが現状である．セキュリティ要因の洗い出しは，ヒントワードに対して青字で記載した事項の追加により対処することになっているが，これらのヒントワードはハザード要因分析のヒントワードに部分的，悪い形状の情報オペレーションを追加しただけである．なぜ，セキュリティ要因の洗い出しのこれらの追加がなされたのかの説明がなされていない．本来，セキュリティ誘発要因（SCF）には，悪意ある者の攻撃にもとづいた脅威分析が必要であるとであると筆者らは考えた．

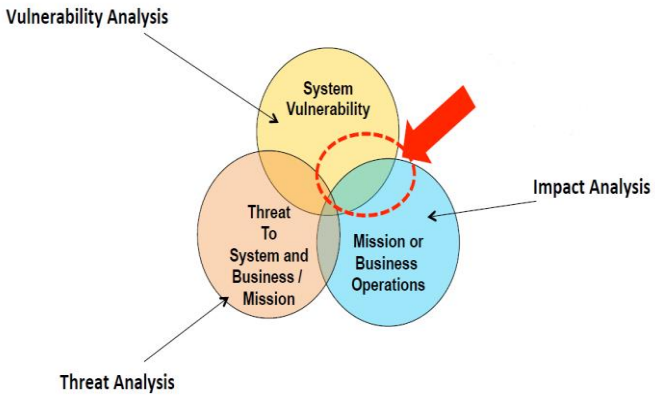


図 5. 現在の STPA-Sec の焦点

3.2 STPA-SafeSec とその課題

STPA-SafeSec[7][8]は安全性と脆弱性を統合して分析できる STPA の拡張である．

STPA-SafeSec では，①機能 CSD（Control Structure Diagram）と物理 CSD の二つの CSD を利用，②Step2 で使用する HCF ヒントのセキュリティを拡張という特徴とその派生効果として③安全性・セキュリティ対策を統合できるとされている[8]．①に関しては，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 CDS 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．さらに，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とされている．

しかしながら，セキュリティに係る脆弱性分析は下位層にあたる物理 CSD だけではなく，上位層である機能 CSD でも可能ではないかという疑問が生じる．STPA-Sec では SoS（System of Systems）の考え方に従い，機能レイヤと物理レイヤの手順を変えることなく，どのレイヤにおいても同様な手順でセーフティと共にセキュリティの分析を行うのに対し，STPA-SafeSec では物理層のみでセキュリティ分析を行うとしている点が大きく異なる．これに対して，双方の主張に従ったときにどのような結果をもたらすのかを具体的な事例を通じて検証することが適切だと思われる．

②に関しては，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 HCF ヒントとして，安全性に係るヒントとして，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，要求仕様不具合等を用いることが一般的である．STPA-SafeSec では，物理 CSD のコンポーネントに既知の脆弱性としてなりすまし（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）を課すという利用をし，これらの従来のヒントにセキュリティに係る HCF ヒントを追加し，HCF としてなりすまし等のセキュリティに係る誘発要因を特定できるようにしている．

なお，STPA-SafeSec では採用されているセキュリティに係る HCF ヒントとは，物理 CSD においてのセキュリティ上の既知の脆弱性を利用するものである．個別の機器，ソフトウェアに対するセキュリティ上の脆弱性は数も多く，絶えず変化するものである．また，どのような攻撃に対してその脆弱性が脅威になるのかを明確化しないと，設計への指針レベルでの対策がうまれづらい．攻撃による脅威を明確化した上での網羅的な要因分析になりづらく，セキュリティに係るヒントとして STRIDE の利用が考えられる．

4. STPA-Sec の脅威分析拡張事例

4.1 STPA-Sec と STPA-SafeSec の課題解決の方向性

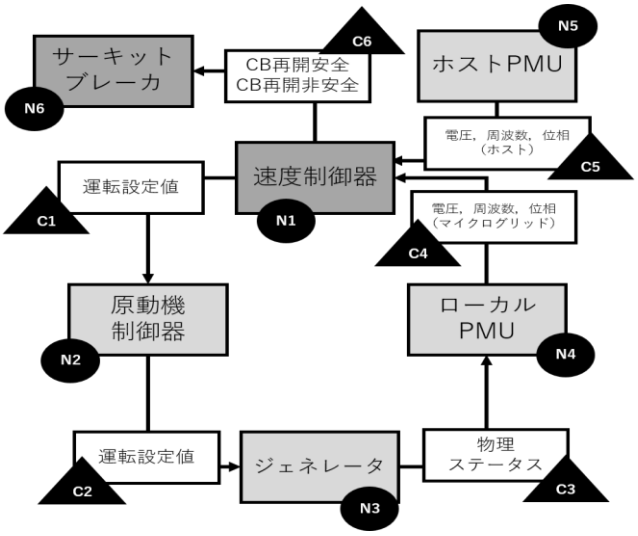
STPA-Sec と STPA-SafeSec の課題解決には，両者が言及していない脅威分析を追加することがどのような価値を生むのかについて示すことが重要である．本稿では脅威分析に有効性について事例を通じて検討するため，STPA-SafeSec

の簡単な解説をしたうえで，脅威分析を適用してみる．

4.2 STPA-SafeSec 論文のマイクログリッド事例

文献 [7]は事例としてマイクログリッドを用いており，特に広域電力網と局所電力網の接続（併入）における分析を実施している．

事例の内容にそって,セキュリティに係る手順を考察する．Step 0 準備 1（STPA-SafeSec II～IV）では，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．システムに対する高抽象度の安全・セキュリティ制約を，ハザードの否定形を取ることで識別しており，制約は安全制約（Safety, CSTR-Sn），可用性制約（Availability, CSTR-An），完全性制約（Integrity, 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[8]. Step0 準備 2（STPA-SafeSec V） で構築される CSD は機能レイヤレベルの CSD である．



Step1（STPA-SafeSec VI～IX）UCA を抽出する．

Step2a（STPA-SafeSec X, XI）で物理層 CSD を構築する．機能レイヤレベルでの速度制御器が物理層レベルではアナログ・デジタル変換器と Rasberry Pi，USB などの具体的な構成で表現されている．

（STPA-SafeSec XII）で抽象的安全・セキュリティ制約を物理レイヤ CSD の要素に割り振る．

Step2c（STPA-SafeSec XIII）抽象的ハザードシナリオを物理レイヤ CSD へ詳細化する．

3.2 項で記述したように STPA-SafeSec 論文では機能 CSD と物理 CSD に分けて手順を考えている．STRIDE を機能 CSD に適用した場合の具体的事例を 4.3 節，4.4 節に示す．

4.3 STRIDE 分析によるセキュリティ事例の具体化

本事例で示されたセキュリティ制約は CSTR-I-5（フィードバック信号へのインジェクション攻撃），CSTR-I-7（FB 信号の不正操作）の 2 種類に限られており，抽象的である．また，セキュリティ由来の HCF になりそうな例として脆弱性が挙げられているのみである．これらが実際に HCF となるか否かの判定のためには攻撃者の視点で分析をするセキュリティ分析手法の活用が期待される．

そこで本稿では参考文献[7]ではセキュリティ分析はなされていない機能レイヤの CSD の 1 つである速度制御器に対して，ハザードシナリオ導出に攻撃者の視点に基づく脅威分析手法である STRIDE 分析を組み合わせて実施してみた．脅威モデリングにおいて STRIDE はリファレンス アーキテクチャを基に図を作成し，システムの全体像を把握したのちに，脅威の列挙，軽減，軽減策の検証などに用いられる．なお，脅威分析で作成される図として，CSD を用いることは可能であろう．脅威モデル化の目的は攻撃者がどのような方法でシステムに侵入しうるかを把握したうえで、適切な軽減策を講じることであり，システムのデプロイ後ではなく設計段階で軽減策を考慮することはコスト上の無駄を省くことにつながり，重要である．

表 1.N1-1 のセキュリティ誘発要因

STRIDE 属性	必要な属性	N1-1 のセキュリティ誘発要因（SCF）
Spoofing identity なりすまし	Authentication 認証	N1-1（速度制御器 CPU） への正しい認証がなされない（N1-1-S）
Tampering 改ざん	Integrity 完全性	N1-1（速度制御器 CPU） に誤った FB 信号が挿入される（N1-1-T）
Information Disclosure 情報漏えい	Confidentiality 機密性	N1-1（速度制御器 CPU） の FB 信号が漏えいする（N1-1-I）
Denial of Service サービス不能	Availability 可用性	N1-1（速度制御器 CPU） が破壊される
Elevation of Privilege 権限昇格	Authorization 認可	正当な権限をもたないものに権限が奪われ、N1-1（速度制御器 CPU） が正しい動作をしない

前述の STIRDE の 6 つの分類はそれぞれ表 1. のように認証，完全性，機密性，可用性，認可というセキュリティ上の必要な属性を意味し，異なる観点で列挙した脅威は属性ごとに軽減策の方向性が示される．さらに STRIDE にリンクした脅威メカニズムを示した分類 Threat Tree[16]を利用することで，STPA で攻撃の起こりうる個所に対して洗い出した脅威ごとに，開発と運用による脅威の低減策が得られる．STRIDE は従来，主に情報システムの脅威モデリングに用いられてきたが，最近本事例のような特殊な用途をもって接続されるデバイスを含め IoT セキュリティへの適用も解説がなされている[**17**].

そこでこの解説を参考にしつつ SafeSec の事例のなかの「シナリオ 1.1：CPT-N1 速度制御器は，正しいフィードバックを間違って認識する.」に対して，そのセキュリティ誘発要因を STRIDE で分析する．STRIDE をヒントワードとして抽出したこの事例の速度制御器（CPU）に対するセキュリティ誘発要因を表 2 に示す.

表 2. 速度制御器（機能レイヤ）の脅威シナリオと対応策例

SCF	想定される脅威シナリオ	対応策例
N1-S	ホストＰＭＵがローカル PMU になりすます	認証機能をもつ IC チップ等を利用する
N1-T	・速度制御器上で実行されているソフトウェアの一部または全部が，攻撃者によって差し替えられる	メッセー認証コード（MAC），改ざん防止機構を速度制御器に適用
N1-I	・速度制御器上で実行されるソフトウェアが改変されていた場合，その改変されたソフトウェアから，許可していない相手に平文が漏えいするおそれがある。	マルウェア対策を実施， 安全なキー管理
N1-D	・速度制御器は，受信接続や一方的に送りつけられるデータグラムをネットワーク上で常時待機する形で DoS の脅威にさらされる. ・攻撃者は，多数の接続を同時に開き，何も処理を行わないか，処理に極端に時間をかけることもある．一方的なトラフィックで速度制御器の処理能力をパンクさせる場合もある． どちらの場合も速度制御器は事実上，ネットワークで機能不全の状態となる. ・妨害電波やケーブルの切断によって，速度制御器の機能が停止したり，通信できない状態になる	攻撃元や同じ IP からのアクセス回数を制限する. 大規模トラフィックに耐えうる速度制御器にする
N1-E	・上位権限と下位権限に分かれて運用している場合，権限昇格され速度制御器が，他の目的に利用される可能性がある	速度制御器のアクセス制御を行う． 承認スキームを設ける.

なお，Repudiation（否認）はユーザーがあるアクションを行ったこと否認し相手はこのアクションを証明する方法がないことであるため，N1-1 の場合は該当要因がない.

さらに，IoT セキュリティの STRIDE 分析[17]を参考にして想定した N1-1 のセキュリティ誘発要因（SCF）ごとに具体的に想定される脅威のシナリオと対応策例を表 2 に示す. この分析で攻撃者の視点でどのような攻撃が対象とするデバイスにおいて可能であるのが具体化された．また各脅威はセキュリティ属性ごとに分類されているため，必要なセキュリティ対策に導きやすくなっている． N1-1 というデバイスに対する分析以外にも，通信，ストレージなど対象に応じて脅威のシナリオと対策を導くことができる.
SafeSec ではセキュリティ上の脅威になりうる脆弱性を洗い出してはいるが攻撃者視点での脅威分析がなされていない．また，可用性と完全性のみをセキュリティ属性として分析しているが，アクセス制御，暗号化につながる機密性や認証，認可につながる真正性も分析対象とすべきだと考えられる．これらは STRIDE 分析を組み合わせることで可能となる.

機能レイヤのコンポーネントの 1 つである速度制御器に対して，表 2 にように STRIDE 分析した結果，属性ごとに想定される具体的な脅威のシナリオと対応策を出すことができた.

4.4 STRIDE の適用による物理レイヤ CSD への影響

機能レイヤの速度制御器に STIRDE 適用をしたのちの物理レイヤ CSD の選択にどのような影響がありうるかを考察する．物理レイヤのコンポーネントの 1 つである RaspberryPi に対して，表 3 にように STRIDE 分析した結果，属性ごとに想定される具体的な脅威のシナリオと対応策を出すことができた.

5. 考察

4 章の STRIDE 分析を追加した事例からわかったことは以下の通りである.

機能レイヤと物理レイヤの双方で STRIDE 分析が可能である．機能レイヤと物理レイヤでは抽象度が異なるが，それぞれで STRIDE による脅威分析は可能であり，階層化されているため，機能レイヤでリスクを洗い出し，セキュリティ機能を作りこみなどの対策ができる.さらにその結果，選定された物理レイヤの各コンポーネントで更に詳細化されたコンポーネントに対して脅威分析を実施することができる．これは要求分析，設計の各段階に適したセキュリティの作りこみをすることになる．この階層化されたセキュリティの作りこみとはセキュリティ・バイ・デザインそのものである.

表 3. RaspberryPi（物理レイヤ）の脅威シナリオと対応策例

SCF	想定される脅威シナリオ	対応策例
N1-1-S	・OS のユーザ設定が適切に行われていない場合、攻撃者がなりすますおそれがある。	パスワードを適切に設定，秘密鍵を用いた SSH ログイン
N1-1-T	・暗号鍵またはその暗号鍵を保持する暗号機構に違法なプログラムがアクセスできた場合，差し替わったソフトウェアによって，速度制御器の本物の ID が悪用される ・攻撃者が，抽出した暗号鍵を利用して，速度制御器からのデータを通信経路上で傍受，遮断し，偽のデータに置き換えて，盗んだ暗号鍵で認証をパスする	メッセー認証コード（MAC），暗号等の改ざん防止機構を速度制御器に適用
N1-1-R	RaspberryPi のユーザが行った処理内容、通信内容のログが残っていない場合、ユーザが不正に行った操作の事実を否認するおそれがある	各種ログの取得，保全
N1-1-I	・攻撃者が，暗号化した鍵を悪用し，速度制御器とコントローラー（フィールド ゲートウェイまたはクラウド ゲートウェイ） との間の暗号鍵と復号鍵を入手し，それによって平文を手に入れる	マルウェア対策を実施，安全なキー管理
N1-1-D	・WAN やイーサネット経由で不正なアクセスが行われたり、大量のデータを受け取った場合、機能が停止するおそれがある。	応答回数制限の適用
N1-1-E	・OS の管理者設定が適切でない場合、本来 OS の管理者権限を持っていないユーザに管理者権限を持たせたり、管理者権限での実行を不正に利用されるおそれがある。	「管理者として実行」や「管理者権限を取得できるユーザの制限」

また STPA-SafeSec と STPA-Sec のセキュリティ分析を比較した結果と今後，対応すべきことを述べる．

① セーフティとセキュリティの枠組み設定に関して

STPA-SafeSec の場合，より抽象度の高い機能レイヤでは安全性リスク分析は行うが，セキュリティ分析は実施しない．そしてさらに具体的な物理的な機器ベースではじめてセキュリティ上の脆弱性を見つけ，対策を図るというある種セーフティ優先の枠組みを設定している．

一方，STPA-Sec は機能レイヤと物理レイヤに分け，セキュリティだけ物理レイヤで対処するという区別はなく，システムに必要な個所を具体化する System of Systems の発想で対処する．階層化を内包した枠組みである．この STPA-Sec の手順の方が詳細度の自由さがあると考えられる．

② セキュリティ・バイ・デザイン

セキュリティ要求を開発の早期段階から行い，適切なセキュリティ機能をシステムの機器自体に実装するセキュリティ・バイ・デサインは大変重要である．セキュリティ機能の不適切な機器が選択されてしまったら，セキュリティ対策を考えるのは手遅れを招き至り，手戻りを起こすことになるからである．

STPA-SafeSec の事例でいえば，機能レイヤの段階ではセキュリティ分析をしないのでセキュリティ・バイ・デザインとはいいがたい．

一方, STPA-Sec はセーフティとセキュリティの分析段階に区別をせず，何が問題であるかを元に掘り下げていくアプローチであり，セキュリティ・バイ・デザインに通じる発想である．

③ 脅威に対するモデリングに関して

STPA-SafeSec は，セキュリティ制約の設定，物理レイヤにおけるセキュリティ制約の割り付け，セキュリティに係る脆弱性への対応をしており, STPA-Sec よりセキュリティに係る脆弱性の具体化がなされている．ただし，攻撃視点により脅威を網羅的にとらえたシナリオを作成し、対策に結びつけるためには，STRIDE 分析などによる脅威モデリングが必要である．

またセキュリティ上の脆弱性対処に従った発想で物理レイヤでのセキュリティ分析を行う．この分析は STRIDE による脅威を事前に洗い出すモデリングとは異なり，物理レイヤのコンポーネントの既知のセキュリティ脆弱性への対処をするものである．

一方，STPA-Sec は，脆弱性分析と問題分析は明示しているものの，脅威分析は明示していない．脅威分析を含めないセキュリティ分析は, 要因の根拠が示しづらい. STIRIDE 分析を含んだ脅威モデリングの追加により，この問題は解決する．

④ 機密性への対応に関して

STPA-SafeSec は安全性だけでなく，完全性および可用性レベルの脅威を考慮しているが，情報セキュリティ上重要視されている機密性には触れていない．

制御セキュリティの場合，機密性は，可用性と完全性に比べて重要度が低いとされていることも起因しているかもしれない．

一方, STPA-Sec でシステム開発における対応セキュリティ属性は明確にされていないが，情報漏えいなどの機密性やプライバシーをアクシデントの事例としてあげている．ただし，MIT より具体的な的な内容が明示されているのは防衛上のセキュリティ調達要件などを分析するアプローチが多い．

6. おわりに

本論文では STAMP／ STPA -SafeSec に対して脅威分析の観点からより網羅性，実用性をもったセキュリティシナリオを導出するため，STRIDE を用いた脅威分析手法の追加を提案した．開発の早期段階でセーフティとセキュリティのリスクを洗い出すためには STPA -SafeSec の手順より STPA-Sec の手順に STRIDE を組み込む方が良いのではないかと考えられる．また，STPA、STPA-Sec の独自性である相互作用や非技術的問題に対するアプローチを実施したうえで，STRIDE は追加分析としてセキュリティ要因分析に用いることが向いていると考えられる．

今後の課題は，提案方式の検証，より詳細な事例の作成とできるだけ定量的な検証が挙げられる．セキュリティサイドの人にとって安全を含めて分析できるセキュリティ要求分析手法であり，セキュリティに縁のなかったセーフティサイドの人でも確実に分析できる統合手法として確立がされることが望まれる．

参考文献

1) Nancy G. Leveson, Engineering a Safer World, Systems Thinking Applied to Safety ,2012
2) ナンシー・G・レブソン，セーフウェア，安全・安心なシステムとソフトウェアを目指して
3) IPA，はじめての STAMP／STPA,2016
4) IPA，はじめての STAMP／STPA（実践編）, 2017
5) William Young, Nancy Leveson. Systems Thinking for Safety and Security, Proceedings of the 29th Annual Computer Security Applications Conference （ACSAC 2013）, pp.1-8 （2013）.
6) William Young, Reed Porada, System-Theoretic Process Analysis for Security（STPA-SEC）:Cyber Security and STPA, 2017 STAMP Conference
7) Ivo Friedberg, Kieran, Paul Smith, David Laverty and Sakir Sezer. STPA-SafeSec: Safety and security analysis for cyber-physical systems, Journal of Information Security and Applications, Volume 34, Part 2, pp.183-196 （2017）.
8) 岡本圭史，岡野浩三，STAMP　海外事例の紹介：STPA-SafeSec，SEC journal 52 号
9) NISC ,ww.nisc.go.jp/conference/seisaku/dai15/pdf/15siryou02.pdf
10) NISC,安全な IoT システムのためのセキュリティに関する一般的枠組
11) Schneier, B, Attack Trees. Dr. Dobb’s Journal of Software Tools 24（12） （1999） 21–29
12) Barbara Kordy, Sjouke Mauw, Saša Radomirović, Patrick Schweitzer, Foundations of Attack–Defense Trees
13) Sindre, G. and Opdahl. L. A : Eliciting security requirements with misuse cases, Requirements Engineering, Vol.10, No.1, pp. 34-44（2005）.
14) Steve Lipner ,Michael Howard,:信頼できるコンピューティングのセキュリティ開発ライフサイクル, https://msdn.microsoft.com/ja-jp/library/ms995349.aspx
15) Adam shostack, Threat Modeling: Designing for Security, Wiley 2014
16) 金子朋子・高橋雄志・大久保隆夫・勅使河原可海・佐々木良一，安全解析手法 STAMP/STPA に対するセキュリティ視点からの脅威分析の拡張提案，Computer Security Symposium 2017，pp.1273-1279，2017.10
17) Microsoft，Azure IoT リファレンス アーキテクチャの脅威のモデル化，https://docs.microsoft.com/ja-jp/azure/iot-suite/iot-security-architecture#threat-modeling-the-azure-iot-reference-architecture