一般財団法人 日本科学技術連盟
第４１年度（2025年度）ソフトウェア品質管理研究会成果発表会
演習コースⅣセーフティ＆セキュリティ
*2026*年　*3*月　*13*日（金）

STAMP/STPAを用いたセキュリティ分析設計：生成AIを活用した反復開発プロセスへの導入

リーダー	安樂 啓之（インフォテック）	吉村　隆広（アイホン）
研究員	藤井　広宣（ブライシス）	石原　佑一
	佐々木　瑛太（アズビル）	（東京海上日動システムズ）
	池上　達也（デンソー）	戸田　優作（アズビル）
	紫藤　紀行	永野　哲
	（三菱電機ソフトウエア）	(ミラクシアエッジテクノロジー)

主　　査	金子　朋子	（創価大学）
副 主 査	髙橋　雄志	（東日本国際大学）
アドバイザー	佐々木　良一	（東京電機大学）

アジェンダ

- 背景
- STAMP/STPAの概要
- 課題
- 提案手法
- 実験
- まとめ
- 今後の課題
- コースの感想
- 参考文献

背景

- 現代の情報システムは、複数のシステムが複雑に相互接続されるよう発展してきた

  個々の構成要素ではなく、要素間の「相互作用」に起因するセキュリティリスクが顕在化している

- 複雑なリスクを分析する手法が存在する。

  STAMP：システム理論に基づく事故モデル

  ⇒ STPA：STAMPを活用した分析手法

  ⇒ STPA-sec：STPAをセキュリティ分野へ応用

制御構造の観点から分析が可能

STAMPの概要

- **STAMP (System-Theoretic Accident Model and Processes: システム理論に基づく事故モデル)**
  - システムの事故の多くは構成要素の故障ではなくシステムの中で制御を行う**「制御要素(コントローラー)」**と**「被制御要素(被コントロールプロセス)」**の**相互作用**が適切に働かないことでおきているという前提
  - システムの構造を**「安全制御構造」(Safety Control Structure)**と名づけこれを作図するところからはじめる

システム理論に基づく事故モデルSTAMP

「STAMPガイドブック ～システム思考による安全分析～ 2019年3月公開」

STPAの概要

- **STPA（ System-Theoretic Process Analysis ）**
  「システムとシステム」「システムと人」といった要素間の関係性に注目し解析する手法

STPA手順　準備	
Step0 準備1	▪ **アクシデント・ハザード・安全制約の識別** アクシデント、ハザードを定義し、ハザードを制御するためのシステム上の安全制約を識別
Step0 準備2	▪ **コントロールストラクチャーの構築** 安全制約の実現に関係するコンポーネント(人・機器・システム・組織)とコンポーネント間の相互作用を分析し、コントロールストラクチャー図を構築する

「はじめてのSTAMP/STPA～システムに基づく新しい安全性解析手法 2016年3月公開」

STPAの概要

- **STPA（ System-Theoretic Process Analysis ）**
  「システムとシステム」「システムと人」といった要素間の関係性に注目し解析する手法

STPA手順　分析	
Step1	▪ **非安全なコントロールアクション（UCA）の抽出** コントロールストラクチャーから、安全制約の実行に必要なコントローラーによる指示（コントロールアクション）を識別し、4種類のガイドワードを適用して、ハザードにつながる非安全なコントロールアクション（UCA）を抽出する。
Step2	▪ **ハザード要因（HCF）の特定** 非安全なコントロールアクション（UCA）毎にどのような原因によって安全制約違反となりえるかといったハザード要因(HCF)を特定する。

※STPA-secではStep2でセキュリティ要因（SCF）も特定する

「はじめてのSTAMP/STPA～システムに基づく新しい安全性解析手法 2016年3月公開」

課題

- セキュリティバイデザインの原則に従うと、
  企画・要件定義段階からセキュリティ分析を行いたいが…

課題① 詳細なモデル化に要する時間と労力

- 制御構造図の作成・UCA識別に多大な工数が必要
- 大規模・複雑なシステムほど負担がさらに増大

→ 反復的な分析が現実的でない

＋

課題② 幅広い専門知識の必要性

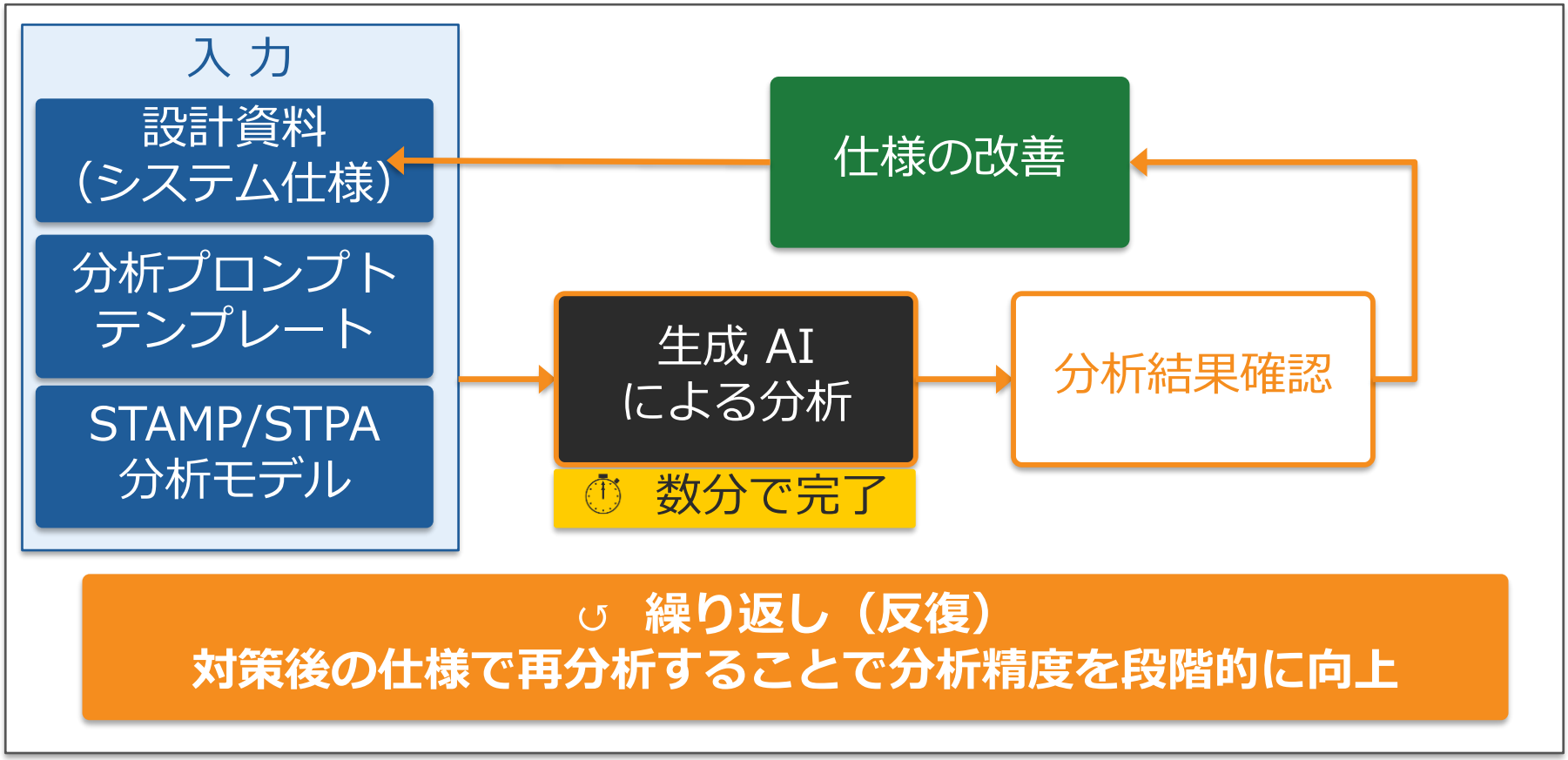
- セキュリティ専門知識 ＋ 対象業界の知識が不可欠
- 専門家の確保・育成が困難

→ 組織全体での導入が困難

反復的な STAMP/STPA セキュリティ分析の実施が困難

提案手法

- STAMP/STPAの分析モデルとシステム仕様を分析プロンプトとともに生成AIで解析
- 仕様を変えながら繰り返し分析することが容易に



期待される効果

- 繰り返し（反復）分析が可能となることからSTAMP/STPAセキュリティ分析がより身近なものとすることができる

分析時間の短縮

生成AI＋テンプレート化により数分で分析を完了

分析品質の確保

テンプレートに分析手順・セキュリティ観点を明示することで
専門家無しでも一定品質の分析を担保

専門知識の補完

規格・脅威情報・業界観点をテンプレートに組み込むことで
システム固有の文脈に基づく分析が可能

期待される効果

- 繰り返し（反復）分析が可能となることからSTAMP/STPAセキュリティ分析がより身近なものとすることができる

反復的な分析の実現

システム変更を行っても容易に再分析が可能となり
開発全体を通じた継続的なセキュリティ改善が可能に

セキュリティバイデザインの実現

企画・要件定義の段階からセキュリティ分析・改善が実施可能となり、開発初期からリスクを継続的に評価可能

実験

「STAMP/STPAによる反復的な分析」について、
下記観点で実験を行いました。

課題 1

短時間にリスクを識別することができるか？

課題 2

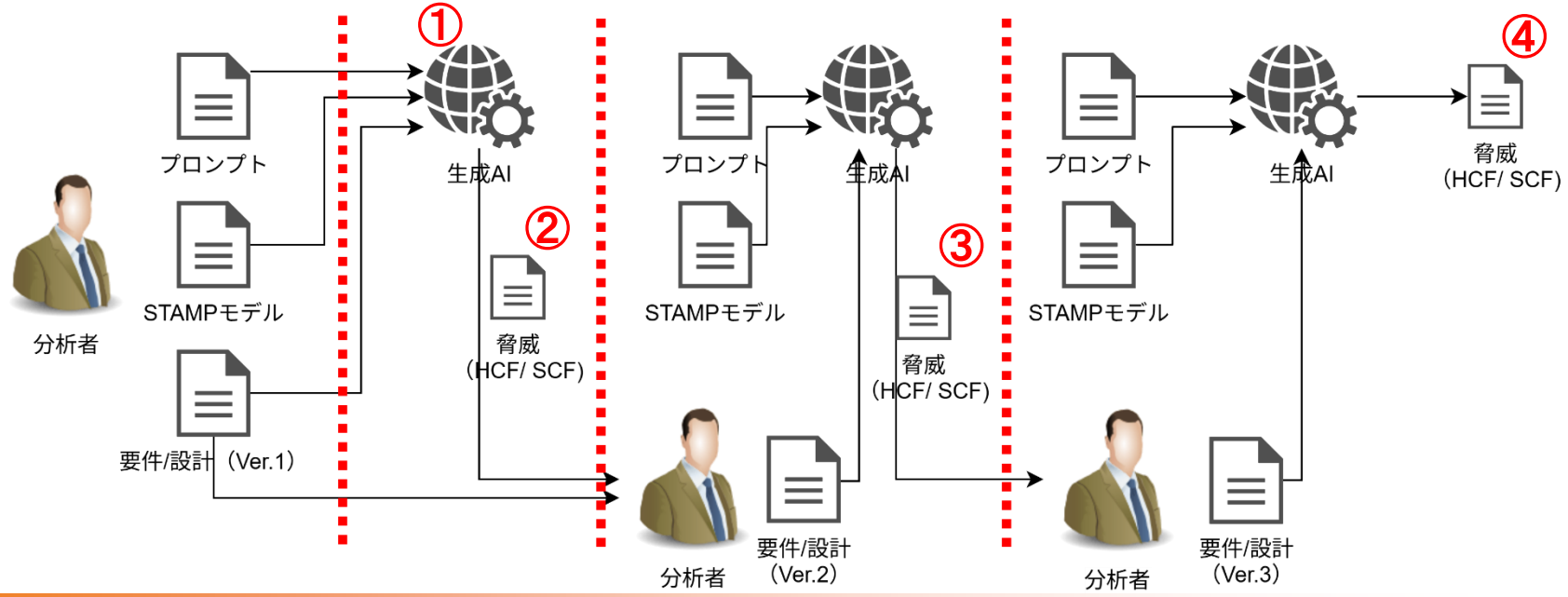
リスク対策済みの設計情報の再評価にも適用できるか？

課題 3

分析を繰り返すことで、
システム内のリスクコントロールを改善できるか？

実験

- **実験の手順は以下の通り**
  - 手順(１)：入力情報の準備
  - 手順(２)：実行時間を測定（下記①）
  - 手順(３)：対策済みのリスクは識別内容に反映されているか（②と③を比較）
  - 手順(４)：繰り返し実施した場合も同様か（④と③を比較）



手順(１)　入力情報の準備

- 生成AIによる分析を行えるよう、以下の入力情報を準備

準備1
プロンプト

- 生成AIを用いてSTAMP/STPAによる分析を行うプロンプト

準備2
STAMP/STPA
モデル

- STAMP/STPA分析に必要な分析情報
- （分析対象のアクシデント、ハザード、安全制約など）

準備3
システム仕様

- 分析対象のシステムに関する仕様書や設計資料
- （分析対象は**【スマートロックシステム】**とした）

手順(２)　実行時間の測定

- 実験(1)で準備したプロンプトを生成AIに読込ませ、STAMP/STPA分析を実行。分析時間を計測した。

①入力情報
- プロンプト
- STAMPモデル
- システム仕様

②生成AI分析
STAMP/STPA分析を実行

③分析完了

数分以内(5分未満)で完了

課題１：短時間にリスクを識別することができるか？

手順(３)　分析の反復（一回目）

- 実験(2)の分析結果に含まれるリスクに対して、システム仕様に対策を施し、再分析を実施した。

分析・対処前		対策実施	対策内容	分析・対処後		リスク変化	
UCA番号	識別されたリスク（HCF/SCF）			UCA番号	識別されたリスク（HCF/SCF）	発生頻度	影響度
UCA-1	・フィードバックの不十分・欠落・遅延	○	開錠操作の動作定義を改善	–	–	解消	解消
UCA-2	・プロセスモデルの不一致・不完全	○	開錠操作の動作定義を改善	–	–	解消	解消
UCA-3	・不十分な制御アルゴリズム	○	開錠操作の動作定義を改善	–	–	解消	解消
UCA-4	・上位からの指示や外部情報の誤り・欠落	○	施錠ロジックの制御異常への対策を実施	UCA-8	・プロセスモデルの不一致	減少	変化なし
UCA-7	上位からの指示や外部情報の誤り・欠落	○	FWアップデート処理の改善	–	–	解消	解消
UCA-11	・コントロールアクションの不適切・無効・欠落	○	開錠時の認証処理を改善	UCA-5	・モバイルアプリまたはクラウドサーバーの認証情報管理の不備	減少	変化なし
–	–	―	新たに検知された	UCA12	・　不十分な制御アルゴリズム	―	―
–	–	―	新たに検知された	UCA15	・　不正確な情報（ペアリング認証の不足）	―	―
UCA-1	Denial of Service（サービス拒否）	○	クラウドサーバの通信要件を具体化	UCA-1	Information Disclosure / Repudiation（情報漏洩/否認）	減少	減少
UCA-4	Tmpering（改ざん）	○	通信の暗号化対象を全体と明示	–	–	解消	解消
UCA-7	Denial of Service（サービス拒否）	○	FWアップデート処理の定義を追加	–	–	解消	解消
UCA-9	Spoofing（なりすまし）	○	スマートロックの設定機能の定義を追加	UCA-15	Elevation of Privilege（権限昇格）	変化なし	減少
UCA-11	Spoofing（なりすまし）	○	認証情報管理について具体化	UCA-5	Spoofing（なりすまし）	減少	変化なし
UCA-14	Information Disclosure（情報漏洩）	―		UCA-17	Information Disclosure（情報漏洩）	変化なし	変化なし
–	–	―	新たに検知された	UCA-4	Denial of Service（サービス拒否）	―	―
–	–	―	新たに検知された	UCA-12	Tampering / Elevation of Privilege（改竄/権限昇格）	―	―

手順(３)　分析の反復（一回目）

- **実験(2)の分析結果に含まれるリスクに対して、システム仕様に対策を施し、再分析を実施した。**

対策前

UCA 識別番号

識別されたリスク

実験(2)の分析で識別されたリスクを記載

UCA番号	識別されたリスク（HCF/SCF）	対策実施	対策内容	分析・対処後 UCA番号	分析・対処後 識別されたリスク（HCF/SCF）	リスク変化 発生頻度	リスク変化 影響度
UCA-1	・フィードバックの不十分・欠落・遅延	○	開錠操作の動作定義を改善	–	–	解消	解消
UCA-2	・プロセスモデルの不一致・不完全	○	開錠操作の動作定義を改善			解消	解消
UCA-3	・不十分な制御アルゴリズム	○					
UCA-4	・上位からの指示や外部情報の誤り・欠落	○					
UCA-5	・上位からの指示や外部情報の誤り・欠落	○					
UCA-11	・コントロールアクションの不適切・無効・欠落	○	開錠時[illegible]を改善	UCA-5	・モバイルアプリまたはクラウドサーバーの認証情報管理の不備	減少	変化なし
–	–	–	新たに検知された	UCA12	・ 不十分な制御アルゴリズム	–	–
–	–	–	新たに検知された	UCA15	・ 不正確な情報（ペアリング認証の不足）	–	–
UCA-1	Denial of Service（サービス拒否）	○	クラウドサーバの通信要件を具体化	UCA-1	Information Disclosure / Repudiation（情報漏洩/否認）	減少	減少
UCA-2	Tampering（改ざん）	○	通信の暗号化対象を全体と明示	–	–	解消	解消
UCA-4	Denial of Service（サービス拒否）	○	FWアップデート処理の定義を追加	–	–	解消	解消
UCA-15	Spoofing（なりすまし）	○	スマートロックの設定機能の定義を追加	UCA-15	Elevation of Privilege（権限昇格）	変化なし	減少
UCA-5	Spoofing（なりすまし）	○	認証情報管理について具体化	UCA-5	Spoofing（なりすまし）	減少	変化なし
UCA-14	Information Disclosure（情報漏洩）	–		UCA-17	Information Disclosure（情報漏洩）	変化なし	変化なし
–	–	–	新たに検知された	UCA-4	Denial of Service（サービス拒否）	–	–
–	–	–	新たに検知された	UCA-12	Tampering / Elevation of Privilege（改竄/権限昇格）	–	–

手順(３)　分析の反復（一回目）

- 実験(2)の分析結果に含まれるリスクに対して、システム仕様に対策を施し、再分析を実施した。

分析・対処前		対策実施	対策内容	分析・対処後		リスク変化	
UCA番号	識別されたリスク（HCF/SCF）			UCA番号	識別されたリスク（HCF/SCF）	発生頻度	影響度
UCA-1	・フィードバックの不十分・欠落・遅延	○	開錠操作の動作定義を改善				
UCA-2	・プロセスモデルの不一致・不完全	○	開錠操作の動作定義を改善				
UCA-3	・不十分な制御アルゴリズム	○	開錠操作の動作定義を改善				
UCA-4	・上位からの指示や外部情報の誤り・欠落	○	施錠ロジックの制御異常への対策を実施				
UCA-7	上位からの指示や外部情報の誤り・欠落	○	FWアップデート[illegible]の改善				
UCA-11	・コントロールアクションの不適切・無効・欠落	○	開錠時の認証処理を改善				
–	–	○	新たに検知された	UCA12			
–	–	○	新たに検知された	UCA15	[illegible]情報（ペアリング認証	—	—
UCA-1	Denial of Service（サービス拒否）	○	クラウドサーバの通信要件を具体化		Information Disclosure / Repudiation（情報漏洩/否認）	減少	減少
UCA-4	Tmpering（改ざん）	○	通信の暗号化対象を全体と明示	–	–	解消	解消
UCA-7	Denial of Service（サービス拒否）	○	FWアップデート[illegible]の定義を追加	–	–	解消	解消
UCA-9	Spoofing（なりすまし）	○	スマートロック[illegible]設定機能の定義を追加	UCA-15	Elevation of Privilege（権限昇格）	変化なし	減少
UCA-11	Spoofing（なりすまし）	○	認証情報管理について具体化	UCA-5	Spoofing（なりすまし）	減少	変化なし
UCA-14	Information Disclosure（情報漏洩）	—		UCA-17	Information Disclosure（情報漏洩）	変化なし	変化なし
–	–	—	新たに検知された	UCA-4	Denial of Service（サービス拒否）	—	—
–	–	—	新たに検知された	UCA-12	Tampering / Elevation of Privilege（改竄/権限昇格）	—	—

対策実施の有無

対策実施の内容

実験(2)で識別されたリスクに対して、対策を施した内容を記載

手順(３)　分析の反復（一回目）

- 実験(2)の分析結果に含まれるリスクに対して、システム仕様に対策を施し、再分析を実施した。

分析・対処前		対策実施	対策内容	対策後		リスク変化	
UCA番号	識別されたリスク（HCF/SCF）			UCA番号	識別されたリスク（HCF/SCF）	発生頻度	影響度
UCA-1	・フィードバックの不十分・欠落・遅延	○	開錠操作の動作定義を改善	–	–	解消	解消
UCA-2	・プロセスモデルの不一致・不完全	○	開錠操作の動作定義を改善	–	–	解消	解消
UCA-3	・			–	–	解消	解消
UCA-4	・り		策を	UCA-8	・プロセスモデルの不一致	減少	変化なし
UCA-7	上欠			–	–	解消	解消
UCA-11	・切			UCA-5	・モバイルアプリまたはクラウドサーバーの認証情報管理の不備	減少	変化なし
–	–		新た	UCA-6	・不十分な制御アルゴリズム	–	–
–	–	–	新たに検知された	UCA-15	・不正確な情報（ペアリング認証の不足）	–	–
UCA-1	Denial of Service（サービス拒否）	○	クラウドサーバの通信要件を具体化	UCA-1	Information Disclosure / Repudiation（情報漏洩/否認）	減少	減少
UCA-4	Tmpering（改ざん）	○	通信の暗号化対象を全体と明示	–	–	解消	解消
UCA-7	Denial of Service（サービス拒否）	○	FWアップデート処理の定義を追加	–	–	解消	解消
UCA-9	Spoofing（なりすまし）	○	スマートロックの設定機能の定義を追加	UCA-15	Elevation of Privilege（権限昇格）	変化なし	減少
UCA-11	Spoofing（なりすまし）	○	認証情報管理について具体化	UCA-6	Spoofing（なりすまし）	減少	変化なし
UCA-14	Information Disclosure（情報漏洩）	–		UCA-17	Information Disclosure（情報漏洩）	変化なし	変化なし
–	–	–	新たに検知された	UCA-4	Denial of Service（サービス拒否）	–	–
–	–	–	新たに検知された	UCA-12	Tampering / Elevation of Privilege（改竄/権限昇格）	–	–

対策後に再分析を行い、識別されたリスクを記載

対策後

UCA識別番号

識別されたリスク

手順(３)　分析の反復（一回目）

- 実験(2)の分析結果に含まれるリスクに対して、システム仕様に対策を施し、再分析を実施した。

分析・対処前		対策	対策内容	分析・対処後		リスク変化	
UCA番号					ク（HCF/SCF）	発生頻度	影響度
UCA-1						解消	解消
UCA-2						解消	解消
UCA-3						解消	解消
UCA-4					の不一致	減少	変化なし
UCA-7	上位からの指示や外部情報の誤り・欠落	○	FWアップデート処理の改善			解消	解消
UCA-11	・コントロールアクションの不適切・無効・欠落	○	開錠時の認証処理を改善	UCA-5	・モバイルアプリまたはクラウドサーバーの認証情報管理の不備	減少	変化なし
–	–	―	新たに検知された	UCA12	・　不十分な制御アルゴリズム		
–	–	―	新たに検知された	UCA15	・　不正確な情報（ペアリング認証の不足）		
UCA-1	Denial of Service（サービス拒否）	○	クラウドサーバの通信要件を具体化	UCA-1	Information Disclosure / Repudiation（情報漏洩/否認）	減少	減少
UCA-4	Tmpering（改ざん）	○	通信の暗号化対象を全体と明示	–	–	解消	解消
UCA-7						解消	解消
UCA-9					ilege（権限昇	変化なし	減少
UCA-11					）	減少	変化なし
UCA-14					情報漏	変化なし	変化なし
–	–	―	新たに検知された	UCA-4	Denial of Service（サービス拒否）	―	―
–	–	―	新たに検知された	UCA-12	Tampering / Elevation of Privilege（改竄/権限昇格）	―	―

対策前後に識別された同一リスクについて、**そのリスクが発生する頻度の変化**を記載

対策前後に識別された同一リスクについて、**そのリスクがもたらす影響度の変化**を記載

発生頻度の変化

影響度の変化

手順(３)　分析の反復（一回目）

- 識別されたリスクに対して対策することで、
  リスクの発生頻度・影響度が**減少**することを確認できた。

分析・対処前		対策実施	対策内容	分析・対処後		リスク変化	
UCA番号	識別されたリスク（HCF/SCF）			UCA番号	識別されたリスク（HCF/SCF）	発生頻度	影響度
UCA-1	・フィードバックの不十分・欠落・遅延	○	開錠操作の動作定義を改善	–	–	解消	解消
UCA-2	・プロセスモデルの不一致・不完全	○	開錠操作の動作			解消	解消
UCA-3	・不十分な制御アルゴリズム	○	開錠操作の動作				解消
UCA-4	・上位からの指示や外部情報の誤り・欠落	○	施錠ロジックの制御要件への対策を実施	UCA-8	・プロセスモデルの不一致	減少	変化なし
UCA-7	上位からの指示や外部情報の誤り・欠落	○	FWアップデート処理の改善	–	–	解消	解消
UCA-11	・コントロールアクションの不適切・無効・欠落	○	開錠時の認証処理		・モバイルアプリまたはクラウド管理の不備	減少	変化なし
–	–	―	新たに検知された		ゴリズム	―	―
–	–	―	新たに検知された		の不足）	―	―
UCA-1	Denial of Service（サービス拒否）	○	クラウドサーバの通信要件を具体化	UCA-1	Information Disclosure / Repudiation（情報漏洩/否認）	減少	減少
UCA-4	Tmpering（改ざん）	○	通信の暗号化対象を全体と明示	–	–	解消	解消
UCA-7	Denial of Service（サービス拒否）	○	FWアップデート処理の定義を追加	–	–	解消	解消
UCA-9	Spoofing（なりすまし）	○	スマートロックの設定機能の定義を追加	UCA-15	Elevation of Privilege（権限昇格）	減少	減少
UCA-11	Spoofing（なりすまし）	○	認証情報管理について具体化	UCA-5	Spoofing（なりすまし）	減少	変化なし
UCA-14	Information Disclosure（情報漏洩）	―		UCA-17	Information Disclosure（情報漏洩）	変化なし	変化なし
–	–	―	新たに検知された	UCA-4	Denial of Service（サービス拒否）	―	―
–	–	―	新たに検知された	UCA-12	Tampering / Elevation of Privilege（改竄/権限昇格）	―	―

■発生頻度：４件減少

■影響度：２件減少

手順(３)　分析の反復（一回目）

- 識別されたリスクに対して対策することで、
  リスクの発生頻度・影響度が**減少**することを確認できた。

分析・対処前		対策実施	対策内容	分析・対処後		リスク変化	
UCA番号	識別されたリスク（HCF/SCF）			UCA番号	識別されたリスク（HCF/SCF）	発生頻度	影響度
UCA-1	・フィードバックの不十分・欠落・遅延	○	開錠操作の動作定義を改善	–	–	解消	解消
UCA-2	・プロセスモデルの不一致・不完全	○	開錠操作の動作定義を改善	–	–	解消	解消
UCA-3	・不十分な制御アルゴリズム	○	開錠操作の動作定義を改善	–	–	解消	解消
UCA-4	・上位からの指示や外部情報の誤り・欠落	○	施錠ロジックの制御異常への対策を実施	UCA-8	・プロセスモデルの不一致	減少	変化なし
UCA-7	上位からの指示や外部情報の誤り・欠落	○	FWアップデート処理の改善	–		解消	解消
UCA-11			錠時の認証処理を改善	UCA			変化なし
–			たに検知された	UCA			–
–			たに検知された	UCA			–
UCA-1			ラウドサーバの通信要件を具体化	UCA-1	（情報漏洩/否認）	減少	減少
UCA-4		○	通信の暗号化対象を全体と明示	–		解消	解消
UCA-7	of Service（サービス拒否）	○	FWアップデート処理の定義を追加	–	–	解消	解消
UCA-9	Spoofing（なりすまし）		追加	UCA-15	Elevation of Privilege（権限昇格）	変化なし	減少
UCA-11	Spoofing（なりすまし）	○	認証情報管理について具体化	UCA-5	Spoofing（なりすまし）	減少	変化なし
UCA-14	Information Disclosure（情報漏洩）	—		UCA-17	Information Disclosure（情報漏洩）	変化なし	変化なし
–	–	—	新たに検知された	UCA-4	Denial of Service（サービス拒		
–	–	—	新たに検知された	UCA-1			

すべての利用権限を
奪取されるリスク

一部の利用権限のみ
奪取されるリスク

被害範囲が限定され、
影響度が**減少**したと判断

手順(３)　分析の反復（一回目）

- 識別されたリスクに対して対策することで、
  リスク自体が**解消**することも確認できた。

分析・対処前		対策実施	対策内容	分析・対処後		リスク変化	
UCA番号	識別されたリスク（HCF/SCF）			UCA番号	識別されたリスク（HCF/SCF）	発生頻度	影響度
UCA-1	・フィードバックの不十分・欠落・遅延	○	開錠操作の動作定義を改善	–	–	解消	
UCA-2	・プロセスモデルの不一致・不完全	○	開錠操作の動作定義を改善	–	–	解消	
UCA-3	・不十分な制御アルゴリズム	○	開錠操作の動作定			解消	
UCA-4	・上位からの指示や外部情報の誤り・欠落	○	施錠ロジックの制 実施		の不一致	減少	変化なし
UCA-7	上位からの指示や外部情報の誤り・欠落	○	FWアップデート処理の改善	–	–	解消	
UCA-11	・コントロールアクションの不適切・無効・欠落	○	開錠時の認証処理を改善	UCA-5	・モバイルアプリまたはクラウドサーバーの認証情報管理の不備	減少	変化なし
–	–	―	新たに検知された	UCA12	・ 不十分な制御アルゴリズム	―	―
–	–	―	新たに検知された	UCA15	・ 不正確な情報（ペアリング認証の不足）	―	―
UCA-1	Denial of Service（サービス拒否）	○	クラウドサーバの通信要件を具体化	UCA-1	Information Disclosure / Repudiation（情報漏洩/否認）	減少	減少
UCA-4	Tmpering（改ざん）	○	通信の暗号化対象を全体と明示	–	–	解消	
UCA-7	Denial of Service（サービス拒否）	○	FWアップデート処理の定義を追加	–	–	解消	
UCA-9	Spoofing（なりすまし）	○	スマートロックの設定機能の定義を追加	UCA-15	Elevation of Privilege（権限昇格）	変化なし	減少
UCA-11	Spoofing（なりすまし）	○	認証情報管理について具体化	UCA-5	Spoofing（なりすまし）	減少	変化なし
UCA-14	Information Disclosure（情報漏洩）	―		UCA-17	Information Disclosure（情報漏洩）	変化なし	変化なし
–	–	―	新たに検知された	UCA-4	Denial of Service（サービス拒否）	―	―
–	–	―	新たに検知された	UCA-12	Tampering / Elevation of Privilege（改竄/権限昇格）	―	―

6件のリスク解消

手順(３)　分析の反復（一回目）

- リスク対策後の再分析により、発生頻度や影響度が**減少**、リスクが**解消**することを確認できた。

分析・対処前		対策実施				リスク変化	
UCA番号	識別されたリスク（HCF/SCF）				（HCF/SCF）	発生頻度	影響度
UCA-1	・フィードバックの不十分・欠落・遅延	○	開錠操作の			解消	解消
UCA-2	・プロセスモデルの不一致・不完全	○	開錠操作の			解消	解消
UCA-3	・不十分な制御アルゴリズム	○	開錠操作の			解消	解消
UCA-4	・上位からの指示や外部情報の誤り・欠落	○	施錠ロジッ実施		不一致	減少	変化なし
UCA-7	上位からの指示や外部情報の誤り・欠落	○	FWアップデート処理の改善	–	–	解消	解消
UCA-11	・コントロールアクションの不適切・無効・欠落	○	開錠時の認証処理を改善	UCA-5	・モバイルアプリまたはクラウドサーバーの認証情報管理の不備	解消	変化なし
–	–	―	新たに検知された	UCA12	・ 不十分な制御アルゴリズム		
–	–	―	新たに検知された	UCA15	・ 不正確な情報（ペアリング認証の不足）		
UCA-1	Denial of Service（サービス拒否）	○	クラウドサーバの通信要件を具体化	UCA-1	Information Disclosure / Repudiation（情報漏洩/否認）	減少	減少
UCA-4	Tmpering（改ざん）	○	通信の暗号化対			解消	解消
UCA-7	Denial of Service（サービス拒否）	○	FWアップデー			解消	解消
UCA-9	Spoofing（なりすまし）	○	スマートロック追加		権限昇	変化なし	減少
UCA-11	Spoofing（なりすまし）	○	認証情報管理に			減少	変化なし
UCA-14	Information Disclosure（情報漏洩）	―			（情報漏	変化なし	変化なし
–	–	―	新たに検知された	UCA-4	ービス拒否）	―	―
–	–	―	新たに検知された	UCA-12	Tampering / Elevation of Privilege（改竄/権限昇格）	―	―

■発生頻度
- リスク解消：６件
- リスク減少：４件

■影響度
- リスク解消：６件
- リスク減少：２件

課題2:リスク対策済みの設計情報の再評価にも適用できるか？ → ○

手順(４)　分析の反復（二回目）

- 手順(3)の分析結果に含まれるリスクに関して、再度システム仕様に対策を施し、再分析を実施。

分析・対処前		対策実施	対策内容	分析・対処後		リスク変化	
UCA番号	識別されたリスク（HCF/SCF）			UCA番号	識別されたリスク（HCF/SCF）	発生頻度	影響度
UCA-8	・プロセスモデルの不一致	○	スマートロック利用条件としての認証要件を明示	UCA-2	・プロセスモデルの不一致	変化なし	変化なし
UCA-5	・モバイルアプリまたはクラウドサーバーの認証情報管理の不備	○	多要素認証による認証エラー対策	–	–	解消	解消
UCA-12	・ 不十分な制御アルゴリズム	○	FWの署名実施を追加	UCA-4	・不十分な制御アルゴリズム	減少	変化なし
UCA-15	・ 不正確な情報（ペアリング認証の不足）	○	セキュアなBluetooth通信を要件に追加	UCA-7	・不適切なコントロールアクション	減少	変化なし
–	–	ー		UCA-1	・動作の遅れ	–	–
–	–	ー		UCA-5	・意図しない外乱	–	–
–	–	○		UCA-8	・不十分な制御アルゴリズム	–	–
UCA-1	Information Disclosure / Repudiation（情報漏洩/否認）	○	通信のセッション管理を厳格に変更/暗号要件追加	–	–	減少	減少
UCA-15	Elevation of Privilege（権限昇格）	○	アプリケーションの権限を最小にする要件追加	UCA-7	・Information Disclosure（情報漏洩）	減少	減少
UCA-5	Spoofing（なりすまし）	○	サービスの認証要件を具体的に記載	–	–	解消	解消
UCA-17	Information Disclosure（情報漏洩）	○	セキュアなBluetooth通信を要件に追加	UCA-8	・Denial of Service（サービス拒否）	減少	変化なし
UCA-4	Denial of Service（サービス拒否）	○	多様所認証など認証の厳格化	–	–	解消	解消
UCA-12	Tampering / Elevation of Privilege（改竄/権限昇格）	○	アプリケーションの権限を最小にする要件追加	UCA-4	・Tampering（改ざん）	減少	変化なし
–	–	ー		UCA-2	・Spoofing（なりすまし）	–	–
–	–	ー		UCA-9	・Denial of Service（サービス拒否）	–	–

手順(４)　分析の反復（二回目）

- 手順(3)の分析結果に含まれるリスクに関して、再度システム仕様に対策を施し、再分析を実施。

分析・対処前		対策実施		分析・対処後		リスク変化	
UCA番号	識別されたリスク（HCF/SCF）			UCA番号	識別されたリスク（HCF/SCF）	発生頻度	影響度
UCA-8	・プロセスモデルの不一致	○	スマートロ 証要件を明		ー致	変化なし	変化なし
UCA-5	・モバイルアプリまたはクラウドサーバーの認証情報管理の不備	○	多要素認証			解消	解消
UCA-12	・　不十分な制御アルゴリズム	○	FWの署名実		リズム	減少	変化なし
UCA-15	・　不正確な情報（ペアリング認証の不足）	○	セキュアなBluetooth通信を要件に追加	UCA-7	・不適切なコントロールアクション	減少	変化なし
–	–	ー		UCA-1	・動作の遅れ	–	–
–	–	ー		UCA-5	・意図しない外乱	–	–
–	–	○		UCA-8	・不十分な制御アルゴリズム	–	–
UCA-1	Information Disclosure / Repudiation（情報漏洩/否認）	○	通信のセッション管理を厳格に変更/暗号要件追加	–	–	–	–
UCA-15	Elevation of Privilege（権限昇格）	○	アプリケーションの権限を最小にする要件追加	UCA-7	・Information Disclosure（情報漏洩）	–	–
UCA-5	Spoofing（なりすまし）	○	サービスの認証要件を最終的に記載			解消	解消
UCA-17	Information Disclosure（情報漏洩）	○	セキュアなBluetooth 追加		ス拒	減少	変化なし
UCA-4	Denial of Service（サービス拒否）	○	多様所認証など認			解消	解消
UCA-12	Tampering / Elevation of Privilege（改竄/権限昇格）	○	アプリケーション る要件追加			–	変化なし
–	–	ー				–	–
–	–	ー		UCA-9	・Denial of Service（サービス拒否）	–	–

■発生頻度
- リスク解消：３件
- リスク減少：６件

■影響度
- リスク解消：３件
- リスク減少：２件

- 再分析よりリスクが**解消**、発生頻度と影響度が**減少**した。

課題3：分析を繰り返すことで、システム内のリスクコントロールを改善できるか?

→ ○

実験結果

実験結果から、提案手法を用いることで、
「STAMP/STPAによる反復的な分析」は**可能**と考える

課題１

短時間にリスクを識別することができるか？

➡ 数分以内で分析が完了したため可能と判断

課題２

リスク対策済みの設計情報の再評価にも適用できるか？

➡ リスクが解消・減少したため可能と判断

課題３

分析を繰り返すことで、
システム内のリスクコントロールを改善できるか？

➡ 再分析でもリスクが解消・減少したため有効と判断

まとめ

生成AIとプロンプトテンプレートを組み合わせることで、
「STAMP/STPAによる反復的なセキュリティ分析」
が可能であることを検証できた。

これにより、**以下も確認することができた。**

- **専門知識を持たない開発者でもSTAMP/STPAに基づくセキュリティ分析が実施可能であること。**
- **開発の早期段階からセキュリティリスクを継続的に評価することが可能であること。**

今後の課題

本研究の課題は以下の通り。

(1)多様なシステムでの汎用性検証

(2)AIエージェント化と組織展開

(3)複数生成AIモデルの比較評価とテンプレート可搬

(4)RAGによる知識更新の実装

(5)経済性を考慮したレビュー体制の確立

今後の課題(1)

(1)多様なシステムでの汎用性検証

本実験ではスマートロックシステムの検証にのみに留まっている。

異なる規模や複雑なシステムへ適用し、提案手法の汎用性の検証が必要

(2)AIエージェント化と組織展開

本研究で構築したプロンプトテンプレートをAIエージェント化することで担当者間の分析結果のばらつき抑制、組織全体の分析品質の均一化が期待

本実験では実装および運用効果の測定ができず、確認が必要

(3)複数生成AIモデルの比較評価とテンプレート可搬

構造化されたテンプレートは特定のAIモデルに依存しない設計となっているが、本研究ではGeminiの利用のみに留まっている。

Claude，ChatGPT等、複数モデルでの出力品質の比較検証、プロンプトテンプレートの可搬性評価が課題

今後の課題(２)

(4)RAGによる知識更新の実装

脅威はサービス提供開始後も進化し続けるため，RAGを活用し，
最新の脅威情報やセキュリティ規格を分析に取り込む仕組みの実装が必要

本実験では実装および運用効果の測定ができず、確認が必要

(5)経済性を考慮したレビュー体制の確立

生成AIは網羅的にリスクを抽出できるため，費用対効果が低いものも
提示する場合がある。

以下を考慮した効率的なレビュー体制の確立が課題

- **守るべき資産の優先順位付け**
- **インシデント発生確率**
- **インシデントの影響度**

コースの感想

参加者	感想・コメント
安樂 啓之（インフォテック）	今回は、STAMP/STPAによる分析の自動化として、セキュリティをテーマとしたプロセス改善につながる内容について、取り組めてよかったです。自社でのSecurity by Designについての取り組みもかなり進み、とても充実した活動をすることができました。
藤井　広宣（ブライシス）	セーフティ・セキュリティにおいて何を目的として、何をすべきかの初歩が学べたと感じています。また、AIが運用ハードルを下げる実例に遭遇できたことも価値があると感じました。
佐々木　瑛太（アズビル）	システムに潜むリスクを分析する方法としてSTAMP/SMTPを学ぶことができました。これまでリスク分析は過去のリスク一覧を基に思いついたものを列挙するという方法でやっていたので、システムの相互作用に着目するという方法を学べたことは勉強になりました。
池上　達也（デンソー）	STAMP/STPAのセキュリティ適用について知見が深まりました。また生成AIによる解析結果を何度もかけることで堅牢なセキュリティが確保されていくのが実感でき、大変良かったです。
吉村　隆広（アイホン）	去年に引き続き、2回目の分科会参加となりました。希望していたセキュリティ分野の研究やSTAMP/STPAの手法理解など活動の中で学ぶことも多く感謝しています。

コースの感想

参加者	感想・コメント
石原　佑一（東京海上日動システムズ）	セキュリティを分析・評価するうえでのSTAMPについて触れることができたことが大きな成果でした。どうしても概念の理解に時間がかかるところはありますが、今回の研究で生成AIの活用でその点を短縮できることにメリットがあると感じました。
戸田　優作（アズビル）	個人でセキュリティを学習すると、技術的分野ばかりに目が行きがちです。本コースを通してセキュリティの品質管理的な観点での分析手法を体系的に学べたことで、より実用的な知識を得ることが出来たと感じています。
永野　哲（ミラクシアエッジテクノロジー）	製品セキュリティに関する要求が高まっていますが、弊社ではあまり知見がないため、参加させていただきました。相互作用に着目するSTAMP/STPAやFRAMなど、上流設計での手法が学べ、知見が広がりました。ありがとうございました。
紫藤　紀行（三菱電機ソフトウエア）	これまでなんとなく実施していたセキュリティ分析を、STAMPという理論的枠組みで体系的に理解することができました。さらに、生成AIを活用して分析を自動化することで、実運用での扱いやすさが向上し、社内への導入がしやすくなったのはよかったと感じています。

SQiP
Software Quality Profession

Q&A

ご清聴ありがとうございました。
ご質問があればお願いいたします。

参考文献

- 独立行政法人 情報処理推進機構(IPA)，制御システムのセキュリティリスク分析ガイド 第2版，2023年
- 経済産業省商務情報政策局，第8回産業サイバーセキュリティ研究会 事務局説明資料，2024年
- 独立行政法人 情報処理推進機構(IPA)，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，2019年
- 独立行政法人 情報処理推進機構(IPA)，制御システム関連のサイバーインシデント事例3：Stuxnet攻撃，2020年
- 独立行政法人 情報処理推進機構(IPA)，はじめての STAMP/STPA ～システム思考に基づく新しい安全性解析手法～ Ver.1.0，2016年
- Nancy G. Leveson, “Engineering a Safer World”, MIT Press, 2012
- 西澤賢一ほか, セーフティ&セキュリティ開発におけるSTAMP/STPAの有効性検証 日本科学技術連盟SQiP 2018年
- 福島 祐子, 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, (日本ユニシス技報 41(2))，2021
- Gregory M. Pope, Systemic Theoretic Process Analysis (STPA) Used for Cyber Security and Agile Software Development, Lawrence Livermore National Laboratory, 2021
- 森川聡久ほか，「開発SEが使える!今注目のリスク分析手法 STAMP/STPAのシステム開発への適用～システム開発でのSTAMP/STPAの実践を通じて得られたテーラリングのエッセンス～」，ソフトウェアプロセス改善カンファレンス（SPI Japan 2020），2020
- 独立行政法人 情報処理推進機構(IPA)，はじめてのSTAMP/STPA（実践編）～システム思考に基づく新しい安全性解析手法～，2017年
- 安樂啓之ほか, 日本科学技術連盟SQiP, STAMP/CAST分析における生成AIの支援～羽田港宇高航空機衝突事故を題材として～, 2025年
- Nancy G. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety，The MIT Press，2012
- 独立行政法人 情報処理推進機構(IPA)，はじめての STAMP/STPA ～システム思考に基づく新しい安全性解析手法～ Ver.1.0，2016年
- Kalle Rindell et al., Security in agile software development: A practitioner survey, Information and Software Technology Volume 131, 2021.
- 大森淳夫ほか, 日本科学技術連盟SQiP, セーフティ＆セキュリティ開発のための技術統合提案と事例作成 ～STAMP/STPAとアシュアランスケースの統合～, 2018年
- [ ]金子朋子ほか, 情報処理学会研究報告, 安全性解析手法STAMP／STPAへの脅威分析（＝STRIDE）の適用, 2018年
- Yi Qi et al., Safety Analysis in the Era of Large Language Models: A Case Study of STPA using ChatGPT, Machine Learning with Applications, 2025
- Simon Diemert, Jens H. Weber, Can Large Language Models assist in Hazard Analysis?, arXiv preprint, 2023