COTTAGE:セキュアなシステム開発の実現に必要なアタックディフェンスツリー作成ツールの提案

大久保研究室
修士課程M1　山本 渓太

2023/08/21

目 次

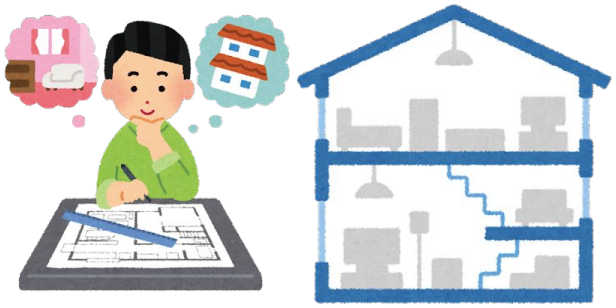
- 背　景
- 先行研究
- 方　針
- 検証(ツール比較
- 今後の課題
- まとめ

目 次

背　景

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

- **セキュリティ・バイ・デザイン**の普及



運用から開発まで一貫したセキュリティの実装

「信頼性の向上」「コスト削減」「点検・改修が簡単」

情報の収集

・媒体固有の弱点(**脆弱性**)
・発見された**脅威・攻撃方法**
・上記の**対策**
を収集・公開する

脅威分析

・守るべきもの(**資産**)はなにか
・資産に対してどんな**脅威**があるか
・どの様な**リスク**があるか
を洗い出し、対策を検討する

背　景

情報の収集

MITRE（マイターと読む
①**米国**土安全保障局が公開（権威性あり
②**継続的な**更新（信頼性あり
③**関連情報の記載**（拡張性あり

CAPEC™

攻撃方法について記載

セキュリティに焦点

軽減策が記載

関連するCWE・CAPECの**リンクあり**

CWE

脆弱性について記載

使用目的：
①**脅威モデリング**
②開発者の教育・訓練
③侵入テスト

背　景

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

CAPEC

攻撃方法
カペック

表 1　CAPEC の記載内容

情報名	内　容
Name	攻撃パターン名
Description	概　要
Relationships	他の攻撃パターンとの関係
Consequence	攻撃後に攻撃対象に起こる結末
Prerequisites	攻撃が成立するための事前条件
Execution Flow	攻撃の実行手順
Mitigations	軽減策
Related Weaknesses	関連する CWE

CWE

脆弱性
シーダブリューイー

表 2　CWE の記載内容

情報名	内　容
Name	脆弱性パターン名
Description	概　要
Extended Description	詳細の概要
Alternate Terms	別の用語
Relationships	他の脆弱性パターンとの関係
Modes Of Introduction	脆弱性がどの段階で発生するか
Applicable Platforms	該当プラットフォーム
Common Consequences	一般的な結果
Observed Examples	関連する CVE
Potential Mitigations	考えられる軽減策

背　景

脅威分析

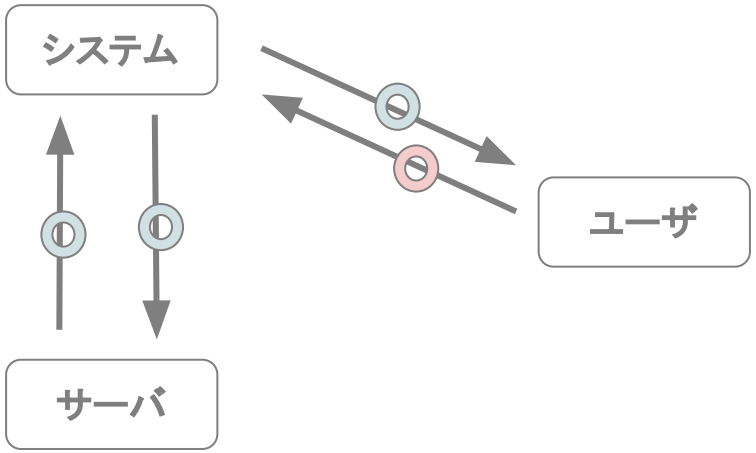
資産ベース

・**ミスユースケース**

⇨誤った使い方で意図しない挙動を洗い出す

・データフローダイヤグラム

⇨データの移動に着目して媒体の境界に発生する脅威を洗い出す

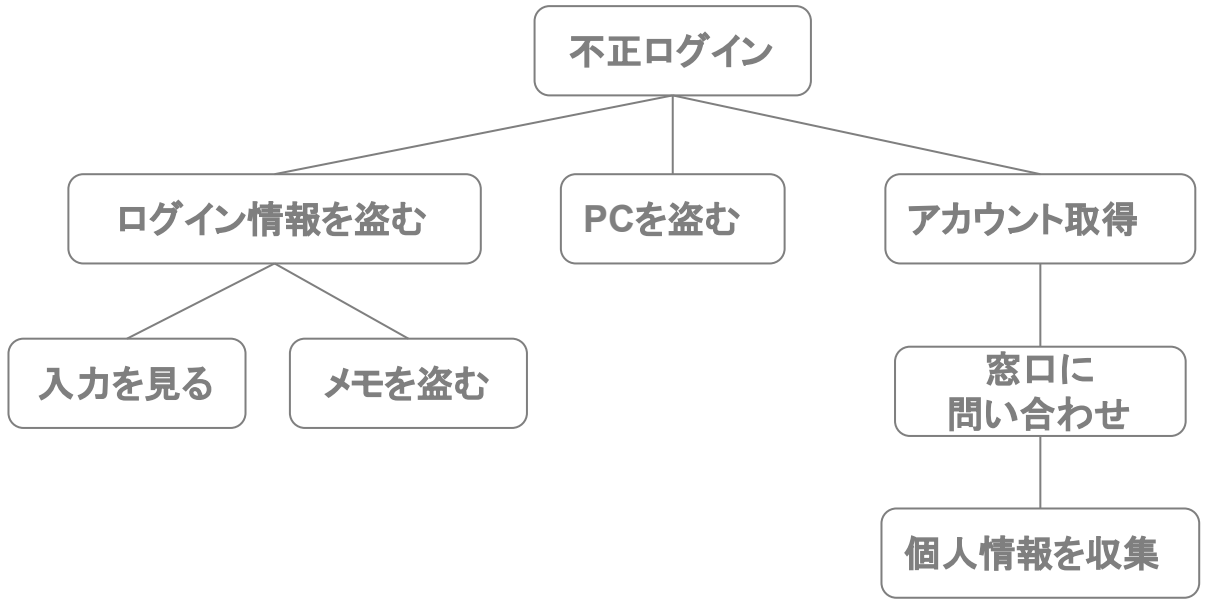


詳細まで分析出来ない
⇨**シナリオベース**などで追加の分析が必要

シナリオベース

・**アタックツリー**など

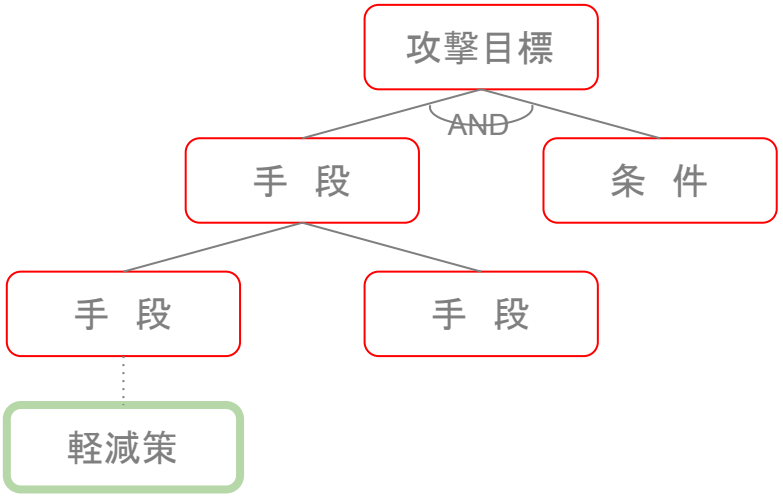
⇨ツリー構造の頂点にゴールを設定して手段を記載していく



背　景

アタックディフェンスツリー（以降ADツリー

攻撃者が達成したい最終目標を頂点にどういった**条件があるか**を下層に記載する。
また、その軽減策も同一ツリーに記載する



赤：アタックノード
緑：ディフェンスノード

- メリット👍
  ①特定の攻撃にどんな条件が必要か把握できる
  ⇨**実行可能性が高い**攻撃手段、**コストの低い**軽減策を検討できる

  ②**軽減策を踏まえた攻撃**策も検討できる

  ③手段に対して策が講じられているかを**一元的**に管理できる

- デメリット
  ①**専門性**が高い
  ⇨業界・セキュリティに対する知識が必要

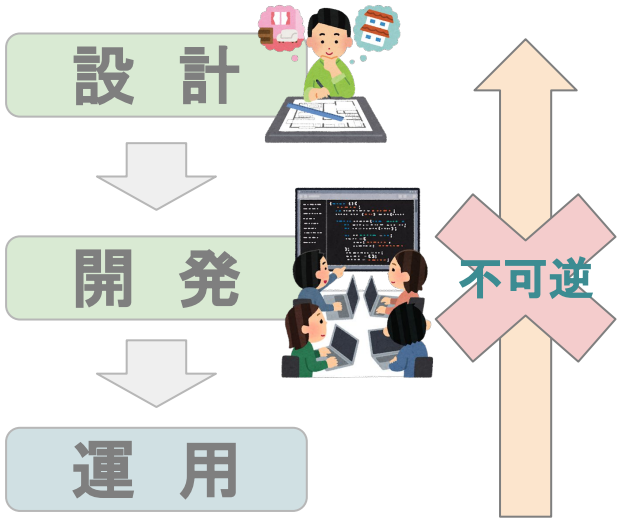
  ②**時間**がかかる

高い専門性を軽減する

背　景

- 開発手法の**多様化**

設計、開発、リリースの各段階で達成すべき目標を明確にしてから開発する手法



DevOps 設計、開発、リリースを一定期間で反復する手法

開発と運用の担当が連携する組織体制のための概念

DevOpsも対応した脅威分析の作成ツールを開発しよう

目 次

先行研究

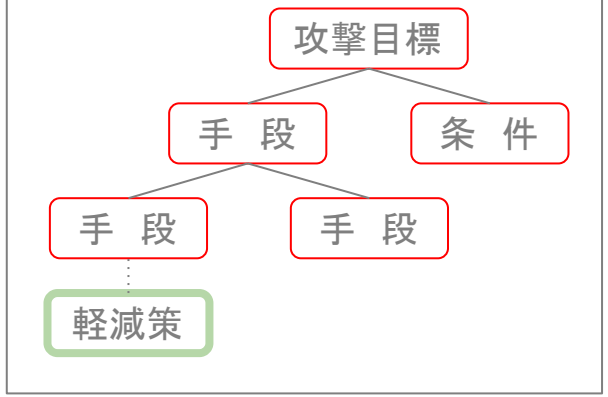
- **DevOps**で脅威分析は何が有効か？

①Efficient secure DevOps using process mining and Attack Defense Trees.
OKUBO, Takao; KAIYA, Haruhiko.
Procedia Computer Science, 2022, 207: 446-455.

プロセスマイニングと**ADツリー**を用いた**効率的なセキュアDevOps**
⇨**DevOps**には**ADツリー**が有効だと提案

DevOps：繰り返す開発方法

ADツリー：脅威分析手法



- 脅威分析の**作成ツール**

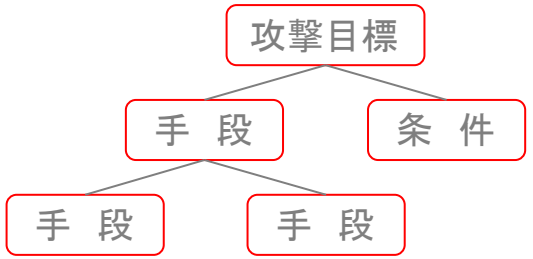
②**FRAT&RATTATA**：**アタックツリー**再利用フレームワークの提案及び実践ツールの開発、2023年、大矢
⇨**CAPECの自動分類**フレームワーク**FRAT**とアタックツリー**作成ツールRATTATAを作成**

CAPEC™

攻撃方法
カペック

＋

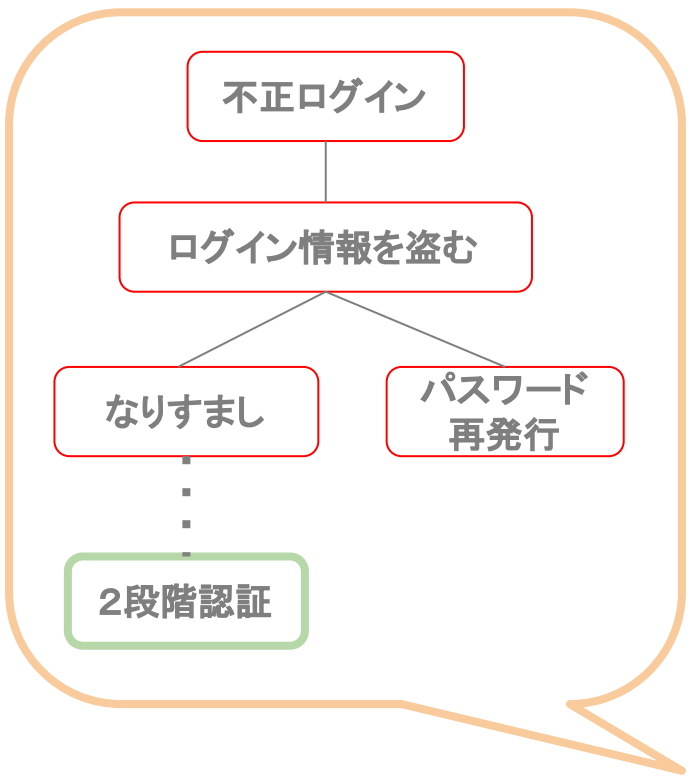
アタックツリー：
脅威分析手法

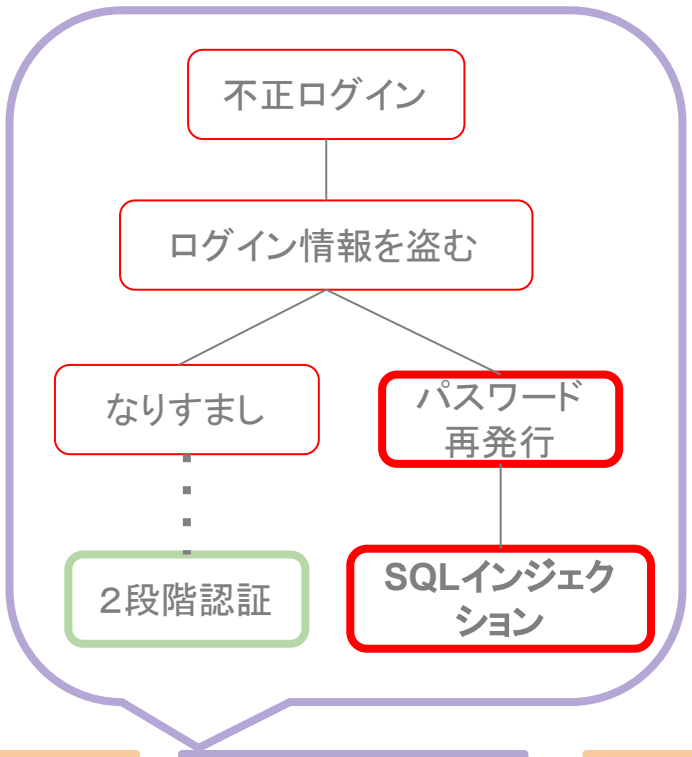


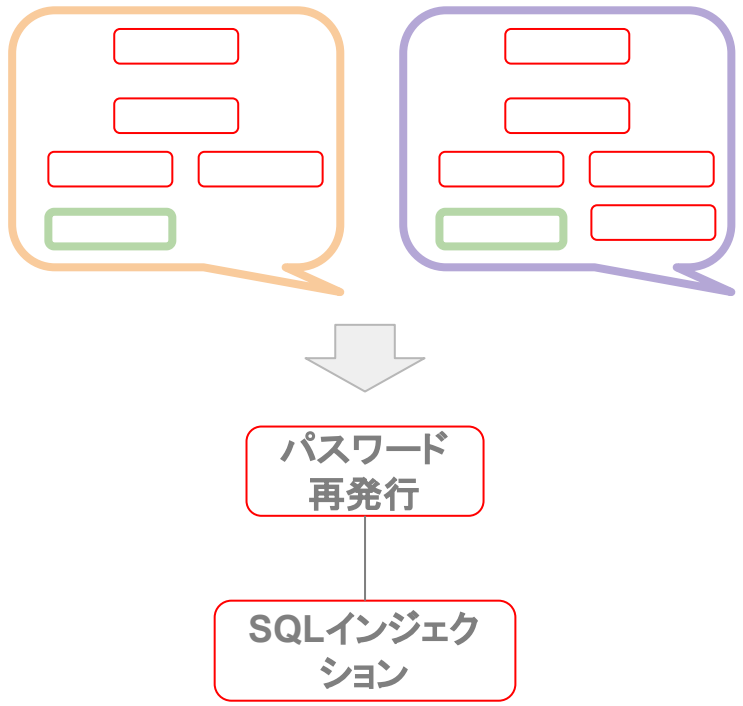
先行研究①(大久保ら

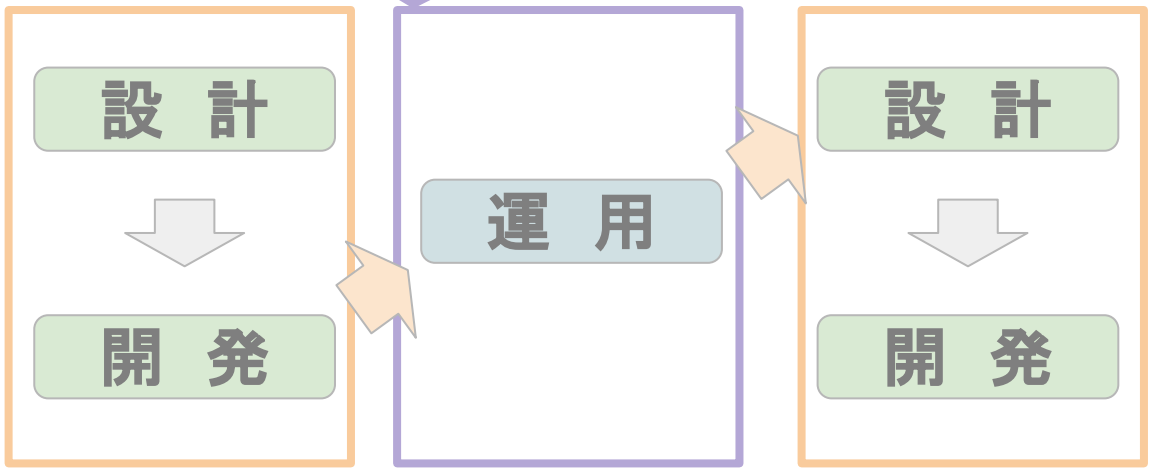
DevOps:繰り返す開発方法

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







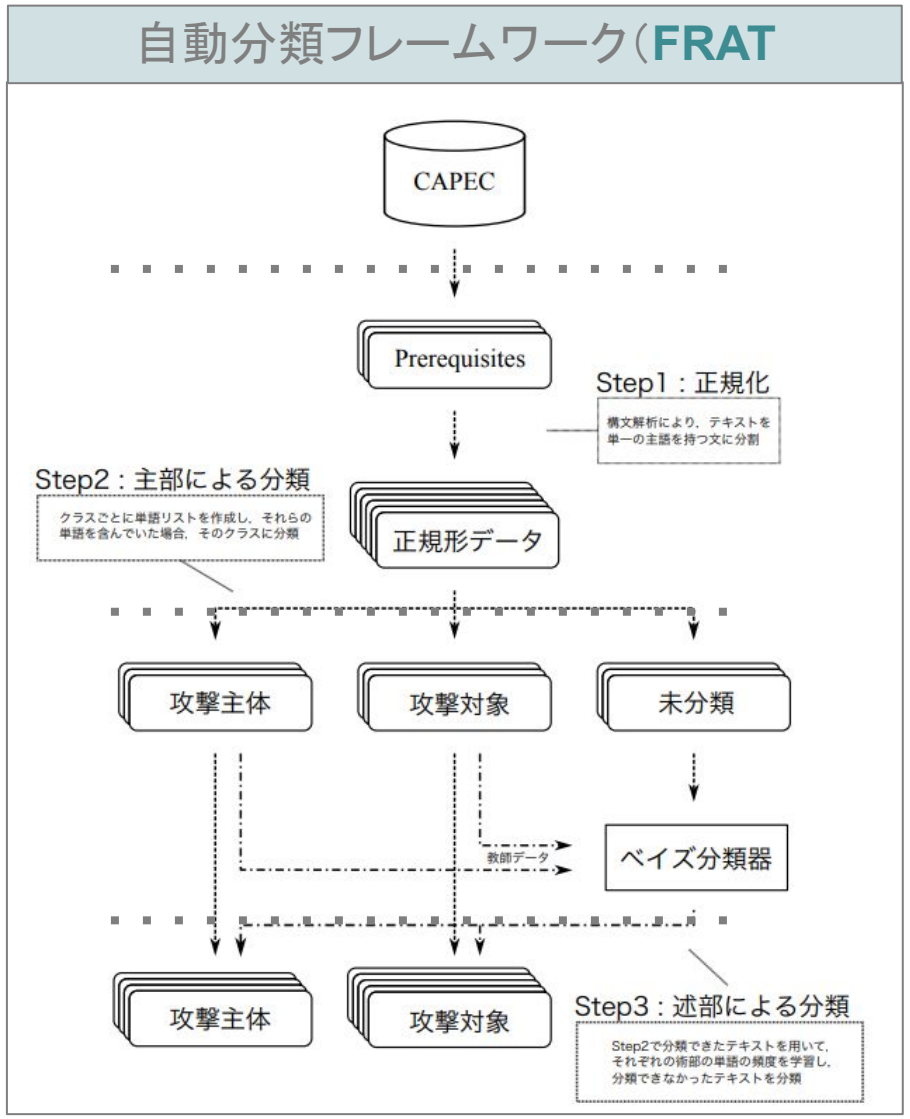


運用で発見された脆弱性を同一分析上で共有できる

先行研究②(大矢 FRAT & RATTATA

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

自動分類フレームワーク(FRAT



Prerequisites　攻撃が成立するための事前条件

事前条件A　事前条件B　事前条件C

A-1　A-2

攻撃主体:
敵の行動

攻撃対象:
我の状態

未分類:
その他

攻撃主体:
敵の行動

攻撃対象:
我の状態

攻撃主体のみをツリーとして使用

CAPEC™

攻撃方法
カペック

目 次

- 背　景
- 先行研究
- **方　針**
- 検証(ツール比較
- 今後の課題
- まとめ

提　案

- 提　案

ADツリー作成ツール「**COTTAGE**」の提案
Collaborative **T**ool of **T**hreat **A**nalysis **G**azes at **E**nemies

①CAPEC・CWEから**引用ツリー**を自動生成する
②ADツリー**作成ツール**の開発

- 目　的

① **DevOps**含めた広い開発手法に対応した脅威分析を実施できる

② より**初心者**の脅威分析をサポートする

③FRAT＆RATTATAの弱点を克服する

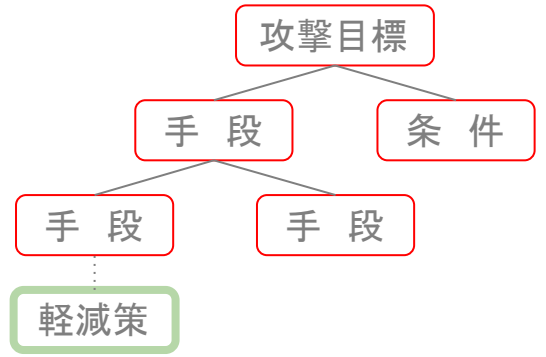
DevOps：繰り返す開発方法

CAPEC™　CWE

攻撃方法
カペック

脆弱性
シーダブリューイー

ADツリー：脅威分析手法



FRAT＆RATTATA

FRAT：先行研究の自動分類フレーム
RATTATA：アタックツリー作成ツール

方　針

- 先行研究の差分

大久保らの研究では**提案**のみ、大矢ツールは**アタックツリー**作成ツール
⇨**DevOps**に役立つよう**ADツリー**に拡張

大矢のFRATは**CAPECのみ**
⇨脆弱性知識ベース**CWE**を追加（初心者＋DevOps

FRAT＆RATTATAの**弱点を克服**する
⇨次スライド

FRAT：先行研究の分類フレーム

CAPEC™

攻撃方法
カペック

＋

アタックツリー：
脅威分析手法

CAPEC™　CWE

攻撃方法
カペック

脆弱性
シーダブリューイー

＋

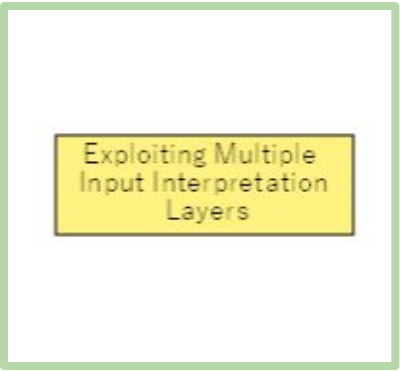
ADツリー：脅威分析手法

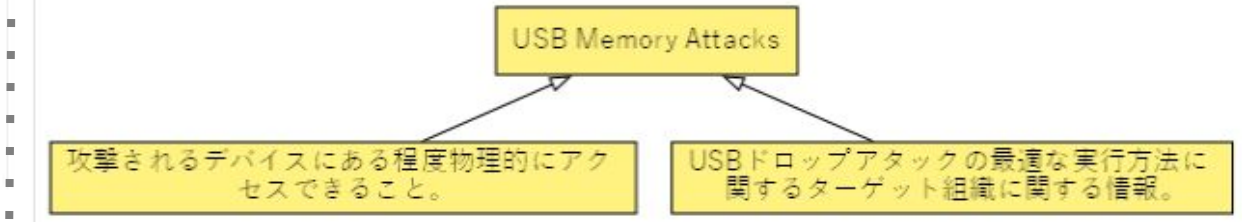
方　針

- FRAT＆RATTATAの弱点とは？

大矢のFRATはCAPECの**攻撃主体**のみを引用ツリーに使用
理由：
①アタックツリー定義：「**攻撃者の行動**」のみを記載する
②**攻撃対象を1つでも**満たしていない場合はツリーを**使用できない**
③**再利用**にフォーカスした研究

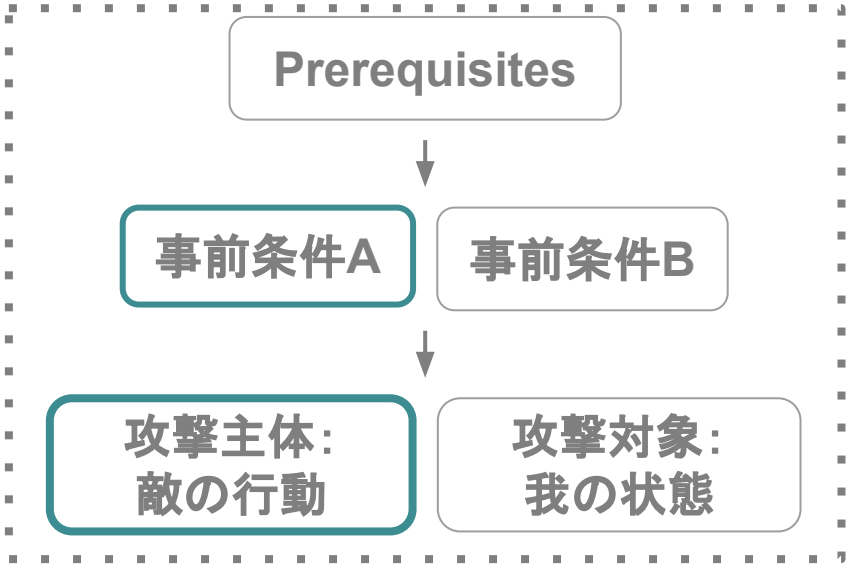
◯COTTAGE：**事前条件を使用**
理由：
①情報が少なすぎる（約6割が下黄緑枠かつ約3割は分類誤り
②属人性が高い・**攻撃対象を満たすか**は**設計で**決める
③DevOpsでの相互の認識統一を重視

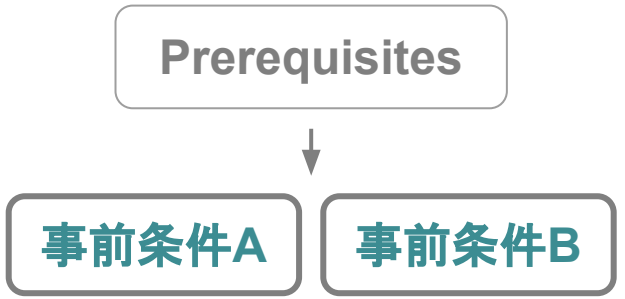




情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

FRAT：先行研究の分類フレーム





目 次

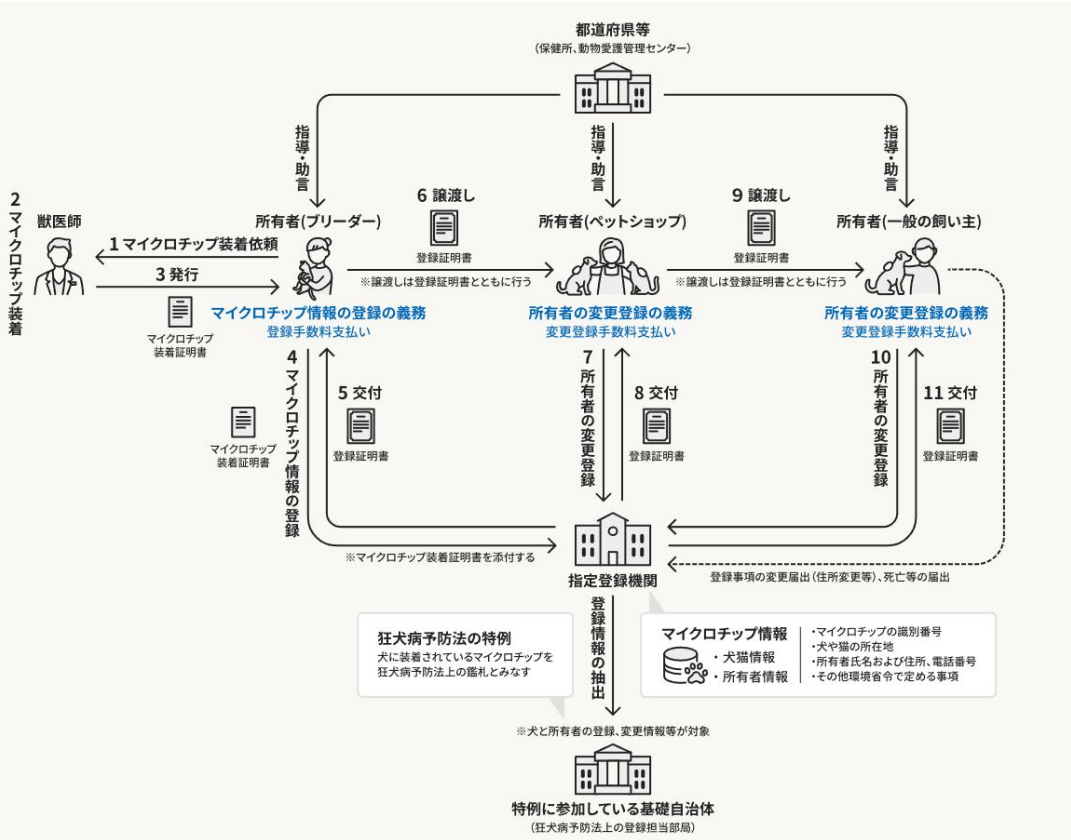
- 背　景
- 先行研究
- 方　針
- **検証（ツール比較**
- 今後の課題
- まとめ

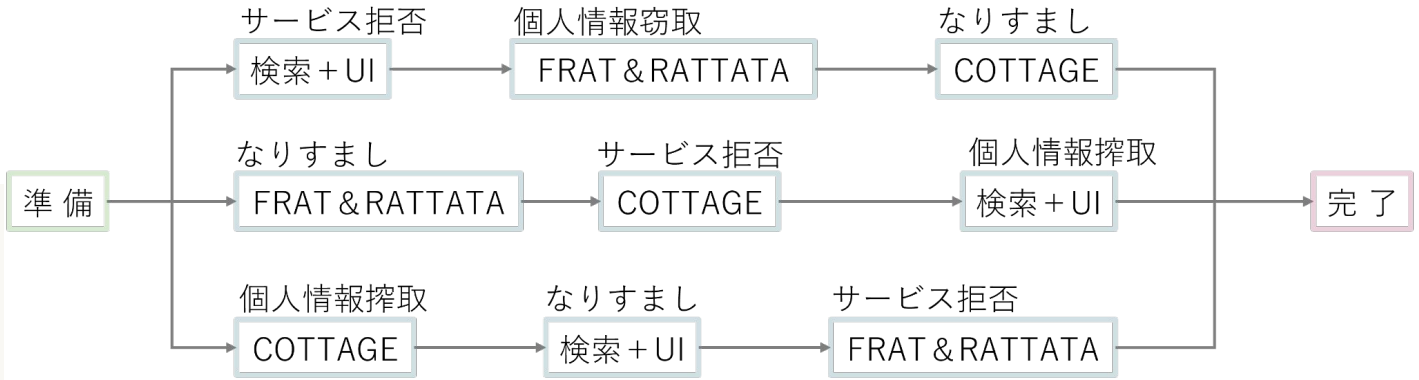
検証（ツール比較

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

- 概　要

情セ大の6名に対して、システムの仕様書を配布し、下記を脅威分析
検索＋UI⇨引用機能を潰したCOTTAGE





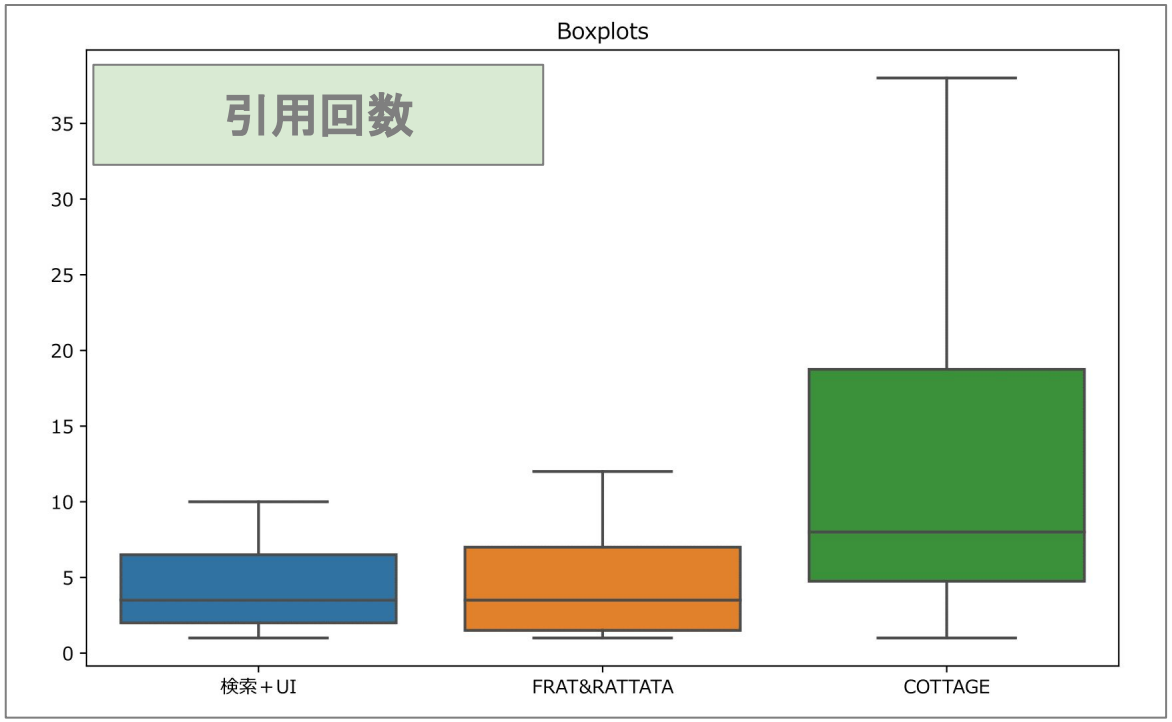
段階（時間）	項目	概要
準備（40 分）	全般説明	検証に関する全般説明
	事前アンケート	業務経験や資格の有無等
	アタックディフェンスツリー	アタックディフェンスツリーの説明
	CAPEC	CAPEC の概要説明
	ツール	共通するツール操作の説明
	仕様書	今回使用する仕様書の説明
検証（160 分）	検証 1 回目	1 つ目のツール練習と検証を実施
	検証 2 回目	2 つ目のツール練習と検証を実施
	検証 3 回目	3 つ目のツール練習と検証を実施
終了（10 分）	事後アンケート	各ツールに関する意見や感想

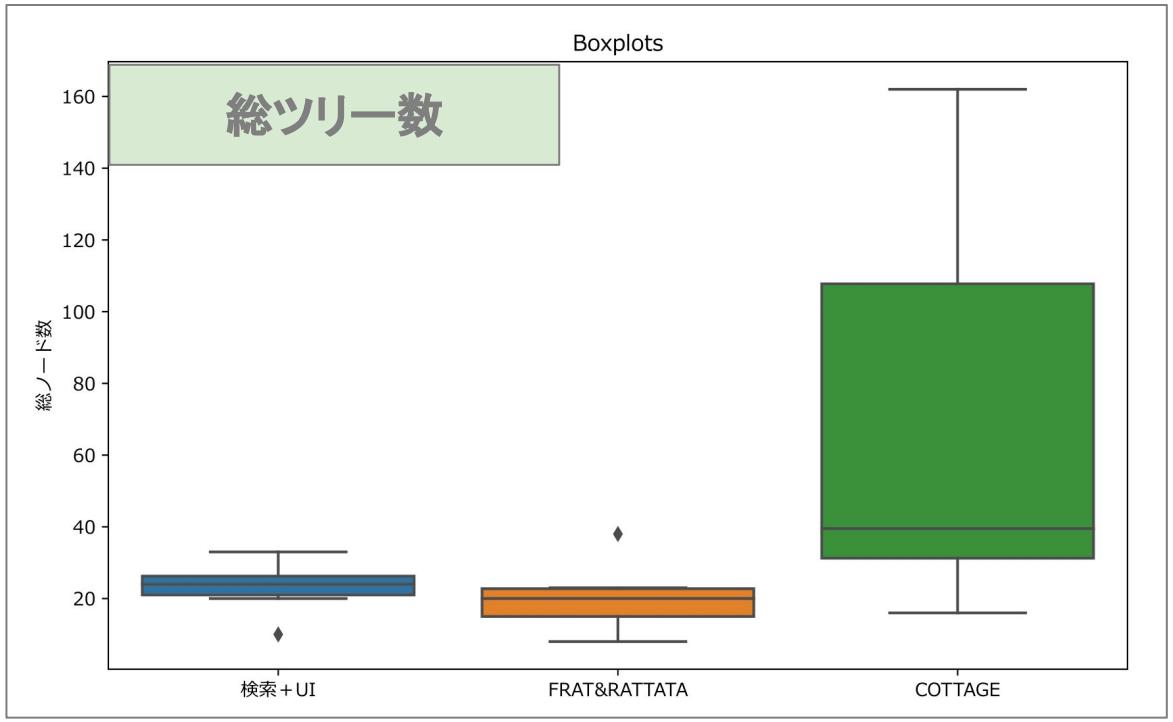
検証(ツール比較

- **統計的有意差**は確認できなかった
  - ⇨原因:人数が少ない
  - ⇨**中央値**では**COTTAGE**が優勢

- 事後アンケート

「一番使いやすいツールは？」で**COTTAGEが一位(回答率100%**





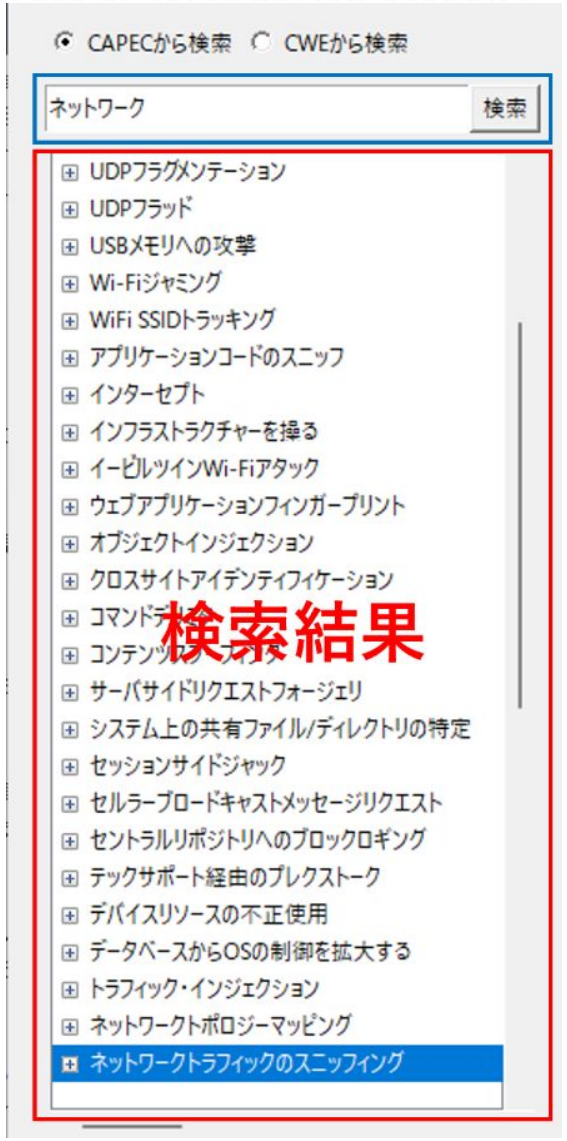
目 次

今後の課題

- 引用ツリー**検索結果が多すぎ**て選びにくい
  - ⇨各知識ベースの**「抽象度」**で区分して表示する

- 自動生成の**テキスト取得**方法
  - 知識ベースから必要なテキストを取得するところで失敗する
  - ⇨取得する方式を変更する

```
<xhtml:br/>define(`LOCAL_SHELL_FLAGS',
<xhtml:div style="margin-left:10px;">ifdef(`LOCAL_SHELL_FLAGS',
    <xhtml:div style="margin-left:10px;">`translit(LOCAL_SHELL_FLAGS, `9')',
    <xhtml:br/>`eu'))
    </xhtml:div>
</xhtml:div>
```

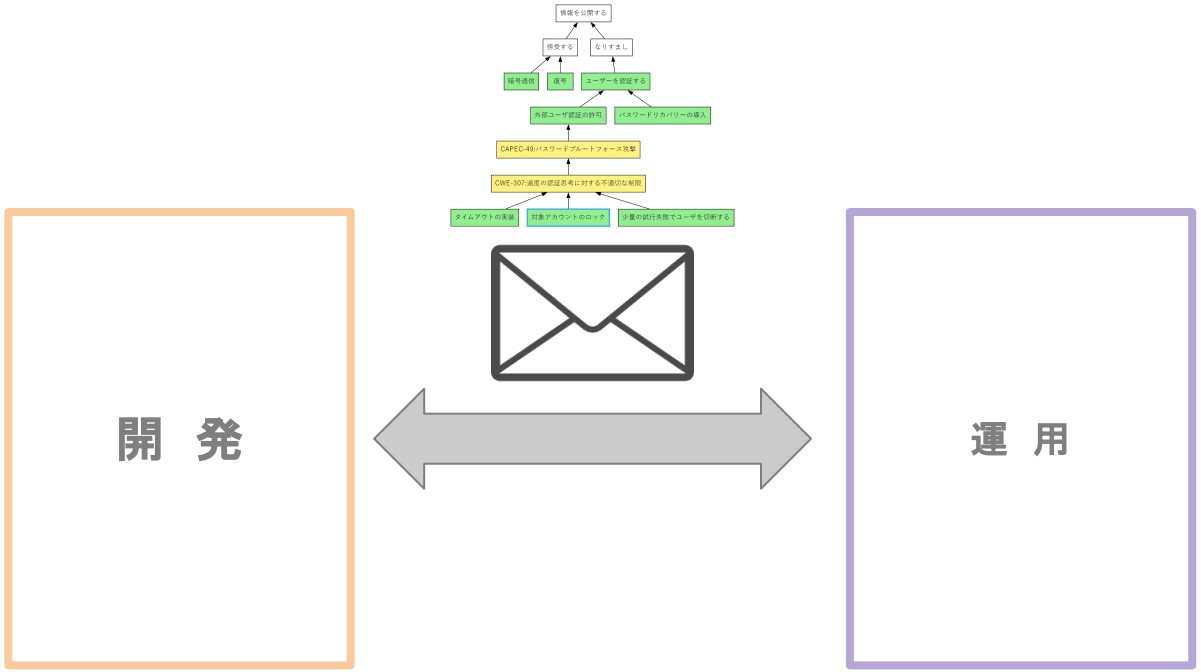


今後の課題

- ADツリーコミュニティの実現

企業、組織が作成したADツリーを他組織、取引先が**確認・再利用**できるようにする

本ツールは**個人用**（ダウンロード、引用ツリーの生成も各自で
⇨作成した**ツリーのやり取り(公開・非公開)を簡単**に
⇨**Webアプリ**に再構築する



目 次

- 背　景
- 先行研究
- 方　針
- 検証(ツール比較
- 今後の課題
- **まとめ**

まとめ

- セキュリティ・バイ・デザインの重要性が高まった
- サイクルする開発手法DevOpsに対応した脅威分析
  ADツリー作成ツール「COTTAGE」を作成した
- 6名によるツールの比較で、COTTAGEの優位を確認できた
- 今後の課題
  ①引用ツリーの検索、自動生成

  ②ツールのWebアプリ化

終わり

ご清聴ありがとうございました。

ATT&CKは？

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

- CAPEC・CWEとATT&CKとの違い

CWE
CAPEC™

ATT&CK®

アプリの**セキュリティ**に焦点

システムの**脆弱性**を列挙

軽減策が記載

使用目的:
①脅威モデリング
②開発者の教育・訓練
③侵入テスト

ネットワーク防御が焦点

悪意のある動作を理解

防御オプションのテスト・分析を支援

使用目的:
①新たな脅威の探索
②脅威インテリジェンスの強化
③敵対者シミュレーション演習

補　足①

CAPEC™

CAPEC-1: ACL によって適切に制限されていない機能へのアクセス

攻撃パターンID：1
抽象化:標準

カスタマイズされた情報を表示します。　概念的な　稼働中　マッピングに優しい　完了

説明

アプリケーション、特に Web アプリケーションでは、機能へのアクセスは承認フレームワークによって軽減されます。このフレームワークは、アクセス制御リスト (ACL) をアプリケーションの機能の要素にマップします。特に Web アプリの URL。管理者が特定の要素の ACL を指定しなかった場合、攻撃者は何の罰も受けずにその要素にアクセスできる可能性があります。ACL によって適切に制限されていない機能にアクセスできる攻撃者は、機密情報を取得し、アプリケーション全体を危険にさらす可能性があります。このような攻撃者は、より高い特権レベルのユーザーのみが利用できるリソースにアクセスしたり、アプリケーションの管理セクションにアクセスしたり、本来は想定されていないデータのクエリを実行したりする可能性があります。

人間関係

実行フロー

前提条件

アプリケーションは、アプリケーションの要素 (サブセクション) を ACL に関連付ける方法でナビゲート可能である必要があります。

さまざまなリソースまたは個々の URL は、攻撃者が何らかの方法で検出できる必要があります。

管理者は ACL を関連付けるのを忘れたか、不適切に許可された ACL を特定のナビゲート可能なリソースに関連付けた可能性があります。

緩和策

J2EE 設定では、管理者は、「NoAccess」など、オーセンティケーターがユーザーに付与することが不可能なロールを、ユーザーが表示およびアクセスできる限られた数のサーブレットによってアクセスが保護されているすべてのサーブレットに関連付けることができます。

これを行うと、それらの保護されたサーブレットへの直接アクセスが Web コンテナによって禁止されます。

より一般的な設定では、管理者は、ユーザーに公開されると想定されているリソース以外のすべてのリソースを、ユーザーが引き受けることが不可能な役割によってアクセスできるものとしてマークする必要があります。デフォルトのセキュリティ設定では、アクセスを拒否し、ビジネス ロジックによって意図されたリソースへのアクセスのみを許可する必要があります。

関連する弱点

https://capec.mitre.org/data/definitions/1.html

補　足②

https://cwe.mitre.org/data/definitions/285.html

情報セキュリティ大学院大学

CWE-285: 不適切な認証

弱点ID：285
抽象化:クラス
構造:単純

カスタマイズされた情報を表示します。 概念的な | 稼働中 | マッピングフレンドリー | 完了 | カスタム

説明

アクターがリソースにアクセスしたりアクションを実行しようとしたときに、製品は承認チェックを実行しないか、誤って実行します。

- 拡張説明
- 代替用語
- 人間関係
- 背景の詳細
- 導入方法
- 適用可能なプラットフォーム
- 一般的な結果
- エクスプロイトの可能性
- 実証例
- 観察された例
- 潜在的な緩和策

フェーズ: アーキテクチャとデザイン

製品を匿名領域、通常領域、特権領域、および管理領域に分割します。役割をデータと機能に注意深くマッピングすることで、攻撃対象領域を削減します。ロールベースのアクセス制御 (RBAC) を使用して、適切な境界でロールを強制します。

このアプローチは水平認可から保護できない可能性があることに注意してください。つまり、同じロールを持つ他のユーザーを攻撃することからユーザーを保護することはできません。

フェーズ: アーキテクチャとデザイン

- 検出方法
- メンバーシップ
- 分類マッピング
- 関連する攻撃パターン
- 参考文献
- コンテンツ履歴

提出物

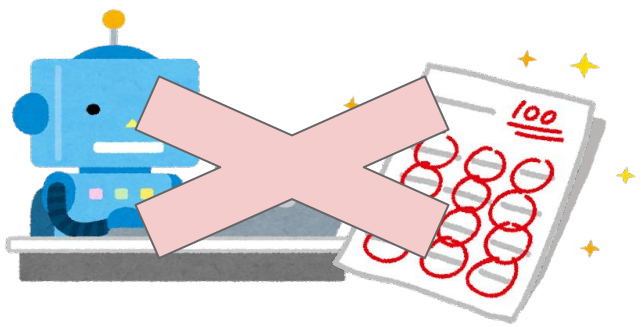
提出日	提出者	組織
2006-07-19	7 有害な王国	

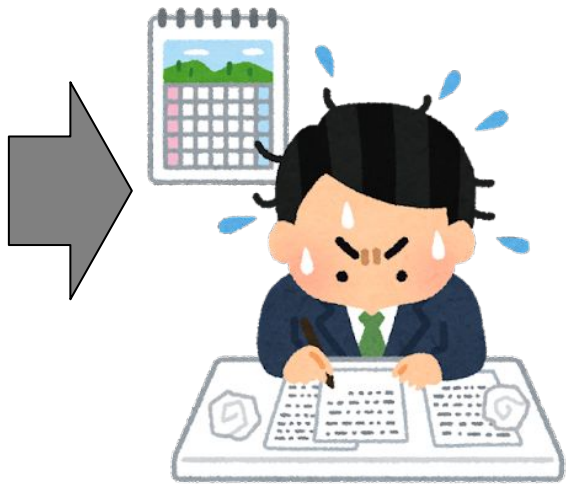
- 修正
- 以前のエントリ名

方　針

- 想　定

①対象：CAPEC・CWEを**利用できる企業**
②本研究で脅威分析が**完全自動化・網羅**されるのではなく、**補助的な役割**である
③検討する知識ベースは**CAPEC・CWE**とする
④質より**量を優先**する





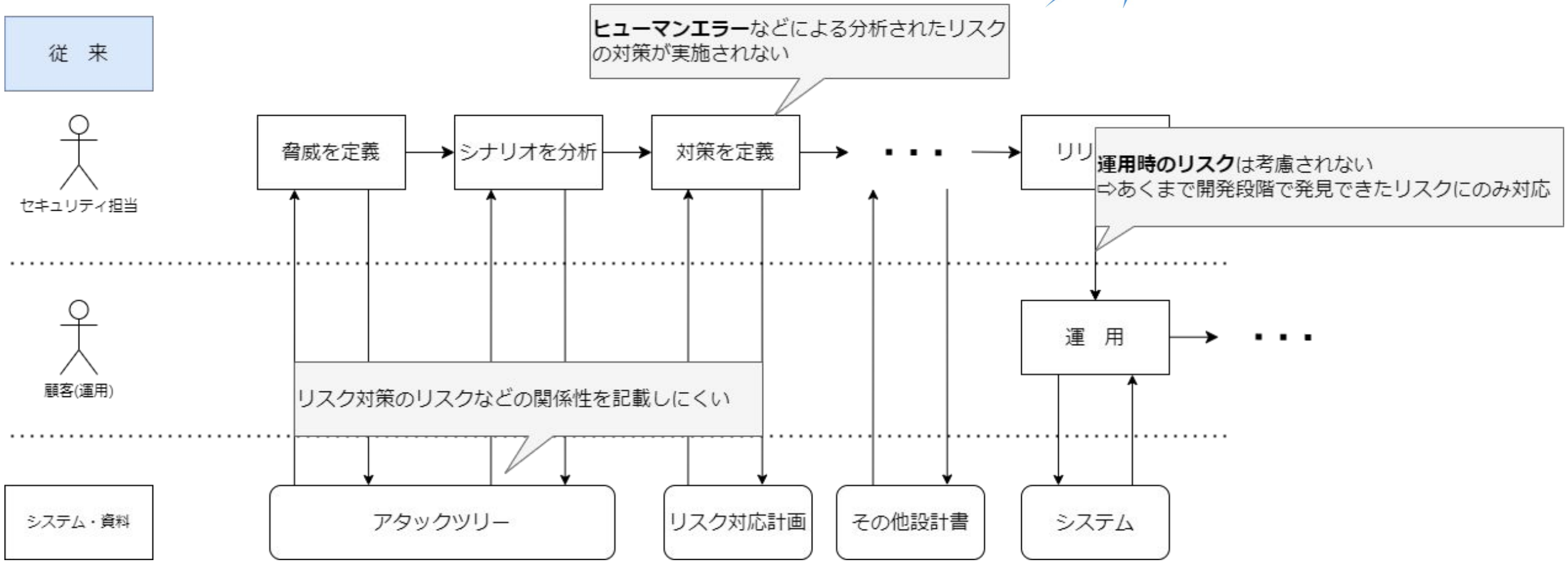
先行研究①(大久保ら

- 問題提起
  DevOpsの**速度**を追求する開発と**網羅性**を重視するセキュリティに**ギャップがある**
  十分な分析をしないでリリースするのは**致命的な事案**を引き起こす可能性がある

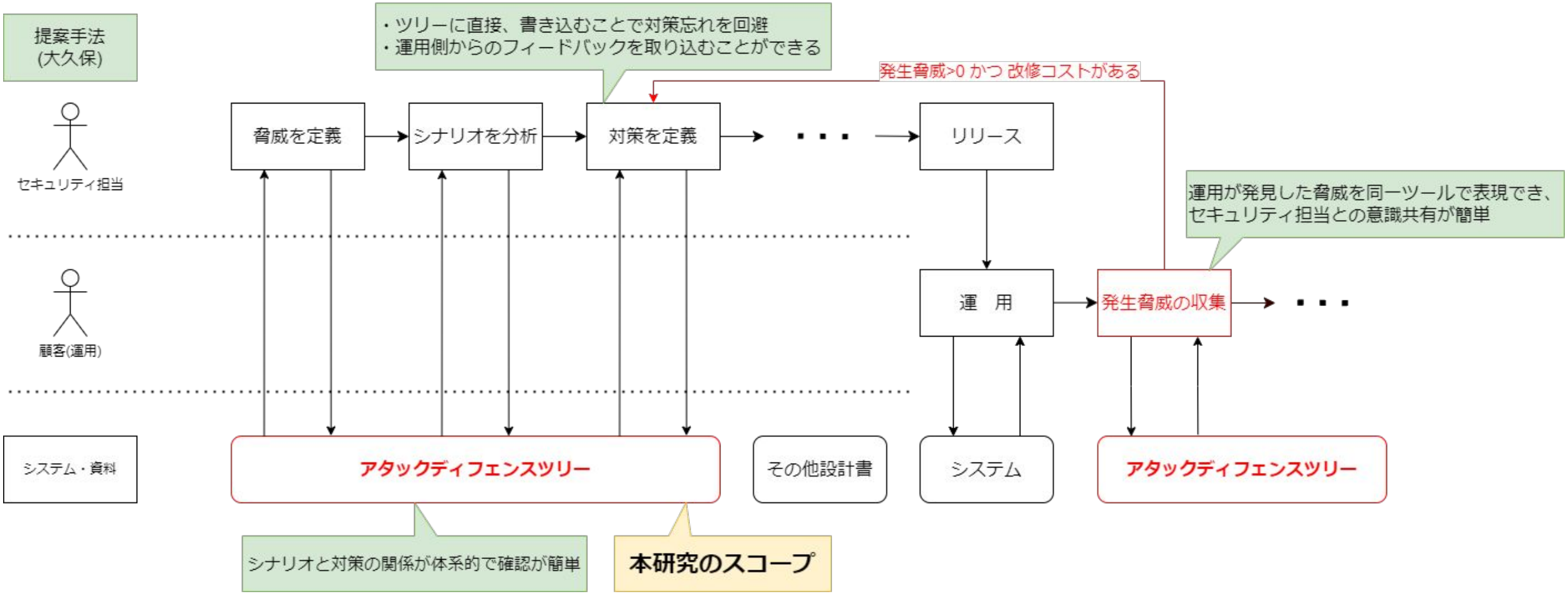
- 提案手法
  ADツリー、**既存知識ベース**を使用することで解決する

- 貢　献
  ①運用中に発生するセキュリティリスクを次の開発要件に**的確にフィードバック**する
  ②既存のデータベースとADツリーを使用した**脅威分析の省力化**

先行研究①(大久保ら


従　来
ヒューマンエラーなどによる分析されたリスクの対策が実施されない
セキュリティ担当
脅威を定義
シナリオを分析
対策を定義
・・・
リリ
運用時のリスクは考慮されない
⇨あくまで開発段階で発見できたリスクにのみ対応
運　用
・・・
顧客(運用)
リスク対策のリスクなどの関係性を記載しにくい
システム・資料
アタックツリー
リスク対応計画
その他設計書
システム

先行研究①(大久保ら



CAPEC･CWE･ATTACKを補助資料として使用することを推奨

先行研究②(大矢

- 問題定義 FRAT & RATTATA
  アタックツリーの効率性の問題は再利用性が乏しいからなのでは？

- 提案手法
  過去・CAPECから生成したツリーを利用できるツールとフレームワークを作成する

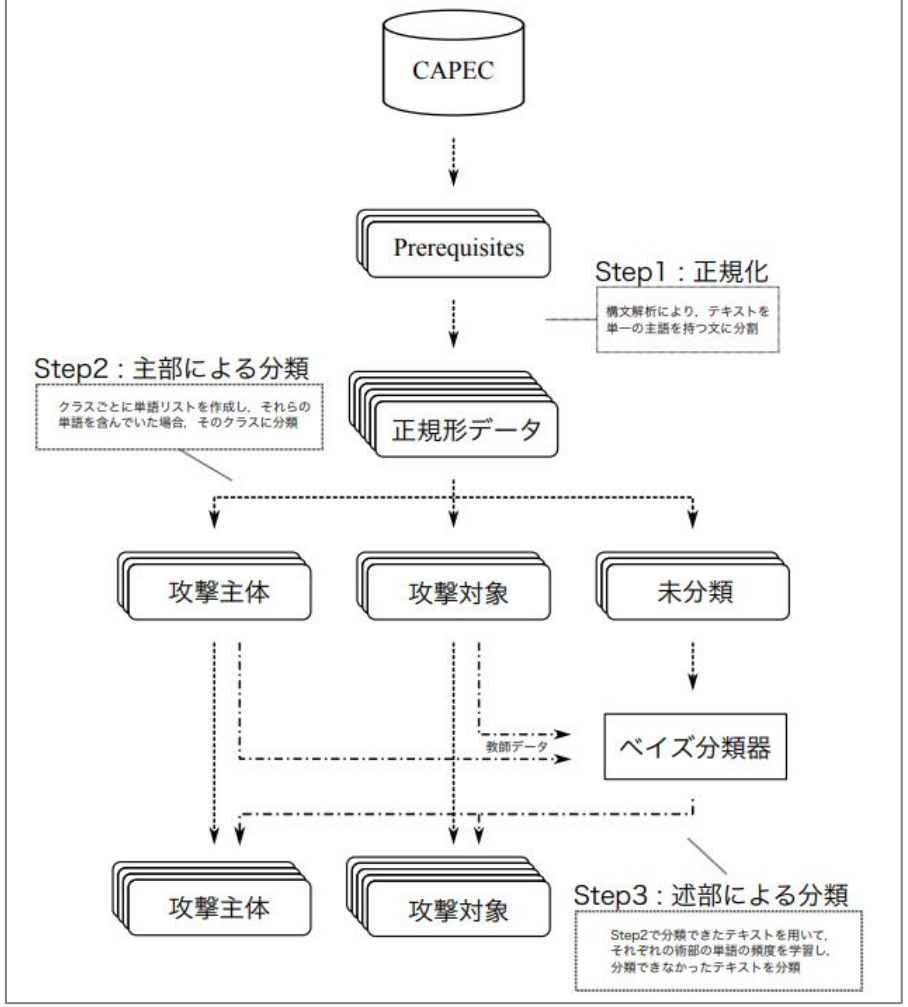
- 貢　献
  ①アタックツリーの**再利用フレームワーク**の提案・開発した
  ②パーツ作成及び更新にかかるコストを**自動化**により**削減**した

CAPECからアタックツリーに**自動分類**するフレームワーク：**FRAT**(フラット
アタックツリー**作成ツール**：**RATTATA**(ラッタッタ

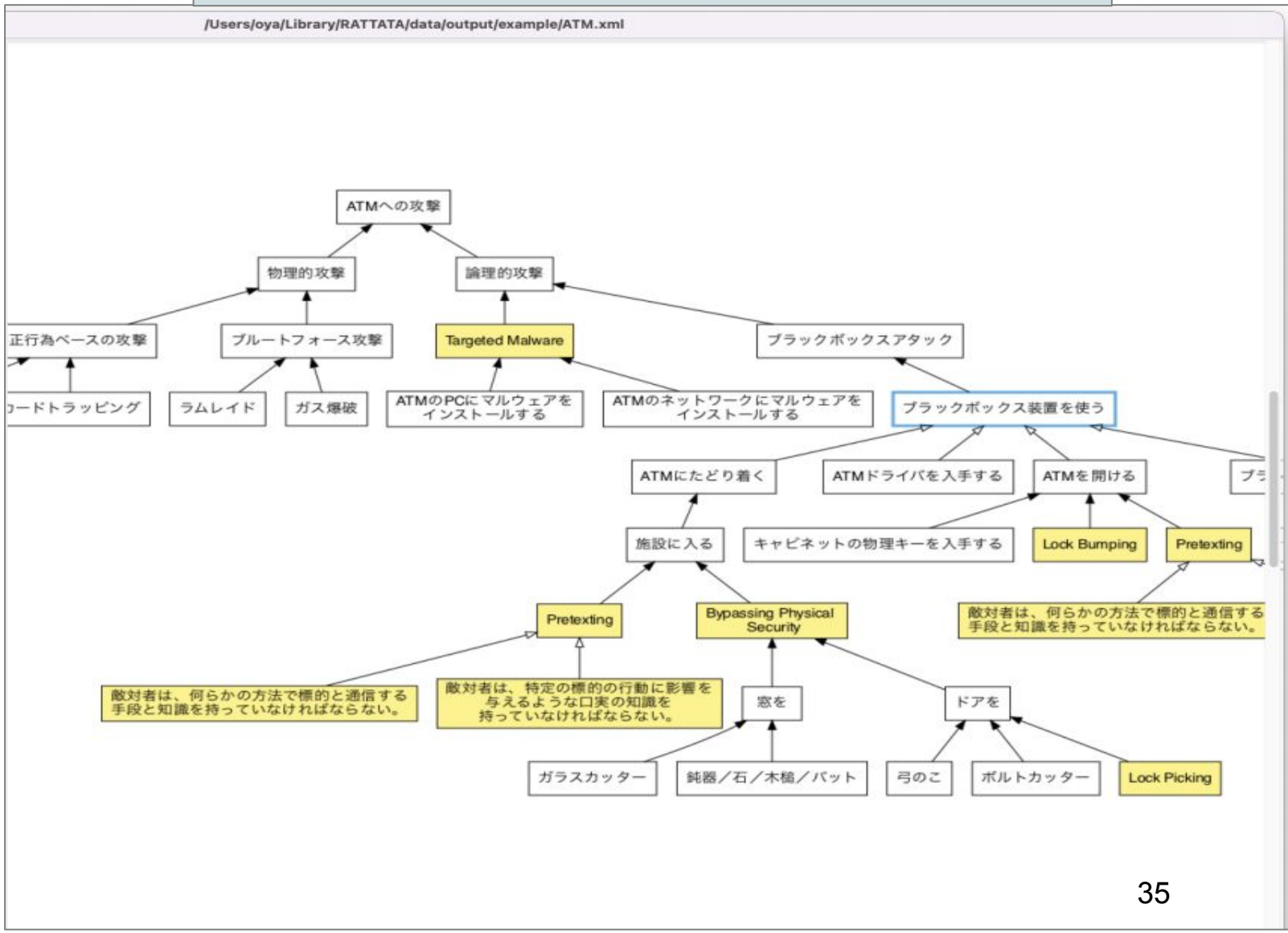
先行研究②(大矢 FRAT & RATTATA

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

自動分類フレームワーク(FRAT



再利用のためのツールを開発(RATTATA



失敗の理由

例：CAPEC-42 MIME Conversion

For example, a sendmail.cf file with these changes applied should look simi

```
Mlocal, P=/usr/libexec/mail.local, F=lsDFMAw5:/|@qrmn, S=10/30, R=20/40,
  T=DNS/RFC822/X-Unix,
  A=mail -d $u

Mprog, P=/bin/sh, F=lsDFMoqeu, S=10/30, R=20/40,
  D=$z:/,
  T=X-Unix,
  A=sh -c $u
```

This can be achieved for the "Mlocal" and "Mprog" Mailers by modifying the

```
define(`LOCAL_MAILER_FLAGS',
  ifdef(`LOCAL_MAILER_FLAGS',
    `translit(LOCAL_MAILER_FLAGS, `9')',
    `rmn'))

define(`LOCAL_SHELL_FLAGS',
  ifdef(`LOCAL_SHELL_FLAGS',
    `translit(LOCAL_SHELL_FLAGS, `9')',
    `eu'))
```

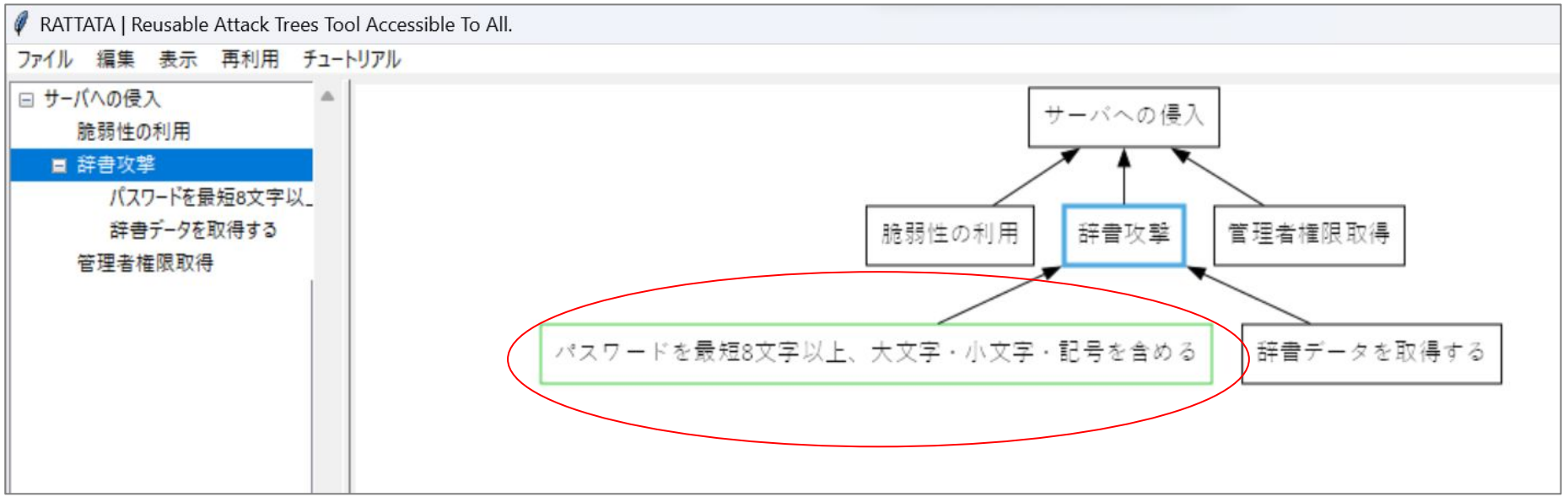
```
<Raw>Mlocal, P=/usr/libexec/mail.local, F=lsDFMAw5:/|@
      T=DNS/RFC822/X-Unix,
      D=$z:/,</Raw>
</Mitigation>
<Mitigation>
  <Raw>This can be achieved for the &quot;Mlocal&quot; a
</Mitigation>
<Mitigation>
  <Raw>define(`LOCAL_MAILER_FLAGS',
      ifdef(`LOCAL_MAILER_FLAGS',
      `translit(LOCAL_MAILER_FLAGS, `9')',
      ifdef(`LOCAL_SHELL_FLAGS',
      `translit(LOCAL_SHELL_FLAGS, `9')',</Raw>
</Mitigation>
```

```
<xhtml:br/>define(`LOCAL_SHELL_FLAGS',
<xhtml:div style="margin-left:10px;">ifdef(`LOCAL_SHELL_FLAGS',
  <xhtml:div style="margin-left:10px;">`translit(LOCAL_SHELL_FLAGS, `9')',
  <xhtml:br/>`eu'))
  </xhtml:div>
</xhtml:div>
```

・失敗の規則性
タブで挟まれていない
単独で完結するタブで記述されている

進　捗（RATTATA

- 自作防御ノードは実装できた



- 検索エンジンの改修を6/1までに実施

他の研究

- 問題点・原因

CAPECから**ツリーを生成**するのは**ユーザ**のため、データの**更新の必要**があり面倒
⇨個人使用が対象であったため

属人化の促進と**再利用性**が落ちている
⇨個人利用と再利用性に矛盾がある

自然言語処理に**新技術**が来た際に**プログラミング知識**がないと更新できない
⇨ツールを作成した新規問題点

多くの研究は手法を提案したのみで販売方法までの検討には至っていない（？

- 必要な事

①根本的な大矢提案の業務フローを変更する（個人
②社内使用のベストプラクティスを提案する（社内
③OSSなどにして広く公開する（世界、業界
でメリット・デメリットを検討する

大矢のフローおよびユーザインターフェースの改修が必要な点を洗い出す

検証（DevOps

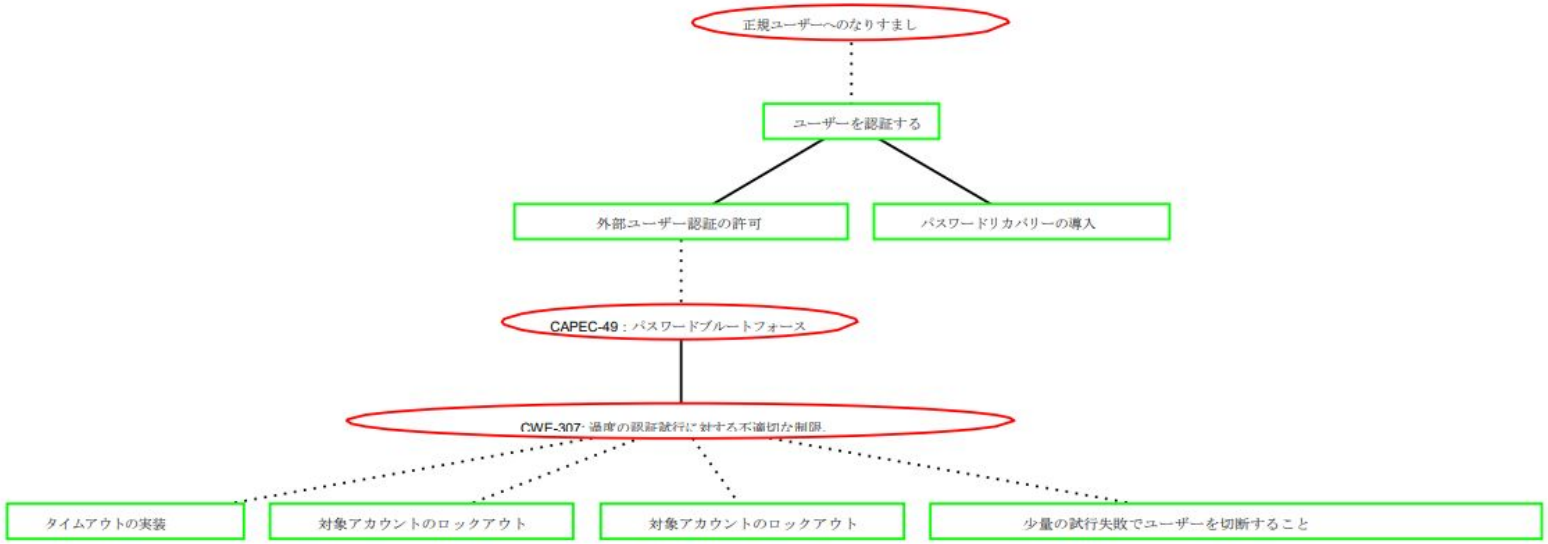
情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

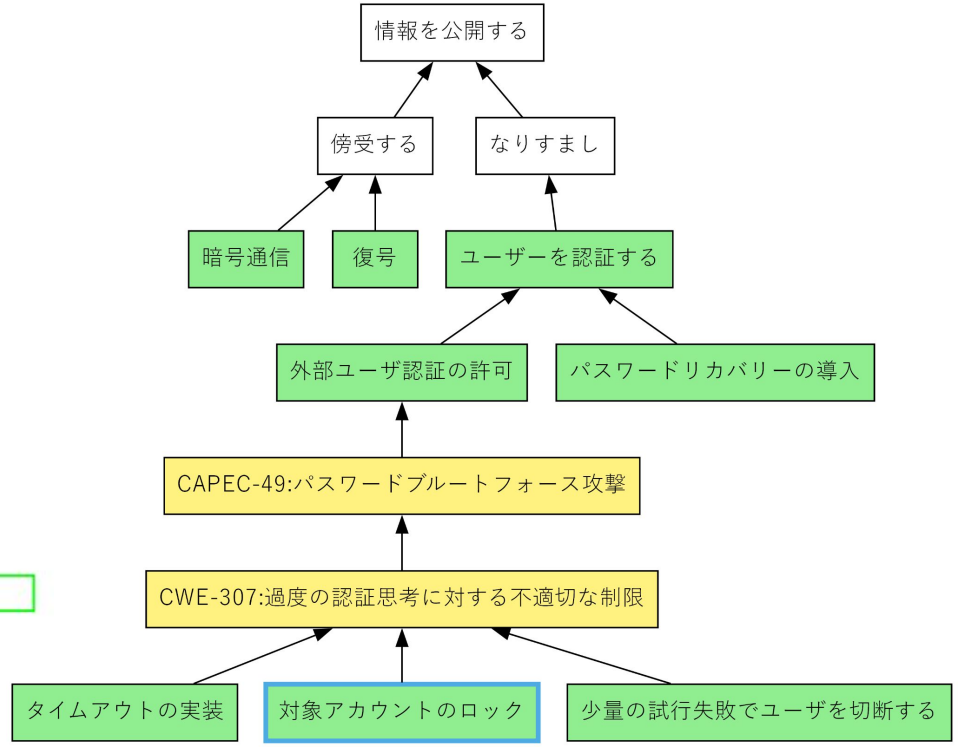
- 想　定

  大久保らが使用したケーススタディを実施
  ⇨大学の学士課程における入試問題の草案を共有する**文書共有システム**

- 項　目

  ①引用ツリーで**ADツリー分析を支援**できるか？⇨**出現した単語**のみでADツリーを作成
  ②**DevOps**で問題なく使用できるか、残課題は？⇨**作成した**ツリーと**大久保ら**のツリーを**比較**

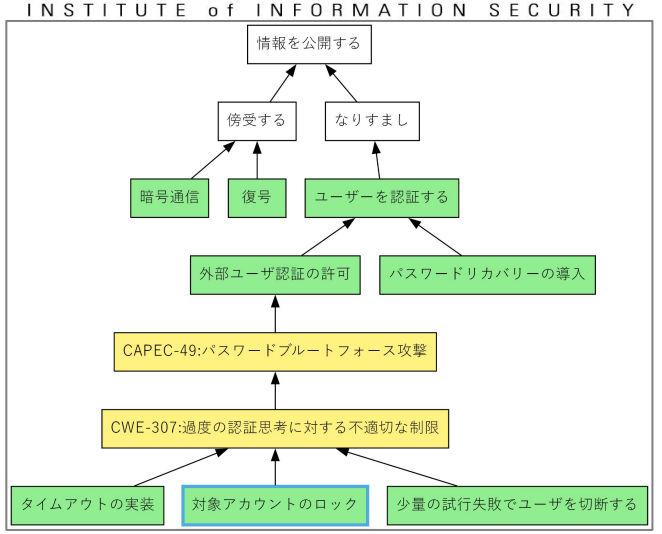


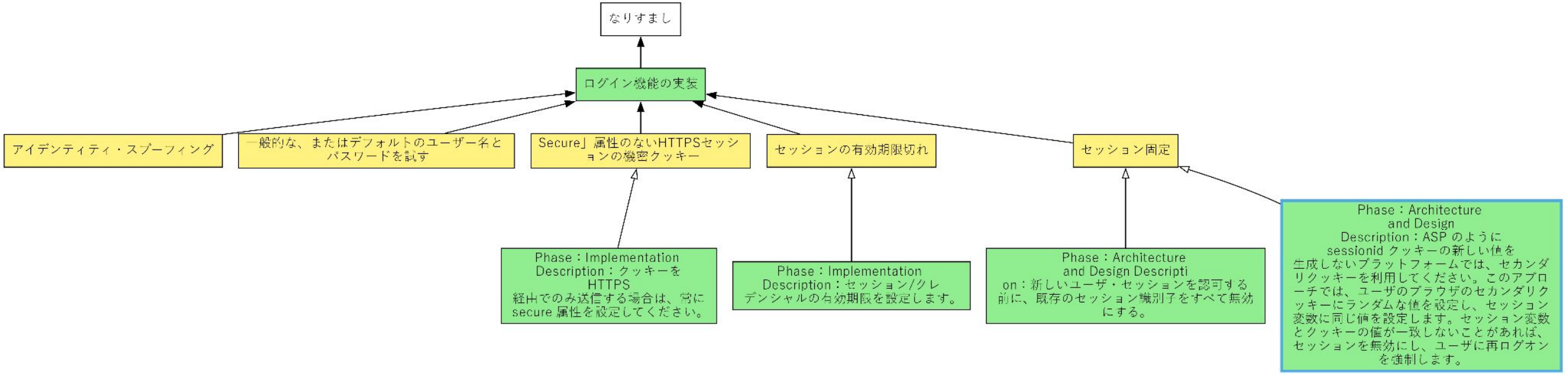


検証（DevOps

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

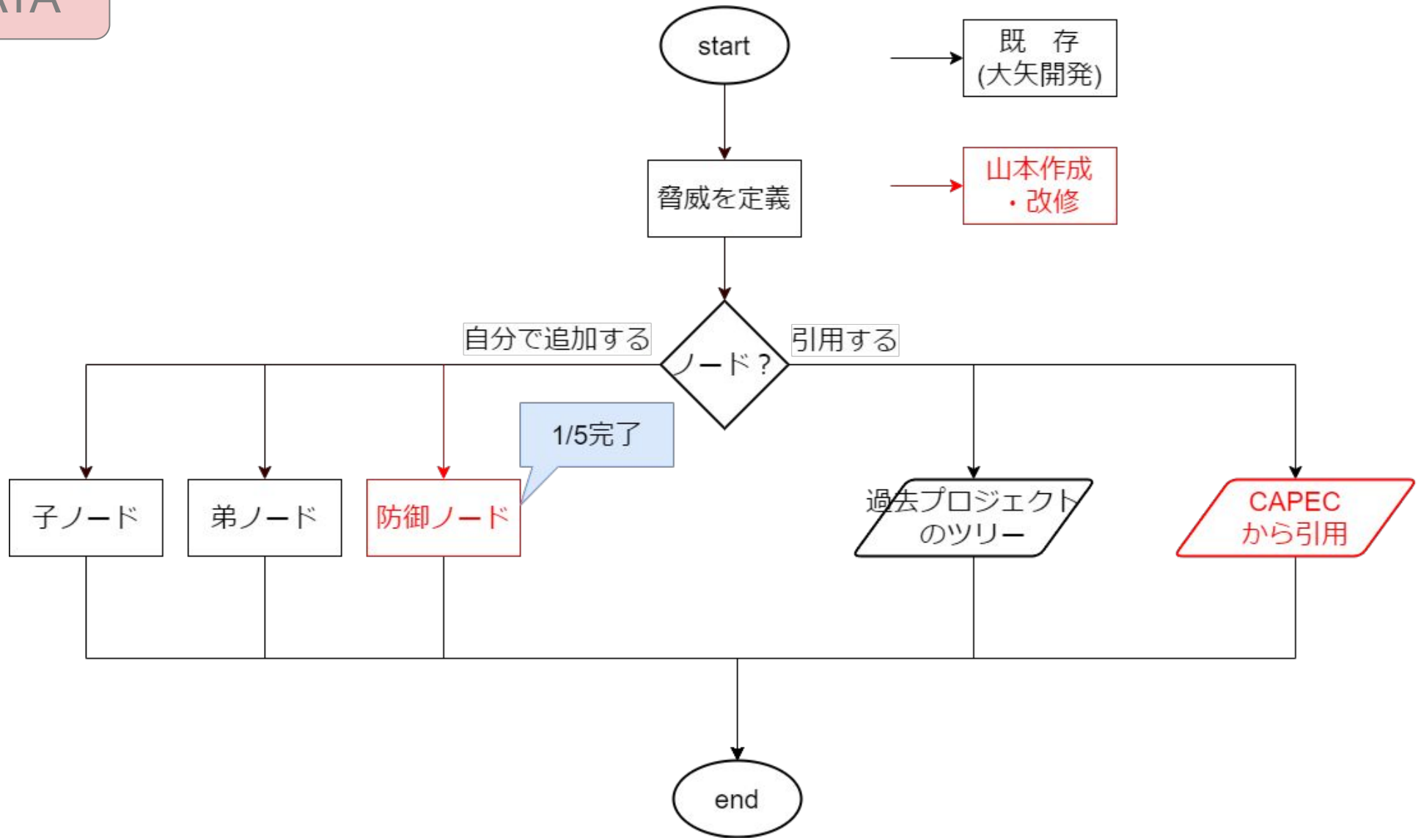
- 結　果（COTTAGE｜大久保ら





改修箇所①

RATTATA



改修箇所②

FRAT

CAPEC

山本作成

CAPEC.xml

攻撃名
事前条件
軽減策
.csv

翻訳
.csv

xml構造体に
再生成

攻撃名
＋
事前条件
.csv

カラムを追加

構文解析

無意味な
レコードを削除

主部で分類

述部で分類

改修箇所③

FRAT

CWE

CWE.xml

脆弱性名
軽減策
.csv

翻訳
.csv

xml構造体に
再生成

parts.xml

完全新規

改修箇所④

FRAT

CAPEC&CWE

CAPEC翻訳後
.csv

CWE翻訳後
.csv

xml構造体に再生成

parts.xml

完全新規

進　捗

- 残ToDo
  ①UIの改修
  ②ツリー構造体の検証
  ③DevOps環境での有用性
  ④論文を書く

検証方法

- 準　備
  ①全般説明
  ②事前アンケート
  ③環境構築

- 練　習
  用語解説（ADツリー、CAPECとは
  UIの操作に関する説明
  作成練習

- 検　証
  準備でわかった経験年数で
  ケースA：RATTATA＋CAPEC手動検索で実施
  ケースB：COTTAGEで実施
  に分けて約30分で作成

検証方法

- 事前アンケート内容

①IT勤務歴
②セキュリティ関連業務経験
③保有IT・セキュリティ資格
④ADツリー作成経験
⑤CAPECについて知っているか

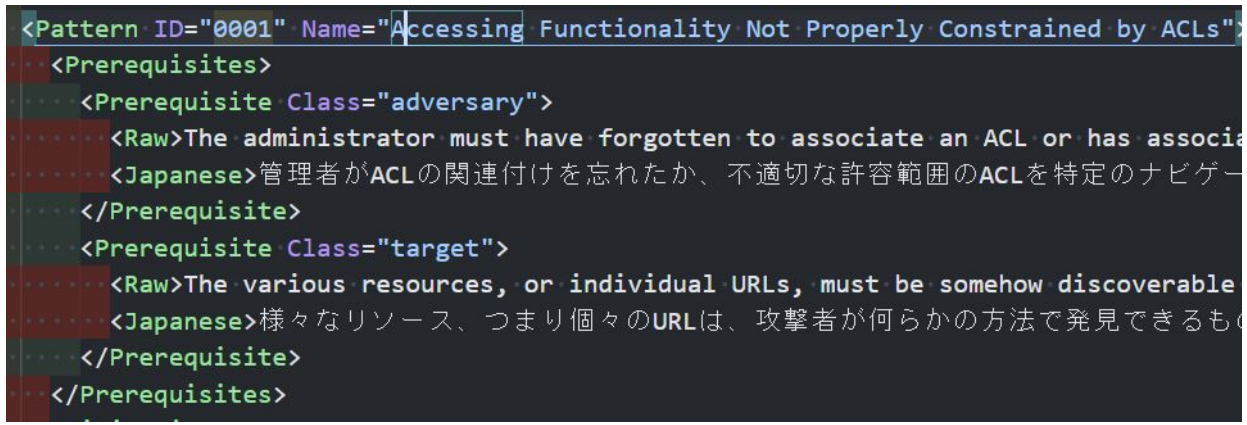
- 事後アンケート内容

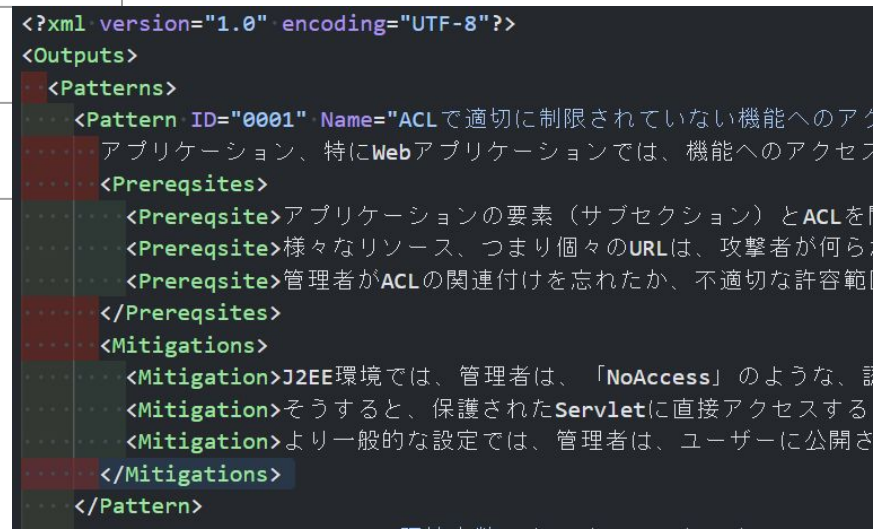
①ツリー検索の内容について良かった点、悪かった点
②ツリー作成の良かった点、悪かった点
③ユーザーインターフェースに関する意見
④アタックツリー作成難度

進　捗(CAPEC

- 差　分

	FRAT	FLATTA
処理時間(正規化)	約3500s	約20s
処理時間(最後)	約8000s	約25s
生成アタックツリー	387	865
生成ADツリー	-	1154
生成csvファイル	11	3
実行ファイル	9	3
正規化コードステップ数	128	65

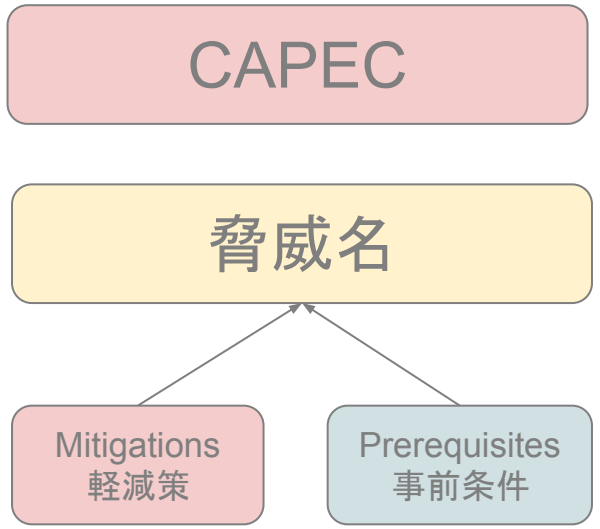




FRATとの比較（CAPEC

- 比　較

	FRAT	COTTAGE
処理時間(事前条件)	約4000s	約20s
処理時間(最後)	約8000s	約25s
事前条件ツリー	387	865
軽減策ツリー	-	1146



{OS : WinOS, CPU : 2.1GHz 4 コア AMD Ryzen 5 3500U, Memory : 8GB}

CAPEC™

攻撃方法
カペック

- コメント

①正規化の段階での精度・速度は**COTTAGE**が有利
FRATは詳細に分解しているツリーが32箇所あるが、16箇所は単独では使用できないツリー
⇨文字化け、意味の通じない箇所での分解

②軽減策ノード化：1146/1148(99.83%)
⇨単純に取り出してるだけ

FRAT：先行研究の分類フレーム

新規作成分（CWEとCAPEC＆CWE

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

- 課　題

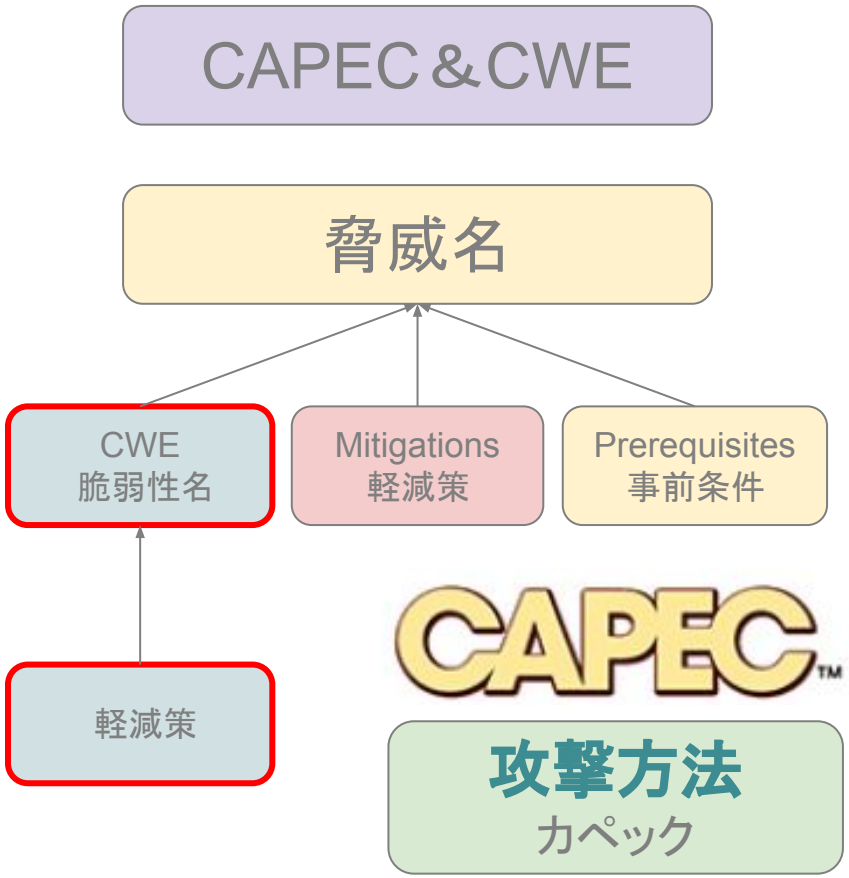
軽減策が長文、この策は有効な策ではない添削が必要
⇨bert-extractive-summarizerという
文章要約のオープンソースを試験的に利用した。

要約条件：①最初のセンテンスを残さない
②英文を250文字以内に要約

- コメント

①文章要約の処理時間：**約3700秒**（前スライドと同スペック
⇨FARTの**全処理時間より早く処理**できるので
ツリー数をとにかく増やしたい場合にも時間的効率もよし

②CWEの軽減策ツリー化　⇨1600/1601(99.99%)



FRAT：先行研究の分類フレーム

まとめ

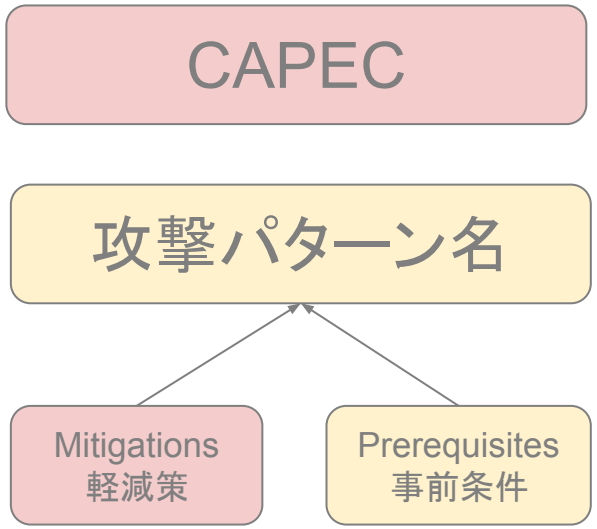
- DevOpsに対応したADツリー作成ツール「COTTAGE」を作成した

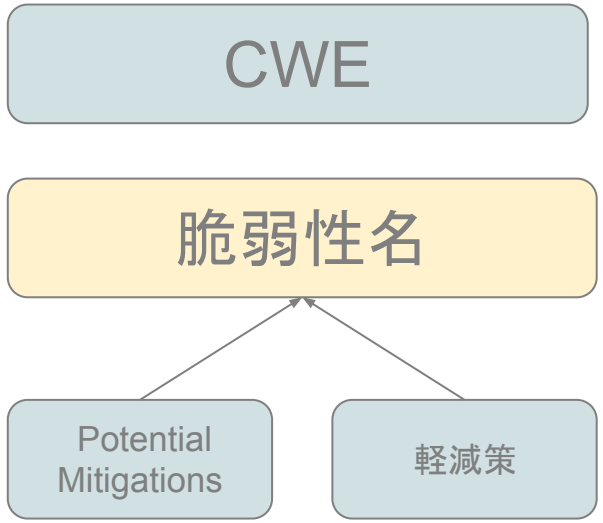
- ツリー生成についての有利は出たが、
  DevOpsでの有用性
  CAPECのツリー引用で脅威分析がどのくらい効率化されるか
  などが未検証の状態であるため、6/7 15:00~17:00y::に検証を実施

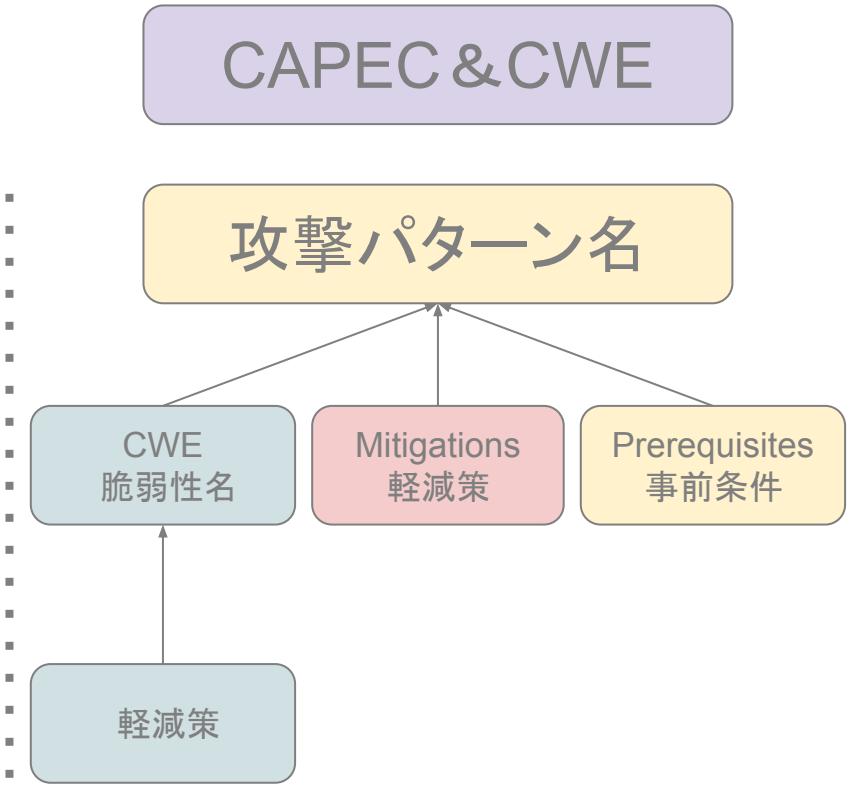
- 5月中にUI部分の作成を完了させる

引用で作成したパターン

- 作成パターン







CAPEC™

攻撃方法
カペック

CWE

脆弱性
シーダブリューイー