

ビルダーのための脅威モデリング

2025/11/21

Amazon Web Service Japan G.K.
Solutions Architect
Wataru Izumi



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.



自己紹介

□ □ ル

エンタープライズ技術本部
ソリューションアーキテクト
主に、人材系の企業を担当

□ 主な経歴

小売系システムのインフラエンジニア
EC サイトの DevOps、SRE リード

□ 好きな AWS サービス

AWS CDK
AWS Lambda
AWS Security Hub



Wataru Izumi / 泉 航

アジェンダ

- 脅威モデリングとは
- 脅威モデリングの効果的な進め方
- AI を活用して効率的に脅威モデリングを進める

脅威モデリングとは

SDLC フェーズごとのセキュリティ施策

※ SDLC = Software Development Life Cycle (ソフトウェア開発ライフサイクル)

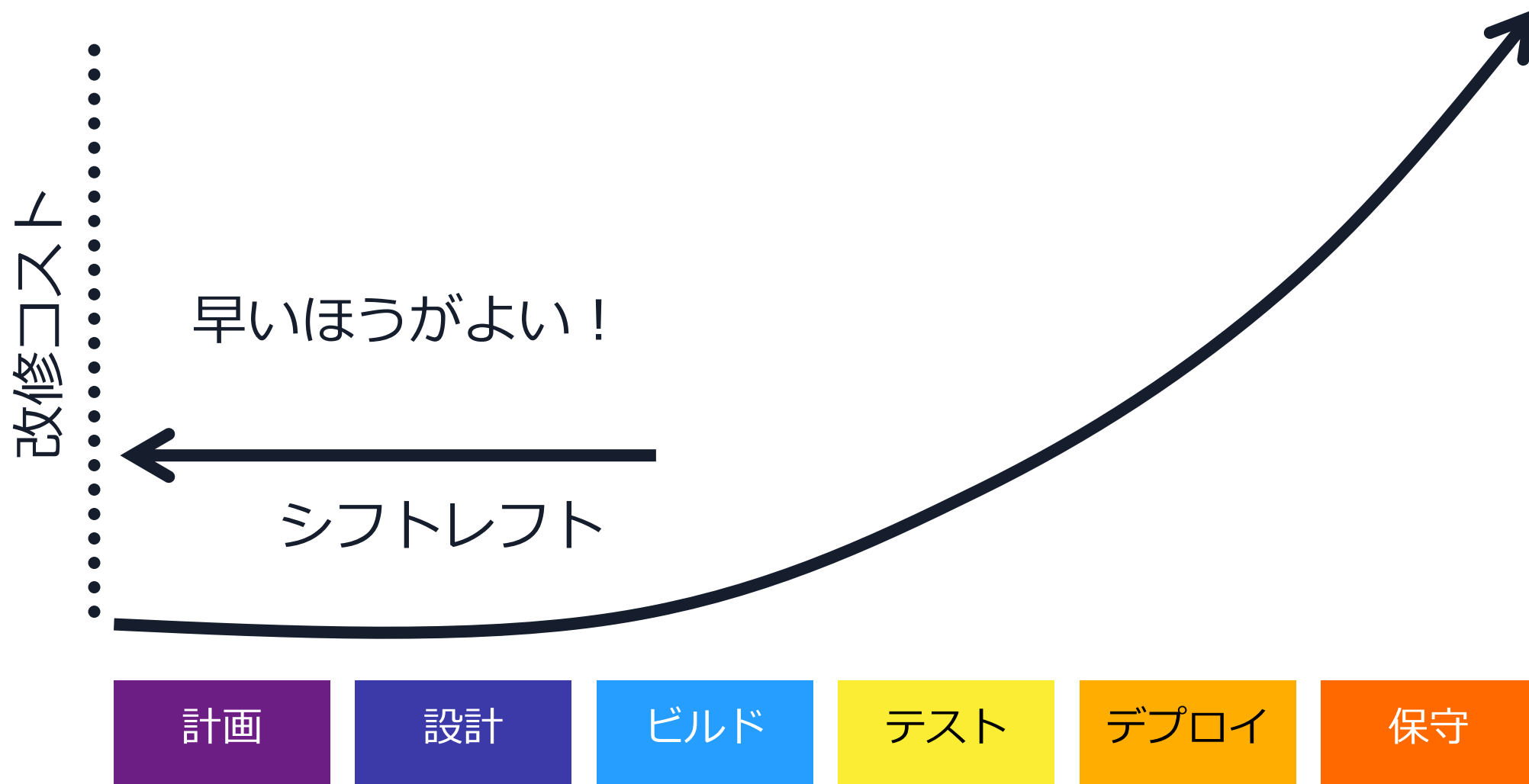


SDLC フェーズごとのセキュリティ施策

※ SDLC = Software Development Life Cycle (ソフトウェア開発ライフサイクル)



不具合修正のコストと “シフトレフト”



脅威モデリングとは

脅威モデリングは、構造的欠陥※や適切な保護手段の欠如などの潜在的な脅威を特定して列挙し、対応策に優先順位を付けるプロセスです。 ※脆弱性、不具合など

(from wikipedia)

脅威モデリングとは

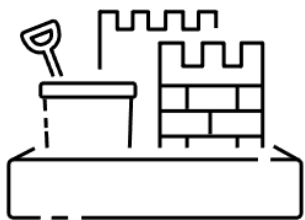
脅威モデリングは、構造的欠陥※や適切な保護手段の欠如などの潜在的な脅威を特定して列挙し、対応策に優先順位を付けるプロセスです。 ※脆弱性、不具合など

脅威モデリングのアウトプット例
(サンプル EC サイトに対して実施したもの)

S - Spoofing (なりすまし)

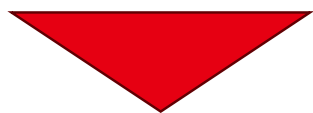
脅威 ID	脅威名	説明	優先度	対策
S-1	ユーザー認証の欠如	認証機能が実装されていない状態で、悪意のあるユーザーが、他人の名前や住所を使って注文することによって不正注文やなりすまし注文が発生し、その結果顧客情報の信頼性や注文データの整合性に悪影響が及ぶ可能性がある	中	<ul style="list-style-type: none">・ユーザー登録・ログイン機能の実装 (JWT/OAuth2.0)・注文時の認証必須化
S-2	注文番号の推測可能性	注文番号が予測可能な実装になっている状態で、悪意のあるユーザーが、注文番号を推測して他人の注文情報にアクセスすることによって注文履歴の不正閲覧が発生し、その結果顧客のプライバシーや個人情報の機密性に悪影響が及ぶ可能性がある	中	<ul style="list-style-type: none">・UUID v4またはULIDの使用・注文照会時の認証必須化

脅威モデリングを行う流れ

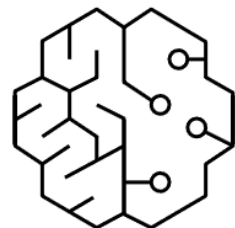


私たちは何に取り組んでいるのか？

What are we working on?



ワークロードの図式化



何が問題になり得るのか？

What can go wrong?

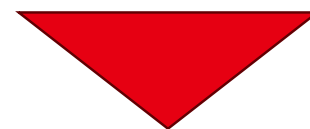


想定される脅威の洗い出し



私たちはそれに対して何をすべきか？

What are we going to do about it?

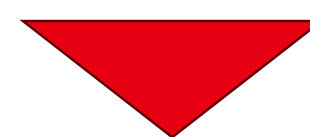


想定される脅威への対応



それは十分に良い仕事だったか？

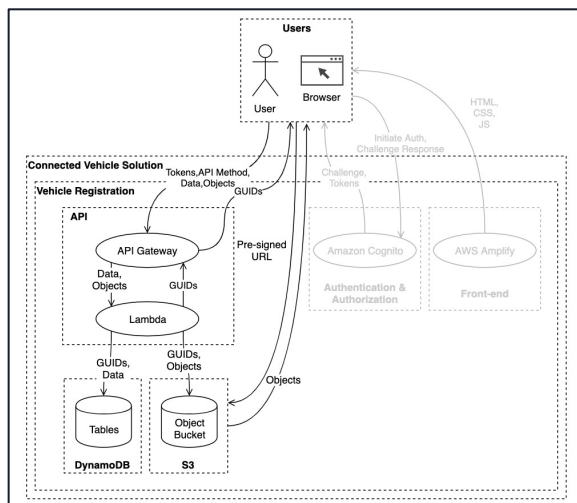
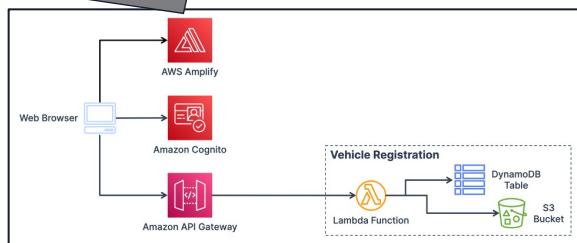
Did we do a good enough job?



対応の評価

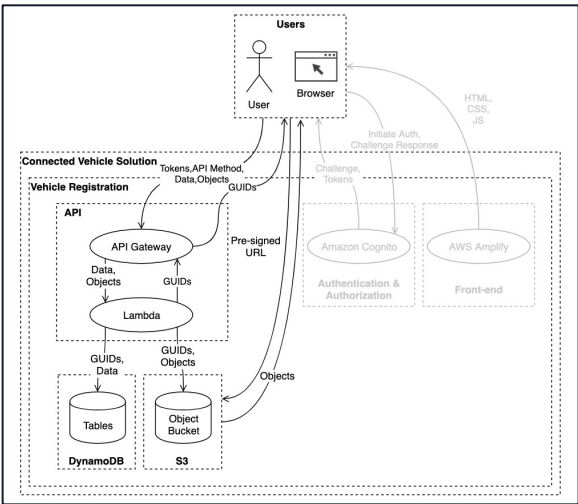
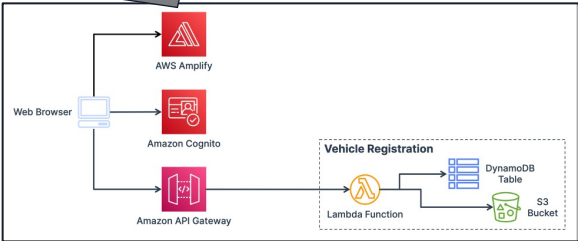
脅威モデリングを行う流れ

① ワークロードの図式化
アーキテクチャ図、データフ
ロー図等を作成



脅威モデリングを行う流れ

① ワークロードの図式化
アーキテクチャ図、データフ
ロー図等を作成



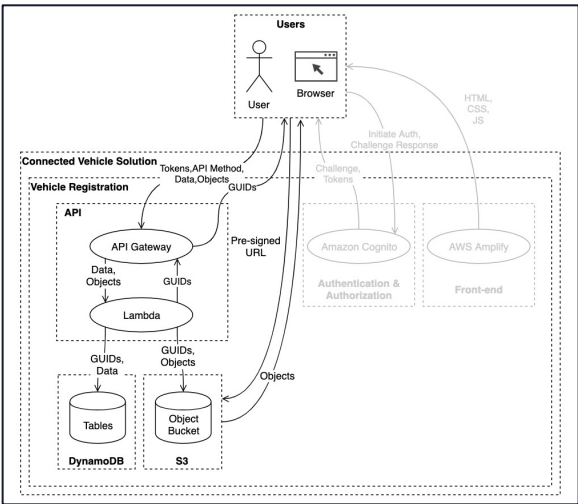
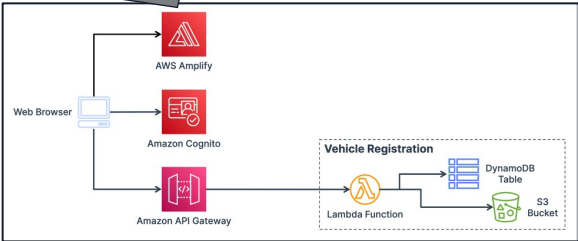
② 想定される脅威の洗い出し
③ 想定される脅威への対応
をチームで議論



優先度	脅威	対策
1	脅威 1	対策 1
2	脅威 2	対策 2
3	脅威 3	対策 3

脅威モデリングを行う流れ

① ワークロードの図式化
アーキテクチャ図、データフ
ロー図等を作成



② 想定される脅威の洗い出し
③ 想定される脅威への対応
をチームで議論



優先度	脅威	対策
1	脅威 1	対策 1
2	脅威 2	対策 2
3	脅威 3	対策 3

④ 対応の評価
一連のサイクルを評価し、流れをブラッシュアップ

脅威モデリングの効果的な進め方

脅威モデリングを効果的に進めるポイント

1. フレームワークの活用
2. 脅威はドメイン観点を中心に考える
3. チームで取り組む
4. 脅威への対応戦略を適切に考える
5. 開発ライフサイクルに脅威モデリングを組み込む

1. フレームワークの活用

脅威モデリングに活用できるフレームワークの例

名称	フォーカスポイント	概要
STRIDE	ソフトウェア開発者	Microsoft 社が開発した方法論で、セキュリティ上の脅威を特定するための記憶術。「Spoofing (なりすまし)、Tampering (改ざん)、Repudiation (否認)、Information disclosure (情報漏洩)、Denial of service (サービス拒否)、Elevation of privilege (権限昇格)」の頭文字。
DREAD	脅威のリスク評価	Microsoft 社の別の方法論で、特定された脅威のリスク評価に使用。DREAD は損害の可能性 (Damage potential)、再現性 (Reproductivity)、悪用可能性 (Exploitability)、影響を受けるユーザ (Affected users)、発見可能性 (Discoverability)。これらの要素はスコア付けされ、その結果は特定された脅威の優先順位付けに使用される。
PASTA	攻撃者視点	「Process for Attack Simulation and Threat Analysis (攻撃のシミュレーションと脅威分析のプロセス)」の略。PASTA は、脅威の特定、列挙、スコア付けを 7 段階のプロセスで行う。
Trike	受容リスク	資産の防御に焦点を当てたリスクベースの方法論。リスク管理の観点から始まり、その文脈で脅威と脆弱性を考える。
VAST	脅威の視覚的表現	Visual (視覚的に)、Agile (柔軟に)、Simple (簡単に)、Threat (脅威)に対応する方法論。アジャイル開発環境に統合されやすく、アクセスしやすいことを目指す。
OCTAVE	組織のリスク評価	オペレーショナルクリティカルな脅威資産と脆弱性の評価。特定の資産に基づく脅威プロファイルの構築、インフラストラクチャの脆弱性の特定、セキュリティ戦略と計画の策定の 3 フェーズがある。

1. フレームワークの活用

脅威モデリングの分析手法として
STRIDE がよく用いられる



1. **Spoofing・なりすまし**：ID のなりすましでは、どのような認証脅威が発生する可能性があるか
2. **Tampering・改ざん**：移動中か保管中かを問わず、データの改ざんにつながるインテグリティの脅威にはどのようなものがあるか
3. **Repudiation・否認**：発生する可能性のあるアクションを拒否する原因となる監査上の脅威の種類（アクションを拒否したり、発生したアクションを隠したりできる）
4. **Information Disclosure・情報漏洩**：どのような機密保持の脅威が発生して情報開示につながる可能性があるか
5. **Denial of Service・サービス拒否**：どのような種類の可用性脅威が発生してサービス拒否の脅威が発生する可能性があるか
6. **Elevation of Privilege・権限昇格**：権限昇格ではどのような権限の脅威が発生する可能性がありますか

2. 脅威はドメイン観点を中心に考える

例えば「EC サイトに対する脅威を考えよう」と言われたら、何を思い浮かべますか？

2. 脅威はドメイン観点を中心に考える

- 「EC サイトに対する脅威」の例
 - SQL インジェクションによる情報漏洩
 - DDoS 攻撃によるサイトダウン
 - 悪意ある内部犯による注文履歴の改竄
 - 不正な大量注文による正規ユーザの注文機会喪失

2. 脅威はドメイン観点を中心に考える

• 「EC サイトに対する脅威」の例

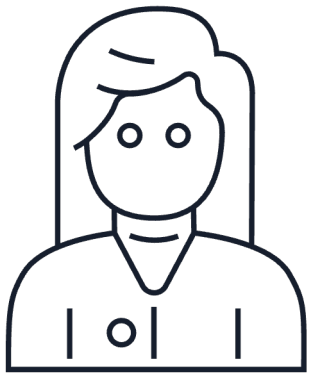
- SQL インジェクションによる情報漏洩
- DDoS 攻撃によるサイトダウン

- 悪意ある内部犯による注文履歴の改竄
- 不正な大量注文による正規ユーザの注文機会喪失

これらは重要なポイントではあるが、
どんなシステムでも言えることで、
汎用的なセキュリティレビューでカバー
できる範囲

これらは EC サイト特有 (=ドメイン特有) の脅威であり、
汎用的なセキュリティレビューではカバーしづらい
→ こういった点をリストアップできると、脅威モデリングを行う価値が高まる

3. チームで取り組む



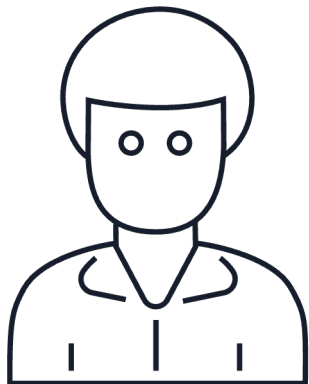
ビジネスペルソナ



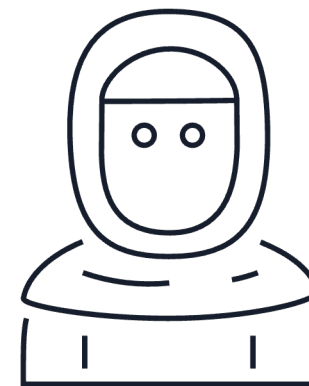
アプリケーションセキュリティ
専門家ペルソナ



敵対的ペルソナ

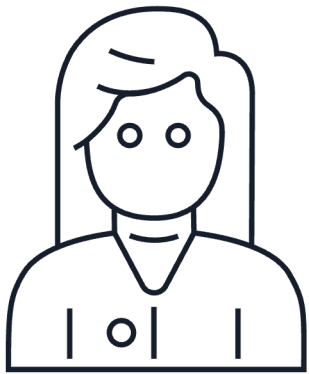


デベロッパーペルソナ



ディフェンダーペルソナ

3. チームで取り組む



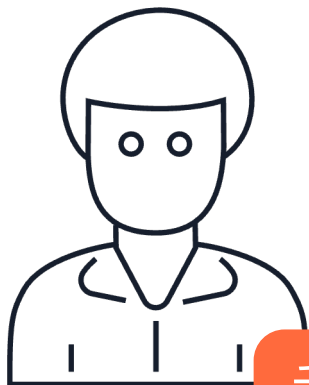
ビジネスペルソナ



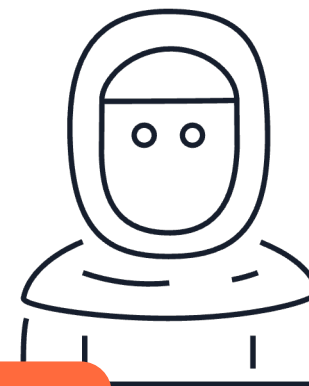
敵対的ペルソナ



アプリケーションセキュリティ
専門家ペルソナ



デベロッパーペ



ンダーペルソナ

チーム全体でさまざまなペルソナを想定することで、
多様な観点で脅威を洗い出せることが見込める

4. 脅威への対応戦略を適切に考える

- よくある誤解



脅威を洗い出したからには、全部対応せねば。。。！

4. 脅威への対応戦略を適切に考える

- よくある誤解



脅威を洗い出したからには、全部対応せねば。。。！

- 現実的な脅威への向き合い方



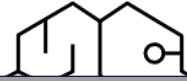
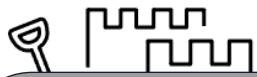
各脅威のリスク分析を行い、適切な対応戦略を考えよう！

4. 脅威への対応戦略を適切に考える

リスクへの対応戦略

対応戦略	対策内容	対策例
1. 回避	<u>リスクを発生させないように変更を加える</u> リスクの可能性をゼロにするか、その重大性を無関係にする。	実装方法を完全に変更するなど。
2. 低減	<u>許容可能なレベルまでリスクを小さくする</u> 脆弱性の可能性を低くするように設計された手法 (セキュアな設計、特定のプログラミング言語、APIなど) を使用するなど。	認証情報の有効期間を短くすることで、認証情報が漏洩して悪用された場合の影響を低減することができる。
3. 移転	<u>リスクを第三者に転嫁する</u> サイバー保険に加入したり別の第三者の管理するコンポーネントに変更するなど。	AWS のマネージドサービスを利用することで、システムリスクの責任の一部を AWS に転嫁したものとも考えることもできる。
4. 受容	<u>意図的にリスクを受け入れる</u> リスクを緩和するためにかかるコストがリスクを悪用された場合の代償よりも大きい場合などは、あえてリスクを許容することもある。	-

5. 開発ライフサイクルに脅威モデリングを組み込む



1回やって終わりでは不十分
開発プロセスに組み込み、メカニズムとして
脅威モデリングを実践できる体制にすること
が重要

れに対し
べきか？



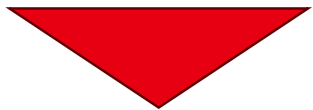
それは十分に良い
仕事だったか？

What are we
working on?

What can
go wrong?

What are we
going to do about
it?

Did we do a good
enough job?



ワークロードの図式化

想定される脅威の洗い出し

想定される脅威への対応

対応の評価

5. 開発ライフサイクルに脅威モデリングを組み込む

効果的な脅威モデリング実施のタイミング

- 新しいサービスを開発する時
- 既存サービスに新しい機能を追加する時
- 既存サービスの仕様を大きく変更する時
- . . . など

※ ちょっとした機能変更やバグ改修時など、サービスの設計が大きく変わらないケースでは、必ずしも毎回脅威モデリングを実施する必要はない

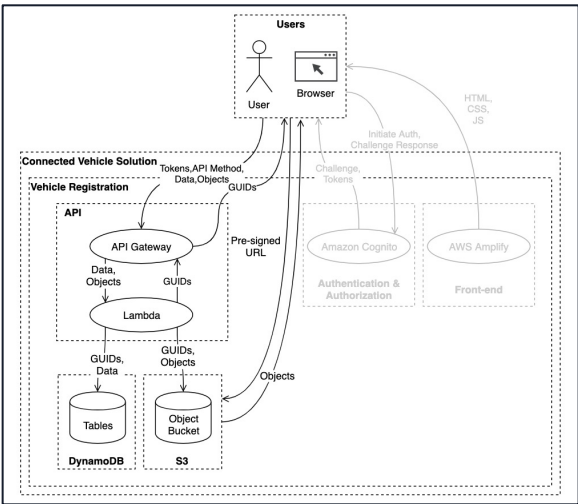
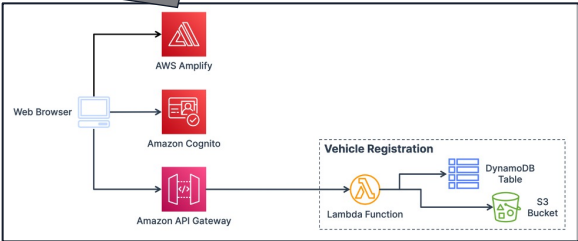
脅威モデリングを効果的に進めるポイント（再掲）

1. フレームワークの活用
2. 脅威はドメイン観点を中心に考える
3. チームで取り組む
4. 脅威への対応戦略を適切に考える
5. 開発ライフサイクルに脅威モデリングを組み込む

AI を活用して効率的に 脅威モデリングを進める

脅威モデリングを行う流れ（再掲）

① ワークロードの図式化
アーキテクチャ図、データフ
ロー図等を作成



② 想定される脅威の洗い出し
③ 想定される脅威への対応
をチームで議論



優先度	脅威	対策
1	脅威 1	対策 1
2	脅威 2	対策 2
3	脅威 3	対策 3

④ 対応の評価
一連のサイクルを評価し、流れをブラッシュアップ

AI を活用した効率化の例

- ① ワークロードの図式化

→ 仕様書やソースコードをインプットに、AI でデータフロー図を作成

- ② 想定される脅威の洗い出し

→ 仕様書やデータフロー図をインプットに、AI で脅威をリストアップ

- ③ 想定される脅威への対応

→ 脅威リストをインプットに、AI で対策案をリストアップ

- ④ 対応の評価

→ 対策と結果のデータをインプットに、AI で評価をして改善点をリストアップ

AI を活用した効率化の例

- ① ワークロードの図式化

→ 仕様書やソースコードをインプットに、AI でデータフロー図を作成

- ② 想定される脅威の洗い出し

→ 仕様書やデータフロー図をインプットに、AI で脅威をリストアップ

- ③ 想定される脅威への対応

→ 脅威リストをインプットに、AI で対策案をリストアップ

- ④ 対応の評価

→ 対策と結果のデータをインプットに、AI で評価をして改善点をリストアップ

AI によるアウトプット例 1 - 脅威リスト



User

本ECサイトアプリケーションに対し、脅威モデリングを実施します。まずは想定される脅威を、STRIDEのフレームワークを活用して洗い出してください。脅威を洗い出す際は、本ECサイトのドメイン要素を考慮した観点を中心に検討するようにしてください。



AI

このECサイトアプリケーションの脅威モデリングを実施します。まず、アプリケーションの構造を確認してから、STRIDEフレームワークに基づいて脅威を洗い出します。
・・・(以下略)

AI によるアウトプット例 1 - 脅威リスト

脅威一覧概要			
STRIDEカテゴリ	脅威ID	脅威名	優先度
S - Spoofing (なりすまし)	S-1	ユーザー認証の欠如	中
	S-2	注文番号の推測可能性	中
T - Tampering (改ざん)	T-1	クライアントサイドでの価格改ざん	高
	T-2	カート内容の操作	高
	T-3	注文詳細の改ざん	中
R - Repudiation (否認)	R-1	注文履歴の欠如	高
	R-2	監査ログの不在	中
I - Information Disclosure (情報漏洩)	I-1	決済情報のクライアント保持	高
	I-2	個人情報のURL露出	中
	I-3	商品在庫情報の露出	低
D - Denial of Service (サービス拒否)	D-1	在庫枯渇攻撃	高
	D-2	大量注文による処理負荷	中
	D-3	カート肥大化攻撃	低
E - Elevation of Privilege (権限昇格)	E-1	管理者機能の欠如による権限概念の不在	低
	E-2	クライアントサイドルーティングの脆弱性	低
合計: 15件の脅威 (高優先度: 5件、中優先度: 6件、低優先度: 4件)			

STRIDE カテゴリごとの脅威を一覧表示

各脅威の詳細表示 (一部抜粋)

S - Spoofing (なりすまし)				
脅威ID	脅威名	説明	優先度	対策
S-1	ユーザー認証の欠如	認証機能が実装されていない状態で、悪意のあるユーザーが、他人の名前や住所を使って注文することによって不正注文やなりすまし注文が発生し、その結果顧客情報の信頼性や注文データの整合性に悪影響が及ぶ可能性がある	中	・ユーザー登録・ログイン機能の実装 (JWT/OAuth2.0) ・注文時の認証必須化
S-2	注文番号の推測可能性	注文番号が予測可能な実装になっている状態で、悪意のあるユーザーが、注文番号を推測して他人の注文情報にアクセスすることによって注文履歴の不正閲覧が発生し、その結果顧客のプライバシーや個人情報の機密性に悪影響が及ぶ可能性がある	中	・UUID v4またはULIDの使用 ・注文照会時の認証必須化

AI によるアウトプット例 2 - 対策サマリ



User

それでは次のステップとして対策検討に移ります。リストアップしてもらった各脅威に対して対策を検討し、先ほど作成いただいた表に対策を追記してください。



AI

各脅威に対する対策を追記しました。対策は以下の観点で検討しています：
・・・(以下略)

AI によるアウトプット例 2 - 対策サマリ

脅威別リスク対応戦略・実装工数・優先度評価					
脅威 ID	リスク 対応戦略	実装工数 (人日)	工数内訳	実装 優先度	理由
T-1	低減	15-20日	バックエンドAPI構築(10日) + 商品マスタDB設計(3日) + 価格検証ロジック(2日) + テスト(3-5日)	最優先	金銭的損失が直接発生。ROI最大。バックエンド基盤は他の脅威対策にも流用可能
I-1	移転	5-8日	決済ゲートウェイ選定(1日) + Stripe/PayPal統合(3-5日) + テスト(2日)	最優先	PCI DSS準拠を外部サービスに移転。法的リスク回避。比較的短時間で実装可能
R-1	低減	10-15日	DB設計(3日) + 注文永続化API(4日) + メール送信機能(2日) + 注文履歴UI(3-4日) + テスト(2日)	最優先	ビジネスの根幹。T-1のバックエンド基盤と並行開発可能
T-2	低減	8-12日	在庫管理API(4日) + バリデーション実装(2日) + 楽観的ロック(2-3日) + テスト(2-3日)	第2優先	T-1のバックエンド基盤完成後に実装。在庫管理の信頼性確保
D-1	低減	6-10日	在庫仮確保機構(3-4日) + レート制限(2日) + CAPTCHA統合(1-2日) + テスト(2日)	第2優先	T-2と密接に関連。同時期に実装することで効率化

各脅威に対する対応戦略と実装工数を考慮し、対応優先度を AI が判断して表示

AI を有効活用するには

- 人間がカバーすべきこと
 - 脅威モデリングとは何なのかの把握
 - 何を対象に、何のために脅威モデリングを行うのかの把握
 - どの脅威への対策を優先して行うかの最終的な判断
- AI を活用するポイント
 - 脅威のリストアップや対策検討など、時間がかかる箇所の効率化
 - 脅威モデリングに必要なインプット情報（設計書等）の準備の効率化

最後に

より理解を深めていただくために

ビルダーのための脅威モデリングワークショップ

- <https://catalog.workshops.aws/threatmodel/ja-JP>
- 原題 'Threat modeling for builders' の日本語訳版

The screenshot shows the AWS Workshop Studio interface. The top bar is dark blue with the 'aws workshop studio' logo. The left sidebar contains a table of contents for the workshop, including sections like 'ワークショップの概要', '脅威モデリングの概要', and 'AWS における脅威モデリング'. Below the sidebar is a 'Content preferences' section with a 'Language' dropdown set to '日本語'. The main content area is titled 'ビルダーのための脅威モデリング' and features a blue box with the text '「ビルダーのための脅威モデリング」 によるこそ'. Below this, there is a paragraph explaining the course content, followed by a bulleted list of topics: 'システムのモデル化', '脅威の特定', and '対策の選択'. At the bottom of the main content area, there are 'Previous' and 'Next' buttons.

Thank you!

