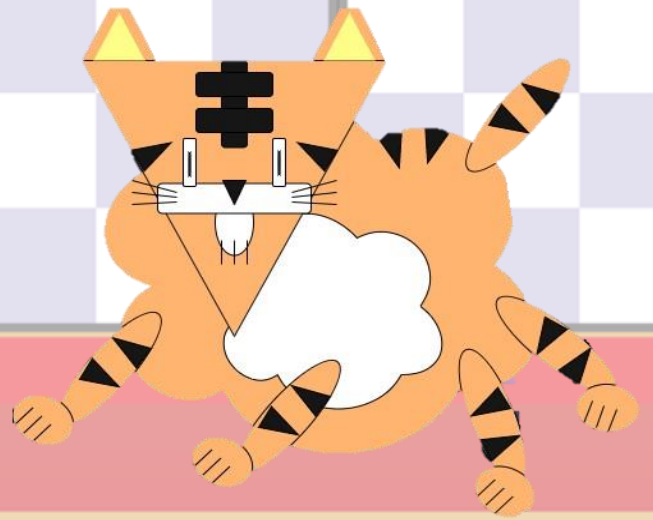
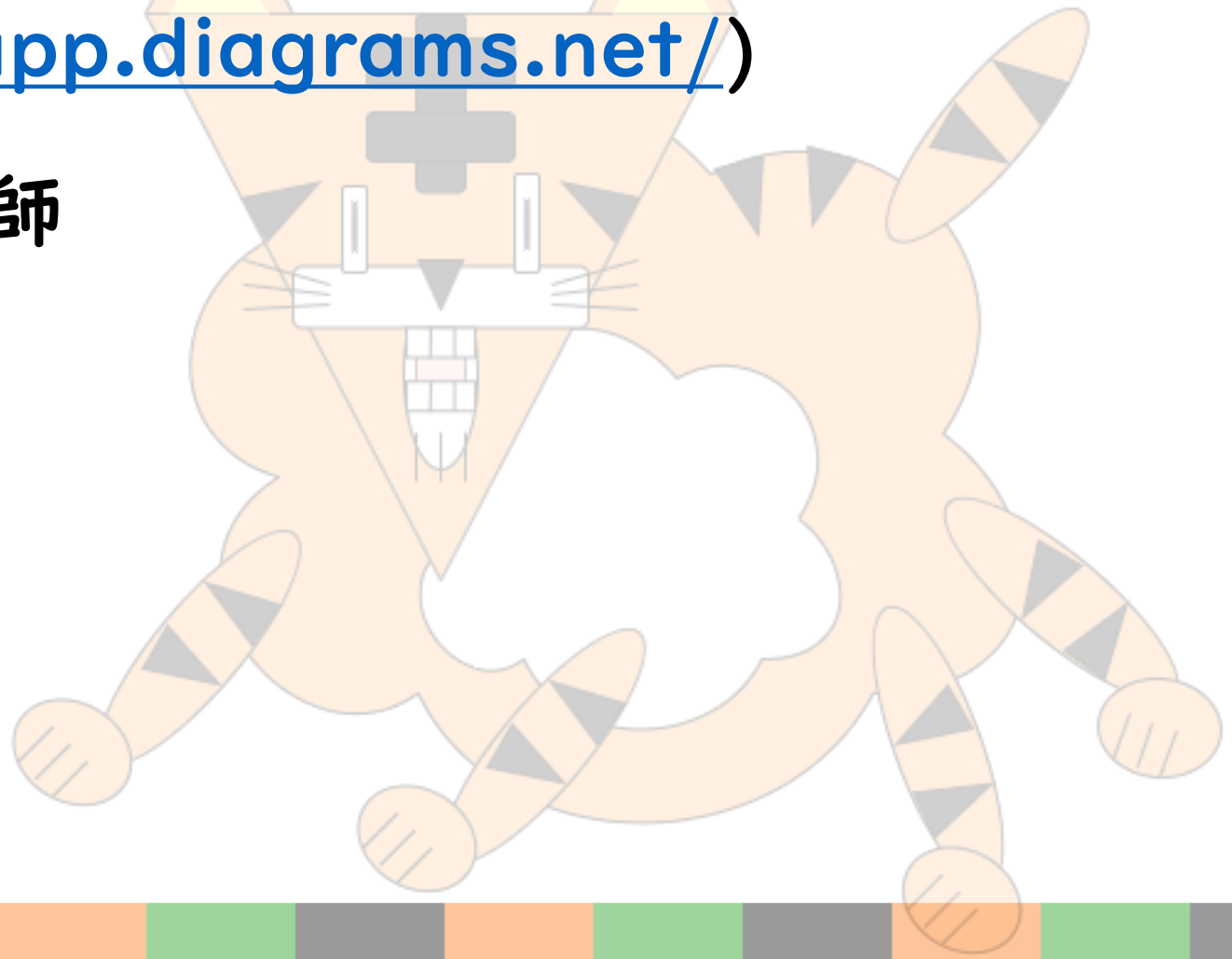
Jasst関西'23
セキュリ亭ジャストラ

よりよいテストを思索するための、組み込みセキュリティテスト入門



講演者紹介

- 名前：セキュリ亭 ジャストラ (JaSST‘23 Kansai実行委員アバター)
- 出身：Draw.io （https://app.diagrams.net/）
- 職業：セキュリティテスト講談師



本日の講演内容

1. 組み込みソフトウェアとセキュリティ

2. セキュリティテストのプロセス

3. セキュリティテストの技法

4. セキュリティテストのベストプラクティス

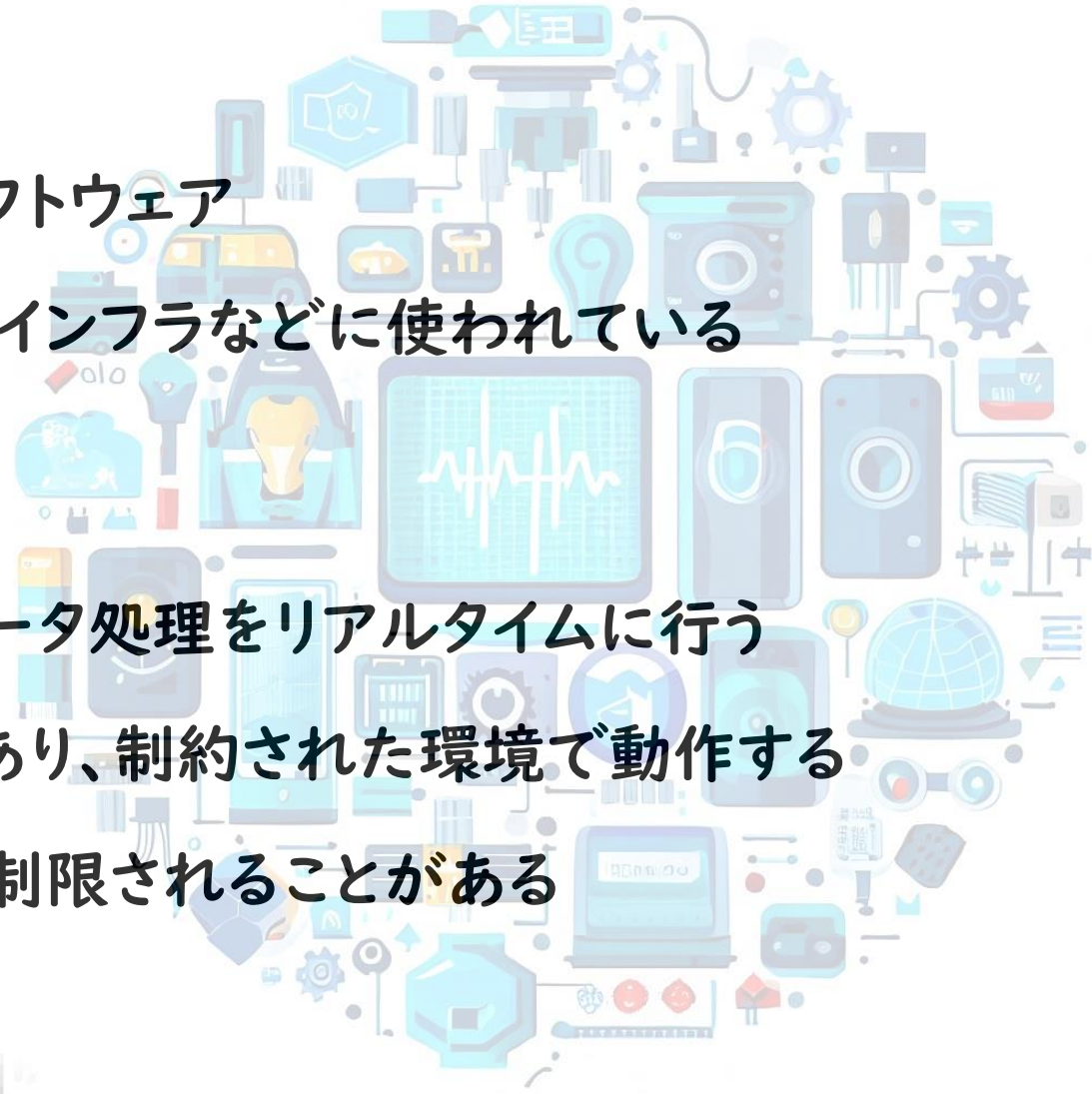
5. まとめ

組み込みソフトウェアとセキュリティ

組み込みソフトウェアの概要

•組み込みソフトウェアとは？

- 様々な電子機器やシステムに埋め込まれたソフトウェア
- 家電製品、自動車、産業機器、医療機器、交通インフラなどに使われている



•組み込みソフトウェアの特徴

- **リアルタイム処理**: ハードウェアへの制御やデータ処理をリアルタイムに行う
- **制約された環境**: メモリや処理能力の制約があり、制約された環境で動作する
- **物理的な制約**: 機器への物理的なアクセスが制限されることがある

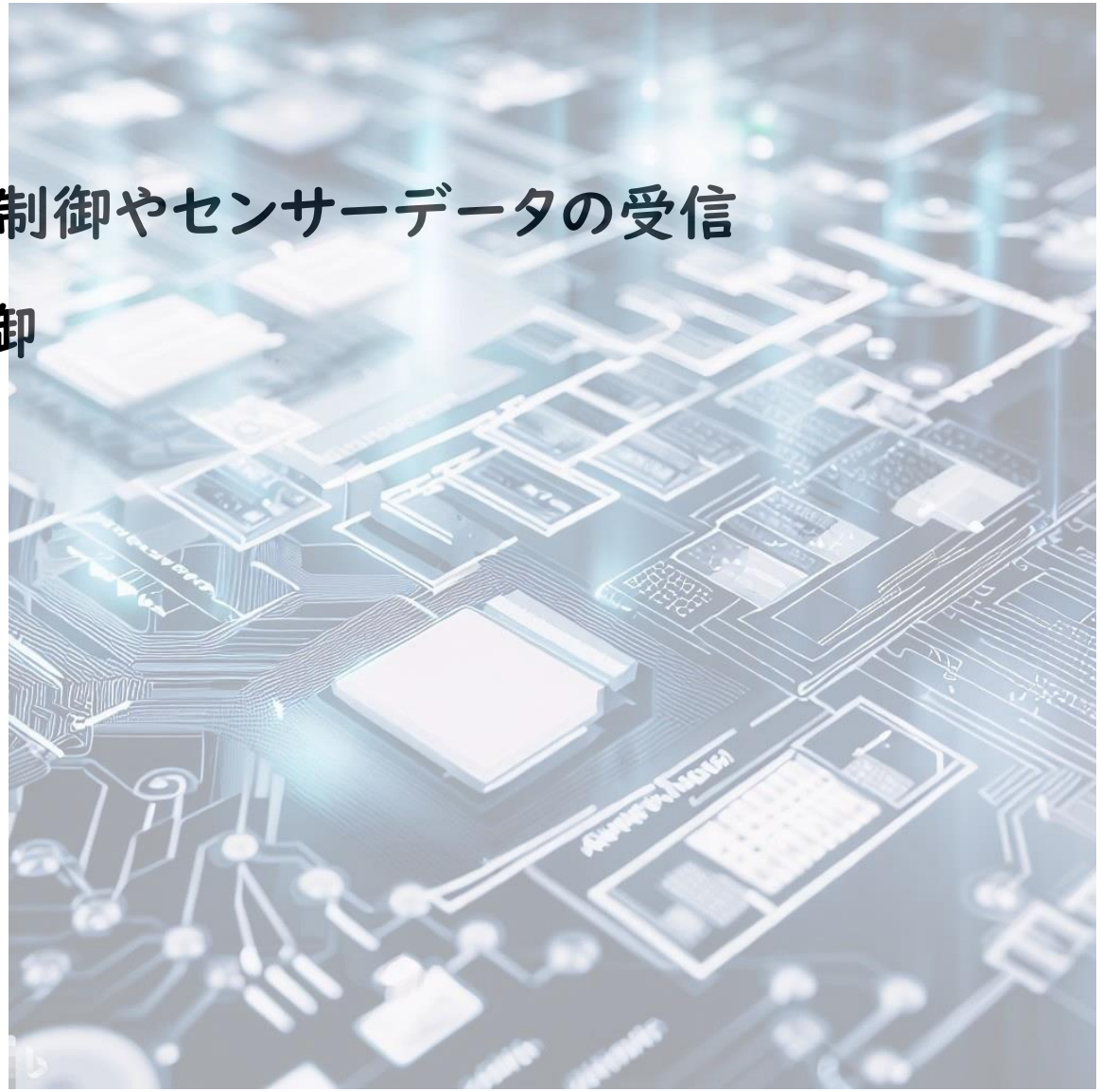
組み込みソフトウェアの役割と機能

•組み込みソフトウェアの役割

- ハードウェアとのインタフェース: ハードウェア制御やセンサーデータの受信
- システムの制御: 機器やシステムの動作や制御
- データ処理: センサーデータの解析や処理

•組み込みソフトウェアの機能

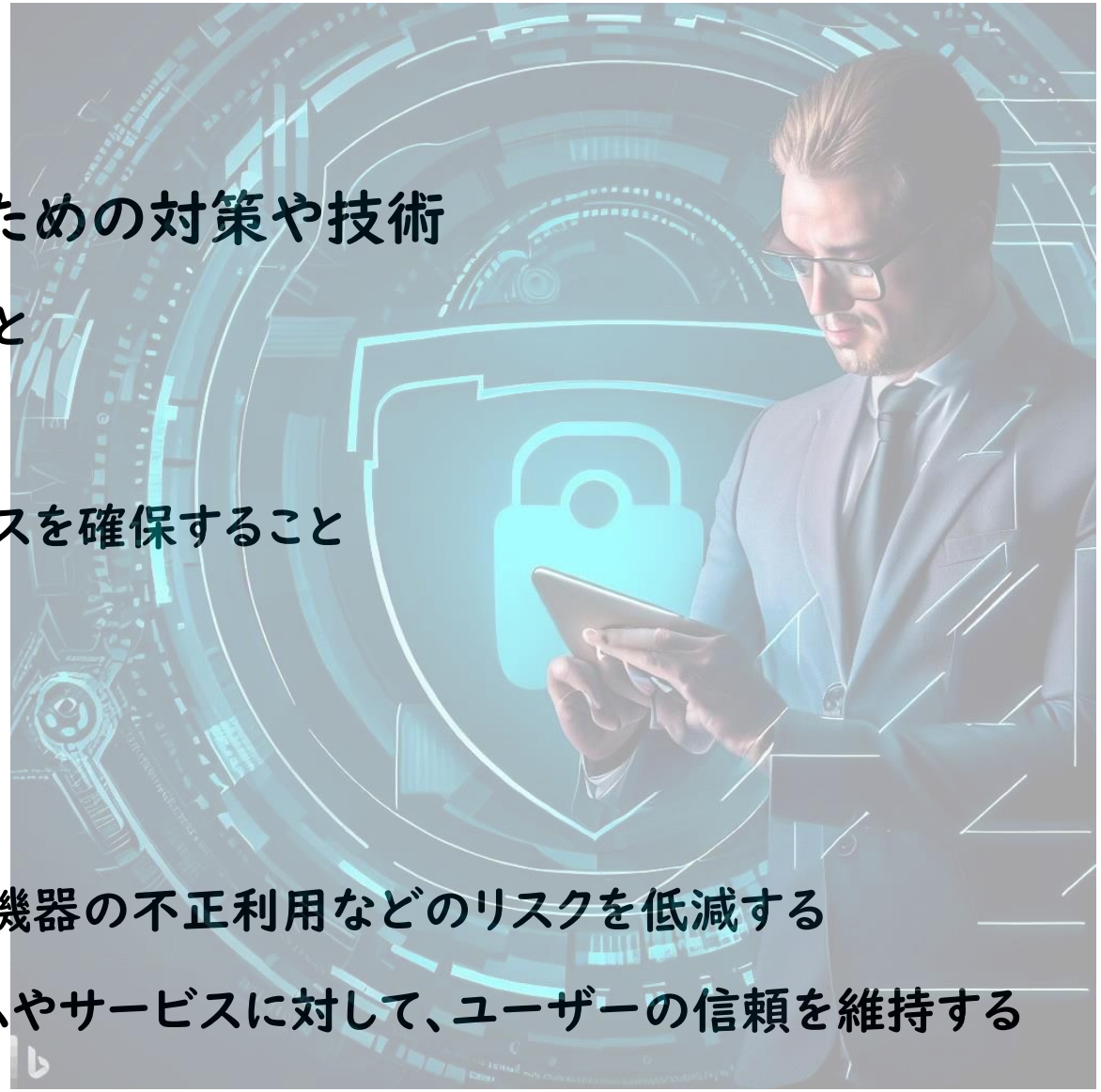
- 外部システムとの通信やデータの送受信
- 制御アルゴリズムの実行
- センサーデータの収集と処理



セキュリティの重要性

•セキュリティとは？

- 「機密性」、「完全性」、「可用性」を確保するための対策や技術
  - 機密性: データや情報への不正アクセスから守ること
  - 完全性: データの改竄や破壊から守ること
  - 可用性: システムやサービスの正常な動作とアクセスを確保すること



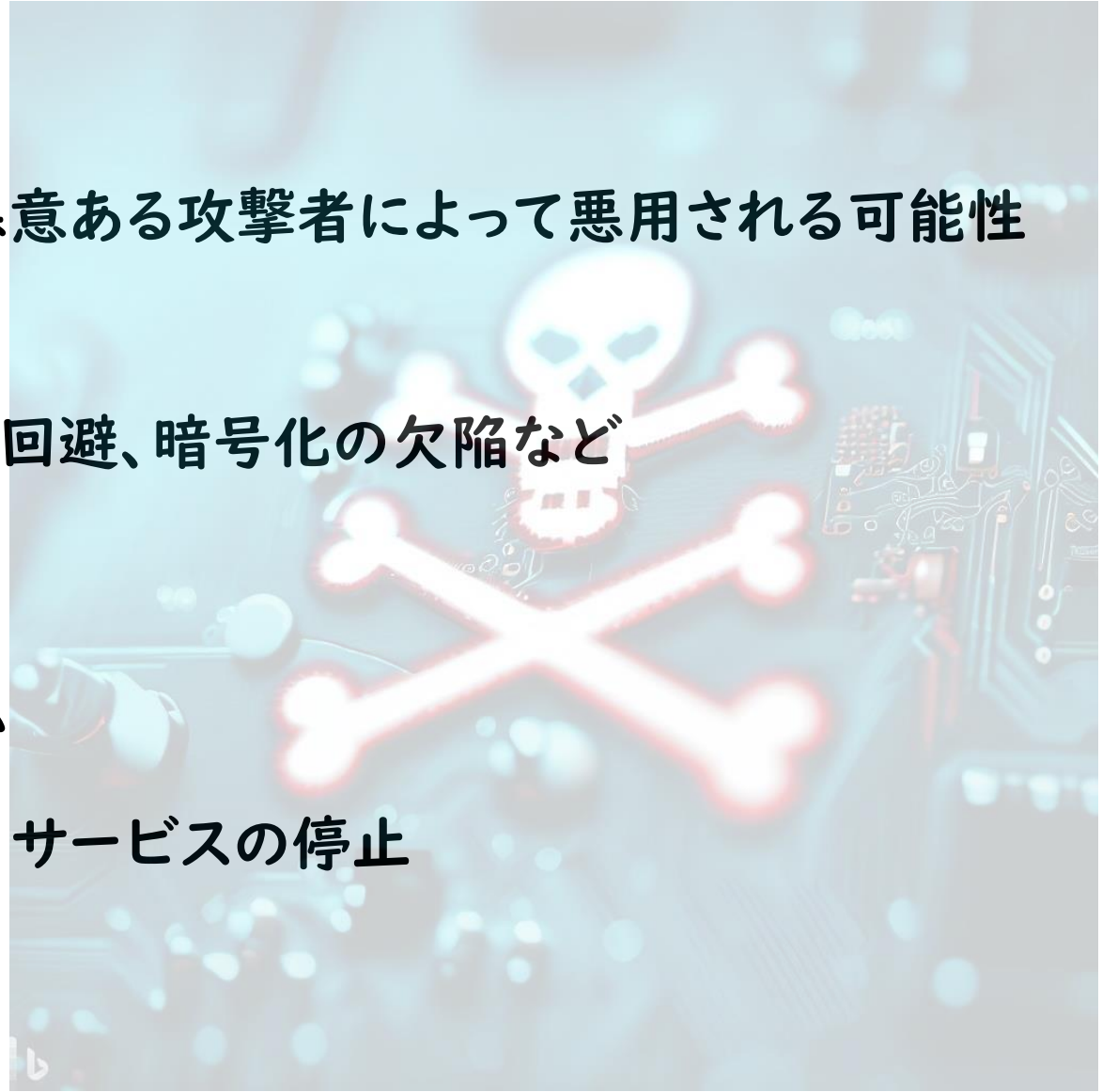
•セキュリティの重要性

- 「リスクの軽減」と「信頼の維持」
  - リスクの軽減: セキュリティ対策により、情報漏洩や機器の不正利用などのリスクを低減する
  - 信頼の維持: セキュリティが確保されているシステムやサービスに対して、ユーザーの信頼を維持する

セキュリティの重要性

項目	セキュリティ事項	セキュリティ対策 具体例
機密性の確保	重要な情報や設定、個人情報を外部からの不正アクセスや盗聴から守る必要がある	• 暗号化されたプロトコルを使用して無線通信することで、不正アクセス者が通信内容を傍受できないようにする • パスワードポリシーを強化したり、2段階認証を導入することで認証機能を強化する • ログの記録と監視によって不正アクセスを検知する
完全性の確保	データの改ざんや不正な機能変更から情報や機能を守り、正当性を保護する必要がある	• データの暗号化やハッシュ化を使用してデータ改ざんをチェックする • ファームウェアの署名と検証によってファームウェアの書き換えを防止する
可用性の確保	機器を物理的な破壊や攻撃、電力障害から守り、正常な動作と利用を確保する必要がある	• 冗長構成を採用してシステムを構築し、物理的な損傷にも耐えられるようにする • 不正な電波侵入を防ぐために電波遮断素材を使用したり、電波攻撃時に周波数を変更したり、管理者への警告や通知を行う

組み込みソフトウェアのセキュリティの必要性

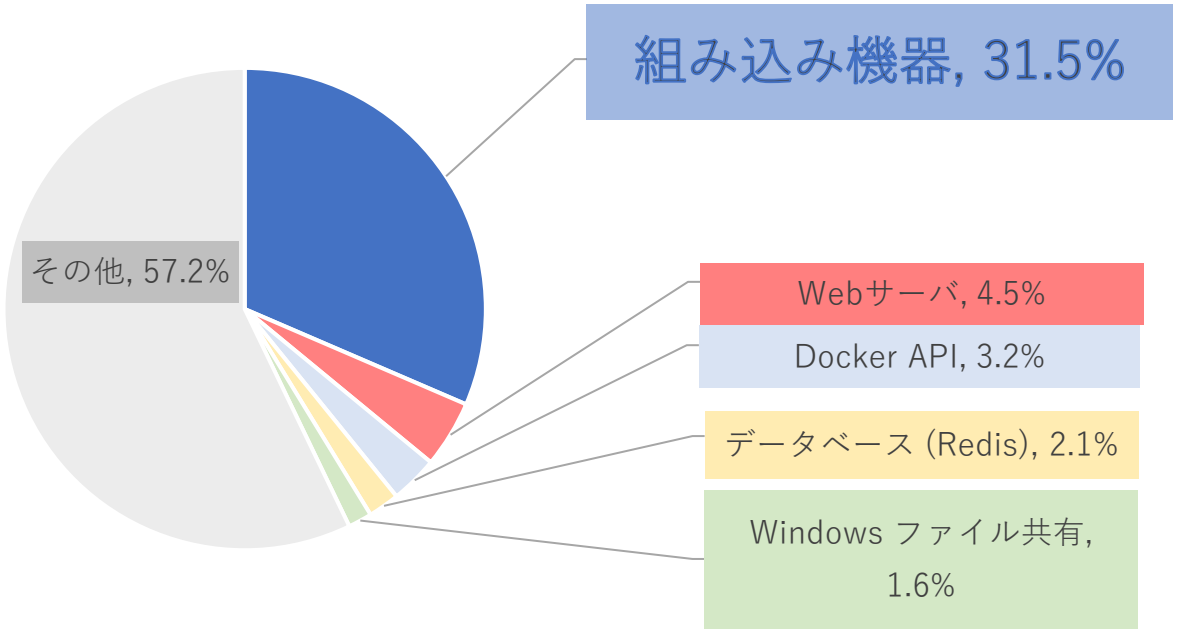


- 組み込みソフトウェアの脆弱性
  - 組み込みソフトウェアには脆弱性が存在し、悪意ある攻撃者によって悪用される可能性がある
  - 脆弱性の例: バッファオーバーフロー、認証の回避、暗号化の欠陥など
- セキュリティの影響範囲
  - データ漏洩: 重要な情報や個人情報の漏えい
  - サービス停止: システムの脆弱性攻撃による、サービスの停止

組み込みソフトウェアのセキュリティの必要性

組み込み機器に対するサイバー攻撃の増加

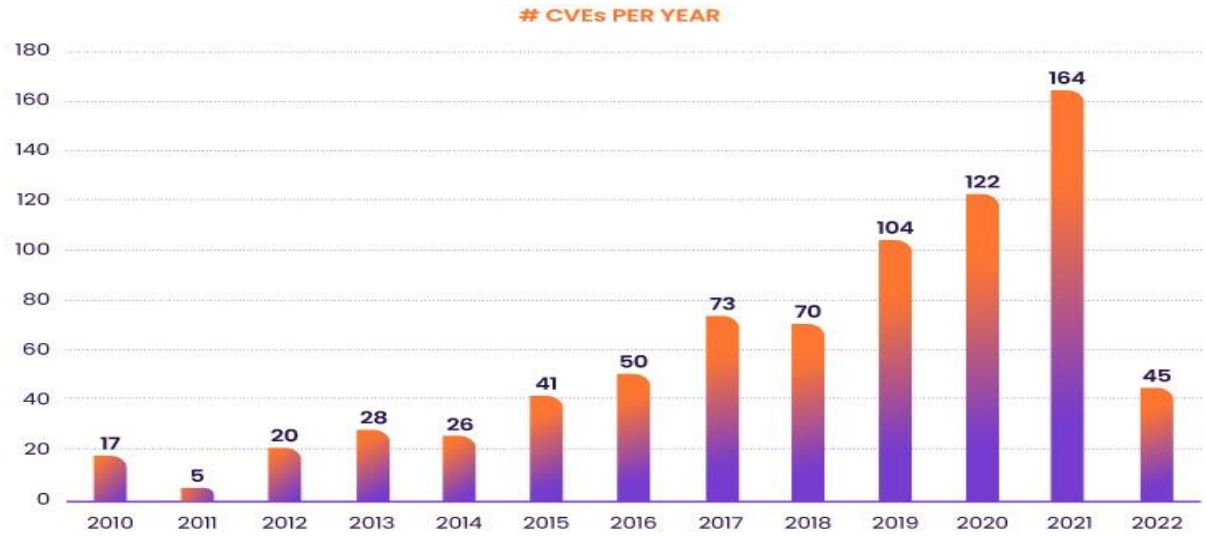
観測対象のサイバー攻撃5,226億パケットのうち、
3割以上が、IoT機器を対象としている



NICTER観測レポート2022
(https://www.nict.go.jp/press/2023/02/14-1.html) を引用

ソフトウェアの脆弱性の年別検出数

790件の脆弱性と公開年度を示したグラフ。2020年以前に公開された脆弱性を抱えたまま、450万台以上の組み込み機器が稼働している

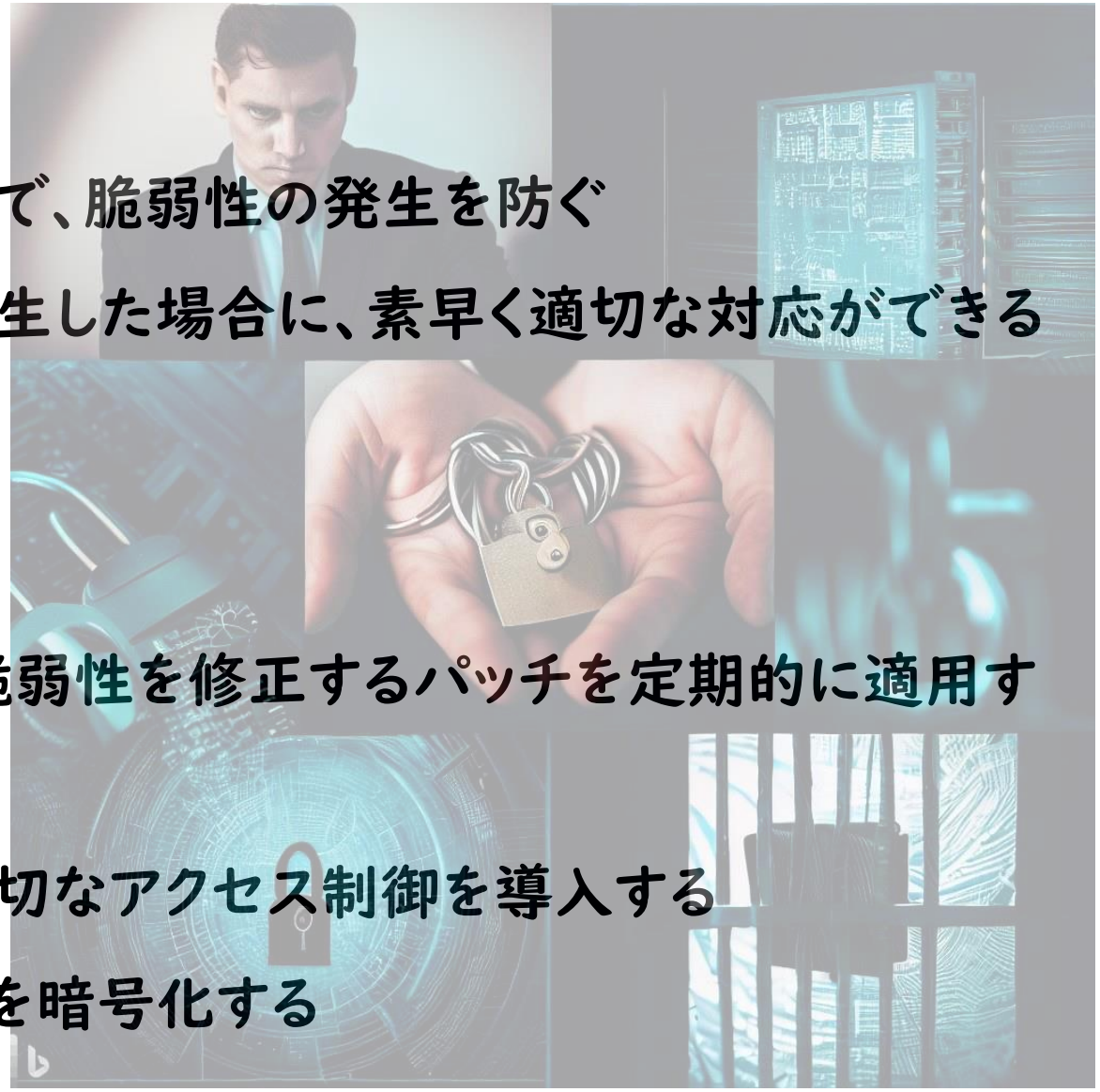


Tech Republic.
(https://www.techrepublic.com/article/vintage-security-vulnerabilities-still-threaten-businesses/)を引用

組み込みソフトウェアのセキュリティ対策

セキュリティ対策の重要性

- **事前予防**: セキュリティ対策を事前に行うことで、脆弱性の発生を防ぐ
- **レスポンス能力**:セキュリティインシデントが発生した場合に、素早く適切な対応ができるようにする



セキュリティ対策の例

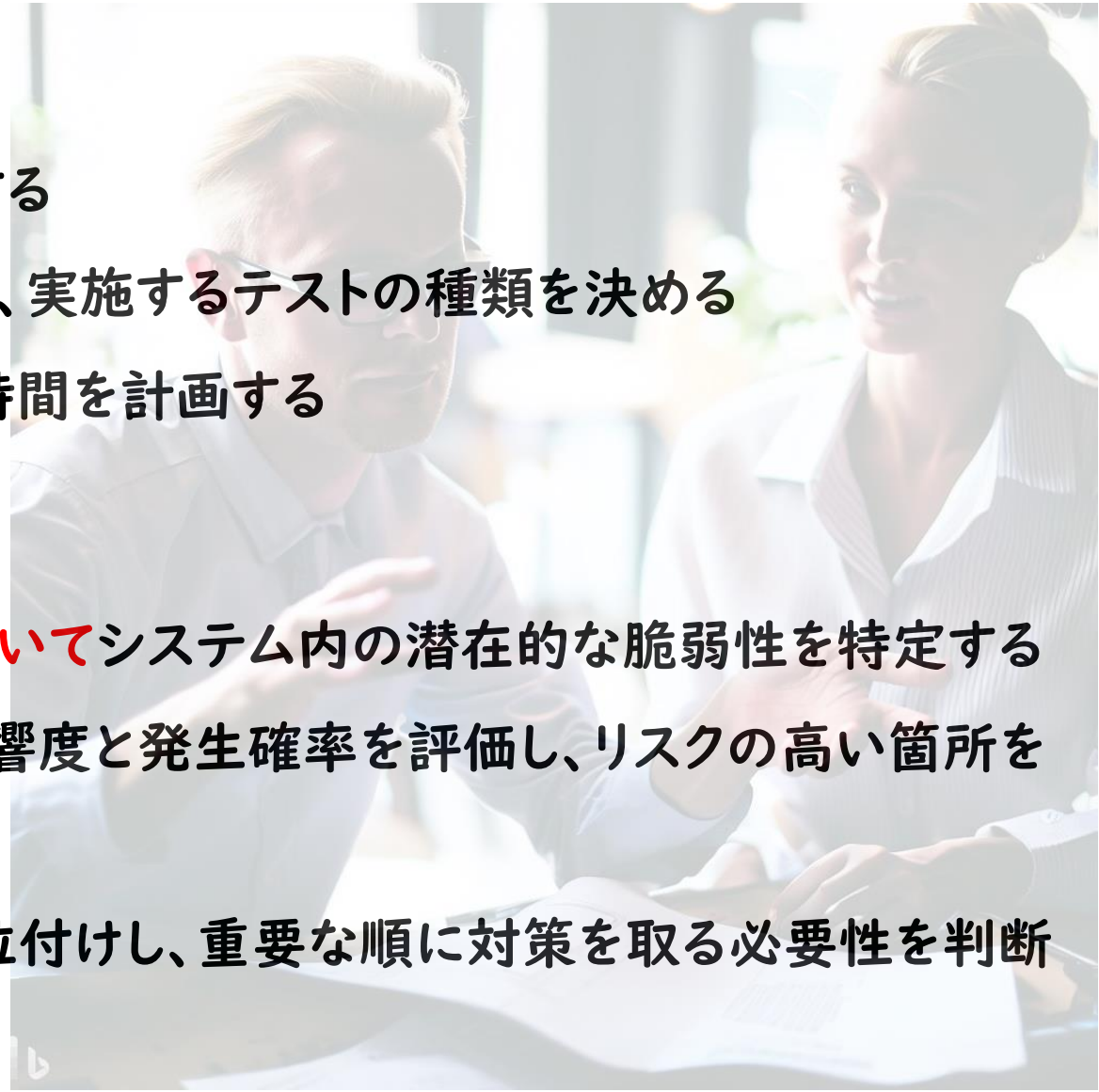
- **パッチ適用**: ソフトウェアやファームウェアの脆弱性を修正するパッチを定期的に適用する
- **アクセス制御**: 不正アクセスを防ぐために、適切なアクセス制御を導入する
- **暗号化**: データの保護のために、重要な情報を暗号化する

組み込みソフトウェアとPC/Webのセキュリティの違い

比較項目	組み込みソフトウェア	PC/Webセキュリティ
リソース制約	メモリや処理能力が限られており、セキュリティ機能の実装に制約がある	比較的リソースが豊富で、高度なセキュリティ機能が実装されている
物理的制約	ネットワーク経由以外に、機器への物理的な攻撃など、様々な攻撃経路がある	ネットワーク経由での攻撃や悪意あるソフトウェアの実行など、攻撃経路が限定的
脆弱性の影響範囲	脆弱性がある場合、デバイスそのものや接続されたシステムに直接影響を及ぼす可能性がある	脆弱性がある場合、ユーザーのデータやプライバシーが漏洩する可能性がある
テスト環境の制約	ハードウェアの特性やリアルタイム処理を考慮したテスト環境が必要	テスト環境の構築が比較的容易で、様々なツールや仮想環境が利用可能

セキュリティテストのプロセス

テスト計画の作成とリスクアセスメント



テスト計画の作成

- 目標の設定: テストの目的や範囲を明確に定義する
- テストの種類: 機能テスト、セキュリティテストなど、実施するテストの種類を決める
- テストのスケジュール: テストの実施予定や所要時間を計画する

リスクアセスメント

- システムの脆弱性の特定: セキュリティテストを用いてシステム内の潜在的な脆弱性を特定する
- 脆弱性の影響度と発生確率の評価: 脆弱性の影響度と発生確率を評価し、リスクの高い箇所を特定する
- リスクの優先順位付け: 特定したリスクを優先順位付けし、重要な順に対策を取る必要性を判断する

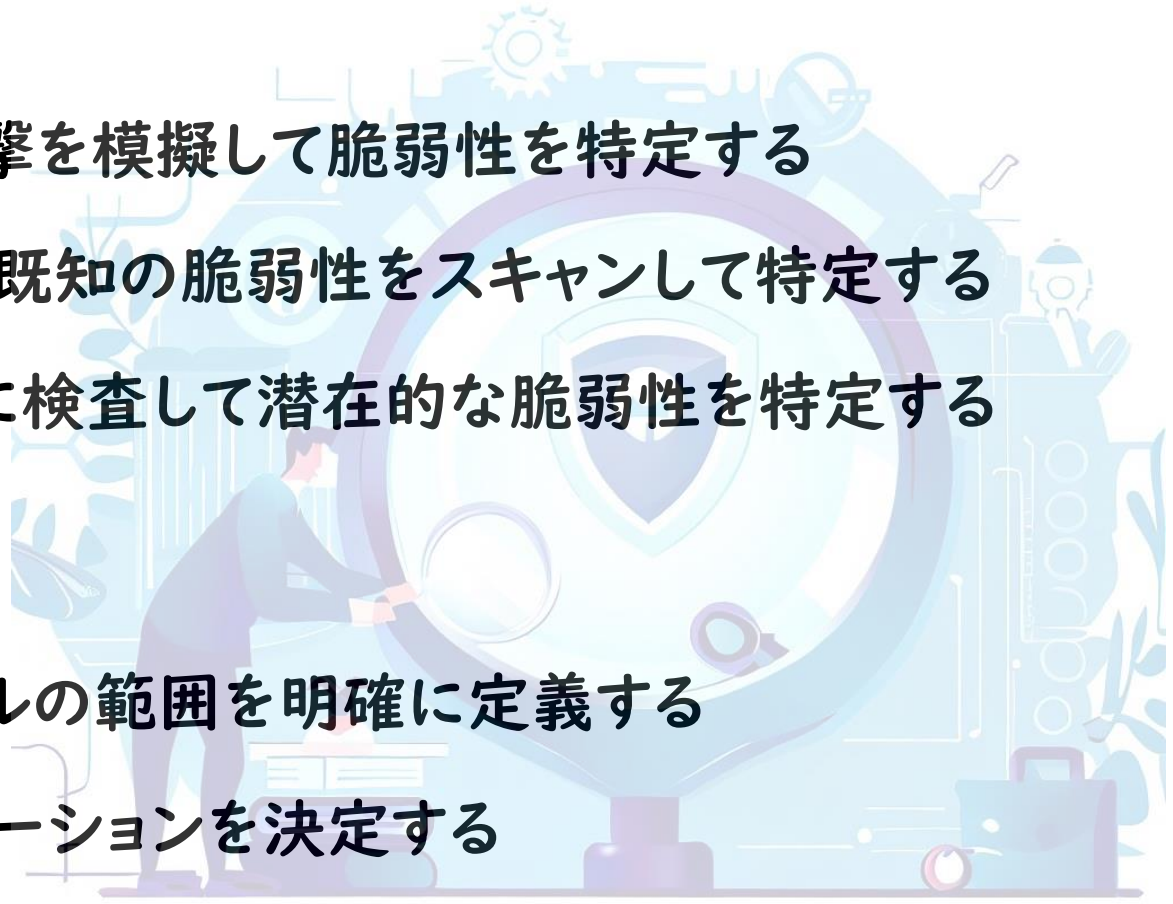
システムの脆弱性の特定方法

セキュリティテストのタイプ

- **ペネトレーションテスト**: システムに対する攻撃を模擬して脆弱性を特定する
- **脆弱性スキャン**: システムやネットワーク上の既知の脆弱性をスキャンして特定する
- **コードレビュー**: ソフトウェアのコードを詳細に検査して潜在的な脆弱性を特定する

セキュリティテストの範囲の決定

- **機能範囲**: テスト対象となる機能やモジュールの範囲を明確に定義する
- **アプリケーション対象**: テスト対象のアプリケーションを決定する
- **システム範囲**: システム全体の範囲や関連するネットワークインフラを考慮する



セキュリティテストの範囲を決めるためのキーワード

- システムのアーキテクチャ
  - システムの概要と構成要素
  - ネットワークトポロジー
  - インタフェースやデータフロー
- アクセス制御と認証
  - ユーザー認証と認可のメカニズム
  - アクセス制御リストや権限設定
  - ロールベースのアクセス制御
- 機密性と暗号化
  - 機密データの保護方法
  - 暗号化アルゴリズムと鍵管理
  - データの保管と伝送時の暗号化
- エラーハンドリングと例外処理
  - エラーメッセージやログの処理方法
  - 例外処理の仕組みと回復手段
  - セキュリティ上の問題を引き起こす可能性のあるエラーパス
- データの入力検証
  - 入力フィールドの制約と検証方法
  - クロスサイトスクリプティングやSQLインジェクションなどの脆弱性
  - 不正なデータや操作の検証
- セキュリティ監視とログ管理
  - セキュリティイベントの監視とログ収集
  - 監視ツールやSIEM（セキュリティ情報・イベント管理）の活用
  - 異常なアクティビティの検知と対応手順

リスクアセスメント(脆弱性評価と脅威モデリング)

- **脆弱性評価**
  - **脆弱性の特定**: システムやソフトウェアに存在する潜在的な脆弱性を特定する
  - **脆弱性の分類**: 脆弱性をカテゴリーに分類し、理解を深める
  - **脆弱性の評価**: 脆弱性の深刻度や影響度を評価し、優先順位付けを行う
- **脅威モデリング**
  - **脅威の特定**: システムに対する潜在的な脅威を特定する
  - **脅威の分析**: 脅威の攻撃手法や影響範囲を分析し、リスクを把握する
  - **脅威のモデリング**: 脅威をモデル化し、セキュリティテストの対象となるシナリオや攻撃手法を洗い出す

脆弱性評価

脆弱性の特定

- ペネトレーションテスト: 実際の攻撃手法を模擬し、システムの脆弱性を特定
- コードレビュー: ソフトウェアのソースコードを詳細に分析し、潜在的な脆弱性を特定
- ファジングテスト: 不正な入力やランダムなデータを使用してシステムをテストし、脆弱性を発見
- 脆弱性スキャン: システムやネットワークをスキャンし、既知の脆弱性を検出
- セキュリティ監査: システムのセキュリティポリシーや設定を評価し、脆弱性を特定

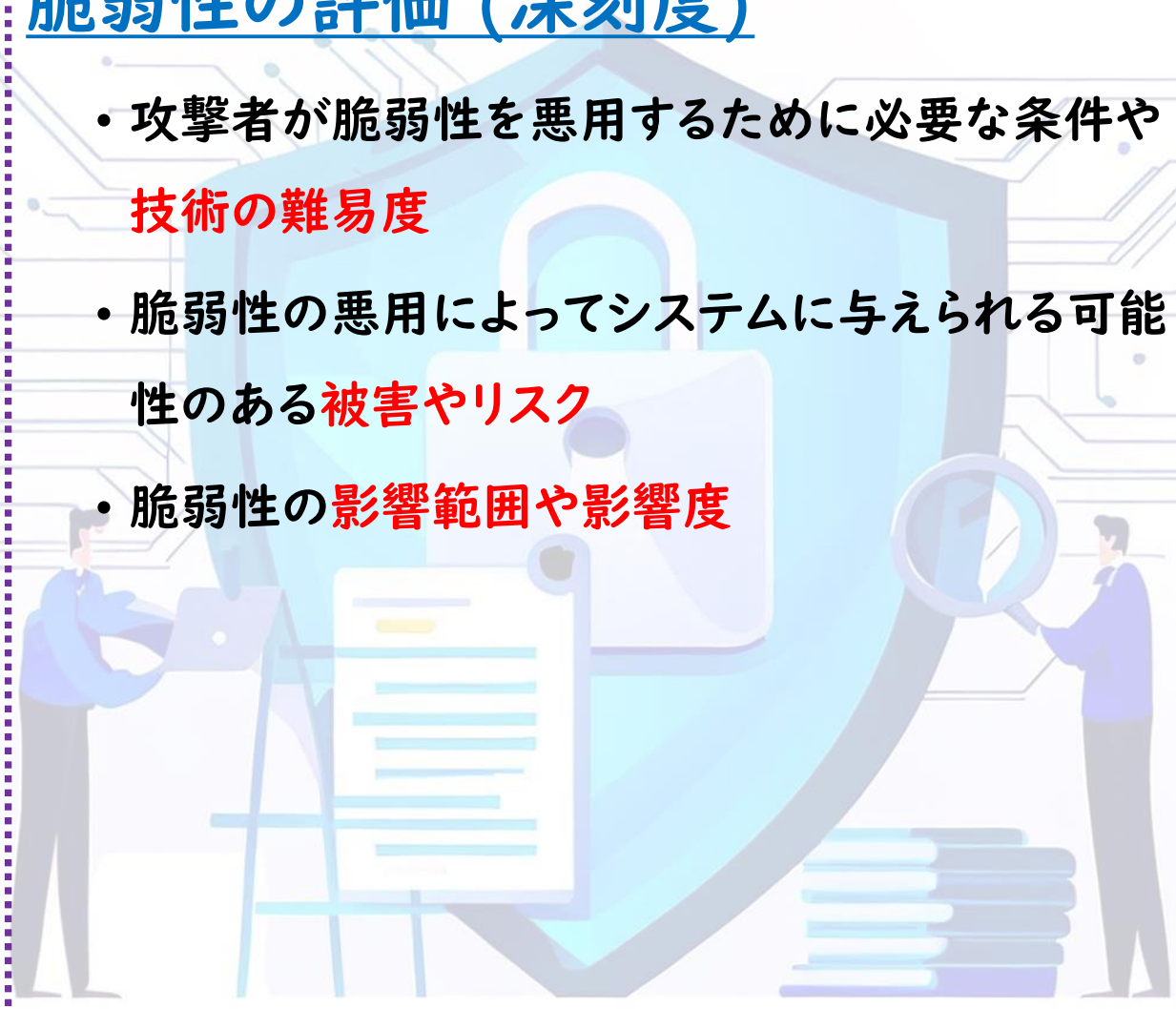
脆弱性の分類

- 認証やアクセス制御に関連する脆弱性
- 暗号化やセキュア通信に関連する脆弱性
- バッファオーバーフローやインジェクションに関連する脆弱性
- 不適切なエラーハンドリングや例外処理に関連する脆弱性
- 情報漏えいやデータ破壊に関連する脆弱性

脆弱性評価

脆弱性の評価 (深刻度)

- 攻撃者が脆弱性を悪用するために必要な条件や技術の難易度
- 脆弱性の悪用によってシステムに与えられる可能性のある被害やリスク
- 脆弱性の影響範囲や影響度



脆弱性の評価(影響度)

- 脆弱性が悪用された場合の被害の重大度
  - 機密情報の漏洩、システムの停止、改ざん、利用者のプライバシーの侵害など・・・
- 脆弱性が影響を与えるリソースの重要度
  - 重要なデータベース、システムの制御機能、ネットワークへのアクセスなど、システムの中核的な要素や重要な資産が影響を受ける・・・

脅威モデリング

脅威の特定

- システムやアプリケーションに対して考えられる様々な脅威を特定する
- 脅威情報の収集を行い、既知の脅威や攻撃手法、セキュリティの脆弱性についての情報を把握する脅威情報を収集するために、セキュリティベンダーや情報共有のプラットフォーム、脆弱性データベースなどを活用する

脅威の分類

- 特定した脅威を分類することにより、それらの特性や特徴を理解する
  - 技術的脅威と社会的脅威
  - 内部脅威と外部脅威
  - 物理的脅威と仮想的脅威
  - アクティブ攻撃とパッシブ攻撃

脅威モデルの作成

- 脅威モデルは、システムやアプリケーションにおける脅威の振る舞いや影響を表現するためのモデル
- 脅威モデルは、攻撃者の目的や手法、攻撃の経路、脆弱性の利用などを考慮して作成する
- モデリング手法は、攻撃ツリーやデータフローモデル、攻撃グラフなどを用いて作成する
- 脅威モデルを作成することにより、潜在的な脅威や攻撃シナリオを可視化し、セキュリティ対策の優先順位付けや設計の検証に活用する

セキュリティテストの技法

分類別セキュリティテスト技法

- システムの攻撃性を評価するためのテスト技法
  - ペネトレーションテスト、ファジングテスト、負荷テスト
- ソフトウェアの潜在的な脆弱性を特定するためのテスト技法
  - 脆弱性スキャン、静的レビュー、ログ分析
- その他のテスト技法
  - ソーシャルエンジニアリングテスト、サードパーティ製品評価

システムの攻撃性を評価するためのセキュリティテスト技法

- ペネトレーションテスト
  - 実際の攻撃手法を模擬し、システムの脆弱性を特定するテスト
  - 脆弱性の特定と悪用の可能性の評価
  - 攻撃シナリオの作成と実施
- ファジングテスト
  - 不正な入力やランダムなデータを注入してシステムの応答をテスト
  - バッファオーバーフローやクラッシュなどの脆弱性を特定
  - 自動化ツールを使用して効率的にテストを実施
- ストレステスト
  - 負荷やストレスをかけてシステムの耐久性やレスポンスをテスト
  - ディナイオブサービス（DoS）攻撃やリソースの枯渇などのテスト
  - システムの弱点や障害への対応能力を評価

脆弱性を特定するためのセキュリティテスト技法

- **脆弱性スキャン**
  - システムやネットワーク上の既知の脆弱性を自動的に検出
  - ポートスキャン、脆弱性データベースの利用などを行う
  - 定期的なスキャンによって脆弱性の変化を監視
- **静的レビュー**
  - ソフトウェアのコードを審査して脆弱性を特定
  - 不適切な認証、データの検証不備などの脆弱性を発見
  - 静的解析ツールや手動のコードレビューを活用
- **ログの解析**
  - セキュリティイベントやログから脆弱性を特定
  - 不正なアクティビティや侵入の痕跡を追跡
  - ログ分析ツールやデバッグツールを使用して解析

その他、セキュリティテスト技法

- ソーシャルエンジニアリング
  - 人間の社会工学的手法を使用して組織やユーザーのセキュリティ意識をテスト
  - フィッシング攻撃、電話での情報収集などを実施
  - 教育やトレーニングの強化に役立つ結果を得る

- サードパーティ製品の評価
  - 使用するサードパーティ製品のセキュリティをテスト
  - ソフトウェア、ハードウェア、ライブラリなどを対象にテスト
  - セキュリティパッチやアップデートの適用を検討

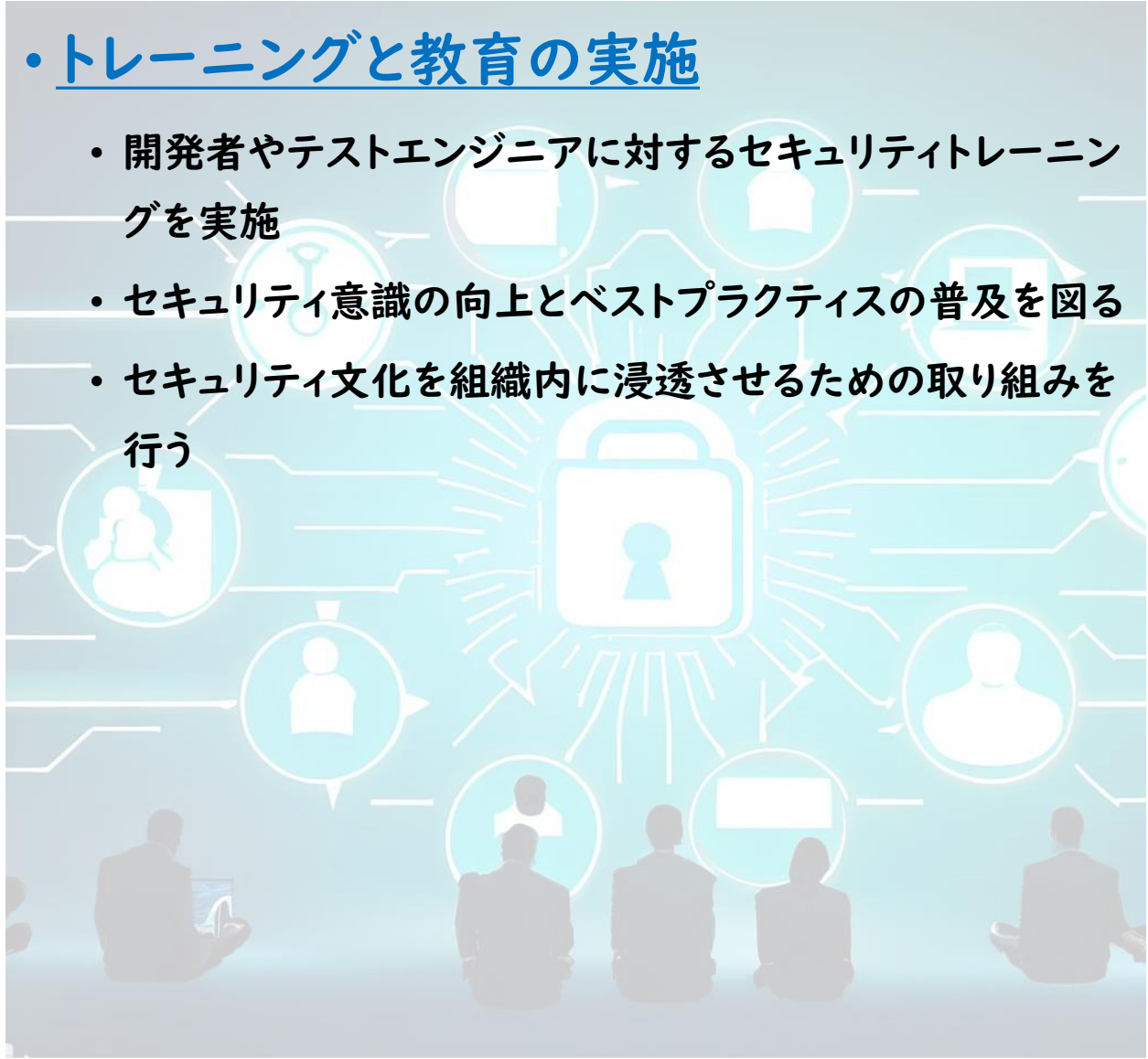
セキュリティテストのベストプラクティス

セキュリティテストのベストプラクティス

- **継続的なテスト**
  - セキュリティテストはプロジェクトの初期段階から終了まで**継続的に実施**
  - 開発プロセスに組み込まれたテストと監視を行う
  - 変更やアップデートに応じてテストを**継続的に更新**
- **レッドチームとブルーチームの活用**
  - レッドチームは攻撃者の立場からシステムをテスト
  - ブルーチームは防御側の役割でセキュリティ対策を検証
  - 両チームの連携により包括的なセキュリティテストを実現
- **実際の攻撃手法の模擬**
  - 実際の攻撃手法や脅威を模倣してテストを実施
  - フィッシング攻撃やゼロデイ攻撃などをシミュレーション
  - システムの実際の脆弱性に対する対策の有効性を評価
- **ドキュメンテーションの重要性**
  - テスト計画、テストケース、結果のドキュメンテーションを行う
  - テストの実施方法、発見された脆弱性、対策の記録を残す
  - ドキュメントによりテスト結果の可視化と報告を行う

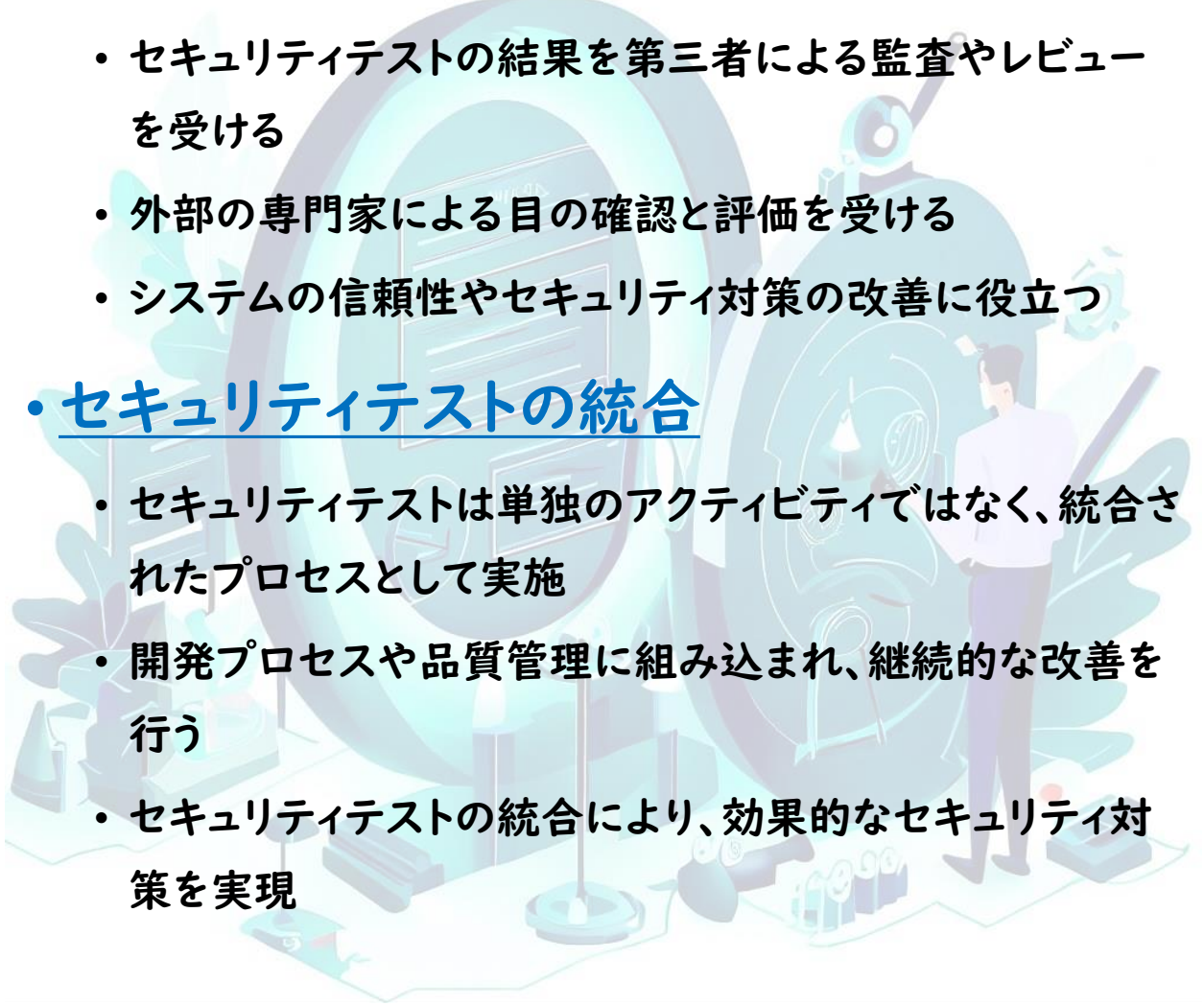
セキュリティテストのベストプラクティス

- パッチとアップデートの管理
  - システムやソフトウェアの脆弱性に関する情報を追跡
  - セキュリティパッチやアップデートの適用を適時に行う
  - 脆弱性情報の共有や脅威インテリジェンスの活用も重要
- 結果のフィードバックと改善
  - セキュリティテストの結果をフィードバックループに組み込む
  - 発見された脆弱性や改善点を追跡し、適切な対策を講じる
  - 繰り返しテストを行いながらセキュリティを改善していく
- トレーニングと教育の実施
  - 開発者やテストエンジニアに対するセキュリティトレーニングを実施
  - セキュリティ意識の向上とベストプラクティスの普及を図る
  - セキュリティ文化を組織内に浸透させるための取り組みを行う



セキュリティテストのベストプラクティス

- **セキュリティテストの基本知識の理解**
  - セキュリティ脆弱性の種類: バッファオーバーフロー、クロスサイトスクリプティング、SQLインジェクションなど、一般的な脆弱性を理解する
  - セキュリティテストツール: 脆弱性スキャナやペネトレーションテストツールなどの基本的なセキュリティテストツールの使用方法を学ぶ
- **プロトコルや技術の理解**
  - ネットワークプロトコル: TCP/IP、HTTP、SSL/TLSなどの基本的なネットワークプロトコルを理解する
  - 暗号化技術: 共通鍵暗号化、公開鍵暗号化、ハッシュ関数などの暗号化技術について理解する
- **第三者の監査とレビュー**
  - セキュリティテストの結果を第三者による監査やレビューを受ける
  - 外部の専門家による目の確認と評価を受ける
  - システムの信頼性やセキュリティ対策の改善に役立つ
- **セキュリティテストの統合**
  - セキュリティテストは単独のアクティビティではなく、統合されたプロセスとして実施
  - 開発プロセスや品質管理に組み込まれ、継続的な改善を行う
  - セキュリティテストの統合により、効果的なセキュリティ対策を実現



まとめ

まとめ

- 組み込みソフトウェアのセキュリティテストの重要性
  - 組み込みソフトウェアは様々なシステムで重要な役割を果たしている
  - セキュリティの脆弱性が悪影響を及ぼす可能性がある
  - セキュリティテストはシステムの安全性を確保するために不可欠
- セキュリティテストのプロセスと必要な知識
  - 脆弱性評価や脅威モデリング、テストのタイプと範囲の決定が重要
  - テスト設計には適切な情報の収集とドキュメンテーションが必要
- セキュリティテストの技法
  - ペネトレーションテスト、ファジングテスト、コードレビューなどの技法を活用
  - セキュリティテストハーネスやログ分析なども有効な技法として活用

まとめ

- セキュリティテストの重要性
  - セキュリティテストは組み込みソフトウェアの脆弱性を特定し、修正するための手段
  - 脆弱性の発見と対策の実施により、システムの信頼性とセキュリティを向上させる
  - セキュリティテストの適切な実施は企業や顧客の信頼を築く重要な要素
- 今後の必要性
  - 組み込みソフトウェアの利用はますます拡大しており、セキュリティの脅威も進化している
  - セキュリティテストは新たな脅威に対応し続ける必要があり、継続的な改善が求められる
  - セキュリティテストの専門家やツールの需要は今後も高まっていく

ご清聴ありがとうございました