# GeneXus の**セキュリティ**

GeneXus™

サイバーセキュリティ

GeneXus

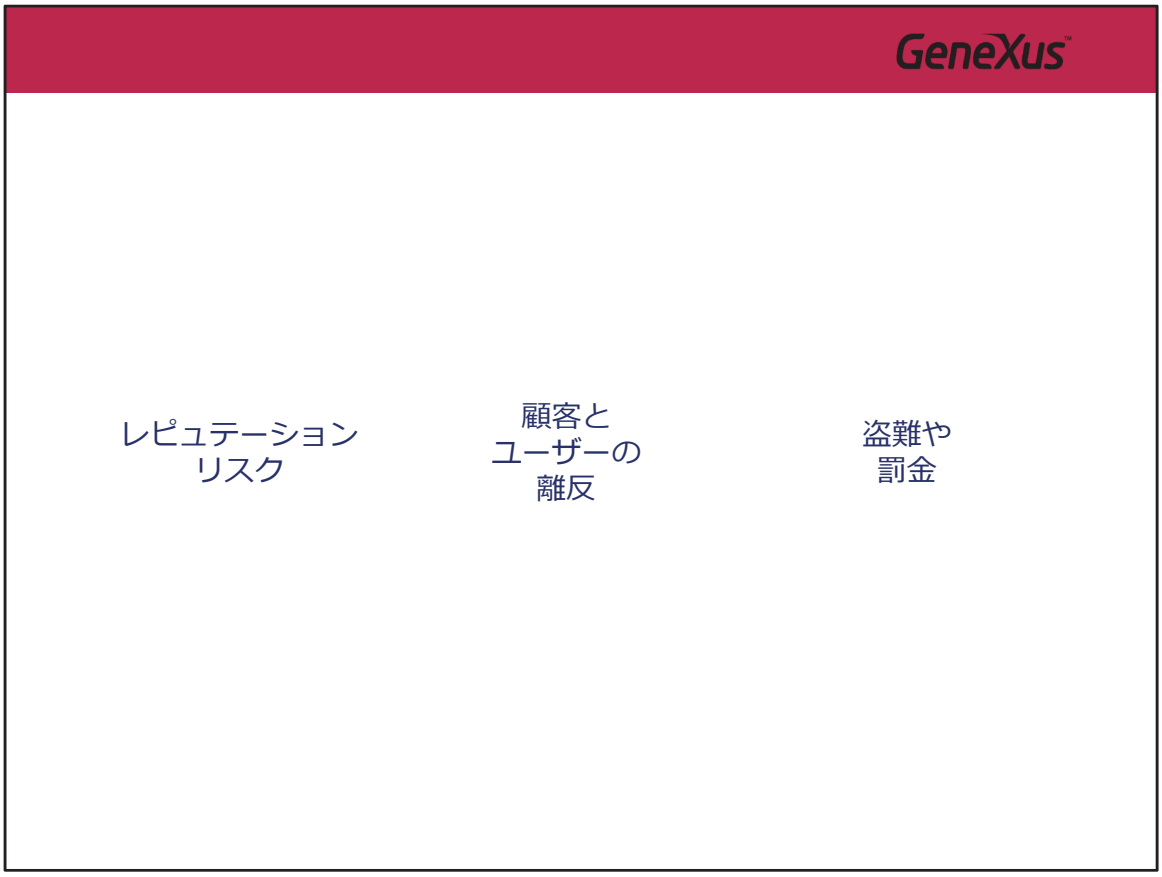
## サイバーセキュリティ


**コンピューターセキュリティ** (サイバーセキュリティ) では、ソフトウェアインフラストラクチャ、より具体的にはデータの保護に重点が置かれます。この領域には、セキュアで信頼性に優れたシステムの開発を支援するメソッド、標準、および技術の設計が含まれます。

サイバーセキュリティは、ネットワーク、サーバー、データベース、アプリケーションなど、システム内のさまざまなレイヤーに関係し、それぞれのレイヤーに脅威とその対策があります。GeneXus で開発したアプリケーションも例外ではありません。

システムのセキュリティが重要である理由を説明します。

GeneXus




セキュリティギャップのもたらす影響は、各当局からの罰金のほか、許可されていない情報が公開されることによるユーザーと顧客を失うことにつながる組織のレピュテーションリスク、組織からの金品の盗難、巻き込まれたユーザーからの要求など、多岐にわたります。


現在、プライバシー、データ保護、およびセキュリティギャップに関連する、より厳しい規制が策定されつつあります。コンピューターシステムの不正使用がますます増加するなかで、その一例となるのが EU 一般データ保護規則 (EU GDPR) です。

**http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/**

脆弱性

GeneXus

## 脆弱性


**脆弱性**とは、システム内に存在する欠陥やバグを意味します。これを利用してエージェントがシステムに不正アクセスし、情報を盗んだりリソースを不正利用したりすると、システムが正常に機能できなくなる可能性があります。

# 優れたプラクティス

このようなリスクは、プログラミングのベストプラクティスを使用することで軽減できます。


ベストプラクティスには、たとえば Open Web Application Security Project (OWASP) が提唱しているものがあります。これは Web 上で公開されており、コーディングに関するガイドライン、マニュアル、および手続きが一通り示されています。

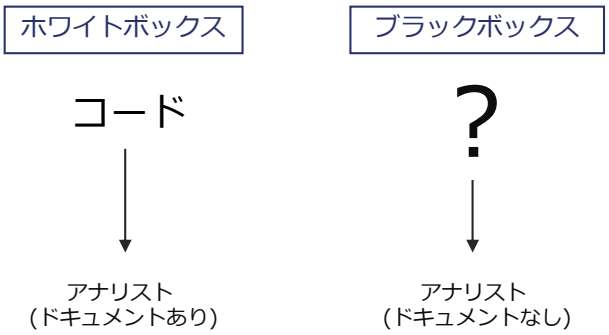
GeneXus

## ペンテスト (侵入テスト)


これに関して、セキュアな開発サイクルでよく知られている実例の 1 つに「ペンテスト (侵入テスト)」があります。事前の承認に基づいてシミュレーションとして攻撃を行い、システムのセキュリティを評価し、開発段階で予期していなかった脆弱性を特定するためのテストです。

GeneXus

## ペンテスト (侵入テスト)

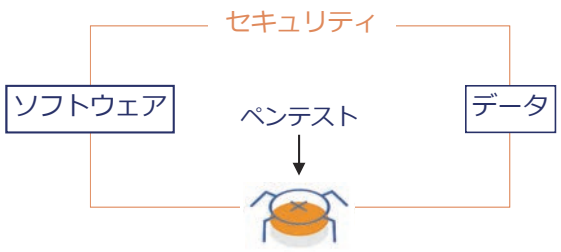



ペンテストは、アナリストがシステムのコードやドキュメントを入手しているかどうかに応じて、ホワイトボックスかブラックボックスのいずれかになります。

これには静的または動的なコードに適用される自動分析ツールを使用するのが一般的ですが、その有効性は、セキュリティアナリストの解釈と、分析対象のシステムのコンテキストに限定されます。

GeneXus

## ペンテスト (侵入テスト)




ペンテストは、セキュアな開発サイクルでよく知られている例の 1 つではありますが、唯一の手段というわけではありません。この段階で検出された脆弱性の修正には、これより前の段階で検出、防止された脆弱性に比べてコストがかかります。

ペンテストが成功して満足な結果が得られたとしても、機能テストでシステムにバグがないことが保証されないのと同じように、システムに脆弱性がないという保証はされません。

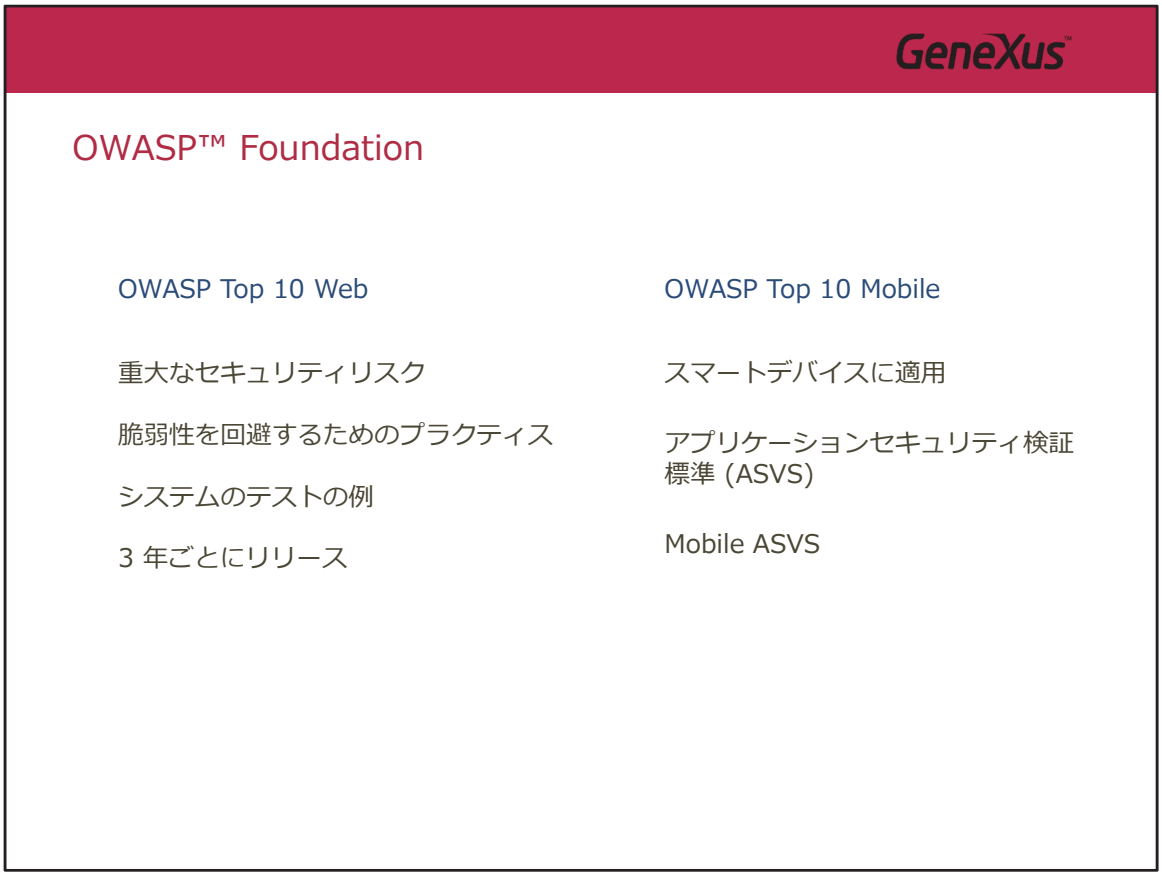

Open Web Application Security Project (OWASP) は、信頼できる組織向けアプリケーションの開発、取得、およびメンテナンスを促進する、テクノロジー企業と関わりのない非営利のオープンコミュニティです。

GeneXus

## OWASP™ Foundation

標準に関する情報

優れたプラクティス



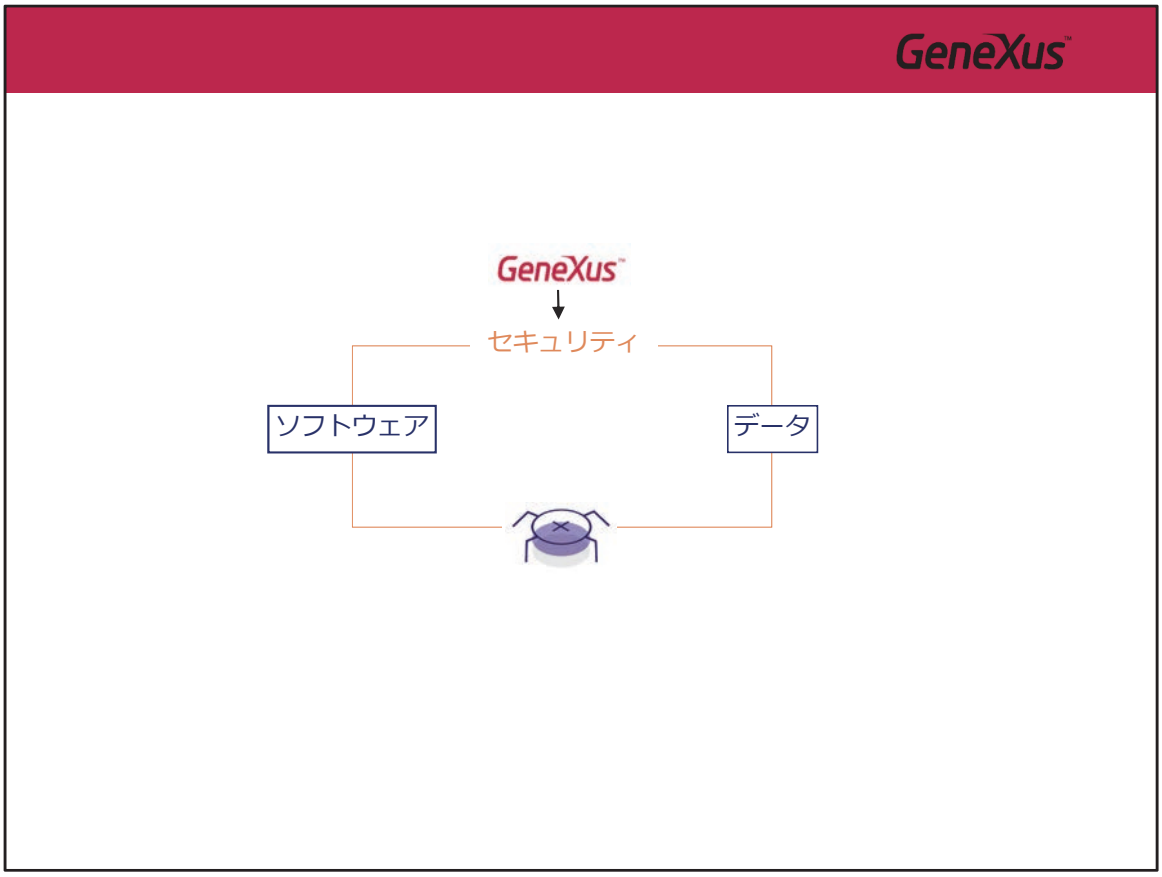

Open Web Application Security Project (OWASP) には、いくつかのプロジェクトがあります。主要かつ重要なプロジェクトの 1 つが **Top 10 Web** プロジェクトです。これは、専門家の間で合意が取れた、影響および頻度を考慮したうえで最も重要なセキュリティリスクを示すドキュメントで構成されています。

このドキュメントの目的は、システムセキュリティの問題に対する意識を高め、脆弱性を回避するためのベストプラクティスに関するガイドラインを提供すること、脆弱性にさらされる可能性のあるシステムのテストの例を提供すること、そして必要な分析を行うための自動テストツールを推奨することです。

Top 10 Web は 3 年ごとにリリースされており、現在は 2021 年版が最新リリースとなっています。

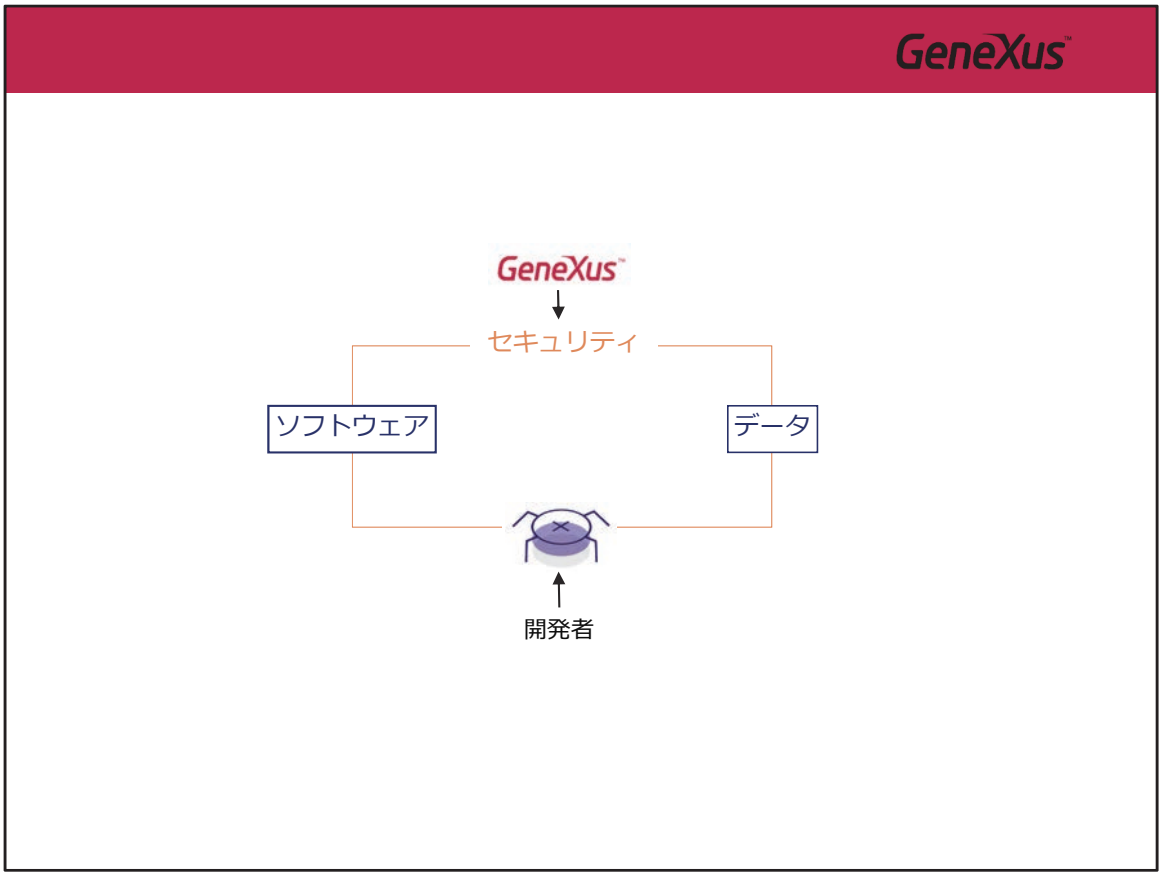
スマートデバイスを対象とした **Top 10 Mobile** もあり、こちらは 2016 年版が最新リリースとなっています。

OWASP には、検証のレベルを定義する ASVS というプロジェクトもあります。これは、アプリケーションのコンテキストに応じた必要最小限のセキュリティ要件として認識されています。これには **Mobile ASVS** もあります (2022 年 8 月時点の最新バージョンは v1.4.2)。




GeneXus とセキュリティコントロールについて見てみましょう。

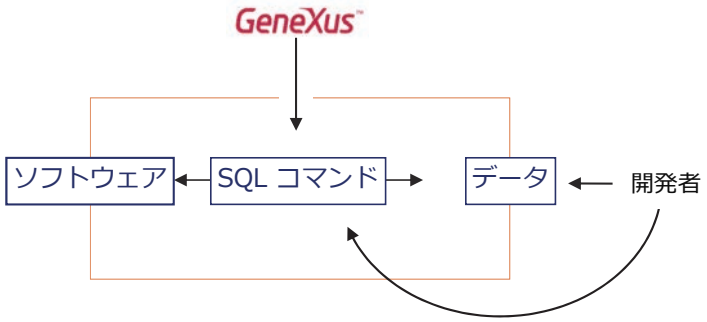
GeneXus は、ナレッジベースから推論されるセキュリティコントロールを自動的に実装します。つまり、GeneXus のようなツールを使用すると、コーディングの誤りに起因する脆弱性を軽減できます。




しかし、ツールを使用する場合でも、ビジネスロジックのレベルや GeneXus の手続き型コードの関係で、開発者によって脆弱性やセキュリティ上の不備が組み込まれ、欠陥やエラーが生じる可能性があります。

たとえば、2017 年版の OWASP Top 10 では、GeneXus が自動的に行う軽減対策や、GeneXus の対策にもかかわらず、開発者が欠陥を組み込んでしまうケースを挙げることができます。

## 1 - SQL インジェクション

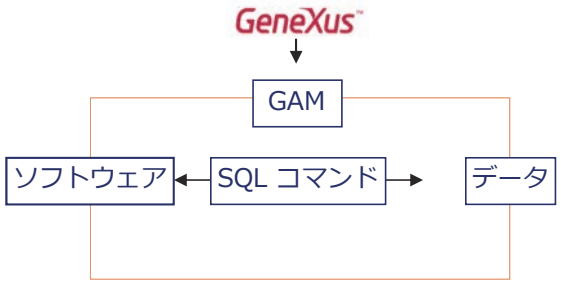



2017 年版の OWASP Top 10 の 1 つ目の項目は、インジェクションです。

SQL インジェクションは特に有名な脆弱性です。この脆弱性は、ユーザーが入力した情報が検証されずにデータベースに送信されるときに生じます。

GeneXus は、開発者がデータベースをリソースとして使用できるように、アプリケーションの汎用性を向上させる SQL コマンドを公開しています。これを使用する場合、開発者は、SQL インジェクションを回避するようにコマンドの入力を調整する必要があります。

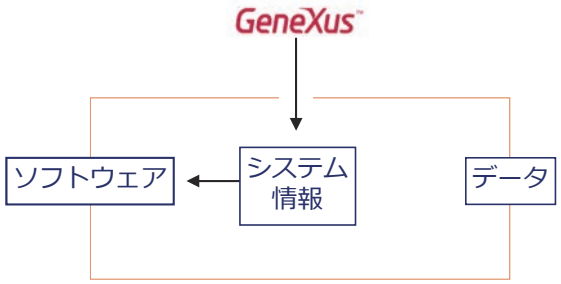
## 2 - 認証の不備




2 つ目の項目は、認証の不備です。GeneXus には、認証関連のタスクを自動化する GAM (GeneXus Access Manager) という認証/認可モデルが用意されています。このツールの適切なセットアップは、開発者やシナリオに応じて異なります。

また、開発者は、パスワードやセッションなどを管理するための専用のモジュールを使用することもできます。

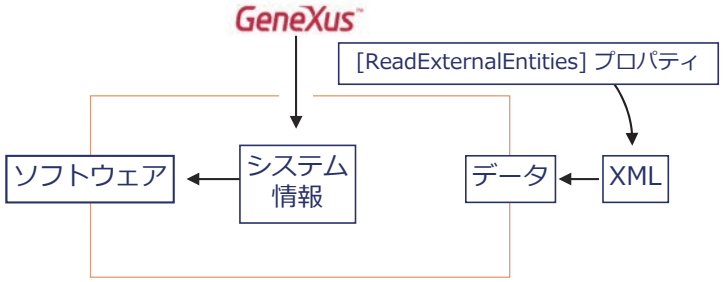
## 3 - 機微な情報の露出




3 つ目の項目は、機微な情報の露出です。要件の分析には、システムの機密情報の特定が含まれます。

そのため、GeneXus では、特定のアクションを実行するのではなく、暗号化、ハッシュ、パスワードの管理、証明書など、機密情報を保護するために必要な機能を提供しています。

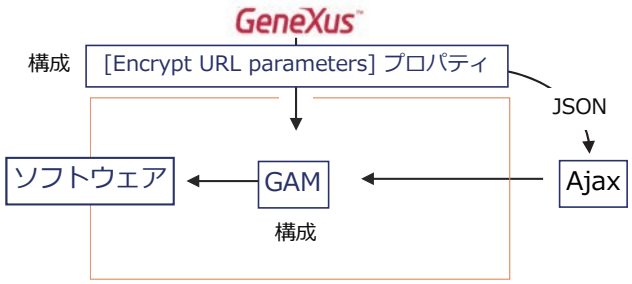
## 4 - XML 外部実体 (XXE) 攻撃




4 つ目の項目は、XML 外部実体 (XXE) 攻撃です。

図に示すように、既定では、GeneXus は XML 外部エンティティの処理は行いません。外部エンティティの処理を有効にするには、[ReadExternalEntities] プロパティを設定する必要があります。その場合、システムの境界を越えた XML の制御については、開発者が責任を持つことになります。

## 5 - アクセス制御の不備




5 つ目の項目は、システムへの不正アクセス、URL の変更、Ajax 呼び出しなどによって発生するアクセス制御の不備です。

GeneXus には URL パラメーターの暗号化を有効にするプロパティが含まれており、これを使用することで、オブジェクトが受け取るすべてのパラメーターを暗号化できます。

また、GAM によるアクセスコントロール、Ajax 呼び出しでの JSON のエンコーディング (既定)、クライアント側だけでなくサーバー側での検証チェックも実行されます。イベントへのコントロールの追加、GAM の適切な設定、URL パラメーターの暗号化を有効にする [Encrypt URL parameters] プロパティの構成は、開発者の責任で行います。

## 6 - セキュリティの設定ミス

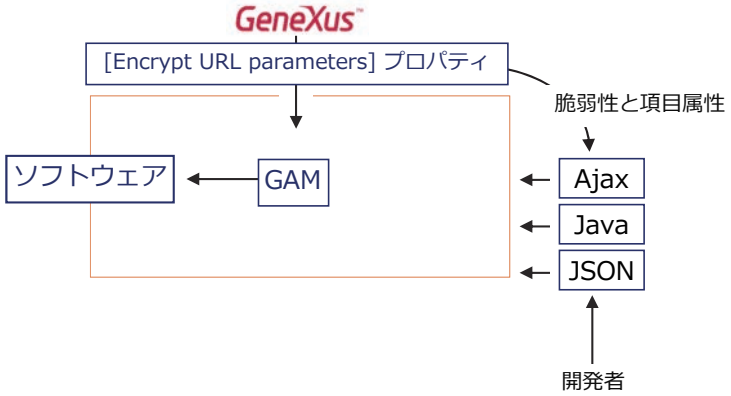

https://www.genexus.com/ja-JP/japan/community-and-support-jp/product-help

6 つ目の項目は、セキュリティの設定ミスです。ほとんどの場合、本番環境のアプリケーションサーバーでのセキュリティ設定の誤りを指します。このために、GeneXus にはいくつかのプロパティがあります。これらを無効にすることが推奨されない場合でも、開発者はプロバイダーのニーズを満たすために調整することができます。

GeneXus の Wiki には、セキュアなサーバーの設定に関するアドバイスとガイドラインが含まれています。

ただし、本番環境のセキュリティをどの程度にするのかは、アプリケーションのサーバーをセットアップするインフラストラクチャ担当者に大きく依存します。

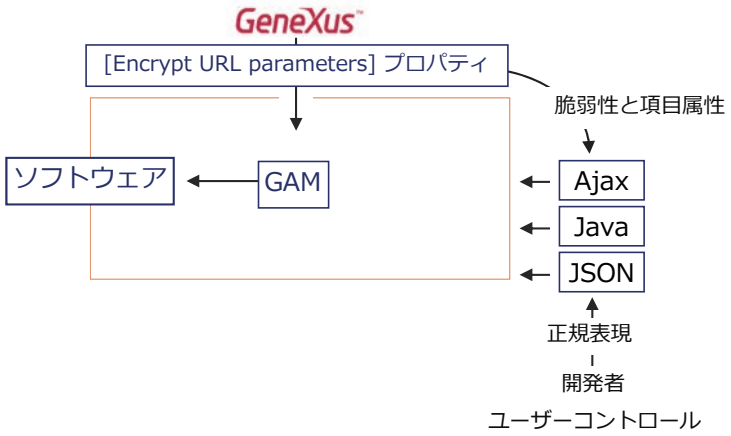
## 7 - クロスサイトスクリプティング (XSS)




7 つ目の項目は、クロスサイトスクリプティングです。この種の脆弱性については、GeneXus は変数および項目属性の長さやタイプを自動的に検証して、システムが受け取るデータをコンテキストに応じてエンコーディングします (HTML JavaScript や JSON などにも自動的に対応します)。

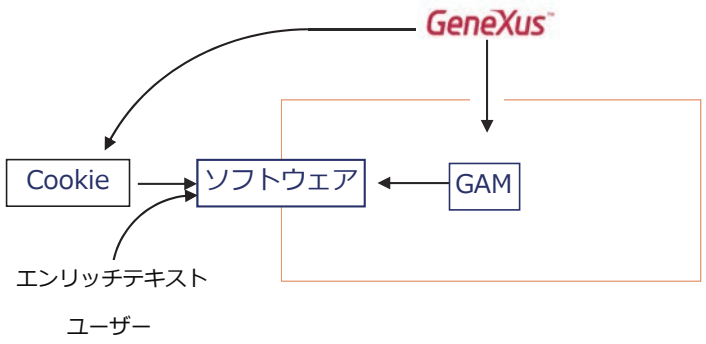
独自の JavaScript や HTML コードを使用する場合は、開発者の責任で検証、制御する必要があります。

## 7 - クロスサイトスクリプティング (XSS)




同様に、GeneXus では、ユーザーの入力が完全に安全であり、事前定義済みの検証コントロールに準拠していると確信できるように、ユーザー入力を検証するための正規表現を定義する機能や、独自またはサードパーティ (GeneXus 以外) のユーザーコントロールを使用する機能が開発者に提供されています。

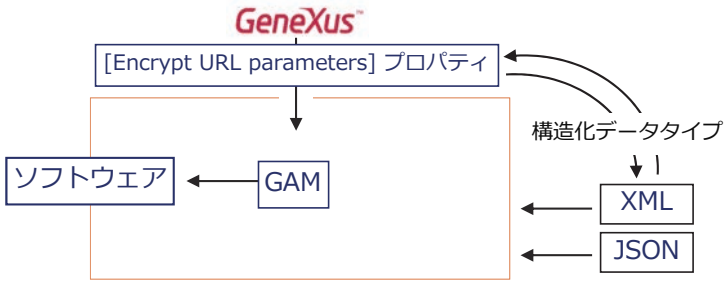
## 7 - クロスサイトスクリプティング (XSS)




Cookie を適切に設定することも忘れないでください。また、GeneXus が自動的に生成したチェックは (そうする必要がない限り) 無効にしないでください。

ユーザーからのエンリッチテキストを受け入れる場合は、この形式の情報をセキュアかつ一貫した方法で制御、検証、および定義することも忘れないようにする必要があります。

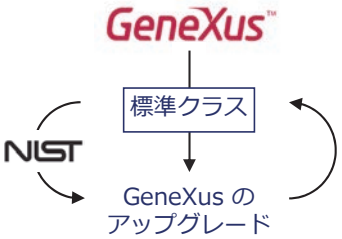
## 8 – 安全でないデシリアライゼーション




8 つ目の項目は、安全でないデシリアライゼーションです。GeneXus は、JSON や XML のような構造化データタイプに対応するために、安全なシリアライゼーションメカニズムを実装しています。

メソッドやデータタイプを使用して XML や JSON からオブジェクトをシリアライズする場合、入力をエンコーディングする必要があります。

## 9 - 既知の脆弱性のあるコンポーネントの使用

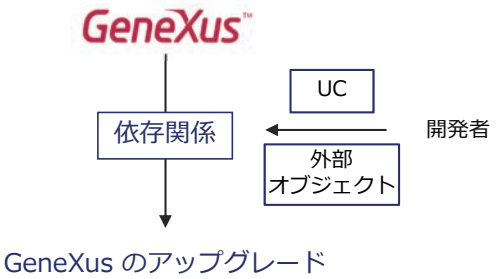



9 つ目の項目は、既知の脆弱性のあるコンポーネントの使用に関するものです。

これに関して、GeneXus は独自の標準クラスを用意しています。この標準クラスは、今後のアップグレードで継続的に確認、修正されます。


また、GeneXus のアップグレードごとに必要に応じて更新される、米国国立標準技術研究所の脆弱性データベースと比較されるサードパーティの依存関係も使用します。

## 9 - 既知の脆弱性のあるコンポーネントの使用




これに関して、GeneXus は独自の標準クラスを用意しています。この標準クラスは、今後のアップグレードで継続的に確認、修正されます。

これは、GeneXus が独自の依存関係を管理し、アップグレードの際にはそれらを更新することを意味します。開発者がユーザーコントロールや外部オブジェクトを追加する場合は、OWASP が提唱するセキュアな開発のベストプラクティスに従うこと、また、依存関係を確認することをおすすめします。

GeneXus

## 9 - 既知の脆弱性のあるコンポーネントの使用

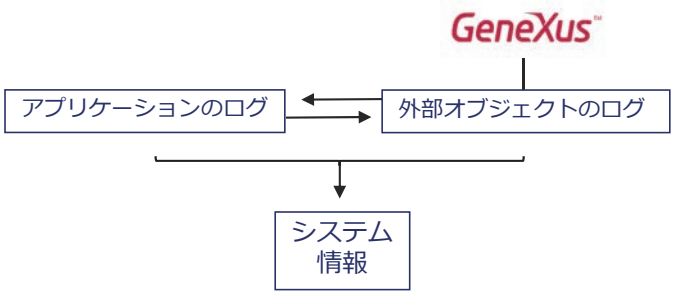
OWASP による優れたプラクティス

OWASP Dependency Check – RetireJS

DBMS の更新

そのためには、DBMS およびそのドライバーを最新の状態に維持し、アップデートおよびセキュリティパッチをサーバーにインストールするほか、さまざまなツールを使用することをおすすめします。

## 10 - 不十分なロギングとモニタリング




最後の項目は、不十分なロギングとモニタリングに関するものです。

どのような場合であっても、セキュリティ関連で起こりうるイベントやインシデントを検出するモニタリングプロセスを定義する必要があります。

GeneXus には、アプリケーションのログにカスタマイズ情報を追加するために、アプリケーションレベルおよび外部オブジェクトレベルでログを収集するメカニズムがありますが、関係するさまざまなデバイス (Web サーバーやファイアウォールなど) から提供される関連情報についても定義する必要があります。また、収集した情報を分類し、処理するプロトコルも定義する必要があります。

これにより、システムにとって脅威となるイベントの検知が可能になり、インシデントを防止することができます。

セキュリティスキャナー


セキュリティスキャナーを使用すると、開発者が組み込んでしまう可能性のある既知の問題を検出することができます。

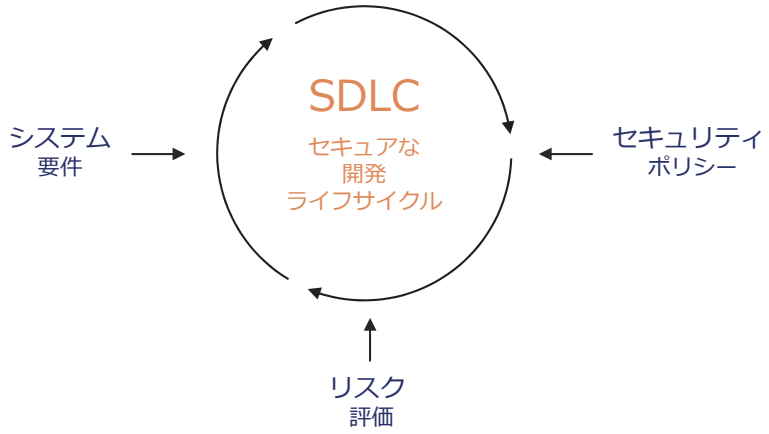
セキュリティスキャナーは、マーケットプレイスで入手可能な GeneXus IDE の拡張機能であり、ナレッジベースのオブジェクトの静的分析を実行し、セキュリティに関する既知の問題の痕跡を探します。

この拡張機能の最新バージョンは GeneXus 16 と互換性があり、2017 年版の OWASP Top 10 に基づいてスキャンするよう更新されています。詳しくは、こちらの Wiki に含まれているドキュメントのリンクを参照してください。

# セキュアな開発ライフサイクル

GeneXus

## セキュアな開発ライフサイクル




セキュアなシステムの開発では、**セキュアなライフサイクル**を実装する必要があります。

つまり、システムの要件をまとめる段階からセキュリティを考慮します。これは開発サイクル全体、つまり、設計、実装、テスト、セットアップ、およびメンテナンスの各段階で適用する必要のあるセキュリティポリシーに基づき、リスク評価を行うことで実現します。