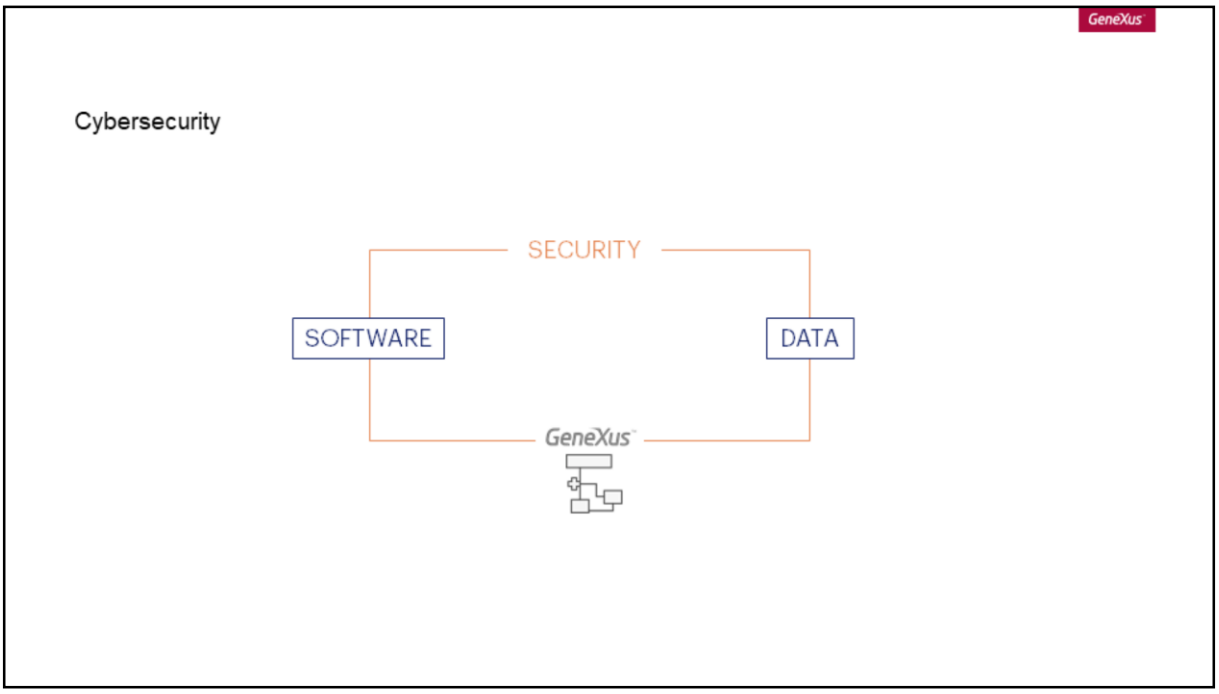
Segurança no GeneXus

GeneXus 16

GeneXus

CYBERSECURITY



A **segurança informática** (ou cibersegurança) se concentra na proteção da infraestrutura do software e, principalmente, na proteção de dados. Nesta disciplina são projetados métodos, normas e técnicas que ajudam a desenvolver sistemas seguros e confiáveis.

A cibersegurança afeta diferentes camadas dos sistemas: rede, servidores, bases de dados, aplicações, etc., e em cada uma delas existem ameaças e contramedidas. As aplicações desenvolvidas com GeneXus não são exceção.

A segurança dos sistemas é importante.

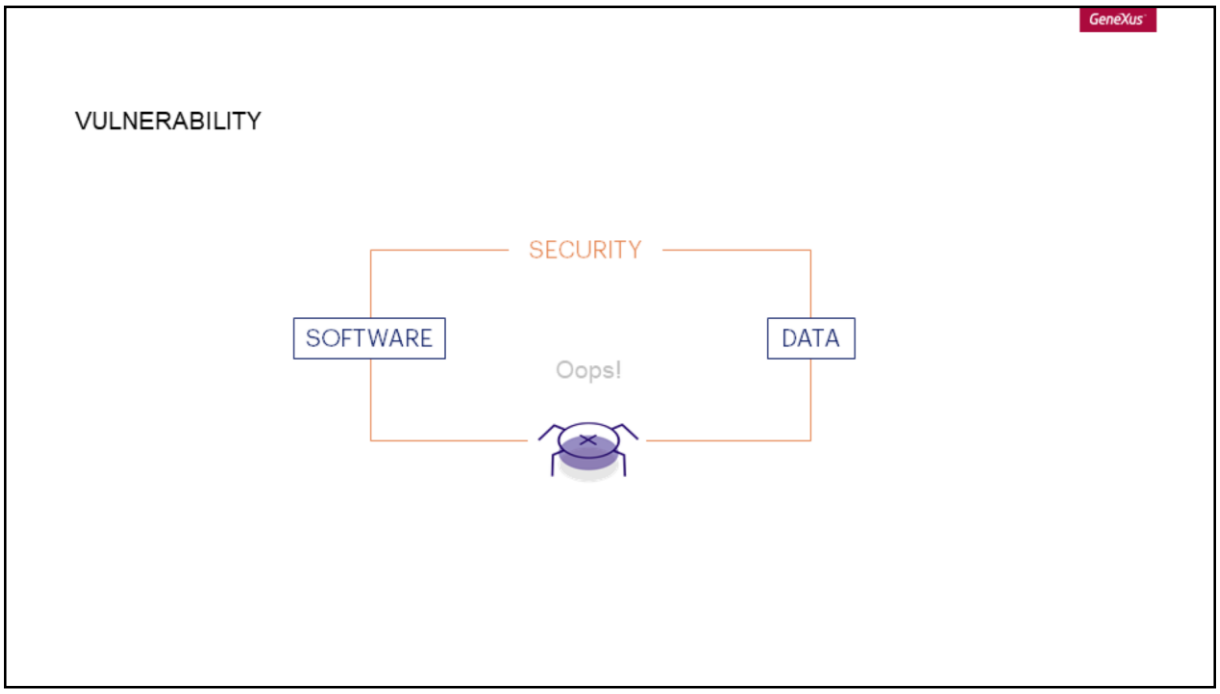
. As consequências de falhas de segurança vão desde um risco à reputação da organização, levando à perda de usuários ou clientes em caso de publicação de informações não autorizada, até roubos em dinheiro ou mercadoria da organização, processos por parte dos usuários envolvidos e multas de diferentes órgãos públicos.

Atualmente estão sendo emitidas novas e mais rigorosas regulamentações sobre privacidade, proteção de dados e publicação de falhas de segurança. Um exemplo desse tipo de regulamentação é a Regulamentação Geral Proteção de Dados europeia, em um mundo no qual os abusos de sistemas informáticos são mais comuns.

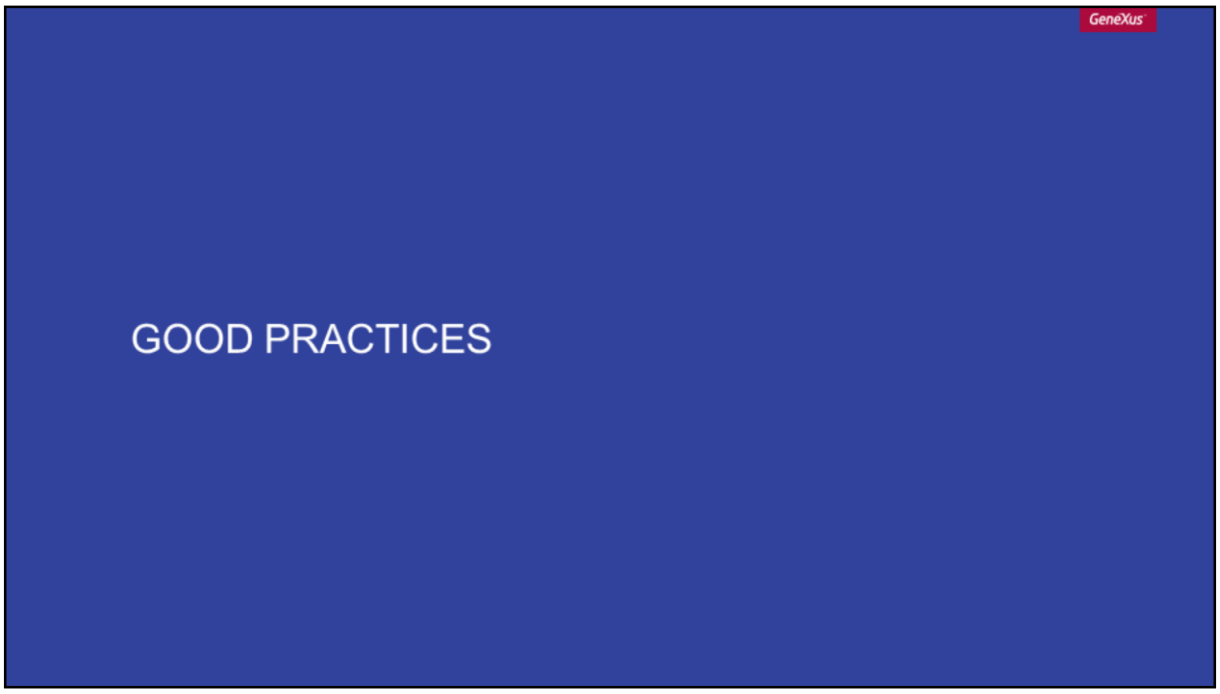
http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

GeneXus

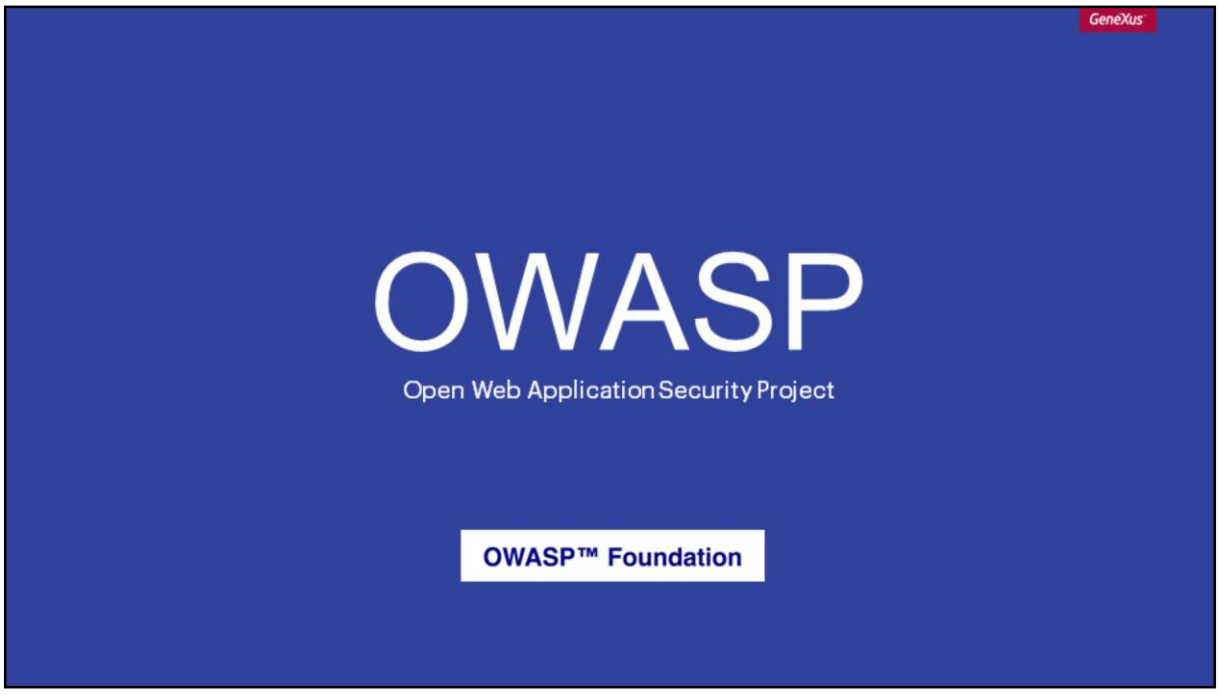
VULNERABILITY



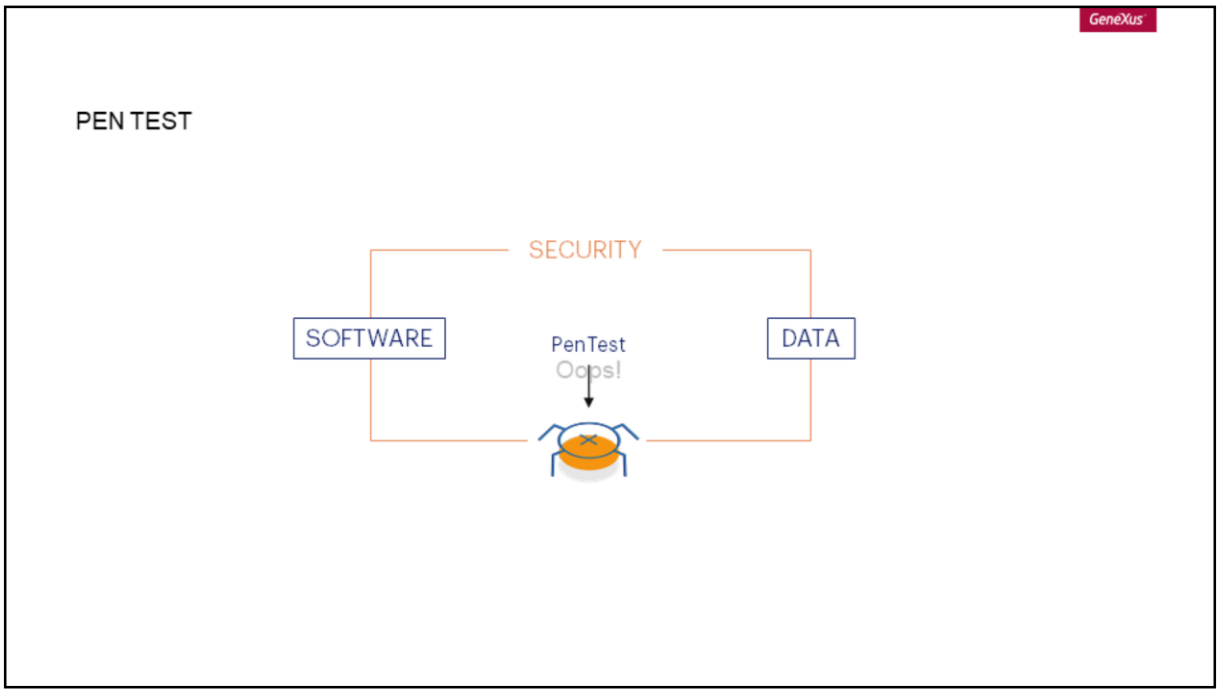
Uma **vulnerabilidade** é uma falha ou bug do sistema que algum agente pode aproveitar para acessá-lo de forma não autorizada, para roubar informações ou abusar dos recursos de tal maneira que não possa funcionar corretamente.



Utilizando boas práticas de programação é possível reduzir estes riscos,

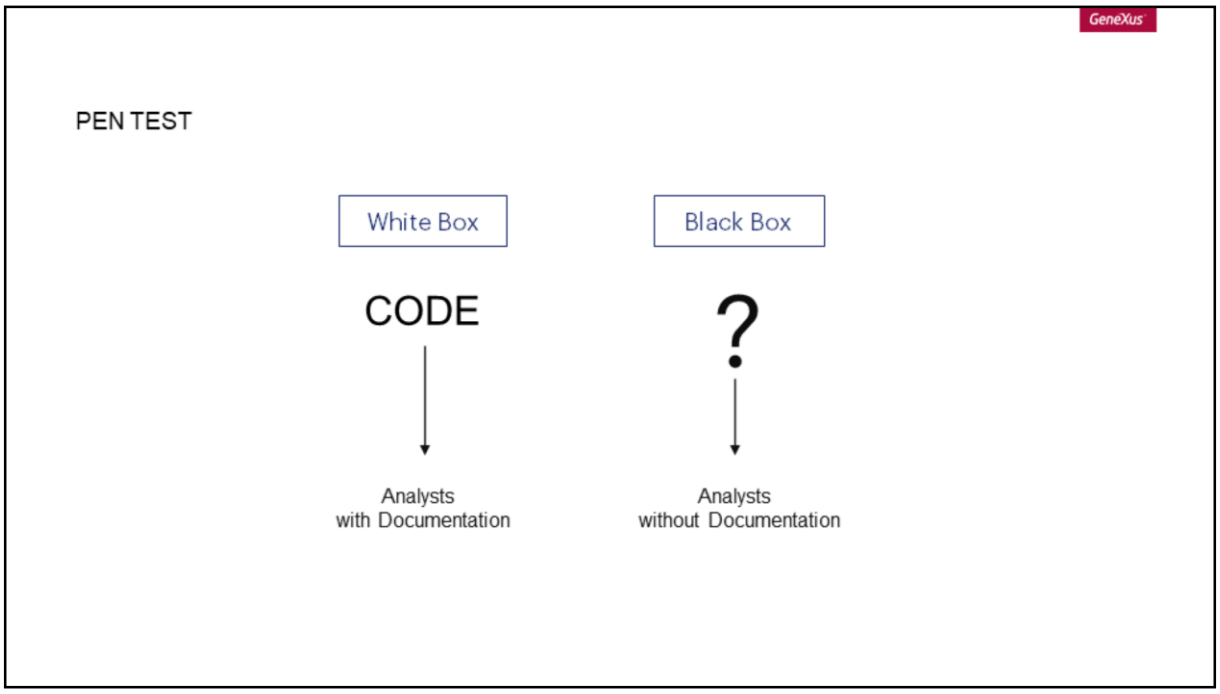


conforme proposto pelo Projeto Aberto de Segurança de Aplicações Web através de diretrizes, manuais e procedimentos de codificação disponíveis na web.



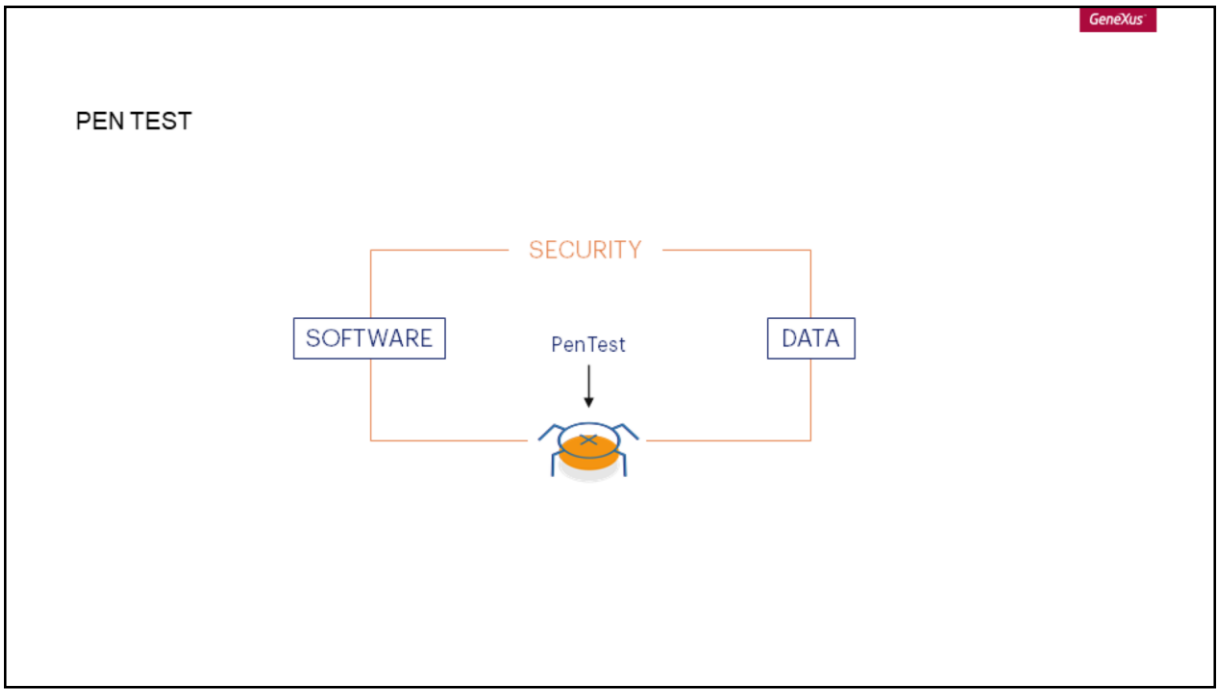
Neste sentido, a instância mais conhecida do ciclo de desenvolvimento seguro é o “pen test”.

Trata-se de uma instância de ataque simulado previamente autorizado para avaliar a segurança do sistema e identificar vulnerabilidades que não foram previstas durante a instância de desenvolvimento.



Os pen test podem ser caixa branca ou caixa preta, dependendo se o analista tem à sua disposição o código e a documentação do sistema ou não.

Para isso, geralmente são utilizadas ferramentas de análise automatizada de código estático ou dinâmico, cuja validade é limitada à interpretação do analista de segurança e ao contexto do sistema analisado.



É importante destacar que o pen test, apesar de ser a instância mais conhecida do ciclo de desenvolvimento seguro, não é a única. A correção de vulnerabilidades detectadas neste estágio é mais cara quando comparada a vulnerabilidades detectadas ou prevenidas em estágios iniciais.

Um pen test bem-sucedido e com bons resultados não garante que um sistema não tenha nenhuma vulnerabilidade, da mesma forma que um teste funcional não garante que um sistema não tenha nenhum bug.

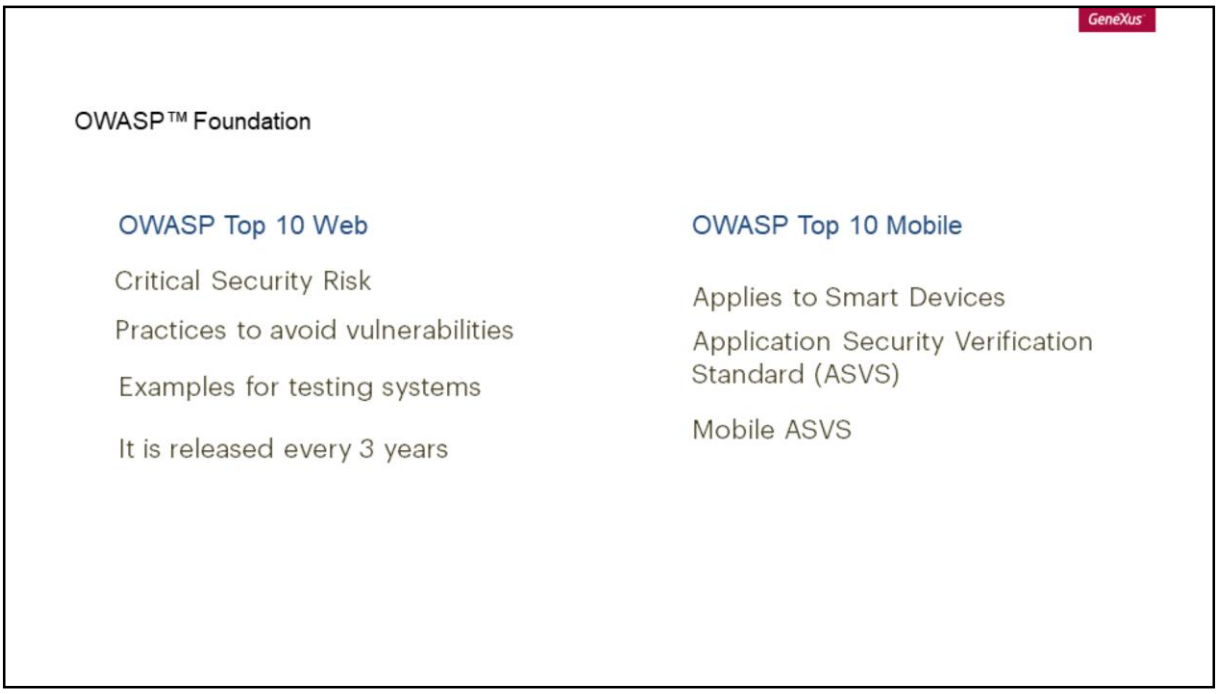
El Proyecto Abierto de Seguridad de Aplicaciones Web, es una comunidad abierta, sin fines de lucro y agnóstica de la tecnología dedicada a facilitar a las organizaciones el desarrollo, compra y mantenimiento de aplicaciones confiables.

GeneXus

OWASP™ Foundation

Standards information

Good practices



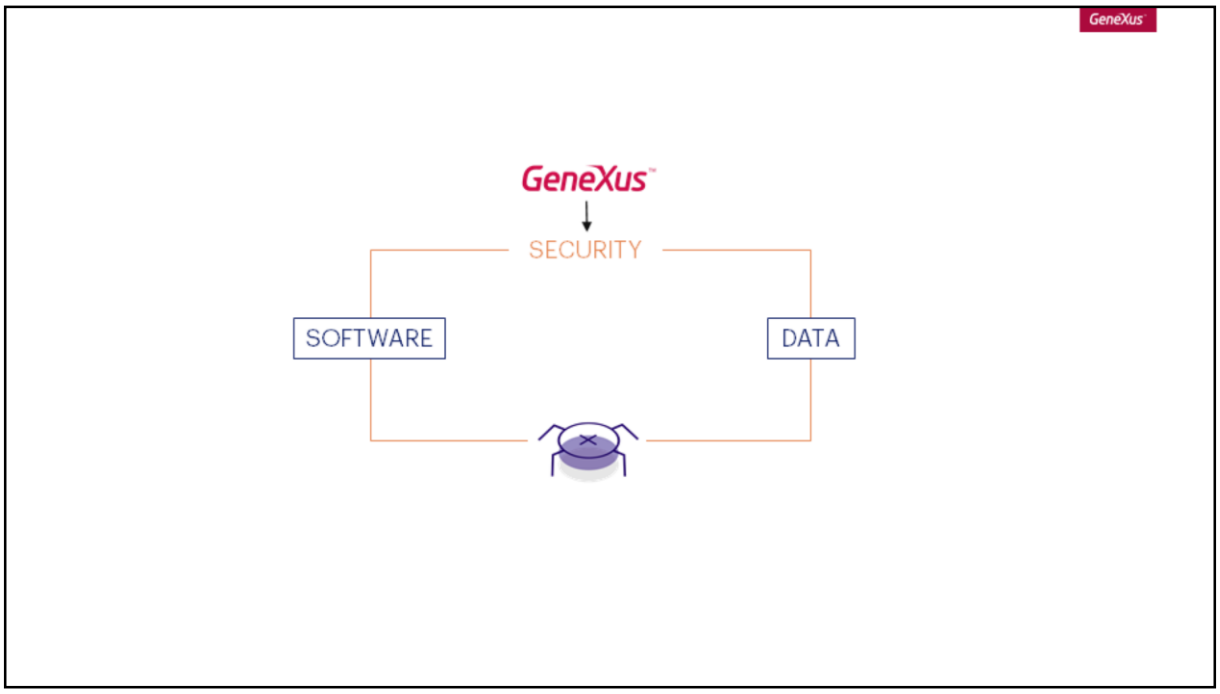
O Projeto Aberto de Segurança de Aplicações Web é uma comunidade aberta, sem fins lucrativos e independente da tecnologia, dedicada a facilitar para as organizações o desenvolvimento, compra e manutenção de aplicações confiáveis. Define e fornece informações sobre padrões, boas práticas e ferramentas para o desenvolvimento e a verificação de código e sistemas informáticos sob uma perspectiva de segurança.

Dentro do Projeto Aberto de Segurança de Aplicações Web, existem vários projetos. Um dos mais destacados e com maior relevância é o **Top 10 Web**, um documento que representa o consenso dos especialistas sobre os riscos de segurança mais críticos, devido ao seu impacto e frequência. O objetivo do documento é gerar consciência sobre o problema de segurança de sistemas, fornecer guias de boas práticas para evitar as vulnerabilidades listadas, fornecer exemplos para testar sistemas que possam ter essas vulnerabilidades e recomendar ferramentas de teste automatizado para sua análise.

Este Top 10 Web é lançado a cada 3 anos, sendo o último lançado e atualmente vigente, o correspondente ao ano de 2017.

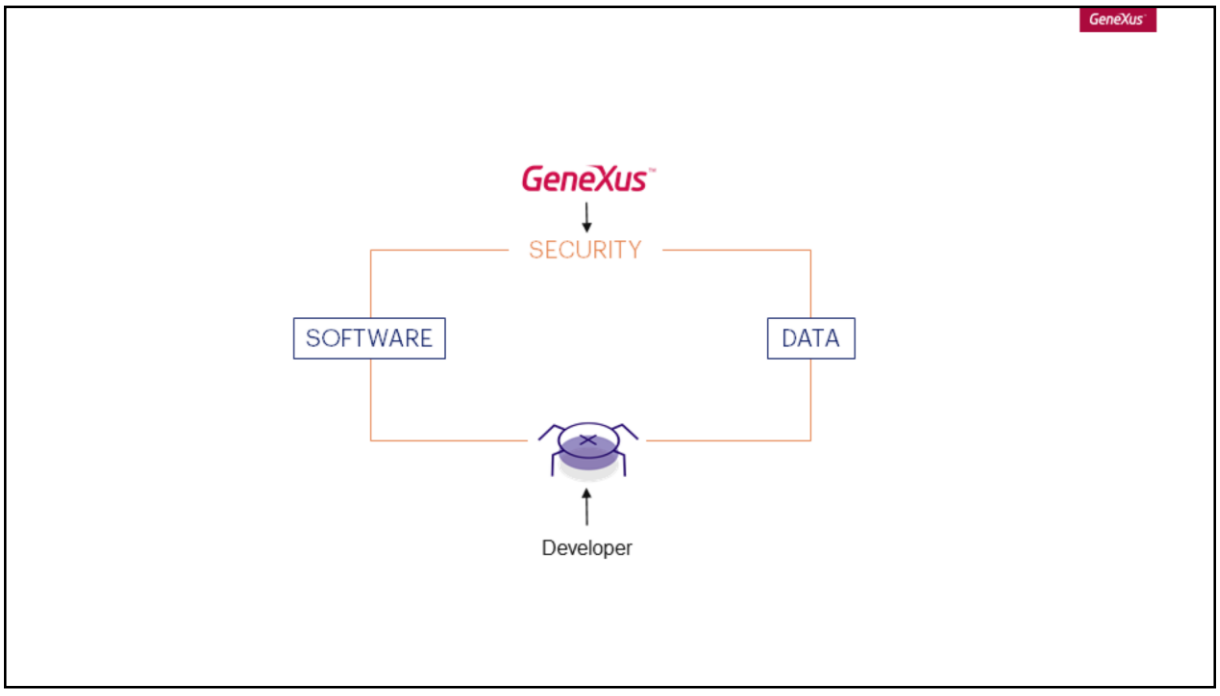
Existe também o **Top 10 Mobile**, sendo o último lançado o correspondente ao ano de 2016 que se aplica a Smart Devices. Além disso, o Projeto Aberto de Segurança de Aplicações Web inclui um projeto chamado *ASVS* que define níveis de verificação, também entendidos como requisitos de segurança mínimos e necessários, de acordo com o contexto da aplicação.

Há também o **Mobile ASVS**, sendo a versão vigente a 1.1 (para o ano de 2018).



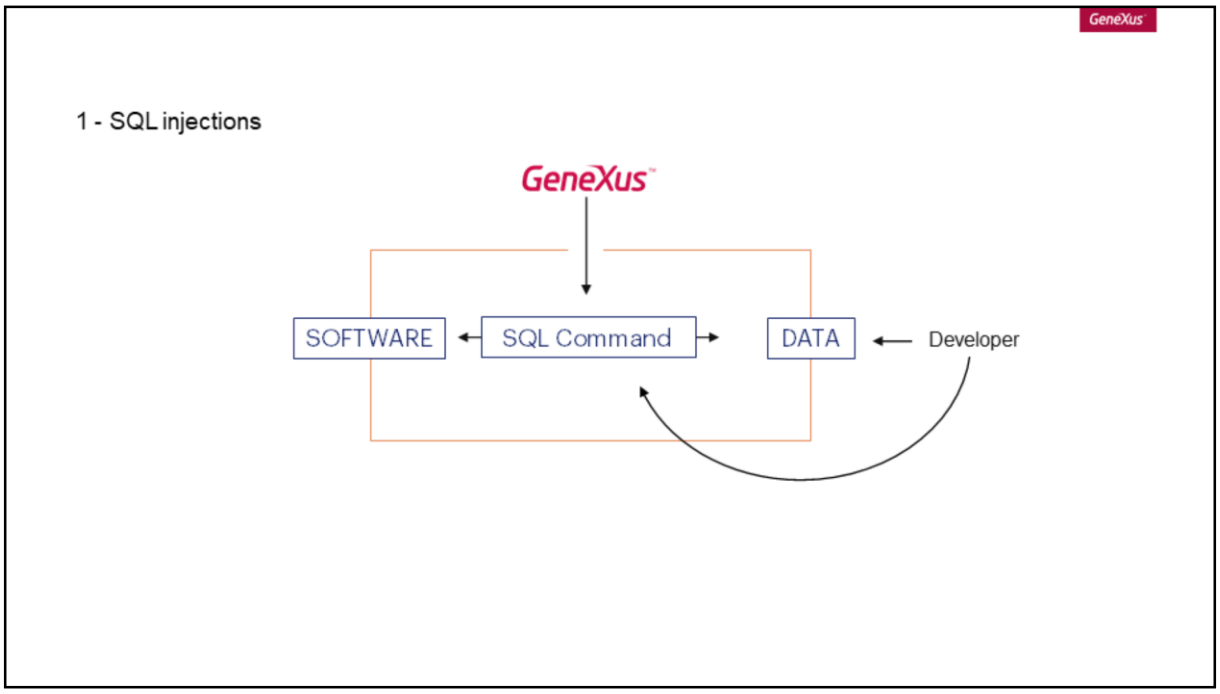
Agora, o que acontece com GeneXus e os controles de segurança?

GeneXus implementa automaticamente aqueles controles de segurança que podem ser inferidos a partir da base de conhecimento. Em outras palavras, aquelas vulnerabilidades provenientes de erros de codificação são atenuadas ao utilizar uma linguagem como GeneXus.



No entanto, ao utilizar a ferramenta, o desenvolvedor pode introduzir vulnerabilidades ou defeitos de segurança que se transformam em falhas e erros, tanto no nível da lógica de negócios quanto do código procedural GeneXus.

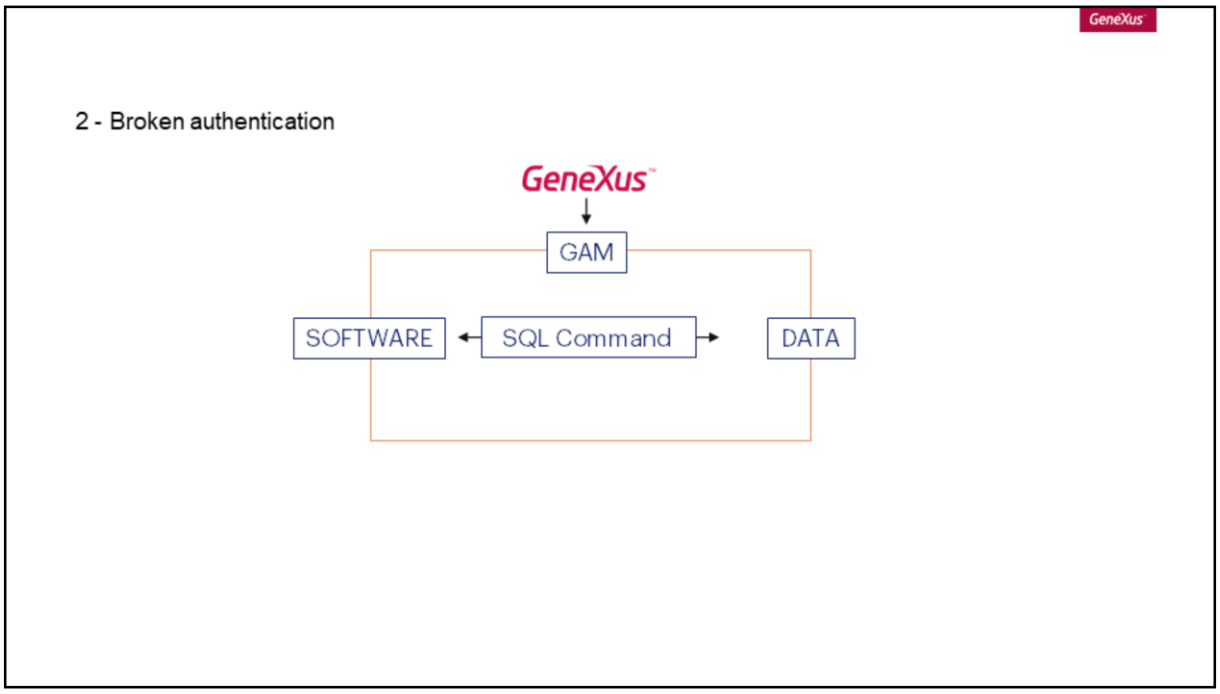
Por exemplo, usando o Top 10 2017 do Projeto Aberto de Segurança de Aplicações Web, é possível listar certas atenuações que GeneXus realiza automaticamente e casos nos quais o desenvolvedor pode introduzir falhas, apesar das medidas tomadas pelo GeneXus.



O primeiro item do Top 10 2017 são as injeções.

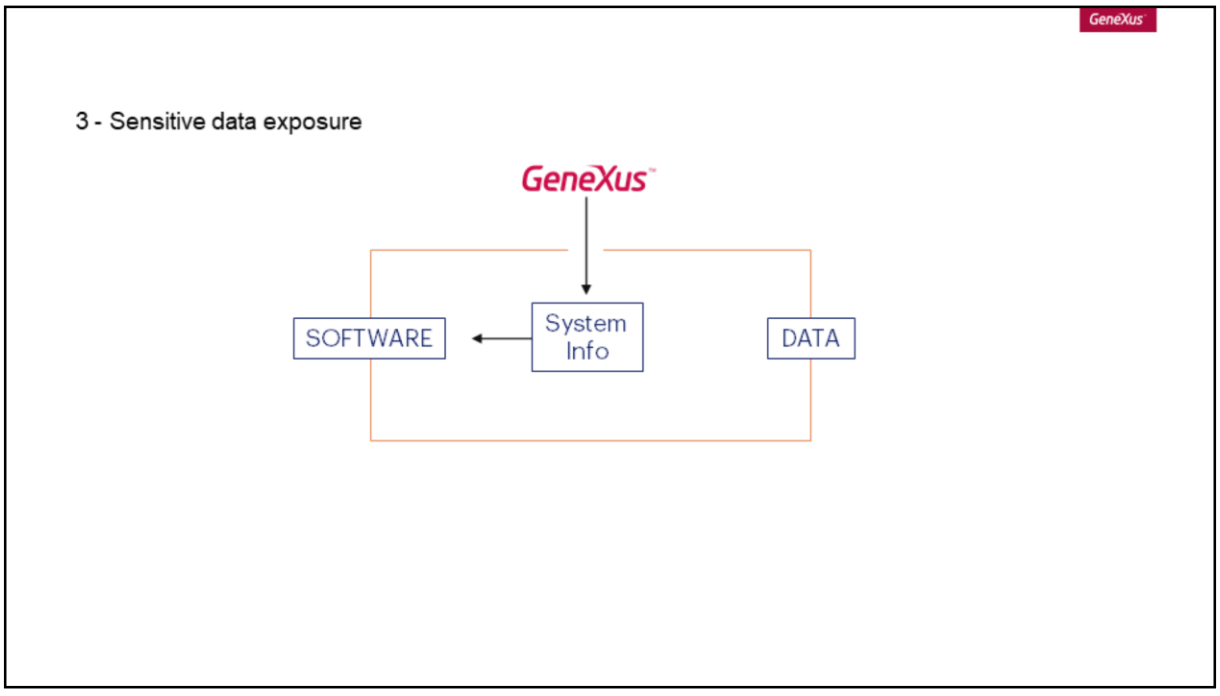
As mais conhecidas são as injeções de SQL, embora não sejam as únicas. Estas vulnerabilidades ocorrem quando informações provenientes do usuário são enviadas à base de dados sem validar.

GeneXus expõe o Comando SQL, que oferece mais versatilidade à aplicação, para que o desenvolvedor possa utilizar a base de dados como recurso. Neste caso, é o desenvolvedor quem deve controlar as entradas do comando para impedir uma injeção de SQL.



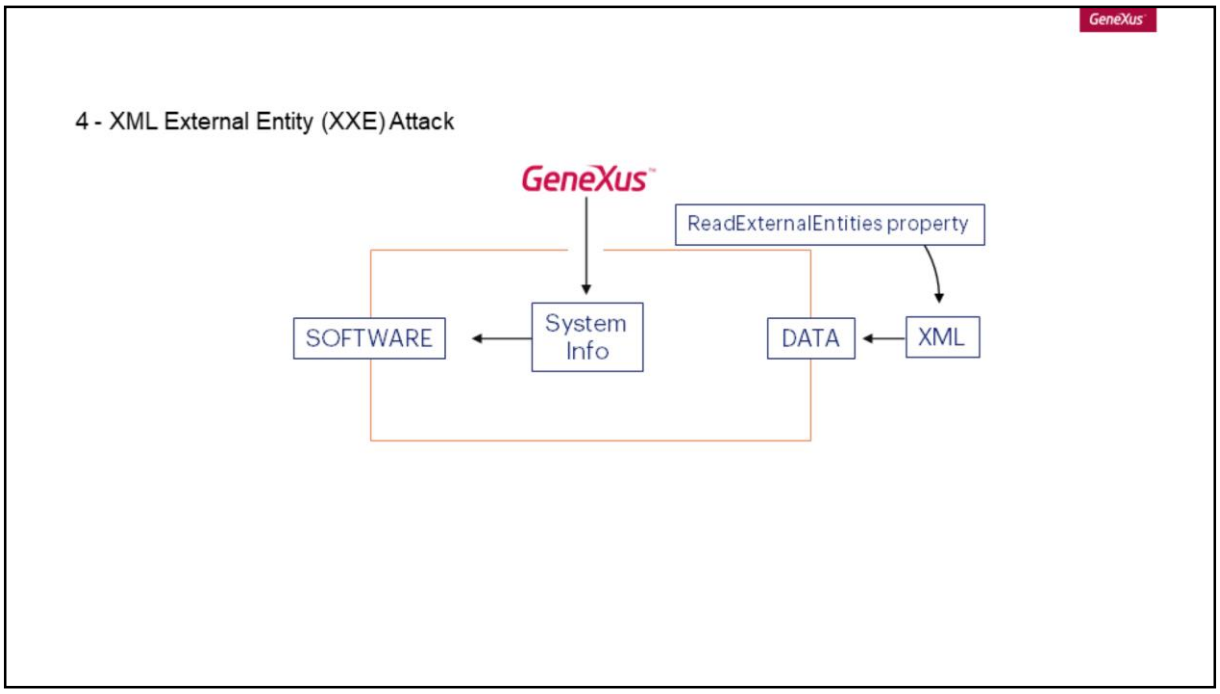
Quanto ao segundo item, a Broken authentication, GeneXus oferece um módulo de Autenticação/Autorização chamado GAM (***Genexus Access Manager***), permitindo executar estas tarefas automaticamente. A configuração adequada da ferramenta depende do desenvolvedor e de seu cenário específico.

Também é possível utilizar um módulo próprio, sendo responsabilidade do desenvolvedor o manuseio de senhas, sessões, etc.



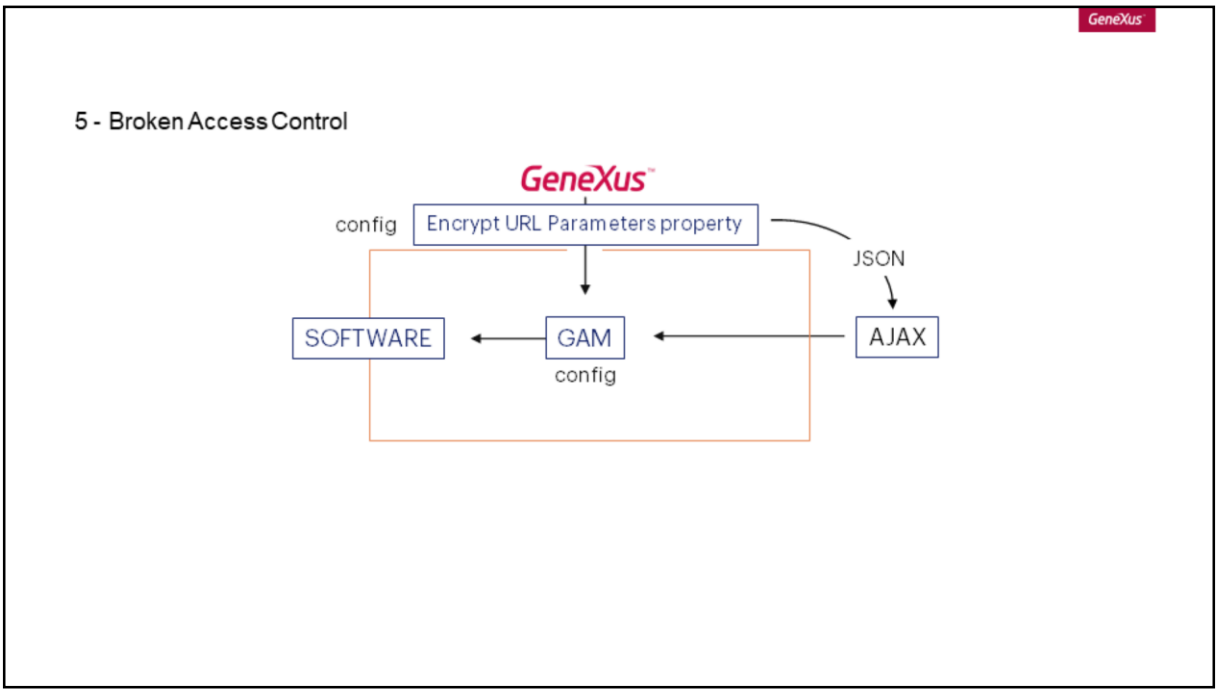
O terceiro ponto do Top 10 2017 corresponde à Exposição de Dados Sensíveis e faz parte da análise de requisitos determinar quais são as informações sensíveis do sistema.

Portanto, GeneXus não realiza nenhuma ação específica, mas fornece as funções necessárias para a proteção de informações sensíveis; como funções de criptografia, hash, gerenciamento de chaves e certificados.



O próximo ponto é o Ataque de uma Entidade externa XML.

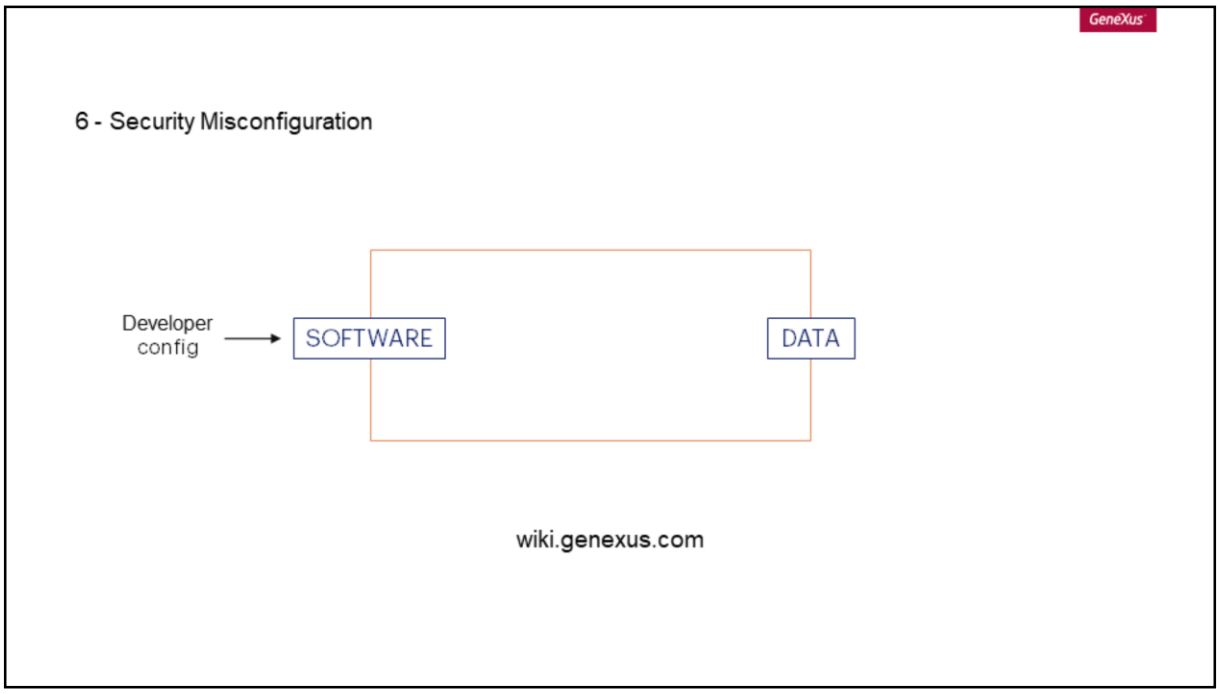
Neste caso, por padrão, GeneXus não processa nenhuma entidade externa de XML. Para habilitar o processamento de External Entities, é necessário configurar a propriedade ReadExternalEntities. Nesse caso, também é responsabilidade do desenvolvedor controlar o XML se cruza o limite do sistema.



O item seguinte corresponde ao Controle de Acesso Quebrado, que ocorre quando são concedidos acessos não autorizados ao sistema, alterando URLs, chamadas AJAX etc.

Para isto, GeneXus possui uma propriedade que permite criptografar parâmetro URL com a qual são criptografados todos os parâmetros que recebe um objeto, controla o acesso com GAM, executa a codificação para JSON nas chamadas AJAX por padrão e executa verificações de validação, não apenas do lado do cliente, mas também do lado do servidor.

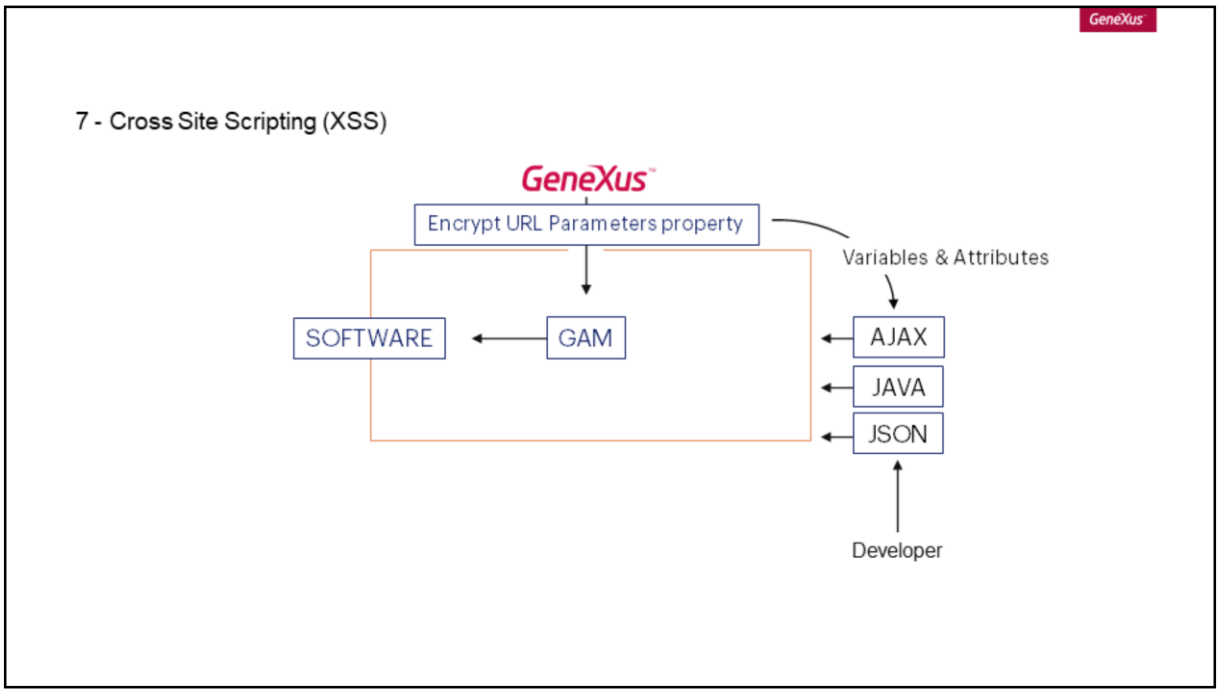
Neste caso, é responsabilidade do desenvolvedor adicionar controles nos eventos, configurar o GAM adequadamente e configurar também esta propriedade que permite criptografar os parâmetros URL..



O sexto ponto deste Top 10 corresponde às Configurações Incorretas de Segurança e refere-se principalmente a configurações incorretas no servidor de aplicações de produção. Para este caso, GeneXus possui algumas propriedades disponíveis que, embora não seja aconselhável desativar, o desenvolvedor pode ajustá-las para atender às necessidades do provedor.

Na Wiki Do GeneXus, é possível encontrar algumas dicas e diretrizes para uma configuração segura de servidores.

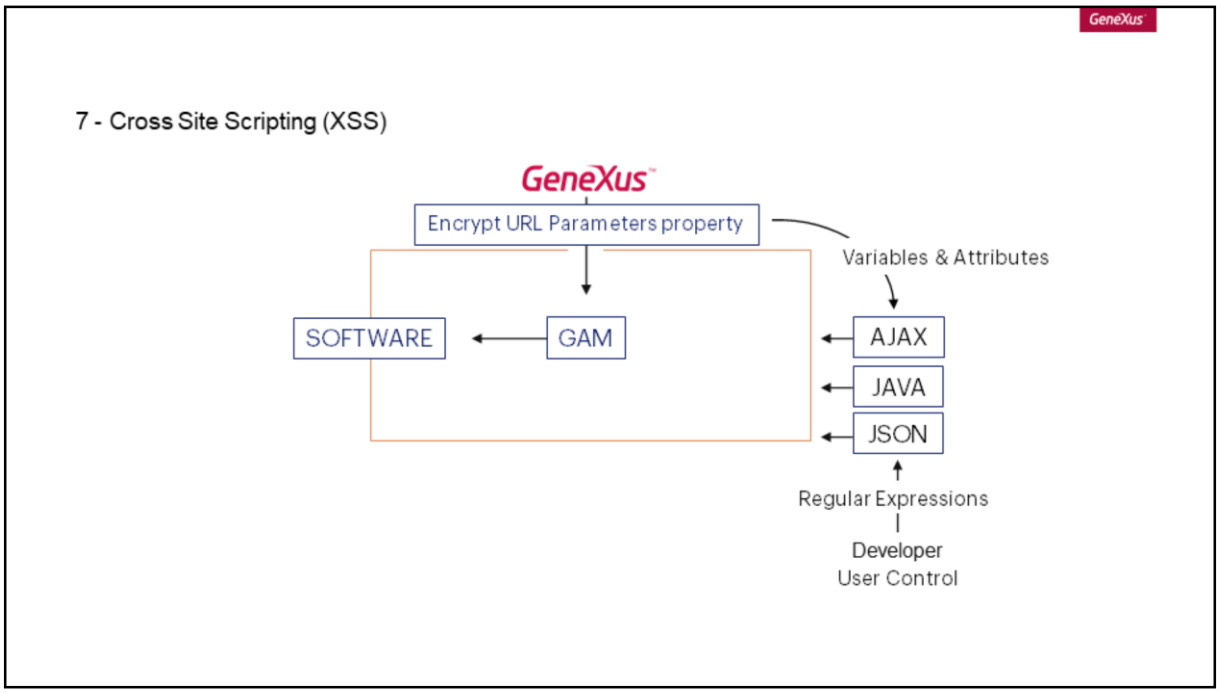
No entanto, o nível de segurança em produção depende em maior medida do responsável por infraestrutura que configura o servidor da aplicação.



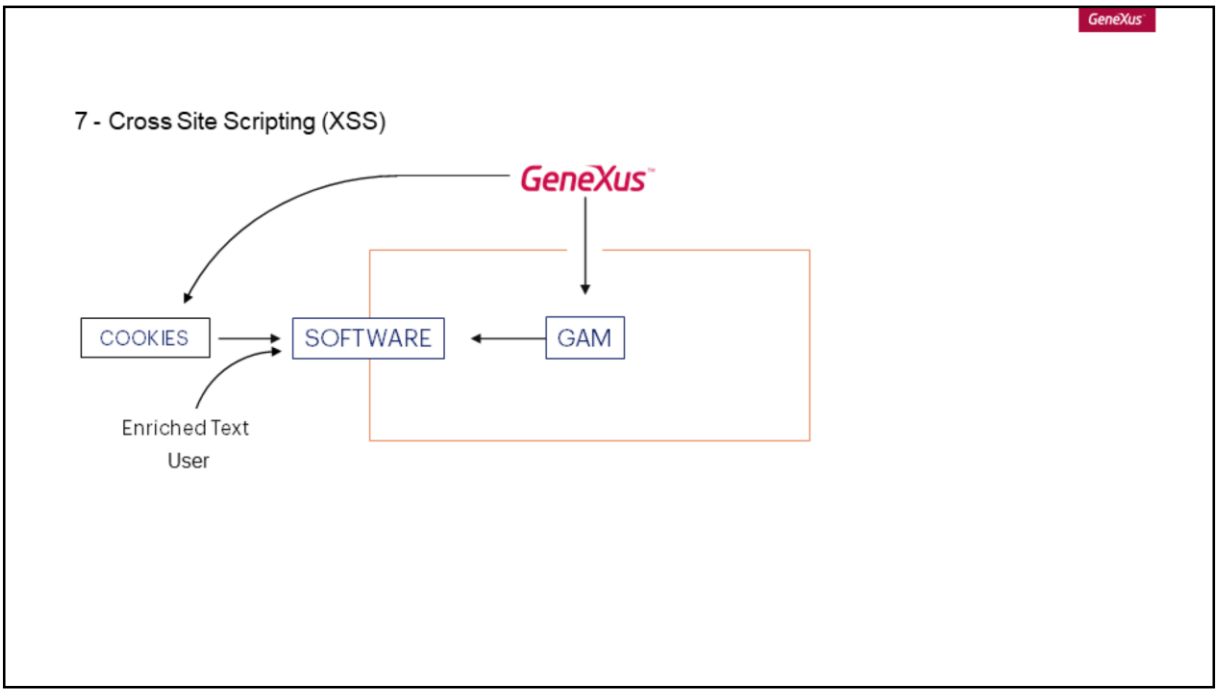
Como sétimo ponto do Top 10 2017 aparece a Sequência de Comandos nos Sites Cruzados.

Para este tipo de vulnerabilidades, GeneXus valida automaticamente o tamanho e o tipo das variáveis e atributos e executa a codificação dos dados que recebe o sistema de acordo com o seu contexto (HTML JavaScript, JSON etc. -- também automaticamente).

Caso o desenvolvedor insira código HTML ou Javascript próprio, é seu dever validá-lo e controlá-lo.

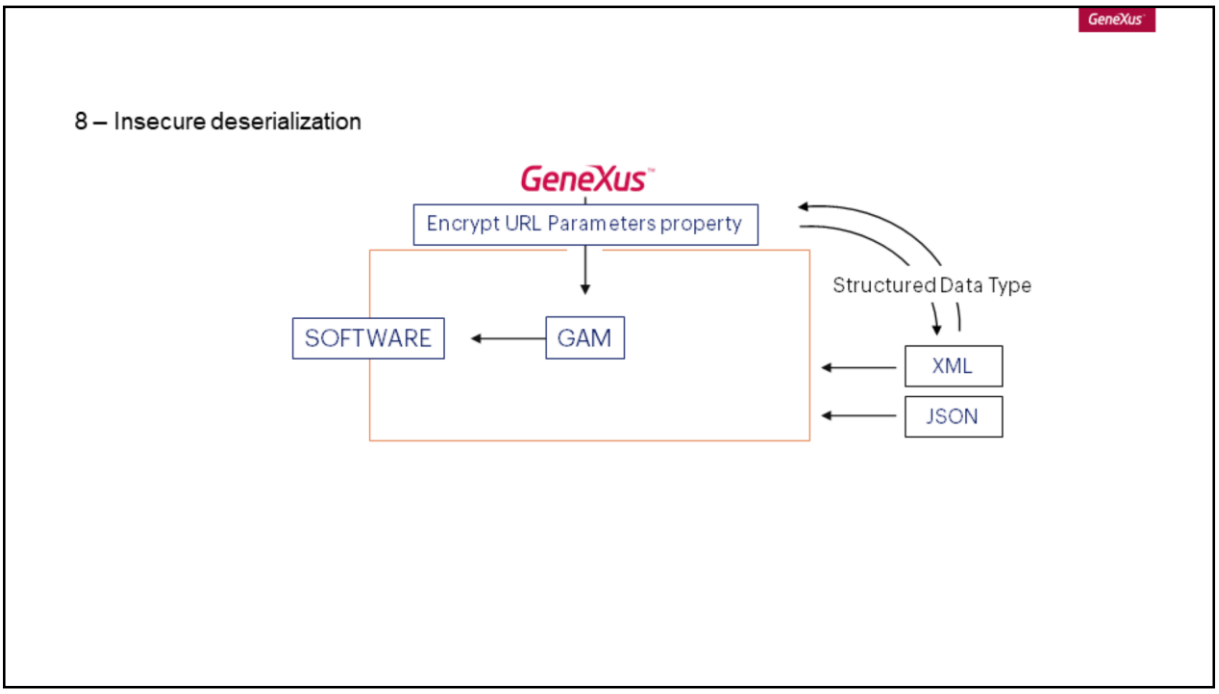


Da mesma forma, GeneXus oferece a possibilidade aos desenvolvedores de definir Expressões Regulares para validar as entradas dos usuários ou utilizar algum User Control próprio ou de terceiros (ou seja, externo ao GeneXus), desde que o desenvolvedor tenha total segurança e confiança que seja seguro, e esteja em conformidade com os controles de validação definidos previamente.



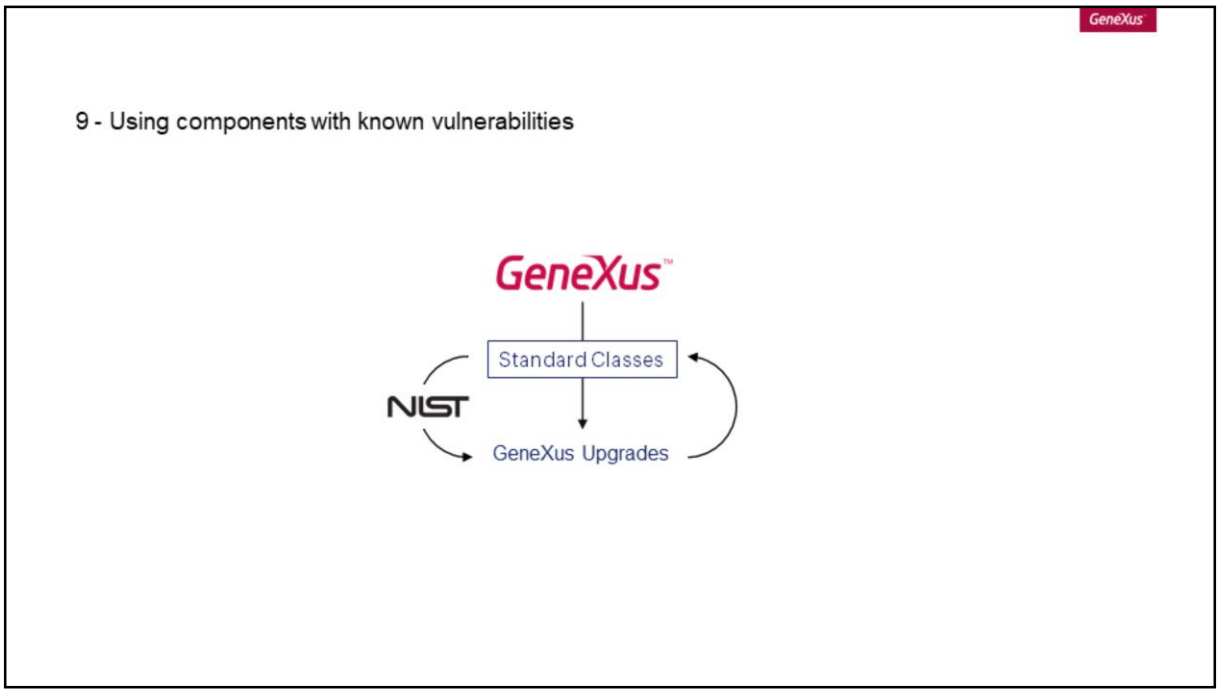
Também deve ser levada em consideração a configuração apropriada dos *cookies* e evitar desativar as verificações geradas automaticamente pelo GeneXus (se não é essencial).

No caso de admitir rich text por parte do usuário, nunca esquecer de controlar, validar e formatar estas informações de maneira segura e consistente.



O oitavo ponto refere-se à Desserialização Insegura, para a qual GeneXus implementa mecanismos de serialização seguros quando são tratados tipos de dados estruturados para JSON e/ou XML.

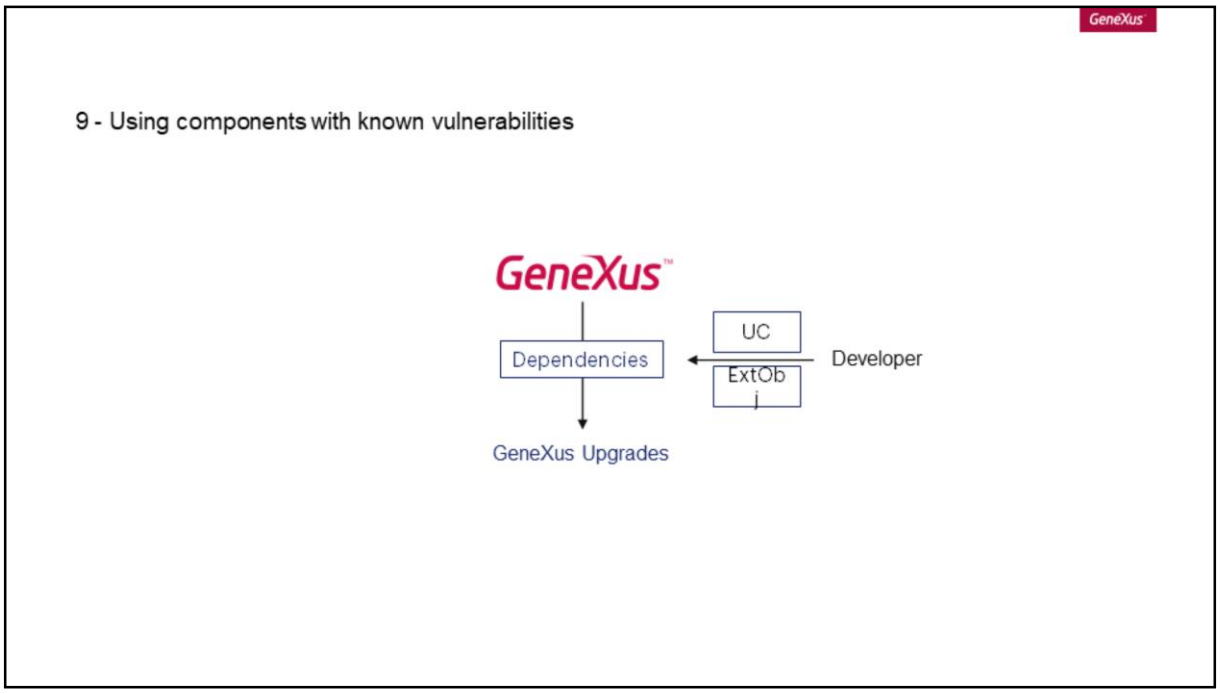
Quando são serializados objetos a partir de XML ou JSON utilizando os métodos ou tipos de dados, é necessário codificar as entradas.



O nono item da lista do Projeto Aberto de Segurança de Aplicações Web corresponde ao Uso de Componentes com Vulnerabilidades conhecidas.

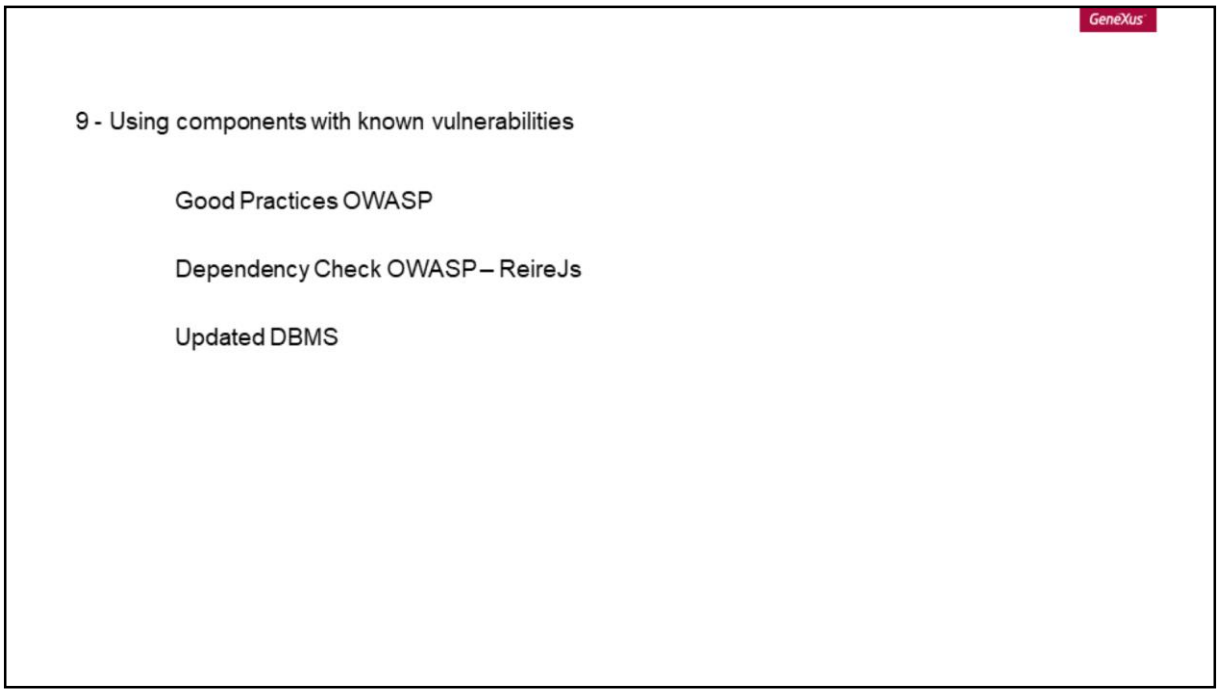
Para este caso GeneXus possui suas próprias classes padrão, que estão sendo constantemente revisadas e corrigidas nos sucessivos *upgrades*.

Mas também utiliza dependências de terceiros que são comparadas com a base de dados de vulnerabilidades do Instituto Nacional de Padrões e Tecnologia, sendo atualizadas, se necessário, em cada upgrade do GeneXus.



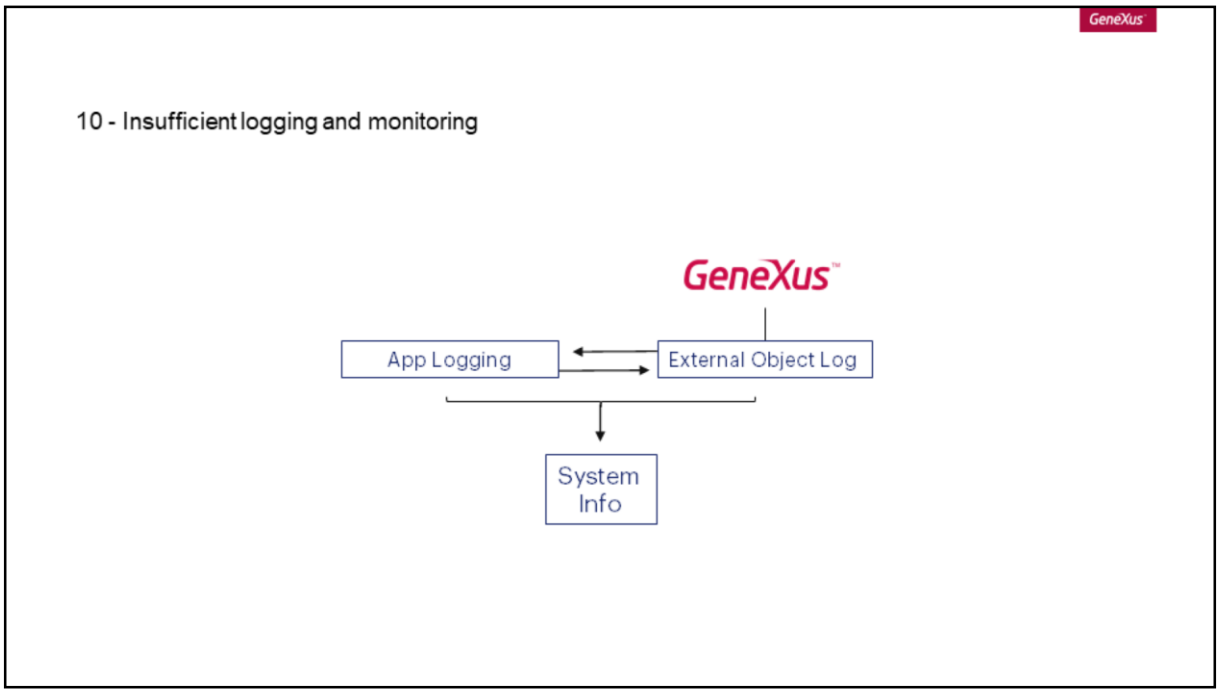
Ou seja, GeneXus se encarrega de suas próprias dependências, atualizando-as com os distintos upgrades.

Caso o desenvolvedor decida adicionar um User Control ou um External Object, é recomendável seguir as boas práticas de desenvolvimento seguro deste Projeto Aberto de Segurança de Aplicações Web e revisar as dependências incluídas.



É sugerido para isto o uso de diversas ferramentas; e também é recomendável manter os DBMS e seus drivers atualizados, bem como instalar atualizações e patches de segurança nos servidores.

.



O último ponto da lista corresponde ao Logging e Monitoramento Insuficientes.

Em todos os casos deve ser definido um processo de monitoramento para detectar possíveis eventos ou incidentes de segurança.

Embora GeneXus ofereça mecanismos de logging no nível da aplicação e o External Object Log para adicionar informações personalizadas ao log da aplicação, é necessário definir as informações relevantes fornecidas pelos diferentes dispositivos envolvidos (como os servidores web ou firewalls), e também definir os protocolos para a retransmissão e categorização das informações coletadas.

Desta forma podem ser detectados os eventos que ameaçam o sistema e, assim, impedir um posterior incidente.

GeneXus

Security Scanner

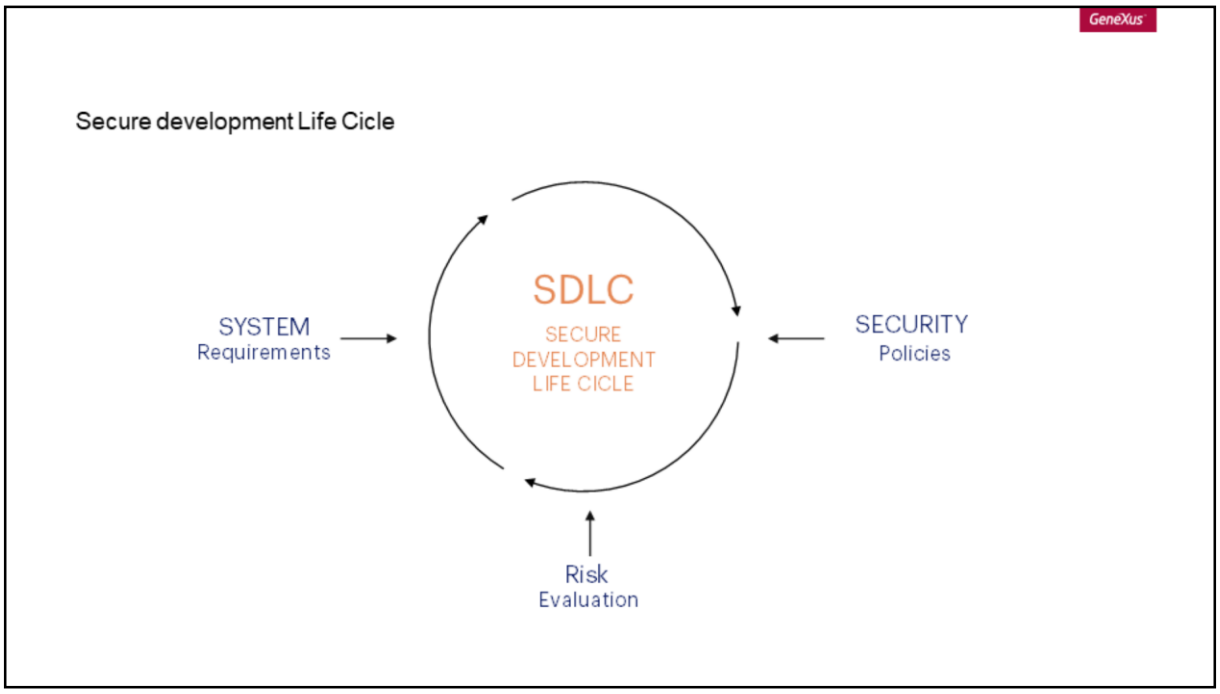
Para detectar casos conhecidos nos quais os desenvolvedores podem introduzir erros, é possível utilizar o Security Scanner.

Trata-se de uma Extensão para o IDE do GeneXus, disponível no Marketplace, que executa uma análise estática dos objetos da base de conhecimento em busca de sinais de problemas de segurança conhecidos.

A versão mais recente da extensão é compatível com o Genexus 16 e está atualizada para executar uma revisão com base no Top 10 2017 do Projeto Aberto de Segurança para Aplicações Web, com documentação disponível na Wiki.

GeneXus

Secure development Life Cicle



Para desenvolver sistemas seguros, é necessário implementar um **ciclo de vida seguro**;

Isto é, levar em consideração a segurança a partir do momento em que sejam coletados os requisitos do sistema. Isto é feito por meio de uma avaliação de riscos baseada em uma política de segurança que deve persistir durante todo o ciclo de desenvolvimento, ou seja, durante o desenho, implementação, teste, implantação e manutenção.

GeneXus™

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certificactions	training.genexus.com/certifications