Estudo dirigido etapa 1 – 2º bimestre/2017

Atividade

Prazo 17/10, por e-mail, 1 slide por item – resumido.

Em grupo - colocar o nome dos integrantes do grupo.

Identificar nos textos:

Motivação para desenvolvimento de software seguro

O que é desenvolvimento de sw seguro

Características de sw seguro

Metodologias, processos e ferramentas disponíveis para desenvolvimento de sw seguro

A Importância da Segurança no Desenvolvimento de Softwares

25 set 2014   | Ezequiel Juliano Muller http://www.ezequieljuliano.com.br/?p=73

[](http://www.ezequieljuliano.com.br/wp-content/uploads/2014/09/seguranca_web_edefense.jpg)

A cada dia um número maior de computadores está conectado a Internet e junto a esta expansão crescem também os ataques aos sistemas informatizados, colocando softwares, organizações e usuários em grande risco. Mesmo as organizações adotando mecanismos de controles e políticas de segurança, o software sempre foi e será o vetor de ataques.

Oferecer um software seguro é obrigação do desenvolvedor e para que este objetivo seja alcançado é necessário avaliar a segurança de todo o ciclo de vida de desenvolvimento da aplicação. Esta avaliação melhora os produtos de TI, pois identifica erros ou vulnerabilidades que podem ser corrigidas pelo desenvolvedor, reduzindo a probabilidade de futuras falhas de segurança, além disso, faz o desenvolvedor tomar mais cuidado com a estrutura e desenvolvimento do software.

Para ter segurança no desenvolvimento é necessário manter fontes em segurança, evitando roubos ou indisponibilidade da equipe de desenvolvimento, seguir especificações de segurança evitando vulnerabilidades que comprometam a segurança e oferecer garantias aos clientes quanto à segurança e integridade da aplicação.

Um ambiente de desenvolvimento seguro se caracteriza por ter um espaço físico restrito, com controle de acesso físico e proteção lógica dos servidores, separação entre ambiente de desenvolvimento, teste e construção (build), gerência de configuração dos fontes e processos de desenvolvimento bem estabelecidos e controlados.

A segurança no desenvolvimento do software é alcançada ao se adotar normas e práticas de boa programação como funções intrinsecamente seguras, verificar os códigos de erro retornado por funções ou métodos, atentar para o tamanho de buffers e arrays do software e documentar o código. A garantia desta segurança é obtida com a construção do software conforme o especificado e na execução e validação de testes para verificar se atende às especificações iniciais.

A segurança no desenvolvimento de softwares está apoiada na norma ISO/IEC 15.408 (Common Criteria for Information Technology Security Evaluation) também chamada de Common Criteria ou CC. Esta norma trata-se de uma metodologia para desenvolvimento de software seguro e possui como objetivo fornecer um conjunto de critérios fixos que permitem especificar a segurança da aplicação de forma não ambígua a partir de características do ambiente e definir formas de garantir a segurança do sistema para o cliente final, sendo definida em quatro níveis de garantia da segurança (EAL – Evaluation Assurance Level): EAL 1 – testado funcionalmente; EAL 2 – testado estruturalmente; EAL 3 – metodicamente testado e verificado e EAL 4 – metodicamente projetado, testado e verificado.

**Referências**

BRASIL. Núcleo de Informação e Coordenação do Ponto Br. **Análise e Interpretação de Logs**. 2003. Disponível em: <http://www.cert.br/docs/palestras/nbso-gter15-tutorial2003.pdf>. Acesso em: 12 fev. 2013.

REGO, Bruno M.; BROSSO, Inês**. Segurança no Desenvolvimento de Sistemas com Metodologia Ágil SCRUM**. Disponível em: <http://www.slideshare.net/BrunoMottaRego/segurana-no-desenvolvimento-de-sistemas-com-metodologia-gil-scrum>. Acesso em: 12 fev. 2013.

Importância do Desenvolvimento de Software Seguro

[](http://2.bp.blogspot.com/-bjxyttYJIQQ/Uheooxue5NI/AAAAAAAAAdI/JUA11CeCTWE/s1600/%C3%8Dndice.jpg)

By Daniel Willian *http://informaticainteligente.blogspot.com.br/2013/08/importancia-do-desenvolvimento-de.html*

É importante que o software seja desenvolvido com segurança, a fim de diminuir o número de vulnerabilidades, que a cada ano vem crescendo. Para tanto, os requisitos de segurança da informação precisam ser incorporados em cada etapa do desenvolvimento, desde o início do desenvolvimento de software até a etapa de implantação, pois quanto mais cedo forem identificadas as vulnerabilidades, menores serão os gastos no projeto, sendo de suma importância que os níveis de segurança de um produto sejam mensurados de acordo com o risco que apresenta.

No desenvolvimento do software é importante conhecer as vulnerabilidades de cada etapa do desenvolvimento para que as ameaças iminentes possam ser removidas da forma mais célere possível, eliminando assim os riscos da má gestão de custos e da descoberta de vulnerabilidades já na finalização do software. Quando a empresa sofre essas ameaças (ataques ou tentativas de ataques) é importante coletar as situações anteriores, a fim de que, os dados possam ser analisados sempre que um novo sistema for desenvolvido.

Como já dito, os níveis de segurança alteram de acordo com o âmbito de negócio que a empresa possui, logo é importante é importante reunir os *\*stakeholders*para propor quais serão os critérios de avaliação para a implantação da segurança e quais serão os níveis de segurança.

Para que os problemas no desenvolvimento de softwares sejam atenuados, é preciso adotar padrões de códigos, normas e manuais de segurança a fim de que se possam evitar erros no código fonte, além da etapa de testes que será muito importante a cada nova implementação de levantamento de requisitos.

\*stakeholders - **parte interessada**ou **interveniente.**

# A importância de desenvolver sistemas seguros, do projeto à produção: Mais um treinamento com certificação internacional da Academia Clavis

[MARÇO 17, 2016](https://seginfo.com.br/2016/03/17/a-importancia-de-desenvolver-sistemas-seguros-do-projeto-a-producao-mais-um-treinamento-com-certificacao-internacional-da-academia-clavis/) https://seginfo.com.br/2016/03/17/a-importancia-de-desenvolver-sistemas-seguros-do-projeto-a-producao-mais-um-treinamento-com-certificacao-internacional-da-academia-clavis/

Quando falamos sobre brechas de segurança, que acabam resultando em ataques, é comum imaginar o atacante driblando mecanismos de proteção de perímetro (como firewalls) ou explorando falhas em sistemas operacionais. Isso não deixa de ser verdade, mas ao tomarmos essa imagem como padrão, deixamos de perceber que as falhas podem estar nos mais variados ativos de uma companhia e um dos tipos de ativos mais vulneráveis nas empresas são os seus próprios sistemas, aqueles que são desenvolvidos internamente, seja por colaboradores ou por fábricas de software.

O problema é muito sério, tanto que a maioria das normas e padrões de segurança em vigor no mundo como, por exemplo, o PCI DSS, dedicam uma seção ao tema da aplicação de segurança no processo de desenvolvimento de software. No lado das empresas, ainda há uma crença de que segurança no processo de produção de software é algo que somente tem a ver com a produção de código seguro, ou que aplicar segurança ao software é uma atividade a ser feita somente ao final do projeto, quando o sistema está pronto.

Na realidade, a atividade de aplicar segurança ao software é algo que começa na concepção do sistema – na fase de requisitos – e acompanha todo o projeto, até a sua entrega em produção. Definir controles de segurança nas fases finais do desenvolvimento atrapalha todo o projeto, incorrendo no aumento de esforço, custo e atrasos, sem dar a garantia de que o software estará protegido contra ataques.

Felizmente os desafios sempre atraem pessoas interessadas em se diferenciar no mercado. Assim se desenvolveu uma especialidade totalmente nova na indústria de segurança da informação, formada por pessoas que querem implementar controles de segurança em todo o processo de desenvolvimento, se tornando especialistas em fortalecer sistemas.

(continua)

**Qual é o processo de desenvolvimento de software seguro?**

4 de novembro de 2015/ Conviso / http://blog.conviso.com.br/qual-e-o-processo-de-desenvolvimento-de-software-seguro/

Durante o desenvolvimento de um software, os gerentes de TI buscam garantir a entrega de um produto com uma baixa quantidade de bugs no menor tempo possível. No entanto, durante esse processo, não são raros os casos em que gestores ignoram práticas de segurança que evitam, no futuro, vazamento de dados sigilosos. Diante desse cenário, é fundamental que certas rotinas sejam tomadas, ainda que os modelos de criação de software mais tradicionais não sejam focados no desenvolvimento de software seguro.

Nem todos os modelos de desenvolvimento garantem um bom sistema

Peguemos como exemplo o Modelo em Cascata (ou Waterfall Mode, em inglês). Por ser um dos primeiros modelos a serem criados, foi altamente adotado, ainda que possuísse falhas. Suas etapas são:

* verificação dos requerimentos de software e sistema;
* analise e criação da estrutura do software
* desenvolvimento do código do programa;
* teste, depuração e busca por erros;
* instalação, suporte e manutenção de todos os sistemas interligados ao software.

Ainda que algumas versões modificadas do modelo tenham sido criadas, a original peca pela falta de uma etapa em que são implementadas políticas e rotinas de segurança de software. Já modelos como o espiral são focados na capacidade da equipe para desenvolver um programa com um bom aproveitamento do tempo e do esforço de todos os profissionais envolvidos. Dessa forma, é possível eliminar rapidamente elementos que podem atrasar a criação do sistema ou diminuir a produtividade do time de desenvolvedores.

Adotando boas práticas de segurança de software

A adoção de um modelo de desenvolvimento tradicional não deve ser visto como um motivo para a não implementação de rotinas de segurança. Ao buscar práticas que permitam o desenvolvimento de um software estável e com poucas falhas, o gerente de TI faz com que o seu sistema seja lançado sem falhas graves de segurança. Isso diminui a possibilidade dos dados de segurança do usuário — e da sua empresa — ficarem vulneráveis.

Em um mundo onde pessoas mal-intencionadas trabalham buscando por brechas que permitam a obtenção de dados pessoais e corporativos, o cuidado com a segurança deve passar por todas as etapas de criação de um software e ser incorporado no cotidiano dos profissionais de TI. Veja abaixo alguns exemplos de como tornar o seu sistema mais seguro:

* Na etapa de classificação de requisitos e formação da equipe de desenvolvimento, é necessário identificar a necessidade de utilização de métodos de criptografia, segurança de usuários, treinamento e, em alguns casos, questões legais envolvendo licenciamento e manipulação de dados;
* Quando a arquitetura do software for projetada, é necessário levar em conta o nível de segurança da arquitetura e da linguagem escolhida, as possíveis vulnerabilidades que o sistema pode apresentar e quando e como os métodos de autenticação e envio de dados seguro serão utilizados;
* Procure oferecer uma documentação com todos os requerimentos de segurança listados para a sua equipe. Além disso, garanta que o seu time trabalhe de acordo com as definições de segurança do projeto, usando as ferramentas da arquitetura e buscando vulnerabilidades da maneira correta;
* Antes de lançar o seu sistema para o público geral, faça testes nos seus métodos de segurança e tente pensar como alguém que deseja comprometer a segurança do sistema. Nessa etapa, é importante garantir que todos os blocos de código que envolvem informações sensíveis não tenham brechas que possam comprometer a segurança do usuário;
* Por fim, quando o produto for finalmente adotado pelo mercado, continue fazendo monitoramento em busca de falhas de segurança. Permita (e incentive) que o seu usuário envie logs com dados de uso e falhas de software. E caso alguma brecha seja encontrada, trabalhe com o seu time para que um patch de segurança seja lançado o mais rápido possível.

Prevenir é melhor que remediar

Com a dinamização dos modelos tecnológicos e de gestão, empresas de desenvolvimento e consultoria de software passaram a ter um papel protagonista na busca por falhas de sistema. Se antes isso era conquistado por hackers que buscavam vulnerabilidades para obtenção de senhas e outros dados privados, hoje as desenvolvedoras trabalham na busca por brechas diversas. Para que essa nova postura ganhasse força, requisitos regulatórios foram criados buscando responsabilizar os criadores — e não os usuários — pelos problemas causados por um software não seguro. Não tenha medo de adaptar as rotinas e protocolos existentes a sua necessidade.

Na era da internet das coisas, a tecnologia tornará a nossa vida melhor dentro e fora do ambiente de trabalho. Nesse cenário em que softwares serão responsáveis pela manipulação de um número cada vez maior de informações, garantir que esses dados estejam seguros é fundamental para as empresas e para os seus usuários. Ao adotar boas práticas que levam ao desenvolvimento de software seguro, o gerente de TI foca os seus recursos na criação e manutenção de um sistema benéfico para a sua empresa e para todos os que dela dependem. Ainda que os desafios estejam cada vez maiores, um sistema que deixe a organização menos vulnerável a ataques criminosos gera menos gastos com contenção e resolução de problemas.

Conhece alguma dica de segurança que não foi apresentada no post? A sua empresa já pratica essas rotinas? Compartilhe a sua experiência conosco!

Introdução ao desenvolvimento de software com foco em segurança

## http://www.devmedia.com.br/introducao-ao-desenvolvimento-de-software-com-foco-em-seguranca/28161

## **Veja neste artigo os fundamentos de segurança da informação aplicados ao desenvolvimento de software.**



A qualidade de software pode ser medida de várias formas. Entre as informações de qualidade, a NBR ISO/IEC 9126-1 destaca diversos itens, entre eles o de confiabilidade, que está relacionado a maturidade, tolerância a falha e recuperabilidade. Além disso, há no item 6.1.4 Segurança da Informação, que é a capacidade do produto de software de proteger informações e dados, de forma que pessoas ou sistemas não autorizados não possam lê-los nem os modificar e que não seja negado o acesso às pessoas ou sistemas autorizados. Estes itens fazem parte não apenas do desenvolvimento de software, mas também da segurança da informação como um todo.

A segurança da informação é essencial para as organizações. O Brasil, segundo o IPEA, é um dos países mais vulneráveis do mundo, por tanto, é necessária a implementação de processos de segurança e boas práticas de desenvolvimento com foco em segurança. A NBR ISO/IEC 27002 é um guia para as práticas de segurança, servindo tanto para o desenvolvimento como para o administrador de redes. A norma define segurança como sendo o conjunto de preservação da confidencialidade, integridade e disponibilidade, além disso, especialistas adicionam a autenticidade, formando a sigla ACID.

O objetivo é proteger um dos ativos mais importantes da organização, a informação, e para isso, é importante saber o valor da informação, para determinar o investimento em tempo e recurso para proteger tal informação. Mas para os desenvolvedores, é sempre importante desenvolver focado na segurança, evitando-se assim que incidentes de segurança ocorram.

“Um incidente de segurança da informação é indicado por um simples ou por uma série de eventos de segurança da informação indesejados ou inesperados, que tenham uma grande probabilidade de comprometer as operações do negócio e ameaçar a segurança da informação” ABNT NBR ISO/IEC 27002, pag. 2.

Para evitar os incidentes, devemos desenvolver buscando atender aos critérios dos pilares da segurança, para isso, vamos entender o que significa cada um dos pilares:

Segundo Alex:

**Confidencialidade**: É a garantia de que a informação é acessível somente por pessoas autorizadas;

**Integridade**: É a salvaguarda da exatidão da informação e dos métodos de processamento;

**Disponibilidade**: É a garantia de que os usuários autorizados obtenham acesso à informação e aos ativos correspondentes sempre que necessário;

Além dos três citados, temos a **autenticidade**, que completa a confidencialidade, visa estabelecer a validade da transmissão, da mensagem e do seu remetente. O objetivo é que o destinatário possa comprovar a origem e autoria de um determinado documento.

Atendendo os pilares dentro da aplicação

Mesmo sabendo quais são os pilares, a preocupação agora é como desenvolver sistema em conformidade com as normas de segurança?

Vamos iniciar pela disponibilidade, ou seja, a aplicação deve estar disponível. Isto quer dizer que se a empresa trabalha 24 horas, o sistema estará disponível 24 horas. Parte desta disponibilidade está vinculada a rede, servidores, etc., mas podemos utilizar também sistemas em nuvem, de forma que a preocupação seja apenas ter Internet no momento do acesso.

Entre as soluções mundiais de nuvem mais conhecidas temos a Amazon e a mais nova solução Livre, com recursos gratuitos, disponibilizada pela RedHat, a OpenShift.

Outra solução é a implementação de cluster´s Java, neste artigo, é possível aprender como configurar um cluster Tomcat: Configurando um Cluster de Tomcat com Balanceamento de Carga. Além disso, temos no artigo de Abraham como montar cluster JEE: J2EE clustering.

Claro que sempre é necessário analisar os pontos positivos de cada solução, manter a aplicação em clusters dentro da empresa ou em nuvem?

Agora que já temos a disponibilidade, é necessário ampliar a integridade.

Esta implementação, pode ser criada em diversos pontos da aplicação, entre os locais temos a integridade dos dados para a persistência, que pode ser utilizado o Beans Validation. No artigo Beans Validation - Solucionando problemas de integridade dos dados persistentes da Francieli, é apresentado como utilizar este recurso.

Além disso, é possível a implementação de recursos de validação de integridade no banco de dados e através de regras nos métodos.

Passado a integridade e a disponibilidade, vamos ao próximo pilar a confiabilidade, que serve para garantir que apenas pessoas autorizadas acessem informações. Isto é possível criando sistemas de login. O Thiago apresenta algumas formas de criar login em Java e também há a possibilidade de utilização do Spring Security que permite a customização da autenticidade e controle de acesso. Acesse:

* Spring Security
* Artigo Java Magazine 69 - Spring Security
* Tutorial: Começando com Spring Security
* Segurança com Spring Security 3.0 utilizando banco de dados em apenas alguns minutos

Por último, mas não menos importante a autenticidade, para isso, é possível utilizar certificados e assinaturas digitais, neste artigo é possível ver a implementação de certificados digitais (Autenticação de Usuários com Certificados Digitais, Certificação Digital com Java - Revista Java Magazine 110 ) e para a assinatura digital veja ASSINATURA E VALIDAÇÃO DIGITAL DE DOCUMENTOS EM JAVA.

Conclusão

Vimos que segurança é importante e que o Brasil necessita evoluir muito no quesito implementação e gestão de segurança, foi apresentado então como atender aos pilares da segurança com o objetivo de melhorar a segurança da informação.

Referência

* ABNT NBR ISO/IEC 27002.