Processo de desenvolvimento de software seguro através da identificação de níveis de segurança

Rosana Wagner, Josiane Fontoura dos Anjos Brandolt, Fábio Diniz Rossi
Instituto Federal Farroupilha
Campus Alegrete

{rosanawagner, josianefab, fdrossi@al.iffarroupilha.edu.br}

Resumo: *As organizações enfrentam uma série de dificuldades para atender às exigências previstas pelas normas e modelos de segurança de software, além do crescimento contínuo das exigências relacionadas à segurança em sistemas. Uma série de normas e modelos de segurança estão disponíveis na literatura a fim de guiarem o desenvolvimento de software seguro, porém a aplicação destas normas gera uma série de interferência no desenvolvimento de software, como aumento de custos, cronograma e demais recursos. Neste sentido, este trabalho visa propor níveis de segurança a serem aplicados em sistemas, bem como a avaliação de cada empresa e o modelo de segurança a ser utilizado em cada nível distinto. A proposta consiste na utilização de processos já consagrados na literatura, porém que podem ser utilizados de diversas formas possibilitando o entendimento de que a segurança precisa ser quantificada de forma distinta para cada projeto.*

1. Introdução

A crescente necessidade de produtos de software que suportam processos de negócios tem motivado consideravelmente pesquisadores no melhoramento de processos de desenvolvimento de software.

Da mesma maneira que cada projeto de sistema possui diferentes requisitos de software, também possuirá diferentes requisitos de segurança. Requisitos de segurança da informação são identificados por meio de uma avaliação sistemática dos riscos de segurança da informação. Gastos com os controles de segurança precisam ser balanceados de acordo com os danos causados aos negócios gerados pelas falhas potenciais na segurança da informação. Os resultados da avaliação de riscos ajudarão a direcionar e a determinar as ações gerenciais apropriadas e as prioridades para o gerenciamento dos riscos da segurança da informação [Compagna et al. 2008].

Além disso, considera-se importante a criação de diferentes níveis de segurança em projetos. Estes níveis são aplicados em projetos a partir de uma avaliação de diversos aspectos da empresa e do projeto em desenvolvimento.

Desta forma, este artigo tem como objetivo propor e avaliar níveis de segurança a serem aplicados em diferentes tipos de projetos. Estes níveis estão baseados em requisitos e medidas de segurança a serem tomadas pelo responsável pelo projeto.

O artigo está organizado da seguinte maneira: seção 2 apresenta níveis de segurança propostos para o desenvolvimento de software. A seção 3 demonstra a proposta de avaliação do nível de segurança a ser empregado no projeto. Na quarta seção é apresentada a ferramenta desenvolvida para suportar a proposta deste artigo e a seção 5 apresenta uma aplicação da proposta. A seção 6 apresenta trabalhos relacionados a este artigo. Por fim, na seção 7 são apresentadas as conclusões e trabalhos futuros, posteriormente é feita referência a bibliografia utilizada.

2. Critérios de segurança no desenvolvimento de software

A melhor maneira de desenvolver software seguro é incorporar a segurança desde o início do desenvolvimento de software. Além disso, o desenvolvedor deve conhecer as vulnerabilidades em diferentes artefatos do ciclo de vida do desenvolvimento do software para que estes possam ser removidos assim que possível. Caso contrário, a remoção as vulnerabilidades, numa fase posterior irá aumentar o custo significativamente [Khan Zulkernine 2008].

Os autores [MEAD MCGRAW 2005] apresentam uma iniciativa de segurança de software denominada *Build Security In* (BSI) a qual foi desenvolvida com a colaboração do *US National Institute of Standards and Technology* (NIST), o *International Organization for Standardization* (ISO), e o IEEE em padrões de atividades focados no desenvolvimento de subconjuntos de linguagens críticas e seguras e guias de estilos de garantia de softwares. O BSI busca alterar o caminho com o qual o software é desenvolvido, tendo assim menores vulnerabilidades de ataques através da construção de segurança desde inicio do desenvolvimento do projeto.

Outro processo de desenvolvimento de segurança em projetos é proposto por [Khan et al. 2009], intitulado SSDP (*Secure Software Development Process*). O SSDP está dividido em 4 fases: Engenharia de Requisitos; Design; Implementação; Garantia. Cada uma dessas fases está dividida em uma sequência de atividades que devem ser seguidas, além do relacionamento entre as fases.

3. Níveis de segurança propostos para processos de desenvolvimento de software seguro

Através dos vários autores citados anteriormente [MEAD MCGRAW 2005] [Khan et. al. 2009] [MEAD et. al. 2001], chega-se a um momento onde a revisão da bibliografia nos permite criar níveis de segurança.

Considera-se então, três níveis de critérios que podem ser utilizados: Nível Baixo; Nível Médio; Nível Alto.

Nível Baixo - O nível baixo é considerado para uso em sistemas que não demandem de maiores preocupações de segurança. Como exemplo, podemos citar um sistema de biblioteca que é acessado através de um único computador que possui uma única conexão a um servidor e não está ligado a Web.

Um sistema com este nível de segurança pode ser considerado como um sistema doméstico e geralmente não está interligado em várias estações de trabalhos, além de ter baixo nível de importância para a empresa.

Neste trabalho, será entendido como nível baixo de segurança a implantação de menos de 50% dos componentes catalogados no contexto do BSI.

Nível Médio - O nível médio pode-se considerar sistemas que demandem de segurança, como sistema de cadastro de alunos, notas, cursos e todas as informações de uma grande universidade. Este nível de segurança é considerado mediano, pois envolve uma série de requisitos de segurança a serem implantados, como normas, modelos e treinamento de segurança além bom senso aos usuários do sistema.

Neste trabalho, será entendido como nível médio de segurança a implantação de pelo menos 50% dos componentes catalogados no contexto do BSI. Pode também ser considerado um nível médio de segurança qualquer abordagem de segurança de alto nível que não seja criteriosamente seguida.

Nível Alto - Como sistemas de nível alto, pode-se citar como principal clientela sistemas bancários, transações web, transferência de dados particulares de cliente como no caso do imposto de renda entre outros sistemas deste gênero. Estes sistemas

necessitam do nível mais alto de segurança bem como implementação constante de novas técnicas desenvolvidas através de pesquisas recentes.

Como exemplo de modelos de desenvolvimento de software seguro pode-se citar Khan e Zulkernine (Khan, Zulkernine 2009) que desenvolveram um modelo de processo de desenvolvimento de software através da exibição de atividades e artefatos, intitulado de SSDP - *secure software development process*. No sentido de quantificar a segurança, propõe-se que todo o modelo SSDP deve ser implementado.

A proposta do nível de segurança a ser empregado no sistema deve dar-se através da avaliação das informações da empresa e do projeto. Identificar e propor aos *stakeholders* os critérios de segurança (baixo, médio ou alto) que através do entendimento do engenheiro de segurança seja necessário para garantir a segurança de tal projeto, explicando claramente ao cliente que se pode alterar o nível de segurança, porém isso irá alterar os custos e o cronograma de desenvolvimento do projeto.

4. Ferramenta desenvolvida para validação

A ferramenta desenvolvida permite selecionar as atividades que deverão ser realizadas para que o padrão de segurança seja atingido, após o usuário realizar a seleção das atividades necessárias, o sistema gera um WebSite que apresenta a descrição, o responsável, os artefatos de entrada e os artefatos de saída que deverão ser gerados, inclusive outras informações para que a equipe possa basear-se no momento de incluir a segurança no projeto.

Porém, como já descrito anteriormente existem projetos que demandam de diferentes níveis de segurança, o que resulta também em diferentes regras de associações entre requisitos e padrões de segurança. Para atender a esta especificação o usuário já deve cadastrar o nível de segurança que deseja para o projeto, com base nas definições acima, no momento do cadastro do projeto.

5. Utilização dos níveis de segurança

Para descrever como devem ser utilizados os níveis de segurança é adotado um sistema de informação para clínica médica a ser desenvolvido pela empresa especializada no desenvolvimento de sistemas relacionados à saúde.

Durante o cadastro dos dados será necessário o uso do bom senso e a ética do profissional, mas uma vez que os dados estão no sistema e a empresa tem acesso a todas as contas, valores, folha de pagamento e notas fiscais de seus clientes a responsabilidade passa a ser do sistema da empresa.

Através de uma análise da empresa e do projeto a serem desenvolvidos chegou-se a conclusão (que é apoiada pelos conceitos citados por [SBIS 2010] de que o nível de segurança demandado pelo projeto é alto.

Então, conforme a proposta deste trabalho, para que este nível seja atingido é necessária a implementação e verificação das atividades do SSDP. O detalhamento das atividades a serem implementadas no projeto, para que este tenha um nível alto de segurança serão descritas abaixo.

Tabela 1 - Atividades Relacionadas à Engenharia de Requisitos

R1 - O ambiente de funcionamento do sistema é constituído apenas de hospitais, postos de saúdes, e o banco de dados está alocado junto à empresa desenvolvedora do software, a qual também é responsável por total segurança de tais dados e informações.
R2 - Inspeções nas especificações de requisitos são realizadas durante todo o tempo de desenvolvimento de software e sempre que novas versões forem criadas.

R3 - Informações sobre ataques e tentativas de ataques anteriores são estudadas e coletadas, para que se tenha um conjunto de dados a serem analisados sempre que novos sistemas forem desenvolvidos.
R4 - Priorização de riscos e ameaças em trabalhos anteriores são realizadas, para utilização em trabalhos futuros.
R5- Especificação de requisitos de alto nível de segurança, tais como preservação da confidencialidade, integridade e disponibilidade são especificadas para atenuar as ameaças identificadas.
R6 - Verificação de mecanismos de segurança que são capazes de cumprir um requisito de segurança.
R7 - Requisitos de segurança de alto nível são priorizados através de uma análise custo-benefício. Mecanismo de segurança com maior prioridade serão implementados primeiramente, caso haja restrições orçamentárias.
R8 - Inspeções são realizadas a fim de identificar erros de segurança em software e requisitos de segurança de baixo nível.
R9 - Um limite de segurança aceitável é definido através de sua utilização. Caso um mecanismo fraco de segurança for selecionado, então sua fraqueza pode ser tratada como uma vulnerabilidade em relação ao cálculo do índice de segurança.
R10 - Se o índice de segurança calculado for inferior ao inicial, então os requisitos de segurança para remover erros identificados são especificados.
R11. Requisitos de segurança são priorizados com base em uma análise custo-benefício.

Byers et al. (2007) afirma que uma fonte de problemas de segurança é a não consideração de requisitos de segurança no completo desenvolvimento do sistema. Tais afirmações justificam o motivo da integração de requisitos relacionados à segurança desde o início do projeto. Através de 11 atividades o modelo proposto atinge um nível alto de segurança nesta fase.

Tabela 2 - Atividades relacionadas ao design

D1 - O design funcional do sistema é especificado de forma segura e ao mesmo tempo de fácil manuseio provendo ao usuário uma aplicação intuitiva as características peculiares do ambiente clínico, bem como a diminuição de navegação em telas do sistema para a obtenção de dados específicos, procurando assim diminuir a rejeição encontrada pelos clínicos na utilização de tais sistemas. Para que estes requisitos de design sejam atendidos será utilizada a linguagem de design UMLsec.
D2- O projetista do design inspeciona regularmente o projeto para identificação de possíveis erros de software.
D3 - O modelo de ameaças é reforçado e corrigido mais uma vez através da atividade anterior.
D4 - Análise das ameaças identificadas é realizada a fim de obter dados de sua proveniência e prever futuras ameaças.
D5 - Ameaças relacionadas a requisitos de segurança são removidas.
D6 - Decisões de design seguro para remoção de ameaças são priorizadas com base numa análise custo/benefício.
D7 - Erros de segurança de software e decisões de design seguro previamente especificados são identificados. Erros de segurança ou vulnerabilidades relatado no software existente similar podem ser usado como uma lista de verificação.
D8 - Um nível inicial de segurança aceitável deve ser definido utilizando, por exemplo, o índice de segurança.
D9 - Se o índice de segurança calculado é inferior ao inicial, então as decisões de design seguro para remover os erros devem ser especificados.
D10 - Decisões de *design* são priorizadas com base numa análise custo-benefício.

As decisões relacionadas ao design de projetos com alto nível de segurança devem ser baseadas nos requisitos de segurança definidos para o projeto inicialmente. As 10 atividades relacionadas ao *design* neste modelo são diretamente relacionadas e demonstram preocupações em relação a segurança modelada para este sistema.

Tabela 3 - Atividades relacionadas à implementação

I1. Utilização de linguagens de programação de nível superior em segurança, as quais oferecem perspectivas de código menos inseguro, como a linguagem Java que tem um alto grau de segurança devido as aplicações rodarem de forma "stand Box", nenhuma aplicação burla os requisitos de segurança especificados na linguagem Java, pois elas não tem acesso diretamente ao Sistema Operacional, desta forma elas são obrigadas a pedir que a Maquina Virtual Java aloque recursos do Sistema Operacional para a aplicação, assim essas requisições da aplicação passam pelas políticas de segurança especificadas na linguagem. No entanto, às vezes é necessária a utilização de linguagem de baixo nível para obter acesso direto ao nível de hardware.
I2. Padrões de códigos, guias de segurança, normas de codificação e orientações de segurança são seguidas no intuito de evitar erros de código-fonte. Padrões de desenvolvimento como (i) *Facede* que atua como uma camada mediadora entre a regra de negócio da aplicação e a interface gráfica, permitindo que caso regras de negócios mudem em algum determinado momento, isso não impacte na reestruturação de toda a aplicação novamente, (ii)

Data Acess Object (DAO) que permite a separação de código destinado a manipulação de dados do banco de dados para a aplicação, assim diminuindo a interferência da estrutura do banco diretamente na aplicação desenvolvida.
I3. Testes de unidade serão realizados com preocupações com a segurança em mente, Podendo ser utilizado o *firebug* para tais testes. Erros de segurança relatados em software similares serão usados como uma referência.

Em relação à implementação da segurança pode-se considerar que todos os requisitos de segurança e o design especificado serão implementados de maneira correta, através da utilização de linguagem de programação segura e da utilização de padrões, guias e normas de codificação.

Tabela 4 - Atividades Relacionadas à garantia

A1. Com base em erros anteriormente descobertos, bem como, nos requisitos de segurança do sistema serão realizadas inspeções no código e análise estática.
A2. Casos de testes são gerados com base nos requisitos funcionais e requisitos de segurança.
I3. Testes de integração, penetração e aceitação são realizados através dos requisitos de segurança relatados.

De acordo com as atividades relacionadas à garantia, o software em desenvolvimento pode ser considerado no final de sua implementação que possui um alto nível de segurança, bem como uma preocupação continua com os dados relacionados a este banco de dados.

Todos os níveis descritos acima devem ser aplicados de forma conjunta tendo um mesmo sincronismo, como, por exemplo, testes devem ser realizados sempre que novos requisitos forem levantados, testes devem ser aplicados sempre que esses novos requisitos forem implementados. Este sincronismo entre as partes deve existir para que no final do processo o produto resultante esteja em conformidade com as regras de segurança estabelecidas.

Analisando os resultados verifica-se que ao desenvolver o sistema de Informática em Saúde é necessário considerar as fases de engenharia de requisitos, design, implementação e garantia, além de cumprir todas as atividades especificadas em cada fase de acordo com explicação de cada uma dessas atividades relacionadas ao sistema de Informática na Saúde. Algumas das atividades explicadas podem ser consideradas genéricas para qualquer sistema de alto nível segurança e outras atividades são especificamente explicadas para a área de Informática na Saúde. A meta principal do desenvolvimento visa realizar a implementação de um sistema que tenha requisitos de segurança especificados no inicio do projeto e que cumpra todas as necessidades de segurança do sistema em desenvolvimento.

6. Conclusão

Considera-se o tema do trabalho importante, pois a segurança na qual é baseado o processo de software e consequentemente o desenvolvimento de sistemas tem se mostrado cada vez mais importante para a organização, para os desenvolvedores e para seus usuários.

A elaboração de níveis de segurança através de modelos encontrados na literatura, que já tiveram uma aceitação da sociedade científica, além de terem sido criados por renomados autores auxilia na compreensão e na validação da eficácia de tais modelos.

Para a decisão de quais modelos devem ser utilizados em cada empresa e cada projeto, foi criado um esquema que permite ao gerente de projetos em conjunto com o engenheiro de segurança uma avaliação de qual nível deve ser utilizada, além de dados concretos que possam demonstrar a usuários de produto final porque um determinado nível de segurança foi escolhido.

A abordagem apresentada neste artigo é explicada através de um estudo de cada atividade aplicado em uma empresa desenvolvedora de softwares relacionados à informática médica, o qual possibilita o entendimento e a forma com que as atividades relacionadas à segurança podem ser implementadas.

Trabalhos futuros representam o projeto de uma dissertação de mestrado que está em desenvolvimento e objetivam o desenvolvimento de um processo de software para segurança baseado em requisitos gerais de segurança bem como da atribuição de níveis de segurança baseados nesses requisitos de segurança.

7. Bibliografia

CARDOSO et al. (2009) "Um Modelo de Controle Formal para o gerenciamento de riscos de processo em fábricas de software". Publicado em: Anais do IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais

CERT (2010), Coordination Center Statics. Disponível em: www.cert.org/stats/cert_stats.html.

MEAD, NANCY R., LINGER R., MCHUGH J., LIPSON H., Managing Software Development for Survivable Systems, Annals Software Eng., vol. 11, no. 1, 2001, pp. 45-78. BYERS, DAVID e SHAHMEHRI, NAHID (2007) "Design of a Process for Software Security" Second International Conference on Availability, Reliability and Security (ARES'07).

CAMPOS A., (2007) "Sistema de segurança da informação - Controlando os riscos" Santa Catarina: Visual Books.

COMPAGNA L., KHOURY P., KRAUSOVÁ A., MASSACCI F., ZANNONE N. (2008) "How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns" Publicado em: Springer Science+Business Media

KHAN, Muhammad U. A., Zulkernine, Mohammad. "Activity and Artifact Views of a Secure Software Development Process". International Conference on Computational Science and Engineering. , 2009, pp. 339- 404.

KHAN Muhammad U. A., Zulkernine, Mohammad. "Quantifying Security in Secure Software Development Phases" Annual IEEE International Computer Software and Applications Conference, 2008, pp. 905-960.

MEAD NANCY R., MCGRAW GARY, A Portal for Software Security, IEEE SECURITY & PRIVACY, 2005, pp. 75-79.

NBR ISO/IEC 27001; Tecnologia da informação - Sistemas de gestão de segurança da informação . Rio de Janeiro - RJ (2006)

NUNES, F. J. B. BELCHIOR, A. D., ALBUQUERQUE A. B. (2009) "A knowledge Management Aproach to Support a Secure Software Development" Universidade de Fortaleza - Fortaleza.

SBIS (2010) - Sociedade Brasileira de Informática na Saúde. Disponível em: http://www.sbis.org.br/

SOMMERVILLE I. (2007) "Engenharia de software" São Paulo: Pearson Addison - Wesley.