SEGURANÇA DA INFORMAÇÃO NO DESENVOLVIMENTO DE SOFTWARE: UMA PROPOSTA DE PROCEDIMENTOS PARA OS SISTEMAS CORPORATIVOS DO EXÉRCITO BRASILEIRO

Francisco José Prates Alegretti[1], Éldman de Oliveira Nunes[2]

Resumo. O presente artigo tem por objetivo apresentar uma proposta de procedimentos de segurança da informação em sistemas corporativos do Exército Brasileiro. A proposta de procedimentos inclui controles em todas as fases do ciclo de vida do desenvolvimento de software (CVDS), sendo composta por quatro itens principais: treinamento do pessoal, revisão do código-fonte, verificação automatizada de vulnerabilidades no código e, por fim, casos de teste de segurança e vulnerabilidade. Estão inclusas na proposta deste trabalho as diretrizes para uma instrução de desenvolvimento de aplicações seguras, uma listagem de verificações e boas práticas a serem realizadas na revisão do código-fonte, sugestões de ferramentas automatizadas disponíveis atualmente e, também, recomendações sobre a elaboração dos testes de segurança e vulnerabilidade. Serão abordados conceitos fundamentais da área: confidencialidade, integridade e disponibilidade, assim como os de autenticação, autorização e auditoria. A validação da proposta é feita através da aplicação dos procedimentos sugeridos, com o acompanhamento e medição dos resultados ao longo do ciclo de vida do desenvolvimento do software. O resultado do trabalho consiste não apenas dos procedimentos de segurança, mas, também, da sua forma de aplicação e validação.

Palavras-chave: Segurança da Informação. CIA+A³. Segurança no CVDS.

Abstract. This paper has the objective to present a proposal of information security procedures for the corporate systems of the Brazilian Army. The procedures proposed include security controls in all levels of the software development life cycle (SDLC), and is composed of four main items: training, source-code review, automatic code vulnerability check, security and vulnerability test cases. This work includes the directives to be used as a basis for the proposed instruction of secure application development. A check-list of best practices to be adopted during the source-code peer review is also included. Suggestions of automated tools for code analysis available to date are presented, as well as recommendations about the elaboration of the security and vulnerability tests. The fundamental concepts of confidentiality, integrity and availability, as well as those of authentication, authorization and

[1] Mestre em Ciência da Computação. Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brasil. alegretti@gmail.com.
[2] Doutor em Computação. Universidade Federal Fluminense (UFF), Niterói, Brasil. eldman.nunes@gmail.com.

auditing are employed. Validation of the proposal is done by applying the recommended procedures, following up and measuring the results through the software development life cycle. The result of this work consists not only of the security procedures, but also of the manner in which they are applied and validated.

Keywords: Information Security. CIA+A³. Security in SDLC.

1 Introdução

Segurança da Informação é a área do conhecimento que protege os ativos de informação contra acessos não autorizados, alterações indevidas ou, ainda, a sua indisponibilidade (SÊMOLA, 2003). Trata-se, portanto, de uma área que envolve conhecimentos de criptografia, integridade e disponibilidade.

O presente trabalho enquadra-se no tema de Segurança da Informação. Especificamente, será dado enfoque aos sistemas corporativos do Exército Brasileiro e aos desafios de segurança que estes enfrentam, atualmente, com relação aos conceitos citados. Em outras palavras, este trabalho versa sobre a Segurança da Informação nos Sistemas Corporativos do Exército Brasileiro.

2 Segurança da Informação

Até recentemente, os ataques a sistemas informatizados concentravam-se nas redes de computadores. Os sistemas corporativos que utilizam a rede não eram alvos comuns dos *hackers*. Atualmente, esse cenário mudou. Os anos de esforços em desenvolvimento de mecanismos de proteção para as redes, como *Firewalls*, Anti-Vírus, Sistemas de Detecção de Intrusão (IDS - *Intrusion Detection System*), entre outros, resultaram em redes mais seguras e com poucas vulnerabilidades a investidas externas. Assim, com o aumento de segurança das redes, tornou-se mais fácil atacar o sistema computacional em si, ou seja, os aplicativos que compõem o sistema informatizado que rodam na rede. De acordo com a ISC² (2009), a maior parte das falhas de segurança estão relacionadas aos aplicativos, "*Some 80% of all security breaches are application related. Every person involved in the Software Lifecycle should consider security as an essential element*". Portanto, verifica-se que o foco dos ataques, atualmente,

mudou das redes para os *softwares;* o ponto crítico de segurança da informação hoje são os sistemas corporativos, e não mais as redes de computadores.

O Exército Brasileiro, cuja missão constitucional primordial é a defesa da Pátria (BRASIL, 1988), possui diversos sistemas corporativos que o auxiliam a cumprir essa missão. E, como qualquer corporação de grande porte, o EB também depende dos seus sistemas de informação para executar as suas tarefas diárias. Assim, uma falha de segurança em um sistema corporativo do Exército pode prejudicar o cumprimento da sua missão constitucional de defender a Nação. "A visão corporativa da segurança da informação deve ser comparada a uma corrente, em que o elo mais fraco determina seu grau de resistência e proteção. A invasão ocorre onde a segurança falha!" (SÊMOLA, 2003, p. 40). Portanto, não adianta ter uma rede corporativa forte e bem protegida se os sistemas corporativos que rodam nessa rede são fracos em termos de segurança e vulneráveis a ataques; a invasão ocorrerá no elo mais fraco do conjunto rede e aplicativos que, na atualidade, são justamente os sistemas corporativos.

Dessa forma, tendo em vista que a Segurança da Informação é indispensável para o Exército Brasileiro e que, na atual conjuntura da Tecnologia da Informação, os ataques virtuais concentram-se nos programas e não mais na rede em si (PAUL, 2009), verifica-se a importância de estabelecer-se um conjunto de procedimentos de Segurança da Informação para os sistemas corporativos do EB. Este é, justamente, o problema abordado pelo presente trabalho.

3 Procedimentos de Segurança

Segundo Sêmola (2003), segurança consiste em implementar controles que reduzam o risco a níveis adequados, viáveis e administráveis. Os procedimentos de segurança propostos neste artigo constituem, nas palavras do autor citado, um controle; este controle em particular, objetiva a redução do risco de falhas de segurança durante o desenvolvimento de *software*. Ou seja, esses procedimentos têm por finalidade evitar que os sistemas corporativos do Exército Brasileiro sejam colocados em produção com *bugs* de segurança presentes no seu código. Tudo da maneira mais eficiente e, portanto, menos custosa possível.

3.1 Visão Geral

Tendo em vista que o custo da detecção de falhas de segurança aumenta ao longo do tempo (FARIS, 2006), deve-se planejar e trabalhar com a segurança da informação desde a primeira fase do ciclo de vida do desenvolvimento de *software* (ARNOLD, et. al, 2007), a fim de evitar-se riscos de segurança e custos adicionais de correções de falhas no sistema. Quanto mais tarde no ciclo de vida do sistema uma falha for detectada, mais cara será a sua correção. Portanto, a segurança deve estar intimamente relacionada e envolvida com o ciclo de vida do desenvolvimento de software. Ela deve ser implementada e testada já na fase de desenvolvimento e não somente quando o sistema está em produção e os riscos, assim como os custos, são exponencialmente maiores.

A proposta de procedimentos de segurança deste trabalho abrange todas as fases do ciclo de vida e é composta de quatro itens principais: treinamento, revisão do código-fonte, verificação automatizada de vulnerabilidades no código e casos de teste de segurança e vulnerabilidade. Todos esses controles, assim como a sua disposição ao longo do ciclo de vida do desenvolvimento do *software*, podem ser visualizados na figura 1.

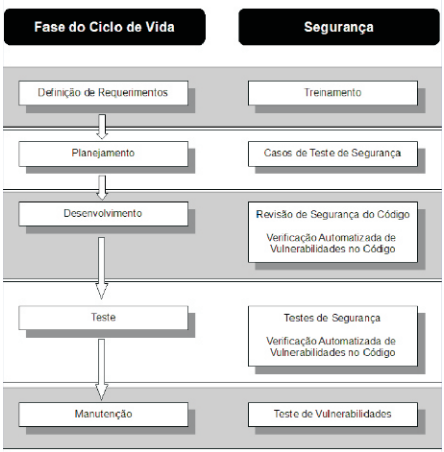

Figura 1: Segurança no Ciclo de Vida
Fonte: O autor

Analisando a figura, verifica-se que o treinamento deve ser realizado logo no início do projeto, ou seja, durante a fase de definição de requerimentos. Os casos de teste de segurança, por sua vez, devem ser planejados antes mesmo da codificação da primeira linha do *software*, na fase de planejamento. Durante o desenvolvimento são implementados dois controles de segurança: a revisão de segurança do código e a verificação automatizada de vulnerabilidades no código; este último controle é executado novamente durante a fase de teste, desta vez, pela equipe de testadores. A fase de teste também inclui a execução dos testes de segurança, que haviam

sido elaborados durante a fase de planejamento. Finalmente, durante a fase de manutenção do *software*, são executados, periodicamente, testes de vulnerabilidades contra o sistema corporativo.

3.2 Treinamento

O treinamento é o primeiro e mais importante passo para a obtenção de aplicações mais seguras: qualificar o pessoal (desenvolvedores) com treinamento sobre os conceitos básicos de segurança da informação. "O nível de segurança de uma corrente é equivalente à resistência oferecida pelo elo mais fraco. O *peopleware* representa justamente esse elo; por isso deve ser alvo de um programa contínuo e dinâmico, capaz de manter os recursos humanos motivados a contribuir, conscientes de suas responsabilidades e preparados para agir diante de antigas e novas situações de risco" (SÊMOLA, 2003, p. 136). Com *peopleware*, o autor citado refere-se aos recursos humanos da empresa. No caso de interesse particular deste artigo, o *peopleware* é constituído pelos desenvolvedores de software do Exército Brasileiro. Ou seja, não adianta ter ferramentas de desenvolvimento modernas e boas políticas de segurança - os elos fortes da corrente - se o pessoal não está treinado para utilizar essas ferramentas de forma adequada, nem qualificado para seguir as políticas adotadas. Por isso, o treinamento do pessoal é indispensável para forjar a resistência do elo constituído pelo *peopleware*.

A importância de treinar o pessoal em segurança da informação também é salientada no parágrafo único do artigo 3º do Decreto número 4.553, de 27 de dezembro de 2002, que diz: "Toda autoridade responsável pelo trato de dados ou informações sigilosos providenciará para que o pessoal sob suas ordens conheça integralmente as medidas de segurança estabelecidas, zelando pelo seu fiel cumprimento". Não menos importante, o caput do referido artigo determina que "A produção, manuseio, consulta, transmissão, manutenção e guarda de dados ou informações sigilosos observarão medidas especiais de segurança". Portanto, um treinamento que qualifique o pessoal responsável pelo trato de informações sigilosas nas medidas de segurança estabelecidas é um requerimento legal.

Os conceitos de segurança da informação abordados neste artigo podem servir como referência para a elaboração de um treinamento

interno específico de segurança, para o público alvo de programadores do Exército Brasileiro. Os assuntos mais propícios a serem abordados numa instrução sobre segurança da informação são, nominalmente: Confidencialidade, Integridade, Segurança, Autenticação, Autorização, Auditoria, Criptografia, *Hashing* e a classificação das informações conforme o seu grau de sigilo, segundo o Governo Federal.

Propõe-se que o treinamento seja intitulado de "Instrução de Desenvolvimento de Aplicações Seguras", e que tenha como público alvo todos os programadores de *software* do EB. O treinamento deve ser presencial, composto de uma parte teórica dos conceitos fundamentais de segurança anteriormente mencionados; mais um laboratório prático, com a aplicação prática dos conceitos teóricos. A duração sugerida para o treinamento é de duas semanas.

3.3 Revisão do Código-Fonte

Com o pessoal qualificado e, principalmente, conscientizado, pelo treinamento de desenvolvimento de aplicações seguras, essa etapa das políticas de segurança constitui-se em adotar as técnicas aprendidas durante o treinamento no dia a dia do trabalho de desenvolvimento de *software*. Para facilitar tal tarefa, convém-se uma série de itens a serem verificados pelo programador durante o desenvolvimento do software. De maneira geral, esses itens incluem os controles de confidencialidade durante o processamento, trânsito e armazenamento das informações. Os itens de controle de integridade, por sua vez, constituem-se em validar todas as entradas de dados, falhar de forma segura e fazer a correta manipulação de erros. Já os controles de disponibilidade envolvem a realização regular de cópias de segurança, seu correto armazenamento e o respectivo procedimento escrito para a restauração de dados em caso de falha. Por fim, a lista deve abordar os controles de autenticação, autorização e auditoria. Sugere-se que esses itens sejam acrescentados ao teste de *peer-review* adotado nas seções de informática das OM.

Adicionalmente, salienta-se que o sub-sistema de *logs* é indispensável, pois é de extrema importância para detectar a ocorrência de uma falha de segurança, possibilitar a análise e compreensão do ataque e, por fim,

o saneamento da vulnerabilidade. Também deve-se evitar o uso de rotinas conhecidamente inseguras, especialmente as de manipulação de strings nas linguagens C e C++. Com relação ao banco de dados, deve-se sempre utilizar *Stored Procedures* (SP) e *views*. Da mesma forma, na montagem de *strings* de comandos de SQL, deve-se tomar as devidas precauções de validação de entradas para não sofrer *SQL-Injection* ao formar comandos concatenando *input* do usuário. Ressalta-se, mais uma vez, que dados sensíveis devem ser sempre criptografados durante o trânsito e armazenagem. É necessário, ainda, verificar a adequação do algoritmo de criptografia utilizado para garantir o nível de proteção adequado à classificação de sigilo da informação em questão. Por fim, implementar e executar periodicamente mecanismos de *backup*, ou seja, realizar regularmente cópias de segurança das informações armazenadas pelo sistema.

3.4 Verificação Automatizada

Essa verificação constitui-se em utilizar uma ferramenta automatizada de verificação de falhas de segurança no código fonte dos sistemas corporativos. A verificação manual de uma série de itens contra todas as linhas de código de um programa, se realizada por um operador humano, tornar-se-ia uma tarefa repetitiva, tediosa, cansativa e, assim, propensa a erros; além disso, consumiria muito tempo para ser concluída com sucesso. Tendo isso em vista, verifica-se que o uso de uma ferramenta automatizada para a realização de tal tarefa é altamente adequada e aconselhável.

Atualmente existem diversas opções desse tipo de ferramenta disponíveis, tanto comerciais como de *software* livre. Das opções comerciais pagas, a ferramenta Fortify (FORTIFY, 2009) constitui, até a data de elaboração do presente texto, a melhor escolha. Essa ferramenta é composta por três verificadores que detectam as vulnerabilidades do programa. O Fortify inclui um módulo para ajudar os desenvolvedores a sanar as vulnerabilidades identificadas. Adicionalmente, a ferramenta também possui um console de relatórios e gerenciamento.

Entre as opções de *software* livre, a ferramenta YASCA é uma escolha popular entre os programadores (YASCA, 2009). O nome da ferramenta é a sigla, em inglês, de *Yet Another Source Code*

Analyzer, ou seja, "mais um analisador de código fonte". YASCA é um programa de código aberto que procura por vulnerabili-dades de segurança, verifica a qualidade do código, o seu desempenho, e a conformidade do mesmo com as melhores práticas estabelecidas de desenvolvimento de código. Ao contrário da opção comercial paga, a ferramenta não possui uma interface muito elaborada; seu funcionamento é via linha-de-comando e a apresentação dos seus resultados é através de um arquivo HTML.

Essa etapa de verificação do código fonte tem o potencial de causar uma melhora significativa na qualidade do *software* produzido em termos de segurança. Mas requer a colaboração dos programadores para ser colocada em prática, pois são os desenvolvedores de cada sistema que devem executar a verificação no código produzido e realizar a correção das falhas encontradas. Adicionalmente, tanto as opções comerciais como as de código livre apresentam um certo índice de falsos-positivos em seus resultados. Um falso-positivo é um trecho de código que a ferramenta julga ser uma falha de segurança mas que, na realidade, não é um *bug* de segurança. Portanto, é necessário uma revisão manual dos resultados apresentados pela ferramenta para a verificação dos falsos-positivos.

Nota-se que existe uma dificuldade em potencial na adoção dessa etapa dos procedimentos de segurança. A prática demonstra que, apesar de eficientes, esse tipo de ferramenta encontra certa resistência de ser utilizada pelos programadores, pois: acarreta mais trabalho para os mesmos com a correção dos *bugs* encontrados; e, principalmente, o temor (na maioria das vezes, não fundamentado) dos desenvolvedores de que a detecção e exposição de falhas de segurança no seu código coloque em dúvida a qualidade do seu trabalho. A melhor forma de enfrentar essa dificuldade é com treinamento, pois a qualificação vai conscientizar o programador da importância do cuidado com segurança da informação, bem como demonstrar a utilidade de ferramenta que, na realidade, vai auxiliar o desenvolvedor a melhorar a qualidade do seu código. Assim, ao invés de temer a ferramenta, o programador deve utilizá-la como atestado da qualidade do código que ele produz.

3.5 Teste de Segurança e Vulnerabilidade

Existem dois tipos diferentes de testes a serem executados: os testes de segurança e os testes de vulnerabilidades. Estes testes têm por objetivo verificar se nenhuma

falha de segurança passou pelos controles anteriores de segurança da informação (TIPTON, KRAUSE, 2001). A eficiência e confiabilidade desses testes baseia-se no princípio de que quem programa um sistema não testa o mesmo; ou seja, os testes devem ser conduzidos por pessoal específico - os testadores - e não pelos programadores. Salienta-se, portanto, a importância de que pessoas diferentes testem o sistema que foi implementado.

Os Testes de Segurança são elaborados na fase de planejamento do projeto, ou seja, antes mesmo do código-fonte ser implementado. Diferem do teste normal pois não estão focados nas funcionalidades de negócio do programa mas, sim, em potenciais problemas de segurança. A sua execução será realizada concomitantemente com os teste funcionais, somente após a finalização da codificação e o início da fase de teste propriamente dita. Esses testes são fundamentados na primeira parte do treinamento de desenvolvimento de aplicações seguras, ou seja, nos conceitos fundamentais de segurança. São testes de fundamentação teórica, e verificam os controles de confidencialidade, integridade e disponibilidade que devem estar presentes no código do programa, assim como os controles de autenticação, autenticidade e auditoria. Os testes de segurança são elaborados a partir de casos de teste baseados nos requerimentos do projeto em particular.

O teste de vulnerabilidade, por sua vez, é executado periodicamente após a implantação do sistema; ou seja, esse teste é executado em ambiente de produção. Seu propósito é justamente testar o sistema contra vulnerabilidades quando o mesmo está inserido dentro de seu ambiente de operação, levando-se em consideração o sistema operacional, a rede e o próprio sistema em si. Devido a sua natureza periódica, esse tipo de teste tem por função capturar defeitos injetados no sistema a partir de atualizações de código do programa, ou mesmo a partir de mudanças realizadas no ambiente em que o sistema executa, como atualizações do sistema operacional ou mudanças no banco de dados. Os testes de vulnerabilidades são elaborados com base na segunda parte da instrução de desenvolvimento de aplicações seguras, isto é, a parte prática de laboratório. Basicamente, os testes de vulnerabilidades são a execução, na prática, dos exercícios

realizados em laboratório durante o treinamento. Salienta-se, entretanto, a importância do testador em manter-se sempre atualizado com relação a novas técnicas de ataque e verificação de vulnerabilidade, assim como a importância do treinamento ser constantemente atualizado, para que passe a cobrir as novas técnicas à medida que as mesmas forem surgindo.

Um exemplo de teste de segurança, para um sistema que trabalhe com informações classificadas como secretas, seria verificar se os controles confidencialidade estão implementados, ou seja, testar se o programa está criptografando adequadamente as informações secretas que armazena. Nesse exemplo em particular, seria necessário verificar se o algoritmo de criptografia utilizado é considerado seguro, ou seja, se ele não foi quebrado ou se não possui vulnerabilidades conhecidas em sua implementação. Já um teste de vulnerabilidade seria, por exemplo, verificar se o software está vulnerável a um ataque de *SQL-Injection*, seja esse decorrente da programação original do sistema, ou advinda de alguma atualização do código feita posteriormente, ou mesmo decorrente de mudanças nas *stored procedures* no banco de dados do sistema. Por isso, é importante realizar periodicamente o teste de vulnerabilidades: para que sejam detectadas possíveis falhas introduzidas em atualizações funcionais do sistema.

3.6 Considerações Adicionais

Salienta-se a necessidade de conscientização do comando com relação à importância de uma política, ou regulamento, para a devida adoção dos cuidados de segurança indispensáveis durante o ciclo de vida do desenvolvimento de software. “Somente com apoio executivo as ações de segurança ganharão autonomia e abrangência capazes de incidir corporativamente sobre os frutos de segurança” (SÊMOLA, 2003, p. 40).

Em artigo publicado na revista de produção científica da ESAEX, Santos, Silva e Nalin (2006) propõe a criação de um grupo de segurança da informação em cada organização militar. Tal medida é válida e, de fato, possui o potencial para melhorar a segurança da informação no âmbito de cada OM. Mas também existe a necessidade de organizar e padronizar os procedimentos de segurança da informação no Exército como um todo. Portanto,

um órgão geral ou grupo de trabalho para a coordenação dessas atividades específicas no EB é uma boa prática de segurança.

4 Validação

A validação das propostas de segurança da informação apresentadas neste trabalho exige a aplicação dos procedimentos, a mensuração da eficácia dos mesmos e a análise dos resultados obtidos. Assim, a validação é dividida em duas grandes etapas: a primeira é constituída pela aplicação da instrução de desenvolvimento de aplicações seguras e a mensuração do aproveitamento do pessoal com relação ao treinamento; a segunda etapa, por sua vez, é composta pela aplicação na prática dos procedimentos ministrados na instrução, bem como pela mensuração da eficácia dos procedimentos em comparação com os métodos anteriores. A listagem a seguir, detalha cada uma das atividades da primeira etapa da validação, com a indicação do tempo necessário para cada atividade.

· Questionário de Avaliação de Conhecimentos (1 hora);
· Instrução de Desenvolvimento de Aplicações Seguras (2 semanas);
· Questionário de Avaliação de Conhecimentos (2ª aplicação, 1 hora).

As atividades relativas a segunda etapa da validação são listadas abaixo. O tempo necessário para todas essas atividades é 1 ciclo de vida de desenvolvimento de *software*.

· Medir aplicações previamente desenvolvidas;
· Desenvolvimento de Aplicações Seguras;
· Revisão do Código-Fonte;
· Análise Automatizada do Código-Fonte;
· Testes de Segurança e Vulnerabilidade;
· Mensurar aplicação desenvolvida com as propostas deste trabalho.

4.1 Primeira Etapa

A primeira etapa da validação constitui-se, basicamente, na realização da instrução de desenvolvimento de aplicações seguras, com a aplicação de questionários de avaliação de conhecimentos para verificar-se a eficácia da instrução ministrada. A validação deve ocorrer conforme descrito a seguir.

Inicialmente, selecionar um grupo de desenvolvedores de software do Exército Brasileiro.

Aplicar, antes da realização do treinamento, o questionário de avaliação de conhecimentos em Segurança da Informação (ALEGRETTI, NUNES, 2009). Em seguida, realizar o treinamento de Desenvolvimento de Aplicações Seguras. Ao término do curso, aplicar novamente o mesmo questionário. A partir da comparação dos resultados obtidos no questionário antes e depois da ministração da instrução, será possível medir a melhora do nível de conhecimento técnico do pessoal com relação ao assunto. Com base nas tendências apresentadas no questionário, também será possível efetuar ajustes e melhorias específicas na instrução.

4.2 Segunda Etapa

A segunda etapa inicia-se com a mensuração das aplicações previamente desenvolvidas pelo grupo de desenvolvedores de *software* selecionados durante a primeira etapa. Para cada uma das aplicações a serem analisadas, será necessário investir o tempo correspondente a uma fase de teste do ciclo de vida da aplicação em particular. A mensuração em si consiste em executar a ferramenta automatizada de verificação de vulnerabilidades no código e analisar os resultados apresentados, filtrando os falsos-positivos.

A seguir, é desenvolvida uma nova aplicação. Desta vez, com o pessoal devidamente treinado e seguindo as propostas apresentadas neste artigo; ou seja, os programadores realizarão a revisão do código-fonte e, também, executarão a ferramenta automatizada para detecção de vulnerabilidades na aplicação. Da mesma forma, também serão devidamente planejados e executados os testes de segurança e de vulnerabilidade do sistema. Todos esses procedimentos consumirão um ciclo de vida de desenvolvimento de *software* completo para serem implementados. Ao término do ciclo, a mesma mensuração executada no início da etapa, para aplicações anteriormente desenvolvidas, deverá ser realizada para a nova aplicação desenvolvida com as propostas de segurança deste artigo. Com base nesses dois índices, será possível comparar e medir a eficiência dos controles de segurança. Validar-se-ia, assim, a proposta do presente trabalho.

5 Conclusão

Atualmente, a maioria dos

ataques a sistemas informatizados ocorrem através de vulnerabilidades nas aplicações corporativas e não mais por falhas de segurança nas redes de computadores. O Exército Brasileiro, que possui diversos sistemas corporativos que o auxiliam a cumprir sua missão constitucional, precisa adotar medidas de segurança da informação para sua proteção. Assim, torna-se necessário direcionar o foco das medidas de segurança para o desenvolvimento de *software*. Adicionalmente, sabe-se que corrigir defeitos de *software* é muito mais caro depois que o sistema já está em produção do que quando ele está em desenvolvimento. Assim, é mais fácil e menos custoso corrigir as falhas de segurança durante a fase de codificação do aplicativo. Por isso, os procedimentos de segurança propostos neste artigo foram voltados para o desenvolvimento de aplicações seguras, incluindo controles em todas as fases do ciclo de vida do desenvolvimento do *software*.

Na proposta apresentada neste trabalho, os procedimentos de segurança da informação incluem o treinamento de pessoal, através de instrução especialmente elaborada para o Exército Brasileiro; a revisão do código-fonte, com o auxílio de uma série de controles de segurança listados e explanados neste trabalho; a verificação automatizada de vulnerabilidades no código, por meio de ferramentas específicas sugeridas neste texto; e, por fim, casos de teste de segurança e vulnerabilidade, com a elaboração e aplicação dos testes nas fases apropriadas do projeto.

A validação da proposta de procedimentos de segurança da informação para os sistemas corporativos do Exército Brasileiro exige o treinamento de um grupo de programadores por meio da instrução de desenvolvimento de aplicações seguras e a medição, através de questionário, da eficácia do treinamento; também é necessário auferir o nível de segurança das aplicações previamente desenvolvidas pelo grupo de desenvolvedores, a fim de medir a melhora obtida com a adoção dos procedimentos propostos. Dessa forma, comprovar-se-á que a execução das propostas de segurança pode garantir propriedades desejáveis, como confidencialidade e integridade dos dados, eliminar vulnerabilidades do sistema corporativo e reduzir riscos de segurança do *software*.

A contribuição deste trabalho, para o Exército Brasileiro, está no

fortalecimento de seus sistemas corporativos, com a adoção das medidas propostas. Também contribui para a qualificação do seu pessoal, com a implantação da instrução sugerida. Com isso, abre-se a possibilidade de se criar uma cultura de segurança da informação no desenvolvimento de sistemas para o EB. No âmbito geral da segurança da informação, este trabalho apresenta a sua parcela de contribuição em uma área que só recentemente vem sendo trabalhada pela comunidade especializada, isto é, a exploração e defesa de vulnerabilidades nos sistemas corporativos e não apenas das redes de computadores.

Finalmente, devido ao tempo exigido para ministrar a instrução de treinamento proposta, acrescida do período exigido para a realização do desenvolvimento de *software*, não foi possível executar a validação planejada dentro da janela de tempo disponível; para tanto, seria necessário empregar um ciclo de vida de desenvolvimento de *software* completo. Tendo em vista essa limitação, sugere-se como trabalho futuro a aplicação da instrução de Desenvolvimento de Aplicações Seguras, com base no material apresentado neste texto. Com a realização do treinamento, poder-se-á desenvolver outro trabalho futuro: a validação das propostas de segurança, conforme descrito neste trabalho.

Referências

ALEGRETTI, F. J. P., NUNES, E. O. **Desenvolvimento de Aplicações Seguras: uma proposta de procedimentos de segurança da informação para os sistemas corporativos do Exército Brasileiro.** Trabalho de Conclusão de Curso. Salvador: EsAEx, 2009.

ARNOLD, T., HOPTON, D.,LEONARD, A., FROST, M. **Professional Software Testing With Visual Studio 2005 Team System: Tools For Software Developers And Test Engineers.** Wiley-India, 2007.

BRASIL. Constituição (1988).

BRASIL. Decreto nº 4.553, de 27 de dezembro de 2002.

FARIS, T. H. **Safe and sound software: creating an efficient and effective quality system for software medical device organizations.** USA: American Society for Quality, 2006.

FORTIFY. Application Security - Fortify Software. Disponível em:

<http://www.fortify.com/>. Acesso em: 29 jul 2009.

ISC2. Certification CISSP. Disponível em: <http://www.isc2.org/cissp/default.aspx>. Acesso em: 16 maio 2009.

PAUL, M. The need for secure software. **Software Community (ISC)² Whitepapers.** Disponível em: <http://www.isc2.org/uploadedFiles/(ISC)2_Public_Content/Certification_Programs/CSSLP/CSSLP_White Paper.pdf>. Acesso em: 12 set 2009.

SANTOS, G. P., SILVA, W. C., NALIN, M. Segurança da Informação: da Constituição e Atuação do Conselho Gestor de Segurança na Organização Militar. **Revista Científica da Escola de Administração do Exército,** Salvador, v. 1, n. 2, p. 6-19, 1º semestre de 2006.

SÊMOLA, M. **Gestão da Segurança da Informação.** Editora Campus, 2003.

TIPTON, H. F., KRAUSE, M. **Information Security Management Handbook.** 4th ed. CRC Press, 2001. 626 p.

YASCA. Yet Another Source Code Analyzer. Disponível: <http://www.yasca.org/>. Acesso em: 29 jul 2009.