UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
TESTES, VERIFICAÇÃO E VALIDAÇÃO DE SISTEMAS
TRABALHO FINAL DA DISCIPLINA

TESTES DE SOFTWARE E A SEGURANÇA DA INFORMAÇÃO: UM ESTUDO ABRANGENTE

Nome: Luis Gonzaga de Paulo

Profª: Maria Cláudia F. P. Emer

CURITIBA

2013

SUMÁRIO

RESUMO

A popularização do uso de computação móvel, principalmente *smart phones* e *tablets* que fazem uso do sistema operacional Android em todo o mundo - e particularmente no Brasil - o aumento do uso dos serviços de comunicação de dados decorrente da redução de custo – quer seja dos equipamentos quer seja dos serviços, e o aumento da velocidade de acesso disponibilizada pelas prestadoras de serviços de comunicação com as redes 3G e 4G alimenta um valioso mercado, estimulando a explosão do crescimento do número de aplicações disponíveis. Entretanto este crescimento não veio amparado do devido esclarecimento, aos usuários, dos riscos aos quais estão submetidos. De forma análoga, desenvolvedores – muitas vezes trabalhando de forma voluntariosa e singular – não são orientados a adotar procedimentos mínimos de segurança no desenvolvimento de software, especialmente os definidos em normas técnicas tais como a ISO 15.408 (RIBEIRO 2002). Empresas e instituições, por sua vez, carecem de parâmetros e ferramental para a avaliação da segurança da informação em tais dispositivos, optando, na maioria das vezes, por não disponibilizar seus serviços para tal plataforma, evitando ou restringindo o uso daqueles dispositivos por seu corpo funcional. Através da pesquisa, da troca de informação com pesquisadores, fabricantes e fornecedores de *software* para o Android, da experimentação e da formulação de procedimentos de avaliação, do desenvolvimento de ferramental de suporte a estes e da divulgação e disponibilização dos mesmos, ambiciona-se, ao término desta pesquisa, colaborar para que a expansão do uso do Android e das suas aplicações se dê de forma mais confiável e segura, elevando a percepção da segurança pelos usuários, desenvolvedores, empresas e instituições que dele fazem uso.

Palavras chaves: Segurança da informação, Teste de software, Computação móvel, Desenvolvimento de software.

1. INTRODUÇÃO

O uso intenso da tecnologia da informação e dos sistemas de informação no cotidiano, quer seja para as tarefas mais simples da atividade profissional, quer seja para o relacionamento social ou entretenimento, aliado às evoluções como a da computação móvel e na nuvem, sistemas embarcados e autônomos, entre outras, tem despertado um crescente interesse pela qualidade do software e pela segurança da informação.

Desde os primórdios da era da computação o ser humano apresenta uma desconfiança tácita às informações digitais, produzidas, manipuladas e armazenadas por máquinas cujo funcionamento foge à capacidade de entendimento da imensa maioria da espécie. Tal desconfiança advém, além do provável desconhecimento e da diferença entre as formas de tratamento dessa informação – uma vez que seres humanos não utilizam o sistema binário por definição – do grande volume de informações tratado pelos computadores, e da velocidade com que são capazes de fazê-lo. Nos dias atuais, em que a capacidade de produção, tratamento e armazenamento de informação cresce de forma exponencial, essa insegurança e a desconfiança crescente na qualidade do software e na segurança da informação provida pelos sistemas computadorizados tem um efeito maléfico, ao dificultar o uso de preciosos recursos e mesmo alijar da sociedade da informação pessoas que dela dependem ou poderiam usufruir com grandes benefícios.

Para combater esse receio a melhor estratégia é busca a garantida da qualidade dos softwares e da segurança da informação através de processos simples, efetivos e viáveis do ponto de vista econômico. A simplicidade implica em facilidade de capacitação dos profissionais para seu uso, bem como a rápida implementação em conjunto com processos já existentes. A efetividade possibilita o incremento na confiabilidade do software – imprescindível para fazer frente à sensação de insegurança por parte do usuário. E a viabilidade econômica responde ao incessante apelo à sustentabilidade, estabelecendo um círculo virtuoso e perenizando empresas e produtos que façam frente a esse desafio.

Este trabalho reflete uma avaliação da relação entre os processos de teste de software, a qualidade do software e a segurança da informação, com o intuito de avaliar as implicações e a adequação dos testes ao processo de desenvolvimento de software. O intuito maior é analisar e reforçar a necessidade da aplicação das atividades de testes precocemente e avaliar seus resultados na qualidade do software e na segurança da informação por ele tratada.

1.1. Contexto

O presente trabalho aborda as atividades de testes de software no contexto do processo de desenvolvimento de software com o objetivo de garantir a qualidade do software produzido e a segurança da informação por ele tratada.

São abordadas as etapas do processo de desenvolvimento de software e a aplicabilidade de diversas técnicas e métodos nessas etapas, considerando a viabilidade, a complexidade e as dificuldades inerentes ao processo e os resultados plausíveis e esperados.

1.2. Objetivos do trabalho

Dentre os principais objetivos deste trabalho destacam-se:

- Ressaltar a indissociabilidade entre testes de software, a qualidade do software e a segurança da informação.
- Avaliar diversos métodos e técnicas de teste de software e sua aplicabilidade nas etapas do processo de desenvolvimento de software, bem como sua efetividade quanto à garantia da segurança da informação.
- Prover um embasamento para que se possa avaliar o processo de teste mais adequado aos projetos de desenvolvimento de software em suas diversas etapas.

1.3. Organização do trabalho

Este trabalho está disposto da seguinte maneira: a seção dois apresenta os conceitos básicos das disciplinas abordadas, a saber, segurança da informação, qualidade do software e testes de software. A seção três discorre sobre trabalhos correlatos e considerações sobre os objetivos e resultados dos mesmos. A seção quatro aborda a proposta do trabalho, as limitações, contribuições e melhorias possíveis, e a seção cinco apresenta as conclusões.

2. CONCEITOS BÁSICOS

2.1. Segurança da Informação

De acordo com (RIBEIRO 2002), a segurança da informação é algo tão esperado pelos usuários quanto o desempenho ou o uso otimizado dos recursos computacionais, mesmo que não faça parte das especificações iniciais do software. O usuário de qualquer software espera que esse seja intrinsicamente seguro a despeito de não tê-lo solicitado assim. E caso não receba o que espera, certamente irá reclamar.

Com o uso cada vez mais conectado dos dispositivos computacionais móveis, a crescente utilização desses dispositivos para as atividades profissionais e o crescimento exponencial da capacidade de processamento e armazenamento dos dispositivos, a preocupação com a segurança da informação tornou-se evidente, e as respostas exigidas são, por ora, inferiores aos desafios apresentados.

Embora a segurança da informação não seja dependente apenas do software, pois além do hardware são necessários controles físicos, normas e procedimentos para garanti-la, é no software que se originam a maioria dos problemas de segurança da informação. Segundo (LYRA 2008), a segurança da informação tem como características principais:

- A confidencialidade, que garante acesso à informação apenas àqueles que possuem autorização para tanto;
- A integridade, garantia de que a informação está correta, é verdadeira e não foi corrompida ou alterada (ou permite identificar caso isso tenha ocorrido);
- A disponibilidade, cujo objetivo é assegurar que a informação esteja acessível àqueles que dela dependem para a realização de suas atividades.

Além dessas características podem ser consideradas também:

- A necessidade de autenticação, que garante ao usuário ou ao sistema que seu interlocutor é realmente quem diz ser.

— O não repúdio, que é a capacidade de comprovar quem executou determinada ação.
— A legalidade, que trata da conformidade com os dispositivos legais.
— A privacidade, restringindo, além do acesso indevido, a vinculação de um usuário a uma atividade executada.
— A auditoria, possibilitando o rastreamento e a reconstituição de toda a movimentação e das alterações da informação em função do tempo.

A segurança da informação tem por objetivo garantir essas características, evitando os incidentes de segurança, quer seja, eventos que atentem contra qualquer uma delas, inviabilizando o uso adequado da informação.

2.2. Qualidade do software

Embora a qualidade possa ser considerada um critério subjetivo, ou seja, dependente das características dos indivíduos, a qualidade do software pode ser avaliada em função de sua aderência aos requisitos. Isto leva a definir a qualidade de software, de uma maneira geral, como sendo o grau em que um conjunto de características inerentes a um software ou sistema atende aos requisitos inicialmente estipulados para estes.

Também há que se considerar que a qualidade está vinculada ao processo e, portanto, a elevação da qualidade de um software passa, necessariamente, pela melhoria dos processos empregados para a construção desse software. Nesse aspecto, propostas como as do CMMI (*Capability Maturity Model Integration* – Modelo de maturidade em capacitação – integração) do SEI – *Software Engineering Institute* – da Universidade Carnegie Mellon, e o MPS.BR – Melhoria de Processos de Software Brasileiro – do Governo do Brasil, têm papel fundamental na evolução dos processos de desenvolvimento de software, levando empresas, instituições e profissionais a

capacitarem-se para a evolução dos métodos, ferramentas e processos utilizados no desenvolvimento de software.

É inquestionável que a segurança da informação está diretamente ligada qualidade do software. Considerando o exposto anteriormente, que os requisitos de segurança para o usuário estão implícitos na especificação do software, e que a qualidade depende do atendimento a esses requisitos, tem-se que garantir a qualidade é algo desafiador, o que justifica a busca contínua da excelência nos processos.

2.3. Testes de software

Para garantir a qualidade do software e, portanto, sua aderência aos requisitos – incluindo-se aí aqueles que se referem à segurança da informação, é necessário submeter os produtos de software a processos de validação, verificação e testes.

Até o início da década de 1990 o processo de testes era a última etapa do processo de desenvolvimento de software, com o intuito de comprovar o funcionamento do produto, e conduzido eventualmente pelos próprios desenvolvedores. Essa visão mudou com o trabalho de (MYERS 1979), que afirmava ser função dos testes descobrir os defeitos, e não provar que o software estava funcionando. (GELPERIN e HETZEL 1988) descreveram uma evolução dos testes, propondo o documento Plano de Testes, que deveria ser escrito a partir dos requisitos do software. A partir de então o teste de software passou a ser incorporado como no processo de desenvolvimento de software.

O processo de teste contempla as etapas de planejamento, projeto, execução e avaliação dos resultados dos testes. Os testes são realizados em fases distintas e com finalidades específicas, a saber:

- *Teste de unidade*, voltado para as menores unidades executáveis de software, buscando problemas quanto à estruturação dos dados, lógica e forma de implementação.

— *Teste de integração*, cujo objetivo é identificar falhas na interface entre os módulos do software.
— *Teste de validação*, que confronta o software com os requisitos funcionais e outros, como desempenho por exemplo.
— *Teste de sistema*, cujo intuito é localizar falhas advindas da interação com outros elementos, incluindo-se a segurança da informação.

Considerando essa abordagem fica evidente que o processo de testes é de crucial importância para a qualidade do software e por conseguinte para a segurança da informação, uma vez que falhas no software certamente levam ao comprometimento da confidencialidade, integridade ou disponibilidade da informação.

3. TRABALHOS RELACIONADOS

A área de testes de software é objeto de vastos estudos com o mais amplo grau de enfoques, incluindo-se a abordagem proposta pelo presente trabalho, voltada para a segurança da informação.

Dentre os inúmeros trabalhos avaliados, pode-se citar o estudo sobre a aplicação de técnicas de teste baseadas em software de busca (ANTONIOL 2009), no qual é abordada a questão das dificuldades – ou mesmo impossibilidade - de garantir a segurança do software devido ao seu tamanho, complexidade, extensibilidade e conectividade, além da busca incessante pela redução de custos de produção de software. A proposta busca oferecer uma alternativa de menor custo e adaptável ao software, incluindo a busca por vulnerabilidades conhecidas e com maior risco para o tipo de software que está sendo testado.

Também considerando aspectos de custo e complexidade dos testes voltados para a segurança do software, (CHANDRAMOULI e BLACKBURN 2004) propõem uma ferramenta de teste de segurança funcional com base na modelagem formal e nas informações de interface, oferecendo um framework

de automação de testes (TAF – *Test Automation Framework*) e uma ferramenta baseada nesse framework (TAF-SFT - *Test Automation Framework – Security Funcional Testing*) usando a linguagem formal SCR – *Software Cost Reduction*.

Propondo uma abordagem mista para a solução do problema do oráculo em testes de segurança para softwares de missão crítica, (DONG, GUO e ZHANG 2013) fazem uma combinação de testes baseados em análise de caminho com testes de mutação, reduzindo assim o esforço de teste e mantendo a cobertura dos testes em um nível satisfatório.

Uma discussão interessante é a avaliação das próprias ferramentas de teste e dos métodos propostos pelas mesmas, como o trabalho de (HE, ZHANG e MEI 2008). Os autores do trabalho avaliam o rápido desenvolvimento da tecnologia da computação confiável e do software que a suporta (TCSS - *Trusting Computer Supporting Software*), e propõem um modelo de teste e um protótipo de sistema para validar a aderência às especificações do TCG – *Trusting Computer Group*. O modelo proposto permite então o gerenciamento dos produtos bem como auxilia na escolha entre produtos classificados como aderentes à especificação de computação confiável.

No que concerne às dificuldades encontradas nas áreas de segurança da informação, qualidade de software e testes de software, uma com muita importância diz respeito à taxionomia. Nesse aspecto, (HUI, et al. 2010) propõem uma taxonomia com base em relatos de defeitos de segurança obtidos em entidades confiáveis que registram tais defeitos, com o intuito de auxiliar os profissionais de testes de segurança na busca por tais defeitos e até mesmo expandir tal lista conforme o padrão adotado.

A automatização do processo de testes com base em métodos conhecidos e o emprego sistemático de ferramentas de testes voltadas para a segurança da informação são deficiências abordadas por (KARPPINEN, et al. 2007). Os autores propõem uma abordagem de testes de segurança com base nos objetivos de segurança, nos requisitos de segurança e nos padrões de segurança, e a sua aplicação em um projeto real, expondo assim a enorme

distância entre o modelo teórico de especificação de requisitos de segurança e as vulnerabilidades de segurança encontradas na prática, o que vem a ser um grande desafio na área.

Ainda tratando de geração de testes, porém com abordagem para testes em nível de código binário para arquiteturas específicas, (LI, et al. 2009) propõem uma técnica de geração de testes dinâmicos em nível de código binário baseada em uma ferramenta (ReTBLDTG – *ReTargetable Binary-Level Dynamic Test* Generation) que faz uso de meta-instruções definidas (MetaISA – *Meta Instruction Set Architecture*), as quais permitem a portabilidade para outras arquiteturas de processador. Nos testes realizados com a ferramenta foi obtido o resultado de 100% de identificação de defeitos conhecidos em aplicações em testes com quatro diferentes arquiteturas de processador, o que á algo promissor.

Uma proposta de geração automatizada de testes de segurança com base em modelos formais de ameaças foi apresentada por (XU, et al. 2012). A abordagem utilizada foi o mapeamento de ameaças comuns que exploram comportamentos não intencionais e entradas inválidas como estabelecido pelo modelo STRIDE – (acrônimo para *spoofing identity, tampering with data, repudiation, information disclosure, denial of service, and elevation of privilegie* – roubo de identidade, adulteração de dados, repúdio, divulgação de informações, negação de serviço e elevação de privilégio) e, através de um mapeamento de modelo de implementação, foram criados códigos executáveis e seus mutantes para explorar as vulnerabilidades. Essa geração, assim como o código, foi executada automaticamente, e o resultado obtido foi a morte da maioria dos mutantes gerados, o que é positivo.

O registro de recursos nos diagramas de sequência para testes de segurança em aplicações web, com o objetivo de possibilitar a geração automática de casos de testes, é o objeto do trabalho de (XU, DENG e ZHENG 2012), cuja abordagem preconiza a mescla de mapeamento de recursos e riscos com diagramas de sequência, possibilitando a rastreabilidade dos riscos inerentes a cada caso de uso. Essa abordagem concentra-se nos recursos

utilizados/explorados por usuários legítimos e atacantes das aplicações web para efetivar os testes de segurança.

(HONG, et al. 2012) propõem, em seu trabalho sobre testes de segurança de software orientado a dados, um método cuja abordagem é voltada para o controle de acesso a dados ao invés da abordagem tradicional voltado para o controle de acesso com papéis, permissões e contextos. Segundo apurado pelos pesquisadores, essa abordagem, embora experimental, mostrou que o método é mais instrutivo e facilita a geração automática de testes.

Uma abordagem pragmática baseada em risco voltada para a validação da segurança de sistemas empresariais é proposta em trabalho de (MURTHY, THAKKAR e LAXMINARAYAN 2009). A abordagem baseada em risco proposta busca prover a cobertura necessária sem pressionar os custos e o esforço de testes. Os autores da pesquisa integraram as melhores práticas de segurança com a abordagem baseada em risco como forma de buscar o equilíbrio entre controles de segurança e usabilidade, viabilizar a aplicação de princípios de teste baseados em risco nos testes de segurança de aplicações, reduzindo o tempo de projeto e os custos do desenvolvimento.

Em seu trabalho voltado para o uso de testes com base no modelo SaaS, (RIUNGU, TAIPALE e SMOLANDER 2010) apresentam uma pesquisa realizada com onze corporações com o intuito de mensurar a aparente crescente demanda de serviços de testes on-line, seus benefícios em termos de redução de custos e flexibilidade, a consonância com serviços na nuvem e os problemas relacionados aos tratamento dos dados de testes e à capacitação do pessoal de testes.

No artigo sobre testes de segurança de (TÜRPE 2008) o autor propõe um aprofundamento da pesquisa no campo de testes de segurança, pontuando questões até então não respondidas ou respondidas insatisfatoriamente, de modo que possibilitassem influenciar o mundo real. O autor relata sua vasta experiência em testes de penetração, sua visão crítica quanto à automação desses testes, e propõe uma agenda para pesquisa com foco na teoria das

vulnerabilidades, na teoria dos testes de segurança e em ferramentas e técnicas voltadas para os testes de segurança. Em contato com o autor foi verificado que, apesar de reconsiderar suas restrições quanto à automação de testes de segurança, o mesmo sustenta suas ponderações quanto às demais questões apresentadas à época do trabalho.

4. PROPOSTA

Com base nas informações obtidas, o presente trabalho busca obter respostas quando à adequação e a efetividade dos testes de segurança da informação dos mais diversos métodos, buscando estabelecer um ponto de equilíbrio entre a flexibilidade, facilidade de implementação, recursos necessários e custos dos processos de teste abordados.

Uma vez estabelecidos critérios viáveis de avaliação, busca-se avançar no conhecimento das técnicas e métodos de testes voltados à segurança da informação como forma de prover aplicações e serviços seguros. Entende-se que, devido às constantes evoluções dos recursos, das arquiteturas e das demandas de software, há uma demanda real para pesquisas nesse segmento, espaço no qual esta pesquisa pretende avançar.

É consenso que o início das atividades de testes deve ser o mais antecipado possível dentro do processo de desenvolvimento do software, indiferente ao modelo e às técnicas empregadas. Os requisitos de segurança, que serão a base para a definição dos casos de teste, devem fazer parte do processo formal e serem definidos juntamente com o processo de especificação funcional do software a ser desenvolvido. A partir daí, a produção e o refinamento de casos de teste deve ter sequência à cada iteração, faze ou ciclo do processo de desenvolvimento, de modo que ao chegar à etapa de testes de homologação pelo usuário tais requisitos tenham sido atendidos satisfatoriamente e a segurança implícita – que faz parte da expectativa do usuário – seja efetiva e comprovada.

Para que isso aconteça é necessário que o desenvolvedor comece o processo de desenvolvimento analisando o risco inerente ao uso da informação para a qual o software está sendo projetado. Desta forma estabelecerá os valores de cada informação manipulada ou produzida pelo software, e estabelecerá a prioridade de tratamento da segurança da informação de acordo com esse valor. Em seguida deverá ser analisada a exposição dessas informações ao risco, isto é, sua vulnerabilidade, e as ameaças conhecidas e possíveis a essas vulnerabilidades. Para isso é necessário que o desenvolvedor raciocine de modo diferente, assumindo o papel de atacante, com o intuito de explorar as vulnerabilidades do software, produzindo os *casos de abusos*, cenários nos quais registrará suas ações para violar a segurança da informação tendo como premissa as vulnerabilidades do software e as ameaças conhecidas.

Os casos de abuso serão estão utilizados para a modelagem dos procedimentos de resposta aos ataques – as contramedidas a serem implementadas para eliminar ou reduzir as vulnerabilidades a níveis aceitáveis, conforme definido na análise de risco. Os casos de abuso também possibilitarão a definição dos casos de testes para a experimentação da efetividade das contramedidas estabelecidas.

4.1.1. Limitações

Os principais problemas de uma abordagem precoce dos testes em geral – e dos de segurança em específico – no processo de desenvolvimento de software decorrem da pressão exercida pelos prazos e custos. Além disso, o baixo nível de automação de tarefas de teste, a geração e manipulação de massa de testes e a necessidade de manter constante atualização sobre as técnicas de ataques e as vulnerabilidades exploradas representam um esforço adicional significativo e que coloca o processo de testes em evidência quando há a necessidade de redução de esforço, de custos ou de prazo.

A abordagem tradicional, na qual os procedimentos de teste de segurança são tratados em sua plenitude somente após os ciclos de desenvolvimento,

fomenta uma prática pouco recomendável de recompor o cronograma e efetuar ajustes de esforço ou de custos reduzindo-se as atividades de teste, e assim comprometendo a qualidade do software. Além disso, uma abordagem tardia impede a implementação de um processo cíclico de testes, interagindo com o desenvolvimento e com a análise de riscos constantemente atualizada e revista, o que só é possível conseguir com uma abordagem proativa e assertiva da questão da segurança da informação no processo de desenvolvimento de software.

4.1.2. Contribuição

A proposta deste trabalho é conscientizar os desenvolvedores para a necessidade de tratamento dos requisitos de segurança da informação no início do processo de desenvolvimento do software, o que reforça a qualidade do software em geral e possibilita uma abordagem proativa e assertiva da segurança da informação.

Nesse aspecto a pesquisa apresentou algumas das iniciativas mais efetivas no que concerne à testes de software com o foco em segurança da informação, provendo um embasamento seguro e modelos de técnicas, práticas e ferramentas que podem ser adotadas com o fim proposto de antecipar as discussões de segurança da informação e de testes no processo de desenvolvimento de software, contribuindo assim tanto para a argumentação que reforça essa necessidade quanto para a adoção desse modelo como resposta efetiva aos problemas de segurança da informação.

4.1.3. Melhorias

Em decorrência dos principais problemas decorrentes da abordagem proposta pelo presente trabalho, há a necessidade de prosseguir no desenvolvimento de técnicas e ferramentas para reduzir o encargo que os testes de software impõem ao processo de desenvolvimento.

A forma mais promissora de contribuição para esse fim é a automação dos procedimentos de teste, notadamente os voltados para a segurança da informação, através da criação de ferramentas específicas que, além da execução dos testes propriamente ditos, permitam a elaboração de análises de risco consistentes, mantenham uma base de dados atualizada com os riscos, as vulnerabilidades e as contramedidas mais adequadas e, com maior ênfase, permita a elaboração de casos de abuso com a respectiva derivação para os casos de teste.

5. CONCLUSÃO

O presente trabalho procurou apresentar a necessidade de incorporar as atividades voltadas para o teste de software – especialmente aquelas com o enfoque de segurança da informação – às atividades iniciais do processo de desenvolvimento de software.

Para isso é proposta a elaboração de casos de abuso com base na análise de riscos aos quais a informação está submetida, a avaliação de vulnerabilidades e as contramedidas aplicáveis. Com base nessas atividades elaboram-se também os casos de teste o mais antecipadamente possível, de forma que a construção do software já privilegie o tratamento desses riscos.

Embora essa abordagem seja um consenso entre os desenvolvedores de software em geral, ainda representa uma prática incipiente, notadamente em função do aumento do esforço, do prazo e consequentemente dos custos envolvidos. É necessário prosseguir com as pesquisas no intuito de simplificar os procedimentos, oferecendo métodos, técnicas e ferramentas que possibilitem a aplicação prática dessa abordagem que impliquem minimamente nos aspectos de esforço, prazo e custos, viabilizando assim a aplicação dessa abordagem.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ANTONIOL, Giuliano. “Search Based Software Testing for Software Security: Breaking Code to Make it Safer.” *IEEE International Conference on Software Testing Verification and Validation Workshops.* IEEE Computer Society, 2009. 88-100.

CHANDRAMOULI, Ramaswamy, e Mark BLACKBURN. “Automated Testing of Security Functions using a combined Model & Interface driven Approach.” *Proceedings of the 37th Hawaii International Conference on System Sciences - 2004.* IEEE, 2004. 1-10.

DONG, Guowei, Tao GUO, e Puhan ZHANG. “Security Assurance with Program Path Analysis and Metamorphic Testing.” IEEE, 2013. 193-197.

GELPERIN, David, e Bill HETZEL. “The growth of software testing.” *Communications of the ACM Volume 31 Issue 6, June*, 1988: Pages 687-695.

HE, Fan, Huanguo ZHANG, e Tang MEI. “A Test Method of Trusted Computing Supporting Software.” *The 9th International Conference for Young Computer Scientists.* IEEE Computer Society, 2008. 2330-2334.

HONG, Yu, Xiao-ming LIU, Song HUANG, e Chang-you ZHENG. “Data Oriented Software Security Testing.” *2012 Second International Conference on Instrumentation & Measurement, Computer, Communication and Control.* IEEE Computer Society, 2012. 676-679.

HUI, Zhanwei, Song HUANG, Bin HU, e Zhengping REN. “A Taxonomy of Software Security Defects for SST.” *978-1-4244-6837-9.* IEEE, 2010. 99-103.

KARPPINEN, Kaarina, Reijo SAVOLA, Mikko RAPELI, e Esa TIKKALA. “Security Objectives within a Security Testing Case Study.” *0-7695-2775-2/07.* IEEE Computer Society, 2007.

LI, Gen, Kai LU, Ying ZHANG, Xicheng LU, e Wei ZHANG. “Decoupling Binary-Level Dynamic Test Generation from Specific Architecture Details.” *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology.* IEEE Computer Society, 2009. 1041-1046.

LYRA, Maurício Rocha. *Segurança e Auditoria em Sistemas de Informação.* Rio de Janeiro: Ciência Moderna, 2008.

MURTHY, K. Kishna, Kalpesh R THAKKAR, e Shirsh LAXMINARAYAN. “Leveraging Risk Based Testing in Enterprise Systems Security Validation.” *2009 First International Conference on Emerging Network Intelligence.* IEEE Computer Society, 2009. 111-116.

MYERS, Glenford. *The Art of Software Testing.* Hoboken, New Jersey: Word Association, Inc., 1979.

RIBEIRO, Ricardo Albuquerque Bruno. *Segurança no Desenvolvimento de Software.* Rio de Janeiro: Campus, 2002.

RIUNGU, Leah Muthoni, Ossi TAIPALE, e Kari SMOLANDER. “Software Testing as an Online Service: Observations from Practice.” *Third International Conference on Software Testing, Verification, and Validation Workshops.* IEEE Computer Society, 2010. 419-423.

TÜRPE, Sven. “Security Testing: Turning Practice into Theory.” *IEEE International Conference on Software Testing Verification and Validation Workshop (ICSTW'08).* IEEE Computer Society, 2008.

XU, Danxiang, Manghui TU, Michael SANFORD, Lijo THOMAS, Daniel WOODRASKA, e Weifeng XU. “Automated Security Test Generation with Formal Threat Models.” *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 9, NO. 4, JULY/AUGUST*, 2012: 526-540.

XU, Weifeng, Lin DENG, e Qing ZHENG. “Annotating Resources in Sequence Diagrams for Testing Web Security.” *2012 Ninth International Conference on Information Technology- New Generations.* IEEE Computer Society, 2012. 874-875.