Uma Revisão Sistemática em Teste de Segurança Baseado em Modelos

Carlos Diego Nascimento Damasceno[1], **Márcio Eduardo Delamaro**[1], **Adenilso da Silva Simão**[1]

[1] Instituto de Ciências Matemáticas e de Computação – ICMC
Universidade de São Paulo – USP
Av. Trabalhador São-carlense, 400 – 13566-590 – São Carlos – SP – Brasil

damascenodiego@usp.br, {delamaro,adenilso}@icmc.usp.br

Abstract. *Model-Based Testing (MBT) is a testing approach which uses explicit formal models to specify and automatize test case generation. It has been successfully applied in funcional testing, however there are still challenges in non-functional requirements testing, as security. This paper presents a systematic literature review about model-based security testing. An MBT taxonomy was used as reference for data extraction to identify tendencies and gaps of research. Even though it was not possible to correlate all the studies found with each dimension of the taxonomy, a predominance of the transition-based modeling paradigm and an increasing number of published papers per year were identified.*

Resumo. *Teste baseado em modelos (TBM) é uma variante de teste que usa modelos explícitos formais para especificar e automatizar a geração de testes. Ela tem sido utilizada com sucesso no teste funcional, entretanto ainda existem desafios no teste de requisitos não funcionais, como segurança. Este artigo apresenta uma revisão sistemática sobre teste de segurança baseado em modelos. Uma taxonomia proposta para TBM foi usada como base para a extração de dados para identificar tendências e lacunas de pesquisa. Apesar de não ter sido possivel estabelecer uma correlação integral entre os estudos e as dimensões da taxonomia, foi identificado um crescente número de publicações nesta área e uma predominância do paradigma de modelagem baseados em transição.*

1. Introdução

A popularização de serviços baseados em mobilidade, virtualização e conectividade têm contribuido fortemente não somente com a flexibilização de negócios de empresas, mas também com o aumento de vulnerabilidades que elas estão se expondo [Rashid et al. 2013]. Consequentemente, a demanda por técnicas de garantia de qualidade, como o teste de software, tem crescido significativamente. Entretanto, o teste de software ainda é frequentemente caracterizado como uma atividade custosa, tediosa e mal documentada [Utting et al. 2012]. Para tratar isso, uma variante denominada teste baseado em modelos propõe o uso de modelos explícitos em notação formal para especificar sistemas em teste (do inglês, *System Under Test* - SUT) e automatizar a geração de testes [Utting and Legeard 2007]. A fim de caracterizar as abordagens de teste de segurança baseado em modelos (TSBM), foi realizada uma revisão sistemática

visando coletar evidências que apontassem os principais resultados obtidos, desvantagens, limitações, tendências e características destas abordagens com base na taxonomia proposta em [Utting and Legeard 2007] para TBM.

A partir dos dados extraídos identificamos que nem todas as dimensões da taxonomia do TBM estão evidentes nos estudos. Foi observada uma predominância no uso do paradigma de modelagem baseado em modelos de transição e de SUTs reais, além de um crescente número de publicações nos ultimos anos.

Na seção 2 é discutido o conceito de segurança da informação, seguido da definição de teste baseado em modelos e da sua taxonomia (3) e de revisão sistemática (4). Após isso, na seção 5, são mostradas as questões de pesquisa, bases de dados e *strings* de busca usadas neste estudo sistemático. Os resultados obtidos são discutidos na seção 6. Em 7 são expostas as ameaças a validade deste estudo e por fim, em 8, são mostradas as conclusões deste estudo.

2. Segurança da Informação

Segurança da informação é a área da computação que lida com o entendimento das diversas questões ligadas às estratégias de defesa e ataque virtual para preservar a confidencialidade, integridade e disponibilidade de dados e recursos computacionais [Jang-Jaccard and Nepal 2014]. Em segurança, mecanismos como *firewalls*, sistemas de detecção de intrusão e de controle de acesso [Jang-Jaccard and Nepal 2014] são usados para monitorar redes de computadores, mediar acesso a recursos e alertar eventos anômalos, como exfiltração de dados.

Exfiltração de dados consiste no vazamento não autorizado de dados sensitivos de sistemas [Sharma et al. 2013]. É um tipo de ameaça de difícil detecção, pois usualmente utiliza-se de técnicas sofisticadas para explorar multiplos canais de comunicação, tem levado empresas a realizar altos investimentos em segurança [Brewer 2014] e que pode se originar tanto em indivíduos internos quanto externos a uma organização e causar danos financeiros significativos [Sharma et al. 2013].

3. Teste Baseado em Modelos

Teste Baseado em Modelos é uma variante de teste de software que permite automatizar a geração de casos de teste a partir de modelos comportamentais e critérios de seleção sistemáticos [Utting and Legeard 2007]. É uma proposta que busca solucionar alguns dos problemas enfrentados pelas metodologias tradicionais de teste que usualmente tendem a ser manuais, mal documentadas e custosas [Utting and Legeard 2007]. O processo de teste baseado em modelos pode ser dividido em cinco etapas [Utting and Legeard 2007]: Modelagem, Geração, Concretização, Execução e Análise.

Na etapa de *Modelagem*, a especificação do SUT é feita com modelos explícitos em notação formal. *Statecharts* e máquinas de estados finitos são exemplos de notações de modelagem que podem ser usadas [Neto et al. 2008]. Durante a *Geração*, critérios de seleção são usados para analisar os modelos do SUT e gerar casos de teste abstratos. Estes critérios podem se basear na cobertura estrutural de um modelo, no domínio de entrada ou no modelo estatístico do SUT [Neto et al. 2008]. Na *Concretização*, os casos de teste abstratos são mapeados para um formato executável que pode ser um código interpretável

ou compilável [Utting and Legeard 2007]. Na *Execução*, os casos de teste concretos são aplicados sobre o SUT. Esta etapa pode ser combinada com a concretização, sendo então denominado teste *online*. Casos de teste também podem ser convertidos para formatos não executáveis, como relatórios [Utting and Legeard 2007]. Na *Análise*, os resultados obtidos durante a execução são avaliados com ajuda de um mecanismo denominado oráculo de teste que determina a aprovação ou falha de casos de teste [Ammann and Offutt 2008].

3.1. Taxonomia do TBM

Idealmente, o desenvolvimento ou adoção de uma abordagem TBM deve ser feito de forma criteriosa e embasada nas necessidades do testador, características do SUT e limitações de projeto [Utting and Legeard 2007]. Para auxiliar nisso, a taxonomia proposta em [Utting and Legeard 2007] pode ser usada para caracterizar abordagens de TBM. A taxonomia TBM, ilustrada na Figura 1, possui sete dimensões: Assunto, Independência, Características, Paradigma, Critério de Seleção de Testes, Tecnologia e Online/Offline.

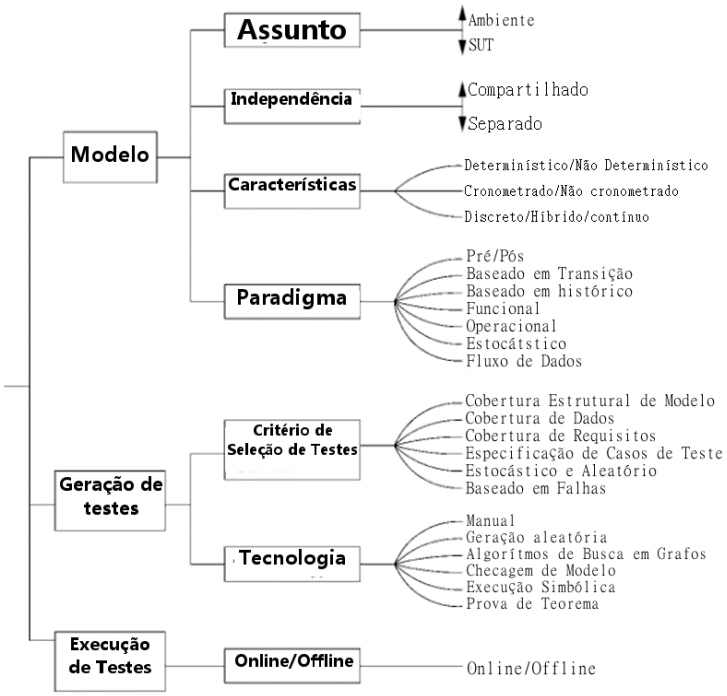

Figura 1. Taxonomia do Teste Baseado em Modelos [Utting and Legeard 2007]

Assunto diz respeito ao elemento descrito que pode ser o SUT e/ou seu ambiente. *Independência* descreve a origem do modelo que pode ser a própria especificação do SUT ou ser independente dela. *Característica* descreve o comportamento modelado que pode ser determinístico, de tempo real, entre outros. *Paradigma* diz respeito à notação usada na modelagem. *Critério de seleção de testes* define o critério de cobertura do modelo. *Tecnologia* diz respeito ao potencial de automação. *Online/Offline* descreve a execução dos casos de teste. TBM também pode combinar múltiplas características de modelo, paradigmas, critérios de seleção e tecnologias [Ammann and Offutt 2008].

4. Revisão Sistemática

Revisão sistemática (RS) é o termo utilizado para descrever metodologias de pesquisa voltadas para a coleta e analise de evidências sobre tópicos de estudo específicos [Biolchini et al. 2005]. Uma RS permite resumir evidências sobre fenômenos ou tecnologias, identificar lacunas de pesquisa e fornecer um arcabouço de conhecimento que posicione novas propostas de pesquisa [Kitchenham and Charters 2007]. Ela pode ser descrita em três etapas: Planejamento, Condução e Documentação.

No *Planejamento*, definem-se os objetivos e o protocolo da RS. Questões de pesquisa (do inglês, *Research Question* - RQ), *strings* e bases de busca, critérios de inclusão, exclusão e extração de dados são definidos nesta etapa. Posteriormente, na *Condução*, é realizada a busca dos estudos. Os estudos encontrados são analisados usando critérios de inclusão e exclusão e, posteriormente, são extraídos dados que ajudem a responder as RQs assim como pré-estabelecido no protocolo. Após a condução, durante a *Documentação*, as informações obtidas são organizadas na forma de um relatório ou artigo científico. Materiais suplementares podem ser disponibilizados para complementar a RS e viéses e ameaças à validade também devem ser evidenciados e discutidos.

5. Execução da Revisão Sistemática

Para esta RS foi estabelecido o objetivo de categorizar os estudos que apliquem TBM no teste de segurança de serviços baseados em mobilidade, virtualização e conectividade segundo a taxonomia do TBM. A partir disso, foram definidas as seguintes questões de pesquisa:

- RQ1 Que resultados o uso de TSBM em serviços ligados a mobilidade, virtualização ou conectividade tem proporcionado?
- RQ2 Quais as características predominantes nos estudos de TSBM?
- RQ2.1 Considerando a taxonomia do TBM, como as abordagens de TSBM podem ser categorizadas?
- RQ2.2 Que tipo de SUTs estas abordagens utilizam em seus estudos empíricos?
- RQ2.3 Quais as desvantagens e limitações destes estudos?
- RQ2.4 Estes estudos podem ser aplicados no tratamento de exfiltração de dados?

Para responder as questões supracitadas, a *string* de busca de [Dias Neto et al. 2007], uma RS sobre TBM em um nível mais geral, foi adaptada usando termos relacionados à segurança da informação, vazamento e exfiltração de dados. Exfiltração de dados foi incluída pois ela é considerada uma ameaça crítica e decisiva em ataques virtuais [Sharma et al. 2013] e, para garantir a segurança de sistemas computacionais, mecanismos de segurança devem passar por processos de teste criteriosos. A *string* de busca definida foi: *("security testing"OR "security"OR "data exfiltration"OR "data extrusion"OR "data theft"OR "data leakage"OR "intrusion"OR "malware"OR "vulnerability") AND ("threat modeling"OR "model based"OR "model based testing"OR "model based security testing")*

A *string* de busca foi adaptada e aplicada nas bases IEEE Xplore Digital Library, ScienceDirect, ACM Digital Library e SpringerLink. Os artigos retornados foram gerenciados usando o *software* Mendeley Desktop [1] e uma planilha eletrônica para extração de dados.

[1] `http://www.mendeley.com/`

A seleção dos estudos foi dividida em duas etapas. Na primeira fase os artigos retornados tiveram o título, resumo e palavras-chave analisados para identificar quais necessariamente discutiam teste baseado em modelos, realizavam algum experimento e abordavam alguma questão ligada a segurança da informação. Estudos não inseridos nesta categoria foram automaticamente excluídos. Na segunda etapa, estudos duplicados ou com menos de 3 páginas foram descartados. Este número mínimo de páginas foi estabelecido a fim de remover resumos e *short papers* desta análise.

Após a identificação, os estudos foram analisados visando extrair informações relacionadas as dimensões da taxonomia do TBM [Utting and Legeard 2007]. Além disso, o tipo de ameaça tratada pela técnica, o tipo de SUT e a possibilidade de ser aplicado no teste de serviços baseados em mobilidade, conectividade ou virtualização ou no tratamento de exfitração de dados também foram extraídos.

6. Resultados

Ao ser aplicada a *string* de busca nas bases de artigos, foi obtido um total de 227 artigos. Desses 227, aproximadamente 22% foram oriundos da IEEE Xplorer, 38% da ScienceDirect, 38% da Springer e 2% da ACM. Eles tiveram seus títulos, resumos e palavras-chave lidos e avaliados segundo os critérios de inclusão e exclusão estabelecidos. Nesta primeira etapa da análise, foram eliminados cerca de 78% dos estudos, restando 50 artigos.

Na segunda etapa, os 50 artigos foram reanalisados visando identificar aqueles que realmente discutissem TSBM e realizassem experimentos. Após esta segunda avaliação, restaram 23 artigos. Estes 23 artigos foram analisados a fim de extrair informações que evidenciassem sua relação com a taxonomia do TBM. Foram extraídos destes estudos, o título do artigo, base de origem, ano de Publicação, as informações que correspondessem as dimensões da taxonomia TBM (Assunto, Independência, Características, Paradigmas, Critérios de Seleção, Tecnologia e Online/Offline), tipo de ameaça modelada, tipo de SUT testado, relação com mobilidade, virtualização, conectividade e exfiltração de dados, vantagens, desvantagens e limitações, sempre que estas informações fossem encontradas. Ao término da extração, os dados extraídos foram analisados a fim de responder as RQs. Na tabela 1 podem ser vistos o título, ano de publicação e base de origem dos 23 estudos retornados pelas bases ACM, IEEE Xplorer (IEEE), ScienceDirect (SD) e Springer.

6.1. Questões de Pesquisa

Com relação a **RQ1**, foi identificado que o TSBM permitiu a especificação do comportamento de atacantes e de sistemas [Salas et al. 2007], uma redução na ambiguidade do plano de testes [Barletta et al. 2011] e facilitou a geração [Fourneret et al. 2011] e replicação [Xu et al. 2012] de testes.

Além disso, 7 dentre os 23 trabalhos foram categorizados como possibilitando o reuso de modelos de especificação, como RFCs (*Request for Comments*) [Rütz and Schmaltz 2011]. Vale ressaltar que este número aumenta se for considerado que 2 trabalhos incluiram reuso em seus trabalhos futuros [Bozic and Wotawa 2013] [Lebeau et al. 2013]. Outro ponto identificado foi que praticamente todos os estudos permitiam a modelagem tanto de dados de entradas como de saída de SUTs. Isso mostra o poder de automação que abordagens de TBM para segurança podem proporcionar.

Tabela 1. Lista dos 23 estudos identificados

Título do Artigo	Ano	Base
A test-based security certification scheme for web services	2013	ACM
An Approach to Modular and Testable Security Models of Real-world Health-care Applications	2011	ACM
A Methodology for Building Effective Test Models with Function Nets	2012	IEEE
A Model-based Approach to the Security Testing of Network Protocol Implementations	2006	IEEE
A Model-Based Fuzzing Approach for DBMS	2013	IEEE
Data vulnerability detection by security testing for Android applications	2013	IEEE
APSET, an Android aPplication SEcurity Testing tool for detecting intent-based vulnerabilities	2014	Springer
Model-Based Vulnerability Testing for Web Applications	2013	IEEE
XSS pattern for attack modeling in testing	2013	IEEE
Testing access control and obligation policies	2013	IEEE
EMV Card: Generation of Test Cases based on SysML Models	2013	SD
Using Labeled Transition System Model in Software Access Control Politics Testing	2012	IEEE
Automated Security Test Generation with Formal Threat Models	2012	IEEE
An Experience Report on an Industrial Case-Study about Timed Model-Based Testing with UPPAAL-TRON	2011	IEEE
Model-Based Security Verification and Testing for Smart-cards	2011	IEEE
Mutation Analysis of Magento for Evaluating Threat Model-Based Security Testing	2011	IEEE
Verified Firewall Policy Transformations for Test Case Generation	2010	IEEE
Model-Based Security Vulnerability Testing	2007	IEEE
Test Generation from Security Policies Specified in Or-BAC	2007	IEEE
A declarative two-level framework to specify and verify workflow and authorization policies in service-oriented architectures	2010	Springer
Fault coverage of Constrained Random Test Selection for access control: A formal analysis	2010	SD
A systematic approach to integrate common timed security rules within a TEFSM-based system specification	2012	SD
Robustness testing for software components	2010	SD

Quanto ao contexto de aplicação, somente 6 estudos identificados aplicavam TBM no teste de serviços ligados a mobilidade [Salva and Zafimiharisoa 2014] [Salva and Zafimiharisoa 2013], virtualização [Barletta et al. 2011] [Brucker et al. 2010] [Wang et al. 2013] ou conectividade [Allen et al. 2006].

Com relação a **RQ2**, foi constatada uma tendência crescente na quantidade de publicações ao longo dos anos (Figura 2 à esquerda). Além disso, foi identificada uma predominância de trabalhos usando modelos baseados em transição (Figura 2 à direita), como máquinas de estados finitos [Bozic and Wotawa 2013], autômatos finitos estendidos [Anisetti et al. 2013] [Li et al. 2007] [Yu et al. 2012] e autômatos temporizados [Rütz and Schmaltz 2011]. Trabalhos usando variantes de redes de petri [Xu and Chu 2012], incluídos na categoria de paradigma operacional [Utting and Legeard 2007], subconjuntos da Unified Modeling Language (UML) [Lei et al. 2010] também foram identificados, assim como abordagens híbridas [Anisetti et al. 2013] [Salas et al. 2007].

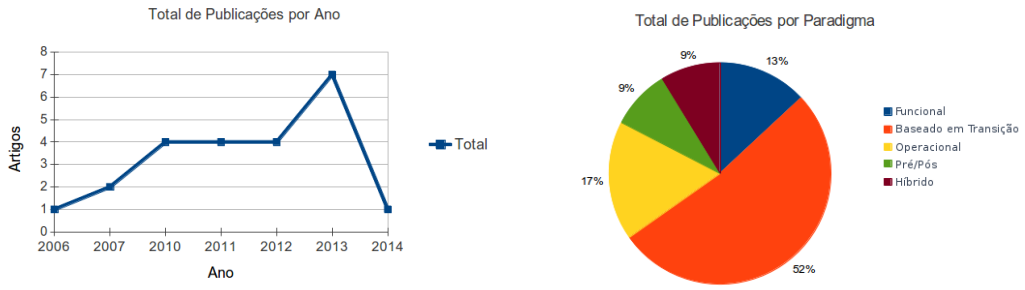

Figura 2. Total de publicações por ano e por paradigma

A questão **RQ2.1** mostrou que nem todos os estudos apresentam claramente informações que preencham as dimensões da taxonomia do TBM. Foi identificado que algumas informações como critérios de seleção e tecnologia de geração de casos de teste utilizados não estavam tão claras ou foram omitidas.

Além da dimensão *Paradigma*, foi possível extrair de todos os estudos o *Assunto*

descrito nos modelos onde somente 1 dos 23 estudo foi categorizado como apoiado na modelagem do SUT e do seu ambiente [Lebeau et al. 2013] e os demais foram categorizados como apoiando somente a modelagem do SUT. A *Independência* predominante foi de modelo separado. As dimensões *Características* de modelo, *Critério de seleção de teste*, *Tecnologia* e *Online/Offline* não estava clara em todos os estudos. Dos estudos em que estas informações puderam ser extraídas, percebeu-se uma predominância de abordagens de teste *offline*, do uso de tecnologias de geração de testes baseadas em busca e em execução simbólica, critérios de seleção de cobertura estrutural e de modelos expressando características determinísticas, de tempo real e eventos discretos. O fato destes estudos não terem evidente certas informações pode dificultar a realização de estudos sistemáticos.

No contexto da **RQ2.2**, foi identificada uma predominância de 74% de estudos experimentais usando sistemas reais ou de grande porte como SUT. Dentre os SUTs foram encontrados programas para *smart-cards* [Fourneret et al. 2011] [Ouerdi et al. 2013], clientes FTP *open-source* [Hsu et al. 2008], softwares da área de saúde [Brucker et al. 2011], políticas de segurança de *firewalls* [Brucker et al. 2010], aplicações móveis reais [Salva and Zafimiharisoa 2014] [Salva and Zafimiharisoa 2013], sistemas bancários [Xu et al. 2013], gerenciadores de conteúdo [Thomas et al. 2011] e sistemas de tempo real [Mammar et al. 2012]. Isso mostra que TSBM é potencialmente usável no contexto de sistemas reais.

Quanto a **RQ2.3**, foi constatado que algumas das abordagens não cobrem o processo de TBM por completo, da modelagem à geração de casos de teste concretos. O estudo de [Ouerdi et al. 2013], por exemplo, não cobre a geração de casos de teste executáveis. O reuso de modelos também é citado como limitação em alguns dos estudos [Bozic and Wotawa 2013] [Lebeau et al. 2013] porém é incluído como trabalhos futuros. O esforço para a modelagem dos sistemas e do comportamento de atacantes também é citado como uma desvantagem [Xu et al. 2012].

Quanto a **RQ2.4**, nenhum estudo tratou especificamente sobre exfiltração. Entretanto, alguns afirmam permitir a modelagem de ameaças como a "quebra de sigilo de dados" e de requisitos de confidencialidade [Xu et al. 2012] [Anisetti et al. 2013] [Wang et al. 2013] que possuem relação com a exfiltração de dados. Isso sugere que o TBM talvez possa ser usado para mitigar ou detectar este tipo de ameaça.

7. Ameaças à Validade

O fato da *string* de busca desta RS ter considerado somente alguns termos da usada em [Dias Neto et al. 2007] foi identificado como uma ameaça que pode ter limitado o alcance deste trabalho. Entretanto, a decisão de descartar uma parte dos termos se embasou no fato que durante um estudo piloto percebeu-se uma grande quantidade de artigos não relacionados a TBM para segurança sendo retornada. A não inclusão de termos referenciando notações específicas, como "máquinas de estados finitos" e "redes de petri", também pode ter limitado o alcance desta RS. Entretanto, considera-se em um trabalho futuro a extensão desta *string* com sinônimos para TBM e nomes de notações de modelagem de SUT específicas.

8. Conclusão

Há um número crescente de publicações sobre teste de segurança baseado em modelos. Os estudos identificados permitiram automatizar a geração e replicação de casos de testes de segurança e redução de ambiguidade no projeto. Entretanto, o fato de haver um maior esforço na modelagem pode afetar sua usabilidade e interferir na sua adoção.

As produções científicas identificadas usam sistemas reais como sistema em teste onde tanto o comportamento esperado como os dados de entrada são modelados. Vários tipos de notações são utilizadas para descrever sistemas em teste. Entretanto, foi identificada uma predominância de abordagens usando modelos baseados em transição, como máquinas de estados finitos.

Nem todos os estudos puderam ser categorizados usando as dimensões da taxonomia de [Utting and Legeard 2007] pois muitas das informações não estavam claras ou explicitadas. Os resultados obtidos apontam que é pequena a quantidade de estudos descrevendo suas abordagens por completo, considerando a explicitação das dimensões da taxonomia teste baseado em modelos como parâmetro de completude. Isso pode dificultar a execução de revisões e mapeamentos sistemáticos que tenham como foco dimensões da taxonomia que não tenham sido totalmente identificadas neste estudo.

Além disso, foi identificada uma lacuna de pesquisas em teste baseado em modelos aplicado na mitigação e detecção de exfiltração de dados. Como trabalhos futuros, considera-se a realização de uma outra revisão sistemática extendendo a *string* de busca com termos relacionados a notações de modelagem específicas, como "máquinas de estados", "redes de petri" entre outros, ampliando o seu alcance.

Referências

Allen, W., Dou, C., and Marin, G. (2006). A model-based approach to the security testing of network protocol implementations. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 1008–1015.

Ammann, P. and Offutt, J. (2008). *Introduction to Software Testing*. Cambridge University Press.

Anisetti, M., Ardagna, C. A., Damiani, E., and Saonara, F. (2013). A test-based security certification scheme for web services. *ACM Trans. Web*, 7(2):5:1–5:41.

Barletta, M., Ranise, S., and Viganò, L. (2011). A declarative two-level framework to specify and verify workflow and authorization policies in service-oriented architectures. *Service Oriented Computing and Applications*, 5(2):105–137.

Biolchini, J., Mian, P. G., and Natali, A. C. C. (2005). Systematic review in software engineering. Technical Report RT-ES 679/05, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

Bozic, J. and Wotawa, F. (2013). Xss pattern for attack modeling in testing. In *Automation of Software Test (AST), 2013 8th International Workshop on*, pages 71–74.

Brewer, R. (2014). Advanced persistent threats: minimising the damage. *Network Security*, 2014(4):5 – 9.

Brucker, A., BruÌgger, L., Kearney, P., and Wolff, B. (2010). Verified firewall policy transformations for test case generation. In *Software Testing, Verification and Validation (ICST), 2010 Third International Conference on*, pages 345–354.

Brucker, A. D., Brügger, L., Kearney, P., and Wolff, B. (2011). An approach to modular and testable security models of real-world health-care applications. In *Proceedings of the 16th ACM Symposium on Access Control Models and Technologies*, SACMAT '11, pages 133–142, New York, NY, USA. ACM.

Dias Neto, A. C., Subramanyan, R., Vieira, M., and Travassos, G. H. (2007). A survey on model-based testing approaches: A systematic review. In *Proceedings of the 1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies*, WEASELTech '07, pages 31–36, New York, NY, USA. ACM.

Fourneret, E., Ochoa, M., Bouquet, F., Botella, J., Jurjens, J., and Yousefi, P. (2011). Model-based security verification and testing for smart-cards. In *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*, pages 272–279.

Hsu, Y., Shu, G., and Lee, D. (2008). A model-based approach to security flaw detection of network protocol implementations. In *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*, pages 114–123.

Jang-Jaccard, J. and Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, 80(5):973 – 993. Special Issue on Dependable and Secure Computing The 9th {IEEE} International Conference on Dependable, Autonomic and Secure Computing.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Keele University and Durham University Joint Report.

Lebeau, F., Legeard, B., Peureux, F., and Vernotte, A. (2013). Model-based vulnerability testing for web applications. In *Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference on*, pages 445–452.

Lei, B., Li, X., Liu, Z., Morisset, C., and Stolz, V. (2010). Robustness testing for software components. *Science of Computer Programming*, 75(10):879 – 897. Selected papers of the 5th International Workshop on Formal Aspects of Component Software (FACSâ08).

Li, K., Mounier, L., and Groz, R. (2007). Test generation from security policies specified in or-bac. In *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, volume 2, pages 255–260.

Mammar, A., Mallouli, W., and Cavalli, A. (2012). A systematic approach to integrate common timed security rules within a tefsm-based system specification. *Inf. Softw. Technol.*, 54(1):87–98.

Neto, A., Subramanyan, R., Vieira, M., Travassos, G., and Shull, F. (2008). Improving evidence about software technologies: A look at model-based testing. *Software, IEEE*, 25(3):10–13.

Ouerdi, N., Azizi, M., louis Lanet, J., Azizi, A., and Ziane, M. (2013). Emv card: Generation of test cases based on sysml models. *IERI Procedia*, 4(0):133 – 138. 2013 International Conference on Electronic Engineering and Computer Science (EECS 2013).

Rashid, A., Ramdhany, R., Edwards, M., Kibirige, S. M., Babar, A., Hutchison, D., and Chitchyan, R. (2013). Detecting and preventing data exfiltration report. `http://goo.gl/epKO48`. Acessado em 10 de maio de 2014.

Rütz, C. and Schmaltz, J. (2011). An experience report on an industrial case-study about timed model-based testing with uppaal-tron. In *Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEE Fourth International Conference on*, pages 39–46.

Salas, P., Krishnan, P., and Ross, K. (2007). Model-based security vulnerability testing. In *Software Engineering Conference, 2007. ASWEC 2007. 18th Australian*, pages 284–296.

Salva, S. and Zafimiharisoa, S. (2013). Data vulnerability detection by security testing for android applications. In *Information Security for South Africa, 2013*, pages 1–8.

Salva, S. and Zafimiharisoa, S. (2014). Apset, an android application security testing tool for detecting intent-based vulnerabilities. *International Journal on Software Tools for Technology Transfer*, pages 1–21.

Sharma, P., Joshi, A., and Finin, T. (2013). Detecting data exfiltration by integrating information across layers. In *2013 IEEE 14th International Conference on Information Reuse & Integration (IRI)*, pages 309–316. IEEE.

Thomas, L., Xu, W., and Xu, D. (2011). Mutation analysis of magento for evaluating threat model-based security testing. In *Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*, pages 184–189.

Utting, M. and Legeard, B. (2007). *Practical Model-Based Testing: A Tools Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Utting, M., Pretschner, A., and Legeard, B. (2012). A taxonomy of model-based testing approaches. *Software Testing, Verification and Reliability*, 22(5):297–312.

Wang, J., Zhang, P., Zhang, L., Zhu, H., and Xiaojun, Y. (2013). A model-based fuzzing approach for dbms. In *Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on*, pages 426–431.

Xu, D. and Chu, W. (2012). A methodology for building effective test models with function nets. In *Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual*, pages 334–339.

Xu, D., Sanford, M., Liu, Z., Emry, M., Brockmueller, B., Johnson, S., and To, M. (2013). Testing access control and obligation policies. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 540–544.

Xu, D., Tu, M., Sanford, M., Thomas, L., Woodraska, D., and Xu, W. (2012). Automated security test generation with formal threat models. *Dependable and Secure Computing, IEEE Transactions on*, 9(4):526–540.

Yu, H., Song, H., Bin, H., and Yi, Y. (2012). Using labeled transition system model in software access control politics testing. In *Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012 Second International Conference on*, pages 680–683.