TESTE DE INTEGRIDADE DE ARQUIVOS ELETRÔNICOS BASEADO EM SISTEMAS IMUNOLÓGICOS ARTIFICIAIS

Thyago MENEZES e Cleonilson PROTÁSIO
CEFET-MA, Av. Getulio Vargas, Nº 4 Monte Castelo. São Luís-MA.
CEP: 65030-000. (98) 3218 9046
(1) e-mail: thyagoma@gmail.com
(2) e-mail: protasio@cefet-ma.br

Resumo. Um sistema natural que pode ser modelo para um detector de mudança é o sistema imunológico biológico por conter vários mecanismos de reconhecimento e de defesa contra organismos patogênicos. A principal finalidade desse sistema é reconhecer todas as células dentro do corpo e classificá-las como próprias ou não-próprias ao corpo. Com essa habilidade de detecção de células não-próprias, o sistema imunológico aparenta ser uma adequada fonte de inspiração para o desenvolvimento de algoritmos de detecção de comportamento anômalo em sistemas. Vários estudos já foram realizados baseando-se nesses fatos, resultando em uma nova área da inteligência artificial, os Sistemas Imunológicos Artificiais. Por outro lado, na sociedade moderna, trocas de informações através de meios eletrônicos de comunicação, por exemplo, pela Internet, tornaram-se atividades essenciais no nosso dia-a-dia e um dos meios principais de armazenamento de informações são os arquivos eletrônicos. Nesse contexto, um princípio fundamental emerge: quanto maior é a importância das informações e da privacidade desses, maior tem que ser a segurança no seu armazenamento para que não haja modificações indesejáveis ou intencionais. Neste trabalho é apresentado um analisador de integridade de arquivos eletrônicos baseado em sistema imunológico artificiais que permite ao usuário verificar a ocorrência de alguma modificação no arquivo após sua transmissão ou armazenamento.

Palavras-chave: Sistemas Imunológicos, Arquivos, Proteção, Integridade.

1. INTRODUÇÃO

Na sociedade moderna, trocas de informações através de meios eletrônicos de comunicação, como, por exemplo, a Internet, tornaram-se atividades essenciais no nosso dia-a-dia. Um dos principais meios de armazenamento de informações são os arquivos eletrônicos. Nesse contexto, um princípio fundamental emerge: quanto maior é a importância das informações e da privacidade desses, maior tem que ser a segurança no seu armazenamento para que não haja modificações indesejáveis ou intencionais.

Por outro lado, um sistema natural que pode ser modelo para um detector de mudança é o sistema imunológico biológico por conter vários mecanismos de defesa contra organismos patogênicos (COSTA BRANCO, 2003). A principal finalidade do sistema imunológico é reconhecer todas as células dentro do organismo e classificá-las como próprias ou não-próprias ao organismo. Com essa habilidade de detecção de células não-próprias, o sistema imunológico aparenta ser uma adequada fonte de inspiração para o desenvolvimento de algoritmos de detecção de comportamento anômalo em sistemas. Vários estudos já foram realizados baseados nesses fatos, resultando em uma nova área da inteligência artificial, os Sistemas Imunológicos Artificiais (SOUZA, 2005; SOUZA, 2004; DE CASTRO, 2003; DASGUPTA, 1997).

Um dos principais algoritmos que compõem o repertório dos Sistemas Imunológicos Artificiais é o Algoritmo de Seleção Negativa (ASN) proposto por Forrest et al (FORREST, 1994). Este algoritmo é utilizado para a detecção de mudanças e é baseado no princípio de discriminação próprio/não-próprio do sistema imunológico biológico. No trabalho de FORREST (1994) foi aplicado o ASN para a detecção de vírus de computador. Outras aplicações do ASN já foram desenvolvidas, como exemplo, tem-se: teste de motores elétricos (DASGUPTA, 1997) e testes de circuitos integrados (SOUZA, 2005).

Baseando-se no Algoritmo de Seleção Negativa, neste trabalho é apresentado o desenvolvimento de um analisador de integridade de arquivos eletrônicos. A motivação deste trabalho deu-se pelo fato que inúmeros documentos eletrônicos, na forma de arquivos, necessitam de alguma constatação que eles foram ou não modificados de forma indesejável ou intencional na sua transmissão ou armazenamento. Dessa forma, a ocorrência de alguma modificação do arquivo poderá ser verificada pelo usuário receptor.

2. ALGORITMO DE SELEÇÃO NEGATIVA

Forrest et al. (FORREST, 1994) propuseram um Algoritmo de Seleção Negativa (ASN) para detecção de mudanças baseado no princípio de discriminação próprio/não-próprio que ocorre em sistemas imunológicos biológicos. Essa discriminação é realizada em parte por células do tipo T, nas quais têm receptores sobre sua superfície que podem detectar proteínas estranhas (antígenos).

Durante o processo de geração das células T, que ocorre nos primeiros meses de vida do corpo, essas células passam por um processo genético de seleção aleatória. Desta forma, essas células são submetidas a um processo de seleção no Timo chamado de seleção negativa onde as células T que reagem com as proteínas próprias do corpo são eliminadas, desta forma somente as células T que não casam com nenhuma célula própria do corpo são permitidas sair do Timo. Essas células T maturadas circulam através do corpo para realizar suas funções imunológicas na proteção contra antígenos estranhos.

O ASN funciona de forma similar, gerado futuros detectores (células T imaturas) aleatoriamente e eliminando aqueles que casam (detectam) com componentes que são considerados próprios, de maneira que os detectores restantes posam detectar qualquer componente não-próprio (DASGUPTA, 1997).

O ASN é baseado nos seguintes procedimentos:

1. Definição dos dados a serem protegidos. Definem-se os componentes próprios de um sistema ou uma coleção que necessita ser protegida ou monitorada, como uma coleção R de *strings* de comprimento L sobre um alfabeto finito.

2. Fase de Maturação. Gera-se um conjunto D de detectores, no qual cada um desses não detecta qualquer *string* em R. Ao invés de usar um processo de casamento exato ou perfeito, o método usa um dado critério de casamento parcial (por exemplo, um critério de casamento parcial pode consistir em comparar duas *strings* da seguinte forma: essas duas *strings* casam se e somente se elas forem idênticas em pelo menos c posições contínuas, onde c é um parâmetro escolhido adequadamente). Essa fase de maturação é ilustrada na Figura 1. Os detectores candidatos são gerados aleatoriamente e, então, testados para verificar se eles casam com alguma *string* própria. Se um casamento é encontrado, o candidato é rejeitado. Esse processo é repetido até que um número desejado de detectores seja alcançado.

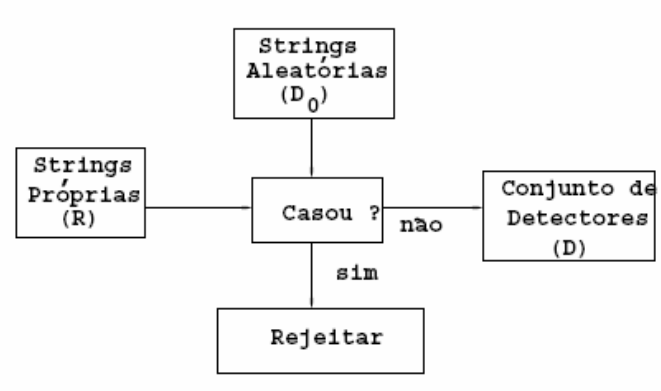

Figura 1. Fase de maturação do ASN.

3. Fase de Monitoramento. Monitora-se R para verificação de mudanças por continuamente verificar se detectores em D casam com R. Se qualquer detector em algum momento casar com algum elemento de R, então uma mudança em R ocorreu, pois os detectores foram projetados de forma a não casar com nenhuma *string* original em R. Essa fase de monitoramento é ilustrada na Figura 2. A fase de maturação é a que consome mais recursos e existem vários conjuntos de detectores para cada conjunto de dados a ser protegido. Por outro lado, a fase de monitoramento é a menos custosa (FORREST, 1994). A maior limitação na geração dos detectores é devido ao fato de ser computacionalmente difícil gerar detectores válidos nos quais cresce exponencialmente com o tamanho do conjunto R. Alguns algoritmos para geração de detectores aplicados no ASN foram propostos a fim de aumentar a eficiência na geração de detectores válidos, como pode ser visto em D'HAESELEER (1996) e HELMAN (1994).

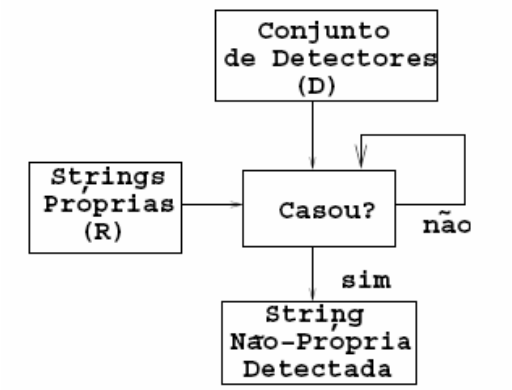

Figura 2. Fase de monitoramento do ASN.

Se o conjunto de dados a ser protegido for visualizado como um conjunto de *strings* pertencentes a um alfabeto finito e uma mudança nesse conjunto como uma *string* que não pertença ao conjunto original, então o ASN propõe-se a gerar detectores para (quase) todas *strings* que não pertença ao conjunto de

dados original e é matematicamente viável no sentido que um pequeno conjunto de detectores tem uma alta probabilidade de verificar uma mudança aleatória no conjunto original (FORREST, 1994). De acordo com Dasgupta et al. (1997), esse algoritmo baseia-se em três princípios fundamentais:

1 - Cada realização do ASN é única, ou seja, cada conjunto de *strings* detectoras é único;

2 - A detecção é probabilística; e

3 - Um sistema robusto deve detectar (probabilisticamente) qualquer atividade estranha ao invés de procurar por padrões específicos de mudanças conhecidos.

Como um exemplo, suponha que se tenha oito *strings* de 4 bits a serem protegidas contra mudanças, dadas por:

R = {0010, 1000, 1001, 0000, 0100, 0010, 1001, 0011}

A coleção R será a coleção de *strings* próprias. A fase de maturação consistirá em gerar *strings* aleatórias (D0) e então verificar se as *strings* em D0 casam com as *strings* em R. As *strings* de D0 que casam com qualquer *string* própria são rejeitadas. As *strings* que não casam com nenhuma *string* própria tornam-se membros da coleção de *strings* detectoras (D).

Suponha que D0 contenha as seguintes 4 *strings* aleatórias:

0111, 1000, 0101, 1001

Então, D consistirá de duas *strings*, 0111 e 0101, pois as *strings* 1000 e 1001 são rejeitadas devido o fato que cada uma delas casam com uma *string* em R. Na prática, o procedimento consiste em gerar *strings* aleatórias continuamente até que o conjunto D tenha um número suficiente de membros.

Na fase de monitoramento, com a coleção de detectores D, o estado do conjunto próprio pode ser monitorado por continuamente verificar se *strings* em R casam com alguma *string* em D.

Suponha que um bit da última *string* própria (0011) mude, produzindo a *string* 0111. Então, em algum ponto do monitoramento, deve-se verificar que a *string* não-própria 0111 casa com algum dos detectores (neste caso, a *string* detectora 0111), e uma mudança em R será relatada.

3. CRITÉRIOS DE CASAMENTO PARCIAL

No Algoritmo de Seleção Negativa é necessário definir um critério de casamento para a realização das fases de maturação e monitoramento. Isso é necessário, pois um casamento perfeito entre *strings* é extremamente raro para *strings* de comprimento razoável (FORREST, 1994).

Um critério de casamento proposto em FORREST (1994) é o de verificar se duas *strings* casam em c posições contínuas correspondentes. Esse critério é denominado de c-contiguous Suponha que duas *strings* x e y tenham símbolos em um pré-definido alfabeto finito. Então, é dito que a função match(x,y) é verdadeira se x e y casam (tenham símbolos iguais) em pelo menos c posições contínuas. Por esse critério de casamento, a probabilidade P_M que duas *strings* aleatórias casem em pelo menos c posições contínuas é dada por:

$$P_M \approx q^{-r}\left[\frac{(l-r)\,(q-1)}{q}+1\right] \qquad (1)$$

em que l é o comprimento da *string* própria, q é o número de símbolos do alfabeto e q << 1 (BRADLEY, 2002).

Por exemplo, as *strings* abaixo casam para todo c ≤ 4.

0**0100**010

1**0100**101

4. TESTE DE INTEGRIDADE DE ARQUIVOS ELETRÔNICOS

Baseando-se no Algoritmo de Seleção Negativa descrito na seção anterior, apresenta-se aqui a proposta de um analisador de integridade de arquivos eletrônicos que tem como objetivo fornecer ao usuário um meio de verificar, com alta probabilidade de acerto, se um documento eletrônico foi modificado após este ser recebido por alguma outra pessoa ou após ter sido armazenado em algum sistema de armazenamento de arquivos.

Foram desenvolvidos para isso programas de computador baseado em JAVA e disponível na Internet no endereço www.dee.cefet-ma.br/~protasio/sia.htm que operacionaliza os procedimentos bases para a proteção proposta.

Na Figura 3 é vista a fase de efetivação da proteção que consiste basicamente em gerar um arquivo auxiliar que contém um conjunto de detectores gerados a partir da fase de geração de detectores do ASN.

FASE DE PROTEÇÃO

Passo 1. Aplique o arquivo, (ex: TESTE.DOC) a ser protegido no programa DETGEN;

Passo 2. Gere o arquivo TESTE.SIA contendo os imuno-detectores;

Passo 3. Transmita ou armazene o arquivo TESTE.DOC.

Figura 3. Fase de efetivação da proteção de arquivo.

Na Figura 4 é vista a fase de efetivação da verificação se o arquivo protegido contra mudanças foi modificado ou não. Esta fase consiste em executar a fase de monitoramento do ASN utilizando o conjunto de detectores gerados e presentes no arquivo auxiliar.

FASE DE VERIFICAÇÃO

Passo 1. Execute o programa de verificação de modificação de TESTE.DOC usando os detectores salvos em TESTE.SIA.

Passo 2. Se algum detector em TESTE.SIA casar com alguam self-string em TESTE.DOC,

Então: TESTE.DOC modificador (FALHA DE INTEGRIDADE)

Caso contrário: TESTE.DOC íntegro (OK).

Figura 4. Fase de efetivação da verificação de arquivo.

5. CONCLUSÃO

Foi apresentado neste trabalho um analisador de integridade de arquivos eletrônicos baseado no Algoritmo de Seleção Negativa. Foi descrito o funcionamento do Algoritmo de Seleção Negativa e apresentado a idéia principal e o princípio de funcionamento do analisador de integridade proposto.

Foram realizados testes usando os analisador de integridade e, em todas as modificações intencionalmente feitas o programa teve alta eficiência na detecção de falha de integridade (modificação). Especificamente: para um arquivo de 100 caracteres, o analisador detectou falha de integridade para somente um caractere foi modificado. Para um arquivo de 10.000 caracteres, o analisador detectou falha de integridade para 1% dos caracteres foram modificados. Em geral, para modificações acima de 1% do tamanho do arquivo, o analisador obteve 100% de efetividade.

REFERÊNCIAS

BRADLEY, D.W.; TYRRELL, A.M. Immunotronics - Novel Finite-State-Machine Architectures with Built-In Self-Test Using Self-Nonself Differentiation. **IEEE Transactions on Evolutionary Computation**.Vol. 6, pp. 227–238, Jun 2002.

COSTA BRANCO, P. J.; DENTE, J. A.; Vilela Mendes, R. Using Immunology Principles for Fault Detection. **IEEE Transactions on Industrial Electronics**. 30(2):302–375, 2003.

D'HAESELEER; FORREST, S.; HELMAN, P. An Immunological Approach to Change Detection: Algorithms, Analysis, and Implication. In: IEEE Symposium on Research in Security and Privacy. **Proceedings…** May 1996.

DASGUPTA, D.; ATTOH-OKINE, N.. Immunity-based systems: a survey. In: IEEE International Conference on Systems, Man, and Cybernetics, vol. 1, pp. 369-374. **Proceedings…** 1997.

DE CASTRO, L.N.; TIMMIS, J. I. Artificial immune systems as a novel soft computing paradigm. **Soft. Computing journal**, Julho de 2003.

FORREST, S.; PERELSON, A. S.; ALLEN, L.; CHERUKURI, R.. Self-Nonself discrimination in a computer. In: 1994 IEEE Symposium on Security and Privacy. **Proceedings…** Maio de 1994.

HELMAN, P.; FORREST, S. **An Efficient Algorithm for Generating Random Antibody Strings**. Technical Report No. CS94-7, Department of Computer Science, 1994.

LUH, G.; CHENG, E. W. Immune model-based fault diagnosis. **Math. Comput. Simul**. 67, 6, 515-539. Janeiro de 2005.

SOUZA, C. P.; Assis, F. M.; Freire, R. C. S. Circuit Testing Using Self-Nonself Discrimination. In: IEEE Instrumentation and Measurement Technology Conference. Ottawa, CA, v. 2. p. 1186-1189. **Proceedings…** 2005.

SOUZA, C. P.; Assis, F. M.; Freire, R. C. S.. A BIST scheme based on self-nonself discrimination of immune system. In: 2004 IEEE Signal Processing Society Workshop. , 2004. p.765 - 774, São Luís. **Proceedings…** 2004.

AGRADECIMENTOS

Agradecemos à Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão (FAPEMA) pela concessão de bolsa de Iniciação Científica Júnior; ao Centro Federal de Educação Tecnológica do Maranhão (CEFET-MA) pelo total apoio ao desenvolvimento desta e de outras pesquisas comprovando assim sua vocação para a tríade ensino, pesquisa e extensão; e ao grupo de pesquisadores do Laboratório de Sistemas Digitais e Instrumentação Eletrônica (LaDiG) do Departamento de Eletro-Eletrônica do CEFET-MA pela sinergia de conhecimentos.