Um módulo de detecção e resposta a intrusões baseado na proteção inata do Sistema de Segurança Imunológico

Victor Hugo de Oliveira Amaducci[1], **Paulo Lício de Geus**[2]

Instituto de Computação - Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6.176 – 13083-970 – Campinas – SP – Brasil

victor@las.ic.unicamp.br, paulo@las.ic.unicamp.br

Abstract. *This paper presents a module that uses the functionality offered by a kernel framework to implement a security system that detects intrusion and generates an active response to unaccepted process behaviour.*

1. Introdução

O projeto Imuno compreende uma arquitetura de segurança computacional proposta em [de Paula 2004]. Essa arquitetura tem como base um *framework* chamado Imuno que faz a interface entre os módulos de segurança e o *kernel*. Neste trabalho será apresentado um módulo de segurança que utiliza o Imuno para efetivar detecção e resposta a intrusões. O sistema imunológico humano é composto pelos sistemas inato e adaptativo, dos quais o primeiro é o foco deste trabalho. O sistema inato reage de maneira semelhante a todas as substâncias estranhas, representando a primeira linha de defesa contra a ação de micróbios, e sua resposta, por não ser específica para um determinado micróbio, é insuficiente, na maioria das vezes. Trabalhos como [e F. González 2001] inspiraram a criação de uma arquitetura de segurança baseada no sistema imunológico, [G. Tedesco 2006] inspirou a detecção de intrusão baseando-se em análise de chamadas de sistemas.

2. Arquitetura geral

Para complementar a arquitetura proposta em [de Paula 2004], este trabalho apresenta um módulo de segurança denominado Módulo de Proteção Inata - MprotI. Este módulo utiliza o *framework* Imuno, introduzido em [Carbone 2005], para detectar e gerar respostas a intrusões. MprotI implementa algumas funcionalidades do sistema de resposta primária de um sistema imunológico biológico. Essas funcionalidades visam a tomada de ações de precaução, não definitivas, que seguem a suspeita de um ataque, geralmente em função da detecção de uma anomalia no sistema.

O módulo MprotI é executado em espaço de usuário, utilizando funcionalidades internas do *kernel* do Linux através do Imuno. O MprotI utiliza *hooks* (funções/comandos) implementados no *kernel* através do *Linux Security Modules* - LSM e *Netfilter* para realizar a detecção de intrusão. Utilizando a funcionalidade CGroups (implementação do *Class-based Kernel Resource Management* – CKRM no *kernel* Linux) o MprotI pode controlar/regular os recursos de *hardware* (CPU, memória e rede) utilizados por processos atacados como forma de resposta a intrusões.

3. Protótipo do módulo MprotI

O MprotI é responsável por implementar um mecanismo de proteção que vai agir, de forma geral, sobre qualquer evento que esteja fazendo o sistema funcionar de forma anormal ou que não faça parte do funcionamento normal do sistema. O objetivo do MprotI

é identificar, retardar o progresso e mitigar os primeiros efeitos de um ataque, para que uma análise forense possa ser conduzida, a ameaça identificada e uma resposta específica elaborada.

3.1. Detecção e Resposta

O MprotI constrói uma máquina de estados para cada processo (que lhe for determinado) através do monitoramento de suas chamadas de sistema. Cada chamada de sistema é um estado, portanto um processo que acabou de executar a chamada de sistema *Bi* encontra-se no correspondente estado *Bi*; a solicitação seguinte de uma chamada de sistema, *Bj*, provoca a mudança de estado do processo de *Bi* para *Bj*. Um processo é definido como suspeito quando sua execução apresenta transições que não estão previstas em sua máquina de estado. A resposta primária produzida pelo MprotI pode ser comparada ao mecanismo da febre, comandado pelo sistema imunológico humano: o sistema entra em "modo de alerta", sendo constantemente avisado sobre irregularidades e garantindo reserva de recursos para suas reações. O MprotI disponibilizará os seguintes agentes de resposta:

- Sinalizador que informa o sistema operacional sobre uma intrusão detectada;
- Controle de recursos, como uso de memória e de processador;
- Monitoramento sobre criação de processos filhos;
- Mecanismo que torna espaços de memória de processos suspeitos (e de seus filhos) totalmente *read-only* (memória virtual);

Processos monitorados são agrupados em classes para permitir o estabelecimento de limites de uso de recursos. Exemplo: Processos da Classe A podem utilizar no máximo 30% da CPU, Processos da Classe B podem utilizar no máximo 5% da memória. Logo, se um processo for determinado como suspeito sua utilização de recursos é reduzida de forma que o sistema de segurança retarde o progresso do ataque que está sendo proferido e garanta maior disponibilidade aos processos que satisfazem a política normal de execução do sistema. Um algoritmo estabelecerá dinamicamente o grau de limitação de uso dos recursos, inclusive até a limitação total, de acordo com a gravidade da situação identificada.

4. Conclusão

MprotI contribuirá para o projeto Imuno acrescentando uma primeira barreira de proteção ao sistema operacional, utilizando-se de um método de resposta inovador que está atualmente sendo desenvolvido: o controle de recursos de *hardware*, que combinado com outras contra medidas poderá ser bastante eficaz no retardamento e contenção de intrusões.

Referências

Carbone, M. P. A. (2005). Kernel framework for an immune-based security system: A work-in-progress report. Simpósio Brasileiro de Segurança em Sistemas Computacionais.

de Paula, F. S. (2004). Uma arquitetura de segurança computacional inspirada no sistema imunológico. Biblioteca Digital da Unicamp.

e F. González, D. D. (2001). An immunity-based technique to characterize intrusions in computer networks. IEEE Transactions on Evolutionary Computation.

G. Tedesco, J. T. e. U. A. (2006). Integrating innate and adaptive immunity for intrusion detection. 5th International Conference on Artificial Immune Systems.