Análise Comparativa de Segurança Entre Sistemas Operacionais de Uso Corrente e Sistemas Operacionais Voltados a Segurança

Karolina Cristina Pacheco, Tiago Costa da Silva

Curso de Bacharelado em Ciência da Computação – Universidade do Vale do Rio dos Sinos (UNISINOS) – Campus de São Leopoldo
93020-190 – São Leopoldo – RS– Brasil

{karolinacp, tiagocsilva}@edu.unisinos.br

Abstract. *This paper describes why the security in an operational system is important and what are the mechanisms that can be used to ensure a user of the operational system will be protected. A comparative analysis between current use Operational Systems and security based Operational Systems is shown with examples of which security mechanisms are used in each one.*

Resumo. *Este artigo descreve o porquê a segurança em um sistema operacional é importante e quais são os mecanismos que podem ser aplicados para garantir ao usuário do sistema operacional que ele estará protegido. Uma análise comparativa entre os sistemas operacionais de uso corrente e os sistemas operacionais voltados à segurança é mostrada, com exemplos de quais mecanismos cada um dos sistemas operacionais utiliza.*

1. Introdução

Os sistemas operacionais surgiram com dois objetivos principais: criar uma camada de abstração entre o hardware e as aplicações e gerenciar os recursos de forma eficiente [Tanenbaum & Woodhull, 1997]. No começo do desenvolvimento dos sistemas operacionais as preocupações de segurança eram mínimas pois os computadores estavam isolados e a única forma de invasão possível era a presencial.

Com o passar do tempo, os sistemas operacionais, foram sendo aprimorados, foram criadas redes para interligar computadores, e os sistemas operacionais passaram a suportar a execução de múltiplas tarefas simultaneamente, permitindo que duas ou mais pessoas compartilhassem o tempo de processamento de uma mesma máquina [Garfinkel e Spafford, 1996, Tanenbaum & Woodhull, 1997]. Com a chegada da internet usuários domésticos passaram a fazer parte das redes, e uma vasta quantidade de informações tornou-se acessível a qualquer pessoa [Nakamura e de Geus, 2002]. Surge, então, a necessidade de tornar os sistemas operacionais seguros, já que os dados agora podem ser compartilhados.

A segurança de um sistema de computação diz respeito à garantia de algumas propriedades fundamentais associadas às informações e recursos presentes nesse sistema [Maziero 2017].

2. Segurança em Sistemas Operacionais de Uso Corrente

Por sistema operacional de uso corrente entende-se os sistemas operacionais mais utilizados em organizações e computadores pessoais. Neste grupo, são enquadrados os derivados do Unix (Linux, Solaris, Fedora), o Windows (2000, XP, W10) e o MacOS. A segurança desses sistemas operacionais reside nos direitos de acesso de um usuário (ou grupo de usuários) a um recurso (arquivo, periférico, memória, etc.). Para estes sistemas operacionais tem-se três paradigmas de segurança:

Usuário como limitador de segurança: o usuário, ao criar arquivos e aplicações, associa a estes permissões referentes à leitura, escrita e execução, para si próprio, para usuários do mesmo grupo, e para os demais usuários do sistema [Garfinkel e Spafford, 1996]. Este conjunto de permissões associados aos arquivos é denominado domínio. Dessa forma, é responsabilidade do usuário restringir os arquivos/recursos que julgar relevantes. De forma similar, quando uma aplicação é executada, esta terá acesso ao domínio do usuário que a executou, podendo interagir com aplicações e/ou arquivos da mesma forma que o usuário do qual obteve o domínio.

Concentração e isolamento dos direitos privilegiados do computador: os sistemas operacionais possuem um usuário, denominado superusuário ou administrador, que tem acesso a todos os recursos e ao qual são restritos todas as operações privilegiadas [Garfinkel e Spafford, 1996]. O domínio do superusuário compreende acesso irrestrito a todos os recursos do computador, inclusive os arquivos e aplicações pertencentes aos demais usuários. Aos usuários comuns e suas aplicações só é permitido o acesso a um subconjunto dessas operações privilegiadas, limitado pelas decisões do núcleo do S.O, através de uma API. O acesso as operações privilegiadas é total ou nenhum, já que a API é responsável por controlar o acesso indireto realizado pelas aplicações. Assim, existem aplicações que por necessitarem de acesso a alguma operação privilegiada não provida pela API, acabam tendo que usar o domínio do superusuário, tendo acesso irrestrito a todas as operações privilegiadas.

Cifragem das informações: Existe um variado conjunto de técnicas de cifragem, empregadas tanto para mensagens a serem enviadas pela rede de forma segura [Tanenbaum, 2003] quanto para armazenar os arquivos em disco rígido [Silbertschatz et al., 2002]. Com a cifragem, o usuário tem a garantia de que, mesmo que alguém consiga acesso não autorizado ao arquivo ou mensagem, esta será ilegível sem a chave apropriada.

Derivado desses paradigmas surgiu o conceito de que as aplicações devem prover a segurança que está faltando e que podem adequadamente garantir tal segurança sem auxílio do sistema operacional [Loscocco et al., 1998]. Uma vez comprometida uma aplicação, todo o domínio ao qual ela estava associada estará comprometido.

Fica sob responsabilidade do usuário preocupar-se com a segurança de seus arquivos, através da configuração correta de permissões, ou através da cifragem dos documentos importantes. Mesmo sobre o superusuário recai essa obrigação, dado o tipo de acesso que possui ao sistema operacional.

3. Sistemas Operacionais voltados a Segurança

A necessidade de que o sistema operacional garanta a segurança dos arquivos, aplicações e recursos do computador contra o eventual comprometimento de uma aplicação é inegável [Loscocco et al., 1998]. Partindo desse princípio, podem ser derivados três novos paradigmas de segurança:

Controle de acesso mandatório (MAC): consiste em um conjunto de políticas de segurança que limitam as permissões que o usuário dispõe sobre arquivos e aplicações. Surge, então, a necessidade de um administrador de políticas de segurança, distinto e isolado do administrador do sistema operacional, cuja responsabilidade é definir quais as políticas a serem aplicadas, visando evitar que os usuários, inclusive o superusuário, exponham acidentalmente arquivos ou aplicações que possam comprometer o sistema operacional. Assim, o superusuário também se submete às políticas, solucionando falhas provindas da concentração e isolamento dos direitos privilegiados do computador.

Privilégio Mínimo: consiste na ideia de que a um usuário ou aplicação não sejam dados nenhum privilégio a mais do que o necessário para realizar suas atividades [Ferraiolo e Kuhn, 1992]. O problema do comprometimento de uma aplicação acontece quando a mesma tem acesso ao mesmo domínio que o usuário que a está executando, ao restringir a aplicação a um subdomínio desse domínio o estrago que poderá vir a ser causado será menor.

Proteger os mecanismos da API: o S.O é responsável por proteger a API contra ataques de spoofing, contorno e interferência. Para que isso ocorra, precisa ser garantido que os mecanismos como canais de comunicação, mecanismos de criptografia, controle de acesso, acesso aos dispositivos de entrada e saída, etc. não tenham seu comportamento alterado seja por qualquer aplicação ou qualquer outra vulnerabilidade do sistema, o que reduz em muito a chance de invasões indesejadas.

Existem diversos sistemas operacionais que foram desenvolvidos com foco nos novos paradigmas de segurança. Destacam-se os baseados em microkernel, que preveem mecanismos primitivos de proteção e que visam suportar a construção de uma arquitetura de segurança flexível e de alto nível. Alguns apresentam limitação em relação ao suporte de controle de acesso mandatório, mas com alto grau de flexibilidade e validação, como o Exokernel [Mazieres e Kaashoek, 1997]. Os mais relevantes são o Flask e o DTOS. Ambos apresentam uma implementação coerente do gerenciador de políticas e restritor de segurança [Loscocco et al., 1998].

4. Security Enhanced Linux - SELinux

Desenvolvido pela agência nacional de segurança dos EUA, a NSA, é um núcleo para o Linux com suporte a controle mandatório [Loscocco e Smalley, 2001]. O SELinux provê uma política de segurança sobre todos os processos e objetos do sistema baseando suas decisões em etiquetas contendo uma variedade de informações relevantes à segurança. A lógica da política de tomada de decisões é encapsulada dentro de um simples componente conhecido como servidor de segurança e toma como base o princípio do mínimo privilégio ao extremo, restringindo até o usuário root. Utiliza um conjunto de regras - conhecidos coletivamente como uma política - para autorizar ou proibir as operações. Os direitos de um processo dependem de seu contexto de

segurança, que é definido pela identidade do usuário que iniciou o processo, o papel e o domínio que o usuário realiza naquele momento. Na prática, o kernel consulta o SELinux antes de cada chamada do sistema para saber se o processo está autorizado a fazer a operação dada. Basicamente, o SELinux pode operar em três tipos diferentes: Enforcing – as regras do SELinux estão aplicadas, e estão sendo gerados logs de todas as operações do mesmo; Permissive – as regras do SELinux estão desativadas, porém, está gerando logs de todas as operações do mesmo; Disabled - as regras e os logs do SELinux estão completamente desativados.

Algumas distribuições Linux já implementam no seu núcleo o sistema SELinux de segurança, tais como Fedora 26, CentOS e Debian. O Ubuntu utiliza o sistema de segurança AppArmor que foi originalmente desenvolvido porque achava-se que o SELinux era complexo demais para que usuários típicos o gerenciassem. O AppArmor inclui um modelo MAC totalmente configurável [Jones, 2008].

5. Conclusão

Os paradigmas de segurança em sistemas operacionais de uso corrente são eficientes se restritos ao contexto em que foram desenvolvidos, com computadores isolados ou pequenas redes isoladas. Nestes casos, o número de atacantes potenciais é extremamente reduzido, restrito às pessoas que têm acesso às máquinas. Entretanto, o quadro muda drasticamente ao interligar esses sistemas ou redes à internet, pois o número de atacantes potenciais aumenta drasticamente. Também é comum o usuário negligenciar a configuração correta das permissões sobre seus arquivos, por não saber como fazê-lo ou por descuido. Outro problema está relacionado com o domínio associado às aplicações, que possuem acesso ao mesmo domínio do usuário que as executou, e, caso sejam comprometidas por um ataque bem-sucedido, o invasor terá acesso ao mesmo domínio do usuário. Depender exclusivamente das aplicações para garantir a segurança é extremamente perigoso, pois elas estão se tornando cada vez mais complexas e garantir a inexistência de falhas é praticamente impossível.

Ao observar qualquer organização é perceptível que os computadores de suas redes ainda utilizam largamente os sistemas operacionais baseados nos paradigmas de segurança dos S.O de uso corrente, seja por parte dos usuários, por causa das aplicações ou devido a questões financeiras envolvendo licenças de sistemas operacionais, ou seja, a substituição por sistemas voltados à segurança não é fácil.

Sendo a substituição do sistema operacional inviável, a solução seria adaptar os sistemas operacionais existentes para se adequarem aos novos paradigmas. O SELinux é uma maneira de contornar este problema, ele pode ser integrado a sistemas Linux atuais como também já é entregue em algumas plataformas.

Referências

Amoroso, E. “Fundamentals of Computer Security Technology”. Prentice Hall PTR, 1994.

Ferraiolo, D. e Kuhn, R. “Role-based access control”. 15ª Conferência Nacional de Segurança em Computação, 1992.

Garfinkel, S. e Spafford, G. “Pratical Unix & Internet Security”. O’Reilly & Associates, Inc., Estados Unidos, 2ª edição, p. 971, 1996.

Jones, M. “Anatomia do Security-Enhanced Linux (SELinux)”. Disponível: https://www.ibm.com/developerworks/br/library/l-selinux/index.html, abril de 2008.

Loscocco, P. e Smalley, S. “Integrating flexible support for security policies into the linux operating system”. FREENIX: USENIX Annual Technical Conference, Boston, 2001.

Loscocco, P. A., Smalley, S. D., Muckelbauer, P. A., Taylor, R. C., Turner, S. J., e Farrel, J. F. “The inevitability of failure: The flawed assumption of security in modern computing environment”. 21ª Conferência Nacional de Segurança de Sistemas da Informação. Páginas 303–314, 1998.

Mazieres, D. e Kaashoek, M. “Secure applications need flexible operating systems”. 6º Workshop de Tópicos de Sistemas Operacionais, 1997.

Maziero, Prof. Carlos Alberto. “Sistemas Operacionais: Conceitos e Mecanismos VIII - Aspectos de Segurança”. Disponível: http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=so:so-cap08.pdf, agosto de 2017.

Nakamura, E. e de Geus, P. L. “Segurança de Redes em ambientes cooperativos”. Editora Berkeley, São Paulo, 1ª edição, 2002.

Silbertschatz, P. B., Galvin, P. B., e Gagne, G. “Operating System Concepts”. John Wiley & Sons, Inc., Nova York, 6ª edição, p.887, 2002.

Tanenbaum, A. S. e Woodhull, A. S. “Operating systems: Design and Implementation”. Prentice Hall, Nova Jersey - Estados Unidos, 2ª edição, 1997.

Tanenbaum, A. S. “Computer Networks”. Prentiche Hall, Nova Jersey - Estados Unidos, 4ª edição, 2003.