UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
ESPECIALIZAÇÃO EM TELEINFORMÁTICA E REDES DE COMPUTADORES

EDDY SANDRO GODOY VAZ

ESTUDO COMPARATIVO ENTRE OS SUBSISTEMAS DE SEGURANÇA GNU/LINUX: SELINUX E APPARMOR

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA
2013

EDDY SANDRO GODOY VAZ

ESTUDO COMPARATIVO ENTRE OS SUBSISTEMAS DE SEGURANÇA GNU/LINUX: SELINUX E APPARMOR

Monografia apresentada como requisito parcial à obtenção do título de Especialista em Teleinformática e Redes de Computadores, do Departamento Acadêmico de Eletrônica da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Armando Rech Filho

CURITIBA

2013

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria de Pesquisa e Pós-Graduação do *Campus* Curitiba
Departamento Acadêmico de Eletrônica
Especialização em Teleinformática e Redes de Computadores

TERMO DE APROVAÇÃO

Estudo Comparativo Entre os Subsistemas de Segurança GNU/Linux: SELinux e AppArmor

por

Eddy Sandro Godoy Vaz

Esta monografia foi apresentada às 18:40 h do dia 26 de AGOSTO de 2013 como requisito parcial para a obtenção do título de ESPECIALISTA EM TELEINFORMÁTICA E REDES DE COMPUTADORES, Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado com a nota NOVE INTEIROS

Prof. Dr. Armando Rech Filho
(UTFPR)

Prof. Dr. Walter Godoy Júnior
(UTFPR)

Visto da Coordenação

Prof. Dr. Walter Godoy Júnior
Coordenador do Curso

À minha querida mãe Raquiel pela paciência e dedicação
comigo, especialmente em momentos difíceis da vida,
onde mais se precisa de apoio.

“Deus sussurra em nossos ouvidos por meio de nosso prazer, fala-nos mediante nossa consciência, mas clama em alta voz por intermédio de nossa dor; este é seu megafone para despertar o homem surdo.”

Clive Staples Lewis

RESUMO

VAZ, Eddy Sandro Godoy. Estudo Comparativo Entre os Subsistemas de Segurança GNU/Linux: SELinux e AppArmor. 2013. 42 f. Monografia (Especialização em Teleinformática e Redes de Computadores) – Programa de Pós-Graduação, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

Nos últimos anos a utilização de sistemas operacionais baseados em Linux vem crescendo, não somente no mundo corporativo mas também na computação pessoal, reflexo da disseminação da Tecnologia da Informação e Comunicação (TIC) no cotidiano da sociedade. Diante deste panorama esta pesquisa apresenta uma abordagem teórico-conceitual sobre modelos de segurança adicionais para os sistemas operacionais GNU/Linux, onde são abordados os subsistemas SELinux e AppArmor, que podem ser implementados tanto em servidores como em computadores pessoais. Também são abordados nesta pesquisa conceitos gerais sobre Segurança da Informação.

Palavras-chave: Linux. Tecnologia da Informação e Comunicação. Segurança da Informação.

RESUMEN

VAZ, Eddy Sandro Godoy. Estudio Comparativo Entre los Subsistemas de Seguridad GNU/Linux: SELinux y AppArmor. 2013. 42 f. Monografia (Especialização em Teleinformática e Redes de Computadores) – Programa de Pós-Graduação, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

En los últimos años el uso de sistemas operativos basados en Linux presenta crecimiento, no sólo en el mundo corporativo pero también en la computación personal, reflejo de la expansión de la Tecnología de la Información y Comunicación (TIC) en el cotidiano de la sociedad. En este contexto, esta investigación presenta un enfoque teórico-conceptual acerca de los modelos de seguridad adicionales para los sistemas operativos GNU/Linux, donde son abordados los subsistemas SELinux y AppArmor, que pueden ser implementados tanto en servidores como en computadoras personales. También son abordados en esta investigación los conceptos generales de Seguridad de la información.

Palabras clave: Linux. Tecnología de la Información y Comunicación. Seguridad de la Información.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE SIGLAS

AVC Access Vector Cache
DAC Discretionary Access Control
GNU GNU's Not Unix
GPL General Public License
LSM Linux Security Modules
MAC Mandatory Access Control
NSA Agência de Segurança Nacional
SCC Secure Computing Corporation
TIC Tecnologia da Informação e Comunicação

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTO

Esta monografia apresenta o estudo comparativo entre dois subsistemas de segurança do sistema operacional GNU/Linux, onde são abordados SELinux e AppArmor. Segundo Crispin Cowan (LEITNER, 2006, p. 42), Arquiteto de Segurança da empresa Novell, “o AppArmor e o SELinux tem objetivos semelhantes de melhorar a segurança no Linux, mas diferentes nos detalhes”.

Ambos são Software Livres, distribuídos sob a licença General Public License (GPL[1]), e o desenvolvimento principal liderado por grandes empresas de tecnologia: sendo a Red Hat responsável pelo SELinux e a Novell pelo AppArmor, após adquirir a empresa Immunix (LEITNER, 2006, p. 42).

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Apresentar de maneira teórico-conceitual a comparação técnica entre os subsistemas de segurança GNU/Linux: SELinux e AppArmor.

1.2.2 Objetivos Específicos

Abaixo a relação dos objetivos específicos pretendidos no estudo:

- Caracterizar a importância da Segurança da Informação;
- Descrever o esquema padrão de segurança em sistemas operacionais GNU/Linux;
- Descrever o subsistema de segurança SELinux;

1 GPL: A General Public License é a mais difundida das licenças de Software Livre, administrada pelo projeto GNU's Not Unix (GNU). Essa licença, entre outros, possibilita a liberdade de utilização, distribuição e modificação de softwares, maiores detalhes em: http://www.gnu.org/licenses/licenses.pt-br.html.

- Descrever o subsistema de segurança AppArmor;
- Realizar a comparação entre SELinux e AppArmor;
- Avaliar em quais aplicações SELinux e AppArmor melhor se enquadram.

1.3 JUSTIFICATIVA

Nos últimos anos a Tecnologia da Informação e Comunicação (TIC) vem ganhando espaço de destaque, pois na atualidade quase todas as atividades sociais, sejam elas de cunho pessoal ou profissional dependem de sistemas informatizados. Assim as pessoas e/ou corporações possuem informações sigilosas em seus computadores, desta forma medidas com a finalidade de proteger os dados se fazem necessárias. A medida inicial é limitar tanto o acesso físico como lógico aos computadores, principalmente em ambientes compartilhados ou que utilizam algum meio de comunicação público. Este cuidado pode ser tomado por meio da aplicação de mecanismos de segurança para proteção de arquivos e de outras informações armazenadas, contando com a automatização de processos e ferramentas de segurança (GUIMARÃES; LINS; OLIVEIRA, 2006, p. 11).

Diante da importância supracitada esta monografia foca segurança em sistemas operacionais GNU/Linux, apresentando dois modelos de controle de acesso: SELinux e AppArmor, que se enquadram como ferramentas de segurança mais avançadas em relação ao esquema padrão de sistemas GNU/Linux. Ambos, por exemplo, possibilitam: monitoramento de aplicações em execução, atribuição de privilégios específicos para usuários dos sistemas e bloqueio de operações potencialmente perigosas à integridade dos dados, dentre outros serviços.

Na sequência estão descritas as justificativas específicas do estudo:

- Por que foco em sistema operacional GNU/Linux?

Considerar a segurança em sistemas operacionais baseados em GNU/Linux como algo primordial se faz necessário principalmente pela grande utilização destes em servidores. A Figura 1 representa a estatística de uso em servidores Web no ano de 2012, onde a distribuição Debian detêm a liderança no setor, com o CentOS em segundo lugar, sendo o Debian usado em 29,4% dos servidores Web baseados em GNU/Linux (9,6% de todos os

servidores). Os números se baseiam em uma pesquisa realizada em 1 milhão dos *sites* mais populares do mundo (DEBIAN, 2012).

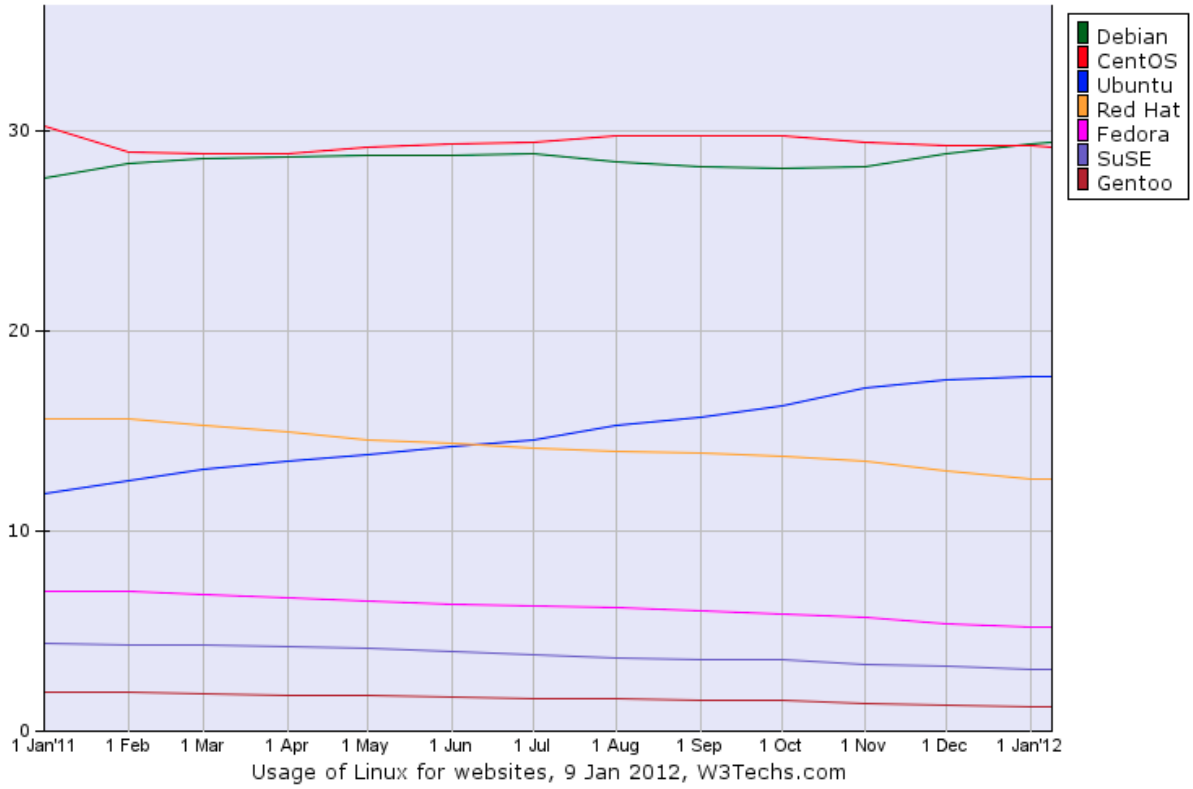

Figura 1 - Servidores Web com GNU/Linux em 2012
Fonte: Gelbmann (2012)

Quanto ao Debian seu uso inclusive cresceu de 2012 para 2013, como pode ser visualizado na Figura 2, que apresenta também uma pequena queda do uso do CentOS, que está sendo ameaçado pelo grande crescimento do Ubuntu (que trata-se de uma distribuição baseada em Debian).

	2012 1 May	2012 1 Jun	2012 1 Jul	2012 1 Aug	2012 1 Sep	2012 1 Oct	2012 1 Nov	2012 1 Dec	2013 1 Jan	2013 1 Feb	2013 1 Mar	2013 1 Apr	2013 1 May	2013 23 May
Debian	30.5%	30.7%	30.8%	31.0%	31.3%	31.7%	32.0%	32.2%	32.6%	32.9%	32.9%	32.7%	32.7%	32.6%
CentOS	28.5%	28.6%	28.7%	28.5%	28.2%	28.0%	27.9%	27.6%	27.5%	27.2%	27.1%	27.2%	27.2%	27.0%
Ubuntu	19.2%	19.6%	19.8%	20.2%	20.5%	20.8%	21.1%	21.7%	22.0%	22.4%	22.8%	23.3%	23.5%	24.2%
Red Hat	11.9%	11.7%	11.6%	11.4%	11.2%	11.0%	10.6%	10.2%	9.8%	9.5%	9.4%	9.2%	9.1%	8.9%
Fedora	4.7%	4.5%	4.3%	4.1%	4.0%	3.8%	3.7%	3.6%	3.4%	3.3%	3.2%	3.1%	3.0%	2.9%
SuSE	2.7%	2.6%	2.5%	2.5%	2.4%	2.4%	2.4%	2.3%	2.3%	2.2%	2.1%	2.0%	1.9%	1.8%
Gentoo	1.2%	1.2%	1.2%	1.3%	1.3%	1.3%	1.4%	1.4%	1.4%	1.5%	1.5%	1.6%	1.6%	1.6%
Scientific Linux	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.2%	0.1%
Turbolinux	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Mandriva	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
CloudLinux	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%		0.1%	0.1%	0.1%

The diagram shows only Linux versions with more than 1% usage.

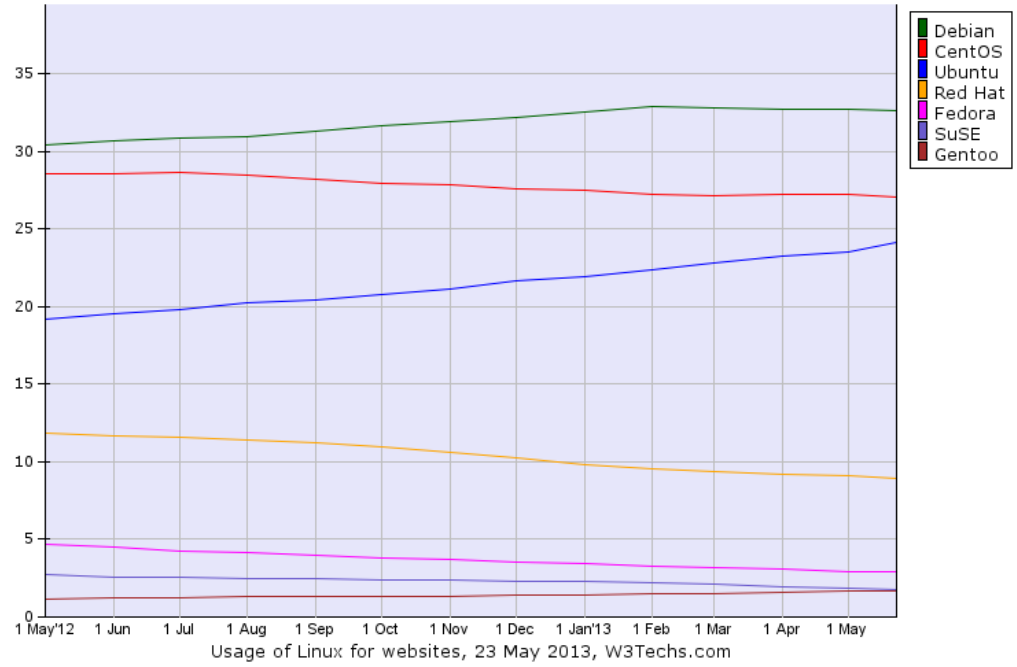

Figura 2 - Servidores Web com GNU/Linux em 2013
Fonte: Historical (2013)

- Por que controle de acesso mais avançado que o padrão presente em sistemas operacionais GNU/Linux?

No contexto de sistemas operacionais baseados em Unix (na atualidade representados, por exemplo, por OpenBSD e GNU/Linux) a questão da Segurança da Informação, ou controle de acesso, sempre foi requisito básico, até porque desde o inicio foram sistemas multiusuários (RAYMOND, 2003 apud CARVALHO, 2011, p. 8).

Portanto, o GNU/Linux herdou a maioria de seus sistemas de controle de acesso dos primeiros sistemas Unix, possuindo matrizes de permissões de arquivos, sendo três as permissões básicas: leitura, escrita e execução, que determinam como serão os privilégios dos

usuários. Este modelo é denominado controle de acesso arbitrário, ou em inglês Discretionary Access Control (DAC), sendo instalado por padrão automaticamente (GOODRICH; TAMASSIA, 2013, p. 138).

Mas o modelo DAC, que é apresentado em detalhes na sequencia do estudo, dependendo da situação pode não ser o ideal (uma empresa, por exemplo, pode apresentar especificidades ou determinadas necessidades adicionais de segurança não atendidas pelo modelo DAC) e de acordo com Goodrich e Tamassia (2013, p. 139) “algumas distribuições Linux têm mecanismos de controle de acesso ainda mais avançados”. Desta maneira o presente estudo tem como intuito contribuir com administradores de sistemas e/ou usuários comuns de sistemas baseados na plataforma GNU/Linux, apresentando dois modelos de controle de acesso adicionais: SELinux e AppArmor, onde com a descrição de cada um deles se poderá apontar qual o mais indicado de acordo com o cenário de instalação pretendido. Existem outros modelos de controle de acesso, mas a escolha de SELinux e AppArmor para esta monografia se deu principalmente pelo fato de serem utilizados respectivamente por padrão nas distribuições GNU/Linux Red Hat e SUSE e também fortemente utilizados em outras distribuições como: Debian, CentOS, Ubuntu e Fedora, que são as mais utilizadas em servidores, verificar Figura 2 com os percentuais de instalações em servidores Web no ano de 2013.

1.4 METODOLOGIA

O estudo trata-se de uma pesquisa bibliográfica exploratória, onde serão utilizadas informações literárias e de *sites* da Internet sobre os assuntos relacionados a subsistemas de segurança, buscando compreender o fenômeno da Segurança da Informação nos ambientes computacionais e identificar mecanismos que podem ser implementados para a proteção de sistemas operacionais GNU/Linux. Abaixo os procedimentos utilizados para alcance dos objetivos:

- Levantamento bibliográfico;
- Coleta das informações;
- Tratamento das informações.

Os exemplos de comandos presentes no estudo foram conseguidos do sistema

operacional GNU/Linux Debian versão estável 7 (codinome Wheezy). O processo de instalação total dos subsistemas de segurança SELinux e AppArmor depende da distribuição Linux utilizada, por esse motivo não será descrito neste estudo e também porque seria uma instalação genérica, sendo que diante da importância da segurança o ideal é uma instalação personalizada para o ambiente e/ou corporação alvo.

1.5 RESUMO DOS CAPÍTULOS

Abaixo a descrição da organização da monografia com o resumo dos respectivos capítulos:

- **Capítulo 2:** serve como fundamentação teórica para melhor assimilação do conteúdo principal específico à controle de acesso SELinux e AppArmor, aborda: informática no cotidiano da sociedade, introdução à Segurança da Informação, contextualização de Linux e descrição sobre controle de acesso;
- **Capítulo 3:** descreve o subsistema de segurança SELinux;
- **Capítulo 4:** descreve o subsistema de segurança AppArmor;
- **Capítulo 5:** compara os dois subsistemas de segurança: SELinux e AppArmor, e descreve suas indicações de utilização;
- **Capítulo 6:** apresenta as conclusões sobre o estudo dos subsistemas de segurança SELinux e AppArmor.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 INFORMÁTICA NO COTIDIANO

O surgimento das tecnologias digitais, caracterizadas principalmente pela informática, proporcionaram o advento de novas práticas sociais, de novos mecanismos de interação, informação e comunicação, onde as relações entre as pessoas não encontram mais as barreiras geográficas de outrora. Esse novo paradigma caracteriza a chamada Sociedade da Informação, que de maneira praticamente contínua e ininterrupta faz o incremento e a profusão dos meios de comunicação e intensifica o fluxo de informações (MARTÍN-BARBERO, 2003 apud ESCOSTEGUY; GUTFREIND, 2007, p. 177).

Essa Sociedade da Informação, em que vivemos na atualidade, foi vislumbrada no século passado no início da década de 1980 pelo escritor Alvin Toffler na obra A Terceira Onda, sobre essas ondas da vida Hamze (2013) descreve que “primeiro foi a revolução agrícola, depois a revolução industrial, atualmente a revolução tecnológica modifica fortemente o arcabouço do conhecimento e da realidade em que vivemos”, ainda sobre A Terceira Onda ou Sociedade da Informação Hamze (2013) reforça que “é a constituição de uma nova sociedade onde a era da informática constitui um moderno estilo de vida precipitando a absorção de informação, transformando intensamente a estrutura do conhecimento e da realidade em que vivemos”.

Evidentemente nessa nova sociedade concebida não se pode de maneira alguma deixar de lado ética e Segurança da Informação, pois são as principais preocupações dos usuários na atualidade. Essas preocupações, por exemplo, influenciam drasticamente se um cliente vai ou não adotar determinada tecnologia digital para conduzir seus negócios na Internet. Nesse contexto, ética e segurança podem interferir diretamente no lucro final de uma empresa. Não é raro encontrar notícias sobre litígio ou queda no preço de ações de empresas que tiveram divulgadas violações de privacidade por ataques ou de maneira consensual repassaram dados de clientes e/ou usuários sem autorização para outras empresas, para as mais diversas finalidades (BALTZAN; PHILLIPS, 2012, p. 91).

2.2 INTRODUÇÃO À SEGURANÇA DA INFORMAÇÃO

A Figura 3 mostra a estatística de incidentes relacionados à Segurança da Informação, referentes ao período de 1999 a junho de 2013, reportados ao Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br). Pode-se constatar no gráfico que o ano passado, 2012, registrou o maior número de incidentes reportados: 466.029. Claro que o número de incidentes tende a crescer porque o número de servidores instalados aumenta com o passar dos anos, mas em contrapartida, os números se referem somente a incidentes reportados, ou seja, certamente muitos outros ocorreram no período e não entraram na estatística do CERT.br. De qualquer forma as informações servem para ilustrar que segurança é merecedora de atenção.

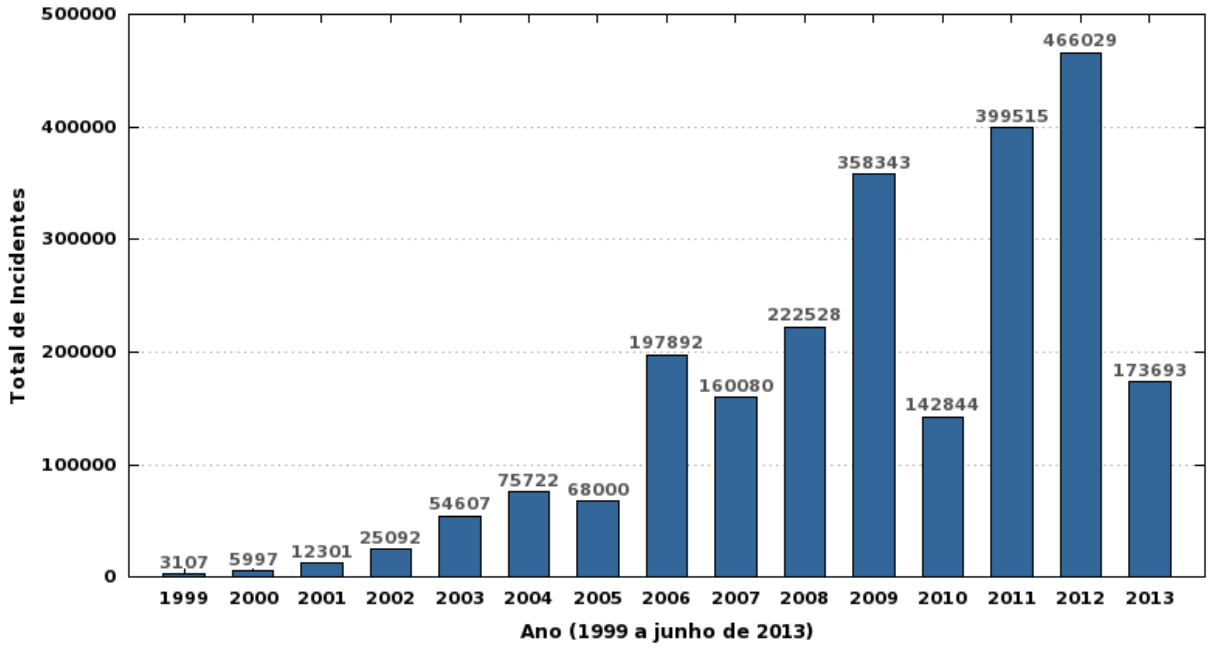

Figura 3 - Total de incidentes reportados ao CERT.br
Fonte: CERT.br (ESTATÍSTICAS, 2013)

A Segurança da Informação portanto deve ser tratada com extrema importância, embora Cheswick, Belovin e Rubin (2005, p. 26) coloquem que em princípio "a idéia de criar uma política de segurança pode sugerir burocracia para algumas pessoas, especialmente para um ávido tecnocrata", sobretudo porque com o advento da Internet não existe fronteira e todos os usuários conectados podem se tornar vítimas em potencial para uma pessoa mal intencionada.

Também para ilustrar a questão da Segurança da Informação com números, em uma

análise dos incidentes reportados entre janeiro e março de 2013 o CERT.br (INCIDENTES, 2013) destaca fatos sobre computadores comprometidos (que tendem a ficar fora de serviço para manutenção):

- No primeiro trimestre de 2013 foram recebidos mais de 5.000 notificações de máquinas comprometidas. Este total foi 146% maior do que o número de notificações recebidas no último trimestre de 2012 e 212% maior que o número de notificações recebidas no primeiro trimestre de 2012;
- A grande maioria das notificações de computadores comprometidos foi referente a servidores Web que tiveram suas páginas desfiguradas (*defacement*).

Como a cada dia as pessoas e/ou corporações possuem mais informações sigilosas em seus computadores, medidas com a finalidade de proteger os dados se fazem necessárias. A medida inicial é limitar tanto o acesso físico como lógico aos computadores, principalmente em ambientes compartilhados ou que utilizam algum meio de comunicação público. Este cuidado pode ser tomado por meio da aplicação de mecanismos de segurança para proteção de arquivos e de outras informações armazenadas, contando com a automatização de processos e ferramentas de segurança (GUIMARÃES; LINS; OLIVEIRA, 2006, p. 11).

2.2.1 Atacantes, Alvos e Motivações

De acordo com Suzuki e Delphino (2007, p. 14-15) em relação a atacantes, alvos e motivações, itens considerados em invasões de redes, os mesmos são descritos como:

- **Atacantes:** usuários mal intencionados que visão comprometer uma determinada rede, violando a política de segurança de um sistema. São conhecidos como *crackers*, ou seja, *hackers* que utilizam seus conhecimentos em informática para atos ilegais. Suas habilidades em mudar constantemente as formas de ataques dificultam as medidas de segurança;
- **Alvos:** usuários comuns (ou domésticos) normalmente tem a falsa impressão que nunca serão alvos de ataques virtuais, porém os atacantes podem utilizar determinado computador alheio para atacar outros ou armazenar informações furtadas. Em síntese todo computador que se encontra em rede é um alvo em potencial para atacantes;
- **Motivações:** as motivações para ataques a sistemas são subjetivas aos criminosos,

normalmente relacionadas, por exemplo, com: dinheiro, ego, diversão, ideologia, status e/ou inclusão em determinado grupo social.

2.3 O QUE É LINUX

Para Siever et al. (2006, p. 17) os sistemas operacionais baseados em Linux estão ganhando importante espaço na área tecnológica nos últimos anos pois "[...] o Linux passou de parque de diversões de estudantes e aficionados para um novo competidor no mercado de servidores, atingindo o estágio de sistema respeitado com lugar certo nas redes educacionais e corporativas".

O Linux foi desenvolvido inicialmente por Linus Torvalds, na Universidade de Helsinque, na Finlândia, posteriormente contou com apoio de programadores espalhados pelo mundo para lançamento da primeira versão estável. Costuma-se utilizar o termo GNU/Linux para referenciar o sistema operacional em geral, pois o Linux trata-se somente do *kernel* (ou núcleo, interface entre o hardware e o software do computador) do sistema operacional e as demais aplicações (como editores de texto, compiladores, terminais, etc) são desenvolvimentos do projeto GNU da Free Software Foundation (SIEVER et al., 2006, p. 17).

O Linux está licenciado sob GPL, significando que nem o próprio criador original Linus Torvalds pode alterá-la ou requerer o sistema novamente para ele, o que quer dizer que o Linux sempre estará disponível para todos como Software Livre, respeitando as liberdades concedidas pela licença GPL (TEIXEIRA, 2008, p. 13).

2.3.1 Distribuições

Com o crescimento da utilização do Linux empresas e pessoas passaram a desenvolver aplicações específicas para interagir com ele, consequentemente disponibilizando para outras pessoas também utilizarem, assim surgiram as chamadas distribuições GNU/Linux. Até porque no início a usabilidade do Linux era satisfatória somente para especialistas, então surgiu a necessidade de criar distribuições mais fáceis de usar para atender públicos diferenciados, inclusive de leigos em informática (TEIXEIRA, 2008, p. 18).

O Linux possui como mascote um pinguim, chamado de Tux, e para Machado, Veneu e Oliveira (2005, p. 13) “[...] há diversas distribuições do Pinguim (Mandriva, Suse, Debian, Slackware etc.) [...]”, nesta monografia foi utilizada a distribuição Debian, que segundo Teixeira (2008, p. 20) “[...] é uma das poucas distribuições que não é mantida por nenhuma empresa. É um conjunto de desenvolvedores que mantém toda a sua estrutura e representa a maior em distribuição do Linux sem ter vínculos com empresas”.

2.4 CONTROLE DE ACESSO

O controle de acesso é realizado por um módulo do sistema de gerenciamento de arquivos presente em qualquer sistema operacional, onde cada nível (representados na Figura 4) do sistema é implementado com técnicas de programação para estabelecerem uma hierarquia. Nessa hierarquia os módulos de nível mais alto passam informações para os de nível inferior para que estes possam desenvolver suas tarefas específicas e assim dar continuidade ao fluxo. Na etapa final acontece a comunicação com o gerenciador de dispositivos e com o dispositivo propriamente dito, que por exemplo pode ser um disco rígido, somente após isso que determinada ação é permitida ou negada ao usuário (FLYNN; MCHOES, 2002, p.189).

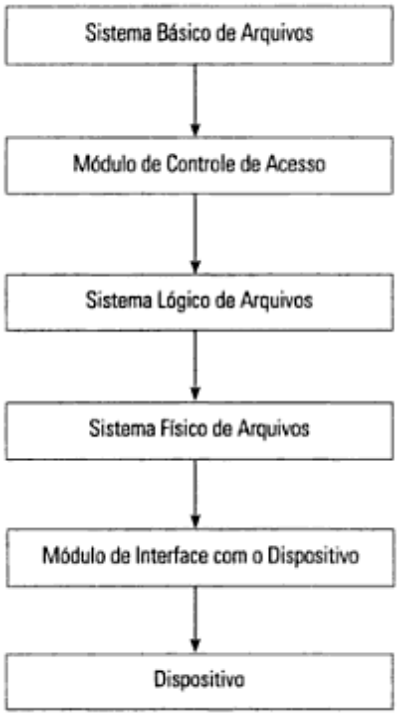

Figura 4 - Níveis do gerenciamento de arquivos
Fonte: Flynn e Mchoes (2002, p. 189)

Para descrever o funcionamento do módulo de controle de controle de acesso, segue exemplo: solicitada a abertura de um arquivo de texto por um aplicativo, o sistema pesquisa no diretório raiz do mesmo sua existência e todas as informações relacionadas a ele são incorporadas à tabela de arquivos ativos mantida pelo sistema operacional. Essas informações incluem: tamanho do registro, endereço do primeiro registro físico, informações de proteção e as informações de controle de acesso. De acordo com estas informações o sistema básico de arquivos do sistema operacional ativa o módulo de controle de acesso para verificar se o usuário possui permissão para executar a operação solicitada. Em caso afirmativo, o arquivo é apresentado na interface do aplicativo e em caso negativo é retornada a mensagem de "acesso negado" ao usuário (FLYNN; MCHOES, 2002, p.190).

Para Goodrich e Tamassia (2013, p. 137) "assim que um usuário for autenticado em um sistema, a próxima questão dever ser dirigida ao controle de acesso: como o sistema operacional determina o que os usuários têm permissão para fazer?". Para responder a questão, se referindo especificamente à sistemas baseados em Linux, Goodrich e Tamassia (2013, p. 138) colocam que: "o Linux possui matrizes de permissões de arquivos, que determinam os privilégios que diversos usuários possuem em relação a arquivos".

O controle de acesso representa aos sistemas, e também a redes de computadores, um item de importância para garantir as características fundamentais de segurança que são: confiabilidade (um dado/sistema não será acessado por usuário que não possua autorização), integridade (prevenir contra a modificação de dados por usuários não autorizados) e disponibilidade (garantir que um usuário, que seja autorizado, terá acesso aos dados/sistemas de maneira ininterrupta) (MADEIRA, 2013).

2.4.1 Modelos de Controle de Acesso

Os modelos de controle de acesso mais utilizados são os denominados como arbitrário e obrigatório. O controle de acesso arbitrário ou DAC trata-se de um esquema onde os usuários possuem a capacidade de alterar as permissões de acesso a seus próprios arquivos e aos dos outros, bem como a recursos do sistema, este é o padrão em sistemas Linux e será tratado com maiores detalhes na subseção 2.4.2. Em contrapartida, o controle de acesso obrigatório ou Mandatory Access Control (MAC) se refere a um esquema mais rigoroso e restritivo para os usuários (GOODRICH; TAMASSIA, 2013, p. 439).

No modelo MAC os usuários não podem alterar permissões de arquivos ou recursos do sistema, qualquer que seja o proprietário. As decisões de segurança são tomadas por um usuário administrador central de política. Onde cada regra de segurança consiste em um **sujeito**, que representa quem tenta obter acesso a determinado arquivo ou recurso; um **objeto**, que se refere ao arquivo ou recurso sendo acessado; e uma série de permissões que definem a extensão em que esse arquivo ou recurso pode ser acessado (GOODRICH; TAMASSIA, 2013, p. 439).

O modelo MAC em síntese apresenta maiores benefícios em relação à segurança que o tradicional DAC. Abaixo, como exemplos, duas potenciais situações de risco de segurança que são encontradas no modelo DAC:

- Existem apenas dois tipos de usuários: administrador e não-administrador. Sendo que para alguns serviços serem executados de maneira eficiente ou eficaz a escolha é resumida em atribuir permissão de administrador ao processo, fato que pode colocar em risco todo o sistema operacional;
- O usuário pode expor todos os seus arquivos a brechas de segurança com atribuições de permissões equivocas.

As duas situações de risco citadas acima são resolvidas no modelo MAC (entre outras) pois a política de acesso é determinada pelo sistema e não pelos proprietários dos recursos. Sendo este modelo utilizado em sistemas com dados altamente sensíveis, como por exemplo, em organizações governamentais e militares. Nesse tipo de controle existe a construção de um sistema que manipula múltiplos níveis de classificação entre sujeitos (nível de privilégios) e objetos (nível de sensibilidade da informação). No MAC, há uma divisão de tarefas entre: os administradores dos sistemas, que definem os níveis de privilégio dos usuários e a política de segurança; e os gestores das informações que estabelecem a rotulação das informações quanto ao seu nível de sensibilidade. O sistema cuida de aplicar as regras da política com base nos privilégios dos usuários e no rótulo das informações (ARANTES FILHO, 2013).

Uma política de segurança possibilita aos administradores ou gestores estabelecerem restrições sobre quais ações os sujeitos em um sistema podem fazer a respeito de objetos desse sistema, com a finalidade de atingir objetivos específicos de segurança (GOODRICH; TAMASSIA, 2013, p. 438). De acordo com Goodrich e Tamassia (2013, p. 438) uma política de segurança contempla os seguintes componentes:

- **Sujeitos:** os agentes que interagem com o sistema;

- **Objetos:** os recursos de informação e computacionais que uma política de segurança deve proteger e administrar;
- **Ações:** as coisas que os sujeitos podem ou não fazer com relação aos objetos;
- **Permissões:** mapeamento entre sujeitos, ações e objetos, que estabelecem claramente que tipos de ações são permitidas ou proibidas;
- **Proteções:** as características específicas de segurança ou regras que são incluídas na politica para ajudar a atingir determinados objetivos de segurança.

Portanto, como o MAC atende aos componentes requeridos pode ser utilizado para implementar políticas de segurança adequadas para atender aos requisitos estabelecidos pelos administradores ou gestores das organizações.

Tanto SELinux como AppArmor são implementações de segurança MAC para sistemas baseados em Linux (ALSBIH, 2005 apud SOUZA, 2007, p. 20). O que possibilita que esses subsistemas de segurança possam ser instalados para funcionar com o *kernel* do Linux, é o Linux Security Modules (LSM). O LSM trata-se de uma arquitetura que possibilita a instalação de subsistemas de segurança na forma de módulos, impossibilitando desta forma o favoritismo de uma implementação sobre outra, deixando a escolha de qual instalar para o administrador do sistema. Os módulos de segurança atualmente suportados de maneira oficial, além de SELinux e AppArmor, são: Smack e Tomoyo (IVASHKO, 2013).

2.4.2 Controle de Acesso Padrão em Sistemas Linux

O GNU/Linux herdou a maioria de seus sistemas de controle de acesso dos primeiros sistemas Unix, o modelo padrão de segurança é o denominado como controle de acesso arbitrário, ou DAC, sendo instalado automaticamente em qualquer distribuição (GOODRICH; TAMASSIA, 2013, p. 138).

Este controle de acesso padrão, ou sistema de permissões como também pode ser chamado, possibilita que vários usuários usem simultaneamente o sistema (pois o GNU/Linux é um sistema operacional multiusuário) sem que um atrapalhe o trabalho do outro, nem possa visualizar ou alterar arquivos que não deveria. O controle de acesso consiste em um conjunto de três permissões (leitura, escrita e execução) e três grupos (dono, grupo e outros), que combinadas permitem fazer muitas configurações de segurança. Além disso o sistema é

utilizado com uma conta de usuário comum, com poucos privilégios de configuração em relação ao sistema operacional como um todo, e os programas são projetados para funcionar desta forma. Apenas os utilitários de configuração e alguns programas para tarefas específicas necessitam ser executados como usuário *root*, que trata-se do usuário administrador e desta forma tem total privilégios sobre o sistema operacional (MORIMOTO, 2006).

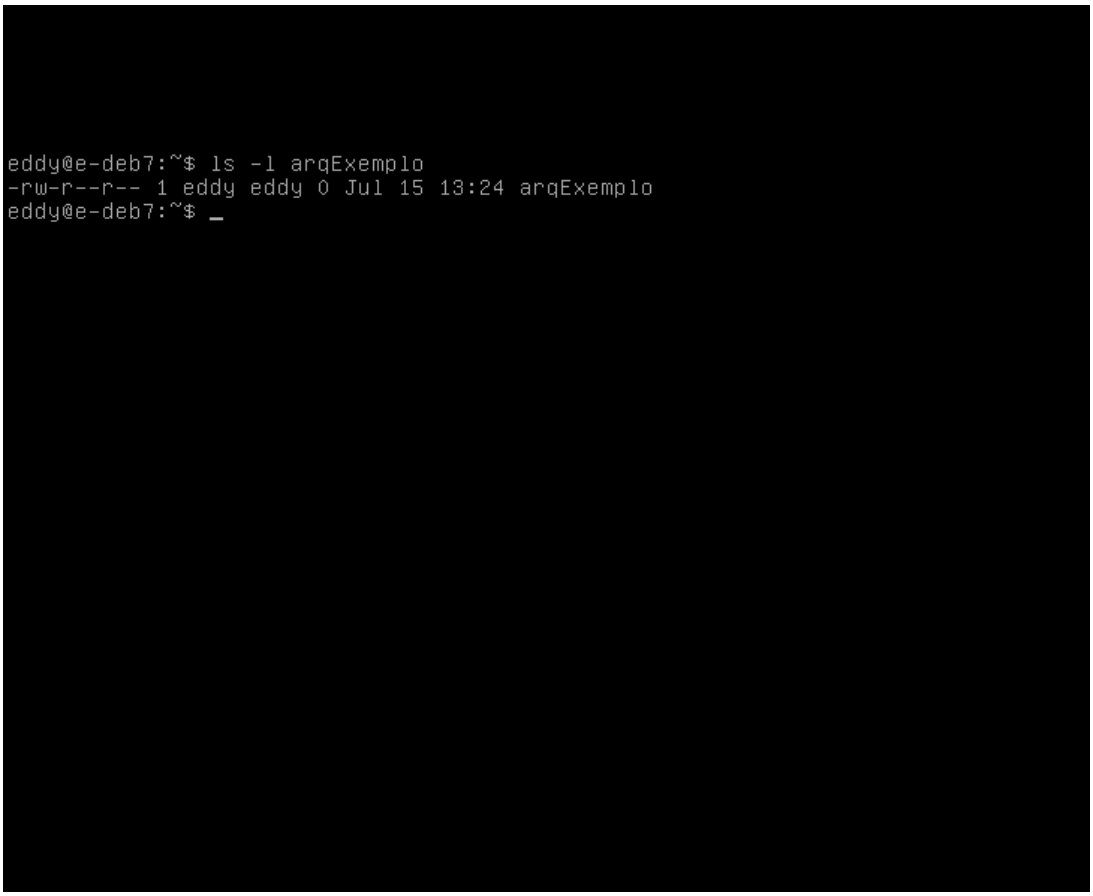

Figura 5 - Exemplo de permissões a arquivo

Como exemplo, na Figura 5 as permissões do arquivo "arqExemplo" estão listadas pelo comando "ls -l" e representadas na primeira coluna como: **-rw-r--r--**, que representam, da esquerda para direita com hifens e letras inicias de palavras em inglês, as permissões de leitura (**r** de **r**ead), escrita (**w** de **w**rite) e execução (**x** de e**x**ecute) para o dono, grupo e outros, sendo que de acordo com Morimoto (2006):

- **-** : primeiro hífen, representa que se trata de um arquivo e não um diretório;
- **rw-** : permissões para dono do arquivo, que pode ler e escrever no arquivo;
- **r--** : permissões para grupo de usuários, que podem somente ler o arquivo;
- **r--** : permissões para outros, ou seja, usuário que não é dono do arquivo e não faz parte

do mesmo grupo do dono, estes outros usuários no exemplo podem somente ler o arquivo também.

O hífen nas configurações de controle de acesso para o dono, grupo e outros representa negação de permissão, no exemplo se pode verificar que somente o dono tem permissão de escrita e ninguém tem permissões de execução (que seria **x** na terceira posição) porque não se faz necessário pois o "arqExemplo" trata-se de um arquivo de texto comum, se aplicaria a questão de execução se o arquivo fosse um programa (aplicação) ou um diretório, para possibilitar ao usuário abrir e listar o conteúdo. Conclui-se que o GNU/Linux trata todos os elementos que o constitui como sendo arquivos ou diretórios, que são sujeitos as aplicações de configurações de acesso.

Esta questão da permissão de execução embora pareça simples é muito importante de maneira geral para a segurança do GNU/Linux, Morimoto (2006) coloca que "[...] o sistema não decide quais arquivos são programas pela extensão, mas sim pelas permissões. Isso aumenta bastante a segurança do sistema [...]".

3 SELINUX

3.1 O QUE É?

O SELinux é uma tecnologia para proteger sistemas informáticos, bem como redes de computadores. Esta tecnologia representa muitos anos de pesquisas de segurança para sistemas operacionais baseados no Linux, tendo como foco um mecanismo de controle de acesso obrigatório poderoso e flexível, incorporado ao *kernel* do sistema operacional, podendo efetivamente reduzir os problemas causados por falhas de softwares, incluindo as falhas ainda não descobertas ou exploradas por pessoas ou organizações mal intencionadas. (MAYER; MACMILLAN; CAPLAN, 2007).

A tecnologia SELinux adota um princípio chamado de menor privilégio, que consiste em limitar cada processo do sistema operacional às permissões básicas mínimas necessárias para o seu funcionamento adequado, esse princípio acaba por minimizar os efeitos de uma falha de segurança que esteja presente em alguma aplicação. Alem disso, ao contrário do controle de acesso DAC, os usuários não possuem permissão para mudar atributos de segurança de seus próprios arquivos, essa permissão somente é delegada a um administrador de política central de segurança. Essas implementações possibilitam ao SELinux estabelecer um ambiente de segurança muito mais restritivo (GOODRICH; TAMASSIA, 2013, p. 139).

Nos sistemas operacionais com o SELinux todos os objetos, sendo eles arquivos ou processos, recebem uma espécie de marca chamada de rótulo de segurança para um nível extra de controle. Esse rótulo, conhecido também como contexto de segurança, geralmente compreende três componentes: usuário, papel e tipo/domínio, que respectivamente representam o sujeito, de que objeto se trata e privilégios atribuídos (SCHERF, 2006, p. 67).

3.2 HISTÓRIA

Em meados da década de 1990, entre os anos de 1992 a 1993, pesquisadores da Agência de Segurança Nacional (NSA) do Estados Unidos da América e da Secure Computing Corporation (SCC) estavam trabalhando na criação de um novo sistema operacional denominado como DTMach, com recursos obtidos de outros dois projetos: o

TMach e o LOCK, além de um mecanismo de controle de acesso do projeto Mach, entre outros recursos deste. Tempos depois, o projeto DTMach foi desenvolvido como parte do projeto DTOS. Após a conclusão do projeto DTOS, os esforços da NSA, SCC e também da Universidade de Utah foram combinados para integrar a arquitetura do sistema de segurança DTOS ao sistema operacional de pesquisa Fluke. O novo projeto foi denominado Flux. Paralelamente, a arquitetura original do sistema de segurança foi aprimorada. A nova arquitetura aprimorada foi denominada Flask. Na etapa seguinte, a NSA implementou essa arquitetura de sistema de segurança no sistema operacional GNU/Linux, valendo-se da arquitetura LSM. Isso foi colocado à disposição do público com o nome de SELinux (IVASHKO, 2013).

O SELinux foi lançado inicialmente como um produto de software de acesso geral (com o código fonte distribuído sob uma licença GPL) em dezembro de 2000, na sequência foi incorporado na distribuição corporativa Red Hat Enterprise Linux 4, depois passou a ser usado também nos sistemas operacionais Debian e Fedora, com isso rapidamente se tornou o controle de acesso adicional mais utilizado em sistemas Linux (IVASHKO, 2013).

3.3 FUNCIONAMENTO

Cardozo (2013) destaca que é “importante deixar claro que o SELinux não substitui o tradicional mecanismo de controle de acesso DAC do Linux. Ele simplesmente complementa-o, verificando se uma determinada operação é permitida após as permissões padrões do usuário já terem sido checadas”.

Para Daniel Riek (LEITNER, 2006, p. 44), Gerente de Produtos da Red Hat, o SELinux procura dar segurança para todo o sistema operacional pois “o SELinux aplica controles de acesso estritos baseados em MAC no nível do *kernel* ”.

No SELinux, identificação é de suma importância. Todos os processos e recursos computacionais são identificados com um determinado tipo que representa seu contexto de segurança, sendo o acesso somente concedido se uma regra aceita uma correspondência entre o tipo do que acessa com o que é acessado. Este é o princípio básico por trás do seu funcionamento (SELINUX, 2013).

Para Scherf (2008, p. 67) “a política é outro componente importante. Uma política SE

define o acesso entre objetos e agentes. A política especifica quais objetos o processo (como um processo do httpd com um papel específico) tem permissão de acessar".

De acordo com Ivashko (2013) o SELinux funciona da seguinte maneira:

1. O sujeito do sistema operacional (processo) tenta realizar uma determinada ação em um objeto específico (por exemplo um arquivo ou outro processo), que é permitida dentro do sistema de segurança padrão DAC. Isso ativa um fluxo de solicitações (*Events flow*) para o objeto;
2. O LSM intercepta todas as solicitações para realizar a ação com o objeto e as transfere, junto com o contexto de segurança do sujeito e do objeto, para o subsistema SELinux abstraction and hook logic, responsável pela interação com o LSM;
3. As informações recebidas do subsistema abstraction and hook logic do SELinux são encaminhadas para o módulo básico Policy enforcement server, que é diretamente responsável por tomar a decisão de permitir o acesso do sujeito ao objeto;
4. Para receber a decisão sobre a permissão ou proibição da ação, o servidor de cumprimento de políticas (SELinux) entra em contato com o subsistema especial Access vector cache (AVC), que muitas vezes armazena em cache as regras que estão sendo usadas para otimizar o desempenho;
5. Se o AVC não contém a decisão em cache referente à política em questão, a solicitação da política de segurança necessária é encaminhada para o banco de dados de políticas de segurança (System security policy database);
6. Quando a política de segurança é localizada, ela é transferida para o servidor de políticas que recebe a decisão;
7. Se a ação solicitada cumpre a política localizada, a operação é permitida. Caso contrário, ela é proibida, e todas as informações de tomada de decisão são gravadas no arquivo de *log* (*Log file*) do SELinux.

A Figura 6, representa o esquema de funcionamento do SELinux:

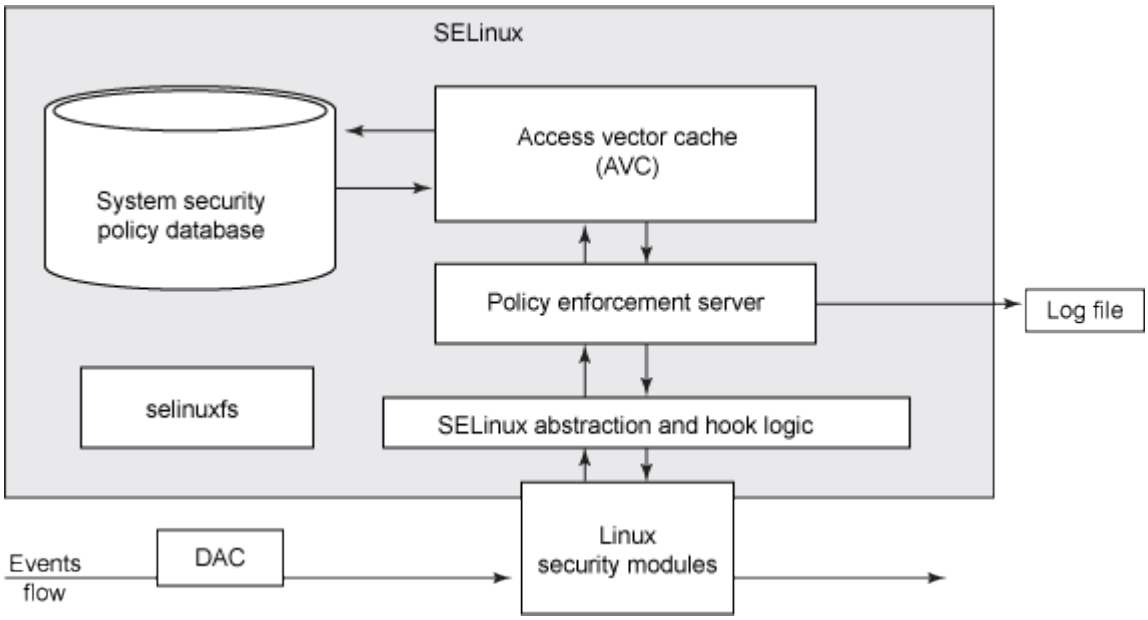

Figura 6 - Visão geral do funcionamento do SELinux
Fonte: Ivashko (2013)

Uma política SE corresponde a um módulo que é carregado pelo sistema SELinux, sendo composta por três arquivos, que são de acordo com a extensão:**.fc**, **.te** e **.if**, onde: o arquivo **fc** contém os contextos de arquivos que definem como o SELinux rotulará arquivos individuais; o arquivo **te** contém as regras de reforço de tipos; e o arquivo **if** contém a definição de interface de módulo e sua respectiva documentação. O administrador pode escrever este três arquivos do zero com um editor de texto comum ou utilizar uma ferramenta chamada de **policygentool** para automatizar o processo (SPENNEBERG, 2006a, p. 40).

Segue exemplo de criação de política SE para o executável foo (fictício): administrador executa a ferramenta **policygentool**, informando respectivamente o nome do novo módulo de política a ser criado e o caminho completo da aplicação que se deseja proteger, ficando o comando: **"policygentool foo /usr/bin/foo"**. A ferramenta então irá solicitar vários detalhes sobre a aplicação, tais como se ele usa um *script* de inicialização, onde armazena seus arquivos de *logs*, etc. Depois de respondidas as questões serão criados os três arquivos: foo.fc, foo.te e foo.if, respectivamente: contexto do arquivo, reforço de tipo e de interface. O arquivo de contexto do arquivo permite que o administrador vincule os arquivos do aplicativo a um rótulo do SELinux, enquanto que o arquivo de reforço de tipo especifica as regras que coincidem com ele, ou seja, o que o aplicativo tem permissão de fazer e o arquivo de interface põe macros à disposição de outros módulos de política. Foi apresentado somente o processo de criação de uma política SE, para que esta política pudesse ser utilizada efetivamente pelo SELinux antes os três arquivos (foo.fc, foo.te e foo.if) passariam por um

processo de compilação, que gera um arquivo com extensão **.pp**, no exemplo resultaria o foo.pp. Mas é possível utilizar a interface gráfica system-config-selinux para automatizar, e consequentemente facilitar, todo o processo de criação de um novo módulo (SCHERF, 2006, p. 73).

3.4 PRINCIPAIS COMANDOS

Abaixo são relacionados os principais comandos de terminal para administração do SELinux:

- **sestatus:** utilizado para verificar o estado geral do subsistema de segurança;
- **policygentool:** permite criar políticas de segurança adicionais para aplicações específicas;
- **seinfo:** relaciona informações sobre a política de segurança usada;
- **semanage:** permite configurar elementos da política de segurança;
- **getsebool:** permite visualizar as variáveis booleanas;
- **setsebool:** permite alterar o valor das variáveis booleanas;
- **chcon:** permite alterar o contexto de segurança dos arquivos;
- **restorecon:** permite restaurar as definições de segurança iniciais dos arquivos.

4 APPARMOR

4.1 O QUE É?

Para Crispin Cowan (LEITNER, 2006, p. 42), Arquiteto de Segurança da empresa Novell, “o AppArmor assegura aplicativos individuais contra defeitos latentes, e protege um sistema inteiro contra uma ameaça em particular, tal como ataques pela rede, protegendo todos os aplicativos que usam a rede”.

Morimoto (2013) coloca que “o AppArmor, usado por padrão no OpenSUSE é o principal 'concorrente' do SELinux, que é usado do Fedora, CentOS e em algumas outras distribuições. Ambos desempenham a mesma função: monitorar os aplicativos em execução [...]”. Apesar dessa semelhança genérica, pois se tratam de sistemas de controle de acesso MAC para sistemas baseados em Linux, Morimoto destaca que “o SELinux e o AppArmor trabalham de forma bastante diferente, seguindo conceitos quase que opostos em termos de usabilidade”.

Usado por padrão nas distribuições GNU/Linux SUSE e openSUSE, ambas desenvolvidas pela empresa Novell, o AppArmor é uma ferramenta de segurança de aplicativos projetado para proporcionar facilidade de utilização em sua estrutura. O AppArmor protege proativamente o sistema operacional e aplicativos de ameaças externas ou internas, prevenindo que falhas de aplicações ainda desconhecidas sejam exploradas. Políticas de segurança, chamadas de “perfis”, definem completamente que recursos do sistema as aplicações individuais podem acessar e com quais privilégios. Uma série de perfis padrão são fornecidos juntamente com a instalação do AppArmor e usando uma combinação de análise estática e ferramentas com interface gráfica baseadas na aprendizagem, mesmo as aplicações muito complexas podem ser configuradas com políticas de segurança adequadas com sucesso em questão de horas (APPARMOR, 2013).

4.2 HISTÓRIA

Nasceu em meados do ano de 1998 como um projeto de complemento de segurança para os sistemas baseados em Linux desenvolvidos pela empresa Immunix, que

posteriormente foi adquirida pela Novell que mantém as distribuições GNU/Linux SUSE e OpenSUSE. O AppArmor foi incorporado ao *kernel* oficial, bem como as distribuições Linux da Novell, a partir da versão 2.6.36. Atualmente a Canonical, empresa mantenedora da distribuição Ubuntu utiliza por padrão o controle de acesso e também contribui com o projeto (SANTOS, 2013).

Logo após se tornar responsável pelo AppArmor, pela aquisição da Immunix em 2005, a Novell desistiu do SELinux e começou a promover o seu novo subsistema de controle de acesso colocando-o no mercado como uma alternativa mais simples (na visão da empresa) de segurança para Linux. A empresa tomou essa decisão em vez de investir numa colaboração dentro da comunidade de código aberto para o desenvolvimento do SELinux, comunidade que na época incluía corporações do ramo tecnológico como: Red Hat, IBM e HP, entre outras (LEITNER, 2006, p. 44).

4.3 FUNCIONAMENTO

O AppArmor foi desenvolvido com o foco em ajudar os administradores a montar um esquema de segurança adicional. O Sistema monitora a forma como os processos acessam os arquivos, distinguindo entre acesso de leitura e de escrita, assim como o uso do privilégio do usuário administrador *root*. A ideia de controlar o acesso e as ações baseando-se no processo em vez do proprietário e/ou usuário do recurso não é novidade, por exemplo, o sistema Systrace (desenvolvido por Niels Provos e disponível para Linux e FreeBSD) já contava com esse princípio. Entretanto, o Systrace monitora as chamadas ao sistema pelos processos e o AppArmor vale-se do LSM, interagindo diretamente com o *kernel* (SPENNEBERG, 2006b, p. 17).

Segundo Venezuela (2013) “Através do 'perfil' de cada programa, o AppArmor pode limitar o que um programa pode fazer e quais arquivos ele pode acessar, gravar ou executar”. Um perfil do AppArmor corresponde a um arquivo de texto simples contendo entradas de caminhos para aplicações e respectivas permissões de acesso (NOVELL, 2013).

Os perfis são arquivos de texto com as configurações de políticas de segurança e se localizam, por padrão, no diretório **/etc/apparmor.d/** do GNU/Linux. Por exemplo, o perfil de segurança para o programa Firefox (navegador para Internet), ficaria em

/etc/apparmor.d/usr.bin.firefox, onde o nome do perfil trata-se do caminho completo para o executável do programa (**/usr/bin/firefox**), substituindo-se cada / por um ponto. No exemplo trata-se do caminho para o atalho do programa principal do Firefox, mas não há problema porque de qualquer forma é o primeiro a ser executado para iniciá-lo. O perfil é associado ao programa no momento em que este é executado. Desta maneira, para aplicar alguma restrição em um programa em execução é necessário reiniciar o mesmo para que entre em vigor (HESS, 2013).

Portanto, para todos os programas que se deseja proteger é necessário criar um perfil associado, pois de acordo com Morimoto (2013) "o AppArmor protege por padrão apenas alguns utilitários básicos de rede, como o ping e o traceroute [...]". Como não se trata de um módulo de política de segurança compilado, como no SELinux, tendo conhecimento da sintaxe da linguagem utilizada pode-se editar os perfis existentes, como coloca Hess (2013): "[...] é possível alterar as permissões de acesso de um programa simplesmente editando o arquivo de perfil e alterando as permissões do perfil sobre arquivos [...]".

Como os demais sistemas de controle de acesso MAC, o AppArmor também possui dois modos de funcionamento: o *enforcement* (modo ativo de segurança do sistema) e o *complain* (modo de reclamação, equivalente ao permissivo do SELinux: relata violação das políticas em arquivos de *logs*, mas não impede nenhuma ação solicita) (HESS, 2013).

A Figura 7 apresenta o esquema de funcionamento do AppArmor, onde ele por meio de perfis (*Application Profiles*) interage com o *kernel* Linux (no exemplo a versão 2.6) por meio do LSM para estabelecer as políticas de segurança para as aplicações, disponibilizando a interface gráfica de administração Yast (presente por padrão no OpenSUSE e SUSE da Novell), e gerando relatórios e alertas, que podem ser visualizados tanto no próprio Yast ou em modo texto no terminal do GNU/Linux.

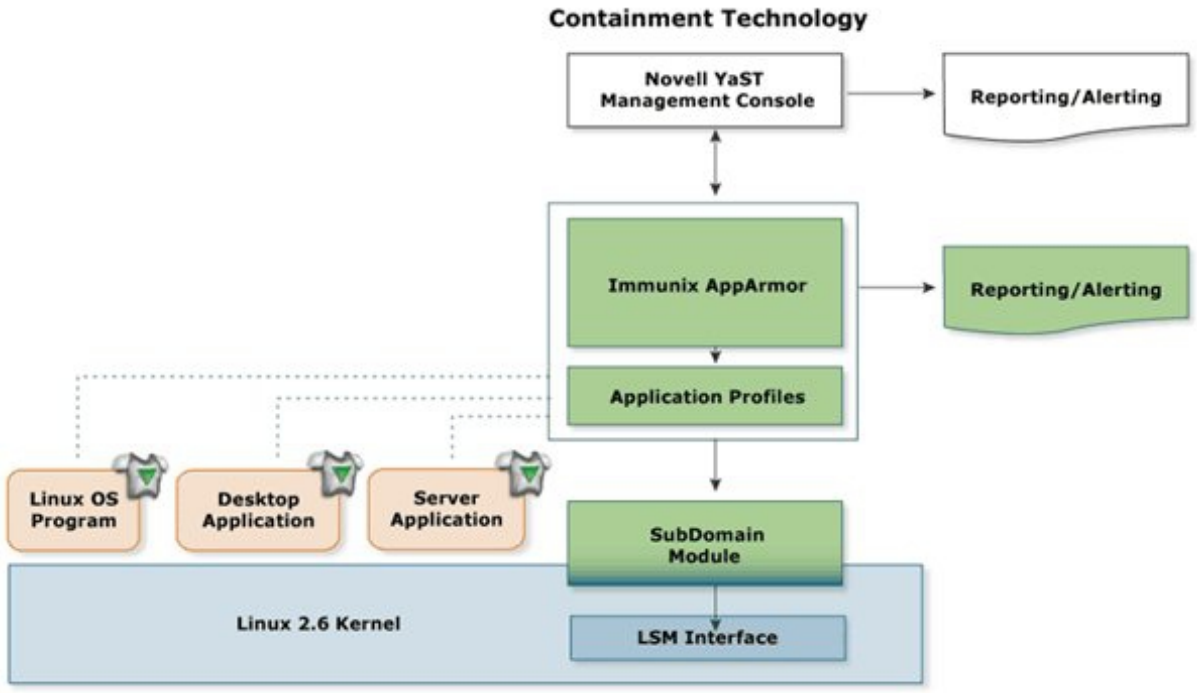

Figura 7 - Visão geral do funcionamento do AppArmor
Fonte: Baader (2013)

4.4 PRINCIPAIS COMANDOS

Abaixo são relacionados os principais comandos de terminal para administração do AppArmor:

- **apparmor_status:** para visualizar o status e o modo de cada um dos perfis configurados;
- **autodep:** realiza suposições sobre os requisitos básicos de configuração para um perfil, cria assim um rascunho de perfil para o programa ou aplicação examinada por não conter todas as entradas necessárias para funcionamento;
- **genprof:** gera ou atualiza um perfil. Se o perfil especificado não existir, o genprof cria usando o autodep ;
- **logprof:** administra os perfis, trata-se de uma ferramenta interativa usada para revisar os *logs* gerados nos dois modos de funcionamento (*enforce* e *complain*) do AppArmor;
- **aa-enforce:** para passar um perfil para o modo *enforce;*
- **aa-complain:** para passar um perfil para o modo *complain*.

5 SELINUX *VERSUS* APPARMOR

Como mencionado anteriormente, AppArmor e SELinux são semelhantes quanto ao objetivo geral de melhorar a segurança em sistemas baseados em Linux, mas que são diferentes nos detalhes. Abaixo são apresentados seus principais itens para comparação:

-Segurança:

O SELinux pode ser considerado um sistema confiável de segurança, pois foi desenvolvido pela Agência de Segurança Nacional do Estados Unidos da América. Além disso, o SELinux é fruto de vários anos de desenvolvimento científico na área de segurança de sistemas. Esse desenvolvimento teve uma base teórica profunda e provou ser altamente efetivo na prática, sobretudo em sistemas militares especializados (IVASHKO, 2013).

O AppArmor oferece segurança para "trechos" de um processo, algo que o SELinux só adicionou recentemente, ele traz também um módulo para o servidor Web Apache usar essa propriedade, assim o administrador pode criar perfis do AppArmor para tarefas pequenas como um *script* em linguagem de programação Perl executado pelo mod_perl, ou até uma página PHP individual. Essa tecnologia de conseguir confinar páginas PHP individuais é desconhecida em outras implementações de segurança (LEITNER, 2006, p. 43).

Santos (2013) destaca que o AppArmor cumpre seu papel de segurança mas que "[...] há discussões sobre sua eficácia uma vez que é possível desabilitá-lo sem necessitar reiniciar o sistema, portanto se um usuário mal intencionado obter acesso administrativo sem comprometer um binário monitorado por ele, acaba se tornando inútil", isso caracteriza o AppArmor como um serviço (*daemon*), ao contrário do SELinux que de acordo com Daniel Riek (LEITNER, 2006, p. 44), Gerente de Produtos da Red Hat, não é um serviço na concepção tradicional pois "o SELinux aplica controles de acesso estritos baseados em MAC no nível do *kernel* ", assim para desabilitar completamente é necessário reiniciar a máquina onde esteja ativo para que a nova configuração entre em vigor.

-Usabilidade:

Crispin Cowan (LEITNER, 2006, p. 42), Arquiteto de Segurança da empresa Novell, à respeito do SELinux coloca que ele "[…] tenta controlar o sistema inteiro, incluindo a segurança de propriedades como o fluxo de informação, e por isso paga o preço da complexidade", Crispin Cowan (LEITNER, 2006, p. 43) acrescenta que "[...] o SELinux parece ter sido feito para atender aos desejos da NSA por políticas arbitrariamente complexas

às custas da usabilidade".

Daniel Riek (LEITNER, 2006, p. 44), Gerente de Produtos da Red Hat, concorda com as alegações de que o SELinux apresenta problemas de usabilidade: "é claro que o AppArmor é mais fácil de configurar, pois ataca um grupo bem menor de problemas de segurança".

Para Ivashko (2013) "[...] uma boa política de segurança do SELinux para o sistema inteiro contém, em média, mais de 100 mil regras! Consequentemente a sua criação, o refinamento e a manutenção, na sua forma atual, requer muito tempo e esforço".

De acordo com Morimoto (2013) SELinux e AppArmor seguem conceitos opostos em relação a usabilidade, coloca que "o SELinux foi desenvolvido com o intuito de ser usado em servidores, priorizando a segurança e não a facilidade de uso (já que a idéia é que o sistema seja configurado por um administrador experiente)", Morimoto (2013) acrescenta que "o AppArmor também é bastante seguro, mas foi desenvolvido com o propósito de ser mais intuitivo e simples de usar, o que nos leva os módulos do Yast [...]". O Yast é uma interface gráfica para instalação e administração presente nas distribuições OpenSUSE e SUSE da Novell, por meio de módulos possibilita a configuração de políticas de segurança para o AppArmor.

Entretanto, segundo Scherf (2008, p. 70) o SELinux também dispõe de interface gráfica para configuração: "a ferramenta mais interessante para o SELinux é o system-config-selinux. Ele permite que os administradores façam configurações básicas, como o modo do SELinux, enquanto suporta tarefas mais complexas como a criação de novos módulos de política".

Em relação a criação de novos módulos de políticas de segurança, no SELinux chamadas de políticas SE e no AppArmor de perfis, o AppArmor apresenta a facilidade de criação e edição pois não existe compilação para o módulo, por outro lado esta característica de usabilidade pode, por exemplo, representar risco de segurança pois uma pessoa mal intencionada pode editar rapidamente uma regra de um perfil e isso passar despercebido pelos administradores do sistema.

-Licenciamento:

Ambos são Softwares Livres com licença GPL.

-Sistemas operacionais recomendados:

- SELinux: Red Hat, Fedora, CentOS e Debian;

- AppArmor: OpenSUSE, SUSE e Ubuntu.

-Suporte:

- SELinux: pela empresa Red Hat para o sistema operacional Red Hat Enterprise Linux;
- AppArmor: pela empresa Novell para os sistemas operacionais OpenSUSE e SUSE e pela Canonical para o Ubuntu.

Diante do pesquisado e exposto, para SELinux e AppArmor, se pode montar um quadro comparativo (Quadro 1):

Item	SELinux	AppArmor
Tipo do controle de acesso	MAC	MAC
Criador da tecnologia	Agência de Segurança Nacional (NSA) do Estados Unidos	Immunix (adquirida pela Novell em 2005)
Usabilidade	Baixa	Alta
Sistemas operacionais recomendados	Red Hat, Fedora, CentOS e Debian	OpenSUSE, SUSE e Ubuntu
Nome da política de segurança	Política SE	Perfil
Tipo do módulo de política de segurança	Compilado	Interpretado
Interface gráfica de administração	system-config-selinux	Yast
Uso recomendado em	Servidores	Computadores pessoais/Servidores
Funcionamento básico	À nível do Kernel	Como serviço (daemon)
Licenciamento	Software Livre (GPL)	Software Livre (GPL)
Suporte	Red Hat para o sistema operacional Red Hat	Novell para os sistemas operacionais OpenSUSE e SUSE; Canonical para o sistema operacional Ubuntu

Quadro 1 - Comparativo entre SELinux e AppArmor

6 CONCLUSÃO

Esta monografia abordou a importância da Segurança da Informação e apresentou os subsistemas de segurança SELinux e AppArmor, ambos representam medidas de segurança adicionais para sistemas operacionais baseados em Linux e podem ser implementados para auxiliar na diminuição da incidência de ataques digitais, ou mesmo acabar com grande parte deles. É evidente que medidas ou políticas de segurança eficazes vão além de um SELinux ou AppArmor configurados de maneira correta. No caso de uma empresa, para Cheswick, Belovin e Rubin (2005, p. 26) "uma política de segurança é um conjunto de decisões que, coletivamente, determina a postura de um organização em direção à segurança". Portanto, o foco para medidas de Segurança da Informação eficazes não deve ser somente a tecnologia empregada nas aplicações, mas a tecnologia é um dos itens mais importantes e até mesmo pode ser o diferencial.

Comparando os dois subsistemas de segurança, especialmente em dois itens: segurança e usabilidade, se chegou a conclusão que em segurança ambos de maneira geral podem representar segurança adicional (MAC) para o sistema operacional GNU/Linux, com o diferencial de que o SELinux é utilizado por uma organização de renome como a Agência de Segurança Nacional do Estados Unidos da América e é executado em nível de *kernel*, assim depende do reinicio do sistema operacional para total desativação, ao contrário do AppArmor que como "serviço" pode ser parado, iniciado ou reiniciado de maneira independente do sistema operacional, desta maneira um usuário com permissão administrativa (*root*) pode desativar o AppArmor e ninguém perceber de imediato. Em contrapartida, no item usabilidade a pesquisa foi favorável ao AppArmor, que é apontado como mais simples de configurar e administrar, principalmente pela disponibilidade da ferramenta Yast (presente no OpenSUSE e SUSE da Novell), embora o SELinux também possua uma interface gráfica, chamada de system-config-selinux.

Mas essa facilidade oferecida pelas interfaces gráficas se aplicam de maneira mais importante em computadores pessoais pois sabidamente servidores GNU/Linux na grande maioria dos casos são instalados sem o modo gráfico. Desta forma uma maneira consistente de aferir se AppArmor é mais simples de utilizar que SELinux seria diante de um laboratório para uso das duas ferramentas de configuração automática e análise da sintaxe de regras de configuração de políticas de acesso, que não foi o escopo deste estudo, ficando como uma possibilidade de trabalho futuro.

Para finalizar, embora a monografia tenha sido, principalmente, baseada na comparação entre o SELinux e AppArmor, não se objetivou de maneira alguma levantar qual é o melhor deles e sim alertar sobre a extrema importância da Segurança da Informação (em todos os âmbitos da Informática, seja pessoal ou corporativa) e também de uma certa forma desmistificar que o GNU/Linux por padrão é extremamente seguro e não merecedor de atenções especiais depois de instalado, diante disto se apresentou duas alternativas de segurança adicional, além do DAC, para o sistema operacional, ficando ao critério do utilizador avaliar qual subsistema instalar.

REFERÊNCIAS

APPARMOR. Disponível em: <http://old-en.opensuse.org/AppArmor>
Acesso em: 04 ago. 2013.

ARANTES FILHO, Sócrates. Segurança da Informação: autenticação. Disponível em: <http://waltercunha.com/blog/index.php/2009/08/19/seguranca-da-informacao-autenticacao/#minimize> Acesso em: 03 ago. 2013.

BAADER, Hans-Joachim. Novell startet AppArmor-Projekt. Disponível em: <http://www.pro-linux.de/news/1/9136/novell-startet-apparmor-projekt.html> Acesso em: 20 jul. 2013.

BALTZAN, Paige; PHILLIPS, Amy. **Sistemas de Informação.** Porto Alegre: AMGH Editora, 2012.

CARDOZO, Heitor Augusto Murari. Introdução ao SELinux. Disponível em: <http://ww.ha-mc.org/node/22> Acesso em: 17 jul. 2013.

CARVALHO, Leandro Inácio Santos de. **Um estudo sobre o uso de Controle de Acesso Pró-Ativo Baseado em Papéis para a Contenção de Falhas de Segurança Desconhecidas em Serviços no Ambiente GNU/Linux.** 2011. 33 f. Monografia (apresentada ao final do curso de pós-graduação lato sensu em Redes de Computadores) – Faculdade de Ciências Exatas e Tecnológicas, Centro de Estudo de Maceió, Alagoas.

CHESWICK, Willian R.; BELOVIN, Steven M.; RUBIN, Aviel D.. **Firewalls e segurança na Internet:** repelindo o hacker ardiloso. 2. ed. Porto Alegre: Bookman, 2005.

DEBIAN ultrapassa CentOS como servidor web Linux mais utilizado. **Linux Magazine Online**, [s.l.], 11 jan. 2012. Disponível em: <https://linuxmagazine.com.br/lm/noticia/a_popularidade_do_debian> Acesso em: 23 mai. 2013.

ESCOSTEGUY, Ana Carolina D.; GUTFREIND, Cristiane Freitas. **Leituras em Comunicação, Cultura e Tecnologia.** Porto Alegre: Edipucrs, 2007.

ESTATÍSTICAS dos incidentes reportados ao CERT.br. Disponível em: <http://www.cert.br/stats/incidentes/> Acesso em: 04 ago. 2013.

FLYNN, Ida M.; MCHOES, Ann McIver. **Introdução aos sistemas operacionais.** São Paulo: Pioneira Thomson Learning, 2002.

GELBMANN, Matthias. Debian is now the most popular Linux distribution on web servers. **W3techs**, [s.l.], 9 jan. 2012. Disponível em: <http://w3techs.com/blog/entry/debian_is_now_the_most_popular_linux_distribution_on_web_servers> Acesso em: 23 mai. 2013.

GOODRICH, Michael T.; TAMASSIA, Roberto. **Introdução à segurança de computadores.** Porto Alegre: Bookman, 2013.

GUIMARÃES, Alexandre Guedes; LINS, Rafael Dueire; OLIVEIRA, Raimundo. **Segurança com redes privadas virtuais VPNs.** Rio de Janeiro: Brasport, 2006.

HAMZE, Amelia. As ondas da vida e do conhecimento. Disponível em: <http://educador.brasilescola.com/gestao-educacional/ondas-da-vida-e-conhecimento.htm> Acesso em: 05 jul. 2013.

HESS, Pablo. Bê-á-bá do MAC no Linux, parte 6: AppArmor. Disponível em: <https://www.ibm.com/developerworks/community/blogs/752a690f-8e93-4948-b7a3-c060117e8665/entry/be-a-ba_do_mac_no_linux_parte_6_apparmor?lang=en> Acesso em: 17 jul. 2013.

HISTORICAL trends in the usage of linux versions for websites. Disponível em: <http://w3techs.com/technologies/history_details/os-linux> Acesso em: 23 mai. 2013.

INCIDENTES reportados ao CERT.br -- Janeiro a Março de 2013. Disponível em: <http://www.cert.br/stats/incidentes/2013-jan-mar/analise.html> Acesso em: 04 ago. 2013.

IVASHKO, Evgeny. Linux Seguro: Parte 1. SELinux – história de seu desenvolvimento, arquitetura e princípios operacionais. Disponível em: <http://www.ibm.com/developerworks/br/library/l-secure-linux-ru/#history> Acesso em: 02 abr. 2013.

LEITNER, Achim. Novell e Red Hat confrontam suas tecnologias: AppArmor x SELinux. **Linux Magazine**, edição 22, ano 2, nº 8, p. 42-45, ago. 2006.

MACHADO, André; VENEU, Aroaldo; OLIVEIRA, Fernando de. **Linux:** comece aqui. Rio de Janeiro: Elsevier, 2005.

MADEIRA, Frederico. Métodos de controle de acesso. Disponível em: <http://www.madeira.eng.br/wiki/index.php?page=Métodos+de+Controle+de+Acesso> Acesso em: 13 jul. 2013.

MAYER, Frank; MACMILLAN, Karl; CAPLAN, David. **SELinux by example:** using security enhanced Linux. United States of America: Prentice Hall, 2007.

MORIMOTO, Carlos Eduardo. **Linux, entendendo o sistema:** guia prático. Rio de Janeiro: GDH Press e Sul Editores, 2006.

MORIMOTO, Carlos Eduardo. Uma introdução ao AppArmor. Disponível em: <http://www.hardware.com.br/dicas/introducao-apparmor.html> Acesso em: 02 abr. 2013.

NOVELL AppArmor Administration Guide. Disponível em: <https://www.suse.com/documentation/apparmor/apparmor201_sp2_admin/?page=/documentation/apparmor/apparmor201_sp2_admin/data/apparmor201_sp2_admin.html> Acesso em: 04 ago. 2013.

SANTOS, Douglas dos. Controle de acesso: AppArmor. Disponivel em: <http://blog.andradesoto.com.br/index.php/controle-de-acesso-apparmor/> Acesso em: 17 jul. 2013.

SCHERF, Thorsten. Controle obrigatório de acesso com o SELinux: acesso restrito. **Linux Magazine**, edição 45, ano 4, nº 8, p. 66-73, ago. 2008.

SELINUX. Disponível em: <http://br.redhat.com/resourcelibrary/articles/article-selinux> Acesso em: 04 ago. 2013.

SIEVER, Ellen; WEBER, Aaron; FIGGINS, Stephen; LOVE, Robert; ROBBINS, Arnold. **Linux:** o guia essencial. 5. ed. Porto Alegre: Bookman, 2006.

SOUZA, Tiago Almeida Barboza de. **SECMEC: Assistente para Instalação de Mecanismos de Segurança voltado para Administradores de Sistema.** 2007. 96 f. Trabalho de Conclusão de Curso (apresentado ao final do curso de graduação Tecnólogo em Informática) – Universidade Tecnológica Federal do Paraná, Curitiba, Paraná.

SPENNEBERG, Ralf. Acesso sob controle: SELinux. **Linux Magazine**, edição 22, ano 2, nº 8, p. 38-41, ago. 2006a.

SPENNEBERG, Ralf. Barrando intrusos com o AppArmor: armadura segura. **Linux Magazine**, edição 22, ano 2, nº 8, p. 34-37, ago. 2006b.

SUZUKI, Luis Henrique; DELPHINO, Alexandre Dantas. **Implantação de um Honeypot e proposta para metodologia de gerenciamento de projetos de Honeynets.** 2007. 79 f. Monografia (apresentada ao final do curso de Bacharelado em Ciência da Computação) – Universidade de Brasília, Brasília, 2007.

TEIXEIRA, Jarbas. **Linux sem segredos.** São Paulo: Digerati Books, 2008.

VENEZUELA, Sandro. Tornando-se um “Artista de Segurança Linux”. Disponível em: <http://www.linux2business.com.br/site/tornando-se-um-artista-de-seguranca-linux-2/> Acesso em: 17 jul. 2013.