

# Mitigação do Risco Operacional através da Cifragem do Sistema de Arquivos em Servidores de Aplicações

Miguel Ângelo Chagas Neumann<sup>1</sup>; André Peres<sup>2</sup>

<sup>1</sup> Acadêmico do Curso de Sistemas de Informação da ULBRA Guaíba  
< miguel@rwx.com.br >

<sup>2</sup> Professor Orientador do Curso de Sistemas de Informação da ULBRA Guaíba  
< andre.peres@ulbra.br >

**Resumo** - Este artigo refere-se ao Seminário de Andamento de TCC do curso de Sistemas de Informação 2009/01. Nele está sendo apresentada uma pesquisa que visa prover processos tecnológicos que assegurem o adequado nível de segurança para as informações contidas nos dispositivos de armazenamento magnético de servidores de aplicações.

**Abstract** - This article refers to the Seminar on Progress of TCC of the course Information Systems 2009/01. It has been presenting a research that aims to provide technological processes that ensure the appropriate level of security for the information contained in magnetic storage devices for applications servers.

(Palavras-chave: cifragem de bloco, file system, LVM, DM-Crypt)

## 1. INTRODUÇÃO

Para viabilizar suas operações, as empresas realizam pesados investimentos em tecnologia da informação. Associado aos investimentos em performance, ferramentas e serviços, também está inclusa a segurança da informação corporativa e de clientes. Tal situação ocorre, pois a confidencialidade dos dados é um dos mais importantes ativos corporativos.

Ao acompanharmos a imprensa, encontramos vários exemplos de roubo ou vazamento de informações que ocasionaram prejuízos às corporações.

Segundo a McAfee, as empresas de tecnologia perderam 1 trilhão de dólares com roubo de dados em 2008. Paralelo a isso, temos a realização de terceirizações de manutenção de hardware, de instalação de aplicativos que mesmo com dispositivos legais e contratuais bem elaborados, pode permitir o vazamento de informações.

Nesse contexto, o presente projeto pretende apresentar a cifragem do *file system* de servidores, como uma alternativa para proteção dos dados corporativos.

#### OBJETIVOS DO TRABALHO

Como objetivo geral, pretendemos fornecer a administradores de servidores de aplicações em plataforma Linux, a possibilidade de adequar seus sistemas à normatização e melhores práticas de segurança da informação, provendo processos tecnológicos que assegurem o adequado nível de segurança para as informações contidas nos dispositivos de armazenamento magnético de servidores de aplicações.

Com a pesquisa, visamos demonstrar a viabilidade da proposta em uma máquina de produção e especificamente, documentar o processo de criptografia de file system e realizar análises de desempenho entre um sistema cifrado e outro não-cifrado.

## 2. REFERENCIAL TEÓRICO

A pesquisa envolve as áreas de Criptografia, Cifragem de File Systems e Análise de desempenho.

### a) Criptografia

A empresa de segurança RSA define criptografia como a arte e ciência de utilizar a matemática para segurança da informação e criar um alto grau de confiança no ambiente eletrônico. Também, dentro do campo da Criptografia encontra-se a Autenticação que utiliza-se da Assinatura Digital para dar legitimidade a documentos originados eletronicamente (RSA Laboratories).

Utilizamos a criptografia para assegurar a privacidade, mantendo a informação escondida para qualquer pessoa para quem não se destina, inclusive para aqueles que possuem acesso aos dados criptografados.

A criptografia é uma operação lógico-aritmética e é representada por algoritmos matemáticos que realizam operações a nível de bit, sobre um conjunto inicial de informações, gerando um grupo de bits aleatórios que é a chave de acesso aos dados a serem criptografados.

Com relação aos tipos de chaves criptográficas, os algoritmos são classificados em três tipos: Simétricos, Assimétricos e de Hash.

Os algoritmos simétricos são caracterizados por possuírem somente uma chave para criptografar e decriptografar as mensagens. A chave deve ser conhecida somente entre as partes que pretendem realizar a comunicação.

Os algoritmos simétricos podem ser divididos em Algoritmo Simétrico de Bloco ou Algoritmo Simétrico de Fluxo.

Um Algoritmo Simétrico de Bloco, a mensagem é dividida em blocos que são cifrados um a um separadamente. Algoritmos de cifragem por bloco são recomendáveis para criptografarmos dados estáticos em dispositivos de armazenamento como discos rígidos, por exemplo. Nesse sentido, foi a alternativa que optamos para utilizar no presente projeto.

Dentre os algoritmos de bloco podemos citar o Advanced Encryption Standard – AES. O AES é o algoritmo criptográfico adotado pelo governo norte-americano e suporta chaves de 128, 192 e 256 bits. Tendo em vista a velocidade de execução do algoritmo AES, sua segurança e a aplicação no mercado será a solução utilizada no presente projeto de pesquisa.

Nos Algoritmos Simétricos de Fluxo, a mensagem é cifrada de forma dinâmica e tipicamente na menor unidade da mensagem, usualmente em bits.

Atualmente os algoritmos de fluxo utilizam chaves criptográficas do tipo one-time pads, que são chaves geradas de forma pseudo-aleatória e utilizadas em uma única sessão. Outra característica é que uma mesma mensagem pode gerar diferentes resultados após a cifragem. Normalmente utilizamos algoritmos de fluxo para comunicação de dados.

Algoritmos de Criptografia Assimétricos também podem ser identificados como criptografia de chave pública e são utilizados para criptografar ou para assinar digitalmente mensagens.

Nos algoritmos assimétricos, cada usuário tem uma chave pública e uma chave privada. As mensagens cifradas com uma chave privada, somente podem ser decifradas com a chave pública e vice-versa. A chave pública é distribuída para as pessoas que se deseja trocar mensagens criptografadas e a chave privada permanece secreta para o proprietário.

Como as chaves pública e privada são matematicamente ligadas e existe notoriedade na chave pública, existe a possibilidade de ataques de derivação à chave privada. A solução é ampliar a complexidade no cálculo entre as chaves e utilizar chaves maiores.

Uma função de *hash* realiza a transformação de uma entrada em uma *string* de tamanho fixo que representa de forma concisa a mensagem ou documento para qual foi calculada.

A principal utilidade de uma função de *hash* está em identificar a integridade de mensagens e checar assinaturas digitais.

#### b) Cifragem de File Systems

Para implementar o projeto utilizaremos Public-Key Cryptography Standards (PKCS), que são especificações publicadas a partir de 1991, pelo RSA Laboratories, com o propósito de agilizar o desenvolvimento de algoritmos criptográficos. São padrões utilizados pelo mercado e, inclusive, sendo transformados em Request For Comments (RFC) pelo Internet Engineering Task Force (IETF) e citados como método OSI para implementação de criptografia pelo OSI Implementer's Workshop (OIW).

O PKCS número 5 (PKCS #5) define recomendações para a implementação de criptografia baseada em senha, cobrindo funções de derivação de chaves, ASN.1 , esquemas de encriptação e de autenticação de mensagens.

Dentre os itens apresentados pelo PKCS #5 e que serão utilizadas no projeto, podemos citar:

a. Derivação – para criptografia baseada em senha, também podemos utilizar técnicas de derivação de chave que sejam relativamente expansivas, incrementando a dificuldade em ataques de força bruta.

b. *Salting* – em muitas aplicações de criptografia de chave pública, a segurança das informações depende em última instância de senhas ou textos secretos. Como as senhas geradas não oferecem a entropia (aleatoriedade) necessária para garantir a segurança dos dados, as aplicações de mercado não utilizam diretamente a senha como chave de criptografia. Uma solução para isso é combinar a senha com dados adicionais denominados de *salt* (sal). O *salt* pode ser entendido como um índice para um grupo de chaves derivadas da senha, cada senha terá um conjunto de chaves correspondentes. Ele não necessita ser mantido em segredo e objetiva evitar ataques por dicionário de senhas e chaves.

$$DK=KDF(P,S)$$

DK=Derived key, KDF=Key derivation function, P=Password, S=Salt

Dicionário Sem Salt

<u>Senhas</u>	<u>Chaves</u>
Mamae	09ds0se9
12/06/1920	54fds564
Brasil	gf4564dr

Dicionário Com Salt

<u>Senhas</u>	<u>Chaves</u>
Mamae	?
12/06/1920	?
Brasil	?

c. *Iteration count* – outra ferramenta utilizada em derivação de senhas é *iteration count* (contador de interação), que indica quantas vezes a função foi executada e o resultado é inserido como parâmetro na função de derivação de senha.

$$DK=KDF(P,S,IC)$$

DK=Derived key, KDF=Key derivation function, P=Password, S=Salt, IC=Iteration count

d. Password-Based Key Derivation Function 2 (PBKDF2) – A Função para Derivação de Chave Baseada em Senha Versão 2, do PKCS #5, está baseada nos conceitos de *salt* e *iteration count*. A PBKDF2 é uma função baseada em três parâmetros: senha, *salt* e *iteration count*. Onde os dois últimos fatores não necessitam ser mantidos em segredo. A PBKDF é muito utilizada para encriptação baseada em senha e em esquemas de autenticação de mensagens. A PBKDF2 aplica uma função pseudo-aleatória para gerar chaves derivadas com tamanho previamente definido. O tamanho da chave derivada torna-se mais um parâmetro da função.

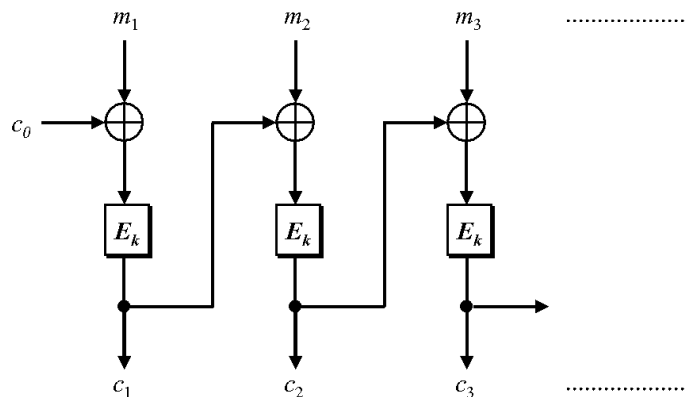
$$DK=PBKDF2(P, S, IC, DKLen)$$

DK=Derived key, KDF=Password-Based Key Derivation Function, P=Password, S=Salt, IC=Iteration count, DKLen=DK Length

Além do PKCS #5, podemos utilizar outras técnicas para melhorar a segurança na criptografia de bloco. Dentre elas podemos citar a utilização de Hierarquia de Chaves, que é um método de gestão das chaves criptográficas em que cada nível do sistema ou ambiente possui tipos de chaves diferentes. Normalmente, as chaves de maior nível não são reveladas, mas somente acessadas através de chaves operacionais com nível inferior. Por exemplo, criptografaremos o disco rígido com a chave mestra e esta com a chave de acesso do usuário ou sistema que montará o dispositivo. Além de permitir a segregação de poderes, essa estrutura permite que em trocas de senhas dos usuários ou sistemas, não seja necessário criptografar todo o dispositivo novamente.

A criptografia de bloco permite a cifragem de tamanho fixo e limitado ao tamanho do bloco. Como dificilmente as mensagens terão tamanho menor ou igual ao bloco, precisamos evitar que a mesma mensagem, após criptografada, gere a mesma saída e, para isso, utilizamos os modos de operação para os algoritmos de criptografia. Dentre eles podemos citar: ECB, LRW-AES, IAPM, CCM, EAX, GCM, OCB, CMC, EME.

Os modos de operação mais avançados, além da confidencialidade, proporcionam autenticação e maior velocidade ao processo, entretanto, alguns ainda não foram totalmente testados. Em decorrência disso, um dos modos mais utilizados é o Cipher-Block Chaining – CBC, que realiza uma operação XOR entre o bloco cifrado anterior e o bloco atual e em seguida, ocorre criptografia do bloco atual. Para iniciar o primeiro bloco é utilizada uma semente chamada *Initialization Vector* (IV) que pode ser o número do bloco ou algum número aleatório.



**Figura 1 - Funcionamento do modo CBC**

O CBC será o modo utilizado no presente trabalho de pesquisa.

O *Logical Volume Manager* – LVM é uma ferramenta para a plataforma \*nix, que realiza o mapeamento entre os dispositivos físicos, como o disco rígido, e os dispositivos lógicos do sistema operacional. Também permite montarmos estruturas RAID por software, criarmos volumes lógicos, possibilita o redimensionamento de

partições sem afetar os dados contidos e amplia as possibilidades de denominação de partições.

O LVM é utilizado pelo módulo DM-Crypt do *kernel* do sistema operacional Linux para realizar a criptografia de dispositivos. O DM-Crypt utiliza uma camada do *kernel* Linux chamada *device manager* para cifrar o acesso ao disco de forma transparente para o próprio sistema operacional.

Para presente projeto iremos utilizar uma ferramenta chamada Cryptsetup-LUKS, que gerencia os cabeçalhos das partições cifradas e implementa o referencial teórico acima citado, como por exemplo, PKCS #5, hierarquia de chaves e ainda possui outras proteções como ESSIV e antiforensics.

c) Análise de desempenho

A análise de desempenho que realizaremos será focada na performance de leitura e escrita de dados criptografados no disco rígido.

Resta-nos ainda, identificar melhor metodologia para avaliar a performance.

### **3. SOLUÇÃO PROPOSTA**

No presente trabalho, pretendemos instalar dois servidores Linux num único host, em ambientes computacionais semelhantes, mas onde um possua o *file system* cifrado e o outro possua o *file system* instalado em modo padrão, ou seja, sem criptografia.

Como os sistemas operacionais carregam para a memória dados dos processos, pretendemos cifrar o dispositivo de *swap*, onde são gravados os arquivos de troca da memória principal, bem como, realizaremos buscas de dados contidos nas partições de *swap* dos dois ambientes para verificar o grau de segurança obtido.

A maioria das soluções para criptografia de dispositivos de armazenamento permite a inserção da chave de acesso de forma manual, no momento da realização do acesso ao dispositivo. Pretendemos que o acesso à chave criptográfica ocorra automaticamente no *boot* do sistema, através de acesso do host a um servidor na rede interna. Isso agilizaria a retomada do serviço após alguma indisponibilidade do servidor.

Realizaremos testes de desempenho focados em requisições de servidores de aplicações.

#### **4. CONCLUSÃO**

Apesar do foco da presente pesquisa não ser análise de técnicas criptográficas, entendemos que seu estudo, afeta diretamente um dos itens do trabalho que é o desempenho de dispositivos criptografados.

Constatamos ainda que, existem outras possibilidades para melhora do desempenho de servidores com *file systems* cifrados, que é a utilização de estruturas de RAID e talvez pudessem ser exploradas em trabalhos futuros.

## 5. BIBLIOGRAFIA

OLIVEIRA, Rômulo de; Carissimi, Alexandre; Toscani, Simão. Sistemas Operacionais 3 Ed. Porto Alegre: Sagra Luzzatto, 2004.

MORIMOTO, Carlos. Redes e Servidores Linux. Porto Alegre: Sulina, 2005.

MORIMOTO, Carlos. Linux: Ferramentas Técnicas. Porto Alegre: Sul Editores, 2005.

LIMA, João Paulo. Administração de Redes Linux. Goiânia: Gráfica Terra Ltda., 2003.

FARNER, Dan; Venema, Wietse. Perícia Forense Computacional Teoria e Prática Aplicada. São Paulo: Pearson Prentice Hall, 2006.

FREITAS, Andrey. Perícia Forense Aplicada à Informática – Ambiente Microsoft. São Paulo: Brasport, 2006.

RUFINO, Nelson Murilo de O. Segurança Nacional Técnicas e Ferramentas de Ataque e Defesa de Redes de Computadores. São Paulo: Novatec, 2001.

DIAS, Cláudia. Segurança e Auditoria da Tecnologia da Informação. São Paulo: Axcel Books, 2000.

IAIB. Instituto dos Auditores Internos do Brasil. Procedimentos de Auditoria Informática. São Paulo: Câmara Brasileira de Auditoria Informática.

ABNT. Associação Brasileira de Normas Técnicas. ABNT NBRISO/IEC 17799:2005. Rio de Janeiro: 2005.

ABNT. Associação Brasileira de Normas Técnicas. ABNT NBRISO/IEC 27005:2008 Rio de Janeiro: 2009.

FEBRABAN. Metodologia de Auditoria Interna com Foco em Riscos. São Paulo: 2004.

FEBRABAN. Novos Conceitos de Auditoria de Sistemas em Bancos. São Paulo: 2004.

RSA. RSA Laboratories. Standards Initiatives. Public-Key Cryptography Standards (PKCS). Disponível em < <http://www.rsa.com/rsalabs/node.asp?id=2124> >. Acesso em: 04 jun 2009.

IETF. Internet Engineering Task Force. Request for Comments. Disponível em < <http://www.ietf.org/rfc.html> >. Acesso em: 04 jun 2009.

DEBIAN. Debian.org. Guia de Instalação de Debian GNU/Linux. Disponível em < <http://www.debian.org/releases/stable/i386/ch06s03.html.pt> >. Acesso em: 04 jun 2009.