SAMBA: Compartilhamento de Arquivos com Auditoria de Segurança

Danilo Henrique dos Santos Balica[1], Tiago Lopes de Aguiar[2]

[1]Acadêmico do Curso Superior de Tecnologia em Redes de Computadores, Turma 2017/1, *e-mail*: danilohenrique.pvh@gmail.com; e
[2]Professor orientador do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia, *Campus* Porto Velho Zona Norte, *e-mail*: tiago.aguiar@ifro.edu.br.

Resumo: Este artigo trata-se de uma pesquisa aplicada/prática, em que, foi utilizado a ferramenta Samba para, com métodos exploratória e descritiva analisar a viabilidade de implementação de um servidor de arquivos com esquema de auditoria em ambiente corporativo, em termos de custo, compatibilidade, e impacto na segurança da informação. A pesquisa foi realizada em ambiente virtualizado, foram implementados 4 (quatro) hosts no total, um *Controlador de Domínio* para criação de unidades organizacionais, grupos e usuários; um *Servidor de Arquivos* para compartilhar arquivos com usuários devidamente autenticados pelo domínio (ambos implementados com Samba) e dois hosts com sistema operacional clientes (Windows 10 e Linux Ubuntu), para testar e analisar as funcionalidades da ferramenta Samba, com objetivo de coletar dados com esquema de auditoria implementado e posteriormente, apontar os fatores positivos e negativos.

Palavras-chave: Samba. Compartilhamento de arquivos. Auditoria de segurança.

SAMBA: FILE SHARING WITH SECURITY AUDIT

Abstract: *This article is an applied/practical research, in which the Samba tool was used to, with exploratory and descriptive methods, analyze the feasibility of implementing a file server with an audit scheme in a corporate environment, in terms of cost, compatibility, and impact on information security. The research was carried out in a virtualized environment, 4 (four) hosts were implemented in total, a Domain Controller for creating organizational units, groups and users; a File Server to share files with users duly authenticated by the domain (both implemented with Samba) and two hosts with client operating systems (Windows 10 and Linux Ubuntu), to test and analyze the functionalities of the Samba tool, with the aim of collecting data with an audit scheme implemented and subsequently, point out the positive and negative factors.*

Keywords*: Samba. File sharing. Security auditing.*

1. INTRODUÇÃO

O desenvolvimento contínuo de informações digitais nos ambientes organizacionais implica a necessidade de sistemas que viabilizem o armazenamento e o acesso fácil a essas informações. A segurança e o compartilhamento de arquivos entre sistemas de diferentes plataformas podem gerar conflitos. Entretanto, ao contar com um servidor, a empresa poderá manter arquivos gerenciados, disponibilizando um local centralizado para armazenar e organizar documentos importantes, garantindo sua disponibilidade conforme a demanda

(GAIDARGI, 2018).

Para compartilhar arquivos na rede existe o SMB (*Server Message Block*), de acordo com Microsoft (2023), é um protocolo cliente/servidor de compartilhamento de arquivos que permite que aplicativos de um computador solicitem serviços de leitura e gravação de arquivo na rede de computadores. Conforme, a ISO/IEC27001 não basta apenas armazenar e compartilhar, é preciso atender todos os requisitos e controles necessários para proteger a privacidade, disponibilidade e integridade de dados e sistemas sensíveis, de acordo com ABNT (2006), devem ser mantidos e estabelecidos registros de evidências de conformidades com suas responsabilidades perante a lei e a necessidade para melhoria contínua.

O objetivo básico da auditoria é estabelecer responsabilidade para entidades do sistema que iniciam ou participam de eventos e ações relevantes à segurança, de acordo com (STALLINGS; BROWN, 2014, p. 520).

Para garantir o processo de compartilhamento de arquivos o Samba por meio do protocolo SMB, permite o compartilhamento de arquivos na rede entre sistemas operacionais distintos, permitindo restaurar arquivos excluídos acidentalmente, por meio da habilitação da lixeira e bloquear armazenamento de arquivos executáveis, para proteger *integridade* e bloquear arquivos grande como *.iso* para não afetar a *disponibilidade*, entre outros, de acordo com desejado pelo administrador. Afirma Brito (2017), podendo inclusive autenticar usuários como controlador de domínio. Além de ser uma ferramenta em código aberto, não gera custo com licenças (SAMBA, 2020).

Portanto, sabe-se a importância de um servidor de arquivo conjugado com critérios de auditoria para ambientes corporativos, cuja necessidade de uma ferramenta que viabiliza o processo de armazenamento e compartilhamento de arquivos de forma segura, que garanta o acesso fácil e rápido a essas informações com baixo custo a organização. Dessa forma, o Samba pode ser usado para esta função, pois, oferece recursos de gestão importantes para administrar o servidor de arquivo com segurança.

2. REFERENCIAL TEÓRICO

2.1. SERVIDOR DE ARQUIVOS

O “servidor de arquivos” de acordo com Costa (2011) é uma máquina que serve (entrega) os dados para os outros computadores (chamados de “clientes”), dedicada ao trabalho de arquivar e gerenciar informações que os usuários inserem em suas pastas, além disso, ele controla a rede e traz vantagens na agilidade de processos.

2.2. PROTOCOLO SMB/CIFS

O protocolo SMB (*Server Message Block*), é um protocolo de compartilhamento de arquivos em rede que permite que os aplicativos de um computador leiam e gravem em arquivos, inicialmente desenvolvida por Barry Feigenbaum em 1980, enquanto trabalhava na IBM (*International Business Machines*), afirma Costa (2011), que em pouco tempo a Microsoft tomou a frente do desenvolvimento do protocolo e em 1996 passou a ser chamado de SMB/CIFS (*Server Message Block/Cammon Internet File System*), conjunto de aplicativos Samba, resume-se em torno de *daemons* que executam as funções e compartilham recursos na rede SMB. O *daemom* SMBD, basicamente cuida dos serviços de arquivos e impressão e controla o sistema de autenticação de usuários, que fazem parte do conjunto CIFS.

2.3. SAMBA

Segundo Carter, Jay e Eckstein (2007), o Samba é uma criação de Andrew Tridgell, que iniciou o projeto em 1991, enquanto trabalhava com um conjunto de *software* da DEC (*Digital Equipment Corporation*) chamado Patchworks. Foi criado para conectar computadores DEC VAX a computadores feitos por outras empresas. O Samba implementa o protocolo SMB (*Server Message Block*) que permite que computadores, executando sistemas operacionais baseados em Unix, se comuniquem com o Microsoft Windows e outros clientes habilitados para SMB e servidores.

De acordo com Costa (2011), o Samba não é apenas um aplicativo, mas sim um conjunto de aplicativos instalados em um servidor Linux que se comunicam através do protocolo SMB/CIFS (*Server Message Block/Cammon Internet File System*), dividido em dois módulos principais: o Servidor Samba, propriamente dito; e o NBNS (*Netbios Name Service*), que permite acessar compartilhamentos em outras máquinas. Usando o protocolo SMB, o Samba permite integração de ambientes heterogêneos entre sistemas Linux, Windows e Mac OS, viabilizando tanto o simples compartilhamento de arquivos e impressoras, com controle de acesso dos usuários, quanto o serviço de diretórios para que o servidor Linux seja controlador de domínio.

2.4. SEGURANÇA DA INFORMAÇÃO

Segurança da informação, explica Sêmola (2014), é o campo do conhecimento destinado à proteção de ativos da informação contra acessos não autorizados, alterações indevidas ou sua indisponibilidade. No sentido amplo, pratica a gestão de riscos e incidentes que comprometam

a integridade dos três princípios básicos de conceitos de segurança: confidencialidade, integridade e disponibilidade, denominado tríade CID explica Stallings (2014). O princípio da confidencialidade exige que a informação privada ou confidencial não fique disponível nem seja revelada a indivíduos não autorizados. O princípio da integridade consiste em garantir que somente pessoas especializadas e autorizadas possam fazer alterações nos programas e dados. O princípio da disponibilidade garante aos usuários autorizados que não haja negação de serviço aos sistemas.

2.5. AUDITORIA

Auditoria é o processo de coleta de evidências de uso dos recursos existentes, a fim de identificar as entidades envolvidas em um processo de troca de informações, ou seja, origem, destino e meios de tráfego de uma informação, afirma Sêmola (2014). De acordo com Kim e Solomom (2014) o objetivo em se realizar uma auditoria é garantir que os sistemas de uma organização e seus controles estejam funcionando de acordo com o esperado, ou seja, que atendam as expectativas previamente definidas.

O alicerce de uma instalação de auditoria de segurança é a captura inicial de dados de auditoria. Stallings e Brown (2014) explicam que isso requer que o *software* inclua “ganchos”, ou pontos de captura, que acionam a coleta e o armazenamento de dados à medida que ocorrem eventos pré-selecionados. Tal função de coleta ou de registro de auditoria depende da natureza do *software* e varia conforme o sistema operacional subjacente e as aplicações envolvidas.

O Samba de acordo com Carter, Jay e Eckstein (2007), lê registros de log de um “tdb” (banco de dados Trivial) associado armazenado na pasta “eventlog” (visualizador de eventos), do seu diretório de bloqueio (por exemplo, “/var/lib/samba/eventlog”). Esses arquivos tdb são criados usando uma ferramenta externa para analisar o sistema “logfiles” (registra eventos no conjunto de arquivos) e, em seguida, grava entradas no banco de dados usando a ferramenta “eventlogadm” (filtro que aceita registros de log de eventos formatados na entrada padrão e os grava no armazenamento de log de eventos do Samba), transformando o Samba em ferramentas de monitoramento para auditar os hosts Unix e suas contrapartes do Windows. Isso permite, ao administrador gerir normas e políticas de segurança da informação com relação ao ativo “dados” armazenado no servidor de arquivo, podendo responsabilizar os envolvidos em caso de quebra de políticas internas nas corporações.

2.6. SOFTWARE LIVRE E SOFTWARE PROPRIETÁRIO

Segundo Samba (2020), software livre é um software ou programa que concede liberdade ao usuário para executar, acessar e modificar o código fonte, redistribuir cópias com ou sem modificações sendo pagas ou gratuitas sua distribuição. Sua definição é estabelecida pela Free Software Foundation (FSF) em conjunto com o projeto GNU (Licença Pública Geral). O *Software* proprietário ou não-livre é o *software* que tem restringido por parte do proprietário a sua redistribuição, cópia e modificação. Negando qualquer uma das quatros liberdades ao usuário. Os direitos são exclusivos do produtor tendo de ser respeitados os direitos autorais e as patentes (SAMBA, 2020).

De acordo com Sebrae (2016).

> O preço de algumas licenças chega a custar 70% do valor do software. Ou seja, imagine um software (o programa em si) que custe R$ 1.000,00, a sua licença a 70% seria algo em torno de R$ 700,00. Além disso, quando se compra um software proprietário só se tem "o direito" de usar em um único computador. A instalação em outro computador é ilegal, o que é considerado Pirataria de Software. Logo a economia em uma empresa que decide pela utilização de software livre é muito alta.

Segundo Microsoft (2019), se as estações de trabalho na sua organização estão em rede, é provável que você dependa de software para servidores em rede para realizar determinadas funções, como compartilhamento de arquivo e impressão. Para acessar esse software para servidores legalmente, uma CAL (licença de acesso para cliente) poderá ser requerida.

3. METODOLOGIA

Trata-se de uma pesquisa aplicada, com vistas à implementação de servidor de arquivos por meio do uso de Samba, testando suas potencialidades no compartilhamento de arquivos com critérios de auditoria. Com experimentações práticas, descrevendo as etapas de instalação e configuração do servidor de arquivos Samba, que se caracteriza por ser exploratória e descritiva. Além disso, a pesquisa classifica-se também em experimental que, de acordo com Cervo (2007), trata-se da manipulação e criação de situações de controle a fim de observar o que acontece com a mesma.

Cervo (2007), define a pesquisa descritiva como o processo de observar, registrar, analisar e correlacionar fatos ou fenômenos sem a manipulação dos mesmos. Portanto, o presente artigo classifica-se como tal, pois permitiu a análise das configurações no compartilhamento de arquivos que impactam diretamente na segurança dos dados armazenados por corporações. Quanto às estratégias adotadas, trata-se de uma pesquisa qualitativa, pois

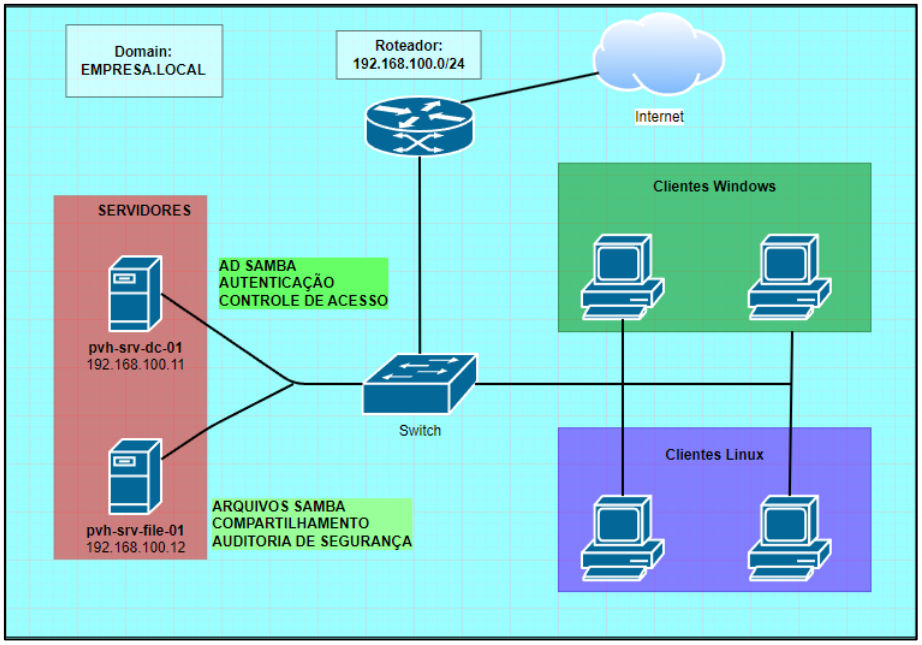
buscou resultados através de testes virtuais com procedimentos práticos.

As experimentações foram realizadas em laboratório virtual, a fim de configurar as ferramentas Samba para testar permissões nos compartilhamentos e coleta de dados para auditoria, apontando ao final a viabilidade na utilização do servidor de arquivos Samba quanto ao compartilhamento de arquivos. Por padrão, para compartilhamento de arquivos o sistema Linux utiliza o protocolo NFS (Network File System) e sistemas Windows utiliza o NetBIOS (Network Basic Input/Output File System), esses protocolos não se comunicam, por isso, a importância de utilizar o Samba para compartilhamento de arquivos.

4. RESULTADOS E DISCUSSÕES

A coleta de dados foi realizada em ambiente de simulação virtualizado por meio da ferramenta Virtual Box, hospedado no *notebook* HP 246Gz, com CPU i5-7200U e 16GB memória RAM. Foi implementado 4 (quatro) *hosts* todos com três interfaces de redes: (1ª interface INTERNA, para comunicação entre os *hosts*; 2ª interface NAT, para comunicação com internet; e a 3ª interface HOST-ONLY, para comunicação no protocolo SSH (*Secure Shell*) com o hospedeiro), no total foram dois *hosts* servidores e dois *hosts* clientes, conforme figura a seguir:

Figura 1 – Topologia da rede.



Fonte: Próprio autor (2023).

O primeiro *host* que foi implementado foi o servidor *"controlador de domínio Samba"* no sistema operacional Linux Debian 10 (*netinst*), com o objetivo de gerenciar computadores e

usuários tendo a principal função a de autenticar usuários aos *desktops* e sistemas internos, proporcionando maior segurança no acesso às informações, identificando pessoas, equipamentos, sistemas e agentes em geral, fundamental para os atuais padrões de informatização. Através deste servidor foi possível implementar unidades organizacionais (OU) como: vendas, tecnologia da informação, diretoria, administração, recursos humanos e financeiro. Em ambos foram criadas contas de usuários e grupos correspondentes aos setores, e para cada setor um compartilhamento foi definido no servidor de arquivos, a fim de simular o âmbito interno de uma corporação. De acordo com o *site* oficial Samba (2020), foi necessário preparar o sistema operacional instalando as dependências de pacotes necessários para a instalação do Samba no Linux Debian:

```
apt-get install acl attr autoconf bind9utils bison build-essential \
debhelper dnsutils docbook-xml docbook-xsl flex gdb libjansson-dev krb5-user \
libacl1-dev libaio-dev libarchive-dev libatev libblkid-dev libbsd-dev \
libcap-dev libcups2-dev libgnutls28-dev libgpgme-dev libjson-perl \
libldap2-dev libncurses5-dev libpam0g-dev libparse-yapp-perl \
libpopt-dev libreadline-dev nettle-dev perl perl-modules-5.24 pkg-config \
python-all-dev python-crypto python-dbg python-dev python-dnspython \
python3-dnspython python-gpg python3-gpg python-markdown python3-markdown \
python3-dev xsltproc zlib1g-dev liblmdb-dev lmdb-utils libsystemd-dev libtasn1-bin -y
```

A partir deste ponto foi possível executar sua instalação por meio de pacotes específicos de cada distribuição ou pela compilação do arquivo disponível no site oficial. A diferença entre as duas instalações está na localização do arquivo de configuração do Samba, pois na compilação fica localizado em "/usr/local/samba/etc/samba/smb.conf", já na distribuição em em "/etc/samba/smb.conf":

O processo de compilação nada mais é do que extrair o pacote fonte e executá-lo. Com a ferramenta wget foi possível baixar do *site* oficial samba.org o pacote, com o comando: "wget -c https://download.samba.org/pub/samba/stable/samba-4.11.6.tar.gz". Após realizar o *download*, os arquivos foram extraídos, executando-se na sequência comando específicos (./configure, make e sudo make install) para finalizar o processo de instalação do servidor Samba. O administrador deve configurar os arquivos "/etc/network/interfaces", "/etc/resolv.conf", "/etc/hosts", "/etc/nsswitch.conf" e "/usr/local/samba/private/krb5.conf" ou "/etc/krb5.conf", antes de realizar o comando (samba-tool domain provision --use-rfc2307 --domian=EMPRESA --realm=EMPRESA.LOCAL) que se refere à criação dos bancos de dados AD (*Active Directory*), este comando ativa a função do servidor. No entanto, não será detalhada sua implementação neste artigo.

O segundo *host* foi o cliente, com sistema operacional Windows 10, que após ingressar

no domínio realizou as funções de acessar e testar as funcionalidades do servidor de arquivos, inclusive os compartilhamentos ativados. Além disso, se o usuário autenticado tiver poderes administrativos no controlador de domínio poderá usar as ferramentas de administração de servidor remoto do Windows (RSAT), permitindo que o administrador gerencie as funções e recursos do servidor através do Windows 10 com interface gráfica, de acordo com a Microsoft (2020) esta função só é permitida nas versões *profissional* e *enterprise* no sistema operacional cliente. O terceiro *host* foi o Linux Ubuntu 18.04 LTS, não diferente do Windows, este após ingresso no domínio teve a função de acessar e testar as funcionalidades do servidor de arquivos, em comparação com o cliente Windows.

O quarto *host* implementado foi o *"servidor de arquivos"* Samba no sistema operacional Linux Debian 10 (netinst), com 2GB de memória RAM, cujo qual foi criada partição em disco "/mnt/samba" para armazenar os compartilhamentos em rede, baseando-se em RAID por *softwares* nível 5, utilizando-se três discos virtuais de 10GB para gerar boa redundância e desempenho no armazenamento dos arquivos. Este servidor será ingressado no domínio, e sua função será de compartilhar arquivos para os usuários devidamente autenticados no controlador de domínio.

Partindo do princípio de que quarto *host* já estivesse integrado ao domínio EMPRESA.LOCAL, e que o objetivo agora seja transformá-lo em um servidor de arquivos, não diferente do controlador domínio, deve-se ser instalado os pacotes Samba. Uma vez instalado, o principal arquivo de configuração é o "/etc/samba/smb.conf". O comando *testparm* é muito importante, pois verifica se há erro nos parâmetros do smb.conf, facilitando o processo para o administrador, apontando eventuais falhas.

Os compartilhamentos de rede configurados no servidor de arquivo Samba têm o controle de acesso feito pelas seguintes permissões: permissões controladas pelo Samba, definidas no arquivo de configuração smb.conf; e permissões atribuídas no sistema de arquivos do Linux. Em ambos os casos, as permissões concedidas devem ser iguais e alinhadas para o bom funcionamento dos compartilhamentos.

Com o comando "mkdir -v -m 770 /mnt/samba/vendas" foi possível criar o diretório "vendas" no sistema operacional Debian com permissões totais ao proprietário e grupo, atribuindo ao grupo "vendas" do Controlador de Domínio Samba como novo proprietário do referido diretório (chmod -v "EMPRESA\vendas" /mnt/samba/vendas/), o comando (chmod -v g+s /mnt/samba/vendas/) conhecido como bit especial (g+s), para qualquer arquivo criado no diretório por usuários diferentes "EMPRESA\vendas" tenha como dono o grupo primário

(vendas). Esse método é o mais simples, ideal para corporações com poucos usuários.

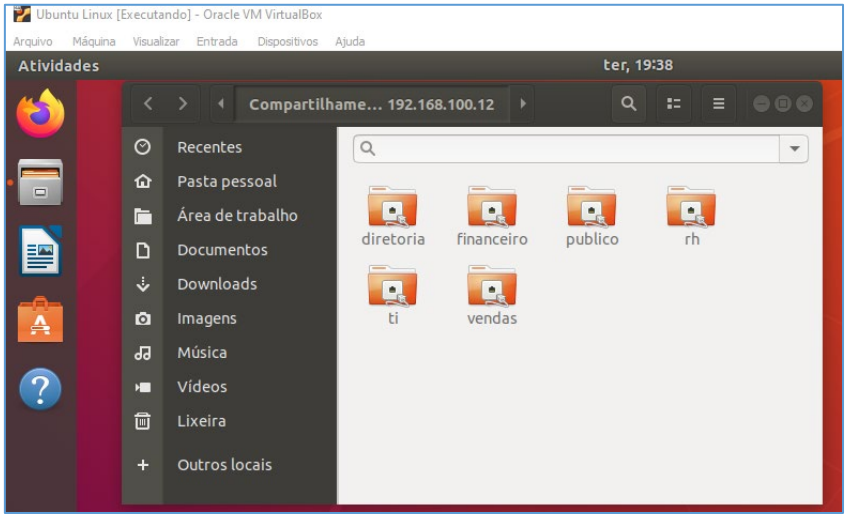
Exemplo de criação de compartilhamento para grupos de usuários no arquivo "smb.conf":

```
[vendas]
    path = /mnt/samba/vendas
    valid users = @"EMPRESA\vendas" @"EMPRESA\financeiro"
    write list = @"EMPRESA\vendas"
    read list = @"EMPRESA\financeiro"
    create mask = 660
    directory mask = 770
    force group = 'EMPRESA\vendas"
```

Considerando os parâmetros acima, a linha "[vendas]" corresponde ao nome do compartilhamento, já "path = /mnt/samba/vendas" é a localização do diretório no sistema operacional. O parâmetro "valid users = @"EMPRESA\vendas" @"EMPRESA\financeiro" se relaciona aos grupos que possuem permissão de acesso, "write list = @"EMPRESA\vendas" somente o grupo vendas terá permissão para escrever e modificar, "read list = @"EMPRESA\financeiro" somente o grupo financeiro terá permissão de leitura, os parâmetros (create mask = 660 e directory mask = 770) representam que todo arquivo criado no diretório será forçado a ter permissões para que somente o proprietário ou grupo proprietário possam criar e editar arquivos. O parâmetro "force group = 'EMPRESA\vendas", força todos os usuários na lista de permissão (valid users = @"EMPRESA\vendas" @"EMPRESA\financeiro") a ter como grupo primário o grupo @"EMPRESA\vendas no sistema de arquivos Debian.

Para cada compartilhamento, deve-se aplicar as mesmas medidas. Observe abaixo a exibição do resultado do compartilhamento:

Figura 2 – Compartilhamentos Linux Ubuntu.



Fonte: Próprio autor (2023).

O sistema de permissões Linux é muito seguro e atende bem as necessidades das corporações que não precisam delegar permissões diferentes ou iguais para várias contas e grupos de usuários no servidor de arquivos Samba. Mas, pode existir a necessidade de delegar permissões para mais de um usuário ou grupo com permissões iguais ou diferentes. De acordo com site oficial Samba (2020), nesse cenário as permissões podem ser implementadas com o suporte a ACLs (Access Control List).

O Samba trabalha com *listas de controle de acesso* **POSIX**, onde, o samba irá usar as permissões definidas diretamente no sistema de arquivos, gerenciadas pelos comandos getfacl (exibe ACLs) e setfacl (define e modifica ACLs), utilizados neste artigo. E com *listas de controle de acesso* **ESTENDIDAS**, estas, são gerenciadas diretamente de clientes com o sistema operacional Windows, usando o console de gerenciamento do computador, no item "pastas compartilhadas", de acordo com site oficial (SAMBA, 2020).

O passo inicial para iniciar o compartilhamento de arquivos utilizando ACLs é habilitar do sistema de arquivo, mas antes é importante verificar sua habilitação com o comando "tune2fs –l .dev/md0 | grep –i option", caso não esteja, deve-se habilitar com o comando "tune2fs –o .dev/md0" e "mount –o remount, acl /dev/md0". Logo em seguida, adicionar as linhas em "smb.conf".

Exemplo de compartilhamento com ACLs arquivo *"smb.conf"*:

```
[financeiro]
    path = /mnt/samba/financeiro
    valid users = @"EMPRESA\vendas" @"EMPRESA\financeiro"
    write list = @"EMPRESA\financeiro"
    read list = @"EMPRESA\vendas"
    create mask = 660
    directory mask = 770
```

Foi necessário criar o diretório financeiro dentro da partição "/mnt/samba" por meio do comando "mkdir -v -m 700 /mnt/samba/financeiro", configurando a permissão para que o grupo e outros usuários não tenham acesso com o comando "chmod -v g=,o= /mnt/samba/financeiro", aplicando inclusive as permissões ACLs com o comando "setfacl -R -m g:"EMPRESA\financeiro":rwx,g:"EMPRESA\vendas":rx /mnt/samba/financeiro". O parâmetro (-R) aplica recursivamente as permissões informadas nos diretórios e subdiretórios, o parâmetro (-m) as modificações de ACLs, (g:"EMPRESA\financeiro") representa o grupo, (:rwx) são as permissões de "leitura, gravação e execução" concedidas. No entanto, para que as permissões sejam herdadas como padrão nos diretórios e subdiretórios devemos executar o comando (setfacl -R -m g:"EMPRESA\financeiro":rwx,g:"EMPRESA\vendas":rx

/mnt/samba/financeiro), adicionando o parâmetro (-d) junto com modificações de ACLs.

Uma vantagem no compartilhamento de arquivos Samba está na triagem de arquivos, adicionando o parâmetro (veto files =) no compartilhamento desejado do smb.conf. O administrador do servidor bloqueia as extensões não desejadas obrigando o usuário do servidor de arquivos a armazenar somente o permitido, com isso, economiza espaço em disco e ganha segurança no compartilhamento ao bloquear arquivos grandes e arquivos executáveis (.iso, .mp4, .exe):

```
[financeiro]
    path = /mnt/samba/financeiro
    valid users = @"EMPRESA\vendas" @"EMPRESA\financeiro"
    write list = @"EMPRESA\financeiro"
    read list = @"EMPRESA\vendas"
    create mask = 660
    directory mask = 770
    veto files = /*exe/*mp4/*mp3/*iso/*msi/*avi/*dll/*bat
```

O modulo “vfs objects = recycle”, mais conhecido como lixeira do servidor Samba, é um recurso de segurança importantíssimo no compartilhamento de arquivos em rede, imagina excluir acidentalmente notas fiscais, recibos de pagamentos e de vendas ou copias de contracheque sem esse recurso ativo, com certeza daria um enorme prejuízo financeiro e dor de cabeça para empresa. Felizmente, este recurso pode ser habilitado no modulo global do Samba de forma que sua configuração atinja todos os compartilhamentos do servidor de arquivo em “smb.conf” ou pode ser habilitado como pasta compartilhada, para este artigo será usado o modulo global. Observe o exemplo:

```
vfs objects = recycle

; Configuração do modulo recycle ;
recycle:repository = .Lixeira/%U
recycle:directory_mode = 770
recycle:keeptree = yes
recycle:versions = yes
recycle:exclude = *.mp4, *.exe, *.msi, *.bat, *.log
```

Explicando os parâmetros: (vfs objects = recycle) ativa a função lixeira do servidor, (recycle:repository = .Lixeira/%U) local onde será salvo os arquivos excluídos, neste caso, o arquivo excluído será salvo em modo oculto no diretório raiz da pasta compartilhada na qual foi excluído. Dentro da lixeira terá o diretório especifico de cada usuário e nelas os arquivos excluídos de cada um. Assim, quando um arquivo é excluído, poderá ser recuperado caso necessário e sem a necessidade de sair do compartilhamento. Entretanto, é preciso habilitar arquivo oculto no sistema operacional dos usuários.

De acordo com Costa (2011), o Samba permite registrar ações de usuários e possíveis mensagens de erros do sistema, esses registros costumam ser chamados de log no Linux e fica armazenado no diretório /var/log. Para ativar basta adicionar as linhas abaixo na seção [global] do smb.conf:

```
log file = /var/log/samba/fileserver.log
max log size = 2048
log level = 2
```

O parâmetro (log file = /var/log/samba/fileserver.log): indica o caminho e o arquivo onde ele será gerado. O parâmetro (log level = 2): mostra o nível das mensagens, o nível 0 mostra apenas mensagens críticas do sistema, o nível 1 mostra detalhes dos acessos e de 2 a 10 mostra diversos níveis de informações uteis para desenvolvedores. O parâmetro (max log size =): indica o tamanho máximo do arquivo em kbytes. Quando o arquivo chega ao tamanho máximo é gerado um novo arquivo e o antigo é renomeado para manter o histórico de informações. A auditoria com modulo foi ativada com o comando “vfs objects = full_audit”. Caso este esteja ativo por conta da lixeira, devem ser listados no mesmo parâmetro, como o exemplo abaixo:

```
vfs objects = recycle full_audit

full_audit:success = write, unlink, rename, mkdir, rmdir, chmod, chown
full_audit:failure = write, unlink, rename, mkdir, rmdir, chmod, chown
full_audit:facility = local7
full_audit:priority = alert
full_audit:prefix = %I|%S|%U
```

O parâmetro (vfs objects = full_audit): ativa a função de auditoria. O parâmetro (full_audit:success =): uma lista de todas as operações que devem ser gravadas no log, caso sejam realizadas com sucesso. O parâmetro (full_audit:failure =): uma lista de todas as operações que devem ser gravadas no log, caso falhem. O parâmetro (full_audit:prefix =): define as informações que serão gravadas no log, por padrão o Samba utiliza os valores %U/%I. O parâmetro (full_audit:facility = e full_audit:priority =): definem o nível de prioridade dos alertas entre os suportados pelo syslog.

O parâmetro (full_audit:success = write, unlink, rename, mkdir, rmdir, chmod, chown): Fara com que todas as ações que incluem open (ler arquivo), opendir (ver arquivo dentro de uma pasta), write (alterar arquivo), unlink (apagar arquivo), rename (renomear arquivo), makdir (criar diretório), rmdir (remover diretório), chmod (alterar permissão de acesso de um arquivo) e chown (mudar dono de um arquivo) sejam gravadas no log em caso de sucesso na ação. O parâmetro (full_audit:failure = write, unlink, rename, mkdir, rmdir, chmod, chown): é o inverso

de “full_audit:success”. O parâmetro (full_audit:prefix = %I|%S|%U): São as informações que serão armazenadas, existem outras variáveis, entretanto, os padrões do Samba: %I (IP da máquina), %S (nome do compartilhamento onde foi feito o acesso ou alteração) e %U (nome do usuário). O parâmetro (full_audit:facility = local7 e full_audit:priority = alert): apenas especificam o nível dos alertas, entre os suportados pelo syslog.

```
Sep 27 19:26:37 pvh-srv-file01 smbd_audit:
192.168.56.1|financeiro|israel|rename|ok|/mnt/samba/financeiro/Nova
pasta|/mnt/samba/financeiro/Israel
Sep 27 19:31:42 pvh-srv-file01 smbd_audit:
192.168.56.1|financeiro|israel|rename|ok|/mnt/samba/financeiro/Israel/Não confirmado
398457.crdownload|/mnt/samba/financeiro/.Lixeira/israel/Israel/Não confirmado 398457.crdownload
Nov  2 14:51:13 pvh-srv-file01 smbd_audit:
192.168.100.20|diretoria|bruno|rename|ok|/mnt/samba/diretoria/Documentos/Comandos
CMD.txt|/mnt/samba/diretoria/.Lixeira/bruno/Documentos/Comandos CMD.txt
```

O segundo passo realizar configurações no arquivo de configuração do syslog. Criar novo arquivo dentro de “/etc/rsyslog.d/” chamado “samba-full-audit.conf“. Neste arquivo vamos informar onde iremos armazenar nossas informações de auditoria. Também devemos dar permissão de escrita ao grupo proprietário do diretório /var/log/samba, com comando (chmod -v g+w /var/log/samba), para finalizar, reiniciar o serviço do rsyslog (systemctl restart rsyslog).

Importante destacar que os diretórios compartilhados foram implementados na partição (/mnt/samba/), criada especialmente para função de compartilhamento no sistema Linux Debian com (RAID 5), a triagem de arquivos está configurada somente para esta partição, podendo ser auditado e analisado o arquivo *full.audit.log* gerado no diretório (/var/log/samba/), cada linha do arquivo corresponde a uma modificação ou tentativa realizada pelos usuários devidamente autenticados no AD do Samba, as informações contidas no arquivo full.audit.log foram definidas pelas variáveis no arquivo smb.conf (full_audit:prefix = %I|%S|%U), existem outras diversas variáveis que podem ser adicionadas e gerar mais informação ao arquivo full.audit.log. Porém, foi decidido aplicar as variáveis básicas. Veja o exemplo do comando para verificar arquivo “full_audit.log” em tempo real.

```
tail  -f /var/log/samba/full-audit.log
```

Portanto, após a realização dos procedimentos técnicos, constata-se que o Samba pode servir como um poderoso servidor de arquivos, possibilitando, por simples linhas de comando ou edição de seus arquivos de configuração, a realização de compartilhamento de arquivos para rede Windows e Linux, controlar acesso para usuários ou grupos com diferentes permissões, além disso, possibilita criar lixeira em rede conjugado com auditoria de segurança, aliado ao

veto files, o armazenamento fica obrigatório ao tipo de arquivo programado pelo administrador.

5. CONSIDERAÇÕES FINAIS

Os resultados obtidos com a implementação da ferramenta Samba foram positivos, somente usuários autorizados conseguiram acessar os compartilhamentos, atingindo os conceitos básicos de segurança da informação: Confidencialidade, Integridade e Disponibilidade. Não ouve gastos ou problemas com licença de softwares e não exigiu hardware com alto nível de processamento de dados para obter bom funcionamento. Contudo, é opcional usar software Active Directory (AD) em conjunto com o servidor de arquivos Samba, mas, é interessante utilizar, porque traz vantagens com autenticação de usuários e controle de acesso aos recursos de rede, que tende facilitar a tomada de decisão na *auditoria*, identificando e permitindo responsabilizar os envolvidos em caso de quebra de políticas de segurança internas das corporações. O ponto negativo está na interpretação dos arquivos coletados, que variam de acordo com variáveis adicionadas na lixeira, cabe ao administrador selecionar apenas as necessárias para não sobrecarregar o arquivo *full_audit.log* e posteriormente poder traduzir as informações coletas corretamente.

Diante da rede simulada foi possível compartilhar arquivos para todos os usuários pertencentes ao Active Directory, inclusive para computadores da rede Linux. Existem diversas opções de configuração para compartilhamento no arquivo smb.conf, possibilitando ao administrador implementar a mais compatível com sua realidade, controlando acesso para usuários, grupos e rede. Foi possível verificar diversos benefícios que ampliam a segurança no compartilhamento de arquivos, através do "*veto files*", por exemplo, pôde-se filtrar o tipo de arquivo a ser armazenado, garantindo que o diretório não seja utilizado para outros fins, protegendo a disponibilidade e a integridade do servidor de arquivo. Além disso, o administrador pode habilitar a lixeira do Samba em modo *recycle full_audit* que protege as exclusões acidentais ou não acidentais, movendo os arquivos para a lixeira, que podem ser recuperados caso necessário. Não apenas isso, este processo armazena todas as modificações feitas nos diretórios compartilhados que estão na partição *(/mnt/samba)* definido como triagem de arquivos para /var/log/samba/full-audit.log.

A auditoria não irá impedir que haja modificação ou exclusão dos arquivos, porém possibilita a identificação do usuário através das informações coletadas no "*full_audit.log*" que possibilita o administrador tomar medidas que o caso requer. Entre todas as vantagens citadas, não encontrei jeito de colocar a data completa nas linhas contidas no arquivo full_audit.log,

onde aparece o mês e o dia. Talvez falte alguma variável a ser implementada ou não, fica a dica a quem tem interesse em estudar o caso.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT. Associação Brasileira de Normas Técnicas**. ISO/IEC 27001 - Tecnologia da informação - Técnicas de Segurança - Sistemas de gestão de segurança da informação – Requisitos**. 1º Ed. Rio de Janeiro: ABNT, 2006.

BRITO, Samuel H. B. **Serviços de Redes em Servidores Linux**. 1.ed. São Paulo: Novatec Editora Ltda, 2017.

CARTER, G.; JAY T.; ECKSTEIN, R. **Using Samba**. 3. ed. Estados Unidos da América: O'Reilly Media, 2007.

CERVO, A. Luiz; BERVIAN, P. Alcino; SILVA, Roberto. **Metodologia científica**. 6. ed. São Paulo: Pearson Prentice Hall, 2007.

COSTA, Paulo H. A. **SAMBA: Windows e Linux em rede**. 2.ed. São Paulo: Novatec Editora Ltda, 2011.

FSF. FREE SOFTWARE FUNDATION. **O que é software livre?** Disponível em: https://www.gnu.org/philosophy/free-sw.html#header. Acesso em: 14 de jun. 23.

GAIDARGI, J. **Escolhendo um servidor de arquivos para sua empresa**. Disponível em: https://www.infonova.com.br/artigo/servidor-de-arquivos/. Acesso em 05 de nov. 2018.

KIM, D.; SOLOMON, M. G. **Fundamentos de segurança em sistemas de informação**. Tradução Daniel Vieira; Rio de Janeiro: LTC, 2014.

MICROSOFT. **Licenças de acesso para cliente e Licenças de gerenciamento**. Disponível em: https://www.microsoft.com/pt-br/licensing/product-licensing/client-access-license. Acesso em 24 jun. 19.

MICROSOFT. **SMB e servidor de arquivos**. Disponível em: https://learn.microsoft.com/pt-br/windows-server/storage/file-server/file-server-smb-overview. Acesso em 01 dez. 23.

NACIONAL, Sebrae: **O que é software livre e quais as vantagens em usá-lo na sua empresa**. https://sebrae.com.br/sites/PortalSebrae/artigos/o-que-e-software-livre-e-quais-as-vantagens-em-usa-lo-na-sua-empresa,2928d53342603410VgnVCM100000b272010aRCRD. Acesso em: 01 dez. 2023.

SAMBA, WIKI. **Configurando um compartilhamento usando ACLs POSIX.** Disponível em: https://wiki.samba.org/index.php/Setting_up_a_Share_Using_POSIX_ACLsAcesso em 01 de dez. 2023.

SÊMOLA, Marcos. **Gestão da Segurança da Informação**: Uma visão executiva. 2. ed. Rio de Janeiro: Elsevier, 2014.

STALLINGS, W.; BROWN, L. **Segurança de computadores**: Princípios e práticas. 2. ed. Rio de Janeiro: Elsevier, 2014.