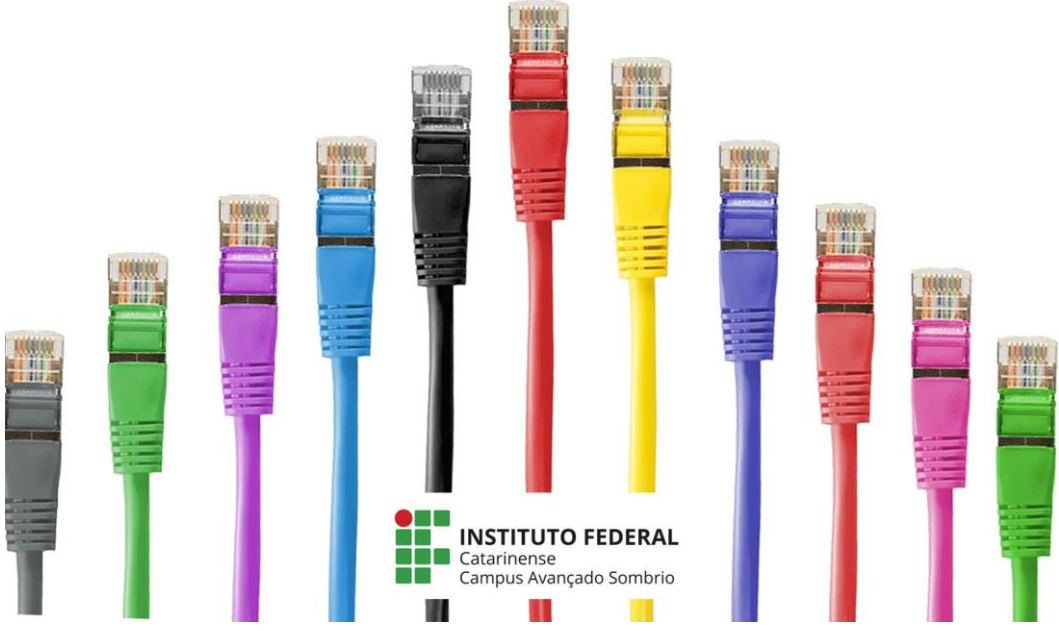
Vanderlei Freitas Junior
Gabriel Cesar Costa
Vinícius dos Santos Fernandes
Organizadores

TECNOLOGIA E REDES DE COMPUTADORES

3ª Edição



INSTITUTO FEDERAL
Catarinense
Campus Avançado Sombrio

Vanderlei Freitas Junior
Gabriel Cesar Costa
Vinícius dos Santos Fernandes
Organizadores

TECNOLOGIA E REDES DE COMPUTADORES

3ª Edição

2017
Instituto Federal Catarinense

Associação Brasileira
das Editoras Universitárias

Direção Editorial	Vanderlei Freitas Junior
Capa e Projeto Gráfico	Vanderlei Freitas Junior
Editoração Eletrônica	Gabriel Cesar Costa
	Vinícius dos Santos Fernandes
Comitê Editorial	Armando Mendes Neto
	Cleber Luiz Damin Ferro
	Gilnei Magnus dos Santos
	Guilherme Klein da Silva Bitencourt
	Jéferson Mendonça de Limas
	Joédio Borges Junior
	Lucyene Lopes da Silva Todesco Nunes
	Marcos Henrique de Morais Golinelli
	Matheus Lorenzato Braga
	Sandra Vieira
	Vanderlei Freitas Junior
	Victor Martins de Sousa
Revisão	Gilnei Magnus dos Santos
Organizadores	Vanderlei Freitas Junior
	Gabriel Cesar Costa
	Vinícius dos Santos Fernandes

Imagens: http://www.morguefile.com/

ISBN 978-855644013-6

Dados Internacionais de Catalogação na Publicação (CIP)

Tecnologias e Redes de Computadores / Vanderlei Freitas Junior, Gabriel Cesar Costa, Vinícius dos Santos Fernandes (organizadores).
– Sombrio: Instituto Federal Catarinense, 2017. 3. ed.

Diversos autores.
90 f.: il. color.

ISBN: 978-85-5644-013-6

1. Redes de Computadores. 2. Tecnologia da Informação e Comunicação. I. Título. II. Freitas Junior, Vanderlei. III. Costa, Gabriel Cesar. IV. Fernandes, Vinícius dos Santos.

CDD 004.6

Esta é uma publicação do
Curso Superior de

Tecnologia em

REDES DE COMPUTADORES

Sumário

Autenticação de usuários em redes *wireless* utilizando protocolo 802.1x, servidor *radius* e base de dados ldap

Tadeu Germano Porto[1], Mateus Souza Trajano[2], Marcos Henrique de Morais Golinelli[3], Tatiana Marcela Rotta[4]

[1234]Curso de Tecnologia em Redes de Computadores - Instituto Federal Catarinense (IFC) – *Campus* Avançado Sombrio

88960-000 – Sombrio – SC - Brasil

tadeugermanoporto@hotmail.com,
mateussouzatrajano@gmail.com,
marcos.golinelli@sombrio.ifc.edu.br,
tatiana.rotta@sombrio.ifc.edu.br

***Abstract**: The security of a network is extremely important these days. Based on this precept, the purpose of this study was the authentication of users of wireless networks with authorization via RADIUS server using the LDAP directory service. In the study, the bibliographic and exploratory research for the development was adopted. Authentication via the RADIUS server in conjunction with the 802.1x protocol becomes a key action to obtain security and performance in network management, since it is pertinent to choose such protocol and service to access users with authentication to the network services. From the experiment, you can perform the authentication of the registered users successfully.*

***Resumo:** A segurança de uma rede é de extrema importância nos dias atuais. Com base nesse preceito, objetivou-se neste estudo a autenticação de usuários de redes wireless com autorização via servidor RADIUS utilizando-se de serviço de diretórios LDAP. No estudo, adota-se a pesquisa bibliográfica e exploratória para o desenvolvimento. A autenticação via servidor RADIUS em conjunto do protocolo 802.1x torna-se ação chave para obter-se segurança e desempenho em gerenciamento de rede, visto que é pertinente a escolha de tal protocolo e serviço ao acesso de usuários com autenticação aos serviços de rede. A partir do experimento, pode-se realizar a autenticação dos usuários cadastrados com êxito.*

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Palavras-Chave: *Autorização; Autenticação; LDAP; 802.1x; RADIUS.*

1. Introdução

Ao deparar-se com o desenvolvimento tecnológico pode-se notar a gama de equipamentos distribuídos entre diversas redes e finalidades. Cada dispositivo que se conecta livremente sem controle em uma rede pode ocasionar ou permitir que brechas de segurança sejam descobertas e atacadas. Obter acesso e privilégios sem autorização ao conectar-se, a uma rede wireless tem sido uma grande preocupação em gerência de redes. Em relação a segurança, define-se diversas funções como: comunicação e tráfego de informações sigilosas, serviços exclusivos para usuários privilegiados e acesso para usuários comuns.

Administrar as redes de maneira adequada para utilização torna-se indispensável, com isso há a necessidade de ferramentas que auxiliem a realizar o gerenciamento da rede, já que se tornou inviável executá-lo de maneira individual para cada conexão wireless. Para tais feitos são necessárias ferramentas e técnicas para maximizar a eficiência de um gerenciamento efetivo de acesso à rede.

Neste trabalho, será abordado como objetivo geral o conceito de autenticação em redes wireless, utilizando um servidor *Remote Authentication Dial In User Service* (*RADIUS*), que consultará uma base de dados de serviços de diretório *LDAP*, ambos com aplicações/aplicativos *open source*. Assim sendo, implementou-se o serviço de autenticação através de protocolo 802.1x com aplicação *FreeRADIUS* em conjunto com serviço de diretório *OpenLDAP* (SANTOS, 2011).

Como objetivos específicos, busca-se implementar serviços de autenticação utilizando software livre *FreeRADIUS* em conjunto com 802.1x, implementar e substituir o serviço de diretório com software livre *OpenLDAP e PHPldapadmin* para controle de interface gráfica, configurar o *FreeRADIUS* para utilizar os dados dos usuários do *OpenLDAP* ao efetuar autenticação.

Na seção seguinte abordaremos os protocolos utilizados, a seção 3 aborda os materiais e métodos a serem utilizados, seção 4 a aplicação e desenvolvimento, seção 5 os resultados obtidos, seção 6 trabalhos futuros e na seção 7 as considerações finais seguidas das referências.

2. Referencial Bibliográfico

Nesta seção, aborda-se a parte teórica do trabalho a fim de embasar a qualidade científica, tecnológica e experimental, utilizando-se de ferramentas e técnicas para fins de maximizar a eficiência do gerenciamento de acesso a redes wireless.

2.1 Protocolo 802.11

O Institute of Electrical and Electronic Engineers (IEEE) é uma organização sem fins lucrativos, a qual se dedica ao avanço da tecnologia visando beneficiar a humanidade, que em 1999, definiu uma norma para redes locais sem fio chamada

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

"Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". O padrão IEEE 802.11, como todos os protocolos da família 802.x, especificam as camadas física e de controle de acesso ao meio como meio cabos e redes sem fio. O padrão 802.11 se divide em duas topologias, sendo elas: ad hoc e infra estruturadas. Na topologia ad-hoc os dispositivos se comunicam diretamente sem que haja a necessidade de um dispositivo intermediando a conexão, já na infra estruturada os dispositivos se comunicam apenas por intermédio de um ponto de acesso, independente da distância entre eles (PAZ, et. al., 2015).

2.2 Protocolo 802.11i

Nas redes *wireless*, inicialmente, utilizava-se do protocolo de criptografia *Wired Equivalent Privacy* (WEP), porém tornou-se rapidamente conhecido como um método inseguro por possuir vulnerabilidade na confidencialidade e integridade da conexão, para tal resolução do problema, nasceu o *Wi-Fi Protected Access* (WPA). Neto (2004, p.15) afirma que o "WPA adotou o padrão 802.1X para resolver o problema de autenticação do protocolo WEP", esse, no entanto, aplica-se não só a rede wireless, mas também a cabeada. O padrão 802.1X surgiu para a resolução do problema com a falta de segurança em redes com fio, mas tornou-se eficaz sendo aplicada em redes sem fio. (NETO, 2004).

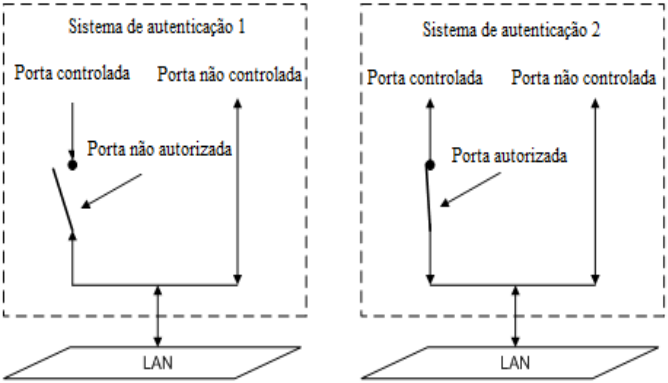
O 802.11i surgiu para corrigir os problemas de segurança conhecidos do protocolo WEP (NETO,2004). O *802.11i* é um padrão IEEE, destinado a proporcionar maior segurança na Camada MAC para o 802.11. A especificação *802.11i* define duas classes de Algoritmos de segurança: *Robust Security Network Association* (RSNA) e *Pré-RSNA* (HE; MITCHELL, 2016). A segurança pré-*RSNA* consiste em *WEP* e 802.11 como autenticação. Por outro lado, a RSNA fornece dois dados de confidencialidade de Protocolos, chamado de *Temporal Key Integrity Protocol* (TKIP) e o protocolo *Counter-mode* / CBC-MAC (CCMP). O procedimento de estabelecimento de *RSNA* inclui autenticação 802.1X e gerenciamento de chaves para os protocolos. Analisa-se os aspectos de segurança do 802.11i, tais como, especificação, considerando a confidencialidade dos dados, integridade, autenticação mútua e disponibilidade projetado a melhorar a segurança das redes sem fio. (HE; MITCHELL, 2016).

2.2 Protocolo 802.1x

O protocolo IEEE 802.1x especifica um mecanismo para autenticação de dispositivos de rede e/ou usuários baseado em portas (a porta do dispositivo só permite o envio e recepção de dados após a autenticação, este mecanismo pode ser utilizado em redes cabeadas e em redes *wireless*) como demonstrado na figura 01, através da utilização de variações do protocolo *Extensible Authentication Protocol* (EAP). O protocolo EAP, na sua definição, permite a utilização de uma grande variedade de mecanismos de autenticação (PERES; WEBER, 2003).

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Figura 01: Autenticação baseada em portas.



Fonte: Disponível em: https://goo.gl/abz45i, 2016.

Acesso em: 15 nov 2016.

2.3 Serviços de Autenticação

A autenticação nos processos é fundamental dentro de um sistema distribuído, onde usuários necessitam provar sua identidade a fim de ter acesso aos recursos que um aplicativo ou serviço possa oferecer (DIÓGENES; MAUSER, 2013). "Os protocolos de autenticação podem ser categorizados para uso de acesso remoto, como *RADIUS* e acesso à rede, normalmente utilizado por sistemas operacionais de rede, como *Active Directory*, na utilização do protocolo *Kerberos*" (DIÓGENES; MAUSER, 2013, p. 198).

O acesso remoto, com autenticação, é um dos processos principais para o acesso à serviços como *Facebook*, *Twitter* e *Hotmail*, em que se precisa apresentar credenciais para acesso aos recursos disponíveis (DIÓGENES; MAUSER, 2013). Serviços de autenticação normalmente possuem um ou mais servidores que armazenam as credenciais de seus usuários, sendo estes servidores AAA (*Autentication, Autorization and Account*) ou IAAA (*Identification, Autentication, Autorization e Acount*). A norma (ABNT ISO IEC 27002, 2005) sugere que AAA, sendo seu significado Autenticação, Autorização e Auditoria. (DIÓGENES; MAUSER, 2013).

2.4 Segurança de Rede

As transações entre o cliente ou ponto de acesso, realizadas com o servidor *RADIUS*, são autenticadas por meio da utilização de uma chave de criptografia compartilhada, que será transmitida em meio criptografado. Nas configurações convencionais de redes sem fio, a chave criptografada é compartilhada entre os usuários, qualquer novo dispositivo que se conecte a rede, necessita digitar tal chave, utilizando protocolo 802.1x de controle de acesso à rede, as chaves criptográficas são geradas de forma dinâmica entre o usuário e o ponto de acesso, não sendo necessário o conhecimento da chave pelos usuários. Além disso, todas as senhas dos usuários são

enviadas criptografadas entre o cliente e servidor, para eliminar a possibilidade de espionagem em uma rede não segura, evitando uma possível descoberta da senha do usuário. (RIGNEY, et. al., 2010).

2.5 RADIUS

RADIUS é um protocolo de rede, um sistema que define regras e convenções para a comunicação entre dispositivos de rede para o usuário remoto (SARL, 2014). É um protocolo utilizado para gerenciar e autenticar o acesso de usuários a diversos serviços de rede. O protocolo define um padrão para ser utilizado na troca de informações entre um cliente e servidor de autenticação, autorização e auditoria de contas (AAA). (SANTOS, 2011). A autenticação é o processo de confirmar a identidade do usuário dentro de uma rede. Normalmente ocorre entre um cliente e um servidor. A autenticação ocorre através da apresentação de uma identidade e as credenciais correspondentes. Por fim o protocolo *RADIUS* se tornou comumente usados pelos *Provedores Serviços de Internet* (*ISPs*), provedores de rede móvel, redes empresariais e educacionais. Dentre suas funções, as três principais são:

- ➢ Autenticar os usuários ou dispositivos antes de permitir-lhes o acesso a uma rede;
- ➢ Autorizar os utilizadores ou dispositivos para serviços de rede específicos;
- ➢ Contabilizar o uso desses serviços.

A partir da criação do *RADIUS*, sistemas de acesso à rede foram distribuídos em uma grande área e foram executados por várias organizações independentes. Administradores de redes desejavam evitar problemas com segurança e escalabilidade, e, portanto, não desejavam distribuir nomes de usuário e senhas, em vez disso, queriam que os servidores de acesso remoto ao entrar em contato com um servidor central, precisassem autorizar o acesso ao sistema solicitado ao serviço (SARL, 2014). Em resposta ao contato do servidor de acesso remoto, o servidor central teria um retorno mensagem “falha” ou “sucesso”, e as máquinas remotas seriam encarregadas de fazer cumprir esta resposta para cada usuário final.

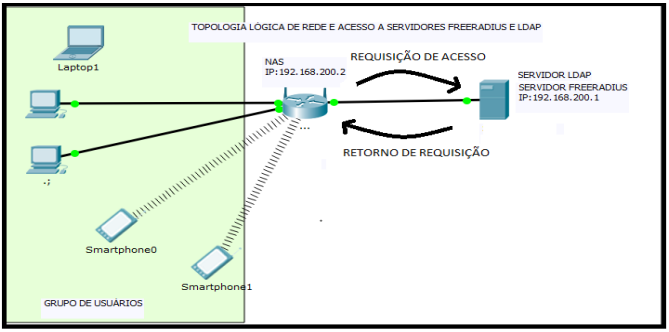
O objetivo do *RADIUS* foi, por conseguinte, criar um local central para a autenticação do utilizador, em que os utilizadores de muitos locais pudessem solicitar o acesso à rede. A simplicidade, eficiência e usabilidade do sistema *RADIUS* levou à sua adoção generalizada pela rede de fornecedores de equipamentos, na medida em que, atualmente, o protocolo *RADIUS* é considerado um padrão da indústria e também é posicionado a se tornar um padrão pela *Internet Engineering Task Force* (IETF). (RIGNEY, et. al., 2010).

O servidor *RADIUS* é, normalmente, um software em execução em uma rede ou servidor independente. O servidor aguarda uma solicitação de um *Network Access Server (NAS*), (que é um servidor entre o usuário final e o servidor *RADIUS*, que recebe a requisição de acesso à internet e envia ao RADIUS, se o usuário tem ou não suas credenciais cadastradas no serviço), o servidor processa os dados e, em seguida, retorna

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

uma resposta ao *NAS*. A resposta pode conter políticas de autorização, ou um reconhecimento dos dados contabilísticos recebidos, ou a não autorização do serviço como demonstra a figura 02 (DIÓGENES; MAUSER, 2013). Um único servidor pode receber e processar muitas solicitações de acesso simultâneas de vários tipos de *NASs* (tal como ADSL, dial-up, ou concentradores de VPN) em muitos locais diferentes (RIGNEY, et. al., 2010). Ao mesmo tempo o servidor pode também interagir com arquivos simples, bancos de dados e diretórios *LDAP* que será falado posteriormente ou outros servidores *RADIUS*. (SARL, 2014).

Figura 02: Autenticação via servidor RADIUS.



Fonte: Autores, 2016.

Uma vez que o servidor toma uma decisão, ele retorna uma resposta ao *NAS* que aplica a política de resposta, ou pode ignorá-lo completamente. O servidor não recebe uma mensagem de resposta do *NAS*, ou se o *NAS* está obedecendo às instruções da resposta. O servidor *RADIUS* não tem controle sobre o que o *NAS* faz com uma resposta. (RIGNEY, et. al., 2010).

Uma ampla e variada gama de empresas, atualmente, utiliza o protocolo *RADIUS* para autenticação e necessidades de contabilidade. (DIÓGENES; MAUSER, 2013). O protocolo cliente-servidor *RADIUS* contém muitas vantagens tecnológicas para os clientes, incluindo uma solução aberta e escalável; amplo suporte de uma grande base de fornecedores; fácil modificação; separação de processos de segurança e comunicação; adaptável à maioria dos sistemas de segurança e viável com qualquer dispositivo de comunicação que suporta o protocolo cliente.

2.5.1 FreeRADIUS

Em 1991, Merit Network, um provedor de internet sem fins lucrativos, buscava uma forma para gerenciar o acesso dial-in para vários *Points of presence* (POP) em toda a sua rede. Em resposta a esta necessidade, foi criado o *RADIUS* (SARL, 2014). O *FreeRADIUS* é o mais popular servidor *RADIUS* de código aberto utilizado pela sociedade académica e no mundo. Ele serve como base para várias ofertas comerciais, e fornece a autenticação, autorização e contabilidade (AAA) de muitas empresas. Foi criado em agosto de 1999, por Alan DeKok e Miquel van Smoorenburg, e foi desenvolvido usando um design modular, para incentivar o envolvimento mais ativo da comunidade acadêmica. (SARL, 2014).

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

A arquitetura *RADIUS* cliente-servidor fornece uma solução aberta e escalável que está amplamente amparada em *Request for Comments (RFC).* Ela pode ser facilmente modificada para satisfazer uma variedade de situações. Os clientes podem modificar o servidor baseado nos servidores de autenticação para trabalhar com um grande número de sistemas de segurança existentes no mercado. Os mecanismos de autenticação flexíveis, inerente ao servidor facilitam a sua integração com sistemas existentes e legados, quando necessário. Outra vantagem da arquitetura é que qualquer componente de um sistema de segurança, que suporta os protocolos, pode utilizar autenticação e autorização do servidor *RADIUS* central. (SARL, 2014).

Como alternativa, o servidor central pode integrar-se com um mecanismo de autenticação separado. A utilidade do protocolo estende-se para além desses sistemas que utilizam dispositivos e servidores de acesso à rede. O protocolo *RADIUS* tem sido amplamente aceito pelos ISPs para fornecer serviços de rede privada virtual (VPN). Neste contexto, sua tecnologia permite que uma organização possa usar a infraestrutura ISP para as comunicações de forma segura. (RIGNEY, et. al., 2010)

A popularidade desse servidor pode ser atribuída ao grande número de benefícios adicionais que ele oferece, muito além dos encontrados na variedade de outros servidores *RADIUS*. Essa ferramenta é baseada em uma característica modular, design escalável, que oferece benefícios e vantagens para os administradores de rede, pois mais tipos de autenticação são suportados pelo *FreeRADIUS* do que por qualquer outro servidor de acesso de código aberto. (SARL, 2014).

O uso de servidores Virtuais significa que implementações complexas terão custos simplificados e baixo custo de manutenção e suporte à rede. Assim, a capacidade do servidor para suportar servidores virtuais, lhe fornece uma enorme vantagem sobre a concorrência. (SARL, 2014).

Enquanto muitos servidores comerciais oferecem diferentes versões de seus softwares para lidar com diferentes necessidades, uma única versão do *FreeRADIUS* é necessária para obter um melhor desempenho, sem necessidade de adquirir licenças adicionais do produto. (RIGNEY, et. al., 2010)

2.6 LDAP

Lightweight Directory Access Protocol (*LDAP*) é um protocolo que permite efetuar o armazenamento de informações em um serviço de diretórios, sendo baseado no protocolo X.500 (HOWES, 1998). Atua de maneira que permite organizar a rede de forma hierárquica, e que possui um diretório principal denominado raiz e subdiretórios, sendo que cada um possui suas peculiaridades e permissões diferenciadas. (ZEILENGA, 2006 A).

O *LDAP* é especificado em uma série de publicações de padronização definidas pelo *Internet Engineering Task Force* (*IETF*) chamadas RFC. A última especificação é a Versão 3, publicada como RFC 4511 (SERMERSHEIM, 2006). Podem-se encontrar diversas aplicações que utilizam o protocolo *LDAP*, sendo que uma das mais conhecidas e difundidas no mercado é a *OpenLDAP*, uma ferramenta *open source*, disponibilizada no site: http://www.openldap.org.

Apesar de utilizar um sistema hierárquico que lembra armazenamento de

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

arquivos em unidades de disco rígido, o diretório *LDAP* é uma base desenvolvida para leitura, diferente de banco de dados relacional, que tende a ser atualizado na proporção em que é lido. (PEREIRA, 2009; SANTOS, 2011).

O protocolo *LDAP* foi desenvolvido em 1993, pela Universidade do Michigan e tinha como objetivo substituir o *Directory Access Protocol* (*DAP*), que era utilizado no modelo OSI. Com a evolução do *LDAP*, notou-se a falta de suporte ao X.500 e a pilha de protocolos *Open Systems Interconnection* (OSI), assim formando uma barreira para a utilização do mesmo. Esse impasse levou pesquisadores e desenvolvedores da Universidade de Michigan a criar um servidor *LDAP standalone*, denominado *slapd* disponibilizando meios para divulgação e aperfeiçoamento do protocolo com auxílio de desenvolvedores de todo o mundo. Com o decorrer do tempo, o protocolo *LDAP* deixa de ser um utilitário resultante do DAP e X.500, para receber padronização IETF, deste modo, se tornando um serviço mundialmente difundido e utilizado para serviços de consulta em diretórios (ESKELSEN; LUCKMANN, 2006).

Por ser especializado na leitura de informações já existentes em sua base de dados, o *LDAP* se torna uma alternativa viável para suprir a necessidade de um banco de dados, podendo ser centralizado ou não, pois pode-se utilizar múltiplas aplicações e serviços em uma rede com apenas uma informação contida na base de dados. O que proporciona esta alternativa é a capacidade de se atribuir permissões diferenciadas a várias informações.

A utilidade do protocolo *LDAP* não se restringe apenas à atribuição de permissões diferenciadas, pode-se efetuar sua configuração para replicação, como é o caso do serviço *Domain Name System* **(***DNS***)**, podendo ser feito balanceamento de carga ao consultar uma informação na base de dados **(**PEREIRA, 2009; SANTOS, 2011).

A utilização do *LDAP* é seguida por alguns modelos básicos de informação, cada um com suas características, podendo distinguir os elementos que compõem modelos de dados, como:

- **Distinção de nomes** – especificados na RFC 4512 Zeilenga (2006b) afirma que cada processo de leitura é feito a partir de um *relative distinguished name* (RDN), o qual é o nome que está em um nível superior da hierarquia, um conjunto de RDN compõem um *distinguished name* (DN). Assim, ao se realizar uma consulta é utilizada uma DN para identificar a que setor está localizado a partir de um nome definido e, sequencialmente, sendo filtrado por RDNs (SANTOS, 2011).

- **Funcionalidades** – o LDAP possui nove funções básicas que o definem, sendo elas: adicionar, modificar, apagar, associar (*bind*), dissociar (*unbind*), buscar, comparar, renomear e abandonar (ESKELSEN; LUCKMANN, 2006).

Ao analisar o estudo das tecnologias, demanda-se uma elaboração de citações de diversos autores da área, a fim de aprofundar o conhecimento, tal necessidade traz o embasamento teórico para o desenvolvimento da pesquisa e escrita do artigo. (NUNES et. al., 2013).

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

3 Materiais e Métodos

Utilizando-se do pensamento de Gil (2010, p.17) “a pesquisa científica se inicia com a construção de um problema passível de solução mediante a utilização de métodos científicos”, parte-se em busca do objetivo do trabalho, a fim de aplicar com sucesso os mesmos.

No estudo, adota-se como métodos, conforme os objetivos do estudo, a pesquisa exploratória, também, conforme os procedimentos do estudo, a pesquisa bibliográfica. Utilizando-se nas elaborações consultas a artigos científicos, embasamento teórico e empírico e em base de dados capes, o conhecimento para a escrita do artigo (LAKATOS; MARCONI, 2003).

A elaboração do artigo inicia-se depois da coleta de dados bibliográficos, esse estudo aplica-se a partir da estruturação das ferramentas em ambiente virtual, com *Virtual Box* versão 5.1.2, *Windows 8.1*, servidor *Ubuntu Server 16.04 LTS* e base de dados *LDAP* versão 3 com a ferramenta *OpenLdap* versão 2.4.42. Na elaboração do plano deve-se observar a estrutura de todo o trabalho científico: introdução, desenvolvimento e conclusão (LAKATOS; MARCONI, 2003). A análise dos resultados será elaborada a partir dos relatórios provenientes da implementação das ferramentas virtuais utilizadas.

4 Aplicação e Desenvolvimento

Primeiramente, será implementado ferramentas em ambiente virtual utilizando a ferramenta Virtual Box. O resultado será apresentado ao final da implementação do experimento, utilizando as aplicações para embasar a parte teórica.

4.1 Implementação das Ferramentas

O estudo inicia-se com a configuração do Sistema Operacional e instalação das aplicações *FreeRADIUS* e *OpenLDAP* e *phpldapadmin* no Ubuntu 16.04 LTS e posteriormente a configuração do Access Point SIROCO EVO-W301AR.

A configuração inicial do servidor se dá pela seguinte ordem de comandos como demonstra-se no quadro 01:

Quadro 01: Comandos configuração inicial *FreeRADIUS*.

Comando	Descrição do comando
sudo su	Alterar para modo root
apt-get update	Atualizar a lista de pacotes de serviços
apt-get install freeradius-ldap	Instalar *FreeRADIUS*

Fonte: Autores, 2016.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Por padrão o *FreeRADIUS* possui apenas o localhost em suas configurações. É necessário adicionar os NAS que são os clientes que possuem permissão para realizar consultas ao *FreeRADIUS*. O cadastro dos NAS's é realizado no arquivo */etc/freeradius/clients.conf.*

Define-se uma segunda interface de rede em modo bridge com IP fixo (192.168.200.2/24) para efetuar a conexão com o equipamento.

Adicionam-se as seguintes linhas, no arquivo, demonstrado no quadro 02:

Quadro 02: Configuração do arquivo clients.conf.

```
client 192.168.200.2/24{
        ipaddr = 192.168.200.2
        secret = tpradius
        shortname = teste
        NAStype = other
        }
client localhost {
        ipaddr = 127.0.0.1
        secret = testing123 NAStype = other}
```

Fonte: Autores, 2016.

Para testar a funcionalidade do *FreeRADIUS*, pode-se criar um usuário no arquivo */etc/freeradius/users*, adiciona-se as seguintes linhas, conforme o quadro 03:

Quadro 03: Comando de inserção de usuário.

```
mateus Cleartext-Password := "123"
```

Fonte: Autores, 2016.

Para verificação inicial do funcionamento do servidor *RADIUS* pode-se utilizar o seguinte comando, omitindo as aspas duplas:

"radtest mateus 123 127.0.0.1 0 testing123"

Após a execução do comando será retornado os valores demonstrando que a autenticação foi realizada com sucesso, como exemplifica o quadro 04:

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Quadro 04: Resposta da solicitação de autenticação.

```
root@radius:/home/teste# radtest mateus 123 127.0.0.1 0 testing123

Sending Access-Request of id 217 to 127.0.0.1 port 1812

        User-Name = "mateus"

        User-Password = "123"

        NAS-IP-Address = 127.0.1.1

        NAS-Port = 0

        Message-Authenticator = 0x00000000000000000000000000000000

rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=217,
length=20
```

Fonte: Autores, 2016.

A instalação do *OpenLDAP* dar-se-á pelos seguintes comandos como apresenta no quadro 05:

Quadro 05: Comandos de instalação *OpenLDAP*.

Comando	Descrição do comando
apt-get install slapd ldap-utils	Instala o *OpenLDAP* e seus pacotes de utilitários.
apt-get install phpldapadmin	Instala a interface gráfica para acessar o *OpenLDAP*.
dpkg-reconfigure slapd	Reconfigurar a base de dados do *OpenLDAP*.

Fonte: Autores, 2016.

Após o comando de reconfiguração da base de dados será necessário fornecer algumas informações, estas sendo necessárias para concluir a reconfiguração. Ao começar a reconfiguração a aplicação pergunta se o instalador deverá omitir a configuração do *OpenLDAP*, deve-se selecionar a opção *<no>*, em seguida deve-se definir um nome de domínio para a base de dados a qual, neste trabalho, foi definida como *domínio.local***.** A próxima configuração será do nome da organização onde se coloca o nome da mesma ou trabalho, em seguida será definido o método de armazenamento que a base de dados utilizará, sendo eles: *BDB; HDB e MDB***.**

Seleciona-se *MDB* que é uma nova forma de leitura de memória, a qual possibilita melhor eficiência na leitura dos dados, ao perguntar se deverá ser purgada, ou seja, limpar a base de dados, seleciona-se *<No>*, seleciona-se a opção *<Yes>* para mover a base de dados antiga, ao perguntar, se deve ser configurada a compatibilidade com *LDAPv2* escolhe-se a opção *<No>*. Para facilitar o gerenciamento do *OpenLDAP*,

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

pode-se utilizar o acesso via navegador, através do *phpldapadmin*, em que é necessário realizar a configuração do arquivo */etc/phpldapadmin/config.php* alterando as seguintes linhas, como demonstrado no quadro 06:

Quadro 06: Configuração do arquivo */etc/phpldapadmin/config.php.*

```
$servers = new Datastore();

$servers->newServer('ldap_pla');

$servers->setValue('server','name','TCC LDAP Server');

$servers->setValue('server','host','192.168.200.1');

$servers->setValue('server','base',array('dc=dominio,dc=local'));

$servers->setValue('login','auth_type','session');

$servers-
>setValue('login','bind_id','cn=admin,dc=dominio,dc=local');
```

Fonte: Autores, 2016.

A consulta na base de dados possui dados pré-definidos, os quais devem ser alterados indicando o novo caminho que deve efetuar as consultas. No arquivo */etc/ldap/ldap.conf* altera-se o nome da base como demonstrado no quadro 07:

Quadro 07: Comando alteração da base de dados.

```
BASE    dc=dominio,dc=local

URI     ldap://192.168.200.1
```

Fonte: Autores, 2016.

No arquivo */etc/freeradius/eap.conf* localizam-se as linhas responsáveis pelo método de criptografia que será aceita nas conexões, editam-se as linhas: *default_eap_type = md5* substituindo por, *default_eap_type = peap*, assim permitindo a compatibilidade com os sistemas operacionais Windows.

Para que o servidor *RADIUS* possa consultar os usuários da base de dados deve-se realizar a configuração. Para isso, edita-se o arquivo */etc/freeradius/modules/ldap*, dentro do arquivo deve-se descomentar as linhas *identity, password e base_filter*. Na linha *server*, se localiza o endereço da base de dados. Na linha *identity*, o usuário para acesso a base. Na linha *password* a senha da base e na linha *basedn* o domínio da base conforme o quadro 08:

Quadro 08: Edição do arquivo /etc/freeradius/modules/ldap.

```
ldap {
server = “192.168.200.1”
identify = “cn=admin,dc=dominio,dc=local”
password = 123
basedn = “dc=dominio,dc=local”
```

Fonte: Autores, 2016.

Após as alterações faz-se necessário reiniciar o serviço, assim utilizando o comando */etc/init.d/slapd restart.*

Após efetuadas tais configurações o *FreeRADIUS* terá acesso a base de dados, porém deve-se configurar os arquivos responsáveis por realizar a verificação dos métodos de autenticação como *PEAP***,** *TLS* e *TTLS* e suas criptografias, para que tenham permissão para verificar as informações da base do servidor *LDAP*.

No arquivo */etc/freeradius/sites-available/default* e no arquivo */etc/freeradius/sites-available/inner-tunnel* deve-se descomentar algumas linhas específicas nas sessões *authorize* e *authenticate* respectivamente informando a utilização da base de dados *LDAP* como demonstra-se o quadro 09:

Quadro 09: Configuração de arquivos default e inner-tunnel.

Authorize {
ldap
}
	Authenticate {
	Auth-type LDAP {
	ldap}
	}

Fonte: Autores, 2016.

Para a captura dos *logs* de conexões faz-se necessária a alteração do arquivo responsável por incluir todos os arquivos configurados a execução do servidor RADIUS. Alterou-se as linhas no arquivo /etc/freeradius/radiusd.conf, onde strypped_names habilita a exibição da identidade que o usuário utiliza ao solicitar permissão para autenticar na base de dados. A linha auth habilita a gravação das tentativas de autenticação e na linha auth_badpass coleta-se a identidade incorreta com que o usuário está tentando se autenticar, conforme o quadro 10:

Quadro 10: Alteração de linhas no arquivo *//etc/freeradius/radiusd.conf*.

strypped_names = yes auth = yes auth_badpass = yes

Fonte: Autores, 2016.

Após a configuração do arquivo radiusd.conf é possível efetuar a coleta dos dados de acesso ao servidor, posteriormente executando-o em modo debug, permitindo que todos os dados de execução processados sejam exibidos na tela. Porém adicionou-se comandos para executar a saída da informação direcionando para um arquivo de texto, permitindo a análise das informações. Para isso utilizou-se os seguintes comandos como demonstrado no quadro 11:

Quadro 11: Alteração de linhas no arquivo *//etc/freeradius/radiusd.conf*

Comando	**Descrição do comando**
/etc/init.d/freeradius stop	Interrompe o serviço em execução
freeradius -X >> testelog.txt	Executa o servidor em modo debug direcionando o resultado para um arquivo de texto.

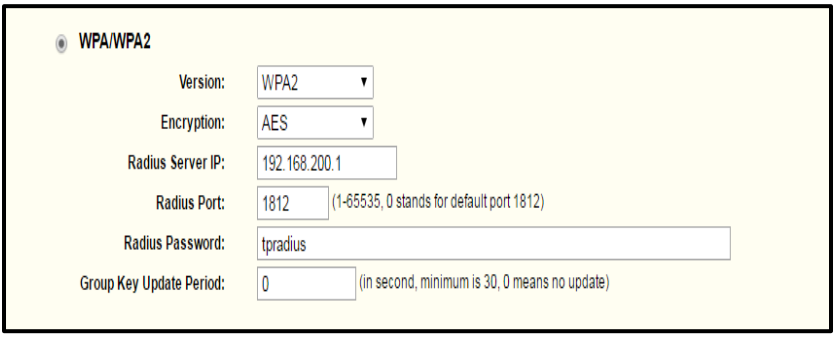
Fonte: Autores, 2016.

Efetuou-se a configuração do Access Point em modo *router*, foi utilizado o endereço de rede IP 192.168.0.0/24 para a interface LAN e o endereço 192.168.200.2/30 na interface WAN. O SSID da rede wireless foi definido com teste *radius*.

O modo de segurança utilizado foi *WPA2* com método de encriptação *AES*, o endereço do servidor RADIUS foi definido como *192.168.200.1* e a senha como *tpradius* como demonstrado na figura 03:

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Figura 03: Configuração do método de segurança utilizado pela rede wireless no AP.

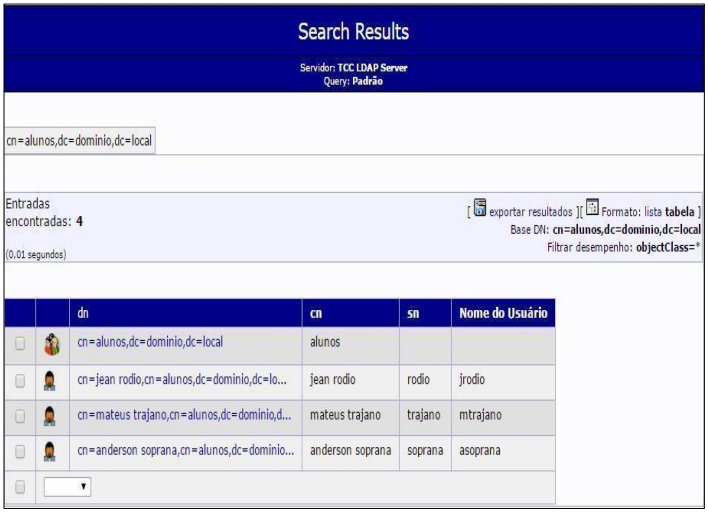


Fonte: Autores, 2016.

5 Resultados

Ao analisar-se os dados contidos no arquivo de log, foi obtido um resultado positivo ao desenvolvimento do trabalho, onde se confirma a veracidade das informações contidas no presente trabalho.

Os dados coletados fornecem informações de suma importância. Ao verificar-se a base de dados pela interface gráfica pode-se notar uma tabela, a qual possui todos os usuários cadastrados, possibilitando dados dos mesmos com o arquivo de log gerado pelas conexões estabelecidas, como demonstra a figura 04:Figura 04: Demonstração de usuários cadastrados na base de dados *LDAP*.



Fonte: Autores, 2016.

A leitura do arquivo contendo os logs de conexões pode ser interpretada da

seguinte forma, seguindo a ordem das linhas destacadas, como demonstrado no quadro 12:

Requerimento de acesso provindo do ponto de acesso com endereço IP 192.168.200.2;

Usuário mtrajano;

Expansão dos dados para comparação com os dados da base;

Expansão da base para localização dos dados;

Busca-se na base pelo usuário que possui o dado uid=mtrajano

Comparação de senha informada para autenticação do usuário;

Confirmação de autenticação do usuário.

Quadro 12: Demonstrativo de acesso de log servidor *FreeRADIUS* em modo debug.

```
Ready to process requests.
rad_recv: Access-Request packet from host 192.168.200.2 port 2050, id=160, length=160
        User-Name = "mtrajano"
        NAS-IP-Address = 192.168.1.1
        NAS-Port = 0
        Called-Station-Id = "54-E6-FC-F5-32-EE:teste radius"
        Calling-Station-Id = "5C-C9-D3-54-33-EB"
        Framed-MTU = 1400
        NAS-Port-Type = Wireless-802.11
        Connect-Info = "CONNECT 0Mbps 802.11"
        EAP-Message = 0x0201000d016d7472616a616e6f
        Message-Authenticator = 0x4861897f6580f4f265f39f6b9c82eaf5
# Executing section authorize from file /etc/freeradius/sites-enabled/default
+group authorize {
++[preprocess] = ok
++[chap] = noop
++[mschap] = noop
++[digest] = noop
[suffix] No '@' in User-Name = "mtrajano", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] = noop
[eap] EAP packet type response id 1 length 13
[eap] No EAP Start, assuming it's an on-going EAP conversation
++[eap] = updated
++[unix] = notfound
++[files] = noop
[ldap] performing user authorization for mtrajano
```

```
[ldap]    expand: %{Stripped-User-Name} ->
[ldap]    ... expanding second conditional
[ldap]    expand: %{User-Name} -> mtrajano
[ldap]    expand:    (uid=%{%{Stripped-User-Name}:-%{User-
Name}}) -> (uid=mtrajano)
[ldap]    expand: dc=dominio,dc=local -> dc=dominio,dc=local
  [ldap] ldap_get_conn: Checking Id: 0
  [ldap] ldap_get_conn: Got Id: 0
  [ldap] performing search in dc=dominio,dc=local, with filter
(uid=mtrajano)
[ldap] No default NMAS login sequence
[ldap] looking for check items in directory...
  [ldap] userPassword -> Password-With-Header == "123"
[ldap] looking for reply items in directory...
  [ldap] ldap_release_conn: Release Id: 0

++[ldap] = ok
```

Fonte: Autores, 2016.

Para realizar uma análise mais detalhada pode-se efetuar uma verificação do arquivo de *logs* do *FreeRADIUS* localizado em **/var/log/freeradius/radius.log**. Ao analisar o arquivo, consta a tentativa de autenticação sendo uma tentativa realizada com sucesso e uma segunda tentativa sem êxito, onde identifica-se a senha que o usuário utilizou, como demonstra o quadro 13:

Quadro 13: logs de acesso do *FreeRADIUS*.

```
Tue Nov 15 18:47:33 2016 : Auth: Login OK: [mtrajano] (from
client 192.168.200.1/24 port 0)

Tue Nov 15 18:47:47 2016 : Auth: Login incorrect (rlm_pap:
CLEAR TEXT password check failed): [mtrajano/tentativa] (from
client 192.168.200.1/24 port 0)
```

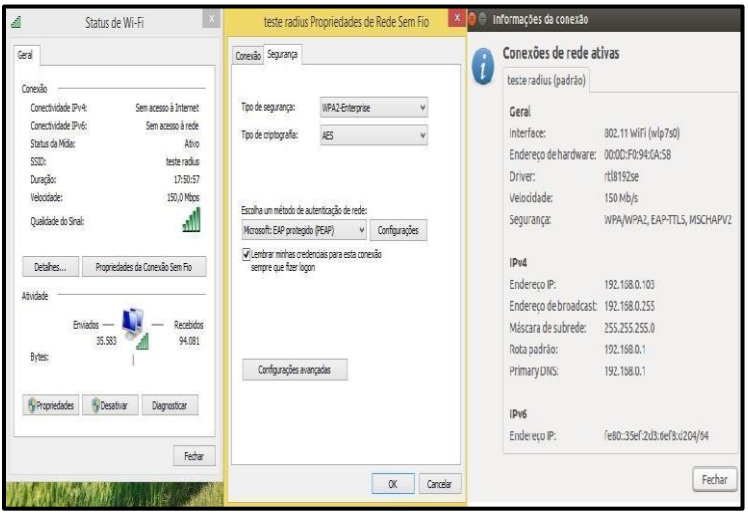
Fonte: Autores, 2016.

Pode-se identificar os status de conexão após efetuar uma autenticação nos sistemas testados, sendo eles o Windows 8.1 e Ubuntu 16.04 LTS, tais conforme demonstra a figura 04:

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Figura 04: Conexões estabelecidas com identificação da criptografia utilizada.

Fonte: Autores, 2016.



6 Trabalhos Futuros

Objetivando-se trabalhos futuros, pode-se utilizar do trabalho como uma base para a implementação e migração a rede *eduroam*[1], com base nas linhas e comandos dados no artigo, seria possível a transferência da rede atual para a rede em questão configurando-se linhas de comando no servidor *FreeRADIUS*, apontando em seu arquivo de configuração, o endereço IP da base de dados em que a *eduroam* faz suas requisições de acesso. A relação deste trabalho com a rede *eduroam* é que ambas utilizam serviço *FreeRADIUS* com protocolo 802.1x, tornando sua principal diferença a base de dados. A rede *eduroam* utiliza a base de dados da rede CAFe[2] pois a mesma é ampla, robusta e confederada. Fazendo com que alunos de todas as instituições federais de ensino superior possam utilizar seu usuário e senha para acessarem a internet de modo automático enquanto estão visitando ou estudando em outras instituições com a mesma rede.

---

[1] Principal iniciativa da RNP dedicada à questão da mobilidade, o eduroam (education roaming) é um serviço desenvolvido para a comunidade internacional de educação e pesquisa que oferece acesso sem fio à internet sem a necessidade de múltiplos logins e senhas, de forma simples, rápida e segura.(RNP, 2016a).

[2]Comunidade Acadêmica Federada reúne informações de identidade de diversas instituições de ensino e pesquisa brasileiras, unificando suas bases de dados. (RNP, 2016b).

7 Considerações Finais

A autenticação via FreeRADIUS torna-se solução para se obter desempenho de gerenciamento em uma rede e maior segurança, visto que é pertinente para a escolha de tal protocolo e serviço, ao acesso de usuários, com autenticação aos serviços de rede onde queira-se implementar os pilares da segurança, sendo eles: confidencialidade, disponibilidade e autenticidade. Tratando-se de segurança em redes, podemos citar a importância que a autenticação com o protocolo *RADIUS* nos traz, com isso provendo um melhor gerenciamento da rede.

Vivemos na era da informação compartilhada e isso nos remete a certos cuidados que devemos ter com relação a segurança de nossos dados pessoais. Nas empresas e instituições não é diferente esse pensamento, visto que seus dados e informações são de extrema confidencialidade. Podemos considerar os serviços de autenticação via *FreeRADIUS* e 802.1x em conjunto com uma base de dados LDAP uma das soluções que melhor se encaixa ao nível de extrema segurança em redes sem fio devido a sua arquitetura e autenticação em base de dados , sendo uma das ferramentas mais seguras com relação a autenticação de usuários.

8 Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - ABNT. **NBR ISO IEC 27002.** Tecnologia da informação- Técnicas de Segurança- Código de prática para a gestão da segurança da informação. Rio de Janeiro: ABNT, 2005.

DIÓGENES, Yuri; MAUSER, Daniel. **Certificação Security +:** Da prática para o exame. Editora Novaterra, 2ª ed. Rio de Janeiro, 2013.

ESKELSEN, Maiko; LUCKMANN, Francisco A. **Interface de Administração de Diretórios Estudo de Caso Open*LDAP*.** Monografia (Graduação em Sistemas de Informação) - Universidade Federal de Santa Catarina, Florianópolis, 2006.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5.ed. São Paulo: Atlas, 2010.

HE, Changhua; MITCHELL, John C.. **Security Analysis and Improvements for IEEE 802.11i.** Stanford, E.U.A. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.1515&rep=rep1&type=pdf>. Acesso em: 4 Nov. 2016.

HOWES, Timothy A.; SMITH, Mark C.; GOOD, Gordon S. - **Undertanding and Depoloying *LDAP* Directory Services**. Nova York: New Riders Publishing, 1998.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade**. Fundamentos de metodologia científica**. 5. ed. São Paulo: Atlas, 2003.

NETO, Roberto Miyano. **A Evolução dos Mecanismos de Segurança para Redes sem fio 802.11.** Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Departamento de Informática. Engenharia de Computação. Rio de Janeiro, Nov. 2004.

NUNES, Helton Lessa., et. al. OpenVPN com autenticação FreeRADIUS com uso parcial de endereços IPv6. In: JÚNIOR, Vanderlei Freitas., et. al. (Org.). **Tecnologia em redes de computadores**: estudos aplicados. Sombrio: Instituto Federal Catarinense - Campus Avançado Sombrio, 2015. p.10-38.

PAZ, Leonardo F., et. al. **Um Método para Minimizar Falhas de Segurança em Redes WLAN 802.11b/g: Controlando Acessos Provenientes de Dispositivos Móveis**. Anal (Anais do EATI Encontro Anual de Tecnologia da Informação e Semana Acadêmica de Tecnologia da Informação), Frederico Westphalen, 2015.

PEREIRA, Marcos H. **Segurança de redes sem fio, uma proposta com serviços integrados de autenticação *LDAP* e *RADIUS***. Artigo (Especialização em Redes e Segurança de Sistemas) Pontifícia Universidade Católica do Paraná, Curitiba, 2009.

PERES, André; WEBER, Raul Fernando. **Considerações Sobre Segurança Em Redes Sem Fio.** Artigo científico. UFRGS – Universidade Federal do Rio Grande do Sul Pós-Graduação em Computação. Porto Alegre, 2003. Disponível em:<http://ceseg.inf.ufpr.br//anais/2003/07.pdf>. Acesso em: 4 jul. 2016.

RIGNEY, Carl; et. al. **Remote Authentication Dial Em User Service (RADIUS).** IETF**.** RFC 2865**.** Pleasanton, E.U.A. Jun. 2010. Disponível em:<https://tools.ietf.org/html/rfc2865>. Acesso em: 4 jul. 2016.

RNP. eduroam: **eduroam**. Disponível em: <https://www.rnp.br/servicos/servicos-avancados/eduroam>. Acesso em: 10 Nov. 2016a.

RNP. CAFe: **Comunidade Acadêmica Federada**. Disponível em:<https://www.rnp.br/servicos/servicos-avancados/cafe>. Acesso em: 31 Out. 2016b.

SERMERSHEIM, Jim.. Lightweight Directory Access Protocol (*LDAP*): **The Protocol**. IETF. RFC 4511. Provo, E.U.A. Jun. 2006. Disponível em: <https://tools.ietf.org/html/rfc4511>. Acesso em: 4 jul. 2016.

SANTOS, Juliano P. **Servidor RADIUS com Conexão LDAP**. 2011. 57 f. Monografia (Especialização em Configuração e Gerenciamento de Servidores e Equipamentos de Redes) – Programa de Pós-Graduação em Tecnologia, Universidade Tecnológica Federal do Paraná, Curitiba, 2011.

SARL, Network RADIUS; FreeRADIUS. **The FreeRADIUS Technical Guide. 2014.** Disponível em: <http://networkradius.com/doc/FreeRADIUS%20Technical%20Guide.pdf>. Acesso em: 4 jul. 2016.

ZEILENGA, Kurt D. Lightweight Directory Access Protocol (*LDAP*): **Technical Specification Road Map**. IETF. RFC 4510. E.U.A. Jun. 2006 A. Disponível em: <http://tools.ietf.org/html/rfc4510>. Acesso em: 4 jul. 2016.

ZEILENGA, Kurt D.. Lightweight Directory Access Protocol (LDAP): **Directory Information Models**. IETF. RFC 4512. E.U.A. Jun. 2006 B. Disponível em: <http://tools.ietf.org/html/rfc4512#section-2.3>. Acesso em: 4 jul. 2016.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Honeypot de Baixa Interatividade como Ferramenta Complementar na Segurança da Rede

Maurício Mesquita Baltazar[1], Sander Cândido Macarini[1], Marcos Henrique de Morais Golinelli[1], Tatiana Marcela Rotta[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia Catarinense – Campus Avançado Sombrio – SC – Brasil

{mauriciomb,sander.macarini}@gmail.com, {marcos.golinelli,tatiana.rotta}@sombrio.ifc.edu.br

Abstract. *The massive use of the Internet attracts the interest of malicious people seeking undue advantages in computer systems. In order to improve network security, this paper aims to implement a low interactivity honeypot using the Honeyd application. Specifically, the information generated will be analyzed to evidence the attacks. The article is made on bibliographic research based on the content of the developers, on scientific articles and applied experimental research. After completing this article, it was verified, through the information generated, the success in the use and the fulfillment of the purpose, demonstrating the effectiveness in complementing the security. Although the experiment was not be exposed to a target environment, it was possible to verify the incidence and types of attacks.*

Resumo. *O uso massivo da Internet atrai o interesse de indivíduos mal-intencionados que buscam vantagens indevidas em sistemas computacionais. Visando aprimorar a segurança da rede, este artigo objetiva implementar um honeypot de baixa interatividade utilizando o aplicativo Honeyd. Especificamente serão analisadas as informações geradas para evidenciar os ataques. O artigo fundamenta-se em pesquisa bibliográfica baseada no conteúdo dos desenvolvedores, em artigos científicos e pesquisa experimental aplicada. Após a conclusão deste artigo constatou-se, através das informações geradas, o êxito na utilização e o cumprimento do propósito, demonstrando a eficácia na complementação da segurança. Embora o experimento não tenha ficado exposto a um ambiente visado, foi possível verificar as incidências e tipos de ataques.*

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

1. Introdução

O uso em massa da Internet para as mais variadas finalidades, entre elas atividades financeiras, atrai o interesse de pessoas mal-intencionadas que se utilizam de técnicas maliciosas para obter vantagens indevidas em sistemas vulneráveis. Diante desta realidade a preocupação com a segurança da informação está presente no cotidiano dos profissionais de tecnologia responsáveis pelas organizações.

O presente artigo aborda o *honeypot*, que é um dos recursos de segurança disponíveis para minimizar riscos. *Honeypot* (pote de mel em português) é um conceito que utiliza ferramentas capazes de simular sistemas operacionais com vulnerabilidades expostas, para assim coletar informações, métodos e técnicas usadas por invasores contra redes e sistemas de computadores.

Para entender a motivação no uso do *honeypot* como solução complementar de segurança, é importante saber um pouco sobre as limitações das outras soluções de segurança disponíveis para redes de computadores, por exemplo, os sistemas de detecção de intrusão em redes. Conforme Provos e Holz (2007) os sistemas de detecção para intrusão em redes (*Network Intrusion Detection Systems* (NIDS)) vêm perdendo a eficácia nas informações fornecidas devido a sofisticadas técnicas de evasão onde um número crescente de protocolos utiliza criptografia para proteger o tráfego de rede dos invasores. Os NIDS também sofrem com incidências de "falsos positivos" diminuindo ainda mais sua eficácia. O fluxo de informações que entram e saem de um *honeypot* permite reunir informações que não estão disponíveis em um NIDS (PROVOS; HOLZ, 2007). Outra solução, conforme NBSO (2003), que operando isoladamente não pode ser considerada infalível é o *firewall*, pois a sua simples instalação não garante que a rede esteja protegida contra invasores. Muito embora o *firewall* seja uma ferramenta indispensável, ele não pode ser a única linha de defesa e sim um dos diversos mecanismos e procedimentos que proporcionam robustez na segurança de uma rede.

Assim, em ambientes onde o fluxo e o armazenamento de informações sensíveis são massivos, o uso exclusivo de NIDS, ou mesmo NIDS combinados com *firewalls*, podem ser consideradas soluções incompletas de segurança. Com a utilização do *honeypot* é possível incrementar o nível de segurança, pois os sistemas simulados são capazes de capturar procedimentos suspeitos dentro de aplicações configuradas, e desta maneira tipificar comportamentos de pessoas e aplicações danosas. Levando em conta que estes sistemas simulados não estão no ambiente de produção da organização, é válido destacar que todo o tráfego destinado a eles deve ser considerado suspeito, sendo assim, o valor de um *honeypot* é mensurado através das informações que podem ser obtidas através dele.

Dessa forma, objetiva-se com o estudo implementar um honeypot de baixa interatividade utilizando o aplicativo *Honeyd*. Especificamente analisar os resultados, com base nas informações capturadas, evidenciando as incidências das sondagens e as tentativas de ataque realizadas nos serviços disponibilizados.

2. Revisão de literatura

Nesta seção serão apresentados fundamentos sobre a segurança em redes de computadores, as ferramentas de segurança, os tipos de ataques, os alvos preferenciais,

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

os *honeypots*, as *honeynets* e o aplicativo *Honeyd*.

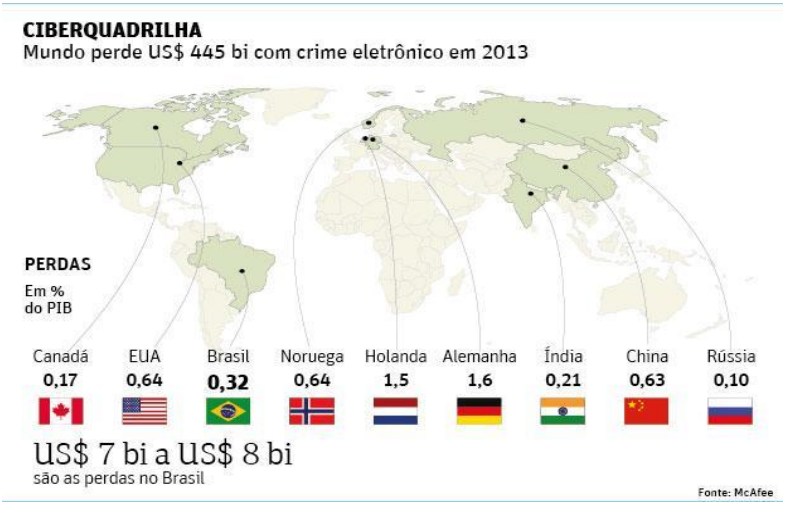
2.1. Segurança em Redes

O grande número de tentativas em obter vantagens indevidas com acesso não autorizado a redes computadorizadas, tornam necessárias providências para evitar que estas tentativas sejam premiadas com êxito. Além das medidas para conter os ataques já conhecidos, é importante também dispor de ferramentas que possam evidenciar novos padrões de ataques, e desta maneira manter as soluções de segurança atualizadas com a maior brevidade, minimizando a possibilidade de perdas.

Segundo Smith (2015), as perdas estimadas no mundo com crimes cibernéticos em 2015 ficaram na ordem dos U$ 500 bilhões e levando em conta a rápida digitalização no comportamento da vida dos consumidores e ainda o aumento dos registros empresariais, estima-se que esse número atinja a cifra de U$ 2,1 trilhões em 2019, multiplicando em quatro vezes o montante de 2015.

A imagem abaixo ilustra o panorama onde, já em 2013, o Brasil e o mundo sofriam perdas grandiosas com crimes cibernéticos.

Figura 01- Perdas com crimes eletrônicos em 2013.



Fonte: Folha de São Paulo. Disponível em: http://www1.folha.uol.com.br/mercado/2014/06/1467110-brasil-perde-ate-us-8- bilhoes-com-crime-cibernetico.shtml. Acesso em: 10 de jun 2016.

Diante do exposto, fica evidenciado que as preocupações relativas à segurança da informação são relevantes e todos os esforços para a melhoria das condições de operação são justificáveis.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

2.2. Tipos de Ataque

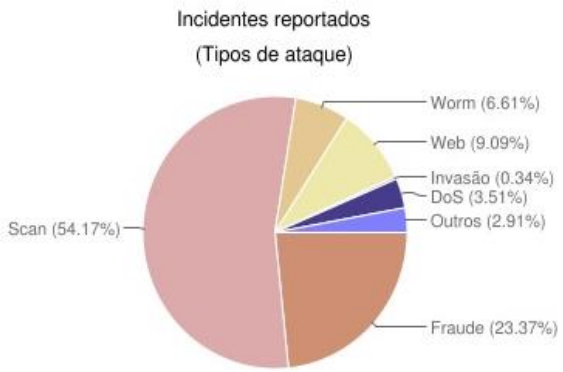
Segundo Shirey (2000), um ataque tem como função atingir a segurança de um sistema que procede de uma ameaça inteligente, utilizando o método ou técnica de burlar, infringindo a política de segurança de um sistema. Ravanello, Hijazi e Mazzorana (2004), complementam que a política de proteção é o conjunto de regras que tende de regulamentar o acesso, o tráfego de informações e recursos computacionais em uma instituição que dita como operar em caso de violação das regras determinadas.

A seguir estão listadas algumas formas de ataques: ataques automatizados e ataques manuais.

- Ataques automatizados são aqueles que não dependem da atenção humana para sua atuação, fazendo uso de *scripts* ou *softwares* específicos para invasão (KLER; PRADO, 2004).
- Ataques manuais são aqueles que escolhem cuidadosamente um alvo e um objetivo antes de executar a invasão. Além de serem mais nocivos por terem o conhecimento técnico que falta aos invasores automáticos, são eles que constroem suas próprias ferramentas (RAVANELLO, HIJAZI e MAZZORANA, 2004).

A Figura 02, ilustra os incidentes reportados ao site [3]CERT.br de janeiro a dezembro de 2015, denotando a porcentagem das formas de ataques.

Figura 02 - Incidentes Reportados ao CERT.br / Janeiro à Dezembro/2015



[3] Sobre o CERT.br; é um Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil é mantido pelo NIC.br, do Comitê Gestor da Internet no Brasil, e atende a qualquer rede brasileira conectada à Internet.

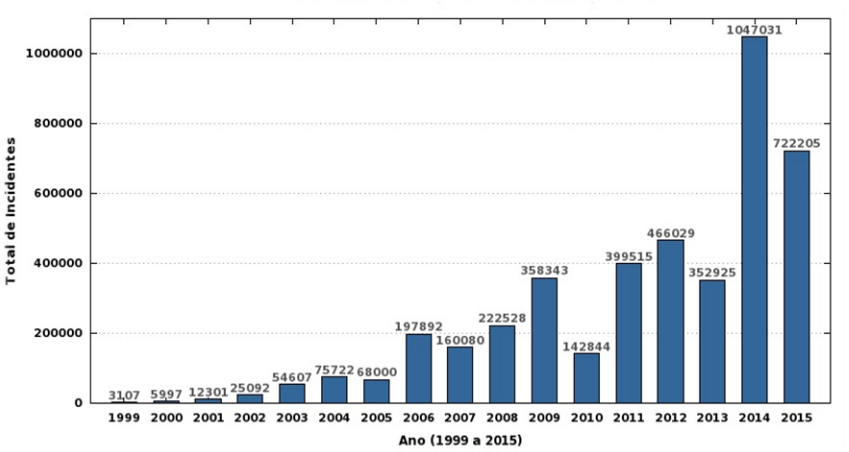
Fonte: Incidentes reportados ao cert.br (2015). Disponível em: http://www.cert.br/stats/incidentes/2015-jan-dec/ tipos-ataque.html. Acesso em: 15 de set 2016.

Para o entendimento da figura 02, destaca-se a seguir a legenda dividida em tipos de ataque (CERT.BR, 2015):

- *Worm:* é um programa malicioso capaz de se auto replicar em computadores ou redes;
- DoS (*Denial of Service*): notificações de ataques de negação de serviço, utilizando um ou mais computadores para tirar de operação um serviço, um computador ou uma rede;
- Invasão: ataque bem-sucedido a um computador ou rede não autorizado;
- *Web*: um ataque específico visando o comprometimento de servidores *web* ou páginas da internet;
- *Scan*: notificações de varreduras em redes de computadores, visando detectar quais computadores e serviços estão ativos ou não e suas vulnerabilidades;
- Fraude: é um ato de má-fé em que ocorre na tentativa de obter vantagem;
- Outros: notificações de incidentes que não se enquadram nas categorias anteriores.

Conforme a figura 03, analisa-se o total de incidentes anuais reportados ao CERT.br de 1999 à 2015:

Figura 03 - Total incidentes Reportados ao CERT.br por ano - 1999 à 2015.



Fonte: Incidentes reportados ao Cert.br por ano (1999 - 2015). Disponível em: http://www.cert.br/stats/incidentes/. Acessado em: 15 de set 2016.

Observa-se um crescente número de incidentes no decorrer dos anos. Nota-se um índice bastante elevado no ano de 2014 em relação aos anos anteriores.

2.3. Ferramentas de Segurança de Redes

Para obter maior segurança no tráfego e na administração dos dados existentes no ambiente de rede, utiliza-se ferramentas de segurança (RAVANELLO, HIJAZI e MAZZORANA, 2004).

A grande maioria dos sistemas computacionais estão expostos a um grande risco de ataques, onde estes ataques estão ficando mais complexos e com maiores possibilidades de danos. Um administrador de redes pode dispor de ferramentas, tais como: *firewalls*, sistemas de detecção de intrusão, antivírus e *honeypots*, definidos a seguir:

- *Firewalls* são pontos de entrada entre redes com um conjunto de hardware e software para analisar o tráfego e os serviços requisitados em uma rede, impedindo acessos externos não autorizados (OPPLIGER, 1996);
- Detecção é definida como percepção de algo através de seus indícios ou sinais, ação de descobrir, de revelar o que estava oculto (DICIO, 2016), já a intrusão é definida como qualquer ação que afete a integridade, confidencialidade ou a disponibilidade do sistema. Reconhecer agentes que não tem permissão para utilizar um sistema computacional, e reconhecer usuários que têm acesso legítimo ao sistema, mas estão ultrapassando de seus privilégios (BALASUBRAMANIYAN *et al*., 1998). NIDS são ferramentas que evidenciam a detecção e intrusão;
- Antivírus é uma ferramenta de segurança que tem a função de detectar, anular e eliminar de um computador vírus e outros programas prejudiciais, podendo atuar com funcionalidade de *firewall* pessoal (SANS, 2014);
- *Honeypot* é definido como um sistema ou um recurso computacional de segurança destinado a ser sondado, atacado ou comprometido (HOEPERS, JESSEN e CHAVES, 2007).

2.4. Honeypots e Honeynets

As primeiras publicações referentes ao conceito do *honeypot* surgiram no início da década de 90. Desde então, novos artigos publicados e produtos comerciais foram desenvolvidos com base no conceito. As primeiras obras públicas que documentam o conceito de *honeypot* são o livro de Clifford Stoll intitulado "*The Cuckoo's Egg"* (O Ovo do Cuco) e um documento técnico chamado de *"An Evening with Berferd in Which a Cracker Is Lured, Endured, and Studied"* (Uma noite com Berferd, na qual um cracker é atraído, tolerado e estudado) escrito por Bill Cheswick. Entretanto não significa que os *honeypots* não tenham sido desenvolvidos antes de 1990. Eles foram propostos e usados por uma variedade de organizações bem antes disso. Uma grande quantidade de pesquisa e desenvolvimento ocorreu dentro de áreas

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

militares, governamentais e organizações comerciais, porém nada sobre este conceito veio a público antes de 1990 (SPITZNER, 2002).

Destaca-se o conceito de *honeypots* como um sistema ou recurso computacional destinados a ser sondado, atacado ou comprometido, em um local que permite o registro e controle dessas atividades.

Honeypot é diferente da maioria das ferramentas de segurança. Grande parte das tecnologias usadas hoje foi projetada para resolver problemas específicos. Por exemplo, *firewall* é uma tecnologia que protege uma rede, controlando o que pode fluir no tráfego. Os *firewalls* são utilizados como um sistema para controle de acesso, são mais comumente implantados em torno do perímetro de uma organização para bloquear a atividade não autorizada. Sistemas de detecção de intrusão são projetados para detectar ataques e monitorar qualquer sistema ou atividade de rede. Eles são usados para identificar a atividade não autorizada (RAVANELLO, HIJAZI e MAZZORANA, 2004).

Honeypot não está limitado para solucionar um problema específico. Em vez disso, é uma ferramenta altamente flexível e pode ser aplicada a uma variedade de situações. É por isso que a definição de *honeypot* pode parecer à primeira vista vaga, porque eles podem ser utilizados para alcançar objetivos. Por exemplo, honeypots geralmente são utilizados para detectar e impedir os ataques, compartilhado com *firewalls*, agindo de forma semelhante à funcionalidade de um sistema de detecção de intrusão. Podem ser usados para capturar e analisar ataques automatizados, tais como *worms*, ou agir como sensores de indicação de alerta precoce, capturando as teclas digitadas ou conversas de atacantes (SPITZNER, 2002).

Montes (2004), afirma que *honeypots* possuem dois níveis de serviços e ressalta que estes níveis são como eles trabalham. Classificados em baixa interatividade e alta interatividade. Possuem a seguinte definição:

- *Honeypot* de baixa interatividade consiste em sistemas operacionais emulados através de ferramentas as quais os atacantes interagem. A instalação e configuração do sistema operacional (S.O.) real deve ser efetuado em modo seguro para ficar inacessível ao atacante. Exemplo de uma ferramenta que implementa *honeypot* de baixa interatividade é o *Honeyd* (JESSEN; CHAVES, 2009).
- *Honeypot* de alta interatividade consiste em máquinas que operam com aplicações e serviços reais, podendo dar total controle desses sistemas ao atacante, se obtiver uma invasão bem-sucedida, oferecendo um grande risco ao sistema. (JESSEN; CHAVES, 2009). Lopes (2013), complementa que sua implementação deve ser em uma rede que tenha um bom controle de proteção e detecção. Pois sua instalação é mais complexa, porém possibilita coletar grande quantia de informações dos atacantes.

O quadro 01 especifica as diferenças entre *honeypot* de baixa e alta interatividade:

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Quadro 01 - Características do *honeypots* de Baixa e alta interatividade.

Características	Baixa Interatividade	Alta Interatividade
Instalação	Fácil	Mais difícil
Manutenção	Fácil	Trabalhosa
Obtenção de Informações	Limitada	Extensiva
Necessidade de mecanismos de contenção	Não	Sim
Atacante tem acesso ao S.O. real	Não (em teoria)	Sim
Aplicações e serviços oferecidos	Emulados	Reais
Atacante pode comprometer o honeypot	Não (em teoria)	Sim
Risco de a organização sofrer um comprometimento	Baixo	Alto

Fonte: JESSEN, Klaus Steding. Uso de Honeypots no auxílio à detecção de ataques Campus Party Brasil. São Paulo. jan de 2011. p. 25-41.

Em 1999, surge a idealização das *honeynets*, dedicada a investigar os ataques e a desenvolver ferramentas de segurança de código aberto para melhorar a segurança na internet. O projeto *honeynet* é uma organização internacional de pesquisa de segurança sem fins lucrativos. Lance Spitzner, fundador do projeto, é um líder internacionalmente reconhecido no campo da pesquisa de ameaças cibernéticas, treinamento e conscientização de segurança (DEPASQUALE, 2015). Denomina-se *honeynet*, redes compostas de uma sub-rede de administração e de uma sub-rede de *honeypots* (FRANCO, BARBATO e MONTES, 2004).

Existem dois tipos de *honeynets* que são: *honeynets* virtuais e reais:

- *Honeynets* reais são todos os dispositivos que a compõem, inclusive os *honeypots*, mecanismos de contenção, mecanismos de alerta e mecanismos de coleta de informação, são físicos. Sendo assim, uma *honeynet* real irá conter diversos computadores para cada *honeypot*, com instalações de sistemas, aplicações e serviços reais (HOEPERS, JESSEN e CHAVES, 2007).
- *Honeynets* virtuais baseia-se no conceito em que todos os elementos de uma *honeynet* são implementados em uma quantidade reduzida de dispositivos físicos. Sendo assim, geralmente utiliza-se apenas um computador com sistema operacional instalado, utilizado como base para

execução ferramentas de virtualização como o VMware, VirtualBox entre outros (HOEPERS, JESSEN e CHAVES, 2007).

Existem vários tipos de ferramentas para implementação de *honeypots* como, *Deception Toolkit (DTK)*, *CyberCop Sting*, *KFsensor*, *Nepenthes*, *Dionaea*, *BackOfficer*, *Friendly (BOF)*, *Specter*, *Honeyd*, entre outras (LOPES, 2013).

A aplicação *Honeyd* foi escolhida para o experimento por ser uma solução *open source* e possuir quantidade razoável de material para composição do estudo.

2.5. Honeyd

A aplicação *Honeyd* foi desenvolvida por Niels Provos em software livre. Ele é também o responsável por manter site do projeto e pelas principais publicações disponíveis.

O *Honeyd* é um *honeypot* de baixa interatividade capaz de criar *hosts* virtuais em uma rede. Estes *hosts* virtuais, conhecidos como "*personalities"* podem ser configurados para executar serviços como se fossem máquinas reais. O *Honeyd* permite que uma única máquina anfitriã divulgue vários endereços IP simulando vários sistemas operacionais e vários tipos de serviços capazes de coletar informações. A aplicação suporta uma grande variedade de funcionalidades deixando-o flexível para vários tipos de simulações (PROVOS, 2008).

3. Materiais e Métodos

O estudo fundamenta-se majoritariamente em conteúdos bibliográficos publicados *online*, o que conforme Gil (2010) podem ser considerados materiais de pesquisa. Dentro deste contexto, na elaboração do projeto, a pesquisa bibliográfica foi desenvolvida utilizando publicações disponíveis na rede e também em livros impressos.

Para o desenvolvimento do estudo, conforme os procedimentos, foi utilizado a pesquisa experimental. Severino (2007) destaca a pesquisa experimental baseada na avaliação de um conjunto de elementos, sendo que estes devem ser estudados, manipulados e analisados. Para isso, utiliza-se de recursos e métodos a fim de atingir os objetivos do estudo.

Para a aplicação prática dos testes, como máquina anfitriã, foi utilizado um *notebook* Dell com 1GB de memória RAM, processador Intel Core 2 Duo e disco rígido de 320GB. Neste computador, foi usado o sistema operacional Ubuntu 12.04 e a aplicação *Honeyd* versão 1.05c.

3.1. Ambiente de pesquisa

O experimento foi desenvolvido inicialmente em ambiente doméstico com rede simulada, onde foram efetuados configurações e testes preliminares. Após a fase inicial, a aplicação foi submetida a um ambiente real; onde na rede do Instituto Federal de Educação, Ciência e Tecnologia Catarinense – Campus Avançado Sombrio (IFC) no período de 31/10/2016 à 04/11/2016 foram coletadas informações referentes às sondagens no *honeypot*.

3.2. Procedimentos para instalação do Honeyd

O processo de instalação do software *Honeyd*, no sistema operacional Ubuntu 12.04, pode ser efetuado através do terminal em linha de comando onde o comando "*sudo apt-get install honeyd"* faz o download dos pacotes e a instalação da *deamon*. Para o funcionamento da aplicação também são necessárias algumas dependências, sendo elas: *libevent, libdnet* e *libpcap*. Além da aplicação, este modelo de *honeypot* necessita do aplicativo Arpd trabalhando em conjunto para que sejam respondidas as solicitações arp destinadas ao *Honeyd*.

3.3 Arquivos do Honeyd

Após a instalação da aplicação, é necessário a configuração do *Honeyd* para que ele possa operar conforme o ambiente e a necessidade do administrador. As configurações e parametrizações do *Honeyd* são definidas através de arquivos localizados no computador anfitrião onde está rodando o *daemon*.

3.3.1. Arquivo padrão de inicialização

O arquivo padrão para inicialização localizado na pasta "/etc/default/" com o nome "honeyd", constam informações sobre a interface onde o honeyd irá efetuar a escuta em modo promíscuo, a faixa de rede utilizada e opções de inicialização. Abaixo segue o conteúdo do arquivo de inicialização com as configurações utilizadas:

Quadro 02: Arquivo de inicialização

```
# Defaults for honeyd initscript
# Master system-wide honeyd switch. The initscript
# will not run if it is not set to yes.
RUN="yes"
# Default options.
# Interface to listen on (if unset honeyd will select an interface himself)
# Note: Use only one! if you wish to use more than one use multiple -i in OPTIONS

INTERFACE="eth1"
# Network Honeyd will listen for. IF this is not set
# Honeyd will claim _all_ IP addresses set on the configured
# interface (which is probably _not_ what you want)
# This "sane" default will prevent you from doing it.
#NETWORK=192.168.1.0/24

NETWORK=172.16.38.0/24
# You can set some other options here for example:
# Notice that some of the options are already  defined in the
```

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

```
        # init.d script and you shouldn't use them here This includes: -f, -
l, -p, -x and -a.
        # Other options are available.
        # This is the default since the webserver uses
        # no authentication and could potentially disclose internal
        # information (and provide a remote user to edit honeyd's
configuration).

        OPTIONS="--disable-webserver"
        # Remove --disable-webserver if you want
        # to start up the web server. You will need to ask# honeyd
        # to fix the permissions of the document root.
        # NOTE: You will not be able to manipulate honeyd's
configuration
        # unless you grant the honeyd user write permission on the
        # configuration files at /etc/honeypot/
      #OPTIONS="--fix-webserver-permissions".
```

Fonte: Os autores. 2016.

No arquivo acima, é possível verificar a existência de duas linhas com a informação *NETWORK*, onde uma está comentada e a outra não. Quando uma linha inicia com o caractere "#", a aplicação não utiliza seu conteúdo, deixando-o apenas como comentário. A representação é feita desta maneira devido as configurações usadas nos diferentes ambientes utilizados. No arquivo acima, a linha comentada faz referência a informação usada na rede doméstica e a não comentada consta a informação da rede do IFC. Dependendo do ambiente onde funcionará o experimento, o caractere é inserido ou retirado, fazendo vigorar a linha que não consta o caractere.

3.3.2. Arquivo de Configuração

O arquivo de configuração localizado na pasta *"/etc/honeypot/"* com o nome *"honeyd.conf"* armazena as informações referentes às personalidades que serão simuladas pelo *Honeyd*. Neste arquivo estão presentes definições dos sistemas operacionais e estações que serão simulados, como por exemplo as portas de serviço liberadas, o endereço MAC, o endereço de IP, os endereços para inicialização dos scripts de serviço, entre outras. O comportamento das estações simuladas e seus serviços dependem das informações inseridas neste arquivo. Abaixo segue o conteúdo do arquivo de configuração com as definições utilizadas no experimento:

Quadro 03: Arquivo de configuração.

```
#HONEYD CONFIG
create default
set default default tcp action block
set default default udp action block
```

```
set default default icmp action block

#WINDOWS XP SP1 TEMPLATE
create windows
set windows personality "Microsoft Windows XP Professional SP1"
set windows default tcp action reset
set windows default udp action reset
set windows default icmp action open
add windows tcp port 445 open
add windows tcp port 135 open
add windows tcp port 5900 "/usr/share/honeyd/scripts/win32/win2k/vnc.sh"
add windows tcp port 23 "/usr/share/honeyd/scripts/telnet/faketelnet.pl"
set windows ethernet "00:01:02:03:04:06"
#bind 192.168.1.201 windows
bind 192.168.208.2 windows
#FTP LINUX TEMPLATE
create linuxftp
set linuxftp personality "Linux 2.4.7 (X86)"
set linuxftp default tcp action reset
set linuxftp default udp action block
set linuxftp default icmp action open
add linuxftp tcp port 445 open
add linuxftp tcp port 21 "sh /usr/share/honeyd/scripts/unix/linux/suse8.0/proftpd.sh $ipsrc $sport $ipdst $dport"
set linuxftp ethernet "00:01:02:03:04:05"
# bind 192.168.1.171 linuxftp
bind 192.168.208.3 linuxftp
##### CRIA UM ROTEADOR #####
create router
set router personality "Cisco 1601R router running IOS 12.1(5)"
set router default tcp action reset
add router tcp port 22 "/usr/share/honeyd/scripts/test.sh"
add router tcp port 23 "/usr/share/honeyd/scripts/router-telnet.pl"
set router ethernet "0a:1b:2c:3d:4e:5f"
#bind 192.168.1.100 router
bind 192.168.208.4 router
```

Fonte: Os autores, 2016.

Da mesma forma que aconteceu no arquivo de inicialização, destacam-se linhas duplicadas, com e sem comentário (#), para o uso nos dois ambientes ao qual o experimento foi submetido.

O primeiro trecho do arquivo de configuração, sob o comentário "*#HONEYD CONFIG*" cria um padrão para descartar todo tráfego que não seja aquele definido posteriormente no arquivo de configuração.

No segundo trecho do arquivo, sob o comentário "*#WINDOWS XP SP1 TEMPLATE*", é criado um dispositivo emulado com a "*personality*" do Microsoft Windows XP Professional SP1, ou seja, em uma sondagem por uma ferramenta de scanner, ele responderá com a informação relativa a este sistema operacional. Ainda neste trecho, existem as definições que liberam as portas 445 (SMB), 135 (EPMAP), 5900 (VNC) e 23 (TELNET). Sendo que, as portas 5900 e 23 chamam scripts nos endereços indicados. As informações restantes deste trecho informam o endereço MAC e endereço IP deste dispositivo.

O terceiro e quarto trechos, fazem referência respectivamente aos dispositivos "Linux 2.4.7 (X86)" e "Cisco 1601R router running IOS 12.1(5)". Neles também constam as suas respectivas configurações.

3.4. Inicialização do Honeyd

A inicialização da aplicação é feita através de linha de comando, onde é enviado um sinal de execução para o programa e são passadas as variáveis. Na execução conforme o experimento temos a seguinte sintaxe:

"honeyd -d -f /etc/honeypot/honeyd.conf -a /etc/honeypot/nmap.assoc -p /etc/ honeypot/nmap.prints -0 /etc/honeypot/pf.os -x /etc/honeypot/xprobe2.conf -u 115 -g 125 -i eth1 -l /var/log/honeypot/honeyd.log -s /var/log /honeypot/servicehoney.log"

As variáveis do comando possuem as seguintes atribuições:

- "-d": Inicializa o aplicativo em modo "*debug*", onde as informações pertinentes serão demonstradas em tela;
- "-f": Indica para o aplicativo o local onde o arquivo de configuração está localizado dentro da máquina anfitriã;
- "-a": Indica para o aplicativo o local para o arquivo que associa impressões digitais do xprobe com o nmap;
- "-p": Indica para o aplicativo o local onde o arquivo com as impressões digitais para o nmap está localizado;
- "-0": Indica para o aplicativo o local onde o arquivo que permite ao Honeyd identificar o sistema operacional do host remoto;
- "-x": Indica para o aplicativo o local onde o arquivo com as impressões digitais do xprobe. Permite que o Honeyd responda corretamente as solicitações de ferramentas ICMP;
- "-u": Indica o usuário em que a aplicação vai executar;
- "-g": Indica o grupo em que a aplicação vai executar;
- "-i": Indica a interface em que a aplicação vai escutar em modo promíscuo;
- "-l": Indica para o aplicativo onde será armazenado o arquivo de log em nível de pacotes;

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

- “-s“: Indica para o aplicativo onde será armazenado o arquivo de log em nível de serviço.

3.5. Análise dos *logs*

A análise operacional é feita através dos logs do Honeyd. A aplicação gera diferentes tipos de logs, onde são capturadas as informações das atividades realizadas na rede e nos serviços configurados. No escopo do estudo são aplicados logs em dois níveis de operação:

- Log de pacotes;
- Log de serviços.

O quadro abaixo, mostra um exemplo ilustrativo de um *log* em nível de pacotes:

Quadro 04 – Exemplos de Log.

Date	Proto	Source IP Port	Destination IP Port	Info Comment
2005-04-02-15:35:15	tcp(6)	10.3.6.139 1827	10.1.2.124 3 128	[Windows XP SP1]
2005-04-02-15:35:56	tcp(6)	10.3.6.139 437 8	10.1.2.84 8 080:	48 S [Windows XP SP1]
2005-04-02-15:36:11	tcp(6)	10.4.7.196 267 1	10.1.2.175 2 380:	40 R [FreeBSD 5.0-5.1]
2005-04-02-15:39:47	tcp(6)	10.3.6.139 1827	10.1.2.123 3 128:	9950 240
2005-04-02-15:40:18	icmp	10.3.5.182	10.1.3.99:	11(0): 56

Fonte: Provoz e Holz (2007)

Analisando o quadro acima, ainda conforme Provos e Holz (2007) tem-se que a coluna *Date* contém a informação da data e hora em que o pacote foi recebido pelo *Honeyd*. A próxima coluna, *Proto*, contém informação sobre o protocolo, geralmente são os protocolos TCP, UDP e ICMP. No entanto, ela pode receber informações de sondagens com pacotes de outros tipos de protocolos. A terceira coluna, rotulada com *T* contém informação sobre o tipo de conexão, onde *S* (Start) indica conexão inicial, *E* (End) conexão final e o símbolo “-” indica que o pacote não pertence a conexão alguma. As próximas quatro colunas mostram a informação de endereço IP e porta para origem

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

e destino. Alguns protocolos, como o ICMP, não utilizam portas, e por essa razão as colunas de porta estarão vazias. A coluna *Info* contém informações sobre uma conexão ou um pacote. No caso da conexão, ao seu término, serão mostrados números de bytes recebidos e enviados respectivamente pelo *Honeyd*. Caso o registro seja relativo a um pacote, o campo irá conter as seguintes informações adicionais do protocolo:

- TCP: O tamanho do pacote e a *flag* definida no cabeçalho. O *Honeyd* conhece as seguintes *flags*: F = Fin, S = Syn, R = Rst, P = Push, A = Ack, U = Urg, E=- ECE e C = CWR;
- ICMP: Em caso de pacote ICMP serão mostradas informações sobre o código, o tipo e o tamanho do pacote;
- UDP: Em pacotes UDP irá conter o tamanho do pacote.

A última coluna nomeada como *Comment*, possui informações adicionais onde, em muitos casos, mostrará uma suposição do sistema operacional remoto.

O quadro abaixo, mostra um exemplo ilustrativo de um log em nível de serviço:

Quadro 05- Log em nível de serviço

Date	Prot	Source IP Port	Destination IP Port	Data
2005-04-10-00:56:48	tcp6	10.3.23.14 3259	10.1.3.222 3128:	CONNECT 10.4.228.113:25HTTP/1.0
2005-04-10-00:59:12	tcp6	10.3.23.14 3343	10.1.3.124 8000:	CONNECT 10.5.167.5:25 HTTP/1.0
2005-04-10-01:04:20	tcp6	10.3.23.14 4116	10.1.3.209 3128:	some@net.em schan @net.em

Fonte: Provoz e Holz (2007)

Também conforme PROVOS e HOLZ (2007), os registros na tabela acima demonstram um endereço de IP caindo em servidores falsos de proxy e SMTP. O atacante tentou enviar mensagem de e-mail anonimamente através de um servidor de e-mail em um proxy aberto. Para o servidor remoto parece que o e-mail se origina a partir do endereço de IP do proxy**.**

Durante o experimento, verificou-se que o log de serviço conta com um arquivo de complemento para informações coletadas. Este arquivo, localizado em /var/log/honeyd.txt armazena informações que serão detalhadas na seção seguinte.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

4. Resultados do experimento com Honeypot

Ao longo do estudo, após as configurações necessárias, onde o experimento foi aplicado em duas situações distintas, foram coletadas informações que puderam evidenciar o funcionamento da aplicação.

No primeiro cenário, onde atuou em rede doméstica, a aplicação recebeu sondagens e tentativas de acesso nas máquinas configuradas. Abaixo, com base nos logs e seus diferentes níveis de captura, é possível identificar um destes incidentes:

- Na linha do arquivo com log de pacotes (variável -l) replicada abaixo, constata-se que em 30 de outubro de 2016 às 11 horas e 56 minutos, houve uma conexão iniciada (S) pelo IP 192.168.1.2 na porta 21 (Servidor FTP) da estação virtualizada com o IP 192.168.1.171 (personalidade criada no arquivo de configuração com o nome linuxftp). Consta também a informação do possível sistema operacional do qual o atacante fez o acesso (Windows 2000 RFC 1323). A linha a seguir demonstra estas afirmações:

2016-10-30-11:56:43.2280 tcp(6) S 192.168.1.2 2955 192.168.1.171 21 [Windows 2000 RFC1323]

- Já a linha abaixo, também do log em nível de pacotes, capturada às 12 horas e 02 minutos do mesmo dia, evidencia o final da conexão (E) iniciada acima. Constam também informações sobre os bytes recebidos e enviados, conforme explicado no capítulo anterior:

2016-10-30-12:02:35.4322 tcp(6) E 192.168.1.2 2955 192.168.1.171 21: 369 989

- Nas linhas do arquivo de log em nível de serviços (variável -s), referente ao mesmo incidente, é possível aprofundar o entendimento. A primeira linha mostra o *Honeyd* iniciando o script (*proftpd.sh*) que atua como emulador de um servidor FTP. A segunda linha mostra o momento em que o script é encerrado:

2016-10-30-11:56:43.3189 tcp(6) 192.168.1.2 2955 192.168.1.171 21: |/usr/share/honeyd/scripts/unix/linux/suse8.0/proftpd.sh: 1:

2016-10-30-12:02:35.4181 tcp(6) 192.168.1.2 2955 192.168.1.171 21: |/usr/share/honeyd/scripts/unix/linux/suse8.0/proftpd.sh: 67: [: no: unexpected operator|

Ainda em nível de serviço, no arquivo de complemento, ficam evidenciadas as atividades do atacante através do comportamento registrado. Foi possível identificar comandos como USER, PORT, XPWD, entre outros dentro do console Telnet. No quadro abaixo temos o trecho que evidencia o exposto:

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Quadro 06 - Arquivo complementar do log de serviços.

```
--MARK--,"Sun Oct 30 11:56:43 BRST 2016 "," pro-ftpd/FTP "," 192.168.1.2 "," 192.168.1.171 ", 2955,21,
"OPTS UTF8 ON
USER user
PORT 192,168,1,2,11,140
PORT 192,168,1,2,11,141
USER admin
HELP
TYPE I
CWD /temp
PORT 192,168,1,2,11,146
PORT 192,168,1,2,11,147
PORT 192,168,1,2,11,148
PORT 192,168,1,2,11,149
PORT 192,168,1,2,11,150
PORT 192,168,1,2,11,151
PORT 192,168,1,2,11,152
dir
user
PORT 192,168,1,2,11,154
PORT 192,168,1,2,11,155
XPWD
verbose
QUIT
",
--ENDMARK—
```

Fonte: Os autores, 2016.

O segundo cenário foi estabelecido na rede do IFC, onde também foram capturadas informações, conforme ataques simulados. Abaixo estão replicados trechos dos logs que os evidenciam:

- No log de pacotes (variável -l), faremos a análise nas seguintes linhas:

  2016-10-31-20:25:13.0612 tcp(6) S 192.168.210.7 44588 192.168.208.4 23 [Linux 2.2 20-25]

  2016-10-31-20:27:21.4365 tcp(6) E 192.168.210.7 44588 192.168.208.4 23: 66 1035

Da mesma forma que o modelo anterior, nas linhas acima constam informações sobre início (S) e fim (E) de uma conexão. Entretanto, desta vez a um servidor *telnet* (Porta 23).

- As linhas abaixo, conforme o *log* em nível de serviço, mostram quais usuários e senha o atacante usou para tentar o acesso:

- 2016-10-31-20:25:52.2155 tcp(6) 192.168.210.7 44588 192.168.208.4 23:
- |Attempted login: user/user| 2016-10-31-20:26:11.8405 tcp(6) 192.168.210.7 44588 192.168.208.4 23:
- |Attempted login: root/root| 2016-10-31-20:27:13.0955 tcp(6) 192.168.210.7 44588 192.168.208.4 23:
- |Attempted login: admin/admin|

Nos exemplos acima, foram demonstradas tentativas de acesso em tipos diferentes de aplicações simuladas. No primeiro exemplo o atacante tentou aplicar comandos dentro de um servidor FTP. O segundo exemplo foram feitas tentativas de login em um servidor telnet. Desta forma, ficou evidenciado que houveram as sondagens ao *honeypot*, bem como ataques à diferentes tipos de serviços.

5. Considerações finais

O processo de implementação do *Honeyd* ocorreu de forma satisfatória e com base nos resultados obtidos é notório que as sondagens e tentativas de ataques ocorreram. Mesmo que o estudo tenha sido aplicado em ambiente doméstico e em ambiente acadêmico, o que não atrai a atenção de atacantes, a incidência de sondagens e tentativas, mesmo que pequena, foi detectada. Com isso foi possível, de maneira satisfatória, analisar o conteúdo e atingir os objetivos propostos com o experimento.

A maior dificuldade encontrada para evidenciar a eficácia do conceito honeypot, foi que os testes puderam ser realizados somente em ambientes pouco atrativos a atacantes, e desta maneira a incidência, crucial para geração de conteúdo analisável, não foi em grande escala.

Após a conclusão do artigo os autores sugerem, para trabalhos futuros, o aprofundamento do tema para elaboração de um trabalho, usando para tal, as funcionalidades de programação (*shell script e python*) para criação de novos recursos e outras funcionalidades como o funcionamento em conjunto com outras aplicações. Analisando os resultados de uma exposição da aplicação em um ambiente com grande fluxo de usuários e conteúdo.

6. Referências

BALASUBRAMANIYAN, Jai Sundar et al. An Architecture for Intrusion Detection using Autonomous Agents. COAST Laboratory: Purdue University, West Lafayette, IN 47907-1398. 12 p.

CERT.BR. **Incidentes Reportados ao Cert.br.** 2015. Disponível em: http://www.cert.br/stats/incidentes/2015-jan-dec/ tipos-ataque.html. Acesso em: 15 de set 2016.

DEPASQUALE, Andea; **An interview with Lance Spitzner, founder of The Honeynet Project.** Disponível em: <https://www.honeynet.org/HPW2015/interview-Lance-Spitzner -founder-Honeynet-Project>: Acesso em: 06 nov. 2016.

DICIO. **Dicionário Online de Português.** Disponível em: <https://www.dicio.com.br/deteccao/>. Acesso em: 09 set. 2016

FRANCO, L. H.; BARBATO, L. G. C.; MONTES, A.: **Instalação e Uso de Honeypot de Baixa Interatividade**. Centro de Pesquisas Renato Archer - CenPRA/MCT. Campinas -SP -Brasil. 2004.

GIL, Antônio C. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010.

HOEPERS, Cristine; JESSEN, Klaus Steding; CHAVES, Marcelo H. P. C. ***Honeypots* e *Honeynets*: Definições e Aplicações.** 2007. Disponivel em: <http://www.cert.br/docs/ whitepapers/honeypots-honeynets/#ref-02> Acesso em: 03 Nov. 2016.

JESSEN, Klaus Steding; CHAVES, Marcelo H. P. C. **Implantação de Honeypots de Baixa Interatividade com Honeyd e Nepenthes**. 2009. Disponível em:<http://www.cert.br/docs/palestras/ certbr-campusparty2008-2.pdf> Acesso em: 15 de outubro de 2016.

KLER, Evelyn Ruth; PRADO, Gelson. **Segurança de Redes Sistema de Detecção de Intrusão**. 75 f. Monografia (Especialização) - Curso de Administração Com ênfase em Análise de Sistemas, Faculdade Internacional de Curitiba, Curitiba, 2004.

LOPES, Alexandre. **Honeypots e Honeynets: As Vantagens de Conhecer o Inimigo.** Faculdade de Tecnologia de Ourinhos – FATEC. 2013.

MONTES, ANTÔNIO. **Honeypots e Honeynets:Contra-Inteligência no Ciberespaço**. 2004. Disponivel em:<http://mtc-m16c.sid.inpe.br/rep/sid.inpe.br/malu/2005/01.28.14.21? languagebutton=pt-BR> Acesso em: 30 de Ago. de 2016.

NBSO. **Práticas de Segurança para Administradores de Redes Internet.** 2003. Disponivel em: <http://www.cert.br/docs/seg-adm-redes/seg-adm-redes.html>. Acesso em: 26 out. 2016.

OPPLIGER, Rolf. **Internet Security: Firewalls and Beyond**. Communication of the ACM, v.40, n. 5, p.92-102, 1996.

PROVOS, Niels; HOLZ, Thorsten. **Virtual Honeypots: From Botnet Tracking to intrusion Detection**. 1. ed. 2007. <Disponível em: http://books.gigatux.nl/mirror/ honeypot/final/toc.html. Acesso em: 16 out. 2016.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

PROVOS, Niels **Developments of the Honeyd Virtual Honeypot.** 2008. Disponível em: <http://www.honeyd.org/index.php>. Acesso em: 02 ago. 2016.

RAVANELLO, Anderson Luiz; HIJAZI, Houssan Ali; MAZZORANA, Sidney Miguel. **Honeypots e Aspectos Legais.** 2004. 85 f. Dissertação - Pós-Graduação em Informática Aplicada, Pontifícia Universidade Católica do Paraná, Curitiba, PR.

SANS - **securing the human. O que é um ativirus?**. Dez. 2014. Disponível em: <https:// securingthehuman.sans.org/newsletters/ouch/issues/OUCH-201412_pt.pdf>. Acessado em: 18 de out 2016.

SEVERINO, Antônio J. **Metodologia do trabalho científico**. 23. ed. São Paulo: Cortez, 2007.

SHIREY, R. **Internet Security Glossary**. 2000. Rfc2828 Disponível em: http://www.ietf.org/rfc/rfc2828.txt, Acesso: 03 nov. 2016.

SMITH, Sam **Cybercrime Will Cost Businesses Over $2 Trillion By 2019.** 2015 <Disponível em: < https://www.juniperresearch.com/press/press-releases/cybercrime-cost-businesses- over-2trillion>. Acesso em: 29 out. 2016.

SPITZNER, Lance. **Honeypots: Tracking Hackers**. (Ed.). Addison Wesley, 2002. 480p.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Redes Definidas por Software (SDN): Filtro de Pacotes utilizando o Floodlight e Mininet

Jeferson de Sousa Oliveira[1], Lizandro da Silva Braga2, Marcos Henrique de Morais Golinelli3

[1,2,3]Instituto Federal Catarinense – *Campus* Avançado Sombrio – Sombrio – SC – Brasil.

[1]bonavex@hotmail.com, [2]lizandrosbraga@gmail.com, [3]marcos.golinelli@ifc-sombrio.edu.br

Abstract. *The centralization of the control of networks goes against the different proprietary software of the manufacturers of equipment (switchs and routers). The SDN project proposes that this control be done in a unified way through controller software. In the production of this work a brief bibliographical revision was realized, the preparation of the environment, installation and configuration of the software and later the insertion of firewall rules for diverse controls. As a result, it was possible to verify the effective functioning of the firewall rules through tests performed in Mininet. The adoption of SDN brings a more favorable performance in order to reduce the complexity in network management, facilitating the performance of network administrators, eliminating the need to configure devices in several ways manually.*

Resumo. *A centralização do controle de redes esbarra nos diferentes softwares proprietários dos fabricantes de equipamentos (switchs e roteadores). O projeto de SDN, propõe que este controle seja feito de forma unificada através de um software controlador. Na produção deste trabalho foi realizada uma breve revisão bibliográfica, a preparação do ambiente, instalação e configuração dos softwares e posteriormente a inserção de regras de firewall para controles diversos. Como resultado, foi possível verificar o funcionamento efetivo das regras de firewall através de testes realizados no Mininet. A adoção de SDN, traz um desempenho mais favorável, no sentido de reduzir a complexidade na gerência das redes, facilitando a atuação dos administradores de rede, eliminando a necessidade de configuração de dispositivos de diversas formas manualmente.*

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

1. Introdução

A Rede Definida por *Software* - SDN (do inglês *Software Defined Network*) é um conceito para a estrutura das redes que traz mais flexibilidade, rapidez e favorece novos serviços. É uma nova arquitetura de rede que evoluiu da ideia de redes virtuais (OPEN NETWORKING FOUNDATION, 2012). Com o crescimento da internet e sua utilização de forma comercial, a expansão das redes se deu de forma inevitável, porém a tecnologia de arquitetura baseada no modelo *TCP/IP* não acompanhou o crescimento no que se refere a sua inovação, além disso, o fato dos dispositivos serem controlados por softwares proprietários, engessou qualquer forma de desenvolvimento e personalização para a configuração das redes. Com este cenário, surge uma maneira diferente em como controlar e configurar as redes, tratando-as com um novo olhar (OLIVEIRA, D. s.d.).

Em geral, SDN significa que as redes são regidas por controladores, em vez dos consoles de gerenciamento de redes e comandos que exigem um grande esforço operacional, tornando complexa a administração em larga escala. As redes atuais ou Redes Legadas, como são conhecidas, para Costa (2013), são definidas por ter uma arquitetura rígida, pelo fato do plano de controle (*software*), e o plano de dados (*hardware*), estarem integrados, impossibilitando a opção de utilizar outro *software*, que não, o instalado pela empresa proprietária.

O objetivo geral do trabalho é apresentar os conceitos desta tecnologia, seus componentes e o seu funcionamento, e seu objetivo específico é simular uma rede definida por software através do aplicativo *mininet* e com o controlador *floodlight*, utilizando *firewall* para verificação.

2 Referencial Bibliográfico

Neste capítulo, são abordados os conceitos de Redes Definidas por S*oftware*, protocolo *OpenFlow*, controlador *Floodlight*, do emulador de redes *Mininet* e, também, é apresentado um comparativo teórico entre Rede Legada e *SDN* e os benefícios da *SDN*.

2.1 Rede definida por software (SDN) e protocolo openflow

Segundo Costa (2013), Redes Definidas por *Software* ou Redes Programáticas, como também são conhecidas, são redes controladas por programas, ou seja, se baseiam na separação do plano de dados, que controla o repasse de pacotes na rede e do plano de controle, que é responsável pelos protocolos e a atualização das tabelas de encaminhamento. Isso faz com que o controle da rede seja feito de um único ponto, e para que isso seja possível, foi criado o protocolo *OpenFlow*.

O Protocolo *OpenFlow* é de código aberto e foi criado para a programação de redes por usuários, programadores ou empresas independentes fornecedoras de software, com o objetivo de baixar os custos de implementação, gerenciamento, e também possibilitar melhorias no desenvolvimento de redes personalizadas, trazendo assim, mais eficiência e confiabilidade a rede.

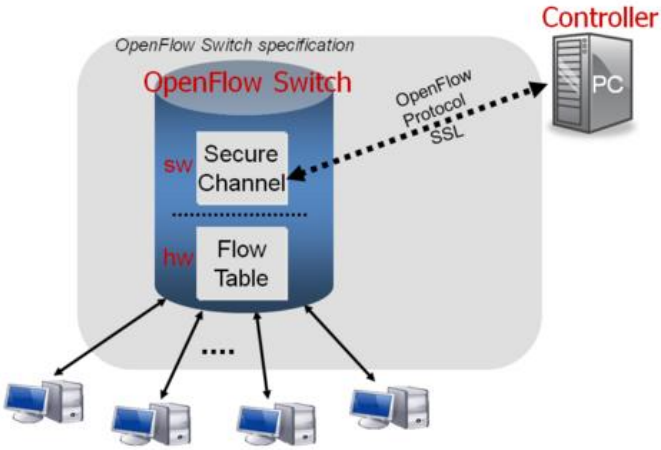
In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Nas palavras de Oliveira, N. (2014), o protocolo *OpenFlow*, tem a função de ligação entre os dispositivos de rede e o controlador, sendo que este último é programado de forma estática ou dinâmica, criando regras pré-definidas para que o protocolo identifique o tráfego da rede utilizando o conceito de fluxo.

O Protocolo *OpenFlow* é responsável pela comunicação entre o plano de dados e o plano de controle, além de reconhecer o fluxo de dados, identificando assim pacotes e utilizando campos já conhecidos, como endereço IP de origem e destino, endereço MAC (do inglês *Medium Access Control*), protocolos de transporte *Tranmission Control Protocol* (TCP) e *User Datagram Protocol* (UDP) e portas de origem e destino, trazendo mais agilidade ao encaminhamento e permitindo que a rede se adapte de maneira mais eficiente em tempo real. Para controlar o fluxo de dados nos dispositivos gerenciáveis, o *Openflow*, utiliza uma tabela de fluxo e também permite a criação de grupos de tabelas que definem um fluxo com uma entrada definida para um método a mais de encaminhamento. Sendo assim, o protocolo abre muitas possibilidades, pois além das tabelas serem manipuláveis, pode-se personalizar o tratamento do fluxo por parte de cada equipamento, além de permitir que o controle seja feito de forma programável (MODA, 2014).

A figura 1, ilustra um esquema de funcionamento de uma Rede Definida por *Software*, onde as estações estão diretamente conectadas ao *switch OpenFlow* e este ao controlador.

Figura 1 – Esquema funcionamento da SDN.



Fonte: Adaptado de Mackeown (2010).

Quando um pacote chega a um *switch* com *OpenFlow* habilitado, conforme Junior et al. (2013), os cabeçalhos são comparados às regras de entrada das tabelas de fluxo (*flow table)*, os contadores são atualizados, então as ações correspondentes são realizadas. Caso não haja correspondência do pacote em alguma entrada de tabela, o

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

pacote é entregue por completo ao controlador, através do canal seguro (*secure channel)*, que define pelo encaminhamento do pacote ou por seu descarte e também se grava uma nova entrada na tabela de fluxos.

2.2 Comparação entre SDN e rede legada

Para entender melhor o funcionamento de uma *SDN*, é preciso saber como funcionam a maior parte das redes atuais, ou legadas, e fazer um comparativo com as definidas por *software*, a fim de trazer os benefícios de uma migração.

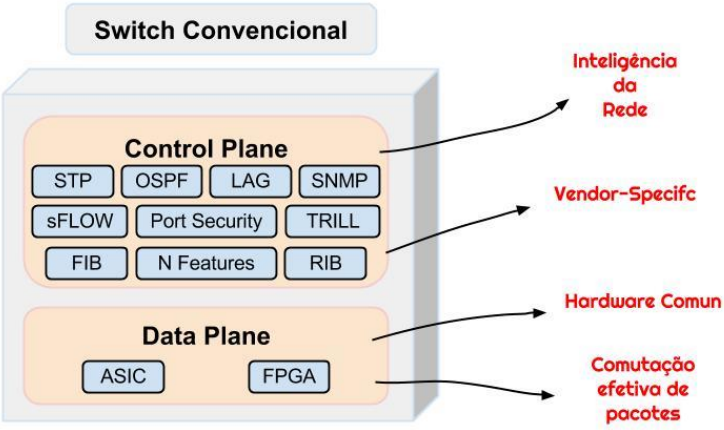
2.2.1 Rede legada

Cada *switch* deste tipo de rede, tem sua inteligência no *software* disponibilizado pelo fabricante. Este *software* é proprietário, o que quer dizer, que não é possível fazer qualquer melhoria ou até mesmo correções de *bugs*, fazendo com que o administrador fique pendente a espera de novas atualizações por parte do fabricante.

De acordo com Rothenberg et al. (2011), uma série de protocolos que formam a camada de *software* de controle são necessários para que a infraestrutura processe os dados para o encaminhamento dos mesmos, fazendo com que cada tipo de rede dispense um trabalho oneroso, além do que os desenvolvimentos de novas funcionalidades sejam restritos ao fabricante do equipamento, o que torna o processo custoso.

O *software* é individual a cada equipamento, o que significa que cada equipamento deve ser configurado individualmente. A inteligência contida no programa é responsável por decidir o que fazer com cada pacote que trafega pela rede e informa ao plano de dados como comutar os pacotes. A figura 3, demostra o esquema de funcionamento de uma Rede Legada.

Figura 2 - Esquema Rede Legada.



Fonte: Adaptado de Andrade (2014).

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

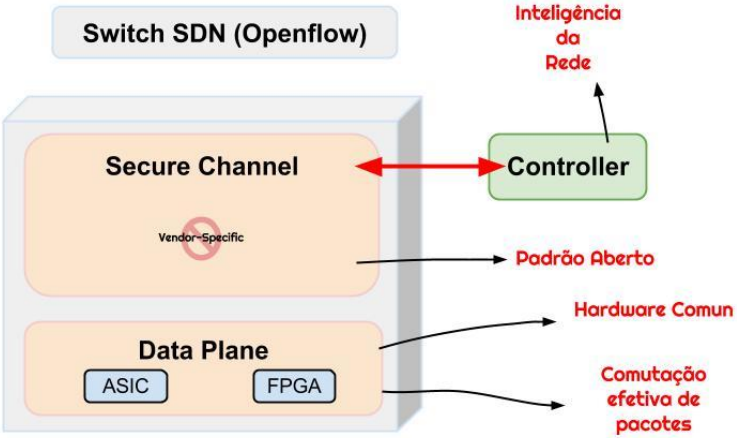
2.2.2 Rede SDN

Diferente da Rede Legada, que tem o controle de plano feito por *softwares* proprietários, a *SDN* faz esse controle direcionado pelo protocolo *OpenFlow*.

O protocolo *OpenFlow* cria uma nova funcionalidade no plano de controle, que é o *Secure Channel;* este se comunica com um *software* localizado no servidor chamado *Controller*, que é a inteligência do *switch*, mas localizado em área externa, o que possibilita a criação de novas funcionalidades, alterações e correções, independente do fabricante do equipamento. A comutação permanece inalterada, porém obedecendo as especificações do *Controller* para aprender o que fazer com cada pacote. A figura 4, demostra a comunicação diferenciada feita em uma *SDN*.

Figura 3 - Esquema Rede Legada.

Fonte: Adaptado de Andrade (2014).



2.3 Benefícios da SDN

Há inúmeros benefícios do uso de *SDN* e os principais deles, segundo a Open Networking Foundation (2012), são:

- O controle da rede é programável diretamente, pois ele é desacoplado das funções de encaminhamento;
- Ajuste dinâmico e ágil do fluxo de tráfego em toda a rede, atendendo as necessidades de mudanças;
- Permite os administradores de redes, configurar, gerenciar, otimizar e proteger os recursos de rede, pois não dependem de *softwares* proprietários;

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

- Simplificação do projeto e operação, pois as instruções são fornecidas pelo controlador (unificado), em vez de vir de vários dispositivos e protocolos diferentes;
- Diminuição no custo operacional, pois elimina a utilização de *softwares* e *updates* proprietários.

2.4 Controlador SDN

Segundo Costa (2013), o Controlador SDN, conhecido como "Sistema Operacional de Rede", oferece uma visão unificada de todos os elementos da rede, convergindo a comunicação com tais elementos, possibilitando o desenvolvimento de programas e através de análises detalhadas, realizar novas funções para o gerenciamento operacional. O autor realça sobre a existência de uma gama de controladores disponíveis como software livre, dentre os quais destacamos alguns, com uma síntese de cada.

- NOX – Referência na utilização do OpenFlow, desenvolvido em C++, mas as aplicações podem ser desenvolvidas em linguagem de programação Python, compatível com a plataforma Linux. Seu funcionamento consiste no gerenciamento do fluxo no *switch*, através de aplicações e serviços, criadas em uma camada de abstração.
- POX – Baseado no NOX, porém mais estável e com uma interface mais moderna e com melhor desempenho. Também desenvolvido em C++ e com suporte a Python, atua nas plataformas Linux, Windows e Mac.
- Maestro – Elaborado em linguagem *Java*, usufrui ao máximo da máquina em que está instalado para não se tornar um gargalo, uma vez que o controlador tem que tomar muitas decisões em um mesmo momento, trazendo assim maior desempenho. O Maestro também opera nas plataformas Linux, Windows e Mac.
- Trema – Não é necessariamente um controlador, mas sua importância se dá, por ser uma plataforma OpenFlow para a criação e desenvolvimento de controladores, em linguagem C e *Ruby*.
- Beacon – Desenvolvido em *Java*, trabalha nas plataformas Linux, Windows, Mac e Android. Suas principais características são, a possibilidade de ser atualizado em tempo de execução, sem interrupção e de ter uma implementação estável.

2.5 Controlador Floodlight

Este controlador tem suas aplicações baseadas em Java; é composto por um núcleo e módulos que permitem ao desenvolvedor, uma manipulação de códigos, a fim de personalizá-los para o desempenho da função desejada, que atendam a rede OpenFlow.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

"O desenvolvimento do controlador *Floodlight* surgiu do controlador *Beacon* e é mantido, testado e apoiado pela *Big Switch Networks* e pela maior comunidade de desenvolvedores do mundo para controladores SDN" (OLIVEIRA, 2014, p. 52)

O Controlador *Floodlight* (PROJETO FLOODLIGHT, 2016), foi desenvolvido para ser de fácil aplicação, com uma interface intuitiva, que suporta *switchs* físicos e virtuais ou redes híbridas, de forma simultânea. A figura 5, ilustra o controlador e os módulos de aplicação.

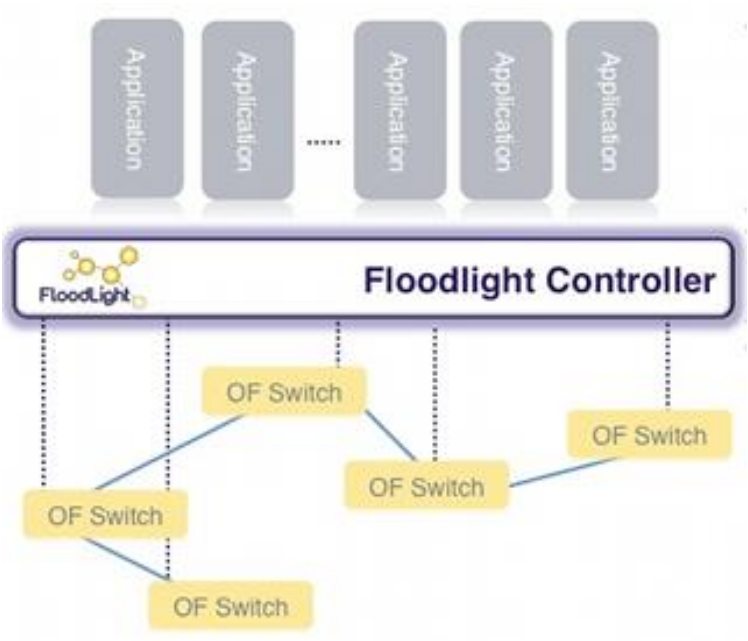


Figura 4 - Esquema Funcionamento Floodlight.

Fonte: Adaptado de Kerner (2012).

Os módulos são implementados em linguagem *Java*, compatível também em *Python*, e a comunicação com o controlador se dá através de uma API. Por padrão o controlador se comunica com o *switch* pela porta 6633, mas a porta 8080, provê uma interface *web*, por onde podem ser inseridas, editadas ou excluídas instruções ao controlador, através de APIs desenvolvidas em *Python* ou *Java*, além de exibir o status dos ativos da rede, a topologia da rede e ainda, é possível alterar algumas configurações. Também é possível efetuar instruções ao *Floodlight*, através da ferramenta *curl*, através de linhas de comando, via terminal.

2.6 Mininet

Desenvolvido para emular redes, se tornou uma ferramenta importante para

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

experimentos utilizando o protocolo *OpenFlow* e Redes Definidas por *Software*. Escrito em linguagem *Python*, este emulador cria vários *hosts*, *switchs* e roteadores em um único computador e se diferencia de outros emuladores, segundo Semedo (2014), por suportar o novo paradigma *SDN*. De acordo com Lantz et al. (2015), o *Mininet* utiliza recursos do *Kernel* do sistema *Linux*, que permite a divisão de um único sistema, sendo que cada divisão possui uma parcela de processamento fixa.

2.7 Firewall

Com o aumento de uso de informações, e de sua importância, ferramentas eficientes, que venham a trazer mais segurança nos dados que trafegam na rede, se fazem necessárias. Neste contexto, o *firewall*, segundo Kurose (2010), consiste em aplicar uma política de segurança, com o objetivo de regrar o tráfego de dados em redes diferentes, bloqueando pacotes não autorizados e/ou danosos. Ainda pode ser definido, de acordo com Panes (2011), como um dispositivo que deve analisar o tráfego de dados na rede, permitindo através da política de segurança, que dados autorizados trafeguem pela rede, devendo ser inviolável.

Neste trabalho, o *firewall* foi utilizado para analisar o controle do tráfego de dados na rede simulada, através de regras aplicadas no controlador *floodlight.*

2.8 Visual Network Description (VND)

O VND é uma aplicação web, onde é possível elaborar uma topologia de rede, configurá-la e exportar suas configurações. Segundo Fontes (2016), a aplicação permite simular e analisar cenários de rede, além da exportação de *scripts* em linguagem de programação *Python*, contendo as regras e tabelas de fluxo para serem utilizadas em Redes Definidas por *Software*. Os *scripts* gerados são inicializados no aplicativo mininet que simula a rede do cenário elaborado.

3. Materiais e métodos

Para a realização deste trabalho foi utilizada a pesquisa bibliográfica, que para Gil (2010) é realizada em materiais já publicados, e que tem como objetivo a possibilidade de alcançar respostas ao problema apontado e sua principal vantagem é a busca das soluções almejadas, através de dados geograficamente disseminados. Também foi utilizada a pesquisa experimental, que segundo Dias et al (2013), consiste na busca do efeito produzido no objeto da pesquisa, através de variáveis que possam provocar alterações, partindo assim, para as formas de observar e controlar tais transformações. A pesquisa experimental foi utilizada neste trabalho, na criação da topologia proposta, através do emulador e também na inserção de regras de *firewall* no controlador, o que possibilitou a coleta dos resultados obtidos. A escolha por um emulador de redes, se deu pela dificuldade em configurar equipamentos reais que dispunham.

3.1 Ambiente de pesquisa

No ambiente de trabalho, como demonstra a figura 6, foi empregado o uso de

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

dois *notebooks*, ambos com 4 gigabytes de memória RAM, porém o que foi utilizado como controlador, possui processador Intel I3, enquanto o que está com o mininet, um processador Intel Celeron QuadCore. Ambos com sistema operacional Ubuntu, versão 16.04, conectados entre si em rede local, utilizando um cabo UTP (do inglês *Unshielded Twisted Pair*), categoria 5, para a comunicação dos mesmos. Para alcançar o objetivo do trabalho, foram criadas sete regras de *Firewall*, entre liberações e bloqueios do tráfego de dados, possibilitando a obtenção dos resultados. Os resultados obtidos com as regras de *firewall* inseridas no controlador, são apresentadas tendo como consequência da aplicação do utilitário Ping, que segundo Silva (2010), verifica a conectividade de uma máquina em rede, apurando o tempo de resposta, sendo realizadas no emulador de rede. Também foi utilizado o *software* livre Iperf, uma ferramenta para medir largura de banda. Apesar de não ser o foco do trabalho, o Iperf, foi a ferramenta ideal para apresentar os resultados em que as regras foram baseadas no bloqueio de protocolos, pois esta ferramenta gera tráfego que pode ser direcionado por protocolos específicos.

Figura 5 - Ambiente de Pesquisa.

Fonte. Autores, 2016.

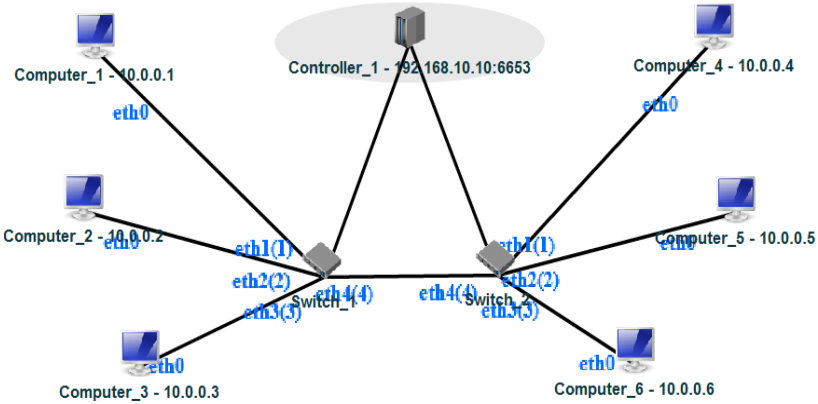
3.2 Instalação Mininet e criação da topologia de rede utilizando o VND

Na a criação da topologia de rede escolhida para realizar a comunicação com o controlador, objetivando a avaliação de *firewall* em uma SDN, os seguintes procedimentos foram adotados:

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

1. Instalação do simulador Mininet, através do comando *"git clone https://github.com/mininet/mininet.git"*. Caso haja erro, é necessário instalar o *git* no sistema, que pode ser feito por "*apt-get install git"*,
2. Após, foi feita a verificação dos arquivos e versão, pelos comandos *"git tag"* e *"git checkout -b 2.2.1 2.2.1"*,
3. Para a geração do *script* da topologia escolhida, conforme a figura 7, foi utilizado a *Visual Network Descripition* (VND), através de interface gráfica baseada na *web*, disponível no endereço site: *"http://www.ramonfontes.com/vnd/#"*,

Figura 6 - Topologia da Rede.



Fonte. Autores, 2016.

3. Após elaborado o *design* proposto para a topologia, é necessário que a exportação do *script*, demonstrado no quadro 1, se dê, dentro do diretório do *mininet*. O arquivo é gerado com extensão *.sh*, sendo necessário, como usuário *root*, renomeá-lo para extensão *.py* (abreviatura para arquivos contendo códigos em linguagem *python*).
4. Para efeito deste trabalho, o nome do arquivo é *mininet.py*,
5. Seguindo os procedimentos, como usuário *root*, o arquivo *mininet.py*, tornou-se executável utilizando o comando "*chmod +x mininet.py",*
6. Para executar o *script*, com a condição de que o controlador já esteja em execução no outro *notebook*, dentro do diretório do *mininet*, foi executado o comando *./mininet.py*, iniciando assim, a rede virtual, entrando em comunicação com o controlador.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Quadro 1. Script Exportado VND

```
#!/usr/bin/python
Script created by VND - Visual Network Description (SDN version)
from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSKernelSwitch, IVSSwitch,
UserSwitch
from mininet.link import Link, TCLink
from mininet.cli import CLI
from mininet.log import setLogLevel
def topology():

 net = Mininet( controller=RemoteController, link=TCLink, switch=OVSKernelSwitch )
 c1 = net.addController( 'c1', ip='192.168.10.10', port=6653 )
 s1 = net.addSwitch( 's1', protocols='OpenFlow13', listenPort=6673, mac='00:00:00:00:00:01' )
 s2 = net.addSwitch( 's2', protocols='OpenFlow13', listenPort=6674, mac='00:00:00:00:00:02' )

 h1 = net.addHost( 'h1', mac='00:00:00:00:00:03', ip='10.0.0.1/8' )
 h2 = net.addHost( 'h2', mac='00:00:00:00:00:04', ip='10.0.0.2/8' )
 h3 = net.addHost( 'h3', mac='00:00:00:00:00:05', ip='10.0.0.3/8' )
 h4 = net.addHost( 'h4', mac='00:00:00:00:00:06', ip='10.0.0.4/8' )
 h5 = net.addHost( 'h5', mac='00:00:00:00:00:07', ip='10.0.0.5/8' )
 h6 = net.addHost( 'h6', mac='00:00:00:00:00:08', ip='10.0.0.6/8' )

   net.addLink(h1, s1)
   net.addLink(h2, s1)
   net.addLink(h3, s1)
   net.addLink(h4, s2)
   net.addLink(h5, s2)
   net.addLink(h6, s2)
   net.addLink(s1, s2)
   net.build()
   c1.start()
   s1.start( [c1] )
   s2.start( [c1] )
   CLI( net )
   net.stop()
if __name__ == '__main__':
   setLogLevel( 'info' )
```

Fonte: Visual Network Descripition.

3.3 Instalação Floodlight

Antes de instalar o controlador *Floodlight*, foi necessário a instalação de alguns programas para seu pleno funcionamento. Após preenchidos os requisitos, foram adotados os procedimentos para instalação e configuração do controlador.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

1. Com o usuário *root* foi feita a instalação dos programas que são pré requisitos para seu pleno funcionamento através do comando *"#apt-get install build-essential default-jdk ant python-dev python-simplejson"*. Caso seja necessário, o *Oracle JDK* é o JDK oficial; contudo, ele não é mais fornecido pela Oracle como instalação padrão no Ubuntu, porém, ainda pode-se instalá-lo utilizando *"apt-get install oracle-java8-installer"*,
2. Após a instalação dos programas necessários, foi feito o download do controlador *floodlight*, através do endereço *web*, "*git clone https://github.com/floodlight/floodlight.git"*,
3. Dentro do diretório do controlador, foi executado o comando "*#java -jar target/floodlight.jar"*, dando início a execução do controlador. O *Floodlight*, tem uma interface *web*, onde podem ser encontradas informações sobre topologia, *switchs*, *hosts*, além de alterar configurações de controle sobre a rede. Tal interface é acessada no servidor pelo endereço *localhost:8080/ui/index.html*, sendo que *localhost*, pode ser substituído pelo endereço ip do servidor.

Uma vez iniciado o *script* no emulador *Mininet*, a rede funciona em total conectividade e se dá a comunicação com o controlador, que à identifica. Para avaliar a funcionalidade decisória do controlador no encaminhamento de pacotes, foi ativado um dos módulos disponíveis do controlador, que permite aplicar regras de *firewall*. Por padrão, após ativado o módulo, todo o tráfego é bloqueado na rede, sendo que o mesmo é habilitado pelo comando:

"curl http://localhost:8080/wm/firewall/module/enable/json -X PUT -d ""

Após o módulo ser habilitado, foram configuradas as regras de *firewall*, porém, para melhor compreensão, destacamos o significado de cada componente, descritos a seguir:

- -X e -d: O -X especifica o método que será utilizado e o -d especifica os dados que são enviados na requisição,
- POST – Método de requisição HTTP,
- src-ip: ip de origem,
- dst-ip: ip de destino,
- priority: prioridade da regra (quanto menor o número maior a prioridade),
- nw-proto: protocolo utilizado na regra. Ex: TCP, UDP, ICMP,
- action: Ação que a regra efetuará. Ex: ACCEPT (Permitir), DENY (Negar),
- src-mac: mac de origem,
- dst-mac: mac de destino.

A seguir, são demonstradas a regras com uma explicação da função cumprida de cada uma, na realização deste trabalho:

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Como a conectividade da rede foi bloqueada com a habilitação do módulo, não há possibilidade de aplicação de análise de tráfego, pois os *switchs* estão bloqueados. Com isso, os *hosts* não se comunicam, e também não é possível criar qualquer regra nesta condição. Sendo assim, a regra 1 foi aplicada para a liberação do tráfego no *switch* 1, com o objetivo de fazer os *hosts* que estão diretamente conectados há este *switch*, se comunicarem.

Regra 1) curl -X POST -d '{“switchid”: “00:00:00:00:00:00:00:01”, “priority”: “20”}' http://localhost:8080/wm/firewall/rules/json

A regra 2, tem como objetivo liberar o tráfego do *switch* 2. Como a rede só possui dois *switchs*, e o primeiro já foi desbloqueado com a regra 1, desbloqueando o segundo, possibilita que toda a rede possua conectividade, ou seja, que todos os *hosts*, possam se comunicar. Destaca-se que, tanto na regra 1, quanto na regra 2, foi utilizado como identificador, os endereços MAC dos *switchs*.

Regra 2) curl -X POST -d '{“switchid”: “00:00:00:00:00:00:00:02”, “priority”: “19”}' http://localhost:8080/wm/firewall/rules/json

Seguindo a criação de regras de *firewall*, a proposta da regra 3, é de bloquear o tráfego entre os *hosts* identificados pelos endereços de IP, 10.0.0.2/32 e 10.0.0.5/32, sendo que o primeiro é a origem de envio de pacotes e o segundo o destino.

Regra 3) curl -X POST -d '{“src-ip”: “10.0.0.2/32”, “dst-ip”: “10.0.0.5/32”, “priority”: “10”, “action”:”DENY”}' http://localhost:8080/wm/firewall/rules/json

Para garantir que não haja conectividade alguma entre os dois *hosts*, citados na regra anterior, instituímos a regra 4, de forma inversa a regra 3, no que se refere a origem e o destino, ou seja, o bloqueio dos *hosts* com endereço IP, 10.0.0.5/32, como origem do envio de pacotes, e o 10.0.0.2/32, como destino.

Regra 4) curl -X POST -d '{“src-ip”: “10.0.0.5/32”, “dst-ip”: “10.0.0.2/32”, “priority”: “10”, “action”: “DENY”}' http://localhost:8080/wm/firewall/rules/json

Na quinta regra, foi proposto o bloqueio dos *hosts* de endereço IP, 10.0.0.3/32 e 10.0.0.4/32, porém, só para dados que utilizem como meio de transporte o protocolo TCP. Nas palavras de Viegas (2008), o TCP é confiável, porque a transmissão de dados deste protocolo, recebe confirmação de recebimento do pacote.

Regra 5) curl -X POST -d '{“src-ip”: “10.0.0.3/32, “dst-ip”: “10.0.0.4/32”, “nw-proto”: “TCP”, “tp-src”: “8080”, “priority”: “5”, “action”: “DENY”}' http://localhost:8080/wm/firewall/rules/json

Já, na regra 6, o bloqueio proposto foi entre os *hosts* de endereço IP 10.0.0.1/32 e 10.0.0.3/32, dos dados trafegados entre eles que utilizem meio de transporte o protocolo UDP, que segundo a RFC 768 (1980), não é orientado à

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

conexão, pois não há confirmação de recebimento do pacote.

Regra 6) curl -X POST -d '{“src-ip”: “10.0.0.1/32, “dst-ip”: “10.0.0.3/32”, “nw-proto”: “UDP”, “tp-src”: “22”, “priority”: “4”, “action”: “DENY”}' http://localhost:8080/wm/firewall/rules/json

A última regra criada se deu entre os *hosts* de endereço IP 10.0.0.3/32 e 10.0.0.6/32, porém, representados na regra pelos seus respectivos endereços MAC. Está regra propõe o bloqueio do tráfego entre os *hosts*, que utilizem como meio de transporte, o protocolo ICMP.

Regra 7) curl -X POST -d '{“src-mac”: “00.00.00.00.00.05”, “dst-mac”: “00.00.00.00.00.08”, “nw-proto”: “ICMP”, “priority”: “18”, “action”: “DENY”}' http://localhost:8080/wm/firewall/rules/json

4 Resultados

O aplicativo *Mininet*, apresenta nos resultados, nomenclatura dos *switchs* e dos *hosts*, de forma diferente de como estão nas regras de *Firewall*. Por esta razão, é apresentado abaixo, a nomenclatura dos *hosts* utilizada pelo *Mininet*, seguida de seus endereços IP e endereços MAC, correspondentes, para que seja facilitada a compreensão dos resultados obtidos:

Switchs

s1 = Switch 1 – Endereço MAC = 00:00:00:00:00:01

s2 = Switch 2 – Endereço MAC = 00:00:00:00:00:02

Hosts

h1 = Endereço MAC = 00:00:00:00:00:03, Endereço IP = 10.0.0.1/32, = Switch 1
h2 = Endereço MAC = 00:00:00:00:00:04, Endereço IP = 10.0.0.2/32, = Switch 1
h3 = Endereço MAC = 00:00:00:00:00:05, Endereço IP = 10.0.0.3/32, = Switch 1
h4 = Endereço MAC = 00:00:00:00:00:06, Endereço IP = 10.0.0.4/32, = Switch 2
h5 = Endereço MAC = 00:00:00:00:00:07, Endereço IP = 10.0.0.5/32, = Switch 2
h6 = Endereço MAC = 00:00:00:00:00:08, Endereço IP = 10.0.0.6/32, = Switch 2

O quadro 2, demonstra os resultados obtidos com as regras criadas para desbloqueio dos *switchs* 1 e 2 respectivamente. A regra do *switch* 1 foi aplicada primeiro, por esta razão, os resultados apresentam somente o desbloqueio dos *hosts* conectados diretamente a ele, enquanto a regra do *switch* 2, demonstra a conexão de todos os *hosts* da rede. Para efeito de compreensão na análise dos resultados, o caractere “X”, atesta que a conexão está bloqueada.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Quadro 2: Resultado das Regras nos switchs.

```
Switch 1:

mininet> pingall

*** Ping: testing ping reachability

h1 -> h2 h3 X X X

h2 -> h1 h3 X X X

h3 -> h1 h2 X X X

h4 -> X  X  X X X

h5 -> X  X  X X X

h6 -> X  X  X X X

*** Results: 80% dropped (6/30 received)
```

Fonte: Autores, 2016.

O objetivo das duas regras, que resultaram no quadro acima, era fazer o desbloqueio dos dois *switchs*, sendo estas criadas no controlador, demonstrando assim, a centralização das configurações no encaminhamento de pacotes, utilizando o módulo de *firewall* do *floodlight*. O resultado do *switch* 1, exibe a conectividade com os *hosts* h1, h2 e h3, enquanto o resultado do *switch* 2, mostra a conexão entre todos os *h*osts, pois os dois *switchs*, estão interligados. O argumento a seguir, ilustrado no quadro 3, propôs o bloqueio de somente duas máquinas, conectadas diretamente em *switchs* diferentes.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Quadro 3: Resultado h2 e h5

```
Bloqueio hosts h2 e h5

IPs: 10.0.0.2/32 e 10.0.0.5/32


*** Ping: testing ping reachability

h1 -> h2 h3 h4 h5 h6

h2 -> h1 h3 h4 X h6

h3 -> h1 h2 h4 h5 h6

h4 -> h1 h2 h3 h5 h6

h5 -> h1 X h3 h4 h6

h6 -> h1 h2 h3 h4 h5

*** Results: 6% dropped (28/30 received)
```

Fonte: Autores, 2016.

É possível observar que, como parâmetros para o bloqueio dos *hosts*, foram utilizados seus endereços de IP, sendo abstraído tal informação no ping, mas incluída na regra de *Firewall*. Este tipo de bloqueio é um facilitador, porque as tabelas de fluxo do protocolo *OpenFlow* são constantemente atualizadas conforme o fluxo, e o IP é uma das formas de coleta de informação da rede. Sendo assim, simplifica o conhecimento total da rede, o que ajuda na criação de regras de *firewall* no controlador.

A quinta e sexta regras apuraram a possibilidade de bloqueio na comunicação de dados, utilizando como fator, os protocolos TCP e UDP. A quinta regra realiza o bloqueio de pacotes enviados a partir do *host* 3 utilizando protocolo TCP, tendo como destino a porta 8080 do *host* 4 e a sexta regra, o bloqueio do h1 e h3 utilizando protocolo UDP e porta 22. Os resultados são demonstrados no quadro 4. Destaca-se que para a obtenção dos resultados, foi utilizado o *software* Iperf.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Quadro 4: Resultado bloqueio TCP e UDP

Regra 5:

Teste TCP sem a regra Firewall

iperf -c10.0.0.03 -p8080 -i1

Client conneting to 10.0.0.3, TCP port 8080

TCP window size: 85.3 KByte (default)

---

[23] local 10.0.0.4 port 54958 connected with

10.0.03 port 8080

[ID] Interval Transfer Bandwidth

[23] 0.0- 1.0 sec 1.40 GBytes 12.0 GBytes/sec

Teste TCP com a regra Firewall

---

iperf -c10.0.0.03 -p8080 -i1

connet failed: Connection timed out

Regra 6:

Teste UDP sem a regra Firewall

[23] local 10.0.0.3 port 41957 connected with 10.0.0.1 port 22
[ID] Interval Transfer Bandwidth
[23] 0.0- 1.0 sec 129 KBytes 1.05 MBits/sec

Teste UDP com a regra Firewall

[23] local 10.0.0.3 port 41957 connected with 10.0.0.1 port 22
[ID] Interval Transfer Bandwidth
[23] 0.0- 1.0 sec 64.6 KBytes 529 kBits/sec
[23] Sent 893 datagrams
read failed: connection refused

Fonte: Autores, 2016.

Na avaliação da regra para o bloqueio do protocolo TCP, observa-se que a mensagem retornada pelo Iperf é de falha na conexão. Por esta razão, constata-se no

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

quadro acima, que mesmo criada a regra de firewall para o protocolo UDP, o Iperf ainda gera tráfego por algum tempo até a conexão ser recusada.

Por último, são apresentados no quadro 5, os resultados da regra 7, criada para bloqueio de hosts h3 e h6, para o tráfego de pacotes transportados pelo protocolo ICMP, utilizando como origem e destino o endereço mac dos computadores. Para apresentar os resultados, foram utilizados o comando ping e o software Iperf. Analisando os resultados, é possível verificar o bloqueio pelos dois macs. O Iperf foi empregado para gerar tráfego transportado pelo protocolo TCP, que foi aplicado para demonstrar uma contra prova da efetividade do bloqueio. Desta forma, evidencia-se que a restrição proposta pelo protocolo de transporte ICMP, se deu de forma efetiva, porém a conectividade entre os hosts, não foi afetada para outros meio de transporte, culminando com o êxito do objetivo deste teste.

Quadro 5: Resultado ICMP

```
mininet> pingall
* Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 X
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 X h4 h5
* Results: 6% dropped (28/30 received)


  mininet> iperf
  mininet> iperf h3 h6
  * Iperf: testing TCP bandwidth between h3 and h6
  * Results: ['12.1 Gbits/sec', '12.1 Gbits/sec']
```

Fonte: Autores, 2016.

5 Considerações Finais

Este artigo buscou apresentar os principais conceitos de Redes Definidas por *Software* e do protocolo *OpenFlow*. A criação de um software que tivesse autonomia na gerência e flexibilidade para as alterações de fluxo necessárias e que facilitasse também a escalabilidade das redes, fez-se necessário, tanto pela expansão de fluxo de dados, quanto pela complexidade elevada das redes, ou seja, a personalização das redes é algo a ser considerado.

As SDN foram criadas para tal personalização e para isso, utiliza-se o protocolo *OpenFlow*, que permite a realização deste controle de forma centralizada pelo software denominado *Controller*, localizado em um dispositivo externo que pode ser

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

um servidor. Com esta modificação, foi possível conseguir o direcionamento do fluxo de dados de uma rede, através de um único ponto, com agilidade e facilidade de fazer qualquer mudança. As Redes Definidas por *Software*, trouxeram uma evolução na estrutura das redes, servindo como forma alternativa, paralela ou agregada as Redes Legadas. O seu crescimento e sua propagação, tem feito com que fabricantes de *softwares* proprietários, criem aplicações compatíveis com o protocolo *OpenFlow* em seus equipamentos. Também é crescente o desenvolvimento de controladores no meio acadêmico, disseminando o conceito de SDN e aumentando a comunidade de colaboradores dos *softwares* de código aberto.

A pesquisa buscou aplicar os conceitos de SDN e constatar, que por intermédio do protocolo *OpenFlow* e do controlador *Floodlight*, o direcionamento de dados trafegados na rede, pautados na segurança da mesma, através de regras de *firewall*, se fez de forma efetiva e estável. Dos quesitos analisados, levando em consideração a rede simulada, objeto deste trabalho, todos demonstraram que a unificação e simplificação do gerenciamento da rede, não só foi possível, mas também apresentaram a flexibilidade nas diversas alterações propostas, certificando a eficiência da utilização de seus componentes na forma de comunicação de dados, porém, testes e análises mais aprofundados, utilizando equipamentos físicos para a montagem de protótipos de redes se fazem necessários.

Os autores tiveram dificuldades na tentativa de utilizar equipamentos físicos, que pudessem engrandecer o trabalho. Especificando, buscou-se utilizar um *Access Point*, marca *TP-LINK*, modelo 740N, com a instalação do *software* de código aberto, *OpenWRT*, mas no ato da compilação do protocolo *OpenFlow*, erros recorrentes impediram a utilização deste equipamento. Ainda, na tentativa de uso de meio físico, buscou-se o *switch Mikrotik*, porém houve incompatibilidade do protocolo *OpenFlow* entre a versão utilizada neste equipamento e o do controlador *Floodlight.*

Para trabalhos futuros, sugere-se a implementação de outros serviços no controlador utilizado neste trabalho, e um comparativo entre SDN e Rede Legada utilizando equipamentos que conduzam ao controle unificado da rede utilizando o protocolo *OpenFlow.*

6 Referências

ANDRADE, Arich. **Entendendo Software-Defined Network (SDN).** 2014. Disponível em: <https://sdnstudent.wordpress.com/entendendo-software-defined-network-sdn/>. Acesso em: 25 jul. 2016.

COSTA, Lucas Rodrigues. **OpenFlow e o Paradigma de Redes Definidas por Software.** 2013. 143 f. Monografia (Graduação) - Curso de Computação - Licenciatura, Ciência da Computação, Universidade de Brasília, Brasília, 2013. Disponível em: <http://bdm.unb.br/bitstream/10483/5674/1/2013_LucasRodriguesCosta.pdf>. Acesso em: 08 ago. 2016.

DIAS, Viviane Borges; SOUZA, Girlene Santos de; SANTOS, Anacleto Ranulfo dos. **Metodologia da Pesquisa Científica: A construção do conhecimento e do pensamento científico no processo de aprendizagem**. 2013. Disponível

em: <https://books.google.com.br/books/about/Metodologia_da_pesquisa_cient%C3%ADfica_a_co.html?hl=pt-BR&id=fba8AQAAQBAJ>. Acesso em: 11 nov. 2016.

Floodlight Is an Open SDN Controller. 2016. Disponível em: <http://www.projectfloodlight.org/floodlight/>. Acesso em: 12 nov. 2016.

FONTES, Ramon. **Visual Network Descripition (VND).** Disponível em: <http://www.ramonfontes.com/visual-network-description/>. Acesso em: 12 nov. 2016.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010.

JUNIOR, José Vitor de Araújo; TONON, Lucas da Silva; SANTOS, Letícia Ferreira da Silva; SILVA, Vinícius Gonzaga da. **PORQUE UTILIZAR REDES BASEADAS EM SOFTWARE (SDN – OPENFLOW).** 2013. 107 f. TCC (Graduação) - Curso de Análise e Desenvolvimento de Sistemas e Ciência da Computação, Centro Universitário Carioca, Rio de Janeiro, 2013. Disponível em: <http://www.ebah.com.br/content/ABAAAgoAYAL/sdn-openflow#>. Acesso em: 14 nov. 2016.

KUROSE James F., ROSS, Keith W. **Redes de Computadores e a Internet.** 5 ed. São Paulo: Pearson, 2010.

LANTZ, Bob; HANDIGOL, Nikhil; HELLER, Brandon; JEYAKUMAR, Vimal. **Introduction to Mininet.** 2015. Disponível em:

<https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>. Acesso em: 13 nov. 2016.

KERNER, Michael Sean. **Big Switch Shines Open Source Floodlight OpenFlow Controller on OpenStack).** Disponível em: <http://www.internetnews.com/infra/big-switch-shines-open-source-floodlight-openflow-controller-on-openstack.html>. Acesso em: 15 nov. 2016.

MACKEOWN, Nick; WINSTEIN, Keith. CS244: Advanced Topics in Networking. 2010. Disponível em: <http://yuba.stanford.edu/cs244wiki/index.php/Overview>. Acesso em: 04 dez. 2016.

MODA, Carlos Spinetti. **Uma Proposta de Redirecionamento de Fluxos de Rede Usando OpenFlow para Migração de Aplicações entre Nuvens.** 2014. 93 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos, São Carlos, 2014. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/573>. Acesso em: 30 jul. 2016.

OLIVEIRA, Natália Queiroz de. **Emprego de SDN para o Balanceamento de Carga em Redes de Computadores com Suporte a Múltiplos Caminhos.**

2014. 102 f. Dissertação (Mestrado) - Curso de Sistemas e Computação, Departamento de Ciência e Tecnologia, Instituto Militar de Engenharia, Rio de Janeiro, 2014. Disponível em: <http://www.comp.ime.eb.br/pos/images/repositorio dissertacoes/2014_Natalia_Queiroz.pdf>. Acesso em: 17 jul. 2016

OLIVEIRA, Davis Victor Feitosa de. **Redes Definidas por Software: A evolução das arquiteturas de redes.** Disponível em: <http://www.serpro.gov.br/tema/artigos-opinioes/redes-definidas-por-software-a-evolucao-das-arquiteturas-de-redes> Acesso em: 02 jun. 2016.

OPEN NETWORKING FOUNDATION. **Software-Defined Networking: The New Norm for Networks**. 2012. Disponível em: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf> Acesso em: 08 jul. 2016.

PANES, Guilherme Gonsales. **Firewall Dinâmico: uma implementação cliente/servidor.** 2011. 71 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Instituto de Biociências, Letras e Ciências Exatas, Universidade Estadual Paulista, São José do Rio Preto, 2012. Disponível em: <http://repositorio.unesp.br/handle/11449/89341>. Acesso em: 02 dez. 2016.

RFC_768. **User Datagram Protocol.** 1980. Disponível em: <https://www.ietf.org/rfc/rfc768.txt>. Acesso em: 18 jun. 2014.

ROTHENBERG, C. E, Nascimento, M. R, Salvador, M. R, Magalhães, M. F. **OpenFlow e redes definidas por software: um novo paradigma de controle e inovação em redes de pacotes**. 2011. Disponível em: <http://www.cpqd.com.br/cadernosdetecnologia/Vol7_N1_jul2010_jun2011/pdf/artigo6.pdf> Acesso em: 02 jul. 2016.

SEMEDO, Gonçalo Miguel Alves. **Load Balancing In Real Software Defined Networks.** 2014. 62 f. Dissertação (Mestrado) - Curso de Engenharia Informática, Departamento de Informática, Universidade de Lisboa - Portugal, Lisboa, 2014. Disponível em: <http://hdl.handle.net/10451/16052>. Acesso em: 12 jul. 2016.

SILVA, Gleydson Mazioli da. **Guia Foca GNU/ Linux.** 2010. Disponível em: <http://www.guiafoca.org/page_id=240>. Acesso em: 20 jul. 2016.

VIEGAS, Diogo Ruviaro. **Um Estudo Experimental Dos Protocolos Tcp, Sctp E Xtp.** 2008. 96 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2008. Disponível em: <http://www.dominiopublico.gov.br/download/texto/cp053019.PDF>. Acesso em: 13 nov. 2016.

Samba 4 como alternativa viável ao Microsoft Active Directory

Rodrigo Gonçalves Mengue[1], Thainá Farias Cardoso[1], Jéferson Mendonça de Limas[1], Marcos Henrique de Morais Golinelli[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia Catarinense – Campus Avançado Sombrio – Sombrio – SC – Brasil.

rodrigogoncalvesmengue@gmail.com,
thainafarias@hotmail.com,
{jeferson.limas,marcos.golinelli}@sombrio.ifc.edu.br

Resumo. *O armazenamento de informações com segurança, é um dos principais objetivos almejados pelas empresas; e para este fim, é necessária a utilização de softwares adequados. No mercado de soluções para armazenamento e disponibilização de arquivos, existem as soluções da Microsoft, como Active Directory, e a alternativa livre, o Samba. Na versão 4, o Samba promete desempenhar o mesmo papel do Active Directory e o objetivo deste trabalho foi verificar se o Samba 4 pode ser uma alternativa viável ao Microsoft Active Directory. Para descobrir se existe verdade nesta proposta, foi utilizada a metodologia de pesquisa aplicada, exploratória e experimental, no qual foram definidos os critérios para análise, partindo da instalação, passando por usabilidade e chegando aos custos. O Samba 4, segundo os critérios adotados nesta pesquisa, pode ser considerado uma boa alternativa ao Active Directory, por cumprir todas as características abordadas de forma satisfatória, com destaques na usabilidade e custos.*

Abstract. *Safe storage of information is one of the main objectives pursued by companies, and for this purpose it is necessary to use appropriate software. In the market for solutions for archiving and file availability there are solutions from Microsoft, Active Directory, and the free alternative, Samba. In version 4, Samba promises to play the same role as Active Directory and the purpose was to verify that Samba 4 can be a viable alternative to Microsoft Active Directory. In order to find out if there is truth in this proposal, the methodology of applied, exploratory and experimental research was used, where the criteria for analysis were defined, starting from the installation, through usability and arriving at costs. Samba 4, according*

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

to the criteria adopted in this research, can be considered a good alternative to Active Directory due to fulfilling all the characteristics approached in a satisfactory way with highlights in usability and costs.

1. Introdução

Em um ambiente corporativo, as informações são responsáveis pela garantia de integridade dos serviços prestados ao cliente, o que faz delas um requisito importante. Caso essas informações, que geralmente são sigilosas, sejam perdidas, ocasionando uso indevido, poderão comprometer a relação da empresa com seus clientes. Para prevenir a possibilidade de ocorrerem tais fatos, é necessária a realização de controles de contas, gerenciamento de arquivos e permissões de acesso (PEREIRA, LIRA e SILVA, 2013).

O uso de ferramentas capazes de limitar e controlar o acesso às informações, são essenciais nas empresas, visto a necessidade de manter a integridade, disponibilidade e confiabilidade dos dados na empresa, os quais, considerados princípios básicos da segurança da informação (PEREIRA, LIRA e SILVA, 2013).

Muitas vezes, empresas e organizações carecem de uma administração de rede para gerenciar e controlar acesso às informações, devido ao alto custo de investimento necessário. Isso se deve, sobretudo, pelo valor de implementação, licenças dos sistemas operacionais e as ferramentas utilizadas para o controle e segurança da informação (CASTRO, 2007).

Das tecnologias existentes no cenário de Redes de Computadores, principalmente as quais utilizam *softwares* da Microsoft, a mais utilizada para sanar estes problemas é o *Active Directory*. Trata-se de uma ferramenta de código fechado proprietária da *Microsoft*, com a característica de realizar funções baseadas no conceito de segurança da informação, ou seja, controle e gerenciamento de contas e políticas de acesso (GROSS, 2015).

Contudo, o Samba 4, emerge como uma opção *open source* em relação ao *Active Directory*. Atualmente na sua versão 4, a qual possui novas características, permite a um servidor *Unix* prover serviços de diretórios em rede para sistemas operacionais Linux e Windows (GROSS, 2015).

Sendo assim, este artigo tem como objetivo demonstrar a viabilidade do uso e/ou transição entre sistemas de diretórios - *open source versus closed source*, contemplando-se em especial os custos de instalação, administração e aquisição de licenças e software, do mesmo modo, sendo necessário conferir aspectos de instalação e configuração do serviço, verificando a viabilidade entre funções *versus* custos para empresas.

Para realizarmos os objetivos propostos, foram selecionados o *Active Directory* e o Samba 4. O *Active Directory* é disponibilizado e comercializado pela *Microsoft*, possuindo como motivação de sua escolha, o fato de o serviço ser conceituado e reconhecido na gerência de domínios e informações. Já o Samba 4, disponibilizado pela organização Samba, possui as características do *Active Directory*,

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

possuindo o diferencial em relação ao licenciamento, despesas e compatibilidade com diversos SO's (sistemas operacionais) linux. Dentre estes, o *Debian* é uma alternativa conceituada e com vasta documentação para uso junto ao Samba 4.

Este artigo está organizado em seções, sendo divididas em: Serviço de diretório, metodologia, procedimentos, análises, considerações finais e referências.

2. Serviço de Diretório

Segurança da informação, conforme a NBR ISO/IEC 27002:2013, é definida como a preservação da disponibilidade, confiabilidade e integridade contra os ativos que são objetos de ameaça, podendo estes ser acidentais ou intencionais. Sendo assim, existem alguns serviços que podem atuar na preservação dos pilares fundamentais da segurança da informação, auxiliando no controle de acesso a sistemas e informações, com o uso de políticas de acesso e propriedade da informação. Um exemplo disso, são os denominados serviços de diretórios.

Serviço de diretório, é composto primordialmente, pelo armazenamento hierárquico da informação, o qual permite agilidade e facilidade na recuperação destas, bem como controle de nomes e domínios, segurança e localização de dados. Permite ainda, funções relacionadas à gestão e infraestrutura de redes, que auxiliam na busca por informações (CRUZ et al., 2004). No mercado atual, duas ferramentas se destacam em prover serviços de diretórios: *Active Directory* e o Samba 4.

Active Directory é denominado como "um serviço de diretório extensível para gerenciar os recursos da rede de modo eficiente. Para tanto, o serviço de diretório armazena informações detalhadas sobre cada recurso de rede, o que facilita ainda mais a pesquisa básica e a autenticação" (STANEK, 2008, p. 1023).

Em contrapartida, tem-se o Samba, que é um software livre que disponibiliza serviço de arquivos e impressão aos clientes que suportam o protocolo CIFS/SMB, como é o caso também dos sistemas operacionais da *Microsoft*. Está disponível gratuitamente para instalação e utilização em sistemas Linux através de repositórios na versão 3, tendo como principal característica, o compartilhamento de arquivos e impressoras, deixando de lado a arquitetura e o modelo de serviço de diretórios. Contudo, encontra-se em arquivos para compilação a versão 4, denominada Samba 4, com enfoque em serviços de diretórios, tornando-se uma alternativa com funcionalidades semelhantes ao *Active Directory* da *Microsoft*. (SAMBA.ORG, 2016)

2.1 Active Directory

Active Directory, é um serviço de diretório proprietário da Microsoft, que tem a capacidade de centralizar informações dos objetos de uma rede de computadores ligada a ele, facilitando o acesso desses dados para os administradores e usuários. Esses objetos são os recursos compartilhados, como: arquivos, impressoras, contas de usuários, dos microcomputadores da rede e de servidores. Dessa forma, os administradores conseguem gerenciar a rede de uma forma mais centralizada, podendo definir quais objetos os usuários autenticados podem ter acesso. (MICROSOFT, 2016)

O *Active Directory,* utiliza alguns protocolos para seu funcionamento, sendo

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

eles:

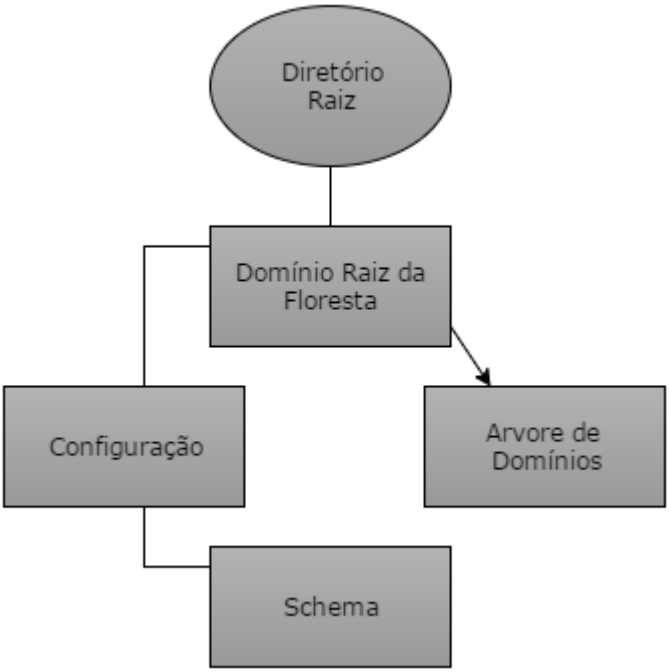
- **LDAP (*Lightweigth Directory Access Protocol*):** Principal protocolo utilizado pelo *Active Directory*, o *LDAP*, roda diretamente sobre o TCP/IP para conexões que tem a confirmação do recebimento de dados; e sobre UDP, para conexões sem confirmação. Os recursos da rede são organizados de forma hierárquica. Primeiramente temos o diretório raiz, posteriormente a rede da empresa e o seu departamento, e por último o computador cliente e os recursos de rede compartilhados. (VIGNATTI, 2007). Traz como principal vantagem, a facilidade de localização de informações na rede. Cada funcionário que possui uma conta autenticada, pode compartilhar arquivos e cadastrar informações (VIGNATTI, 2007).

- **RPC (Chamada Remota de Procedimento):** O serviço de diretórios da *Microsoft*, utiliza este protocolo para o gerenciamento de controladores de domínio, e para realizar comunicação com a interface MAPI (Interface de programação de aplicativo de mensagens). Além de ser um mecanismo eficiente para efetuar a chamada de funcionalidades e troca de dados, pode ocorrer em uma rede local, no mesmo computador ou na rede mundial de computadores (NASCIMENTO, 2010).

- **SMTP (*Simple Mail Transfer Protocol*):** protocolo utilizado para controladores de domínio que não possuem comunicação permanente. Prática sem muita recomendação de utilização (NASCIMENTO, 2010).

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Figura 1 -Arquitetura Lógica.



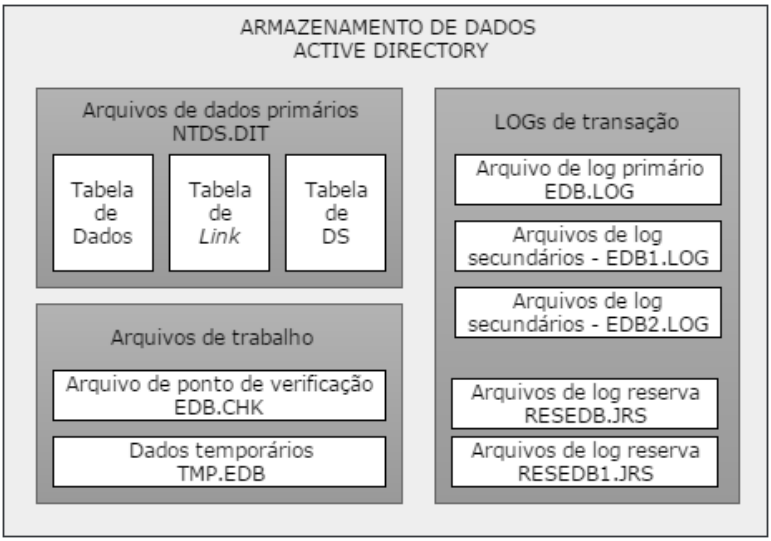
Fonte: Os autores, 2016.

Active Directory, é dividido em duas arquiteturas de armazenamento de dados, sendo elas lógica e física. A estrutura lógica, funciona conforme o modelo hierárquico do LDAP, sendo derivada do *Schema*, onde são localizados os arquivos necessários para a organização das informações, que definem os atributos de um objeto, como: e-mail, nome e senha. Sendo assim, a forma como os objetos são organizados cria uma estrutura em árvore, como ilustra a Figura 01. Cada objeto precisa de um pai, e as informações desse pai sempre são armazenadas juntamente ao objeto. (NASCIMENTO, 2010).

Já a estrutura física, como podemos observar na Figura 02, fundamenta-se do arquivo de banco de dados principal (NTDS.DIT) do *Active Directory*, dos seus arquivos de log, assim como seus arquivos temporários.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Figura 2 - Arquitetura Física.



Fonte: Adaptado de Microsoft, 2016.

Conforme Stanek (2008) os arquivos apresentados possuem as seguintes funções:

- **NTDS.DIT:** arquivo do banco de dados físicos, que guarda todo o conteúdo armazenado. Possui três tabelas indexadas, as quais são: tabela de dados, de *link* e de descrições de segurança.
- **EDB.CHK:** rastreia até onde as transações foram confirmadas no arquivo de log.
- **TMP.EDB**: espaço temporário para processo de transações.
- **EDB.LOG:** arquivo onde as transações são escritas, antes de serem gravadas no banco de dados.
- **Arquivos de log secundário:** Arquivos de logs adicionais.
- **Arquivos de log reserva:** Arquivos utilizados para reservar espaço para os arquivos de logs secundário, caso o primário fique cheio.

Ainda, conforme Stanek (2008), o armazenamento de objetos no banco de dados do *Active Directory* não possui um limite estimado, permitindo assim, uma maior escalabilidade.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

2.2 Samba 4

Conforme Morimoto (2009), o Samba é um serviço que permite realizar o compartilhamento de arquivos e impressoras em Sistemas Operacionais Linux, para microcomputadores com Sistema Operacional *Microsoft* Windows. Conforme o Site Oficial do Projeto Samba, ao utilizá-lo em um Servidor Linux, será possível fazer a autenticação de usuários, compartilhar arquivos e atuar como um controlador de domínios, exatamente igual a um Servidor Windows.

Samba 4, fornece todas as opções das versões anteriores, adicionando a funcionalidade de serviço de diretórios. Sua configuração é feita através do arquivo principal "smb.conf", para compartilhamento de arquivos e pastas, juntamente a ferramenta samba-tool, responsável pela parte administrativa via linha de comando, no terminal do servidor onde se encontra instalado. Em sistema Windows, é possível utilizar a ferramenta RSAT (Remote Server Administration Tools), disponibilizada pela Microsoft, a qual permite o gerenciamento de usuários, permissões em pastas e políticas de segurança em modo gráfico, realizado pela integração do Samba 4 com esta ferramenta (SAMBA-ORG, 2016).

No Samba 4, é possível realizar mais funções do que apenas o gerenciamento de grupos e usuários, como centralizar a autenticação dos usuários em vários serviços, como: *Proxy*, *e-mail* e *web*. Para a realização desta autenticação, tais serviços utilizam a base *LDAP* (BURGARDT, 2010).

Salientando apenas, que o Samba 4 utiliza como mecanismo padrão a base de dados LDB, a qual é caracterizada por um banco de dados baseado em LDAP. Contudo, se necessário, pode-se utilizar serviços baseados em LDAP, sendo o mais popular na plataforma *Unix*, o *OpenLDAP* (LDB-Samba, 2016).

3. Metodologia

De acordo com Rauen (2015), a pesquisa pode ser definida como procedimentos aplicados na busca, apuração ou exploração, tendo a finalidade de conhecimento em torno de um fato ou fenômeno parcialmente ou completamente desconhecido. Nestas circunstâncias, define-se pesquisa como:

> Um conjunto de ações sistemáticas, minuciosas, completas, sustentadas epistemologicamente e metodologicamente, com as quais, partindo-se de evidências disponíveis, de teorias científicas ou de intuições racionais, descobrem-se novos fatos ou fenômenos ou compreendem-se fatos ou fenômenos até então considerados complexos ou inadequadamente explicados (RAUEN, 2015, p. 123).

Sendo assim, o delineamento do estudo será explicitado a seguir.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

3.1 Métodos

Conforme a natureza da pesquisa, o método utilizado definiu-se em pesquisa aplicada que tem como objetivo produzir conhecimento através de uma aplicação prática para resolução de problemas, abrangendo assuntos que envolvem verdades e interesses locais. Sendo utilizado para atender os objetivos do estudo a pesquisa exploratória, a qual fundamenta-se em conhecer melhor um problema a fim de torná-lo mais compreensível para que ocorra o aprimoramento de uma ideia. Possui um planejamento adaptável, ou seja, possibilita o acatamento das diversas questões que se relacionam com a situação estudada. (GIL, 2010).

Em relação aos procedimentos da pesquisa, trata-se de uma pesquisa experimental que, conforme Severino (2007), pega o próprio objeto em sua realidade e põe em condições técnicas de manipulação e observação experimental em laboratórios, na qual situações são criadas para seu tratamento. Com isso, o pesquisador seleciona variáveis, nas quais testará suas funcionalidades, através de formas de controle.

Com base na compreensão dos mecanismos utilizados, caracterizou-se a pesquisa experimental mediante as comparações que serão realizadas; podendo apontar nos resultados, os pontos fortes e fracos de cada uma das ferramentas, analisando suas funcionalidades, instalações e custos, ajudando na escolha de um administrador de redes.

3.2 Materiais

Conforme os objetivos da pesquisa, criou-se um ambiente para a aplicação dos métodos e busca de resultados, possuindo os seguintes equipamentos: Servidor 01, Servidor 02, Cliente e Administrador Cliente. Pode-se observar no Quadro 01, as definições de cada máquina, informando o sistema operacional, serviço e configuração.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Quadro 1 - Materiais Utilizados.

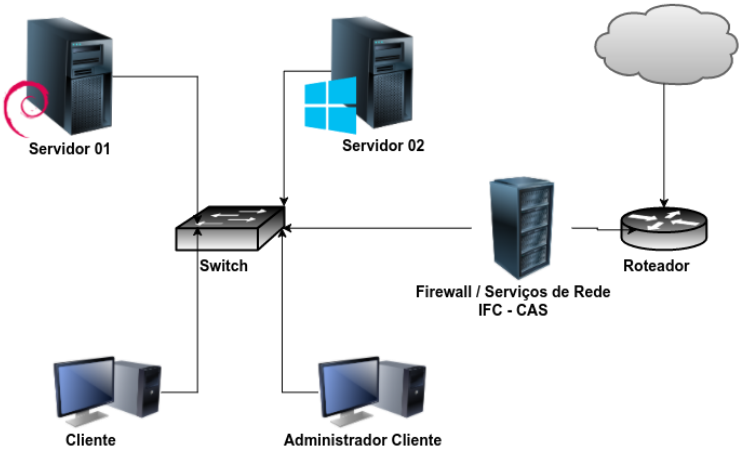
	Sistema Operacional	**Serviço**	**Configuração**
Servidor 01	*Debian* 8 *Jessie*	Samba 4.4.5	HD: 250GB Processador: Core i5 Memória Ram: 4GB
Servidor 02	*Windows Server* 2012 R2	*Active Directory* AD DS	HD: 250GB Processador: Core i5 Memória Ram: 4GB
Cliente	*Windows* 7 Professional	-----	HD: 250GB Processador: Core i5 Memória Ram: 4GB
Admin. Cliente	*Windows* 7 Professional	Ferramenta RSAT	HD: 250GB Processador: Core i5 Memória Ram: 4GB

Fonte: Os autores, 2016.

No andamento da pesquisa, foi necessário a montagem de um ambiente de rede para a interconexão dos servidores e clientes, bem como o acesso à rede mundial de computadores para a obtenção dos pacotes de instalação dos serviços a serem compilados e configurados. A montagem do ambiente físico, seguiu a topologia proposta na figura 03.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Figura 3 - Topologia Lógica da Rede.



Fonte: Os autores, 2016.

Para a montagem física, foi utilizado o Laboratório 37 do Instituto Federal Catarinense – *Campus* Avançado Sombrio (IFC – CAS) Figura 4. Utilizou-se um switch 3COM Baseline 2226-SPF, mantido nos Servidores e Clientes, os serviços já providos neste ambiente, sendo eles: serviço de DHCP (Dynamic Host Configuration Protocol), firewall, dentre outros de menor impacto no ambiente de testes.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Figura 4 - Laboratório.

Fonte: Os autores, 2016.

Salientamos que, para o ingresso dos clientes ao controlador de domínios, foram realizadas a configuração manual de DNS nos microcomputadores clientes. Nos microcomputadores servidores, foi necessário deixar a configuração do IP, bem como as opções de forma manual, a fim de evitar a alteração de arquivos de configurações internos.

4. Procedimentos

Com a finalidade de alcançar o propósito deste artigo, foi estipulado um escopo dos aspectos necessários a serem instalados e configurados, garantindo assim, a possibilidade da análise proposta.

Os primeiros testes de implementação, foram realizados com o auxílio de máquinas virtuais, permitindo realizar os testes básicos, ajudando a definir os componentes importantes a serem implementados na versão final. Para isto, foram utilizadas VM's (*Virtual Machines*), com o software Virtualbox 5.

Ao final da fase de definição dos requisitos, foi utilizado o Laboratório 37 do IFC – CAS, para implementação da solução nos microcomputadores. Conseguinte, optou-se na divisão do processo de pesquisa em três etapas, as quais estão descritas a seguir: prática, a instalação e requisitos de comparação.

4.1 Instalação e configuração do *Active Directory*

A instalação do *Active Directory*, no Windows *Server* 2012 R2, envolve um conjunto de passos que devem ser seguidos. Primeiramente, é necessário

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

acessar o assistente de adição de funções e recursos, localizado no painel "gerenciador do servidor" e adicionar uma nova função.

Ao criar uma nova função, é preciso selecionar um servidor de destino e posteriormente escolher a opção de serviço desejada. Neste caso, foi selecionada a opção "*Active Directory Domain Services (AD DS)"* e instalada. Ao término da instalação do AD DS, para chamar o assistente de instalação, é necessário clicar na opção "Promover este servidor a controlador de domínio", que aparece na última tela da instalação, descrevendo--se que tudo ocorreu como deveria ou acessar o painel de gerenciador do servidor, na qual esta opção estará disponível.

Após chamar o assistente de configuração, o primeiro passo é adicionar uma nova floresta, para que seja indicado o nome raiz do domínio. Subsequentemente, os níveis de floresta são definidos. E, na próxima tela, a opção do serviço de DNS deve ser selecionada, definindo também uma senha para o administrador.

Para finalizar a configuração, é necessário definir um nome NETBIOS, para que seja possível a comunicação do Windows *Server* 2012 com as outras versões, assim como, indicar um caminho onde a base de dados do AD DS será armazenada, os arquivos de LOG e os arquivos do *Sysvol* (*System Volume* – volumes do sistema).

O *Active Directory*, trabalha juntamente ao serviço DNS para realizar a resolução de nomes. Se o serviço não estiver instalado no microcomputador, o assistente de instalação o instalará automaticamente. Para que o DNS funcione corretamente, é necessário checar se no momento da instalação foi criado uma zona de pesquisa direta, na qual fará a resolução de nomes em endereço IP.

Caso essa zona não tenha sido criada, isso dever ser feito acessando o menu "gerenciador do servidor > ferramentas" selecionar DNS e criar uma zona primária. Lembrando que o nome da zona deve ser igual ao nome que será colocado no domínio, para que eles possam ser integrados.

Após o cumprimento de todos esses passos, a máquina reiniciará. Na tela de logon, estará a opção DOMÍNIO/Administrador, na qual o serviço de domínio já estará operacional.

4.2 Instalação e configuração Samba 4

A instalação e configuração do servidor Samba 4, foi realizada utilizando-se o sistema operacional Debian 8 - Jessie. Sendo este escolhido, devido ao reconhecimento do sistema operacional ser uma distribuição Linux que preza pela estabilidade do sistema, possuir suporte ao serviço proposto, uma distribuição Linux conceituada além de apresentar uma gama de materiais de apoio, assim como, manuais e tutoriais no âmbito da rede mundial de computadores.

Sendo assim, foi realizada a instalação do SO Debian, seguindo as diretrizes necessárias ao funcionamento do serviço:

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

- **Particionamento:** a tabela de partições para o sistema, foi definida conforme o Quadro 2, tendo a definição do ponto de montagem, tipo de arquivos e tamanho de cada partição;

Quadro 2 - Partições Sistema

Ponto de Montagem	Sistema de Arquivos	Espaço Alocado
/	Ext4	100 GB
/samba	Ext4	150 GB
swap	---	2 GB

Fonte: Os autores, 2016.

- **Serviços durante a instalação:** foi instalado o pacote openssh-server, o qual permite o acesso remoto ao servidor para uso e manutenção do sistema operacional.

Após a instalação base do sistema operacional, foram iniciados os procedimentos de instalação dos requisitos para a compilação - implementação do serviço através do código fonte - e configuração do Samba 4 na sua versão atual, 4.4.5 estável. Foram seguidos os seguintes passos:

- **Configuração do serviço de nomes de domínios local**, bem como a configuração de endereço ip manual, evitando assim, erros de resolução de nomes e que serviços do SO alterem IP's e endereços internos, bem como alterações das definições manuais efetuadas.

- **Configurar a partição */samba* para suporte à ACL's** (Access Control List); permitindo assim, o uso de permissões avançadas para usuários, necessárias para o funcionamento correto do serviço de compartilhamento de arquivos.

- **Download e compilação do Samba 4:** para que o serviço entre em funcionamento, foram instaladas as dependências, realizado o download do código fonte na versão já mencionada, e realizado os procedimentos de compilação, processo padrão para instalação de programas. Segundo Mota Filho (2012, p.25), "compilar, é executar comandos que ajustam o código-fonte do programa ao nosso sistema, gerando um arquivo executável ideal para cada distribuição".

- **Instalação das dependências:** estas, são bibliotecas de funcionalidades, que se incorporam ao sistema operacional; permitindo assim, a instalação

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

e o funcionamento dos recursos. Para o serviço Samba 4, foram necessárias bibliotecas para seu funcionamento, tais quais, no Debian, podem ser adicionadas com o comando a seguir: "*apt-get install build-essential libacl1-dev libattr1-dev libblkid-dev libgnutls28-dev libreadline-dev python-dev python-dnspython gdb pkg-config libpopt-dev libldap2-dev dns libbsd-dev attr krb5-user docbook-xsl libcups2-dev libkrb5-dev libssl-dev python-software-properties acl quota".*

Após instalação finalizada, são necessários alguns passos para o início da operação básica do serviço. Estes passos, compreendem: provisionamento, inicialização, testes dos serviços e teste de logon.

- **O provisionamento**, é o ato de implantar o serviço, no qual são definidos parâmetros de funcionamento do serviço. Para realizar o provisionamento, aplicam-se alguns comandos no terminal do Sistema Operacional ou através de terminal remoto via SSH. Foi realizado no terminal, o seguinte comando: "*/usr/local/samba/bin/samba-tool domain provision --use-rfc2307 --use-xattrs=yes –interactive"***.**

Os parâmetros utilizados, estão explicados nos itens a seguir, seguidos da foto:

- ***domain provision:*** parâmetro para ativar o domínio;
- ***use-rfc2307:*** conforme a samba wiki, este atributo define a possibilidade de armazenar usuários e grupos em base de diretórios LDAP;
- ***use-xattrs=yes:*** habilita ao serviço o uso de Acl's estendidas junto ao sistema operacional, para controle das permissões e recursos em diretórios dentro do serviço.
- ***interactive:*** faz o provisionamento do serviço de forma interativa, no qual o usuário vai "respondendo" cada passo, estes são:
    - ***realm:*** solicita o FQDN (*Full Qualified Domain Name*), ou seja, o domínio que irá conter o serviço disponível;
    - ***Server Role:*** onde é informado se o domínio faz parte de alguma floresta já existente ou o primeiro controlador para o domínio, este definido como DC (Domain Controller);
    - ***DNS Backend:*** estabelece o serviço responsável por resolver os nomes dentro do domínio; e
        - ***Administrator password:*** onde é definido a senha para o usuário administrador, o qual se denomina com o login no serviço: ADMINISTRATOR.

Após o provisionamento, devemos iniciar o serviço no Sistema Operacional, que é realizado através do comando: "*/usr/local/samba/sbin/samba"***.**

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Figura 5 - Resultado do Provisionamento.

```
root@samba4:~# /usr/local/samba/bin/samba-tool domain provision --use-rfc2307 --use-xattrs=yes --interactive
Realm: tcc.local
 Domain [tcc]:
 Server Role (dc, member, standalone) [dc]:
 DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) [SAMBA_INTERNAL]:
 DNS forwarder IP address (write 'none' to disable forwarding) [172.16.37.7]: 172.16.37.1
Administrator password:
Retype password:
Looking up IPv4 addresses
                    ...
Once the above files are installed, your Samba4 server will be ready to use
Server Role:           active directory domain controller
Hostname:              samba4
NetBIOS Domain:        TCC
DNS Domain:            tcc.local
DOMAIN SID:            S-1-5-21-3992918656-54709893-2378311902
```

Fonte: Os autores, 2016.

Após a inicialização do serviço Samba 4, pode-se realizar alguns testes; assim, verificamos se os serviços complementares estão funcionando como o previsto, como: o serviço de resolução de nomes de domínio (serviço interno do samba); o serviço de chaves do Kerberos e o registro do LDAP.

Com o serviço instalado, já pode ser realizado o ingresso de uma máquina Windows ao domínio configurado recentemente utilizando o usuário ADMINISTRADOR. A partir deste momento, pode-se instalar e utilizar a ferramenta RSAT para realizar mais configurações.

4.3 Configuração dos clientes

Nas máquinas clientes, foram realizadas as instalações do Sistema Operacional Windows 7 Profissional e ativadas as configurações de IP e DNS, para que houvesse comunicação com o servidor. Como especificado no Quadro 01, seção 3.2, haviam dois tipos de clientes, sendo eles: cliente e administrador cliente. Estes dois clientes foram adicionados ao domínio, para que estes permitam o login de contas de usuários, permitindo os testes com os dois serviços.

No microcomputador administrador cliente, foi necessário a instalação da ferramenta RSAT, disponível no próprio site da Microsoft, que permite que os administradores de TI gerenciem as funções do servidor através de um computador remoto (Microsoft, 2016).

O RSAT, permite gerenciar os servidores nativos da Microsoft. A documentação disponível no *wiki* Samba, recomenda a utilização desta ferramenta para o gerenciamento dos serviços do Samba 4. Nela, pode-se realizar a delegação de permissões de diretórios compartilhados, criação de unidades organizacionais, contas de usuários, gerenciamento de diretivas de grupo e os demais componentes no domínio. Assim, torna-se possível o controle do servidor Samba 4 por uma interface gráfica, não sendo necessariamente obrigatória a realização das configurações de controle através de comandos no terminal, tornando a gerência do Controlador de Domínios igual ao *Active Directory* da *Microsoft*.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

5. Análise

Para efetuar a análise de comparação entre as ferramentas, alguns critérios foram definidos, sendo estes: análise das funcionalidades, análise de instalação e manutenção do serviço e custos, que estão explicados a seguir.

5.1 Análise das funcionalidades

O serviço de diretórios padrão Microsoft, destaca-se por cinco funcionalidades bem nítidas, sendo elas: serviço de domínio, LDAP, certificados, federação e gerenciamento de direitos; sendo estas, inclusas no produto original, mas delineadas e correlacionadas aqui, para uma melhor compreensão.

É utilizado o serviço LDAP, para realizar o controle de contas através do serviço de domínio, como: o gerenciamento de contas de usuários, unidades organizacionais, grupos e computadores no domínio. O serviço de domínio, torna-se responsável pela definição do grupo de trabalho onde será realizado tal gerenciamento, sendo descrito como a zona de atuação, constituída pela infra-estrutura da empresa ou organização.

Junto a essas propriedades, há os serviços de certificados e federações. Estes, consistem na emissão e gestão de certificados, sendo utilizados em sistemas de segurança de software, no qual o serviço de federação expande a possibilidade do uso de chaves e certificados, através de redes altamente extensíveis. Exemplo disso é a internet, que permite o uso de computadores em várias plataformas ou associados à um sistema operacional.

Por fim, tem-se o serviço de gerenciamento de direitos, caracterizado pelas definições de direitos legais sobre ações desenvolvidas ou permissões dentro da arquitetura interna da empresa. Este, por sua vez, também se integra aos outros serviços, portanto, aplica-se dentro do domínio da empresa, junto com todas as questões de segurança. O serviço, caracteriza-se pelo gerenciamento de informações e permissões, bem como o uso de GPO's (Objeto de Política de Grupo) para a manutenção do direito de cada usuário.

Para tanto, este trabalho delinea-se com base às funções mais utilizadas em empresas de médio porte, sendo realizado uma análise juntos às empresas da região sobre as principais funções utilizadas em seus sistemas, verificando funções básicas, como: o gerenciamento de logon em domínio, o serviço de arquivos e o controle por GPO's para usuários e informações.

Sendo assim, a pesquisa é baseada nessas prioridades. Porém, com uma análise bibliográfica e conceitual, atráves da wiki do Samba 4 e de caracteríticas de seus plugins, nota-se a possibilidade de realizar todas funções do *Active Directory* no Samba 4.

Em relação à gestão das funções após a instalação, os dois serviços possuem similaridades. Isto se deve ao Samba 4 conseguir integrar-se a mesma plataforma de gerência do *Active Directory*, permitindo suprir os seguintes pontos:

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

- **Criação e gerenciamento das entidades na rede:** unidades organizacionais, usuários, grupos, computadores, impressoras, entre outras, podem ser realizadas da mesma forma que utilizando-se do serviço da Microsoft;

- **Compartilhamento de arquivos na rede:** consegue-se realizar o compartilhamento de arquivos com base em permissões por usuários ou grupos de usuários. No sistema da Microsoft, isto é realizado por meio da criação da pasta e compartilhando através do gerenciamento do computador, seguindo em pastas compartilhadas, disponível no próprio sistema da Microsoft. Para realização, através do Samba, é preciso a criação do diretório e atribuição do usuário do Samba 4, como proprietário do diretório. Em seguida, realiza-se o compartilhamento no arquivo smb.conf, para compartilhamento em rede. Após utilizar a ferramenta, no Windows, Gerenciamento do Computador com a ação de se Conectar à um Computador Remoto, acesar o computador e seguir em pastas compartilhadas, compartilhamento e efetuar as permissões conforme necessário;

- **Criação e gerenciamento de diretivas de grupos:** GPO's são definições de acesso a arquivos ou funções de um sistema Windows. Esta é uma ferramenta utilizada devido à possibilidade de definir normas de uso e segurança da informação dentro da empresa. Esta função, também é disponível na ferramenta RSAT; portanto, pode-se utilizar das GPO's em redes Windows com a utilização de servidores linux para seu gerenciamento.

Tomando como ponto de relevância o uso de GPO's na empresa, foi realizada a configuração do ambiente de testes, para testar esta ferramenta. De início, foram realizadas regras básicas de uso no servidor AD, como regras de bloqueio para trocas de papéis de paredes ou de temas no cliente. Após, foram realizadas as mesmas configurações no servidor Samba 4, nas quais funcionaram corretamente. Realizadas essas configurações básicas, foram feitos bloqueios de leitura ou escrita em discos removíveis, partições ou uso de ferramentas administrativas, como: o console CMD, ou o editor de registro no Windows. Todas as funções foram testadas nos dois servidores havendo sucesso em ambas tentativas. Apenas há um aspecto a notar-se, o Samba 4 precisa ter o serviço reiniciado para a aplicação das Políticas de Grupo, tal ação não é feita manualmente no *Active Directory*, porém com este detalhe o Samba 4 supri esta função corretamente também.

Devido à esta constatação, pode-se concluir que nos quesitos implementados, o Samba 4, corresponde à sua proposta de se equiparar ao serviço de diretórios Microsoft; garantindo assim, que o serviço se comporte de maneira satisfatória para o usuário que irá fazer a gerência e manutenção destas funções.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

5.2 Análise de instalação e gerência

Em relação a complexidade do uso e gerenciamento do serviço Samba 4 ao sistema proprietário da *Microsoft*, foram realizadas a instalação, configuração e gerência , podendo assim, analisar aspectos de implementação de cada um.

O *Active Directory*, mantém a premissa utilizada pela plataforma *Windows*, na qual a instalação de ferramentas é realizada através de uma interface gráfica, tornando o ato de instalação e configuração relativamente simples, para os profissionais da área., estabelecendo uma vantagem em relação ao Samba 4, o qual necessita a utilização de comandos no terminal do servidor para sua instalação.

Apesar disso, outra dificuldade que trazia desvantagem para o Samba era questão da gerência dos serviços, sendo efetuada também por linha de comando através do terminal. No entanto, a integração com a ferramenta RSAT da *Microsoft*, auxiliou nesse quesito, tirando esta desvantagem em comparação ao serviço da *Microsoft*.

O uso da ferramenta RSAT, portanto, propicia um ambiente de equivalência em relação à gerência das ferramentas, visto que as principais dificuldades encontradas nas versões anteriores, proferia justamente no gerenciamento do serviço.

Através destas análises, podemos identificar a possibilidade de propagação do Samba 4, visto que esta dificuldade foi amenizada, apesar da instalação ainda exigir um maior conhecimento do profissional que a realizará, pois este precisa de habilidades em Sistemas operacionais Linux.

5.3 Custos

O principal objeto de pesquisa deste artigo, consiste na avaliação de impactos em redes de médio porte com a transposição do uso de software proprietário à software livre. Para que isto seja viável, tem-se a necessidade de que haja compatibilidade das funções e resultados semelhantes entre estes dois serviços. Sendo assim, instalação, manutenção e serviços tornam-se variáveis de extrema importância ao contexto da pesquisa.

Em redes que utilizam softwares Microsoft, para que ela seja legalizada, tem-se a necessidade de que todos os computadores da rede possuam uma licença. Então, a máquina servidor, precisa de uma licença de uso para servidor, que é definida pela quantidade de núcleos que o servidor possui, sendo que atualmente o mínimo de licenças vendidas são oito. E, para as demais máquinas que acessam ao servidor, além da licença do sistema operacional que está em uso, é preciso também que cada uma das máquinas possuam uma licença CAL (*Client Access License* – licença de acesso para clientes), que é quem dá o direito de acesso aos arquivos do servidor de forma legalizada (MICROSOFT, 2016).

Para saber ao certo os valores destas licenças, contatamos via e-mail, cinco revendas oficiais da Microsoft, nas quais obtivemos respostas de três.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Analisando os orçamentos, notou-se a correlação dos preços com os seguintes componentes de uma rede: quantidade de usuários, quantidade de servidores e núcleos de processamento por servidor.

Dentre as respostas obtidas pelas empresas, optou-se pelo orçamento de apenas duas, devido à quantidade de licenças informadas pela terceira ser de apenas uma para cada produto, não correspondendo as expectativas decorrentes do estudo.

As informações repassadas pelas empresas, informaram que as licenças do Windows 7 Professional, não estão sendo mais comercializadas. Mas, há possibilidade de fazer um *downgrade* para o Windows 7; porém, a *Microsoft* não dará suporte. A versão do Windows *Server* sendo comercializada, é a 2016. Entretanto, ainda é possível fazer um downgrade para a versão 2012, com suporte da *Microsoft*. Importante salientar, que a escolha das versões utilizadas neste projeto, foi devido ao Instituto Federal Catarinense – *Campus* Sombrio possuir os sistemas operacionais licenciados apenas do Windows 7 e Windows *Server* 2012 R2 e não de suas versões posteriores. Mas, a base de instalação das ferramentas é similar em qualquer versão do sistema.

Então, a tabela 1 demonstra os valores das versões atualizadas dos produtos, definindo a empresa A como um orçamento para 25 usuários e a empresa B para 100 usuários.

Tabela 1 - Orçamento Licenças Windows

	Licença Servidor	**Licença CAL**	**Licença Windows 10**	**Total**
Empresa A	08 x R$ 434,00	25 x R$ 150,00	25 x R$ 706,00	R$ 24.872,00
Empresa B	08 x R$ 516,00	100 x R$ 136,00	100 x R$ 710,00	R$ 88.728,00

Fonte: Os autores, 2016.

Já o Servidor Linux, acompanhado do Samba 4, ser de uma comunidade de *software* livre, não possui a necessidade de compra de licença. Para que seu servidor seja legalizado, a única licença necessária será do Windows 10, para legalização de seus usuários. No mesmo esquema da Tabela 1, a Tabela 2 demonstra como ficaria o orçamento para o servidor Samba 4.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

Tabela 2 - Orçamento Licenças *Linux*

	Licença Servidor	Licença CAL	Licença Windows 10	Total
Empresa A	08 x R$ 0,00	25 x R$ 0,00	25 x R$ 706,00	R$ 17.650,00
Empresa B	08 x R$ 0,00	100 x R$ 0,00	100 x R$ 710,00	R$ 71.000,00

Fonte: Os autores, 2016.

Em relação a instalação e manutenção das ferramentas, conforme pesquisas realizadas com empresas no ramo, obteve-se uma média dos preços cobrados para efetivação dos serviços, conforme a tabela 3 demonstra.

Tabela 3 - Orçamento Instalação / Manutenção

	Preço Instalação	Preço Manutenção
Windows	R$ 1.100,00	R$ 350,00
Linux - Debian	R$ 1.500,00	R$ 500,00

Fonte: Os autores, 2016.

Em conformidade com os preços apresentados nas tabelas 1, 2 e 3, pode-se observar que optar por um serviço de diretórios com servidores Linux, ocasiona na diminuição dos custos que serão gastos por uma empresa, sendo que a diferença de preço, somente na parte de compra de licenças, é de 7.222,00 reais para 25 usuários e de 17.728,00 reais para 100 usuários.

Já na instalação e manutenção, o Samba 4 possui um custo maior em relação aos serviços da Microsoft, mesmo assim, se considerarmos somente a diferença da manutenção, que é de 150,00 reais, para chegar a diferença de 17 mil reais, são necessários 118 meses na empresa com 100 usuários e de 45 meses na empresa de 25 funcionários.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

6. Considerações finais

Após análise das ferramentas, notou-se que o Samba 4 pode ser uma alternativa frente ao *Active Directory*. Além de contemplar as funcionalidades testadas, traz a redução de custos, vantagem almejada pela maioria das empresas, sendo possível devido ao fato das licenças Microsoft apresentarem valores elevados. Porém, em contrapartida, os serviços de diretórios implementados utilizando servidores com sistema operacional Linux, são livres de licenças bem como custos da compra de software.

Observa-se também, que a compatibilidade da ferramenta RSAT com o servidor Samba 4, pode causar grande impacto na facilidade gerada pela ferramenta, aos responsáveis pela administração do serviço. Portanto, será possível controlar o serviço como se estivesse em um servidor da Microsoft, o que facilita o trabalho do mesmo.

Apesar da instalação do Samba 4 ainda ser considerada mais difícil que a instalação do *Active Directory*, devido ao fato de exigir conhecimento técnico quanto ao funcionamento do Linux, os critérios de comparação realizados demonstraram que o objetivo deste trabalho foi alcançado, tendo em vista que o Samba 4 é uma alternativa viável para empresas que estão iniciando seu processo de informatização ou que estão procurando reduzir custos de investimento em novas licenças para seu parque tecnológico. Esta viabilidade, é caraterizada pela capacidade do Samba 4 suprir as funções do *Active Directory*, não necessitando de licenças para usuários na organização ou empresa e na compra de software.

Destaca-se como limitação deste trabalho, dificuldades de encontrar estudos que aplicam a ferramenta Samba 4, pois, os materiais bibliográficos disponibilizados, estão apresentados em formatos que não indicam estudos científicos ou tecnológicos, e de fonte não padronizadas.

Como trabalhos futuros, sugere-se que seja colocado em prática as funcionalidades que não foram implementadas neste artigo, entre elas os serviços de certificado e federação. Outra sugestão, seria que além da utilização de Samba 4, implementado em servidor Linux, fosse realizada a utilização dos clientes com sistema operacional Linux, assim os custos com implementação de software, serviços de servidores e clientes seria zero, pois toda a rede funcionaria com softwares *open source*.

Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27002:2013**: Tecnologia da informação — Técnicas de segurança — Código de prática para controles de segurança da informação. São Paulo, 2013. 99 p.

BURGARDT, Flávio. **Infraestrutura com samba 4**. São Paulo: UNASP, 2010.

CASTRO, Tiago de Assis Oliveira. Estudo de caso: Migração de estações para software livre em domínio microsoft. Brasília - DF, 2007.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

CRUZ, Fernando William.; Santos, Giovanni Almeida Santos.; Medeiros, Raissa Dantas Freire de.; Pereira, Lucas Araújo.; Braga, Marcio Carlos.; Diener, Roberto. **Uma Ferramenta para a Administração de Serviços de Diretório Distribuídos Baseados no OpenLDAP**. Porto Alegre, 2004.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa.** 5. ed. São Paulo: Atlas, 2010.

GROSS, Marcelo Luis. **Active Directory sobre Samba**. Passo Fundo, 2015.

LDB, Samba. **Ldb**. 2016. Disponível em: <https://ldb.samba.org/>. Acesso em: 14 Out. 2016.

MICROSOFT. **O que é Active Directory, topologia física e lógica**. 2012. Disponível em: <https://technet.microsoft.com/pt-br/library/jj206711.aspx>. Acesso em: 24 set. 2016.

MICROSOFT, Corporation. **Antipirataria - Guia do CAL**. 2016. Disponível em:<https://www.microsoft.com/brasil/antipirataria/guia_cal.mspx>. Acesso em: 26 set. 2016.

MICROSOFT, Corporation . **Ferramentas de Administração de Servidor Remoto para Windows 10.** 2016. Disponível em: <https://www.microsoft.com/pt-BR/download/details.aspx?id=45520>. Acesso em: 07 out. 2016.

MORIMOTO, Carlos Eduardo. **Servidores Linux:** guia prático. Porto Alegre: Sul Editores, 2013.

MOTA FILHO, João Eriberto. **Descobrindo o LInux:** Entenda o sistema operacional GNU/Linux. 3. ed. São Paulo: Novatec, 2012.

NASCIMENTO, Ulisses Poveda. **Por dentro do Active Directory | Tecnologia da Informação - Microsoft.** 2010. Disponivel em: <https://unascimento.wordpress.com/2010/09/24/por-dentro-do-active-directory/>. Acesso em: 24 Out. 2016.

PEREIRA, Alessandro José Brasil.; LIRA, Claudiano José de; SILVA, Eric Malone Costa Silva. **Análise comparativa entre o Active Directory e o Samba 4.** Recife, 2013.

RAUEN, Fábio José. **Roteiros de iniciação científica:** os primeiros passos de pesquisa científica desde a concepção até a produção e a apresentação. Palhoça : Ed. Unisul, 2015.

SAMBA-ORG. **Samba – opening windows to a wider world**. 2016. Disponível em: <https://www.samba.org/>. Acesso em: 10 Set. 2016.

SEVERINO, Antônio Joaquim. **Metodologia do trabalho científico**. 23. ed. revisada e atualizada. São Paulo: Cortez, 2007.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.

STANEK, Willian Robert. **Windows Server 2008:** guia completo. Porto Alegre: Bookman, 2008.

VIGNATTI, Anderson. **Implantação e Estudo de um Sistema Utilizando o Active Directory.** Lavra - MG, 2007. Disponivel em: <http://repositorio.ufla.br/bitstream/1/5399/1/MONOGRAFIA_Implanta%C3%A7%C3%A3o_e_estudo_de_um_sistema_utilizando_o_active_directory.pdf>. Acesso em: 10 Out. 2016.

In: FREITAS JUNIOR, V.; COSTA, G. C.; FERNANDES, V. S. (Org.). Tecnologia e Redes de Computadores. 3. Ed. Sombrio: Instituto Federal Catarinense, 2017. p. XX – XX. ISBN: 978-85-5644-013-6.