UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

HELIELTON DOS SANTOS MAINARDES

ANÁLISE E SIMULAÇÃO DE FIREWALL

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2016

HELIELTON DOS SANTOS MAINARDES

ANÁLISE E SIMULAÇÃO DE FIREWALL

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Augusto Foronda

PONTA GROSSA

2016

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa

Diretoria de Pesquisa e Pós-Graduação
Departamento Acadêmico de Informática
Tecnologia em Análise e Desenvolvimento de Sistemas

TERMO DE APROVAÇÃO

ANÁLISE E SIMULAÇÃO DE FIEWALL

por

HELIELTON DOS SANTOS MAINARDES

Este Trabalho de Conclusão de Curso foi apresentado em 23 de novembro de 2016 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Augusto Foronda
Prof. Orientador

_____
Luiz Rafael Schmitke
Membro titular

_____
Alessandro Luiz Stamatto Ferreira
Membro titular

_____
Profª. Mônica Hoeldtke Pietruchinski
Responsável pelo Trabalho de Conclusão de Curso

_____
Profª. Dra. Mauren Louise Sguario
Coordenadora do curso

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

Dedico este trabalho à minha noiva, pelos momentos de ausência e pela compreensão e apoio. Também dedico aos meus Pais e irmãos que sempre me incentivaram a valorizar meus estudos.

AGRADECIMENTOS

Acredito que estes parágrafos não são suficientes para agradecer a todas as pessoas que contribuíram de alguma forma com esta importante fase da minha vida, mas que elas tenham a certeza de minha imensa gratidão.

Primeiramente, agradeço a Deus pelo discernimento para buscar o melhor para minha vida e concluir mais uma importante fase.

Agradeço ao meu orientador Prof. Dr. Augusto Foronda, pela sabedoria, simpatia e simplicidade com que me guiou nesta trajetória.

Aos amigos que conquistei durante este período e aos amigos que sempre me incentivaram e deram forças para continuar.

Gostaria de deixar registrado também, o meu reconhecimento à minha noiva e minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio.

Enfim, a todos os que por algum motivo contribuíram para a realização desta pesquisa.

RESUMO

SANTOS MAINARDES, Helielton. **Análise e Simulação de Firewall**. 2016. 50 f. Trabalho de Conclusão de Curso – Tecnologia em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2016.

Este trabalho tem como objetivo demonstrar a importância da segurança de rede utilizando um serviço de *Firewall*. Será feito a revisão teórica dos tópicos importantes para a compreensão do assunto através da literatura disponível, como livros, manuais e guias. A simulação do *Firewall* será executada na ferramenta *Packet Tracer* 6.2 da CISCO, com o intuito de demonstrar como o *Firewall* trabalha utilizando ACLs e CBAC para gerenciar o tráfego de pacotes. Espera-se com este trabalho mostrar a necessidade de se manter um sistema de segurança atualizado, principalmente em nível corporativo, tendo em vista a hostilidade que a Internet possui, além disso, apresentar a eficiência desta ferramenta para evitar ataques, controlar e tornar seguro o tráfego da rede.

Palavras-chave: *Firewall*. Segurança. Gerenciamento. Restrição. Filtragem.

ABSTRACT

SANTOS MAINARDES, Helielton. **Analysis and Simulation Firewall**. 2016. 50 f. Work of Conclusion Course – Graduation in Analysis and Systems Development - Federal Technology University - Paraná. Ponta Grossa, 2016.

This work aims to demonstrate the importance of network security using a firewall service. It will be the theoretical review of topics important to the understanding of the subject through the available literature, such as books, manuals and guides. The simulation Firewall will run in the Packet Tracer 6.2 CISCO tool, in order to demonstrate how the firewall works using ACLs and CBAC to manage packet traffic. It is hoped that this work show the need to maintain an updated security system, mainly at the corporate level, in view of the hostility that Internet has, in addition, present the effectiveness of this tool to prevent attacks, to control and make safe the network traffic.

Keywords: Firewall. Safety. Management. Restriction. Filtration.

LISTA DE FIGURAS

LISTA DE SIGLAS

TCP	*Transmission Control Protocol*
IP	*Internet Protocol*
ACL	*Access Control List*
DoD	*Department of Defense*
HTTP	*Hypertext Transfer Protocol*
SMTP	*Simple Mail Transfer Protocol*
FTP	*File Transfer Protocol*
DMZ	*Demilitarized Zone*
VPN	*Virtual Private Network*
IPv4	*Internet Protocol version 4*
IPv6	*Internet Protocol version 6*
UDP	*User Datagram Protocol*
PC	*Personal Computer*
ICMP	*Internet Control Message Protocol*
DNS	*Domain Name System*

LISTA DE ACRÔNIMOS

CBAC	*Context-Based Access Control*
LAN	*Local Area Network*
OSI	*Open System Interconnect*
BIT	*Binary digit*
ARPANET	*Advanced Research Projects Agency Network*
DARPA	*Defense Advanced Research Projects Agency*
EUA	Estados Unidos da América
NAT	*Network Address Translation*
IPSEC	*IP Security Protocol*
MAC	*Media Access Control*
TELNET	*Telecommunications Network*
BYTE	*Binary Term*
PING	*Packet Internet Grouper*

SUMÁRIO

1 INTRODUÇÃO

A utilização em massa da Internet para transações bancárias, compras, pagamentos, entre outras operações que demandam sigilo das informações, é algo comum nos dias atuais. Infelizmente, tão comum quanto a utilização da Internet para facilitar as tarefas diárias é a utilização deste meio para interceptar estas tarefas e causar danos ao usuário, roubando seus dados ou destruindo suas informações. Problemas como este existem desde a invenção da Internet e evoluem tão rápido quanto ela, gerando a grande preocupação e o grande investimento no desenvolvimento de sistemas de segurança mais avançados que as ameaças (NAKAMURA; GEUS, 2007).

Apesar de existirem vários recursos disponíveis para auxiliar na proteção a ataques, como antivírus, *antispywares*, mecanismos de encriptação, *Firewalls*, protocolos de comunicação, entre outros, diariamente vários ataques e invasões ocorrem com sucesso, transpassando tais barreiras.

O sistema de segurança a ser estudado neste trabalho é o *Firewall*, que foi criado inicialmente apenas com o intuito de restringir o acesso entre redes existentes e atualmente, é um componente comum em todas as redes, desde residências até as maiores corporações do mundo.

Basicamente o *Firewall* é como uma barreira entre a rede interna e o resto do mundo, ele age controlando as entradas e as saídas da rede, impedindo que o vírus se espalhe para o restante da rede e permitindo que o tráfego seja administrado. Para KUROSE (2006), é denominado *Firewall* a utilização em conjunto de hardware e software para isolar e filtrar pacotes que trafegam entre a rede interna e o resto da Internet.

1.1 OBJETIVOS

Os objetivos gerais e específicos a serem atingidos por esse trabalho são descritos abaixo.

1.1.1 Objetivo Geral

Este projeto tem como objetivo principal a análise e simulação de um *Firewall* dentro do roteador na entrada da rede antes da implementação em um ambiente real.

1.1.2 Objetivos Específicos

- Explanação do tema e intensificação da importância das ferramentas de seguranças no ambiente seja ele corporativo ou residencial;
- Análise da teoria de *Firewall*, revisão e descrição dos conceitos sobre o modelo TCP/IP (*Transmission Control Protocol/Internet Protocol*), ACL (*Access Control List*), CBAC (*Context-Based Access Control*);
- Simulação de diversos ambientes de rede para a aplicação e testes da ferramenta, dos modelos e politicas, visando demonstrar a melhor aplicação dentro de contextos diferentes.

A simulação de diversos ambientes na ferramenta *Packet Tracer*, facilitam a visualização e a compreensão da complexidade que pode nos ocorrer quando tratamos de segurança. A utilização de ferramentas de simulação auxilia na criação de um sistema de defesa personalizado para um cliente, já que as partes lógicas (Internet, LANs, políticas) também são visualizadas na criação do ambiente, tornando o sistema de defesa compreensível e menos suscetível a erros quando implantado no ambiente real.

1.2 JUSTIFICATIVA

Todos os dias milhares de ataques virtuais acontecem no mundo e, mesmo conhecendo o assunto e com tantas formas e dicas de segurança, muitas pessoas acabam tendo seus dados roubados, suas contas invadidas, suas informações perdidas. Para piorar a situação, vários destes casos acontecem no âmbito corporativo, onde as informações perdidas ou os dados roubados são ainda de maior valor. Na maioria das vezes, estas invasões ocorrem devido ao fator humano, um usuário mal instruído, com pouco conhecimento ou sobrecarregado, que acaba

clicando ou executando algo que não devia. Entretanto, estamos suscetíveis a invasões mesmo sem a intervenção do usuário, que ocorrem através de ataques robotizados que atingem diretamente nossa rede até conseguirem quebrar nossa barreira.

Como em qualquer contexto, o problema só é descoberto após a invasão ou após a perda de dados importantes, geralmente tarde demais para impedir ou recuperar as informações. E, apenas neste momento as empresas percebem o quão importante é investir em Tecnologias de Segurança.

“A segurança pode ser burlada por vírus e outras pestes digitais, destruindo dados valiosos e demandando muito tempo dos administradores para eliminar todos os problemas causados” (TANENBAUM, 2003).

1.3 METODOLOGIA

Este projeto será desenvolvido com embasamento teórico de manuais, normas, documentos da Internet e bibliografia referente aos tópicos que serão abordados.

Primeiramente será trabalhado com a revisão teórica e com a contextualização do tema principal e das principais apêndices e tópicos necessários para melhor compreensão do assunto.

Será realizada a revisão teórica sobre o modelo TCP/IP, além da revisão teórica do *Firewall* e sobre as demais políticas, como a utilização de ACL, CBAC. A revisão teórica será realizada a partir de bibliografias na área de Rede de Computadores e de sites que abordam tais temas.

Após o término da revisão e contextualização dos assuntos, será iniciada a etapa de testes e análise, que consiste na simulação de ambientes de rede dentro do *Packet Tracer* 6.2. Será criado o ambiente de rede para simular a utilização de ACLs, CBAC, com o intuito de analisar e trabalhar com os atributos e com as politicas revisadas.

A utilização de testes e criação da rede em um ambiente simulado permite que as configurações, o ambiente, as diversas portas de entrada para ataques e as diversas formas de combate a estes ataques, sejam mais bem visualizados e

compreendidos, facilitando assim a demonstração da grande importância de se manter um ambiente com as ferramentas de segurança atualizadas e monitoradas frequentemente.

Ao término da revisão teórica, análise e testes das ferramentas, será possível compreender a grande importância da implantação e do monitoramento de um *Firewall*, principalmente em ambientes corporativos.

2 EMBASAMENTO TEÓRICO

Neste capítulo, será revisada a teoria das tecnologias relacionadas ao *Firewall*, que são base para o entendimento da análise e simulação proposta.

2.1 MODELO TCP/IP

Pessoas em diferentes lugares do mundo se comunicam com muita frequência e estas, utilizam redes com diferentes tecnologias que precisam ser conectadas para permitir a comunicação. Para realizar esta conexão de modo transparente é necessário que uma máquina, conhecida como *gateway,* execute a conversação e tradução necessária para que uma rede compreenda a outra. A conexão de diferentes redes ao redor do mundo é o que conhecemos por Internet (TANENBAUM, 2003).

As redes trabalham utilizando diversas camadas que são responsáveis por serviços específicos para executar a comunicação e o trafego de dados. Cada camada desempenha um serviço que é requisito para que a próxima camada possa trabalhar. Nos projetos o *software* passou a ter muito valor e muito investimento, diferente do que acontecia antes, onde o *hardware* era considerado o item mais importante do ambiente (TANENBAUM, 2003).

O modelo TCP/IP possui diversos protocolos que suas camadas utilizam, dentre eles temos o TCP (*Transmission Control Protocol* – Protocolo de Controle de Transmissão*) e o IP (Internet Protocol* – Protocolo da Internet), que são considerados os mais importantes da Internet (KUROSE, 2006).

Um protocolo pode ser definido como um conjunto de regras que gerenciam a comunicação com o objetivo de detectar e evitar a perda de dados ao longo da transmissão fazendo com que ela seja eficiente e sem erros, além de garantir que um dado chegue a outro ponto da mesma forma que foi transmitido (SOUSA, 1999).

A partir dos conceitos apresentados sobre as redes divididas em camadas, será feito uma revisão teórica das arquiteturas de rede e do modelo de referência OSI (*Open Systems Interconnection*), mas o foco estará no modelo de referência TCP/IP.

“O modelo OSI foi desenvolvido como o primeiro passo para a padronização da comunicação de dados e compatibilizar os diferentes protocolos” (FILIPPETTI, 2008).

Este modelo é composto por sete níveis de protocolos (camadas) que serão descritas segundo Soares (1995):

- Camada física: transmissão de *bits* brutos na rede;
- Camada de enlace de dados: transforma um canal de transmissão não confiável em um canal confiável para o uso no nível de rede;
- Camada de rede: responsável por estabelecer e operar a conexão de rede;
- Camada de transporte: fornece a comunicação fim a fim para garantir a entrega de um pacote, controle de fluxo, detecção e recuperação de erros fim a fim e segmentação e blocagem de mensagens;
- Camada de sessão: permite sessões entre usuários de diferentes máquinas, gerenciamento *token,* controle de diálogo e gerenciamento de atividades;
- Camada de apresentação: realiza transformações nos dados antes de seu envio ao nível de sessão, formatação de dados, seleção de sintaxes e estabelecimento de conexões de apresentação.
- Camada de aplicação: fornece aos processos de aplicação os meios para que utilizem o ambiente de comunicação.

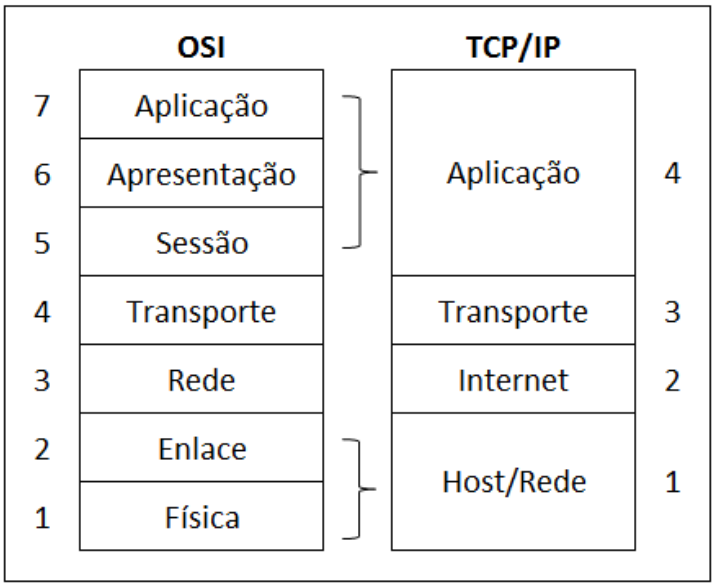
O Departamento de Defesa dos Estados Unidos (DoD) mantinha uma rede de pesquisas conhecida como ARPANET, eles foram os precursores da Internet e utilizavam a rede para fins militares. O crescimento deste departamento e o avanço de suas pesquisas foram os principais fatores que motivaram a criação do modelo de referencia TCP/IP (FILLIPPETTI, 2008).

As redes militares de pesquisa ficaram conhecidas como DARPANET, cuja estrutura ainda possuía diversos problemas. Com o intuito de melhorar este modelo, e criar assim uma rede redundante e resistente a ataques ou até mesmo a panes que poderiam acontecer em algum equipamento ao decorrer da rede, o DoD desafiou diferentes centros de pesquisas e universidades a desenvolver um novo modelo, melhor e mais seguro. O principal objetivo era manter as conexões entre a origem e o destino em funcionamento, não importando o que acontecesse no

caminho a ser percorrido pela informação. Independente de quantas sub-redes ou máquinas parassem de funcionar repentinamente, a conexão e comunicação entre as principais redes deveriam ser mantidas (FILIPPETTI, 2008). Junto com a necessidade de criar um modelo seguro e redundante surgiu a necessidade de um modelo que aceitasse novas tecnologias, como transferência de arquivos e transmissão de voz, e que pudesse se adaptar a elas independente de seus requisitos (TANENBAUM, 2003).

Um conjunto de diferentes protocolos forma uma camada e a junção destas camadas forma o modelo TCP/IP. Entre as diversas funções dos protocolos que as camadas utilizam, estão as tarefas de conectar, efetuar a comunicação e permitir transferência de arquivos entre dispositivos de redes locais e também de redes locais com redes externas que utilizam diferentes tecnologias (SOUSA, 1999).

Figura 1 - Comparação Modelo OSI e TCP/IP



Fonte: Tanenbaum, 2003

O modelo OSI iniciava com as camadas Física e de Enlace, as quais foram englobadas pela camada *Host*/Rede no modelo TCP/IP. Esta camada tem como principais funções o monitoramento dos dados que trafegam na rede também é responsável pela comunicação com o meio físico da rede (FILIPPETTI, 2008).

A camada Internet cuida de todo o trafego entre a origem e o destino e também dos pacotes recebidos das interfaces. A rede de transporte envia

solicitações à camada de Internet para que ela avalie o pacote e defina a forma com que ele deverá ser processado. (SOARES, 1995).

Apesar de a camada Internet cuidar do tráfego de dados entre a origem e o destino, ela não cuida da integridade dos dados, ou seja, ela não verifica se os dados estão sendo entregues na mesma sequencia em que foram enviados e nem se eles possuem erros. Essa tarefa é executada pela camada de transporte, que assegura que os pacotes não possuem erros e os reorganiza para entregar na ordem correta. Esta camada faz com que a rede seja confiável no quesito integridade de dados (LEIDEN; WILENSKY, 2009).

A última camada do TCP/IP é a camada de aplicação onde estão as aplicações de rede e seus protocolos. Entre os protocolos existentes nesta camada, estão alguns mais conhecidos como o HTTP (requisição e transferência de documentos pela *Web*), o SMTP (transferência de mensagem de correio eletrônico) e o FTP (responsável por transferência de arquivos). O protocolo de camada de aplicação é utilizado por diversos sistemas finais para troca de pacotes (KUROSE 2006).

O modelo TCP/IP trata-se de uma evolução do modelo OSI, desta forma, desempenha a mesma função que o OSI, mesmo tendo um número menor de camadas ambos utilizam diferentes protocolos. O modelo OSI era muito flexível e genérico, a grande dificuldade com ele era a necessidade de se adaptar a cada nova tecnologia criada, isto demandava muito tempo de estudo para a criação de funcionalidades e tecnologias ainda desconhecidas. Com o avanço das tecnologias de redes, o modelo OSI passou a ter ainda mais dificuldades para ser compatível com as necessidades e tecnologias. Toda esta evolução foi uma grande vantagem para o modelo TCP/IP já que ele foi criado depois do aperfeiçoamento das tecnologias e dos protocolos, o que lhe deu uma maior compatibilidade. (TANENBAUM, 2003).

2.2 *FIREWALL*

A capacidade de interligar o mundo inteiro através de um computador é algo fantástico. Poder fazer compras sem sair de casa, poder fazer movimentações bancárias, ter acesso a outros computadores, trabalhar em casa, entre as diversas

opções que a Internet oferece. As pessoas em todo o mundo estão tão acostumadas em fazer todas estas tarefas utilizando a Internet, que muitas vezes os cliques se tornam automáticos, o que é algo muito perigoso em um ambiente tão hostil.

No passado, apenas pessoas com um grande conhecimento na área ou estudantes da computação conseguiam originar uma ameaça à rede, porém com o passar do tempo, tais ameaças se multiplicaram e praticamente qualquer pessoa que tenha acesso a Internet pode inicia-la (FILIPPETTI, 2008).

A Internet é grande, internacional e não controlada. Estes atributos a tornam um recurso valioso e crescente, mas também definem o risco para organizações que conectam suas redes a ela (ZACKER; DOYLE, 2000). Tendo em vista a hostilidade do ambiente que é utilizada diariamente, se percebe a grande importância e dificuldade em obter segurança de rede e segurança de informação.

“Nada evitará que tentativas de invasões continuem a existir, mas o que definirá se estas serão bem-sucedidas ou não será o conhecimento embutido em seu *Firewall* e demais ferramentas de segurança” (NETO, 2004). Para Soares (1995) esta necessidade de proteção de informações e dados confidenciais e a utilização de um elemento da rede por meios não autorizados está relacionado diretamente à segurança.

Fornecer segurança de rede e segurança de informação requer proteção de todos os recursos da rede, sejam eles físicos ou abstratos. Embora a segurança física não seja tão mencionada, ela desempenha um papel muito importante na infraestrutura da rede, pois se tratam de cabeamentos, pontes e roteadores que interligam toda a rede, além de equipamentos para *backup*. Todavia, é mais difícil proteger um recurso abstrato do que um recurso físico, como as informações podem ser transferidas e modificadas muito rapidamente, torna-se difícil verificar a veracidade da informação, além disso, manter a integridade e disponibilidade dos dados é uma preocupação da segurança abstrata, já que a informação é algo abstrato (COMER, 1998).

A ameaça no ambiente de rede consiste em uma possível violação da segurança, as principais ameaças são (SOARES, 1995):

- Destruição de informação ou de outros recursos;
- Modificação ou deturpação da informação;
- Roubo, remoção ou perda de informação ou de outros recursos;
- Revelação de informação;

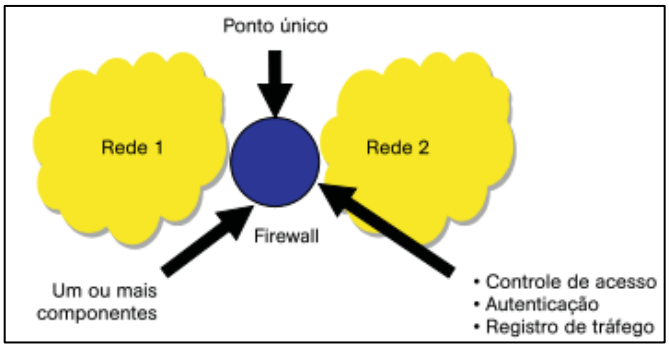
- Interrupção de serviços.

Analisando tais informações sobre as ameaças, se percebe que qualquer uma das principais ameaças, pode trazer consigo prejuízos incalculáveis, principalmente em um ambiente corporativo. A destruição ou roubo de uma informação ou recurso do ambiente pode gerar perdas maiores ainda, sendo que a informação ou recurso atingido muitas vezes é parte de processos vitais para a empresa. A revelação de informações é muito temida no ambiente empresarial, pois essas informações podem estar relacionadas a segredos comerciais, estratégias de mercado, análises financeiras, entre outras, que ao cair na mão de um concorrente, por exemplo, acarretaria amargas consequências. A interrupção de serviços poderia causar a parada de uma linha de produção inteira, poderia parar todo o processo de distribuição de mercadorias, talvez por apenas 30 minutos ou até mesmo por 30 horas, consegue calcular o prejuízo que causaria em uma indústria de grande porte?

Levando em consideração todos os riscos que a rede fornece, desde a utilização pessoal até a utilização por grandes corporações, é possível ter noção da importância de um sistema de segurança que controle todo o tráfego que entra e sai de uma rede. Em um ambiente empresarial, este processo é feito em uma central de segurança através de mecanismos operacionais conhecidos como *Firewalls* e outros sistemas de detecção e prevenção de invasão.

“Pode-se dizer que o *Firewall* é um ponto entre duas ou mais redes, que pode ser um componente ou um conjunto de componentes, por onde passa todo o tráfego, permitindo que o controle, a autenticação e os registros de todo o tráfego sejam realizados” (NAKAMURA; GEUS, 2007).

Figura 2 - Definição de Firewall

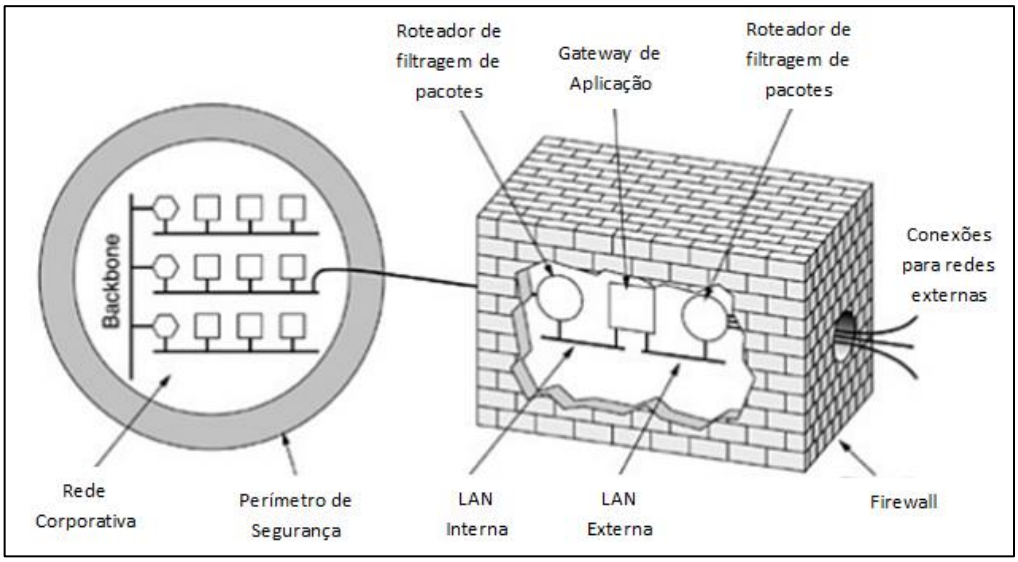


Fonte: NAKAMURA e GEUS, 2007

O *Firewall* não é apenas utilizado para proteger uma rede privada de uma rede pública não confiável, ele também pode ser aplicado dentro de um ambiente corporativo para a separação de grupos de trabalhos ou sub-redes. Como se pode observar na Figura 2, o *Firewall* pode ser composto por um ou mais componentes. Quando utilizado mais componentes, cada um assume uma função que está ligada diretamente ao nível de segurança da rede em que está sendo projetado.

Os *Firewalls* são apenas uma adaptação moderna de uma antiga forma de segurança medieval: cavar um fosso profundo em torno do castelo. Esse recurso forçava todos aqueles que quisessem entrar ou sair do castelo a passar por uma única ponte levadiça, onde poderiam ser revistados por guardas (TANENBAUM, 2003).

Figura 3 - Firewall com dois filtros de pacotes e um gateway de aplicação



Fonte: Tanenbaum, 2003

Na Figura 3, temos um projeto de *Firewall* com dois roteadores para filtragem de pacotes e o *gateway* de aplicação, a principal vantagem deste modelo é que todos os pacotes terão de passar por duas filtragens e pelo *gateway* antes de sair ou entrar na rede.

“Cada filtro de pacotes é um roteador padrão equipado com funções complementares, que permitem a inspeção de cada pacote de entrada ou de saída. Os pacotes que atenderem a algum critério serão remetidos normalmente, mas os que falharem no teste serão descartados” (TANENBAUM, 2003).

O *Firewall* do tipo filtro de pacotes analisa cabeçalhos enquanto os mesmos trafegam. A partir de regras previamente adicionadas, o *Firewall* do tipo filtro de pacotes faz diversas comparações com o pacote que esta sendo analisado para decidir se o mesmo pode trafegar livremente ou deve ser parado totalmente (NETO, 2004).

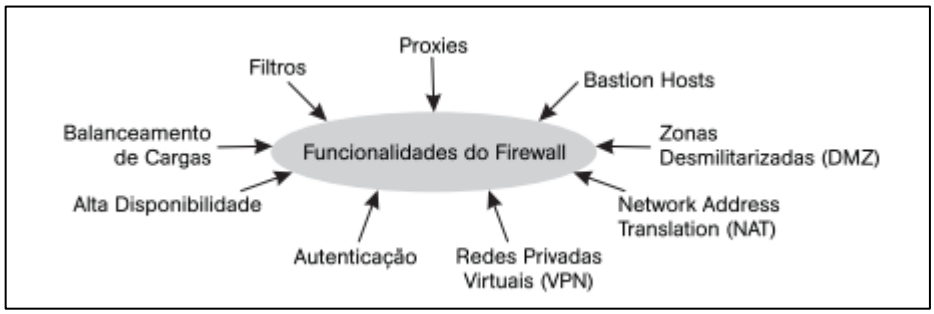
Para Kurose (2006), os *Firewalls* podem ser classificados em três categorias: filtros de pacote tradicionais, filtros de estado e *gateways* de aplicação. No filtro de pacote tradicional, cada pacote é verificado individualmente e então sua passagem é negada ou liberada, com base nas regras especificadas pelo administrador. O filtro de pacote de estado trabalha de uma forma mais inteligente que a tradicional, ele basicamente analisa o trafego de dados e verifica se os padrões utilizados estão de acordo com as regras. Por fim, um *gateway* de aplicação trata-se de um servidor de aplicação que controla todos os dados da aplicação. Desta forma é possível criar grupos de usuários que podem executar uma aplicação que está nas regras de bloqueio do filtro, porém ser deixar de lado a segurança, tendo em vista que o usuário deve irá requisitar a utilização da aplicação primeiramente ao *gateway*, que por sua vez irá verificar as informações sobre a identidade do usuário e suas permissões.

2.2.1 Funcionalidades de *Firewall*

Como visto anteriormente, cada componente acrescentado em um *Firewall* possui uma funcionalidade que está diretamente ligada ao nível de segurança da rede. Alguns componentes são considerados básicos em um *Firewall* e outros foram inseridos conforme o aumento da necessidade de segurança.

Cada componente acrescentado em um *Firewall* trabalha dedicado a um requisito de segurança da rede e, quando mantido e configurado corretamente para trabalhar em conjunto sem interferir na função dos outros componentes, aumenta consideravelmente a segurança do ambiente.

Figura 4 - Funcionalidades do Firewall



Fonte: NAKAMURA e GEUS, 2007

2.2.1.1 Filtros

Verifica os pacotes de entrada e saída com base nas tabelas definidas pelo administrador, permitindo ou negando o trafego do pacote (TANENBAUM, 2003).

2.2.1.2 *Proxies*

O *proxy* é um servidor que intermedia a conexão do *host* interno com um *host* externo. Na sua forma mais utilizada, o usuário se autentica e envia sua requisição para o *proxy*, que verifica se o acesso ao host é permitido e comunica-se com o servidor, verificando a resposta do servidor e permitindo ou bloqueando o trafego. Além disso, o *proxy* permite a criação de *logs* por atividade, usuário e tráfego (NAKAMURA; GEUS, 2007).

2.2.1.3 *Bastion hosts*

Trata-se de equipamentos que ficam além do *Firewall* oferecendo serviços para a Internet. Devido ao contato direto com as redes externas, estão mais suscetíveis a ataques, por isto precisam ser muito bem protegidos e devem possuir apenas os serviços e aplicações essenciais. Os *bastion hosts* não tem contato direto com a rede interna, a interação acontece apenas com a zona desmilitarizada (DMZ) (NAKAMURA; GEUS, 2007).

2.2.1.4 Zona desmilitarizada (DMZ)

A DMZ é uma rede que fica entre a rede interna e a rede externa. Ela cria uma camada adicional de segurança, mantendo os serviços que necessitam de acesso externo (FTP, EMAIL, HTTP) separados da rede interna. Desta forma, se um *bastion host* é comprometido, a rede interna continua intacta e segura (NAKAMURA; GEUS, 2007).

2.2.1.5 *Network address translation (NAT)*

A ideia da NAT, além de traduzir os endereços de rede na comunicação entre a rede interna e externa, é reservar um IP a cada empresa para trafego na Internet. Na rede interna, cada computador tem um IP especifico que serve apenas para trafego interno. Toda vez que existe uma comunicação com o ambiente externo, é necessário que a ocorra à conversão de endereço para que o pacote consiga trafegar na rede externa e o mesmo processo acontece quando este pacote retorna da rede externa para a interna (TANENBAUM, 2003).

2.2.1.6 Rede privada virtual (VPN)

A VPN é utilizada para a comunicação entre diferentes redes de forma segura. Ela cria um túnel entre as redes utilizando criptografia e *IP Security* (IPSec) para manter sigilo, integridade e autenticação dos dados (KUROSE, 2006).

2.2.1.7 Autenticação/certificação

A autenticação do usuário é de grande importância para a segurança do ambiente corporativo. Existem várias formas de validar este acesso, tendo como base o endereço IP, senhas, certificados digitais, *tokens*, biometria ou chaves publicas (NAKAMURA; GEUS, 2007).

2.2.1.8 Balanceamento de cargas e alta disponibilidade

Devido ao *Firewall* ser o único meio de acesso a uma rede e à quantidade excessiva de trafego que passa por ele, torna-se necessária alguma forma de balancear todo este trabalho. O balanceamento de cargas visa o trabalho em paralelo de dois ou mais *Firewalls* para dividir todo o trafego.

A alta disponibilidade visa a disponibilidade em tempo integral do *Firewall*, e isto pode ser mantido através um espelho ou um *backup* que passe a funcionar no local do original, em caso de indisponibilidade ou paradas não esperadas (NAKAMURA; GEUS, 2007).

2.3 LISTA DE CONTROLE DE ACESSO (ACL)

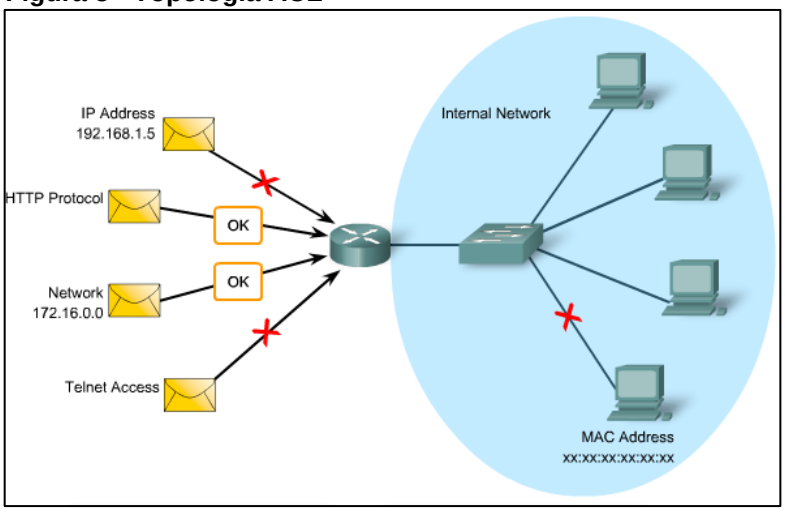
Inicialmente, as ACLs eram os únicos meios utilizados para fornecer proteção de *Firewall*. Outras tecnologias começaram a surgir no final da década de 90, até este momento, o aumento dos tipos de ACLs foi significativo.

Como visto anteriormente, hoje em dia existem vários tipos de *Firewall* que são, na maioria das vezes, utilizados em conjunto em uma mesma rede. As ACLs ainda são muito utilizadas atualmente para controle de trafego e para diminuir a chance de ataques à rede. Através delas, classes são definidas e controladas com base em diversos parâmetros de rede. Praticamente qualquer tipo de trafego pode ser controlado através de ACLs, tendo como base parâmetros de rede que envolvem endereços MAC, IPv4 e IPv6, além da numeração das portas TCP e UDP, como visto na Figura 5.

Resumidamente, as ACLs podem permitir ou negar o tráfego para certo endereço ou tipo de trafego e podem restringir a utilização da rede para um serviço e/ou dispositivo. Elas podem ser configuradas de dois modos: no modo *whitelist* onde todos os pacotes são permitidos, exceto os que estão descritos na ACL ou no modo *blacklist* que, diferente do *whitelist,* bloqueia todos os pacotes, permitindo apenas os que estão descritos na ACL. Os principais tipos de ACLs existentes são ACL Padrão e ACL Estendida, a partir destas surgiram outras variações. Ambas são utilizadas para descrever os pacotes que entram e saem de uma interface (FILIPPETTI, 2008).

As ACLs podem ser identificadas utilizando números ou nomes. Nas ACLs numeradas o número é atribuído com base no protocolo que será filtrado, de 1 a 99 e 1300 a 1999 para ACL IP Padrão e de 100 a 199 e 2000 a 2699 para ACL IP Estendida. As listas de acesso tem grande importância para o controle, entretanto, em grandes redes as listas numeradas não são a melhor opção, levando em consideração o gerenciamento das listas existentes e o fato de que elas não podem ser editadas. Para resolver tais problemas, foram criadas as listas de acesso nomeadas, facilitando o gerenciamento através de nomes intuitivos definidos pelo administrador e facilitando as modificações, permitindo a exclusão ou inserção de uma nova linha. (FILIPPETTI, 2008)

Figura 5 - Topologia ACL



Fonte: Filippetti, 2008

2.3.1 ACL Padrão

Este modelo de ACL é utilizado para filtrar pacotes baseados na origem e assim, permitir ou negar o tráfego de protocolos com base no endereço. Devem estar mais próximos do destino.

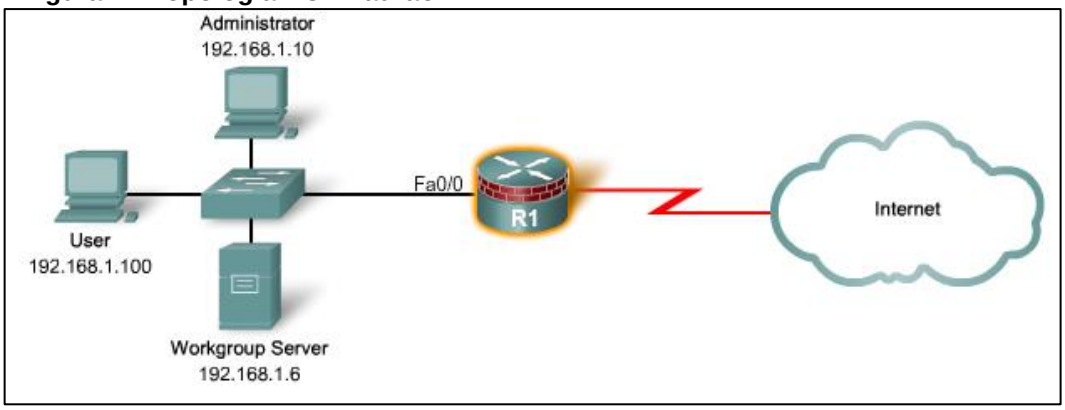
Figura 6 – ACL Padrão

```
access-list {1-99} {permit | deny} source-addr [source-wildcard]
```

Fonte: Filippetti, 2008

Na figura 6 está a sintaxe para configurar uma ACL padrão numerada. O primeiro valor define a ACL, de 1 a 99 para ACL padrão. O segundo valor especifica se o trafego do endereço IP de origem deve ser permitido ou negado. O terceiro valor é o endereço IP de origem que deve ser correspondido e o quarto valor é a máscara padrão a ser aplicada ao IP configurado anteriormente para indicar o intervalo.

Figura 7 - Topologia ACL Padrão



Fonte: Filippetti, 2008

Na Figura 7 o roteador, que está entre a Internet e a rede interna, possui uma ACL padrão configurada na interface Fa0/0, nela pode ser definido se um ou mais *hosts* terão seu acesso concedido ou negado a outra rede ou aos demais *hosts*.

2.3.2 ACL Estendida

Utilizado para filtrar pacotes baseados na origem e no destino através de protocolo (IP, TCP, UDP, etc.) e número de porta. Devem estar mais próximos da origem para evitar o uso desnecessário da rede.

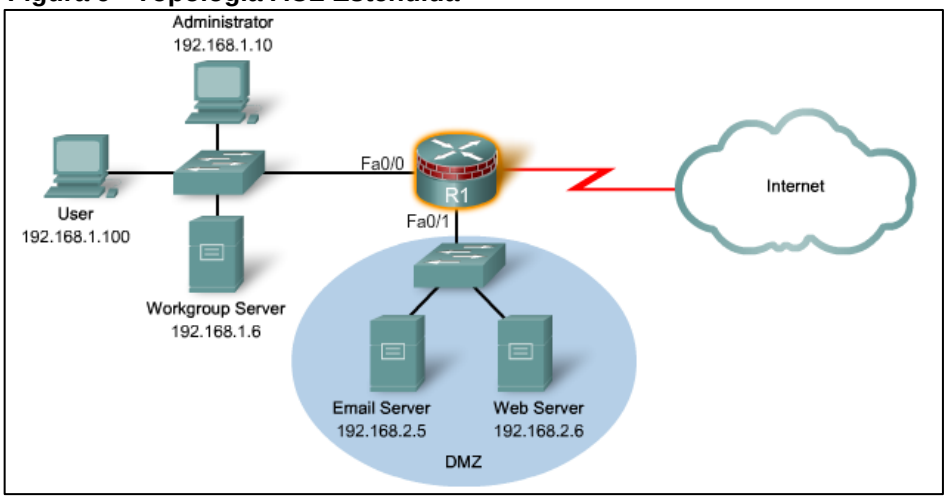
Figura 8 – ACL Estendida

```
access-list {100-199} {permit | deny} protocol source-addr
[source-wildcard][operator operand] destination-addr [destination-
wildcard] [operator operand][established]
```

Fonte: Filippetti, 2008

Da mesma forma que a ACL padrão, o primeiro valor é o identificador da ACL, os valores de 100 a 199 e de 2000 a 2699 indicam que se trata de uma ACL estendida, o segundo valor especifica se deve permitir ou negar. O terceiro valor indica o tipo do protocolo, o administrador deve especificar IP, protocolo TCP, UDP e outros sub-protocolos. O endereço IP de origem e a mascara padrão determinam onde o tráfego se origina. O endereço IP de destino e sua mascara padrão definem o destino final do tráfego. Quando o endereço IP de destino e a mascara são especificados, o administrador deve especificar o número da porta para coincidir com uma porta com número ou nome conhecido, caso contrário, o tráfego para esse destino será descartado.

Figura 9 - Topologia ACL Estendida



Fonte: Filippetti, 2008

Na topologia mostrada na Figura 9 o roteador possui duas interfaces. Na interface Fa0/0 pode ser controlado quais *hosts* podem utilizar os protocolos FTP e HTTP por exemplo. E na interface Fa0/1 pode ser definido quais portas e protocolos estão liberados para cada *host.*

As ACLs são criadas a nível global dentro da rede e então aplicadas a cada interface, elas são capazes de filtrar os pacotes que passam por um roteador e deve ser definido apenas uma ACL por interface, protocolo e direção. É importante lembrar que as ACLs são processadas de cima para baixo, portanto, quando um pacote atende um teste da ACL, seu processamento é parado e as regras sequentes não são testadas neste pacote. Este fator mostra a importância de que as ACLs devem ser muito bem revisadas antes de serem aplicadas à rede para evitar

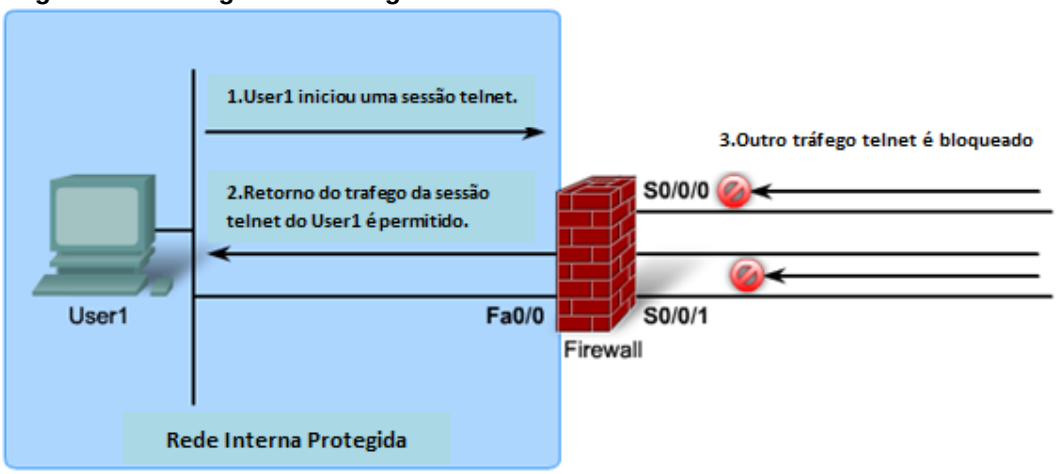
retrabalhos, principalmente quando se utiliza os modos de *whitelist* e *blacklist* (*deny all)*.

Em um ambiente de rede moderno, o *Firewall* é colocado entre a rede interna e a externa, com o intuito de impedir a entrada do tráfego externo. Esta função só é excedida se este tráfego for explicitamente permitido por uma ACL ou se for o retorno de um tráfego iniciado na rede interna. Além das ACLs *padrão* e *estendida*, existem outros tipos que, como dito anteriormente, são variações destas, como *Reflexive IP ACLs, Dynamic ACLs, Time-Base ACLs* e a *Context-based Access Control (CBAC) ACLs* que será revisada na próxima seção.

2.4 *CBAC (CONTEXT-BASED ACCESS CONTROL)*

O CBAC fornece funcionalidades avançadas de filtragem de tráfego com base na camada de aplicação e inspeciona as atividades por trás do *Firewall*. Como visto anteriormente, o CBAC é também uma ACL, a principal diferença é que as demais ACLs analisam apenas a camada IP e de Transporte, enquanto o CBAC analisa a camada de Aplicação. Ele possui quatro funções principais: filtragem de tráfego, inspeção de tráfego, detecção de intrusão e geração de alertas e auditorias.

Figura 10 - Filtragem de Tráfego com CBAC



Fonte: SANTOS; STUPPI, 2015

Na Figura 10 é mostrada a filtragem utilizando CBAC. Ao iniciar uma sessão *telnet,* o User1 está na camada de aplicação onde os protocolos de nível mais alto atuam, quando o retorno do tráfego da sessão *telnet* retorna, o CBAC analisa e

permite sua passagem. Ao receber outro tráfego *telnet,* o CBAC verifica que o mesmo não foi requisitado ou que a sessão não se iniciou na rede interna e interrompe este tráfego.

2.4.1 Filtragem de Tráfego

A filtragem de tráfego feita pelo CBAC examina não apenas a camada de rede e as informações da camada de transporte, mas também examina as informações do protocolo na camada de aplicação. O CBAC pode monitorar os protocolos e conexões manter as informações em uma tabela para acompanhar as sessões ativas.

2.4.2 Inspeção de Tráfego

Como o CBAC inspeciona os pacotes na camada de aplicação e mantem informações sobre as sessões, ele consegue detectar e prevenir muitos ataques de rede, como ataques de inundação ou negação de serviço. Ele verifica se os pacotes estão dentro do limite determinado e derruba pacotes suspeitos.

2.4.3 Detecção de Intrusão

Certos ataques de rede possuem características ou assinaturas específicas. Estas características são monitoradas e quando o CBAC detecta um ataque ele reinicia as conexões que podem ser ofensivas e envia as informações para bloqueio da conexão.

2.4.4 Geração de Alertas e Auditoria

O CBAC cria *logs* com todos os dados necessários para um relatório avançado baseado em sessão. Ele armazena informações de rede, data e hora, origem e destino, portas usadas e quantidade de *bytes* transmitidos o que é um item muito importante, pois todo e qualquer sistema pode falhar de forma inesperada. Além disso, ele gera alertas em tempo real ao detectar qualquer atividade suspeita.

3 SIMULAÇÃO E ANÁLISE

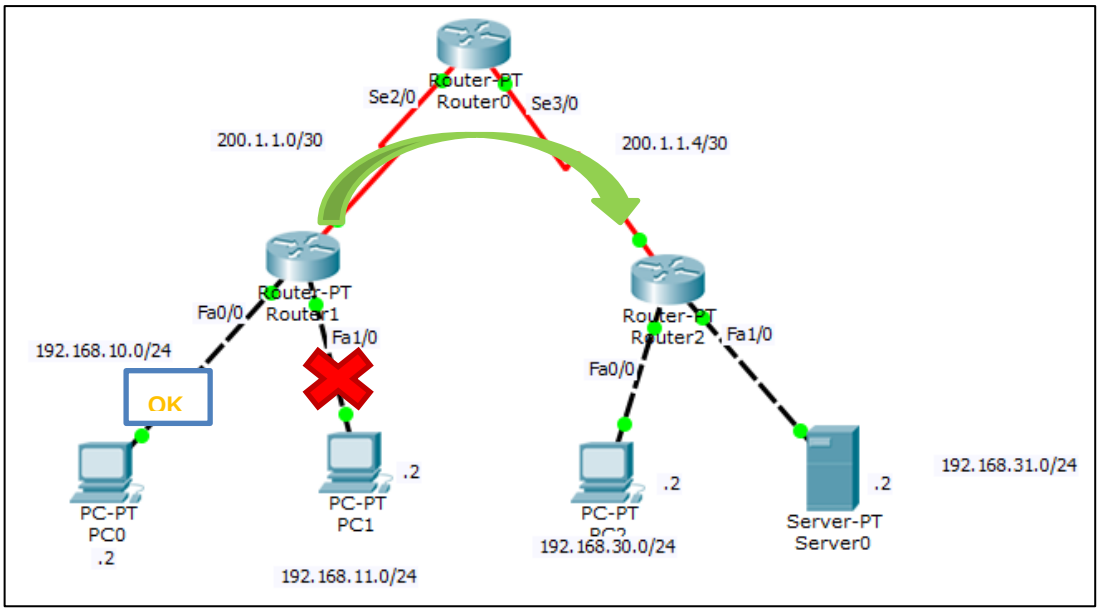
O desenvolvimento deste trabalho envolve diversas simulações executadas no simulador *Packet Tracer*, que permite a criação de várias redes, a interligação entre elas e a aplicação de serviços e testes. Cada simulação deverá bloquear ou liberar um *host* ou um protocolo utilizando ACLs e CBAC.

Para descrever e testar cada bloqueio ou liberação proposto, em cada simulação será exibida a topologia, as configurações aplicadas aos roteadores. Em seguida serão executados testes de *ping* e *telnet* de acordo com o proposto em cada simulação e será apresentado o resultado.

3.1 PRIMEIRA SIMULAÇÃO

Nesta simulação a ACL deverá permitir a rede 192.168.10.0/24 e negar a rede 192.168.11.0/24 de acessar as redes 192.168.30.0 e 192.168.31.0, conforme mostra a Figura 11.

Figura 11 - Topologia Simulações ACL



Fonte: Autoria Própria

Para que a rede 192.168.10.0/24 seja permitida e a rede 192.168.11.0/24 seja impedida de acessar as outras redes, o Roteador 1 possui a configuração

mostrada na Figura 12, onde apenas a rede 192.168.10.0 tem permissão de acesso e a ACL é aplicada na interface Serial2/0 no sentido de saída.

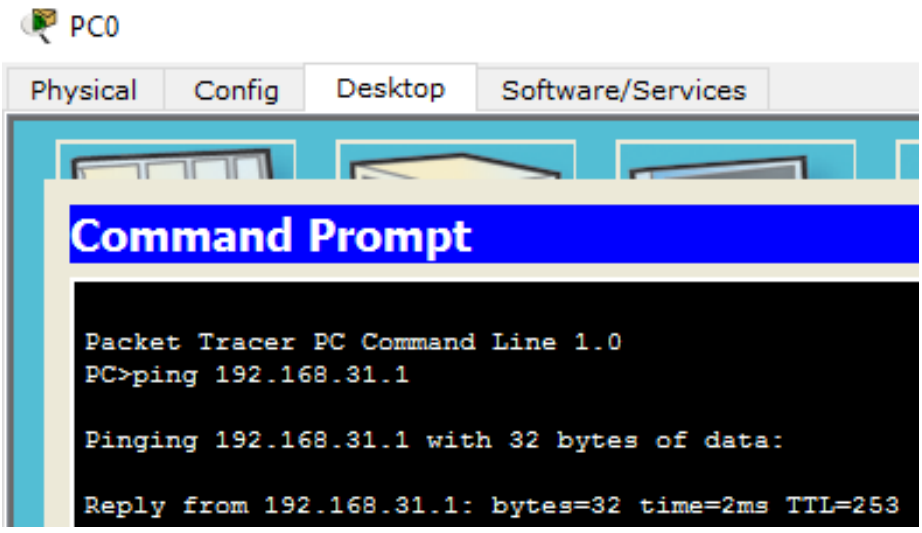
Figura 12 - ACL Primeira Simulação

```
interface Serial2/0
 ip address 200.1.1.2 255.255.255.252
 ip access-group 1 out
 clock rate 2000000
!
!
access-list 1 permit 192.168.10.0 0.0.0.255
!
```

Fonte: Autoria Própria

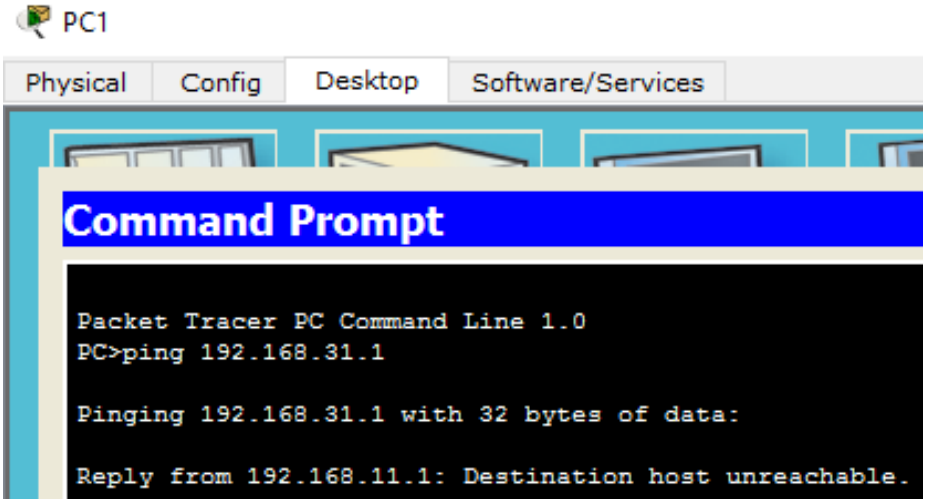
As Figuras 13 e 14 mostram o teste de *ping* executado a partir das redes 192.168.10.0 e 192.168.11.0 para a rede 192.168.31.0. Na Figura 13 o teste recebe a resposta da rede destino e na Figura 14 o mesmo teste tem a resposta negada pelo destino.

Figura 13 – Teste de Ping da rede 192.168.10.0 para a rede 192.168.31.0 aceito



Fonte: Autoria Própria

Figura 14 - Teste de Ping da rede 192.168.11.0 para a rede 192.168.31.0 negado



Fonte: Autoria Própria

Figura 15 - Resultado da ACL

```
Router#show access-list
Standard IP access list 1
    10 permit 192.168.10.0 0.0.0.255 (2 match(es))
```

Fonte: Autoria Própria

A Figura 15 apresenta o resultado da ACL, onde a palavra *match* indica que a ACL foi acionada duas vezes permitindo a rede 192.168.10.0.

3.2 SEGUNDA SIMULAÇÃO

Nesta simulação a ACL deverá negar o host 192.168.11.2 e permitir o host 192.168.10.2 de acessar a rede 192.168.30.0 e 192.168.31.0, conforme mostra a Figura 11. Para que isso aconteça, o roteador 1 possui configuração mostrada na Figura 16, onde apenas o *host* 192.168.10.2 tem permissão de acesso e a ACL é aplicada na interface Serial2/0 no sentido de saída.

Figura 16 - ACL Segunda Simulação

```
!
interface Serial2/0
 ip address 200.1.1.2 255.255.255.252
 ip access-group 1 out
 clock rate 2000000
!
access-list 1 permit host 192.168.10.2
!
```

Fonte: Autoria Própria

A mesma configuração pode ser feita de outra forma, como mostrado na Figura 17. Nesta ACL, ao invés de definir a permissão ao *host* 192.168.10.2, é definido a negação ao *host* 192.168.11.2.

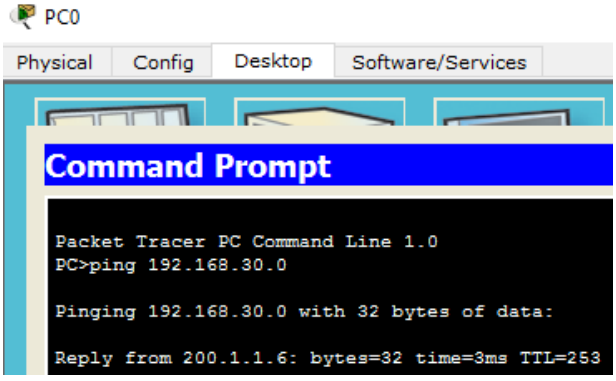
Figura 17 - Variação ACL Segunda Simulação

```
!
interface Serial2/0
 ip address 200.1.1.2 255.255.255.252
 ip access-group 1 out
 clock rate 2000000
!
access-list 1 deny host 192.168.11.2
access-list 1 permit any
!
```

Fonte: Autoria Própria

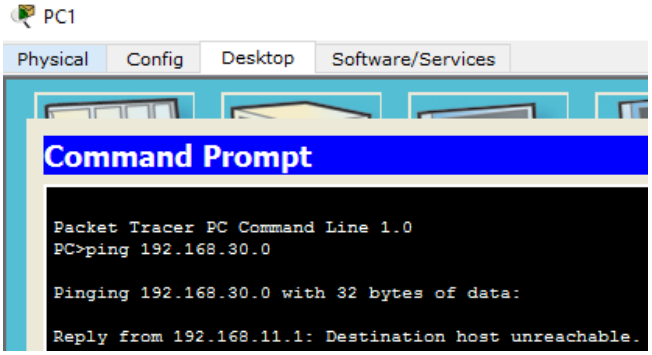
As Figuras 18 e 19 mostram o teste de *ping* executado a partir dos *hosts* 192.168.10.2 e 192.168.11.2 para a rede 192.168.30.0. Na Figura 18 o teste recebe a resposta da rede destino e na Figura 19 o mesmo teste tem a resposta negada pelo destino.

Figura 18 - Teste *Ping* do host 192.168.10.2 para a rede 192.168.30.0



Fonte: Autoria Própria

Figura 19 - Teste *Ping* do host 192.168.11.2 para a rede 192.168.30.0



Fonte: Autoria Própria

Figura 20 - Resultado da ACL

```
Router#show access-lists
Standard IP access list 1
    10 permit host 192.168.10.2 (3 match(es))
```

Fonte: Autoria Própria

A Figura 20 apresenta o resultado da ACL, onde a palavra *match* indica que a ACL foi acionada três vezes permitindo o *host* 192.168.10.2.

3.3 TERCEIRA SIMULAÇÃO

Esta simulação representa uma ACL padrão nomeada para negar uma máquina específica. O roteador 1 possui a configuração mostrada na Figura 21, nesta configuração o *host* 192.168.11.2 é negado de acessar outra rede e a ACL NEGAR_IP é aplicada na interface Serial2/0 no sentido de saída.

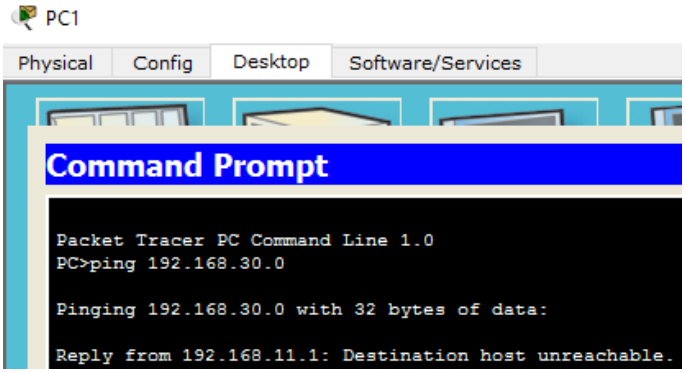
Figura 21 - ACL Nomeada Terceira Simulação

```
!
interface Serial2/0
 ip address 200.1.1.2 255.255.255.252
 ip access-group NEGAR_IP out
 clock rate 2000000
!
ip access-list standard NEGAR_IP
 deny host 192.168.11.2
 permit any
!
```

Fonte: Autoria Própria

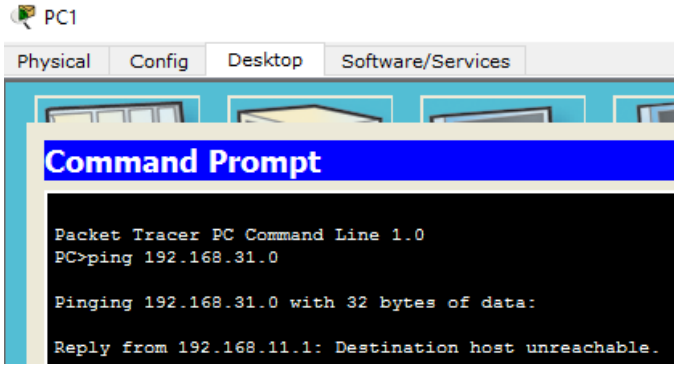
As Figuras 22 e 23 mostram o teste de *ping* executado a partir do *host* 192.168.11.2 para as redes 192.168.30.0 e 192.168.31.0. Na Figura 22 o teste para a rede 192.168.30.0 tem a resposta negada e o mesmo acontece na Figura 23 que executa o teste para a rede 192.168.31.0.

Figura 22- Teste de *Ping* do host 192.168.11.2 para a rede 192.168.30.0



Fonte: Autoria Própria

Figura 23 - Teste de Ping do host 192.168.11.2 para a rede 192.168.31.0



Fonte: Autoria Própria

Figura 24 - Resultado da ACL

```
Standard IP access list NEGAR_IP
    10 deny host 192.168.11.2 (15 match(es))
    20 permit any
```

Fonte: Autoria Própria

A Figura 24 apresenta o resultado da ACL, onde a palavra *match* indica que a ACL foi acionada quinze vezes bloqueando o *host* 192.168.11.2.

3.4 QUARTA SIMULAÇÃO

Nesta simulação será liberado o serviço *Telnet* no roteador 0 e será criado uma ACL para liberar o *Telnet* apenas para um *host* específico. Para liberar o serviço, o roteador 0 possui a configuração mostrada na Figura 25 onde o serviço é permitido e o acesso exige a utilização da senha ‘rede’. Para permitir o serviço apenas para um *host* o roteador recebe a ACL mostrada na Figura 26, onde o *host* 192.168.30.2 é liberado.

Figura 25 - Configuração Telnet

```
!
line vty 0 4
 access-class 1 in
 password rede
 login
!
```

Fonte: Autoria Própria

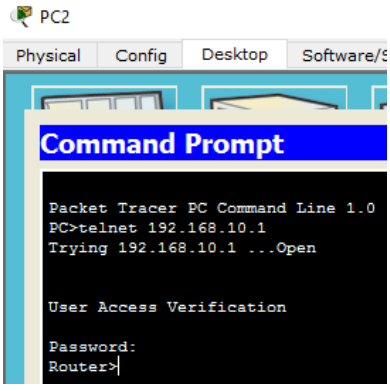
Figura 26 - ACL Quarta Simulação

```
!
access-list 1 permit host 192.168.30.2
!
```

Fonte: Autoria Própria

O teste de *Telnet* executado a partir do *host* 192.168.30.2 para o roteador 1 no IP 192.168.10.1 é mostrado na Figura 27, onde a conexão é aceita após a inserção da senha ‘rede’ definida anteriormente no roteador 0.

Figura 27 - Teste de Telnet no roteador 1



Fonte: Autoria Própria

Figura 28 - Resultado da ACL

```
Router#sh access-lists
Standard IP access list 1
    10 permit host 192.168.30.2 (4 match(es))
```

Fonte: Autoria Própria

A Figura 28 apresenta o resultado da ACL, onde a palavra *match* indica que a ACL foi acionada quatro vezes permitindo o *host* 192.168.30.2.

3.5 QUINTA SIMULAÇÃO

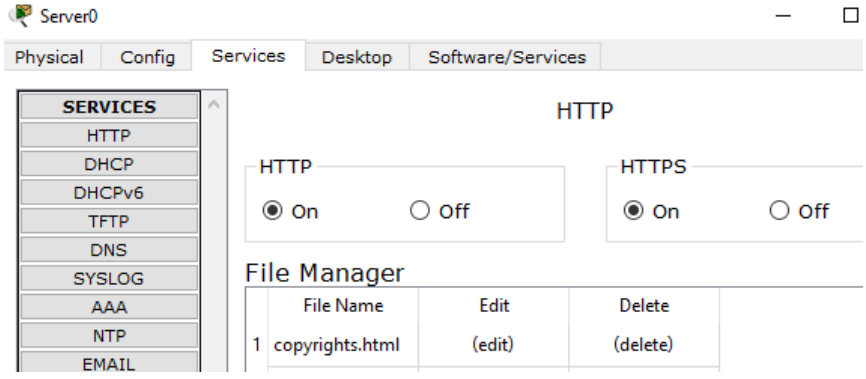
Na quinta simulação será utilizada uma ACL Estendida para permitir que o *host* 192.168.10.2 acesse o servidor *WEB* (192.168.31.2) e para negar o acesso para o *host* 192.168.11.2. Para isto, o roteador 1 possui a configuração mostrada na Figura 29, onde apenas o *host* 192.168.10.2 tem as portas 80 e 443 liberadas e a ACL é aplicada na interface Serial2/0 no sentido de saída. Além disso, o servidor *WEB* também precisa ter os serviços ativos, como mostrado na Figura 30.

Figura 29 - ACL Quinta Simulação

```
!
interface Serial2/0
 ip address 200.1.1.2 255.255.255.252
 ip access-group 100 out
 clock rate 2000000
!
access-list 100 permit tcp host 192.168.10.2 host 192.168.31.2 eq www
access-list 100 permit tcp host 192.168.10.2 host 192.168.31.2 eq 443
!
```

Fonte: Autoria Própria

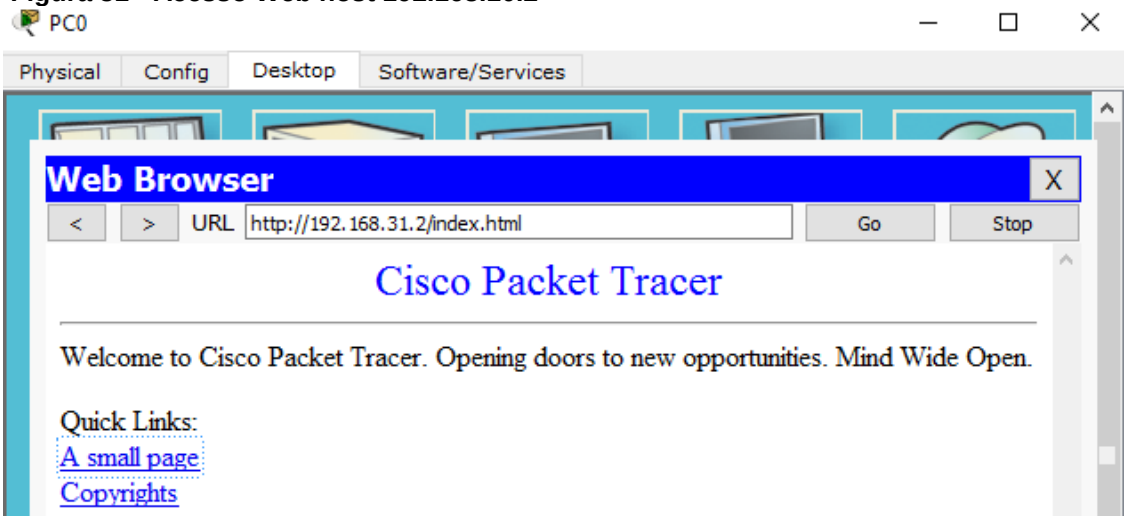
Figura 30 - Serviços ativos no servidor WEB



Fonte: Autoria Própria

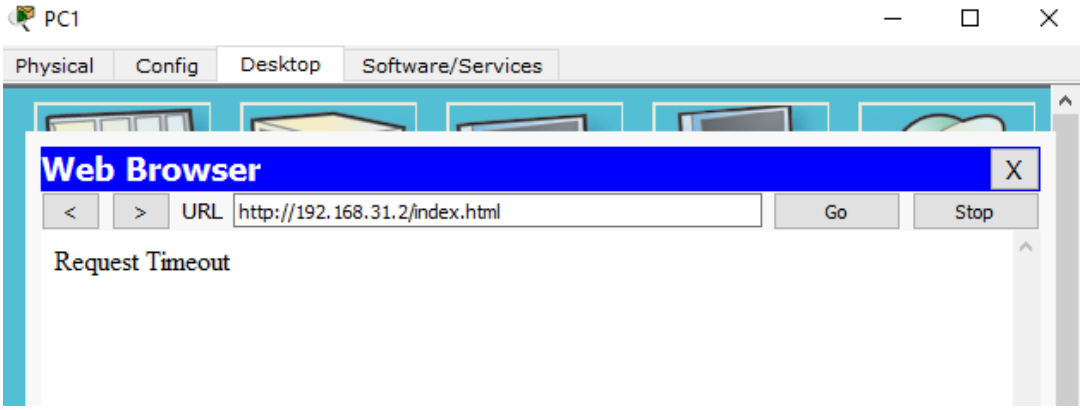
Para testar esta simulação os *hosts* 192.168.10.2 e 192.168.11.2 irão tentar carregar o documento *index.html* que é disponibilizado pelo servidor *WEB* através do navegador. Na Figura 31 o *host* 192.168.10.2 consegue carregar o documento como proposto pela simulação e na Figura 32 o *host* 192.168.11.2 tem o tempo de resposta do servidor *WEB* esgotado, pois não tem permissão de acesso definida.

Figura 31 - Acesso Web host 192.168.10.2



Fonte: Autoria Própria

Figura 32 - Acesso Web host 192.168.11.2 negado



Fonte: Autoria Própria

Figura 33 - Resultado da ACL

```
Router#sh access-lists
Extended IP access list 100
    10 permit tcp host 192.168.10.2 host 192.168.31.2 eq www (103 match(es))
    20 permit tcp host 192.168.10.2 host 192.168.31.2 eq 443
```

Fonte: Autoria Própria

A Figura 34 apresenta o resultado da ACL, onde a palavra match indica que a ACL foi acionada cento e três vezes permitindo o *host* 192.168.10.2.

3.6 SEXTA SIMULAÇÃO

Esta simulação é similar a quinta simulação, a diferença é que será utilizado uma ACL estendida nomeada para permitir que o *host* 192.168.10.2 acesse o servidor *WEB* (192.168.31.2) e para negar que o *host* 192.168.11.2 tenha acesso. Para isto o roteador 1 recebe a configuração mostrada na Figura 34 onde o *host* 192.168.10.2 é o único a ter as portas 80 e 443 liberadas e a ACL LIBERAR_WEB é aplicada na interface Serial2/0 no sentido de saída.

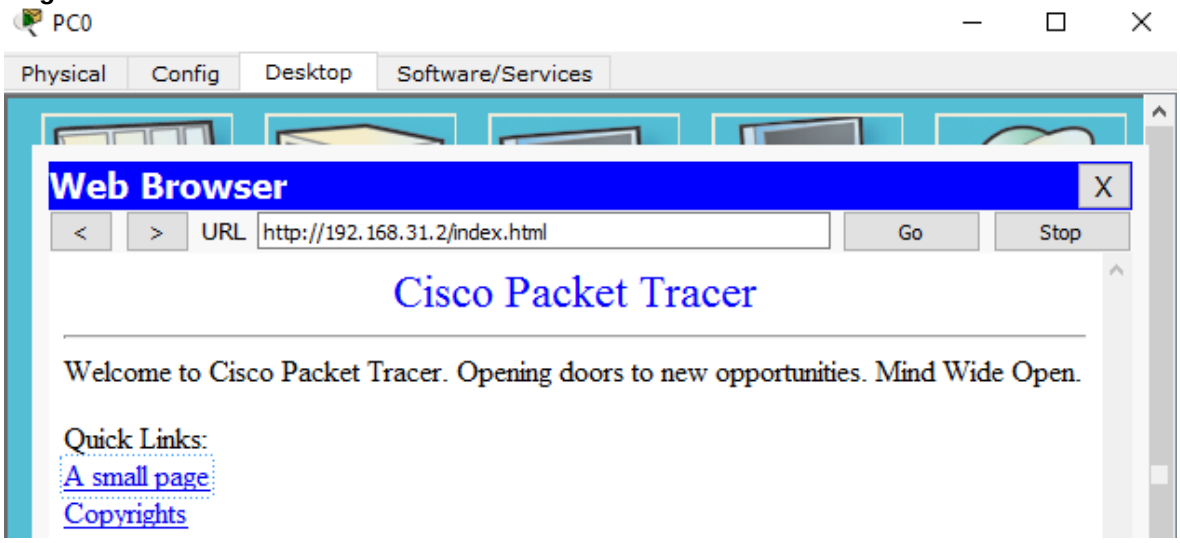
Figura 34 - ACL Sexta Simulação

```
!
interface Serial2/0
 ip address 200.1.1.2 255.255.255.252
 ip access-group LIBERAR_WEB out
 clock rate 2000000
!
ip access-list extended LIBERAR_WEB
 permit tcp host 192.168.10.2 host 192.168.31.2 eq 80
 permit tcp host 192.168.10.2 host 192.168.31.2 eq 443
!
```

Fonte: Autoria Própria

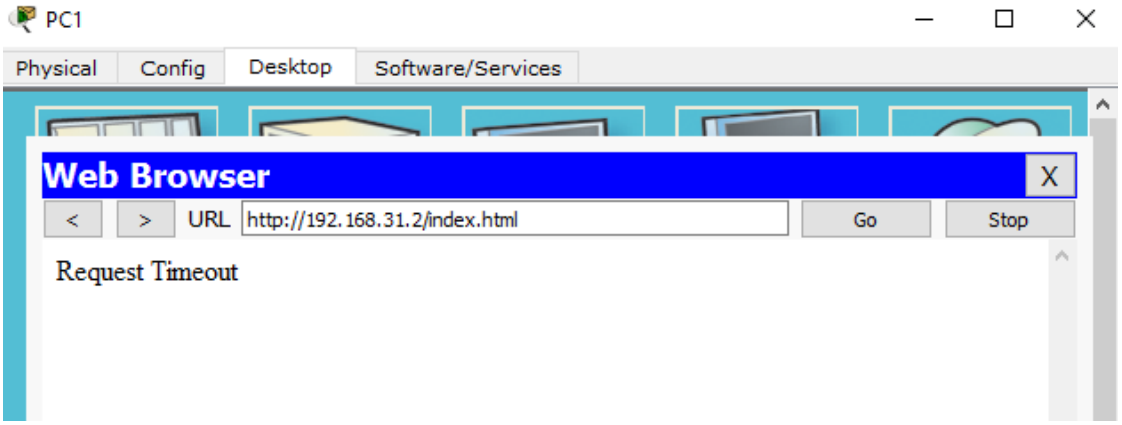
Para testar esta simulação os *hosts* 192.168.10.2 e 192.168.11.2 irão tentar carregar o documento *index.html* que é disponibilizado pelo servidor *WEB* através do navegador. Na Figura 35 o *host* 192.168.10.2 consegue carregar o documento como proposto pela simulação e na Figura 36 o *host* 192.168.11.2 tem o tempo de resposta do servidor *WEB* esgotado, pois não tem permissão de acesso definida.

Figura 35 - Acesso Web host 192.168.10.2



Fonte: Autoria Própria

Figura 36 - Acesso WEB PC1 negado



Fonte: Autoria Própria

Figura 37 - Resultado da ACL

```
Router#sh access-lists
Extended IP access list LIBERAR_WEB
    10 permit tcp host 192.168.10.2 host 192.168.31.2 eq www (11 match(es))
    20 permit tcp host 192.168.10.2 host 192.168.31.2 eq 443
```

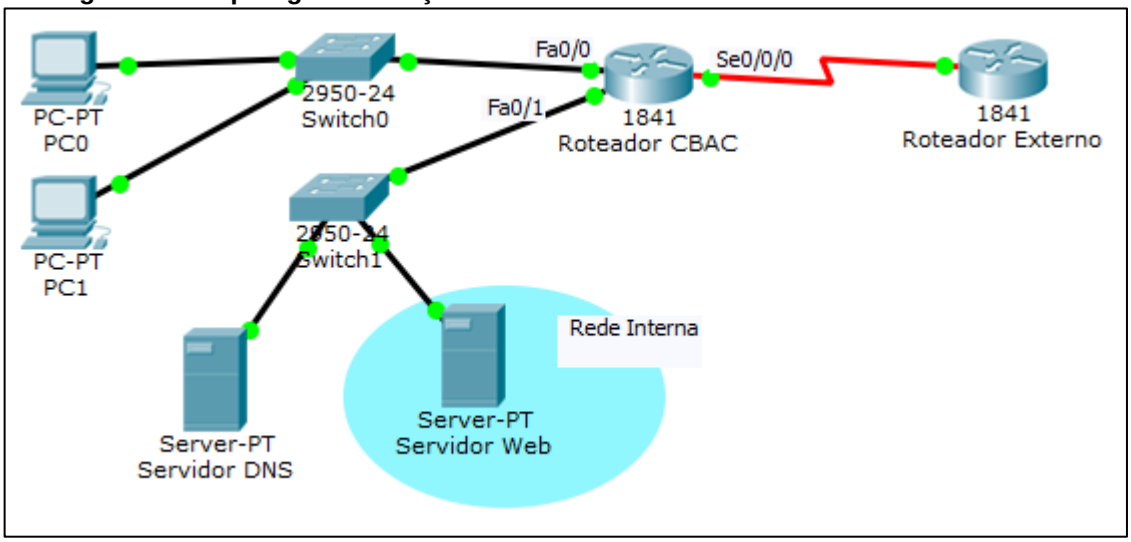
Fonte: Autoria Própria

A Figura 37 apresenta o resultado da ACL, onde a palavra match indica que a ACL foi acionada onze vezes permitindo o *host* 192.168.10.2.

3.7 SÉTIMA SIMULAÇÃO

Na sétima simulação será utilizada a filtragem com CBAC, conforme mostra a Figura 38. O roteador será configurado para negar todo o tráfego da rede externa para dentro da rede interna, permitindo apenas que a rede interna tenha resposta dos testes de *ping* e *telnet*.

Figura 38 - Topologia Simulação CBAC



Fonte: Autoria Própria

Para que tal filtragem aconteça, o Roteador CBAC possui as configurações mostradas nas Figuras 39, 40 e 41 e as ACLs são aplicadas no sentido de entrada às interfaces Serial0/0/0, *FastEthernet*0/1 e *FastEthernet*0/0, sucessivamente.

Figura 39 - ACL interface Serial0/0/0

```
!
interface Serial0/0/0
 ip address 10.1.1.1 255.255.255.0
 ip access-group ext_acl in
 clock rate 125000
!
ip access-list extended ext_acl
 permit icmp any host 192.1.2.2
 deny ip any any
!
```

Fonte: Autoria Própria

A Figura 39 mostra a ACL estendida nomeada “ext_acl” que é aplicada na interface Serial0/0/0 no sentido de entrada. Nesta ACL o Protocolo ICMP (*Internet Control Message Protocol* – Protocolo de Mensagens de Controle da Internet) é liberado para o *host* 192.1.2.2 que é o servidor DNS da rede.

Figura 40 - ACL interface FastEthernet0/1

```
!
interface FastEthernet0/1
 ip address 192.1.2.1 255.255.255.0
 ip access-group host_acl in
 ip inspect host_cbac in
 duplex auto
 speed auto
!
ip access-list extended host_acl
 permit icmp any any
 permit tcp any any
!
```

Fonte: Autoria Própria

A Figura 40 mostra a ACL estendida nomeada “host_acl” que é aplicada na interface *FastEthernet*0/1 no sentido de entrada. Nesta ACL o Protocolo ICMP (*Internet Control Message Protocol* – Protocolo de Mensagens de Controle da Internet) e o Protocolo TCP são liberados para o *host* 192.1.2.1 que é o servidor *WEB* da rede.

Figura 41 - ACL interface FastEthernet0/0

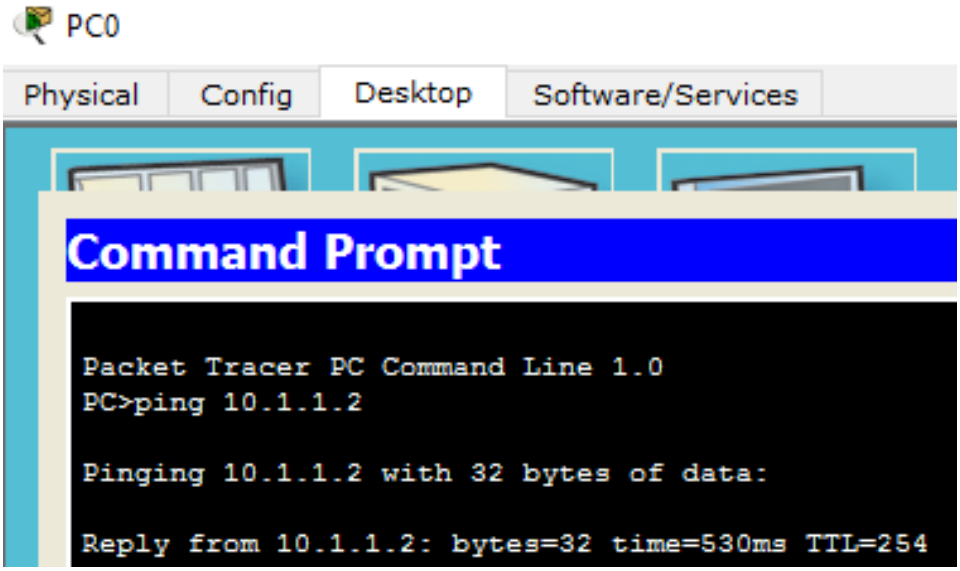
```
!
interface FastEthernet0/0
 ip address 192.1.1.2 255.255.255.0
 ip access-group int_acl in
 ip inspect t_cbac in
 duplex auto
 speed auto
!
ip access-list extended int_acl
 permit ip any any
!
```

Fonte: Autoria Própria

A Figura 41 mostra a ACL estendida nomeada “int_acl” que é aplicada na interface *FastEthernet*0/0 no sentido de entrada. Nesta ACL o Protocolo IP é liberado para o *host* 192.1.1.2 que é o *Gateway* da rede.

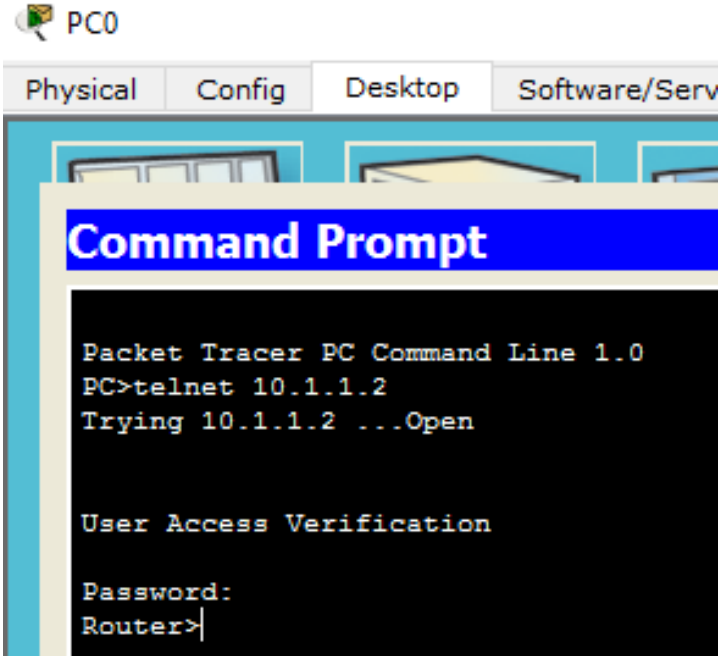
Para mostrar a funcionalidade das ACLs criadas, será testado o *ping* e o *telnet* a partir do PC0 (*host* 192.1.1.1) para a rede externa. As Figuras 42 e 43 mostram o resultado dos testes, na Figura 42 o teste de *ping* obtém resposta do Roteador Externo e o mesmo acontece na Figura 43 para o teste de *telnet.*

Figura 42- Teste de Ping do host 192.1.1.1 para o Roteador Externo



Fonte: Autoria Própria

Figura 43 - Teste de Telnet do host 192.1.1.1 para o Roteador Externo



Fonte: Autoria Própria

Figura 44 - Resultado da ACL

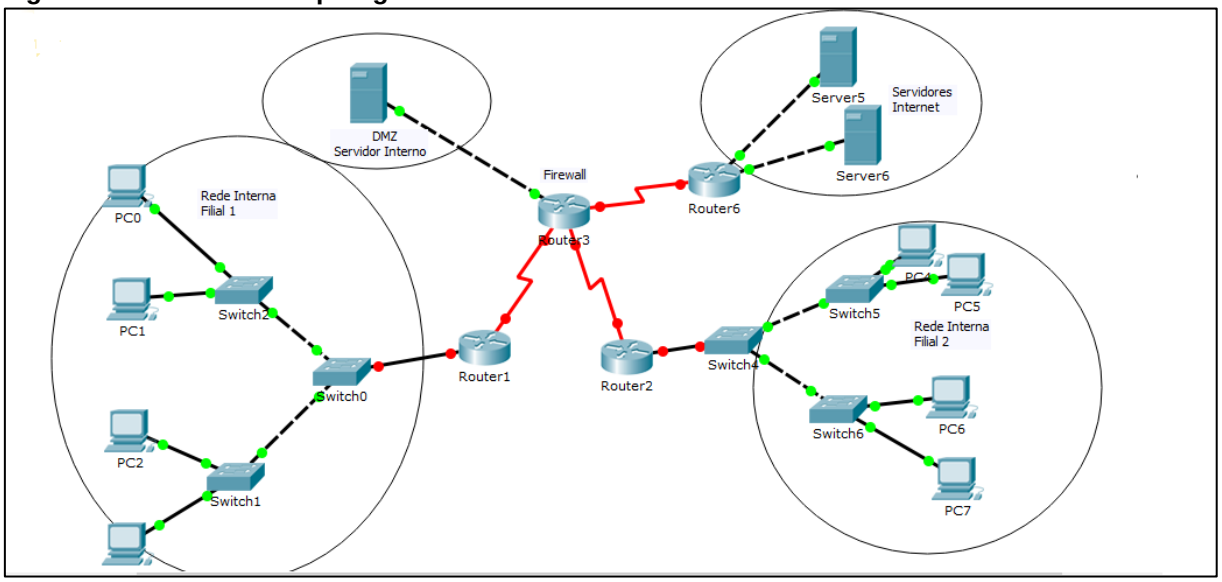
```
cbac_router#sh access-lists
Extended IP access list ext_acl
    10 permit icmp any host 192.1.2.2
    20 deny ip any any (2 match(es))
Extended IP access list host_acl
    10 permit icmp any any
    20 permit tcp any any
Extended IP access list int_acl
    10 permit ip any any (55 match(es))
```

Fonte: Autoria Própria

A Figura 44 apresenta o resultado das ACLs aplicadas ao Roteador Externo, onde a ACL “ext_acl” foi requisitada duas vezes permitindo o *host* 192.1.2.2, que é o DNS da rede interna, a ACL “host_acl” não foi requisitada e a ACL “int_acl” foi requisitada 55 vezes, conforme indicado pela palavra *match*.

As simulações apresentadas foram aplicadas em um contexto mais simples para facilitar o entendimento das diversas formas que *Firewall* pode ser utilizado em uma rede. Seguindo estas simulações é possível aplicar tais configurações em uma topologia real e complexa, como mostrado na Figura 45.

Figura 45 - Modelo de Topologia Real



Fonte: Autoria Própria

A Figura 45 representa uma topologia mais próxima da realidade, onde uma organização possui duas filiais e a matriz, onde os servidores internos e externos ficam alocados. O servidor interno está na região DMZ, os externos na Internet e o *Firewall* está alocado no Roteador 3, controlando o tráfego das Filiais para os servidores internos e externos, o tráfego da Internet para os servidores internos e também o tráfego entre as Filiais.

4 CONCLUSÃO

Com base nas revisões e simulações apresentadas neste trabalho, conclui-se que o estudo, o aprimoramento e a aplicação do serviço de *Firewall* são de extrema importância para a segurança das informações que são partilhadas na Internet mundial, visto que a velocidade em que esta rede se expande é incontrolável e que este meio é utilizado para todos os tipos de tarefas e por todos os tipos de pessoas.

Foram demonstrados dois tipos de *Firewall* entre tantos outros que existem, além de outras ferramentas e aplicações que são base para a contextualização e desenvolvimento de tal funcionalidade. Este trabalho serve ainda como base para diversos trabalhos futuros, onde poderá ser abordados temas para comparação e levantamento de informações sobre outras tecnologias como o *IPTables* e também para a aplicação das regras aqui demonstradas em um ambiente real.

REFERÊNCIAS

COMER, Douglas E., **Interligação de Redes com TCP/IP**. Vol. 1. Rio de Janeiro: Campus, 1998.

FILIPPETTI, Marco A., **CCNA 4.1: Guia Completo de Estudos**. Florianópolis: Visual Books, 2008.

KUROSE, James F. **Redes de Computadores e a Internet**, 5ª Ed. São Paulo: Addison Wesley, 2006.

LEIDEN, Candace; WILENSKY, Marshall. **TCP/IP *for Dummies***, 6ª Ed. Hoboken: Wiley Publishing, Inc., 2009.

NAKAMURA, E. T.; GEUS, P. L. **Segurança de Redes em Ambientes Cooperativos**, 2ª Ed. São Paulo: Novatec Editora, 2007.

NETO, Urubatan. **Dominando Linux *Firewall Iptables***. Rio de Janeiro: Editora Ciência Moderna Ltda., 2004.

SANTOS, Omar; STUPPI, John. **CCNA Security 210-260 Official Cert Guide.** Cisco Press, 2015.

SOARES, Luiz F. G.; LEMOS, Guido; COLCHER, Sérgio, **Redes de Computadores: das LANs, MANs e WANs às Redes ATM**. Rio de Janeiro: Campus, 1995.

SOUSA, Lindeberg Barros de. **Redes de Computadores – Dados, Voz e Imagem**, 3ª Ed. São Paulo: Érica, 1999.

TANENBAUM, Andrews S., **Redes de Computadores**, 4ª Ed. Rio de Janeiro: Campus, 2003.

ZACKER, Craig; DOYLE, Paul. **Redes de Computadores – Configuração, Manutenção e Expansão.** São Paulo: Makron Books, 2000.