UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GIOVANI TOGNON

FIREWALL DE BAIXO CUSTO PARA EMPRESAS DE PEQUENO PORTE COM UM EQUIPAMENTO ARM

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2016

GIOVANI TOGNON

FIREWALL DE BAIXO CUSTO PARA EMPRESAS DE PEQUENO PORTE COM UM EQUIPAMENTO ARM

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Eden Ricardo Dosciatti

PATO BRANCO
2016

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Tecnologia em Análise e Desenvolvimento de Sistemas

TERMO DE APROVAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO

FIREWALL DE BAIXO CUSTO PARA EMPRESAS DE PEQUENO PORTE COM UM EQUIPAMENTO ARM

por

GIOVANI TOGNON

Este trabalho de conclusão de curso foi apresentado no dia 21 de junho de 2016, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho APROVADO.

Banca examinadora:

Prof. Dr. Eden Ricardo Dosciatti
Orientador

Prof. Me. Alisson Andrey Puska

Prof. Dr. Fábio Favarim

Prof. Dr. Edilson Pontarolo
Coordenador do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas

Profª. Me. Soelaine Rodrigues Ascari
Responsável pela Atividade de Trabalho de Conclusão de Curso

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

TONGON, Giovani. Firewall de baixo custo para empresas de pequeno porte com um equipamento ARM. 2016. 75 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2016.

Com o objetivo de fornecer um firewall de baixo custo para empresas de pequeno porte com tecnologia ARM, este projeto usa hardware e softwares de código aberto. Com a instalação do sistema operacional Linux no equipamento ARM foi implementado um firewall, com o intuito de proteger as aplicações contra ataques oriundos da Internet. Também foi implantado um Proxy com autenticação por usuário e senha para liberação do acesso à Internet e com filtros de conteúdo que efetuam a liberação ou bloqueios de acessos indevidos feitos pelos usuários. O projeto contém um sistema para geração de relatórios diários dos acessos de cada usuário com a intenção de aprimorar os filtros de conteúdo. Com o projeto em funcionamento foram aplicados testes a fim de analisar a quantidade de banda suportada pelo equipamento e a quantidade de requisições HTTP que o Firewall consegue processar.

Palavras-chave: ARM. Firewall. Proxy. Linux. Código Aberto.

ABSTRACT

TOGNON, Giovani. Low cost firewall for small businesses with an ARM device. 2016. 75 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2016.

In order to provide a low-cost firewall for small businesses with ARM technology, this project uses hardware and open source software. With the installation of the Linux operating system on ARM equipment we have implemented a firewall, in order to protect the application from attacks originating from the Internet. It was also implanted with a proxy for user authentication and password to release access to the Internet and content filters that effect the release or unauthorized access locks made by users. The project contains a system to generate daily reports of each user access with the intention of improving the content filters. With the project running tests were used to analyze the amount of bandwidth supported by the equipment and the number of HTTP requests that the firewall can process.

Keywords: ARM. Firewall. Proxy. Linux. Open Source.

LISTA DE FIGURAS

FIGURA 1 - ESTATÍSTICAS DE ATAQUES 8
FIGURA 2 - ESQUEMA DE REDE COM FIREWALL 9
FIGURA 3 - PLACA CUBIEBOARD 14
FIGURA 4 - PLACA CUBIETRUCK 14
FIGURA 5 - WHEEZY SERVER DESCOMPACTADO. 21
FIGURA 6 - PROGRAMA SUSE STUDIO IMAGE WRITE. 22
FIGURA 7 - CONFIGURAÇÃO PADRÃO DE REDE. 24
FIGURA 8 - RESPOSTA DO PING. 26
FIGURA 9 - CONTEÚDO DO ARQUIVO PALAVRAS_BLOQUEADAS. 33
FIGURA 10 - CONTEÚDO DO ARQUIVO SITES_BLOQUEADOS. 34
FIGURA 11 - CONTEÚDO DO ARQUIVO IPS_LIBERADOS. 35
FIGURA 12 - CONFIGURAÇÃO DE IP DO USUÁRIO. 37
FIGURA 13 - CONFIGURAÇÃO DO PROXY. 38
FIGURA 14 - CONFIGURAÇÃO AVANÇADA DO PROXY. 38
FIGURA 15 - SOLICITAÇÃO DE USUÁRIO E SENHA. 39
FIGURA 16 - MENSAGEM DE BLOQUEIO. 39
FIGURA 17 - CONTEÚDO DO DIRETÓRIO SARG. 48
FIGURA 18 - PÁGINA DO SARG. 51
FIGURA 19 - PÁGINA DE USUÁRIOS. 51
FIGURA 20 - TOPOLOGIA DA PRIMEIRO ETAPA. 53
FIGURA 21 - PROGRAMA JPERF. 55
FIGURA 22 - GRÁFICO DE RESULTADOS. 56
FIGURA 23 - TOPOLOGIA DA SEGUNDA ETAPA 57
FIGURA 24 - CONFIGURAÇÕES DAS REQUISIÇÕES HTTP. 58
FIGURA 25 - PLANO DE TESTE 59
FIGURA 26 - GRUPO DE USUÁRIOS. 59
FIGURA 27 - RESULTADOS EM TABELA COM UM USUÁRIO. 60
FIGURA 28 - GRÁFICO AGREGADO COM UM USUÁRIO. 60
FIGURA 29 - RESULTADOS EM TABELA COM TRINTA USUÁRIOS. 61
FIGURA 30 - GRÁFICO AGREGADO COM TRINTA USUÁRIO. 61
FIGURA 31 - RESULTADOS EM TABELA COM SESSENTA USUÁRIOS. 62
FIGURA 32 - GRÁFICO AGREGADO COM SESSENTA USUÁRIO. 62

LISTA DE QUADROS

QUADRO 1 - MATERIAIS UTILIZADOS. 13
QUADRO 2 - LISTA DE OPÇÕES DE REGRAS. 17
QUADRO 3 - LISTA DE OPÇÕES DE DADOS. 18
QUADRO 4 - LISTA DE OPÇÕES DE AÇÕES. 18

LISTA DE SIGLAS

A	Amperes
ACK	*Acknowledgement*
ACL	*Access Control List*
ADSL	*Asymmetric Digital Subscriber Line*
ARM	*Advanced* RISC *Machine*
CTRL	*Control*
DDoS	*Distributed Denial of Service*
DNS	*Domain Name System*
DoS	*Denial of Service*
ETH	*Ethernet*
FIN	*Finish*
FTP	*File Transfer Protocol*
GB	Gigabytes
HDMI	*High-Definition* Multimidia Interface
HTTP	*Hyper Text Transfer Protocol*
HTTPS	*Hyper Text Transfer Protocol Secure*
ICMP	*Internet Control Message Protocol*
ICP	Internet Cache *Protocol*
IP	Internet *Protocol*
IPv4	*Internet Protocol version* 4
LAN	*Local Area Network*
MB	Megabytes
MMS	Microsoft Media Server
NAT	*Network Address Translation*
NCSA	*National Center for Supercomputing Applications*
OSI	*Open Systems Interconnection*
P	Pixels
PSH	*Push*
RAM	*Random Access Memory*
RISC	*Reduced Instruction Set Computer*
RST	*Reset*

SD	*Secure* Digital
SH	Shell Script
SSH	*Secure Shell*
SYN	*Synchronize*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
URG	*Urgent*
URL	*Uniform Resource Locator*
V	Volts
VGA	Video *Graphics Array*
W	Watts
WWW	*World Wide Web*

SUMÁRIO

1 INTRODUÇÃO 2
1.1 CONSIDERAÇÕES INICIAIS 2
1.2 OBJETIVOS 3
1.2.1 Objetivo Geral 3
1.2.2 Objetivos Específicos 4
1.3 JUSTIFICATIVA 4
1.4 ESTRUTURA DO TRABALHO 5
2 REFERENCIAL TEÓRICO 6
2.1 TECNOLOGIA ARM E SISTEMAS OPERACIONAIS 6
2.2 SEGURANÇA 7
2.3 FIREWALL 9
2.4 PROXY 11
2.5 AUDITORIA E FILTROS DE CONTEÚDO 12
3 MATERIAIS E MÉTODO 13
3.1 MATERIAIS 13
3.1.1 Cubietruck 13
3.1.2 Armbian 15
3.1.3 Squid 15
3.1.4 Iptables 16
3.1.5 SARG 19
3.2 MÉTODO 19
4 RESULTADOS 21
4.1 IMPLANTAÇÃO DO SISTEMA OPERACIONAL 21
4.1.1 Instalação do sistema operacional 21
4.1.2 Configuração do Armbian 23
4.2 INSTALAÇÃO E CONFIGURAÇÃO DO PROXY 27
4.3 INSTALAÇÃO E CONFIGURAÇÃO DO FIREWALL 40
4.4 INSTALAÇÃO E CONFIGURAÇÃO DO SARG 46
4.5 TESTES E RESULTADOS OBTIDOS 53
5 CONCLUSÃO 64
REFERÊNCIAS 65

1 INTRODUÇÃO

Este capítulo apresenta uma visão geral do trabalho, os objetivos e a justificativa. A organização do texto está de acordo com a da apresentação de seus capítulos, que consiste nas considerações iniciais, objetivos, justificativa e estrutura do trabalho.

1.1 CONSIDERAÇÕES INICIAIS

O uso de equipamentos ARM (*Advanced* RISC *Machine*) tem se tornado cada vez mais frequente devido ao seu baixo custo aliado a sua capacidade de ser aplicado em diversas áreas e com diversos fins. Um dispositivo ARM nada mais é do que um sistema embarcado, utilizado em *smartphones*, calculadoras e diversos periféricos computacionais. O sistema embarcado é um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou do sistema que ele controla (EMBEDDED, 2016). Com essa tecnologia surge a possibilidade de ampliar significativamente o uso de equipamentos utilizadores da tecnologia ARM.

Com o intuito de reduzir custos para micro e pequenas empresas, optou-se pela utilização de um equipamento chamado de Cubietruck (CUBIETRUCK, 2016). Seu hardware se assemelha muito com o de um servidor, contendo um processador AllWinnerTech dual-core, memória de 2GB (Gigabytes), capacidade de armazenamento interno de 8GB com possibilidade de expansão para cartão SD (*Secure Digital*) ou disco rígido e fonte de 5V (Volts) com 2.5A (Amperes) (AllWinner Technology, 2016). Um servidor possui a mesma topologia de hardware do Cubietruck, o que diferencia um do outro é que o servidor pode ser adicionado memórias ou processadores mais robustos caso haja necessidade, enquanto o Cubietruck não permite nenhuma alteração no seu hardware, pois os componentes são soldados diretamente a placa.

O Cubietruck possui uma fonte que gera aproximadamente 12.5W (Watts) de potência, enquanto um servidor possui uma fonte de 300W de potência, se

ambos os equipamentos ficarem ligados por um mesmo período de tempo, pode haver uma economia de 96% no consumo de energia elétrica mensal entre o dispositivo ARM e o servidor (COPEL, 2016), além disso, o Cubietruck pode ser adquirido por $99,99 (CUBIETRUCK, 2016) na loja oficinal do fabricante, enquanto um servidor parte de $329,00 (DELL, 2016) na loja oficial DELL.

O equipamento Cubietruck é compatível com o sistema operacional Linux (LINUX-SUNXI, 2016), sendo assim, foi instalada uma versão em modo texto do Linux, com distribuição Armbian a fim de obter melhor desempenho. Através dessa instalação foi implantado um Proxy, que é uma ferramenta que funciona como intermediário entre um navegador da Web e a Internet. Segundo Microsoft (2016), o Proxy também ajuda a melhorar a segurança porque filtra diversos tipos de conteúdo da Web e softwares mal-intencionados. Cada usuário possui um usuário e senha para acessar a Internet, configurados no Proxy. Junto a esse firewall foi implantado um sistema para auditorias, que gera relatórios diários de acesso de cada usuário. O *firewall* conta também com um conjunto de regras que tem o objetivo de proteger o Cubietruck e suas aplicações contra ataques.

Ao contrário de outros trabalhos realizados em que foi virtualizado o sistema operacional como é apresentado por Scheer (2016), este trabalho é concretizado através do uso único e exclusivo de equipamento ARM.

1.2 OBJETIVOS

O objetivo geral apresenta os resultados pretendidos com a realização deste trabalho e os objetivos específicos o complementam.

1.2.1 Objetivo Geral

Propor um firewall de baixo custo para empresas de pequeno porte utilizando um dispositivo ARM.

1.2.2 Objetivos Específicos

- Efetuar a instalação do sistema operacional Linux no equipamento ARM;
- Realizar as configurações de Proxy e Firewall;
- Implantar um sistema de autenticação de usuário e senha para liberação do acesso à Internet;
- Configurar uma ferramenta para auditoria e aplicação de filtros de conteúdo;
- Realizar testes para estimar a quantidade de banda suportada pelo equipamento e a quantidade de requisições HTTP que pode processar.

1.3 JUSTIFICATIVA

Os servidores em modo geral são uma necessidade indispensável para qualquer tipo de empresa nesta era digital, no entanto essas máquinas custam caro, necessitam de espaço reservado, além de consumir energia elétrica em elevada quantidade, se tornam custoso para qualquer micro e pequena empresa que pensa em adquirir e manter um *firewall*.

Atualmente estão surgindo novas tecnologias no mercado que envolvem dispositivos ARM. Esse tipo de equipamento, juntamente com o sistema operacional Linux e softwares gratuitos torna-se uma nova alternativa para micro e pequenas empresas que necessitam de um firewall e não tem condição de arcar com altos custos do equipamento. Um firewall com a tecnologia ARM agrega qualidade com baixo custo, tanto na sua aquisição quanto na redução drástica do consumo de energia elétrica e do espaço físico que um servidor convencional necessitaria.

O *firewall* tem a função de proteger a rede corporativa de ameaças, e dentro de uma empresa, poucas delas têm controle do conteúdo que seus colaboradores acessam na Internet em horário de expediente. A Internet tem a função de auxiliar na execução de tarefas e na obtenção de informações, porém, nada impede que os usuários acessem conteúdos impróprios ou ilegais, prejudicando a segurança e acarretando na vulnerabilidade da rede corporativa.

Visando a integridade da empresa e a qualidade do trabalho, torna-se necessário a implantação de um sistema de controle de acesso aos usuários de Internet, com ferramentas livres e gratuitas. O sistema funciona com autenticação de usuário e senha pré-cadastrados para liberação do acesso à Internet. Outra medida a ser tomada é a implantação do *firewall*, que irá prevenir contra ataques de hackers e pacotes maliciosos através de bloqueios de portas.

Além disso, uma ferramenta para auditoria é primordial, sendo possível verificar informações sobre o conteúdo que cada usuário acessou e que equipamento usou. Através dos resultados gerados pela auditoria é possível realizar o aprimoramento dos filtros de conteúdo, aplicando bloqueios ou liberações da navegação através da análise das necessidades de cada usuário.

As ferramentas que compõe o *firewall* também podem ser usadas para proteger aplicações que podem ser hospedados no Cubietruck. Um exemplo de aplicação são os bancos de dados e aplicações web, podem estar protegidos pelo *firewall* do equipamento. Assim sendo, o firewall pode ser usando não só para o usuário final mas também para a proteção de outras aplicações.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em capítulos, aqui é apresentado o primeiro capítulo, que mostra a introdução, juntamente com os objetivos e a justificativa. O Capítulo 2 contém o referencial teórico sobre equipamentos ARM e os sistemas operacionais disponíveis, segurança de rede e auditorias. O Capítulo 3 apresenta os materiais e o método utilizado para alcançar os objetivos pretendidos. No Capítulo 4 são exibidos os resultados com a instalação e configuração do sistema operacional, Proxy, firewall e do sistema de auditoria no equipamento ARM, além de apresentar os testes efetuados e os resultados obtidos. No Capítulo 5, a conclusão deste trabalho.

2 REFERENCIAL TEÓRICO

A fundamentação teórica do trabalho tem como objetivo descrever a utilização de dispositivos ARM com o sistema operacional Linux, funcionando como um Firewall e também como Proxy, além de contar com uma ferramenta para auditoria.

2.1 TECNOLOGIA ARM E SISTEMAS OPERACIONAIS

O uso de processadores ARM é tradicionalmente encontrado nos dispositivos de baixa potência tais como calculadoras, telefones celulares e vários periféricos de informática. No entanto, isso está começando a mudar, pois já é possível encontrar a tecnologia ARM em notebooks e em outros equipamentos que anteriormente usavam processadores contendo uma arquitetura diferente (HOFFMAN, 2016).

Historicamente os processadores RISC (*Reduced Instruction Set Computer*) obtiveram menor consumo de energia e são significamente mais baratos se comparados com os processadores convencionais, como os desenvolvidos pela empresa Intel na arquitetura x86 e x64. Atualmente, a tecnologia ARM tem uma evolução enorme em termos de desempenho, equipamentos como IPhones e IPads, juntamente com a maioria dos *smartphones* e tablets equipados com Android, possuem processadores ARM. Sabe-se da evolução no desempenho desses equipamentos ao longo dos anos, e isso se deve ao avanço da tecnologia ARM (HOFFMAN, 2015).

Os chips ARM atingiram um desempenho similar aos seus concorrentes, pois começaram a ser utilizados em Chromebooks (GOOGLE, 2016) e alguns NetBooks. Porém o que diferencia os processadores ARM dos demais, está na sua arquitetura e no conjunto de instruções, por exemplo, um aplicativo desenvolvido para funcionar em uma arquitetura Intel não será compatível com um equipamento ARM (HOFFMAN, 2015).

Os dispositivos ARM necessitam de um sistema operacional específico, que sejam compatíveis com a sua arquitetura. É possível encontrar várias opções de sistemas operacionais dos mais diversos desenvolvedores como o Windows RT da Microsoft, o Android e o ChromeOS da Google e várias distribuições Linux, como Debian e Ubuntu.

O Windows RT é exclusivo para equipamentos ARM fabricados pela Microsoft, como por exemplo, a série de smartphones Lumia. Ainda que seja um sistema operacional desenvolvido pela Microsoft é incapaz de executar programa desktop e programas que não tenham sido desenvolvidos pela própria empresa. Já os sistemas operacionais desenvolvido pela Google atingem uma gama maior de equipamentos ARM, trazem a possibilidade de rodar aplicativos nativos da empresa e os desenvolvidos por outras companhias. No entanto, a característica dos sistemas operacionais Android e o ChromeOS ainda são limitados para configurações avançadas que vão além do uso convencional, ou seja, não é possível utilizar esses sistemas operacionais como servidor.

Os sistemas operacionais Linux se encaixam nos equipamentos ARM por serem sistemas de código aberto, e com isso é possível criar pacotes de instalação exclusivos para essa arquitetura. Além de existir inúmeras distribuições de sistemas operacionais e as mais diversas possibilidades de configurações e aplicações compatíveis. Com a flexibilidade do sistema operacional Linux é possível desenvolver servidores dos mais diversos tipos, como firewall e Proxy.

2.2 SEGURANÇA

O termo segurança, especificamente na área da tecnologia da informação, pode ser referenciado como “a forma de proteger a informação de diversos tipos de ameaças para garantir a continuidade dos negócios, minimizar os danos aos negócios e maximizar o retorno dos investimentos e as oportunidades de negócio” (ABNT, 2016). Os objetivos principais da segurança da informação visam manter a confidencialidade, integridade e disponibilidade. Cada item tem um objetivo e propósito especifico:

- **Confidencialidade**: Garantia de que a informação somente é visualizada ou divulgada por pessoas autorizadas a terem acesso.
- **Integridade**: Garantia da alteração da informação somente por pessoas autorizadas.
- **Disponibilidade**: Garantir que os usuários autorizados obtenham acesso à informação e aos ativos correspondentes, sempre que necessário.

A segurança das informações é mantida através da implantação de uma série de controles, que podem ser procedimentos, práticas, políticas, estruturas organizacionais e funções de software.

Empresas especializadas em monitoramento de ataques e pragas na Internet têm apresentando dados que podem auxiliar na prevenção da segurança da informação. Como mostra a Figura 1, a quantidade de ataques diários varia entre aproximadamente 25 mil e 115 mil ataques no período compreendido entre abril e maio de 2016. A Figura 1 também apresenta os 10 tipos de ataques mais realizados no mesmo período, com a respectiva porcentagem.

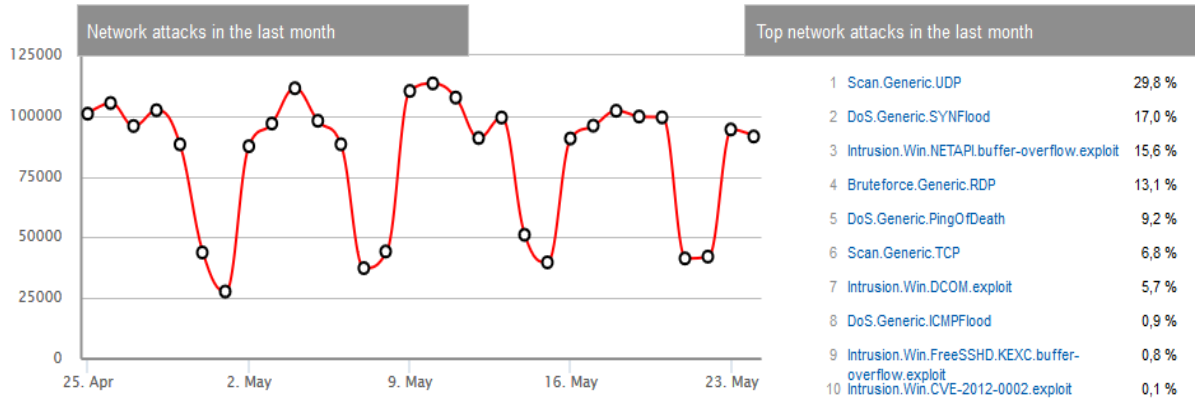

Figura 1 - Estatísticas de ataques
Fonte: SECURELIST (2016).

Na primeira e sexta posições no ranking estão os *scanners* de rede, ferramentas que buscam portas não protegidas e brechas para serem exploradas, tanto na conexão de entrada de dados quanto na de saída. A segunda posição corresponde aos ataques de *SYNFlood*, que é um ataque usando os pacotes que originam uma sessão TCP. O SYN Flood consiste em enviar diversas requisições invalidas de sessão (Pacotes TCP com a *flag SYN (Synchronize)*) para o alvo com o objetivo de saturar sua capacidade de responder a requisições válidas e reais, esse tipo de ataque também é conhecido como ataque DoS (*Denial of Service*). Na quarta

posição está o ataque de força bruta, que consistem na descoberta de usuário e senha de servidores ou equipamentos através de tentativa e erro, seja manualmente ou através de programas específicos. A quinta posição está o PingOfDeath (Ping da Morte) que é um ataque destinado a equipamentos antigos sem atualização do protocolo ICMP (*Internet Control Message Protocol*), este ataque consiste em enviar um pacote ICMP com um *payload* (carga de uma transmissão de dados) que excede a capacidade especificada elo protocolo, sobrecarregando o processamento, causando um *buffer overflow* (transbordamento de dados) que pode travar o sistema ou permitir a injeção de códigos maliciosos. A oitava posição referência DoS por *smurf attacks* ou por DDoS (*Distributed Denial of Service*).

2.3 FIREWALL

Um dos métodos para se precaver dos ataques é a implantação de um *firewall*. O princípio do *firewall* é de que um dispositivo trabalhe como intermediário entre os computadores pessoais e a Internet, como ilustra a Figura 2.

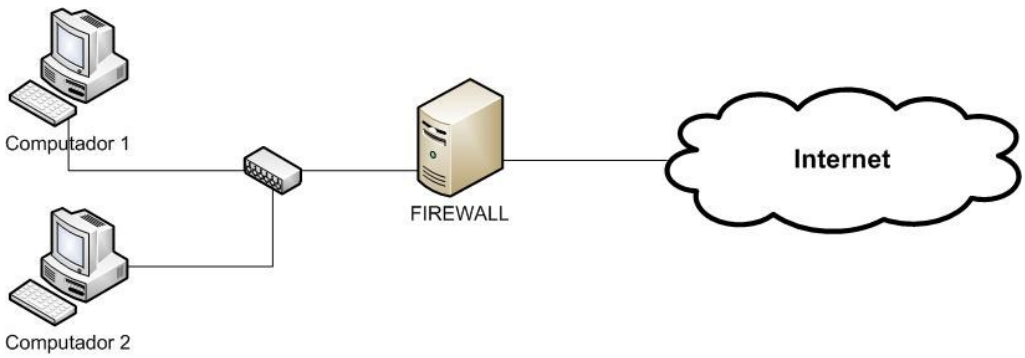

Figura 2 - Esquema de rede com firewall
Fonte: Autoria própria.

O *firewall* funciona de maneira em que a comunicação oriunda de um dos computadores presentes dentro da rede interna, transmita pacotes de dados através do *firewall* que irá filtrar os pacotes, com o intuito de tratar os pacotes que deve passar ou não pelo *firewall*, para depois chegar ao destino desejado na Internet.

Segundo Nakamura e Geus (2010) o *firewall* é formado por diversas funcionalidades, as principais e mais usadas são os filtros, *proxys*, NAT (*Network Address Translation*) e autenticação/certificação.

Os filtros são utilizados para efetuar a análise de pacotes, aceitando ou descartando pacotes por meio de regras de filtragem definidas pela política de segurança da empresa. Os filtros de pacotes atuam na camada de rede e de transporte da pilha TCP, realizando a filtragem com base nas informações como o endereço de origem, o endereço de destino, a porta de origem, a porta de destino e a direção das conexões. Em relação ao ICMP (Internet *Control Message Protocol*), a filtragem é feita com base nos tipos de códigos das mensagens.

Os *proxys* atuam como *gateway* entre duas redes, permitindo a comunicação dos usuários internos conforme a definição da política de segurança. Também podem realizar uma filtragem avançada dos pacotes, pois atua na camada de aplicação do modelo OSI (*Open Systems Interconnection*). O proxy pode trabalhar na camada de sessão, na camada de transporte, ou na camada de aplicação. O funcionamento típico dos *proxys* é que o cliente deve se conectar ao Proxy, que libera o acesso após a autenticação. Uma vez autenticado, o cliente envia sua requisição ao Proxy que repassa ao servidor. A resposta vinda o servidor externo também passa pelo Proxy. Assim, duas conexões são estabelecidas em cada requisição, uma do cliente até o Proxy e outra do Proxy para o servidor.

O NAT não foi desenvolvido para ser aplicado na segurança da rede, seu foco é tratar problemas em redes de grande porte, onde a escassez de endereços IP representa um problema. Sendo o NAT o responsável por converter endereços inválidos e reservados para endereços validos e roteáveis. O NAT também pode ser usado com a intenção de aumentar a segurança da rede, tornando possível esconder os endereços dos equipamentos internos da rede e a sua topologia, precavendo os ataques externos.

A autenticação pode ser baseada por endereços IP (Internet *Protocol*), usuário e senha, entre outros, agregando segurança através da identificação de cada usuário que utiliza a rede.

2.4 PROXY

O Proxy tem a função de intermediar as requisições dos usuários da rede interna com a Internet. Tem o poder de alterar as requisições do cliente ou a resposta do servidor, ou seja, o Proxy funciona como um complemento ao *firewall*, ambos podem trabalhar em conjunto, assim, possibilitando a melhoria do fluxo de dados e da segurança na rede. O Proxy fornece ferramentas mais avançadas que o *firewall*, proporcionando um melhor controle dos usuários, enquanto o *firewall* consegue ser mais eficaz na proteção da rede contra ataques e ameaças vindas da Internet.

O Proxy busca por comunicação através do TCP (*Transmission Control Protocol*) e do ICP (Internet *Cache Protocol*) em determinadas portas. Os pacotes TCP são utilizados para a comunicação entre servidores *web* e clientes, enquanto o ICP é utilizado entre servidores de cache. O servidor Proxy necessita de uma única porta para enviar e receber as respostas requisitadas, tanto para TCP quanto para ICP (MORIMOTO, 2006).

Uma das muitas funcionalidades do Proxy está na possibilidade de impor limites ou restrições, através do endereço IP do computador e do *login* de usuário. As restrições ou bloqueios podem ser definidos para funcionar em horários pré-estabelecidos ou a todo instante, além de possibilitar o bloqueio de páginas com conteúdo indesejado.

Outra possibilidade que o Proxy apresenta é a de funcionar como cache de arquivos e páginas. Quando um usuário acessar pela primeira vez um determinado site, o *cache* armazena dados, como imagens, animações e outros tipos de conteúdo contidos na página. Assim, quando houver um segundo acesso ao mesmo site armazenado, o Proxy irá transmitir o conteúdo armazenado no cache para a máquina do usuário, sem a necessidade de baixar repetidamente o mesmo arquivo da Internet, a cada acesso. Consequentemente o acesso acaba se tornando mais rápido, pois economiza muita banda, além de diminuir o tempo de espera de carregamento da página.

Com o Proxy é possível definir a quantidade de banda que cada usuário poderá usar. A ideia é priorizar usuários que necessitam de mais banda para trabalhar na Internet, dos demais usuários, ou ainda limitar toda a banda utilizada.

2.5 AUDITORIA E FILTROS DE CONTEÚDO

A auditoria realiza-se através das análises dos registros de eventos (*logs*) gerados pelo Proxy ou *firewall.* Esses registros armazenam uma lista de todas as páginas que foram acessadas, de qual máquina da rede partiu o acesso e a data e hora que ocorreu. É possível também obter os acessos baseando-se no *login* de cada usuário, se configurado no Proxy, caso contrário, é exibido o IP das máquinas.

Através dos *logs* é possível implantar os filtros de conteúdo, que têm a função de efetuar o bloqueio de páginas, arquivos, extensões, palavras e outros argumentos. A inspeção manual dos *logs* é maneira mais eficaz de aprimorar os filtros de conteúdo, pois dessa maneira é possível acompanhar as páginas que estão sendo bloqueadas e as que estão tendo acesso livre, assim, controlando possíveis abusos.

3 MATERIAIS E MÉTODO

Este capítulo enfatiza os materiais utilizados no projeto e o método utilizado para a realização das atividades.

3.1 MATERIAIS

As tecnologias e ferramentas que serão utilizadas na implementação do projeto estão descritas no Quadro 1.

Ferramenta / Tecnologia	Versão	Finalidade
Equipamento ARM	Cubietruck	Dispositivo ARM.
Linux	Armbian 4.5	Sistema Operacional.
Iptables	1.4.7	Firewall.
Squid	3.5.7	Proxy.
Sarg	2.3.9	Ferramenta para realização de auditoria.

Quadro 1 - Materiais utilizados.
Fonte: Autoria própria.

3.1.1 Cubietruck

Fundada em 2012, a Cubietech Limited começou com o intuito de pesquisar e desenvolver tecnologias embarcadas, com o objetivo de produzir *hardware* de código aberto. A sua primeira linha de produtos foi a série CubieBoard, um microcomputador de placa única que suporta os mais diversos sistemas operacionais conforme demonstrado na Figura 3.

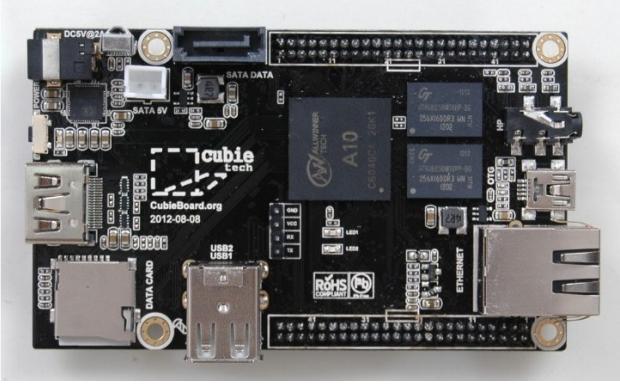

Figura 3 - Placa Cubieboard
Fonte: Lima (2016).

Após 2 anos do surgimento da CubieBoard, foi lançada a sua sucessora que foi chamada de Cubietruck, com hardware mais robusto do que a sua antecessora, foi classificada por comunidades de código aberto como um dos melhores mini computadores já produzidos. Destacando-se pelas suas inovadoras características, a Cubietruck foi lançada com um sistema de comunicação sem fio e antena na própria placa, além de possuir uma placa de rede integrada com velocidade de 1GB e sistema operacional Android como padrão de fábrica. Entre outras características importantes ela tem, memória RAM (*Random Access Memory*) de 2GB e NAND Flash de 8GB. Suas dimensões foram reduzidas, podendo ser utilizada com uma fonte externa de 5V ou com baterias de lítio. Também dispõe de saídas de vídeo VGA (*Video Graphics Array*) e HDMI (*High-Definition Multimidia Interface*) com resolução de 1080p (Pixels), conforme descrito na Figura 4.

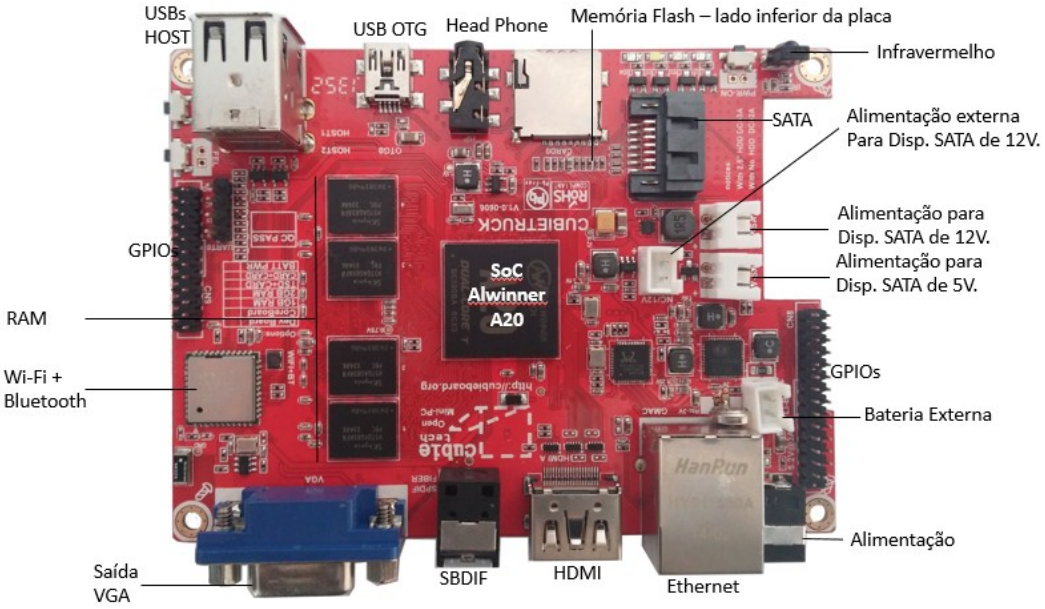

Figura 4 - Placa Cubietruck
Fonte: Lima (2016).

3.1.2 Armbian

A versão do sistema operacional Linux Armbian é baseado no já consagrado sistema operacional Debian, que foi criado em 1993 por Ian Murdock e um grupo pequeno de *hackers* de *softwares* livres. O Debian usa o kernel Linux e possui mais de 43.000 pacotes de *softwares* pré-compilados e livres (DEBIAN, 2016).

O Debian pode ser instalado em qualquer computador ou até mesmo em minicomputadores, pois necessita de apenas 2GB de espaço em disco para funcionar em modo gráfico, no entanto se não for necessário a utilização da interface gráfica o espaço necessário cai para apenas 600MB (Megabytes).

O sistema operacional Debian é usando por várias organizações, sejam elas de grande ou pequeno porte, sem contar os usuários individuais. O principal uso do Debian é na forma de servidores, pois disponibiliza vários serviços, incluindo e-mail, servidor de compartilhamento de arquivos, varredura de segurança e monitoramento, entre outros atrativos.

O que diferencia o Armbian do Debian é que o Armbian é um sistema operacional customizado para equipamentos ARM, ou seja, um sistema pré-configurado com todas os pacotes e funcionalidades que o Debian utiliza mas exclusivo para equipamentos ARM.

O Armbian recebe atualizações constantes, visando sempre aprimorar sua performance e a evolução do sistema operacional voltado para os equipamentos ARM.

3.1.3 Squid

O Squid nasceu em meados dos anos 90, mas há alguns anos vem crescendo cada vez mais e ampliando sua faixa de recursos extras como: um controle de acesso maior e mais eficaz, autorizações, modelagem, registros, distribuição de conteúdo ou replicação e gerenciamento de tráfego (SQUID-CACHE, 2016).

O Squid é um Proxy *cache* que funciona agindo principalmente no controle de acessos para a *web* com suporte HTTP (*Hyper Text Transfer Protocol*), HTTPS (*Hyper Text Transfer Protocol Secure*) e FTP (*File Transfer Protocol*). Sua função é aperfeiçoar o fluxo de dados, reduzindo a largura da banda e aumentando o tempo de resposta do armazenamento em *cache*.

3.1.4 Iptables

O iptables é um *firewall* baseado no tipo endereço/porta de origem/destino do pacote, prioridade, entre outros. O iptables funciona através das regras de comparação, analisando se um pacote tem permissão para passar ou não (SILVA, 2016).

As regras são comandos que determinam a ação a ser tomada com os pacotes, como bloquear ou liberar. As regras são inseridas dentro das chamadas *chains* e processadas na sequência em que são inseridas. Essas regras são escritas em um arquivo e colocadas para inicializar juntamente com o sistema operacional, para que não haja a necessidade de carregar todas as regras a cada vez que o sistema for reiniciado.

As *chains* são regras para classificar o tratamento dos pacotes. Existem dois tipos de *chains*: os padrões ou embutidos como *INPUT*, *OUTPUT* e *FORWARD* e os personalizados pelo próprio usuário.

As tabelas são os locais onde se armazenam as *chains* e o conjunto de regras, no iptables, as tabelas são referenciadas pela opção -t e compostas por 3 tipos:

a) Tabela *filter*: composta por 3 tipos de *chains* padrões:
   INPUT: utilizado para dados que chegam até o firewall.
   OUTPUT: utilizado para dados que saem do firewall.
   FORWARD: utilizado para dados que são redirecionados, seja para outra interface de rede ou outra máquina.
b) Tabela *nat*: tem a função de alterar características de origem ou destino de um pacote através de 3 *chains* padrões:
   PREROUTING: utilizado para modificar os pacotes recebidos.

OUTPUT: utilizado para modificar os pacotes localmente, antes de serem roteados.

POSTROUTING: utilizado para modificar os pacotes após o roteamento.

c) Tabela *mangle*: usada na maioria das vezes para alterações especiais de pacotes. Suas *chains* são as seguintes:

INPUT: utilizado quando os pacotes precisam ser modificados antes de serem enviados para a *chain INPUT* da tabela *filter*.

FORWARD: utilizado quando os pacotes precisam ser modificados antes de serem enviados para a *chain FORWARD* da tabela *filter*.

PREROUTING: utilizado quando os pacotes precisam ser modificados antes de serem enviados para a *chain PREROUTING* da tabela *nat*.

POSTROUTING: utilizado quando os pacotes precisam ser modificados antes de serem enviados para a *chain POSTROUTING* da tabela *nat*.

OUTPUT: utilizado quando os pacotes precisam ser modificados antes de serem enviados para a *chain OUTPUT* da tabela *nat*.

Para a criação de regras é necessário utilizar uma conotação do tipo: iptables [opção] [*chain*] [dados] [ação]. As opções que podem ser utilizadas estão dispostas no Quadro 2, com sua respectiva funcionalidade.

-P	Define uma regra padrão
-A	Acrescenta uma nova regra as existentes. Este tem prioridade sobre a -P
-D	Apaga uma regra
-L	Lista as regras existentes
-F	Apaga todas as regras
-I	Insere uma nova regra
-h	Exibe a ajuda
-R	Substitui uma regra
-C	Faz uma checagem das regras existentes
-Z	Zera uma regra específica
-N	Cria uma nova regra com um nome
-X	Exclui uma rega específica pelo seu nome

Quadro 2 - Lista de opções de regras.
Fonte: Silva (2016).

O Quadro 3 exibe os dados que podem ser utilizados na criação das regras. Cada opção utilizada tem um objetivo e um significado.

-s	Especifica a origem do pacote.
-d	Especifica o destino do pacote.
-p	Protocolo a ser usado. (tcp, upd, icmp).
-i	Interface de entrada, usualmente indica-se a placa de rede
-o	Interface de saída, mesma condição que –i
!	Negação, exclui determinado argumento.
--sport	Refere-se à porta de origem e deve obrigatoriamente acompanhar as funções -p tcp e -p udp.
--dport	Refere-se à porta de destino e deve obrigatoriamente acompanhar as funções -p tcp e -p udp.
--syn	Especificar pacotes para iniciar uma conexão.

Quadro 3 - Lista de opções de dados.
Fonte: Silva (2016).

As ações sempre vêm acompanhadas do -j, que é chamado de alvo da regra. Ele serve para definir o destino do pacote, ou seja, se vai liberar ou bloquear entre outras opções que estão apresentadas no Quadro 4.

ACCEPT	Aceita e permite a passagem do pacote.
DROP	Não permite a passagem do pacote. Não informa se recebeu ou não o pacote.
REJECT	Assim como o DROP, não permite a passagem do pacote, mas envia um aviso.
LOG	Cria um *Log* referente a regra em /var/log/messages.

Quadro 4 - Lista de opções de ações.
Fonte: Silva (2016).

Portanto, para a elaboração de regras basta utilizar as opções apresentadas anteriormente e um pouco de lógica. Um exemplo simples que pode ser feito para melhor entendimento da criação de uma regra é o bloqueio de qualquer acesso a um determinado IP, assim, pode-se criá-la da seguinte maneira: iptables -t filter –A INPUT -d 192.168.2.2 -j DROP. Dessa maneira foi utilizada a tabela (-t filter), em seguida adicionado uma nova regra através dos dados que chegam (-A INPUT), especificado o destino do pacote (-d 192.168.2.2) e a ação a ser tomada, nesse caso efetuando o bloqueio (-j DROP).

3.1.5 SARG

A função do Sarg é interpretar os *logs* do Squid. A partir desta interpretação os acessos são organizados por usuários ou por IP, dependendo da configuração usada pelo gestor optando em usar ou não a autenticação fornecida pelo Squid.

A partir dessas informações o SARG organiza os *logs* gerando listas de páginas acessadas, com as seguintes informações: data, hora, local, máquina, quantidade de dados transmitidos, tempo utilizado em cada acesso e tentativas de acessos bloqueados pelos filtros.

Os relatórios antigos são organizados e armazenados automaticamente por certo tempo, sendo substituídos automaticamente pelo script que executará a rotatividade dos *logs*. Evitando assim a grande quantidade e o acúmulo de dados salvos no servidor.

3.2 MÉTODO

Este projeto apresenta uma alternativa de firewall para empresas de pequeno porte, através de um dispositivo ARM. Para a realização do trabalho foi levado em consideração a utilização de equipamentos e *softwares* com código aberto, pois a ideia é agregar qualidade e desempenho, com baixo custo.

O dispositivo ARM escolhido para desenvolver o projeto foi o Cubietruck. Nele foi instalado o sistema operacional Linux com distribuição Armbian. A versão do sistema operacional é direcionada exclusivamente para equipamentos ARM, ou seja, não é necessário efetuar nenhuma alteração nos pacotes ou ajuste na sua instalação (ARMBIAN, 2016).

Após a instalação do sistema operacional no Cubietruck foi configurado a rede para receber o sinal de Internet. Em seguida foi alterada as configurações de acesso remoto ao equipamento e alterada a configuração de fuso horário.

Com o sistema operacional configurado e a Internet sendo disponibilizada para a rede interna se faz necessário a instalação do Proxy, após a instalação foi configurado a porta de funcionamento, diretório em que o log é armazenado e

configuração da página de bloqueio. Em seguida foram implantados os filtros de conteúdo, esta etapa consiste no que estará bloqueado ou liberado para acesso. Logo após foi feito o cadastro de usuário e senha para os colaboradores da empresa efetuarem o *login* para liberação do acesso à Internet.

Visando proteger as informações que trafegam na rede foi implantado o *firewall*. O *firewall* foi configurado com política padrão de bloquear qualquer trafego que passe pelo equipamento, especificando através das suas regras as portas que são usadas para trafegar dados, a intenção é impedir os principais tipos de ataques que são oriundos da Internet.

Com a implantação do sistema para auditoria foi possível obter uma melhor leitura do *logs* gerados pelo Proxy. O sistema para auditoria foi configurado para gerar relatórios diários, com informações dos sites acessados, os horários de acessos, os usuários que efetuaram os acessos e em que máquina foi feito.

Após todos os sistemas em operação foram efetuados testes de funcionamento em todas as aplicações implantadas, com o intuito de coletar dados para análise e aperfeiçoamento.

4 RESULTADOS

Neste capítulo é apresentado o processo de instalação e configuração do sistema operacional, Proxy, Firewall e do sistema para auditorias no equipamento ARM. Além disso, são demonstrados os testes realizados nas aplicações e os resultados obtidos.

4.1 IMPLANTAÇÃO DO SISTEMA OPERACIONAL

A primeira utilização do equipamento ARM exigiu a utilização de um monitor com entrada HDMI e um teclado. Ao conectar o cabo HDMI e o teclado já é possível ligar a fonte do Cubietruck a tomada de energia, no entanto, antes disso é preciso efetuar a instalação do sistema operacional.

4.1.1 Instalação do sistema operacional

Para iniciar a instalação do sistema operacional é necessário efetuar o *download* da sua imagem, que pode ser obtida no site http://www.armbian.com/cubietruck/. Existem duas versões disponíveis, a versão *Legacy Wheezy* Server e a *Legacy Jessie* Server, sendo que a versão *Jessie* apresenta instabilidade quando transferida para a memória interna do equipamento. Portanto, a versão escolhida foi a *Legacy Wheezy* Server, pois é a versão mais instável até o momento. Ao término do *download* é necessário descompactar o arquivo baixado, para gerar os arquivos mostrados na Figura 5.

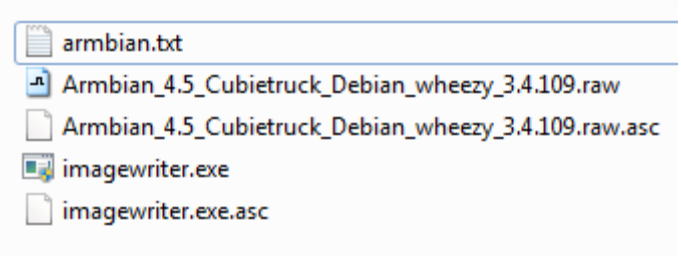

Figura 5 - Wheezy Server descompactado.
Fonte: Autoria própria.

A Figura 5 mostra cinco arquivos, sendo que foi usado apenas dois deles, a imagem do sistema operacional e um programa chamado SUSE Studio ImageWriter. Esse programa serve para realizar cópia da imagem do sistema operacional para o Micro SD. Porém antes de fazer uso do programa é necessário conectar o Micro SD ao computador e formatá-lo em qualquer sistema de arquivo, a única restrição ao Micro SD é que deve conter um tamanho mínimo de 4 GB.

Com o Micro SD formatado e o programa SUSE Studio Image Writer aberto, deve ser selecionado o sistema operacional através do botão *Select*, o programa direciona automaticamente até a imagem, sendo assim, basta selecionar e clicar no botão Abrir. O programa fica parecido ao apresentado na Figura 6.

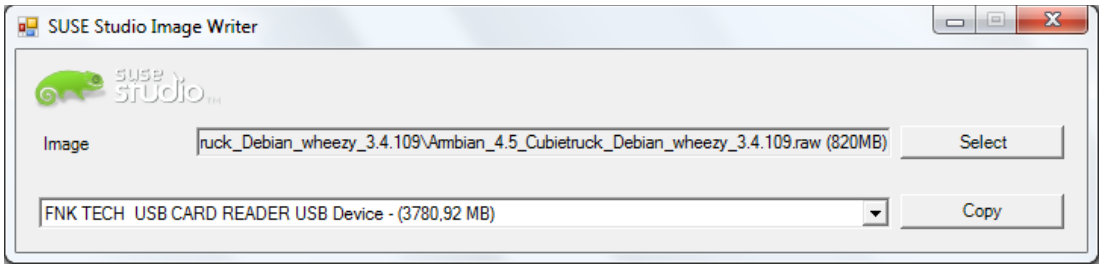

Figura 6 - Programa SUSE Studio Image Write.
Fonte: Autoria própria.

O Micro SD é localizado automaticamente pelo programa, caso haja mais de um Micro SD conectado ao computador, é possível selecionar o dispositivo desejado através da seta do lado esquerdo do botão Copy. Com tudo devidamente selecionado, clica-se no botão Copy. Uma mensagem de confirmação é exibida, informando que são substituídos os arquivos contidos no Micro SD. O processo de cópia tem início, até ser apresentado uma mensagem de sucesso, se tudo ocorreu bem.

Com os arquivos copiados para o Micro SD, deve-se conectá-lo ao Cubietruck. O equipamento ARM irá reconhecer o Micro SD e se inicia a instalação do sistema operacional. Esse processo funciona de forma automática e deve demorar em média 6 minutos.

Ao término da instalação é exibida uma tela solicitando o *login*, por padrão o usuário é root com a senha 1234. No primeiro acesso é solicitado a troca da senha para maior segurança, então, é solicitada novamente a senha de root, ou seja, a senha 1234. É preciso digitar a nova senha e efetuar as confirmações necessárias.

O sistema operacional foi instalado no Micro SD, portanto, é possível iniciar o sistema operacional apenas inserindo o Micro SD no Cubietruck. Mas como o

dispositivo ARM possui memória NAND foi transferido o sistema operacional do Micro SD para a NAND, com a intenção de aumentar a performance do sistema operacional e suas aplicações. Logo após efetuar o *login* é possível listar um arquivo chamado nand-sata-install. Esse *script* é responsável por transferir o sistema operacional para a NAND de forma automática, os comandos para realizar essa tarefa são os seguintes:

```
root@cubietruck:~# chmod a+x nand-sata-install
```

O comando chmod a+x torna o arquivo executável para qualquer usuário, ou seja, o arquivo tem permissão total para ser executado. Logo após ele pode ser executado com o comando:

```
root@cubietruck:~# ./nand-sata-install
```

Ao executar o *script* é apresentada uma tela azul com um cronômetro regressivo no seu rodapé informando o tempo para conclusão da transferência. Ao término da operação uma mensagem informa que o processo foi finalizado e que o dispositivo deve ser desligado e o Micro SD removido do equipamento. A partir de agora o sistema operacional é iniciado através da memória NAND do equipamento ARM, não é mais necessário ligá-lo com o Micro SD conectado.

4.1.2 Configuração do Armbian

A primeira configuração do servidor é a da placa de rede. Por padrão o sistema operacional define a configuração de rede para obter um endereço IP automaticamente, ou seja, se a rede fornecer um endereço IP de forma automática, é atribuído um endereço IP ao servidor, porém a cada nova conexão é atribuído um endereço IP diferente. Se fixado o endereço IP de forma estática, o endereço IP é sempre o mesmo a cada nova conexão, facilitando as configurações das aplicações que serão utilizadas, portanto no console do servidor é necessário entrar com o seguinte comando:

root@cubietruck:~# nano /etc/network/interfaces

Esse comando exibe as configurações de rede, possibilita a configuração de dois adaptadores via cabo e um sem fio, como é mostrada na Figura 7. No entanto, para neste trabalho foi usado apenas o adaptador via cabo.

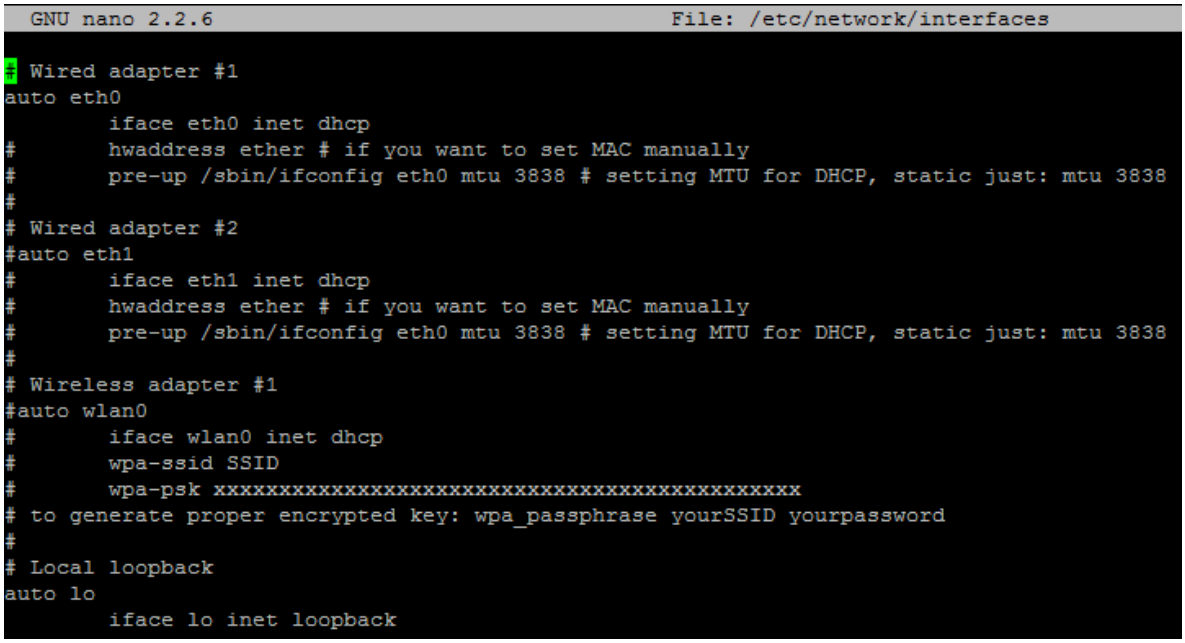

Figura 7 - Configuração padrão de rede.
Fonte: Autoria própria.

É possível notar na Figura 7 que a eth0 (*ethernet*) não está comentada com #, isso significa que ela está ativa e configurada para receber o endereço IP automático. Para funcionar em modo estático é necessário alterar as seguintes linhas:

auto eth0
iface eth0 inet dhcp

Para:

auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
gateway 192.168.1.1

Com as alterações realizadas, salvar as configurações utilizando o atalho Ctrl (*Control*) + O e em seguida Ctrl + X para sair do editor. Para aplicar as configurações é necessário reiniciar o serviço de rede com o comando:

root@cubietruck:~# /etc/init.d/networking restart

Se o sistema reportar um mensagem de OK, as configurações foram executadas com sucesso. Agora falta efetuar a configuração da DNS (*Domain Name System*) que é responsável por traduzir os domínios em endereços de IP, para isso é necessário editar o seguinte arquivo:

root@cubietruck:~# nano /etc/resolv.conf

Neste arquivo de configuração existe somente uma linha para alteração, que é alterada de:

nameserver 192.168.1.1

Para:

nameserver 8.8.8.8

O endereço de DNS configurado é um DNS público, fornecido pela empresa Google. Agora utilizando o atalho Ctrl (*Control*) + O para salvar a configuração e Ctrl + X para sair do editor. Novamente é necessário reiniciar o serviço de rede com o comando:

root@cubietruck:~# /etc/init.d/networking restart

Se o sistema retornar uma mensagem positiva as configurações foram inseridas com sucesso. Neste momento, é possível efetuar o teste de conexão através do comando *ping*, como é mostra a Figura 8.

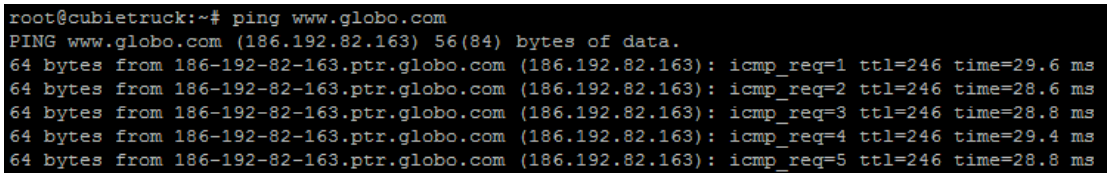

Figura 8 - Resposta do ping.
Fonte: Autoria própria.

A resposta ao teste de *ping* foi positiva, como apresentado na Figura 8, portanto a conexão do servidor com a Internet está em pleno funcionamento.

Com a Internet em funcionamento é necessário alterar as configurações de fuso horário, pois o sistema operacional instalado, por padrão, está configurado com o fuso horário europeu. Para dar início a alteração utiliza-se o seguinte comando:

```
root@cubietruck:~# dpkg-reconfigure tzdata
```

A tela exibida solicita que seja escolhido o continente do novo fuso horário, portanto selecione a opção América. Agora é solicitada a opção de fuso horário. Como o sistema operacional não fornece a opção da capital Brasília e apenas São Paulo, ao escolhe-la as configurações já atualizadas, serão mostradas na tela.

A versão de sistema operacional instalada traz o sistema de SSH (*Secure Shell*) configurada para funcionar na porta 22. O sistema de SSH permite a conexão entre computadores, isto é, é possível acessar o console do servidor remotamente através de outro computador independente do sistema operacional, permitindo executar comandos como se estivesse localmente. Por estar configurado com uma porta padrão, a 22, torna-se necessário efetuar a troca dessa porta. Esta prática agrega mais segurança ao servidor, pois somente o administrador do servidor terá conhecimento da porta que está aberta para uso. No terminal do servidor é inserido o seguinte comando:

```
root@cubietruck:~# nano /etc/ssh/sshd_config
```

Para alteração da porta existe uma linha não comentada que indica Port 22. Está linha deve ser alterada para Port 2202, por exemplo, e em seguida salvando as configurações através do atalho Ctrl (*Control*) + O e usando Ctrl + X para sair do editor, agora é necessário reiniciar o serviço através do comando:

```
root@cubietruck:~# /etc/init.d/ssh restart
```

O sistema deve retornar uma mensagem de OK informando que as configurações foram salvas e portanto a partir de agora qualquer conexão SSH realizada no servidor deve ser feita na porta 2202.

Para finalizar as configurações do sistema operacional, é realizada uma atualização de todo o sistema, isso serve para manter o sistema seguro e para correção de falhas no sistema, além de atualizar as principais aplicações do sistema. No terminal do servidor deve ser aplicado o seguinte comando:

```
root@cubietruck:~# apt-get update && apt-get upgrade
```

O sistema operacional busca por atualizações em diversos servidores e ao final é apresentado os pacotes que necessitam de atualização juntamente com a opção de prosseguir com o processo através da tecla “y” ou cancelá-lo pressionado a tecla “c”. Com o processo em continuidade o servidor começa a baixar e instalar os pacotes, tudo de maneira automática, podendo demorar alguns minutos dependendo da velocidade da Internet. Ao final do processo o sistema retornará ao terminal inicial.

4.2 INSTALAÇÃO E CONFIGURAÇÃO DO PROXY

Antes de iniciar a instalação do Squid é necessário habilitar o modulo de encaminhamento de pacotes IPv4 (*Internet Protocol version* 4). Para isso é necessário editar o seguinte arquivo:

```
root@cubietruck:~# nano /etc/sysctl.conf
```

Agora basta localizar a linha exibida abaixo:

#net.ipv4.ip_forward=1

E alterá-la para:

net.ipv4.ip_forward=1

Com essa alteração já é possível iniciar a instalação do Squid. Para isso basta digitar o seguinte comando:

root@cubietruck:~# apt-get install squid3

O sistema operacional irá solicitar se deseja continuar, para isso basta pressionar a tecla “y” e aguardar a instalação do programa. Após isso, é preciso instalar um complemento, necessário para gerenciar os usuários que forem criados para autenticar no Squid, para instalar basta digitar no console:

root@cubietruck:~# apt-get install apache2-utils

Com tudo instalado é preciso navegar até o diretório em que o Squid está instalado, o comando para isso é:

root@cubietruck:~# cd /etc/squid3/

Dentro desse diretório está o arquivo de configuração do Squid, no entanto ele precisa ficar inalterado, pois nele está contido toda a documentação e exemplos. Este arquivo é útil caso se necessite efetuar alguma pesquisa de utilização do Squid. Essa ação é feita com o comando:

root@cubietruck:/etc/squid3# mv squid.conf squid.conf.original

O comando gerou um arquivo chamado squid.conf.original com o objetivo de deixá-lo apenas para consultas e um novo arquivo chamado squid.conf deve ser criado. Para iniciar a criação do arquivo deve-se executar o seguinte comando:

root@cubietruck:/etc/squid3# nano squid.conf

Dentro do arquivo em branco, serão adicionadas as linhas de comando de configuração do Squid de maneira organizada, visando facilitar a manutenção do Proxy futuramente.

A primeira linha é a porta padrão pela qual o squid vai atuar. Essa porta não precisa ser necessariamente a 3128, ela pode ser alterada para qualquer outra porta que melhor se encaixe na rede.

http_port 3128

Agora a linha que vai exibir o nome do servidor proxy.

visible_hostname cubietruck

Está linha indica o diretório que exibe a página de bloqueio em português brasileiro. Quando o usuário acessar uma página não autorizada ou não efetuar o *login* para acessar a Internet, é exibida uma mensagem de erro.

error_directory /usr/share/squid3/errors/pt-br

Juntamente a mensagem de erro é mostrado o e-mail do responsável pelo Proxy, com a intenção de fornecer algum suporte para os usuários.

cache_mgr giovani123@gmail.com

Nesta linha é possível escolher o diretório onde se quer salvar os registros de conexões que passam pelo Squid.

cache_access_log /var/log/squid3/access.log

Agora é criada uma ACL (*Access Control List*) ou lista de controle de acesso chamada rede_local. A opção src significa origem, isto é, com ela é possível restringir o Proxy apenas para os usuários que estejam dentro da mesma classe de IP e usando a mesma máscara de sub-rede do servidor.

acl rede_local src 192.168.1.0/24

As ACLs criadas com os nomes de SSL_ports e Safe_ports agregam as principais portas usadas para navegação, essas ACLs estão no arquivo padrão de configuração do Squid.

```
acl SSL_ports port 443 # https
acl SSL_ports port 563 # snews
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl Safe_ports port 901 # SWAT
```

A ACL CONNECT tem como elemento o method, que é o método http de requisição (get ou post), então foi acrescentado:

```
acl CONNECT method CONNECT
```

A sintaxe *deny* significa bloqueio ou negado no Squid, portanto está regra bloqueia qualquer acesso ou conexão que não esteja especificado nas ACLs Safe_ports e SSL_ports.

```
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
```

A ACL palavras_bloqueadas busca por palavras chaves dentro do arquivo palavras_bloquedas, que é criado dentro da pasta squid3. A sintaxe url_regex compara as palavras contidas no arquivo palavras_bloquedas com a URL (*Uniform Resource Locator*) em que o usuário está navegando. A opção -i é utilizada para que não tenha distinção entre maiúsculas e minúsculas.

```
acl palavras_bloqueadas url_regex -i "/etc/squid3/palavras_bloqueadas"
```

A ACL sites_bloqueados busca por sites ou domínios contidos no arquivo sites_bloqueados, que é criado dentro da pasta squid3. A sintaxe dstdomain compara os domínios contidos no arquivo sites_bloqueados com a URL em que o usuário está tentando acessar.

acl sites_bloqueados dstdomain -i "/etc/squid3/sites_bloqueados"

A ACL ips_liberados busca os IPs contidos no arquivo ips_liberados, que é criado dentro da pasta squid3. Está regra tem o intuito de que o IP contido nessa lista tenha acesso livre, ou seja, não é afetado por qualquer bloqueio, poderá acessar o site que desejar sem nenhuma restrição.

acl ips_liberados src "/etc/squid3/ips_liberados"

A ACL redes_sociais busca por sites ou domínios contidos no arquivo redes_sociais, que é criado dentro da pasta squid3. Está regra é usada para bloquear as redes sociais e liberá-las em determinado horário.

acl redes_sociais dstdomain -i "/etc/squid3/redes_sociais"

A ACL streaming busca por tipos de arquivos descritos no arquivo streaming, que é criado dentro da pasta squid3. A sintaxe rep_mime_type corresponde ao tipo de arquivo recebido pelo squid. Com essa regra é possível bloquear o recebimento de qualquer formato de arquivo, nesse caso é usado apenas para arquivos de áudio e vídeo.

acl streaming rep_mime_type -i "/etc/squid3/streaming"

A regra abaixo cria a ACL horário_almoco, que define o tempo de intervalo do almoço da empresa.

acl horario_almoco time 12:00-13:30

A sintaxe http_access tem o objetivo de negar ou permitir qualquer ACL criada. A regra descrita abaixo tem a função de bloquear o conteúdo contido na ACL palavras_bloqueadas.

http_access deny palavras_bloqueadas

Bloqueia o conteúdo contido na ACL sites_bloqueados.

http_access deny sites_bloqueados

Libera o acesso aos IPs contidos na ACL ips_liberados.

http_access allow ips_liberados

Libera o acesso ao conteúdo contido na ACL redes_sociais apenas no horário mencionado na ACL horario_almoco.

http_access allow redes_sociais horario_almoco

Bloqueia o conteúdo contido na ACL redes_sociais.

http_access deny redes_sociais

A sintaxe http_reply_access é um complemento à sintaxe http_access, porém só deve ser usada para permitir ou bloquear requisições do tipo rep_mime_type. Por se tratar de uma exceção é preciso liberar a ACL streaming para a ACL ips_liberados.

http_reply_access allow streaming ips_liberados

Bloqueia o conteúdo contido na ACL streaming.

http_reply_access deny streaming

A opção auth_param é a opção para definir o parâmetro de autenticação de usuários. Nesta regra é usado o tipo NCSA (*National Center for Supercomputing Applications*) de autenticação, padrão do Proxy fornecido pelo próprio sistema operacional, ele irá criptografar os dados do usuário no arquivo passwd dentro da pasta squid3.

auth_param basic program /usr/lib/squid3/ncsa_auth /etc/squid3/passwd

A sintaxe proxy_auth checa as combinações de *login* e senha existentes. A opção REQUIRED faz com que seja exigida a autenticação para que seja criada a ACL autenticados.

acl autenticados proxy_auth REQUIRED

Libera o acesso aos usuários contidos na ACL autenticados.
http_access allow autenticados

Libera acesso a ACL rede_local
http_access allow rede_local
Está regra bloqueia tudo que não se encaixe em nenhuma das ACLs.
http_access deny all

Com todas as regras criadas, o próximo passo é salvar o arquivo usando Ctrl + O e Ctrl + X para sair. Após voltar ao terminal de comando é preciso criar os arquivos que foram mencionados nas regras criadas, o comando para a criação dos arquivos é a seguinte:

root@cubietruck:/etc/squid3# touch palavras_bloqueadas

Com o arquivo criado, é necessário editá-lo para inserir as palavras que se julgue necessário bloquear. No console insira o seguinte comando:

root@cubietruck:/etc/squid3# nano palavras_bloqueadas

Foram utilizadas três palavras chaves como mostra a Figura 9. As palavras devem ser inseridas uma abaixo da outra, não existe a necessidade de separá-las por vírgula ou qualquer outro tipo de caractere, além disso, também é possível utilizar expressões regulares, após a inserção basta salvar o arquivo e depois sair.

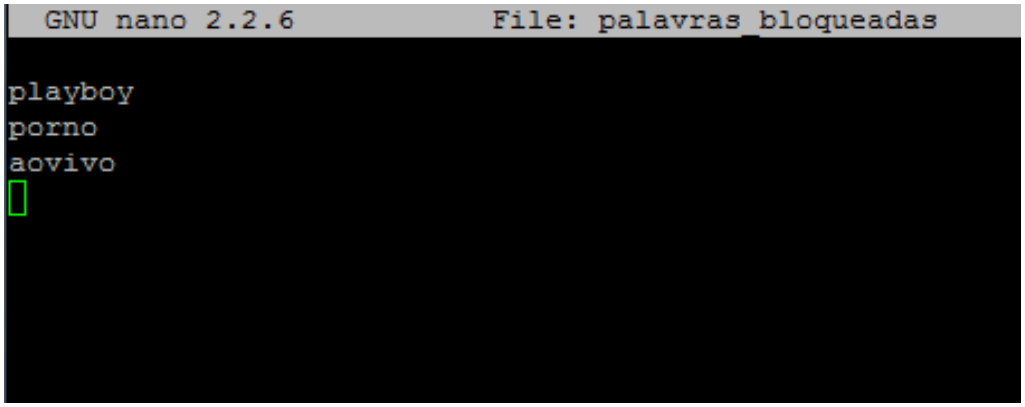

Figura 9 - Conteúdo do arquivo palavras_bloqueadas.
Fonte: Autoria própria.

Outro arquivo que é necessário criar é o de sites bloqueados, então no console digite:

root@cubietruck:/etc/squid3# touch sites_bloqueados

O arquivo recém criado deve ser editado e os sites que serão bloqueados, devem ser adicionados. No console digite:

root@cubietruck:/etc/squid3# nano sites_bloqueados

No arquivo sites_bloqueados foram inseridos três sites como mostra a Figura 10. Os sites devem ser listados um abaixo do outro e não existe necessidade de adicionar o prefixo WWW (*World Wid Web*) antes dos sites, ao final da edição do arquivo é só salvar e sair.

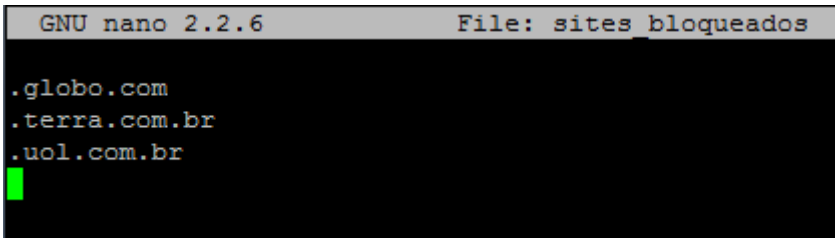

Figura 10 - Conteúdo do arquivo sites_bloqueados.
Fonte: Autoria própria.

O próximo arquivo a ser criado é o de IPs liberados, portanto no console:

root@cubietruck:/etc/squid3# touch ips_liberados

O próximo passo é editar o arquivo recém criado.

root@cubietruck:/etc/squid3# nano ips_liberados

Assim como nos arquivos anteriores os IPs também devem ser inseridos um abaixo do outro como é mostrado na Figura 11. Lembrando que os IPs inseridos no arquivo ips_liberados não sofrem nenhum tipo de bloqueio na navegação, ao final da edição, salvar o arquivo e sair.

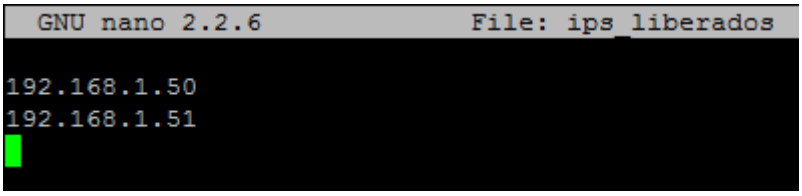

Figura 11 - Conteúdo do arquivo ips_liberados.
Fonte: Autoria própria.

Agora o arquivo que deve ser criado é o das redes sociais.

root@cubietruck:/etc/squid3# touch redes_sociais

Com o arquivo redes_sociais criado, o próximo passo é editá-lo.

root@cubietruck:/etc/squid3# nano redes_sociais

O arquivo redes_sociais deve ser editado como foi feito no arquivo sites_bloqueados, porém com os sites que deverão ser bloqueados em horário de expediente e liberados no horário de almoço. Foram inseridos apenas três sites:

.facebook.com
.twitter.com
.youtube.com

Ao final da edição basta salvar o arquivo e sair para dar sequência a configuração.

O último arquivo a ser criado é o de streaming.

root@cubietruck:/etc/squid3# touch streaming

Editando o arquivo streaming.

root@cubietruck:/etc/squid3# nano streaming

Dentro do arquivo streaming devem ser inseridos os formatos de arquivos que compõem o streaming de áudio e vídeo, os formatos foram listados abaixo:

mms
x-ms-asf
video/flv
video/mp4
video/x-flv
application/x-shockwave-flash

Os arquivos mms (Microsoft Media Server) são transmitidos via streaming através do Windows Media Play. O x-ms-asf são arquivos de áudio do Windows Media Player, esse formato é usado em rádios com transmissão streaming. O formato video/flv e video/x-flv são arquivos de vídeo em Flash. O application/x-shockwave-flash é uma aplicação de animações em Flash. Existem muitos outros formatos de arquivos como, por exemplo, o mp4 que é outro formato de vídeo, os listados foram apenas para exemplificar o funcionamento. Ao final da inserção dos formatos é necessário salvar as alterações e depois sair.

Os arquivos de configuração de Squid estão prontos, mas antes de reiniciar o serviço do Squid para aplicar as configurações é preciso criar os usuários para autenticação no Proxy, o comando para isso é:

```
root@cubietruck:/etc/squid3# htpasswd -c /etc/squid3/passwd giovani
```

A sintaxe htpasswd é um complemento do Apache usado para autenticação e o -c é usado para autorizar a edição do arquivo passwd para criação do usuário giovani, no entanto a sintaxe -c deve ser usada apenas na criação do primeiro usuário, para os demais não é exigido. Após executar o comando, é solicitado uma senha para o usuário. Se tudo foi digitado corretamente uma mensagem é exibida de que foi adicionada uma senha ao usuário criado (*Adding password for user* giovani).

O último passo é reiniciar o serviço do Squid para que as configurações feitas sejam aplicadas, o comando para realizar essa tarefa é:

```
root@cubietruck:/etc/squid3# /etc/init.d/squid3 restart
```

Se ao final for reportado uma mensagem de OK é porque as configurações foram aplicadas sem erros. Com o serviço em operação é preciso fazer a configuração do computador do usuário. A primeira alteração é as configurações de IP e depois a configuração do Proxy no navegador de Internet.

A Figura 12 ilustra a configuração de rede necessária para o funcionamento do computador do usuário através do Proxy.

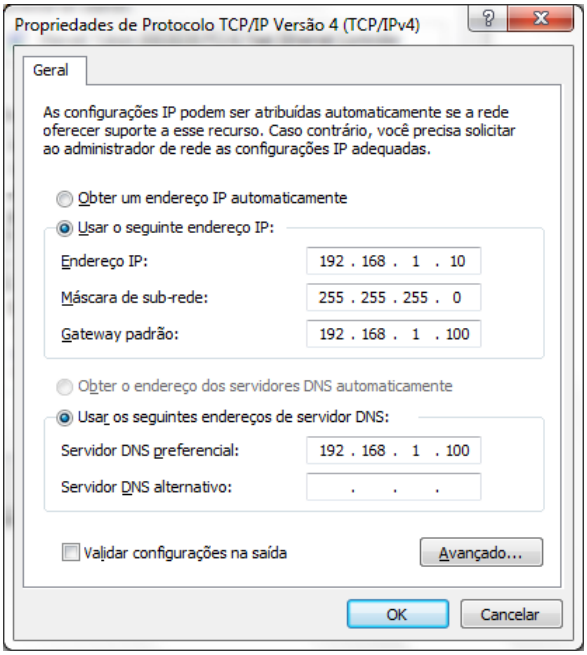

Figura 12 - Configuração de IP do usuário.
Fonte: Autoria própria.

O endereço IP deve ser único em cada computador da rede, a máscara de sub-rede é dada automaticamente, o gateway padrão e servidor devem ser direcionados para endereço IP do Proxy.

Outra configuração necessária é a do endereço de IP do Proxy no sistema operacional como ilustra a Figura 13. Dessa maneira todos os programas que fazem conexão com a Internet precisam ser configurados para funcionar através do Proxy.

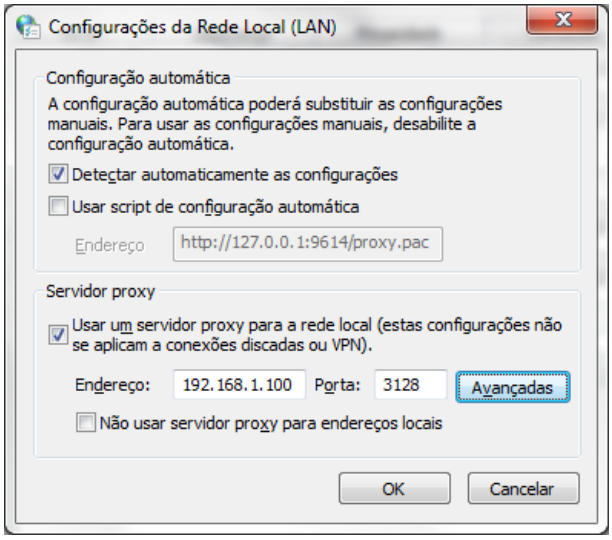

Figura 13 - Configuração do Proxy.
Fonte: Autoria própria.

No sistema operacional Windows é possível localizar as configurações de Proxy dentro do painel de controle no ícone chamado opções da internet, em seguida na aba conexões e depois na opção configurações de LAN (Local Area Network). Marcando a caixa *usar um servidor proxy para a rede local,* irá habilitar a inserção do endereço do Proxy com sua respectiva porta. No botão descrito como avançadas é possível acessar as configurações avançadas de Proxy para diversos protocolos de conexão como é exibido na Figura 14.

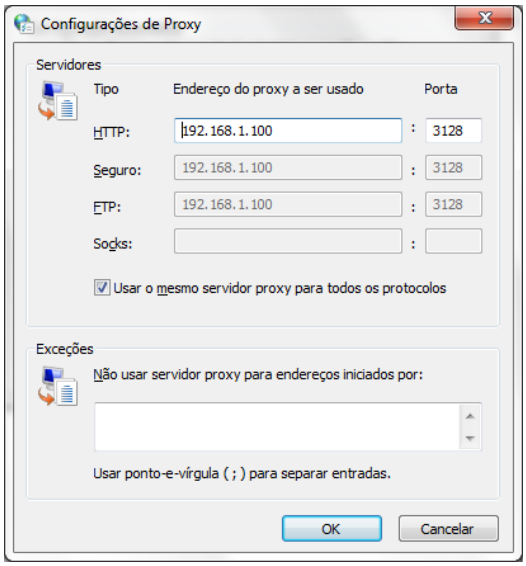

Figura 14 - Configuração avançada do Proxy.
Fonte: Autoria própria.

Para um melhor aproveitamento do Proxy é recomendado que a caixa descrita como: *usar o mesmo servidor proxy para todos os protocolos* seja marcada, pois dessa maneira ao usar qualquer protocolo citado na Figura 14 é solicitado usuário e senha para seu funcionamento.

Ao tentar acessar qualquer site usando o navegador padrão do Windows é solicitado à inserção de usuário e senha para a liberação do acesso, a Figura 15 mostra a mensagem de autenticação solicitada.

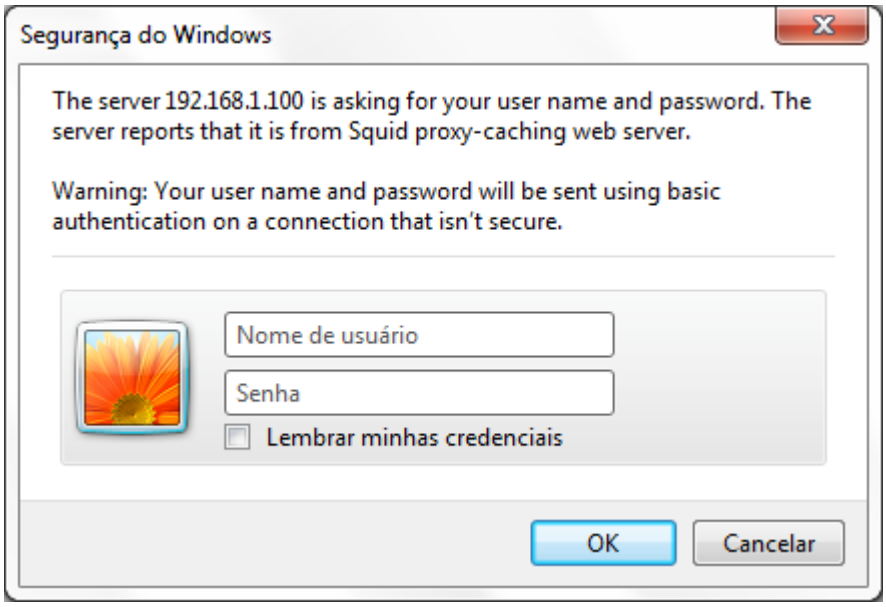

Figura 15 - Solicitação de usuário e senha.
Fonte: Autoria própria.

Após a autenticação ser bem sucedida o usuário terá acesso livre a Internet dentro das regras impostas anteriormente nas configurações do Squid. Caso a tentativa de acesso do usuário seja barrada por umas das regras, é exibida uma página de erro, mostrada na Figura 16.

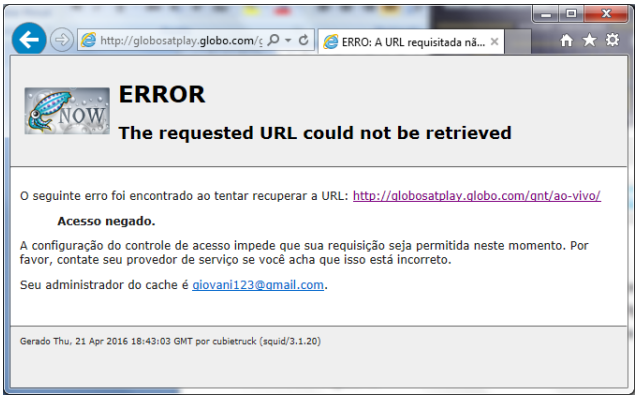

Figura 16 - Mensagem de bloqueio.
Fonte: Autoria própria.

A Figura 16 exemplifica uma das regras aplicadas, a de palavras bloqueadas. Na tentativa do usuário acessar um site contendo a palavra ao-vivo imediatamente o Squid fez o bloqueio e apresentou a página de bloqueio, no conteúdo da página contém o site que foi tentado acessar, seguido da mensagem de acesso negado e o contato do administrador do Proxy para qualquer esclarecimento.

4.3 INSTALAÇÃO E CONFIGURAÇÃO DO FIREWALL

A primeira etapa para criação do firewall é a instalação do Iptables. Está aplicação é responsável por interpretar as regras codificadas no *firewall*. Para iniciar a instalação o comando abaixo dever ser inserido.

```
root@cubietruck:~# apt-get install iptables
```

Com o Iptables instalado é preciso navegar através do console até a pasta onde é criado o arquivo de firewall, no console utilizar o seguinte comando:

```
root@cubietruck:~# cd /etc/init.d/
```

A pasta init.d contém os arquivos de inicialização do sistema operacional, o intuito do firewall é de que ele inicie juntamente com o sistema operacional. Para codificar o firewall é necessário criar um arquivo chamado firewall.sh, este arquivo irá conter todas as regras necessárias para conter as ameaças mais recorrentes oriundas da Internet. Para concretizar o que foi proposto, no console é inserido o seguinte:

```
root@cubietruck:/etc/init.d# nano firewall.sh
```

A extensão *sh* do *firewall* quer dizer que ele é um arquivo Shell Script, ou seja, um arquivo executável, após ter a devida permissão. Dentro do arquivo firewall.sh as primeiras linhas são:

```
modprobe ip_tables
modprobe ipt_multiport
modprobe ipt_LOG
```

O modprobe é usado para adicionar os módulos que são utilizados no firewall, nas linhas acima estão sendo adicionados os módulos de tabelas, de múltiplas portas e o de registro de *logs*.

O objetivo dessas regras são limpar qualquer outra regra que já esteja em operação, para que em seguida sejam carregadas outras regras e não ocasione qualquer tipo de conflito entre as existentes e as novas.

```
iptables -F
iptables -X
iptables -Z
```

A primeira medida de segurança a ser tomada pelo firewall é de bloquear qualquer conexão que tentar entrar ou sair. Dessa maneira é possível definir apenas as conexões que serão utilizadas para o funcionamento do firewall e suas aplicações, impedido qualquer tipo de ataque a portas que estejam abertas, mas sem uso.

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

A sintaxe -m especifica um módulo do iptables. O modulo usado é o *state* e sua função é efetuar a filtragem de pacotes, ou seja, liberar a volta dos pacotes às conexões estabelecidas e relacionadas. Sendo assim pacotes inválidos ou tentativas de conexões são descartados antes mesmo de ser tratado por qualquer outra regra.

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Esta opção de roteamento foi habilitada manualmente dentro do *kernel* anteriormente, porém foi adicionada está regra apenas por precaução, caso seja

desabilitada sem intenção, o *firewall* irá habilitá-la automaticamente ao ser executado.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

O *kernel* do sistema operacional traz uma proteção contra ataques de SynFlood, no entanto está opção vem desabilitada por padrão, sendo necessário efetuar sua ativação manualmente dentro do arquivo /etc/sysctl.conf ou através da regra abaixo.

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

A interface loopback permite testes dos protocolos e das interfaces de rede, além de permitir a intercomunicação entre processos dentro do sistema operacional.

```
iptables -A INPUT -i lo -j ACCEPT
```

A próxima regra efetua o bloqueio dos ataques de DDoS ou Smurf Attacks baseados no protocolo ICMP. O *type* identifica o tipo de mensagem, neste caso ICMP e o *echo-request* indica uma requisição com valor de 8 *bits*. Impedindo que o servidor seja sobrecarregado com mais requisições do que pode tratar. Em seguida efetua o registro da execução no arquivo /var/log/messages.

```
iptables -A INPUT -p icmp --icmp-type echo-request -j LOG --log-prefix "FIREWALL: DDoS: "
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

Este servidor firewall deve ter a opção de acesso via SSH. Como todas as portas foram bloqueadas, é preciso liberar a porta 2202 que foi configurada para o acesso, porém com a técnica de força bruta é obter acesso ao servidor. As regras abaixo foram criadas para evitar os acessos indevidos. A primeira regra é apenas para a geração do registro de acesso, como foi feito anteriormente. A segunda regra corresponde à entrada de uma conexão tcp na porta 2202. O módulo utilizado para essa regra é o *recent*, ele é composto pelo *update*, *hitcount*, *seconds* e *name*. O *update* faz a atualização do *hitcount* das tentativas de conexão, se o *hitcount* atingir duas conexões mal sucedidas na lista NOVOSSH é bloqueado por 60 segundos.

```
iptables -A INPUT -p tcp --dport 2202 -m recent --update --hitcount 2 --seconds 60 --name NOVOSSH -j LOG --log-prefix "FIREWALL: brute force: "
```

iptables -A INPUT -p tcp --dport 2202 -m recent --update --hitcount 2 --seconds 60 --name NOVOSSH -j DROP

A regra abaixo libera o acesso a porta 2202. Ela é a extensão das regras citadas anteriormente. Através do módulo *recent* é atribuído o nome do computador que está tentando acessar o servidor na lista NOVOSSH, assim, as regras entram em funcionamento verificando a lista NOVOSSH. Em uma visão geral, esse conjunto de regras verifica a lista NOVOSSH, se ela não contém o nome do computador que está realizando a conexões ao servidor, é liberada a porta para acesso, caso está lista contenha duas vezes o nome do computador que está tentando acessar o servidor, a porta é bloqueada para esse computador.

iptables -A INPUT -p tcp --dport 2202 -m recent --set --name NOVOSSH -j ACCEPT

O conjunto de regras denominada SCANNER efetuam o bloqueio contra scanners de portas. Primeiramente é criada uma regra com o nome de SCANNER, em seguida é efetuado o registro da sua utilização para depois bloqueá-la. A sintaxe --tcp-flags utilizar *flags* para filtrar pacotes TCPs específicos. O argumento ACK (*Acknowledgement*) implica em confirmar o recebimento dos dados enviados pela origem. O argumento FIN (*Finish*) indica que a conexão atual quer ser interrompida por uma das partes por não ter mais dados a serem enviados. No argumento PSH (*Push*), o computador que recebe os dados é instruído a enviá-los para a aplicação imediatamente. O argumento RST (*Reset*), reinicia ou rejeita alguma tentativa de conexão problemática. O argumento SYN (*Synchronize*) indica um computador deseja transmitir dados e estabelecer uma conexão. O argumento URG (*Urgent*) indica uma característica de urgência, ou seja, os segmentos que forem recebidos primeiro, deverão ser tratados por primeiro (KARPISCHEK, 2016). Segundo Cabral (2016) a regra abaixo permite a proteção eminente contra os scanners de portas:

```
iptables -N SCANNER
iptables -A SCANNER -j LOG --log-prefix "FIREWALL: port scanner: "
iptables -A SCANNER -j DROP
```

As regras a seguir, indicam que todas as *flags* devem ser examinadas (ALL é sinônimo de SYN, ACK, FIN, RST, URG, PSH), porém apenas FIN, URG, PSH devem estar em funcionamento. O argumento NONE significa nenhuma *flag*.

iptables -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j SCANNER

Pacotes nulos ou mal formados.

iptables -A INPUT -p tcp --tcp-flags ALL NONE -j SCANNER

iptables -A INPUT -p tcp --tcp-flags ALL ALL -j SCANNER

iptables -A INPUT -p tcp --tcp-flags ALL FIN,SYN -j SCANNER

iptables -A INPUT -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j SCANNER

iptables -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j SCANNER

iptables -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j SCANNER

Está regra libera a porta 3128 para a entrada. Esta foi à porta configurada para o funcionamento do Squid.

iptables -A INPUT -p tcp --dport 3128 -j ACCEPT

A regra abaixo utiliza o módulo *multiport* que possibilita utilizar mais de uma porta na mesma regra. Esta regra libera as portas de saída 80 e 443 *http* e *https*.

iptables -A OUTPUT -p tcp -m multiport --dports 80,443 -j ACCEPT

Esta regra libera as portas utilizadas para acesso a FTP, tanto para protocolos tcp quanto para udp.

iptables -A OUTPUT -p tcp -m multiport --dports 20,21 -j ACCEPT

iptables -A OUTPUT -p udp -m multiport --dports 20,21 -j ACCEPT

As próximas regras liberam as portas de DNS para saída, tanto para tcp quanto para upd.

iptables -A OUTPUT -p tcp --dport 53 -j ACCEPT

iptables -A OUTPUT -p udp --dport 53 -j ACCEPT

Esta regra é única e exclusiva para atualizar a hora do sistema de forma automática.

iptables -A OUTPUT -p udp --dport 123 -j ACCEPT

Com todas as regras codificadas o próximo passo é salvar o arquivo. Ao retornar ao terminal é preciso atribuir a permissão de execução para o arquivo firewall.sh, para que quando o sistema operacional se inicie ele seja executado e entre em funcionamento. Para executar essa ação é preciso digitar o seguinte comando:

```
root@cubietruck:/etc/init.d# chmod +x firewall.sh
```

O comando chmod é usado para atribuir ou alterar permissões de arquivos, enquanto a sintaxe +x significa que o arquivo terá permissão de execução. Para colocar o *firewall* em funcionamento existem duas maneiras, a primeira reiniciado o servidor e a segunda usando o comando de execução.

```
root@cubietruck:/etc/init.d# ./firewall.sh
```

É possível atribuir mensagens para cada regra criada dentro do arquivo firewall.sh, assim quando o *firewall* for executado irá ser exibida na tela uma mensagem informando que aquela regra foi executada com sucesso. Essa mensagem pode ser inserida como no exemplo abaixo.

```
iptables -A INPUT -p icmp --icmp-type echo-request -j LOG --log-prefix "FIREWALL: DDoS: "
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
echo "Politica Anti Ataque DDoS .....[ OK ]"
```

Quanto mais mensagens forem inseridas mais simples é para localizar algum possível erro de digitação ou criação das regras, pois se pode prever qual regra foi executada e qual não foi.

4.4 INSTALAÇÃO E CONFIGURAÇÃO DO SARG

Para a implantação do sistema para auditorias é preciso efetuar a instalação de uma aplicação chamada Apache na versão 2.2.22, que permite o acesso ao sistema através do navegador de Internet. O método para instalação dessa aplicação é semelhante as realizadas anteriormente, portanto, no terminal é inserido o seguinte comando:

```
root@cubietruck:~# apt-get install apache2
```

É solicitado uma confirmação se deseja prosseguir com a instalação, usando a tecla “Y” para dar continuidade é efetuada a instalação e em seguida o serviço é iniciado automaticamente, estando pronto para hospedar o Sarg. Agora é a vez do Sarg ser instalado na versão 2.3.2, o comando para isso é o seguinte:

```
root@cubietruck:~# apt-get install sarg
```

Novamente é solicitado a confirmação para realizar a instalação, tecla “Y” para prosseguir e aguardar a conclusão automática. Com todos as aplicações necessárias instaladas é necessário efetuar a configuração do Sarg para a geração dos relatórios. O arquivo de configuração do Sarg que precisa ser editado está localizado dentro do seguinte diretório:

```
root@cubietruck:~# cd /etc/sarg/
```

O arquivo que é editado se chama sarg.conf, nele existem vários exemplos de configuração ou possibilidades diferentes de edição. Neste projeto serão descritas apenas as linhas que irão ser editadas, pois cada linha já possui uma descrição de seu funcionamento. Para dar início a configuração deve ser aberto o arquivo através do seguinte comando:

```
root@cubietruck:/etc/sarg# nano sarg.conf
```

Dentro do arquivo sarg.conf é possível observar que existe uma linha que inicia com a palavra TAG, nela é informado o propósito do comando que pode ser inserido e uma breve explicação sobre o mesmo. A primeira linha alterada é indicada com a TAG: access_log file, nesta linha é solicitado informar onde está localizado o arquivo que armazena os registro de eventos do Squid, ao final desta deve ser acrescentado a seguinte linha:

access_log /var/log/squid3/access.log

É importante observar que não existe a necessidade da utilização do caractere #, pois ao utilizá-lo a linha é comentada e não é integrada a configuração da aplicação.

A próxima TAG alterada é a chamada de *title*, ela é composta do título que é exibido quando a aplicação é aberta no navegador de Internet, por mais que se trate de uma alteração estética é personalizado com a função do Sarg, a linha a ser adicionada é:

title "Auditoria do Cubietruck Proxy"

Outra TAG que foi alterada é chamada de output_dir, está TAG é responsável por indicar onde os relatórios serão salvos. Esses relatórios são gerados automaticamente analisando os registro de evento do Squid e em determinado horário o Sarg gera um relatório que permite a visualização através do navegador de Internet. O diretório que é usando é indicado pela linha abaixo.

output_dir /var/www/sarg

A próxima TAG a ser modificada é a TAG user_ip, nela existem apenas duas opções de sim ou não. A opção sim é usada para mostrar no relatório o endereço IP do computador que fez o acesso e não é a opção padrão que busca pelo nome do computador. Aqui é utilizada a opção sim, pois ao decorrer do projeto é atribuído um nome específico para cada endereço IP. Abaixo é demostrado a linha como deve ser configurada.

user_ip yes

A TAG lastlog é a próxima a ser alterada, ela server para remover os registro de eventos antigos, dessa maneira evita que o espaço de armazenamento do Firewall fique cheio. A configuração que é aplicada consistem em manter somente os últimos 30 registros, de cada usuário armazenados, para isso a linha de configuração deve ficar dessa maneira:

lastlog 30

Dentro do arquivo sarg.conf são essas as alterações que precisam ser alteradas, a próxima etapa é salvar as configurações usando Ctrl + O e Ctrl + X para sair. Ao retornar ao diretório do Sarg é possível identificar diversos outros arquivos de configurações como é mostrado na Figura 17.

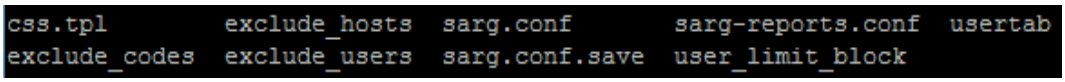

Figura 17 - Conteúdo do diretório Sarg.
Fonte: Autoria própria.

Os arquivos de configuração possibilitam a adição de usuários e nome do computador para que não apareçam nos relatórios de auditoria, esses arquivos são exclude_users e exclude_hosts respectivamente. Não foi realizada nenhuma inclusão em nenhum desses arquivos, pois o intuito de uma auditoria é de que tudo seja analisado. Outro arquivo que é editado é o chamado usertab, nele é possível atrelar um nome ao endereço IP, dessa maneira no relatório irá mostrar o nome atribuído em vez do endereço IP. No console é preciso inserir o seguinte comando:

root@cubietruck:/etc/sarg# nano usertab

Neste arquivo precisa ser inserido as informações da seguinte maneira:

192.168.1.15 GIOVANI
192.168.1.16 PROFESSOR

Em seguida Ctrl + O para salvar e Ctrl + X para sair. A próxima configuração a ser realizada é a geração de relatórios diários de forma automática, para isso é criado um script de execução responsável por filtrar e gerar os relatórios de cada dia. Dentro do diretório /etc/sarg é criado o seguinte arquivo:

```
root@cubietruck:/etc/sarg# nano relDiario.sh
```

Dentro dele é adicionado as seguintes linhas:

```
INICIO=$(date --date "0 days ago" +%d/%m/%Y)
FIM=$(date --date "0 day ago " +%d/%m/%Y)
sarg -f /etc/sarg/sarg.conf -d $INICIO-$FIM -p -x –z
```

As duas primeiras linhas que indicam INICIO e FIM coletam a data atual do sistema, enquanto a terceira linha é a responsável por gerar o relatório. Sendo que sarg é o comando para a geração de relatório; -f indica onde está localizado o arquivo de configuração sarg.conf; -d representa a data, neste caso a sintaxe $INICIO-$FIM representa a data inicial até a data final que é gerado o relatório; -p solicita que seja analisado todos os endereços de IP presentes nos registro de evento; as opções -x e -z são utilizadas para exibir na tela uma mensagem da execução do sarg, caso aconteça algum problema nas gerações dos relatórios, ao final é preciso salvar o arquivo Ctrl + O e depois sair Ctrl + X (YAKUSHEV, 2016). O próximo passo é atribuir a permissão de arquivo executável para o relDiario.sh assim como foi feito com o firewall.sh no capitulo 4.3, portanto no console é executado o seguinte comando:

```
root@cubietruck:/etc/sarg# chmod +x relDiario.sh
```

Agora é preciso enviar uma cópia do arquivo até o diretório sbin, pois a aplicação responsável pela execução automática busca os arquivos dentro desse diretório, então no console é utilizado o comando:

```
root@cubietruck:/etc/sarg# cp relDiario.sh /sbin/
```

Agora é editado o arquivo responsável pela execução automática, ele encontra-se no seguinte diretório /etc e para acessá-lo é preciso entrar com o seguinte comando:

root@cubietruck:/# cd /etc/

Dentro do diretório é editado o arquivo crontab, portanto:

root@cubietruck:/etc# nano crontab

Ao final do arquivo é preciso adicionar a seguinte linha:

00 22 * * * root /sbin/relDiario.sh

Após a inserção Ctrl + O para salvar e Ctrl + X para sair. É possível observar a linha adicionada, onde o 00 significa os minutos, 22 as horas. As sintaxes “dom” os dias do mês, “mon” o mês e “dow” os dias da semana, estas estão marcadas com * para que seja emitido o relatório todos os dias. Em seguida é informado o usuário que executa a ação, neste caso o root, seguido do caminho onde se encontra o arquivo executável relDiario.sh responsável por gerar os relatórios. Portanto todos os dias da semana e todos os meses do ano é gerado o relatório as 22:00 horas, os relatórios podem ser observados através do IP do Cubietruck seguindo de /sarg, neste projeto é possível acessá-los através do navegador de Internet utilizando o seguinte endereço:

192.168.1.100/sarg

A Figura 18 mostra a página inicial do Sarg com apenas um relatório gerado, nela é possível observar o período do relatório 2016May10 até 2016May10, além da data de criação, o número de usuário contidos nela e a quantidade de dados consumidos nesse período.

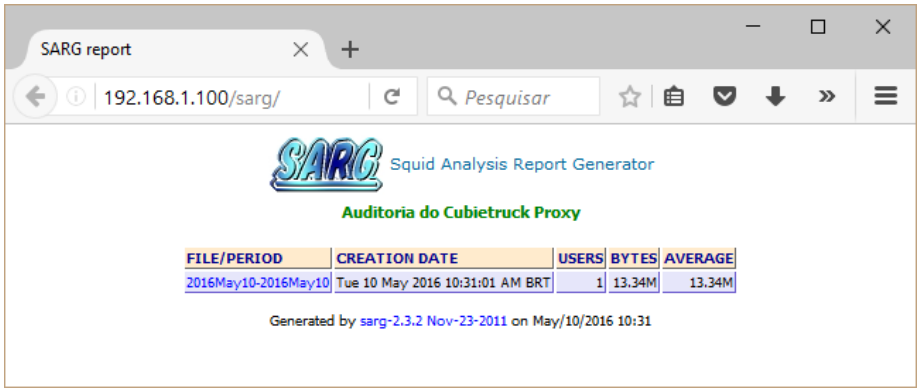

Figura 18 - Página do Sarg.
Fonte: Autoria própria.

Ao clicar sobre o período desejado é exibida a lista de usuários que efetuaram algum tipo de acesso à Internet, a Figura 19 mostra essa tela.

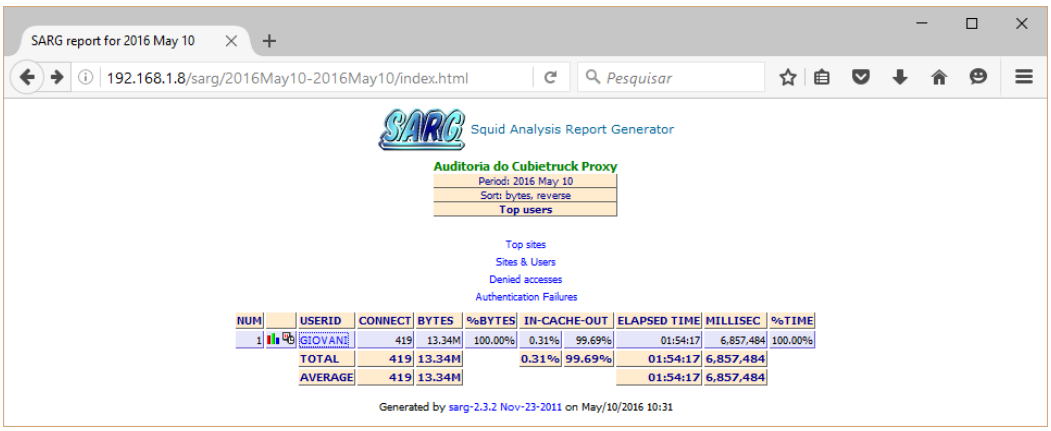

Figura 19 - Página de usuários.
Fonte: Autoria própria.

Na Figura 19 é possível observar que no campo USERID está sendo exibido o nome atrelado ao endereço IP que foi adicionado neste capítulo, caso algum endereço IP não esteja atribuído a nenhum nome é exibido o próprio IP. O relatório também apresenta o número de conexões estabelecidas e quantidade de dados consumidos por cada usuário. Ao clicar no USERID desejado é listado todos os sites acessados, os sites que foram bloqueados pelo Squid são exibidos com uma observação de DENIED.

A função do Sarg é gerar relatórios para fins de auditoria, por isso é composto de informações muitas vezes sigilosas de todos os usuários da rede, é recomendado que apenas pessoas autorizadas tenham acesso a esse tipo de informação e por isso é crucial a implantação de um sistema de autenticação. Para realizar a configuração de autenticação é preciso acessar o diretório /var/www.sarg.

root@cubietruck:/# cd /var/www/sarg/

Em seguida é criado o usuário para acesso ao Sarg, a técnica é muito semelhante aos usuários criados para o Squid no capítulo 4.2, sendo assim, no console é realizada a seguinte tarefa:

root@cubietruck:/var/www/sarg# htpasswd -c htpasswd giovani

Após a execução do comando é solicitado uma senha e a confirmação em seguida. Assim como no capítulo 4.2 a sintaxe -c é usada apenas na criação do primeiro usuário. O próximo arquivo que precisa de alterações está localizado no seguinte diretório:

root@cubietruck:/# cd /etc/apache2/sites-enabled/

Dentro deste diretório é preciso editar o seguinte arquivo:

root@cubietruck:/etc/apache2/sites-enabled# nano 000-default

Neste arquivo é possível notar que as linhas de códigos estão entre <Directory></Directory> que indica o local onde a regra irá ser aplicada, então após o último </Directory> deve ser inserido as linhas descritas abaixo.

```
<Directory "/var/www/sarg">
        Deny from all
        AuthType Basic
        AuthName "Informe o usuário e senha"
        AuthUserFile /var/www/sarg/htpasswd
        Require valid-user
        Satisfy Any
    </Directory>
```

A primeira linha a ser analisada é a *Deny from All*, seu significado é negar para todos, ou seja, bloquear qualquer tentativa de acesso. A segunda linha corresponde ao tipo de autenticação que corresponde ao tipo básico com codificação base64. A terceira é o nome que aparecerá na janela de autenticação. A quarta linha é o local onde é armazenado o usuário e senha. A quinta linha define que o usuário deve estar devidamente cadastrado para obter acesso. A sexta e última linha significa que qualquer acesso estará bloqueado exceto para os autenticados. Após as inserções salvar o arquivo usando Ctrl + O e Ctrl + X para sair.

Para aplicar as modificações realizadas é preciso reiniciar o serviço do Apache2, o comando para realizar tal tarefa é o seguinte:

```
root@cubietruck:/etc/apache2/sites-enabled# service apache2 restart
```

Agora ao tentar acessar novamente o endereço 192.168.1.100/sarg através do navegador de Internet é solicitado a inserção de usuário e senha para liberação do acesso aos relatórios de auditoria.

4.5 TESTES E RESULTADOS OBTIDOS

Nesta fase do projeto foi realizado os testes em duas etapas, a primeira teve o intuito de analisar a quantidade de banda suportada pelo equipamento. A segunda etapa teve o objetivo de analisar a quantidade de requisições HTTP que o Firewall consegue processar. Todos os testes foram realizados utilizando simuladores para gerar as conexões.

A primeira etapa dos testes foi realizada usando um computador equipado com placa de rede *Gigabit*, conectado diretamente ao Firewall através de um cabo de rede com categoria 5e. A Figura 20 representa a topologia de rede preparada para o teste.

Figura 20 - Topologia da primeiro etapa.
Fonte: Autoria própria.

O teste foi conduzido com três níveis de simulação. A primeira simulou a conexão entre um único usuário e o Firewall. O segundo nível foi à simulação com trinta usuários conectados ao Firewall simultaneamente. No último nível foi simulado a conexão de sessenta usuários simultâneos no Firewall. Para a realização deste teste foi usado um programa denominado Iperf no Firewall e Jperf no computador, apesar dos nomes serem distintos ambos são o mesmo programa, a variação de nome se deve ao fato de que o Iperf ser usado em sistemas operacionais Linux, enquanto o Jperf foi desenvolvido para Windows.

A aplicação Iperf é uma ferramenta nativa no Firewall, ou seja, não é necessário nenhuma instalação para seu funcionamento, pois ela já está integrada ao sistema operacional instalado. Para executar o programa no Firewall é preciso executar o seguinte comando:

```
root@cubietruck:~# iperf -s -f M
```

Este comando executa a aplicação Iperf em modo servidor (-s) e exibindo o formato dos resultados (-f) em *Megabytes* (M), sendo assim o Firewall estará aguardando para receber as conexões na porta 5001, porta padrão da aplicação.

A aplicação Jperf foi usada no computador com ambiente Windows na versão 2.0.2, que foi obtida no site http://www.download82.com/download/windows/jperf/. Ao baixar a aplicação é salvo um arquivo compactado, ao descompactá-lo é preciso apenas executar o arquivo jperf.bat. A Figura 21 exibe a tela do programa com as modificações necessárias para a realização do teste.

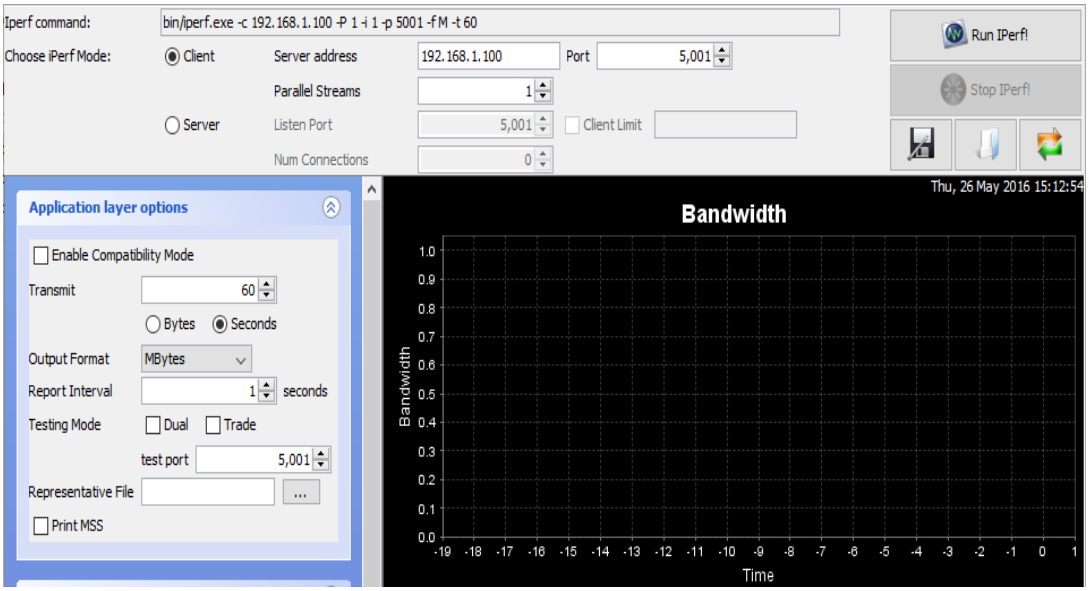

Figura 21 - Programa Jperf.
Fonte: Autoria própria.

A primeira coisa feita foi marcar a opção *Client*, que simulará as conexões no Firewall. No campo *Server address* (endereço do servidor) foi digitado o IP do servidor. No campo *Parallel Streams* (fluxos paralelos), é informado o número de conexões simultâneas a ser feitas, a primeira execução foi realizada com apenas uma. Outra alteração foi no campo *Transmit* (transmissão), nessa opção é informado o tempo de duração do teste, todos os testes foram executados com tempo de sessenta segundos. A última modificação é no campo *Output Format* (formato de saída), que foi alterada para *MBytes*. Ao final é dado início ao teste utilizando o botão *Run Iperf*! (Rodar Iperf).

Analisando os resultados obtidos é possível notar que a velocidade média de conexão foi de 50.80 *MBytes* por segundo e efetuou uma transferência média de 3046 *MBytes* de dados durante os sessenta segundos de teste.

O próximo teste foi executado com trinta conexões simultâneas, portanto no campo *Parallel Streams* o número foi alterado para trinta, as demais configurações permanecem inalteradas. Os dados obtidos apresentaram uma velocidade média total de 61.9 *MBytes* por segundo, com isso é possível calcular a média por usuário, usando a média total dividido pelo total de usuários. A velocidade média por usuário encontrada é de 2.06 *Mbytes* por segundo e uma transferência média de 3759 *MBytes* de dados.

O último teste utilizando o aplicativo Jperf foi simulando a conexão com sessenta usuários simultâneos, no campo *Parallel Streams* o número digitado foi

sessenta. Com os dados coletados é possível observar uma transferência média de 3782 *MBytes* de dados, enquanto a velocidade média total foi de 61.5 *MBytes* por segundo, esse resultado traz uma média de 1.025 *MBytes* por segundo para cada usuário. A Figura 22 demonstra os dados coletados durante todo o teste.

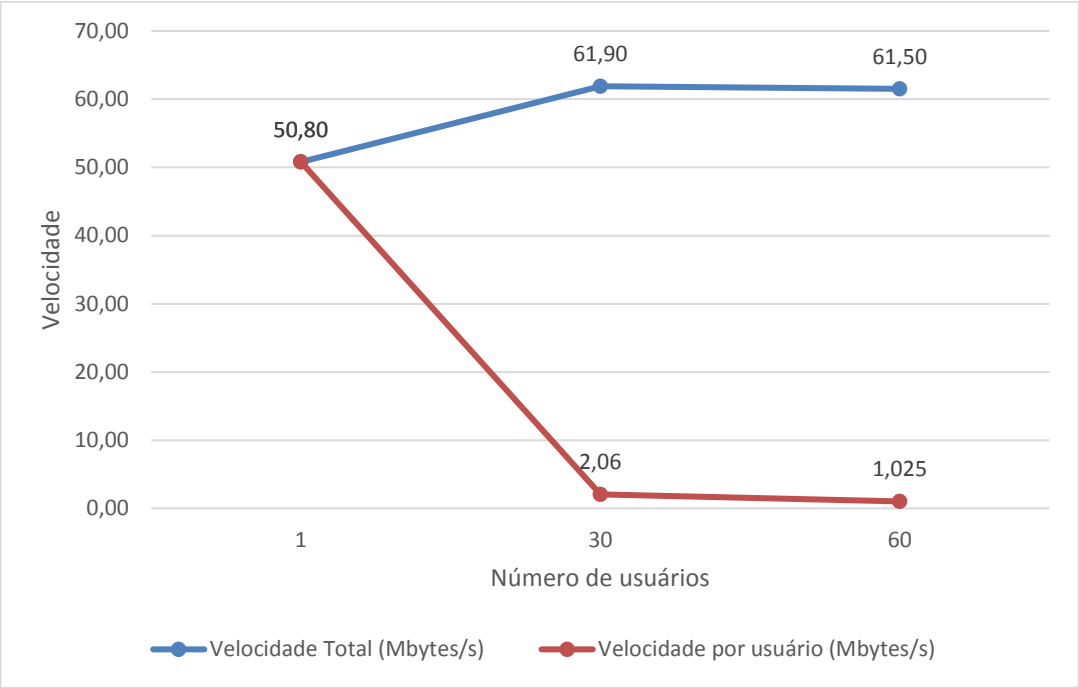

Figura 22 - Gráfico de resultados.
Fonte: Autoria própria.

Na Figura 22 é possível observar o gráfico com a velocidade total e velocidade por usuário obtidas. A velocidade total atingiu as marcas de 50.80, 61.90 e 61.50 respectivamente com um, trinta e sessenta usuários. Enquanto a velocidade por usuário foi de 50.80 com um usuário, 2.06 com trinta usuários e 1.025 com sessenta usuários.

A segunda etapa dos testes foi realizada utilizando o mesmo computador equipado com placa de rede *Gigabit*, conectado ao equipamento de modelo DSL-2730R através de um cabo de rede com categoria 5e, o Firewall também foi conectado ao equipamento DSL-2730R utilizando um cabo de rede com categoria 5e. A Figura 23 representa a topologia de rede preparada para o teste.

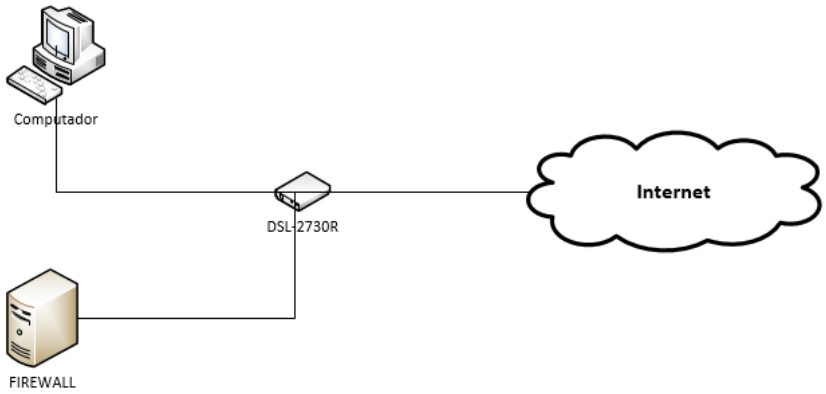

Figura 23 - Topologia da segunda etapa
Fonte: Autoria própria.

O equipamento utilizado tem marca D-Link e através das especificações técnicas foi possível constar que o dispositivo é equipado com entradas ethernet de velocidade 100 Mbps (Megabit por segundo).

Está segunda etapa também foi conduzida com três níveis de simulação. A primeira simulou a conexão entre um único usuário e o Firewall. O segundo nível foi a simulação com trinta usuários conectados ao Firewall simultaneamente. No último nível foi simulado a conexão de sessenta usuários simultâneos no Firewall.

A primeira configuração feita para a condução do teste foi realizada no Firewall, comentando as linhas que se referem a autenticação por usuário e senha para acesso à Internet do Squid. Essa configuração foi necessária devido ao fato do programa usado para gerar as conexões não conseguir se autenticar ao Firewall para liberar o acesso as requisições HTTP. As linhas que foram comentadas são as seguintes:

```
#auth_param basic program /usr/lib/squid3/ncsa_auth /etc/squid3/passwd
#acl autenticados proxy_auth REQUIRED
#http_access allow autenticados
```

Para a realização deste teste foi usado um programa denominado JMeter na versão 2.13, a aplicação mencionada funciona em sistemas operacionais Windows e pode ser encontrada no site https://sourceforge.net/projects/jmeterforwindows/. O arquivo salvo é um executável de fácil instalação, basta seguir as instruções que o programa fornece.

Ao abrir o programa JMeter é possível verificar que ao lado direito da tela consta duas opções Plano de Teste e Área de Trabalho. Ao clicar com o botão direito do mouse na opção Plano de Teste e depois em adicionar é exibido vários tipos de testes que podem ser conduzidos. Para dar início a configuração deste teste foi clicado com o botão direito na opção Plano de Teste, depois em Adicionar, em seguida Threads (Users) e por último em Grupo de Usuários. Esta opção é responsável por gerar as conexões que serão configuradas após serem adicionadas as demais opções.

Em seguida foi adicionada a opção responsável por gerar as requisições HTTP, para isso foi preciso clicar com o botão direito do mouse em Grupo de Usuários, depois em Adicionar, em seguida na opção Testador e por fim em Requisições HTTP. Essa opção foi configurada como mostra a Figura 24, pois as configurações feitas não serão mais alteradas.

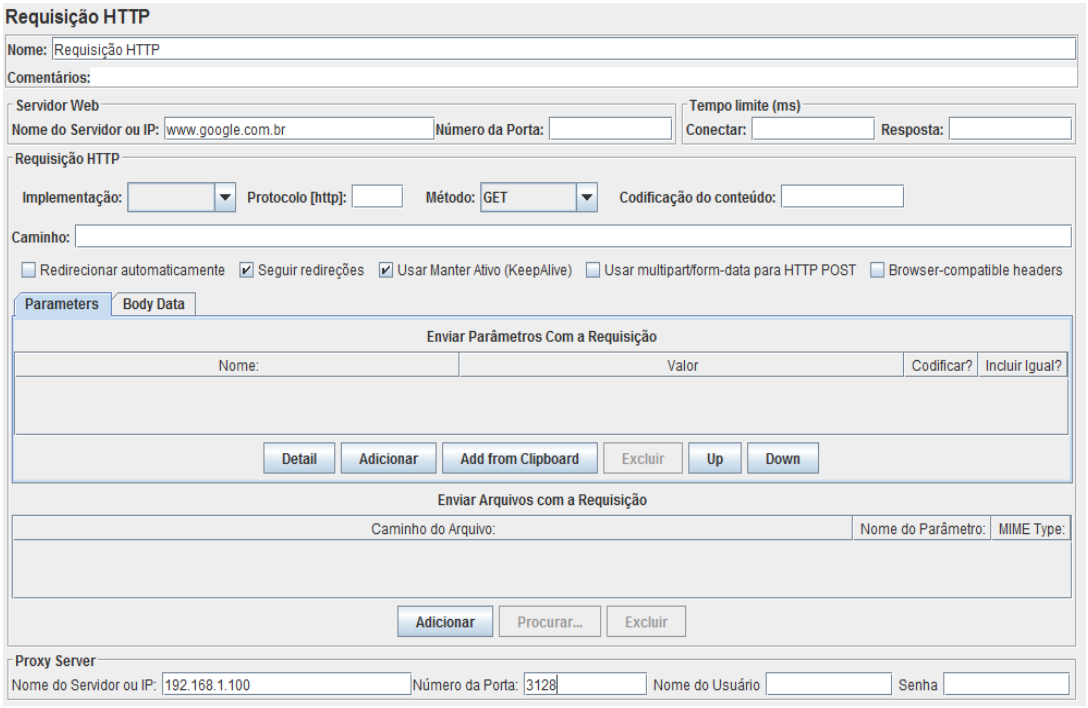

Figura 24 - Configurações das requisições HTTP.
Fonte: Autoria própria.

A Figura 24 mostra que foi inserido o site do Google no campo Nome do Servidor ou IP da aba servidor web, esta opção serve para simular o site em que os usuários estão acessando. Outro campo que foi adicionado informações foi o do nome do servidor na aba proxy server, nela foi inserido o endereço IP do Firewall seguido da porta de atuação.

As próximas opções que foram adicionadas são apenas para fim de análises, são elas as responsáveis por apresentarem os resultados. Portanto para adicioná-las foi clicado com o botão direito do mouse em Grupo de Usuários, em seguida Adicionar, depois em Ouvintes foi adicionado duas opções a de Ver Resultados em Tabela e Gráfico Agregado. A hierarquia do Plano de Teste deve ser seguida como mostra a Figura 25, pois pode influenciar nos resultados do teste.

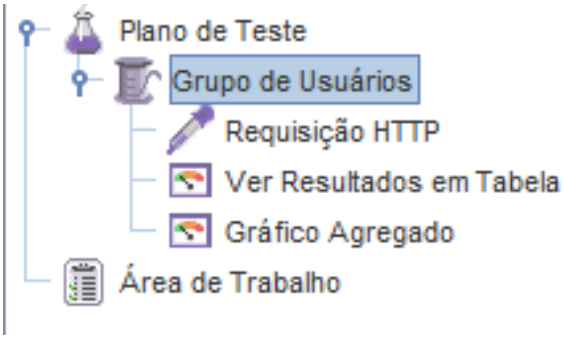

Figura 25 - Plano de Teste
Fonte: Autoria própria.

Na Figura 25 é possível ver como o teste vai ser executado, primeiro ele vai gerar as conexões simultâneas no Grupo de Usuários para depois acessar o endereço configurado nas Requisições HTTP e enviar para o Firewall, por fim os dados são coletados nas duas últimas opções.

Ao clicar sobre a opção Grupo de Usuários é possível definir a quantidade de usuários simultâneos, a duração do teste e quantas vezes o teste vai ser repetido. A Figura 26 demonstra a opção Grupo de Usuários.

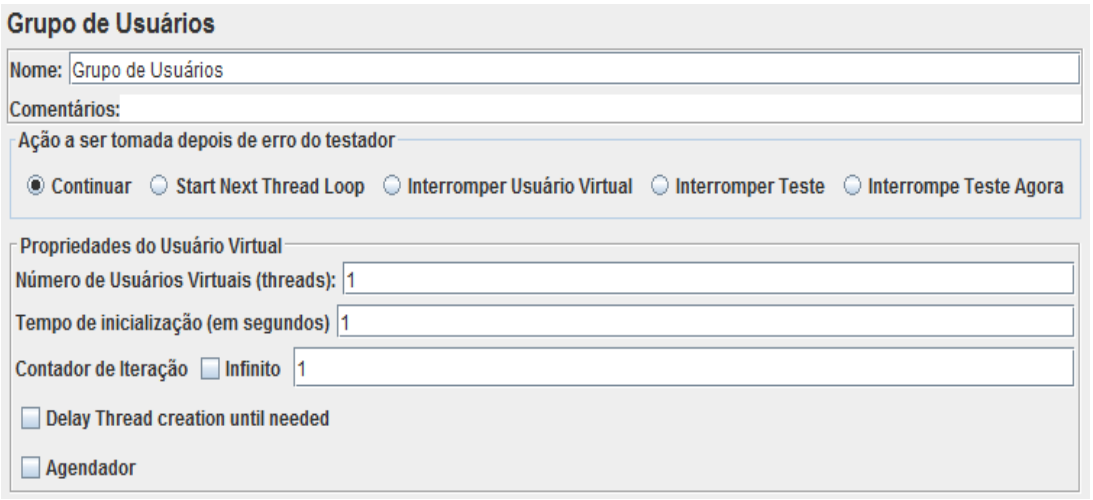

Figura 26 - Grupo de usuários.
Fonte: Autoria própria.

A Figura 26 apresenta as opções sem nenhuma modificação, no entanto, o primeiro nível de teste foi realizado com um usuário simultâneo, com tempo de inicialização de sessenta segundos e dez interações, ou seja, o usuário realiza dez requisições HTTP. O resultado do teste pode ser analisado na Figura 27 que corresponde à opção Ver Resultados em Tabela.

Amostra #	Tempo de início	Nome do Usuári...	Rótulo	Tempo da amost...	Estado	Bytes	Latency	Connect Time(ms)
1	15:29:05.997	Grupo de Usuári...	Requisição HTTP	114		11703	110	3
2	15:29:06.122	Grupo de Usuári...	Requisição HTTP	112		11680	108	3
3	15:29:06.240	Grupo de Usuári...	Requisição HTTP	108		11656	104	2
4	15:29:06.356	Grupo de Usuári...	Requisição HTTP	208		11663	113	8
5	15:29:06.572	Grupo de Usuári...	Requisição HTTP	117		11657	110	2
6	15:29:06.696	Grupo de Usuári...	Requisição HTTP	112		11601	105	2
7	15:29:06.816	Grupo de Usuári...	Requisição HTTP	114		11633	107	2
8	15:29:06.938	Grupo de Usuári...	Requisição HTTP	119		11648	110	3
9	15:29:07.065	Grupo de Usuári...	Requisição HTTP	133		11683	108	2
10	15:29:07.207	Grupo de Usuári...	Requisição HTTP	126		11713	109	3

Figura 27 - Resultados em tabela com um usuário.
Fonte: Autoria própria.

Na Figura 27 é possível observar o tempo de conexão de todas as requisições em milissegundos e o tempo conectado também em milissegundos. Outros dados também são apresentados como a latência, a quantidade de bytes e o estado do pacote, se o símbolo correspondente for verde significa que o pacote foi tratado, caso contrário seria exibido um símbolo em laranja. A Figura 28 apresenta os resultados da opção Gráfico Agregado.

Rótulo	# Amostras	Média	Mediana	90% Line	95% Line	99% Line	Mín.	Máx.	% de Erro	Vazão
Requisição H...	10	126	114	133	133	208	108	208	0,00%	7,5/sec
TOTAL	10	126	114	133	133	208	108	208	0,00%	7,5/sec

Figura 28 - Gráfico agregado com um usuário.
Fonte: Autoria própria.

A opção de Gráfico Agregado exibido na Figura 28 traz informações bem detalhadas, como a porcentagem de erros, ou seja, a porcentagem de pacotes em que o Firewall não conseguiu processar. Outra informação como a coluna 90% *Line,* mostra que mais de 90% dos pacotes atingiram o tempo de 133 milissegundos; na coluna 95% *Line* mostra que mais de 95% dos pacotes atingiram o tempo de 133 milissegundos; e na coluna 99% *Line* mostra que mais de 99% dos pacotes atingiram o tempo de 208 milissegundos, além de um mínimo, um máximo e uma média em milissegundos que foi de 126 milissegundos.

O próximo nível de teste foi realizado com trinta usuários simultâneos, para efetuar a alteração foi acessado a opção Grupo de Usuários e alterado o campo Número de Usuários Virtuais para trinta. Outra configuração feita foi limpar os

registros obtidos no teste anterior para que não influencie no próximo, para tal, foi preciso acessar a opção Executar e em seguida Limpar Tudo. Assim o teste foi executado e as informações da opção Resultados em Tabela podem ser vistas na Figura 29.

Amostra #	Tempo de início	Nome do Usuário...	Rótulo	Tempo da amostra (ms)	Estado	Bytes	Latency	Connect Time(ms)
274	20:20:43.208	Grupo de Usuári...	Requisição HTTP	127		11679	123	2
275	20:20:43.348	Grupo de Usuári...	Requisição HTTP	132		11696	124	2
276	20:20:43.493	Grupo de Usuári...	Requisição HTTP	127		11686	119	2
277	20:20:43.633	Grupo de Usuári...	Requisição HTTP	128		11729	121	2
278	20:20:43.774	Grupo de Usuári...	Requisição HTTP	125		11713	119	2
279	20:20:43.912	Grupo de Usuári...	Requisição HTTP	136		11681	128	3
280	20:20:44.061	Grupo de Usuári...	Requisição HTTP	133		11711	125	2
281	20:20:44.775	Grupo de Usuári...	Requisição HTTP	129		11689	120	2
282	20:20:44.917	Grupo de Usuári...	Requisição HTTP	130		11633	124	2
283	20:20:45.060	Grupo de Usuári...	Requisição HTTP	147		11630	139	2
284	20:20:45.220	Grupo de Usuári...	Requisição HTTP	126		11690	119	1
285	20:20:45.360	Grupo de Usuári...	Requisição HTTP	130		11665	121	2
286	20:20:45.503	Grupo de Usuári...	Requisição HTTP	130		11705	122	2
287	20:20:45.646	Grupo de Usuári...	Requisição HTTP	139		11675	120	2
288	20:20:45.797	Grupo de Usuári...	Requisição HTTP	135		11697	124	3
289	20:20:45.945	Grupo de Usuári...	Requisição HTTP	138		11640	129	2
290	20:20:46.097	Grupo de Usuári...	Requisição HTTP	132		11655	123	2
291	20:20:46.769	Grupo de Usuári...	Requisição HTTP	137		11764	126	2
292	20:20:46.919	Grupo de Usuári...	Requisição HTTP	130		11687	121	2
293	20:20:47.062	Grupo de Usuári...	Requisição HTTP	129		11665	121	2
294	20:20:47.204	Grupo de Usuári...	Requisição HTTP	134		11664	126	2
295	20:20:47.351	Grupo de Usuári...	Requisição HTTP	132		11714	124	2
296	20:20:47.496	Grupo de Usuári...	Requisição HTTP	133		11697	126	2
297	20:20:47.642	Grupo de Usuári...	Requisição HTTP	137		11665	129	2
298	20:20:47.792	Grupo de Usuári...	Requisição HTTP	130		11632	122	2
299	20:20:47.935	Grupo de Usuári...	Requisição HTTP	132		11673	124	2
300	20:20:48.081	Grupo de Usuári...	Requisição HTTP	129		11674	120	3

Figura 29 - Resultados em tabela com trinta usuários.
Fonte: Autoria própria.

A Figurada 29 demonstra que foram realizadas trezentas requisições, pois cada um dos trinta usuários realizou dez requisições. Agora na Figura 30 é apresentado os resultados da opção Gráfico Agregado.

Rótulo	# Amostras	Média	Mediana	90% Line	95% Line	99% Line	Mín.	Máx.	% de Erro	Vazão
Requisição H...	300	132	129	134	138	284	121	351	0,00%	5,0/sec
TOTAL	300	132	129	134	138	284	121	351	0,00%	5,0/sec

Figura 30 - Gráfico agregado com trinta usuário.
Fonte: Autoria própria.

A Figura 30 traz novamente 0% de erro dos pacotes, ou seja, todos os pacotes foram processados, outro ponto a se observar é o tempo máximo e mínimo de resposta que atingiram 351 e 121 milissegundos respectivamente, ocasionando uma média de apenas 132 milissegundos. Vale ressaltar que mais de 90% dos pacotes atingiram o tempo de 134 milissegundos, mais de 95% dos pacotes atingiram o tempo de 138 milissegundos e mais de 99% dos pacotes atingiram o tempo de 284 milissegundos.

O último nível foi realizado com sessenta usuários simultâneos, a alteração foi realizada na opção Grupo de Usuários e modificado o campo Número de Usuários Virtuais para sessenta. Novamente foi limpado os registros obtidos no teste anterior acessando a opção Executar e em Limpar Tudo. Por mais uma vez o teste

foi executado e as informações da opção Resultados em Tabela são apresentadas na Figura 31.

Amostra #	Tempo de início	Nome do Usuário...	Rótulo	Tempo da amostra (ms)	Estado	Bytes	Latency	Connect Time(ms)
574	20:03:38.666	Grupo de Usuári...	Requisição HTTP	131		13092	121	3
575	20:03:38.810	Grupo de Usuári...	Requisição HTTP	128		11680	118	2
576	20:03:38.950	Grupo de Usuári...	Requisição HTTP	128		11658	124	2
577	20:03:39.091	Grupo de Usuári...	Requisição HTTP	158		11657	150	3
578	20:03:39.218	Grupo de Usuári...	Requisição HTTP	128		11630	119	2
579	20:03:39.263	Grupo de Usuári...	Requisição HTTP	133		11656	126	2
580	20:03:39.359	Grupo de Usuári...	Requisição HTTP	127		11681	120	2
581	20:03:39.408	Grupo de Usuári...	Requisição HTTP	147		11673	139	3
582	20:03:39.498	Grupo de Usuári...	Requisição HTTP	127		11671	120	2
583	20:03:39.568	Grupo de Usuári...	Requisição HTTP	144		11640	135	2
584	20:03:39.638	Grupo de Usuári...	Requisição HTTP	133		11682	125	2
585	20:03:39.784	Grupo de Usuári...	Requisição HTTP	127		11648	118	2
586	20:03:39.923	Grupo de Usuári...	Requisição HTTP	126		11672	116	2
587	20:03:40.061	Grupo de Usuári...	Requisição HTTP	139		11698	130	2
588	20:03:40.213	Grupo de Usuári...	Requisição HTTP	130		11738	119	2
589	20:03:40.219	Grupo de Usuári...	Requisição HTTP	130		11688	121	2
590	20:03:40.356	Grupo de Usuári...	Requisição HTTP	127		11681	120	2
591	20:03:40.372	Grupo de Usuári...	Requisição HTTP	130		11686	121	2
592	20:03:40.496	Grupo de Usuári...	Requisição HTTP	135		11657	128	2
593	20:03:40.516	Grupo de Usuári...	Requisição HTTP	133		11696	124	2
594	20:03:40.663	Grupo de Usuári...	Requisição HTTP	130		11697	121	1
595	20:03:40.806	Grupo de Usuári...	Requisição HTTP	132		11713	124	3
596	20:03:40.951	Grupo de Usuári...	Requisição HTTP	129		11657	120	2
597	20:03:41.093	Grupo de Usuári...	Requisição HTTP	132		11710	123	2
598	20:03:41.238	Grupo de Usuári...	Requisição HTTP	129		11705	121	2
599	20:03:41.379	Grupo de Usuári...	Requisição HTTP	134		11656	126	2
600	20:03:41.527	Grupo de Usuári...	Requisição HTTP	125		11703	118	2

Figura 31 - Resultados em tabela com sessenta usuários.
Fonte: Autoria própria.

Na Figura 31 foi possível notar que mesmo com sessenta usuários conectados e seiscentas requisições o tempo de conexão e as demais colunas como latência e tempo conectado se mantiveram na média dos testes anteriores. Para um resultado mais detalhando foi observado a opção de Gráfico Agregado que é apresentado na Figura 32.

Rótulo	# Amostras	Média	Mediana	90% Line	95% Line	99% Line	Mín.	Máx.	% de Erro	Vazão
Requisição H...	600	132	129	137	143	258	119	360	0,00%	9,9/sec
TOTAL	600	132	129	137	143	258	119	360	0,00%	9,9/sec

Figura 32 - Gráfico agregado com sessenta usuário.
Fonte: Autoria própria.

Na Figura 32 novamente ocorreu 0% de erro dos pacotes e o tempo máximo e mínimo de resposta foram de 360 e 119 milissegundos respectivamente, ocasionando uma média de 132 milissegundos. Verificou-se que mais de 90% dos pacotes atingiram o tempo de 137 milissegundos, mais de 95% dos pacotes atingiram o tempo de 143 milissegundos e mais de 99% dos pacotes atingiram o tempo de 258 milissegundos.

Analisando a diferença de velocidade de conexão e tempo de resposta com um, trinta e sessenta usuários simultâneos gerados pelas ferramentas Jperf e Jmeter, constatou-se que a velocidade total de conexão entre um usuário e trinta aumentou em 11.1 *MBytes* por segundo, enquanto o tempo de resposta aumentou

em apenas 6 milissegundos. Comparando a velocidade total de conexão entre um usuário e sessenta ocorreu aumento de 10.7 *MBytes* por segundo, enquanto o tempo de resposta aumentou em 6 milissegundos. A velocidade de conexão entre trinta e sessenta usuários permaneceu quase inalterada com apenas 0.4 *MBytes* por segundo de diferença, a média por usuário foi de 2.06 *MBytes* por segundo com trinta conexões e 1.025 *MBytes* por segundo com sessenta conexões. O tempo de resposta entre trinta e sessenta usuários foram igualitárias, 132 milissegundos.

Devido a fatores como a limitação da estrutura física e lógica da rede montada para o teste e até mesmo da limitações de hardware do Firewall, fez com que o tempo de resposta variasse entre as simulações, apesar de ser uma diferença pequena não ocorreu nenhuma falha ou perda de pacotes de dados.

5 CONCLUSÃO

O projeto desenvolvido tem como objetivo apresentar uma alternativa de firewall com baixo custo para empresas de pequeno porte, através de um equipamento ARM e com softwares livres. O equipamento escolhido para desenvolver o projeto é chamado de Cubietruck, nele foi instalado o sistema operacional Linux com a distribuição Debian. Também foi instalado o serviço de Proxy com a ferramenta Squid, que usa um sistema de autenticação por usuário e senha para liberação do acesso à Internet, além disso, os filtros de conteúdo efetuam a análise do tipo de conteúdo que é bloqueado ou liberado para acesso. Foi implantado um *firewall* com o Iptables com a função de bloquear ataques ou ameaças vindas da Internet. O *firewall* tem uma ferramenta chamada Sarg, que gera relatórios diários de acesso de cada usuário a fim de aprimorar os filtros de conteúdo.

Este projeto apresentou a tecnologia ARM como sendo uma alternativa viável no uso de servidores, pois a redução do consumo de energia elétrica é consideravelmente alta. No entanto o equipamento ARM utilizado traz algumas limitações, tanto na parte de *hardware*, por possuir apenas uma placa de rede, quanto no software, que necessita de sistema operacional e aplicações compatíveis com a tecnologia.

Durante ambos os teste o processador atingiu 100% de uso, enquanto o uso da memória se manteve instável e sem alteração. Apesar disso, os testes sugerem que é viável a implantação de um Firewall em empresas de pequeno porte, pois permitiu gerenciar até sessenta usuários sem nenhuma perda de conexão, com boa velocidade de conexão e com tempo de resposta que não influencia na navegação dos usuários.

Pelo fato do firewall necessitar mão de obra especializada para realizar sua manutenção, pretende-se como trabalho futuro implantar uma interface gráfica, com o intuito de facilitar e agilizar a sua utilização.

REFERÊNCIAS

ALLWINNER TECHNOLOGY CO. **User Manual**. Disponível em: <http://dl.cubieboard.org/model/cubietruck/Hardware/soc/A20%20user%20manual%20V1.0%2020130322.pdf>. Acesso em: 25 mai. 2016.

ARMBIAN. **Cubietruck**. Disponível em: < http://www.armbian.com/cubietruck/>. Acesso em: 28 jun. 2016.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 17799. **Tecnologia da informação - Código de prática para a gestão da segurança da informação**. Rio de Janeiro, 2001. Disponível em: <http://www.saude.al.gov.br/arquivos/documento_tecnico/documento_tecnico_22-07-2014_15-19-45_NBR_ISO_IEC_17799.pdf>. Acesso em: 25 mai. 2016.

CABRAL, Paulo. **Firewall System**. Disponível em: < http://www.inf.ufsc.br/~bosco/ensino/ine5680/material-seg-redes/2014-1/iptables-script.htm > Acesso em: 28 abr. 2016.

COPEL, Companhia Paranaense de Energia. **Simulador de consumo de energia elétrica.** Disponível em: < http://www.copel.com/hpcopel/simulador/index.htm>. Acesso em: 26 mai. 2016.

CUBIETRUCK. **Cubietruck Cubieboard3**. Disponível em: < http://www.cubietruck.com/collections/frontpage/products/cubietruck-cubieboard3-cortex-a7-dual-core-2gb-ram-8gb-flash-with-wifi-bt-1>. Acesso em: 25 mai. 2016.

CUBIETRUCK. **Debian Server**. Disponível em: < http://dl.cubieboard.org/model/cubietruck/Image/Debian-server/>. Acesso em: 26 mai. 2016.

DEBIAN, Software in the Public Interest. **Sobre o Debian**. Disponível em: < https://www.debian.org/intro/about#history>. Acesso em: 26 mai. 2016.

DELL. **Desktop Deals**. Disponível em: < http://www.dell.com/us/business/p/deals/desktop-all-in-one-deals>. Acesso em: 24 jun. 2016.

EMBEDDED, Architects. **O que é um sistema embarcado.** Disponível em: <http://www.embarc.com.br/p1600.aspx>. Acesso em: 25 mai. 2016.

GOOGLE. **Conheça os chromebooks.** Disponível em: < http://www.google.com/chromebook/>. Acesso em: 25 mai. 2016.

HOFFMAN, Chris. **ARM vs. Intel: What It Means for Windows, Chromebook, and Android Software Compatibility.** Disponível em: <http://www.howtogeek.com/180225/arm-vs.-intel-what-it-means-for-windows-chromebook-and-android-software-compatibility/>. Acesso em: 25 mai. 2016.

KARPISCHEK, Ricardo Ueda. **Notas de Aula de TCP/IP.** Disponível em: < http://www.ime.usp.br/~ueda/ldoc/notastcp.html>. Acesso em: 27 jun. 2016.

LIMA, Thiago. **Cubietruck**. Disponível em: < http://www.embarcados.com.br/cubietruck/ >. Acesso em: 26 mai. 2016.

LINUX-SUNXI. **Bootable OS images**. Disponível em: <http://linux-sunxi.org/Bootable_OS_images>. Acesso em: 25 mai. 2016.

MICROSOFT. **What is a Proxy server.** Disponível em: <http://windows.microsoft.com/pt-br/windows-vista/what-is-a-proxy-server>. Acesso em: 25 mai. 2016.

MORIMOTO, Carlos E. **Linux Redes e Servidores: Guia Prático**, 2ª Edição. Porto Alegre: Editora Meridional, 2006.

NAKAMURA, Emilio Tissato; GEUS, Paulo Lício. **Segurança de Redes em ambientes cooperativos**. São Paulo: Novatec Editora, 2010.

SCHEER, Rodrigo de Arruda. **Segurança em pequenas empresas.** Disponível em: < http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/808/1/CT_TELEINFO_XX_2012_11.pdf>. Acesso em: 26 jun. 2016.

SECURELIST, AO Kaspersky Lab. **Network attacks.** Disponível em: < https://securelist.com/statistics/>. Acesso em: 25 mai. 2016.

SILVA, Gleydson Mazioli. **Guia Foca GNU/Linux**. Disponível em: < http://www.guiafoca.org/cgs/guia/avancado/ch-fw-iptables.html>. Acesso em: 26 mai. 2016.

SQUID-CACHE. **About Squid**. Disponível em: <http://www.squid-cache.org/Intro/>. Acesso em: 26 mai. 2016.

YAKUSHEV, Evgeniy. **Sarg - Command line options**. Disponível em: < https://sourceforge.net/p/sarg/wiki/Command%20line%20options/>. Acesso em: 10 mai. 2016.