CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U. nº 198, de 14/10/2016
AELBRA EDUCAÇÃO SUPERIOR - GRADUAÇÃO E PÓS-GRADUAÇÃO S.A.

Felipe Reis Pimentel

CONTROLE DE ACESSO WEB COM PROXY HTTP E FILTRO DNS

Palmas – TO

2020

Felipe Reis Pimentel

CONTROLE DE ACESSO WEB COM PROXY HTTP E FILTRO DNS

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Madianita Bogo Marioti.

Palmas – TO

2020

Felipe Reis Pimentel

CONTROLE DE ACESSO WEB COM PROXY HTTP E FILTRO DNS

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Madianita Bogo Marioti.

Aprovado em: _____/_____/_____

BANCA EXAMINADORA

_____

Prof. M.e Madianita Bogo Marioti

Orientador

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. M.e Jackson Gomes de Souza

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. Esp. Fábio Castro Araújo

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2020

Dedico este trabalho a duas mães que tive durante todo o processo. A primeira delas, Maria Arlete, minha progenitora, que tanto fez para me incentivar e torceu, talvez mais do que eu mesmo, para que o sonho da graduação, ainda que vinda tardiamente, se tornasse realidade.

Minha segunda mãe, a qual não posso deixar de dedicar esse trabalho, é a minha orientadora, Madianita. Não tenho palavras para agradecer a paciência, pelas broncas quando necessário e insistência para comigo. Você me segurou pelas mãos, pernas, orelha, e me acompanhou no trilhar deste caminho de conquista.

AGRADECIMENTOS

Agradeço a todos que colaboraram direta e indiretamente para a realização deste trabalho. Em especial a equipe da Escola Superior de Gestão Penitenciária e Prisional, ESGEPEN, meu local de trabalho, nas pessoas do Daniel Rodrigo de Araújo que estava mais ansioso do que eu para que o êxito fosse alcançado, e ao Carlesandro Ferreira Gaspar por segurar a barra no trabalho por várias vezes para que eu pudesse ter mais tempo para estudar. Agradeço a minha família, aqui representada pela minha mãe, Maria Arlete Reis, e pela minha irmã, Vitória Reis Santos, as quais foram meu suporte afetivo durante esse caminho. Agradeço aos amigos e outras pessoas importantes na minha vida, em especial ao grande Rodolpho Henrique Neiva, por me dar o suporte necessário em muitas coisas, quando eu não tinha como resolver. Tenho uma dívida de gratidão com todos vocês.

RESUMO

PIMENTEL, Felipe Reis. **Controle de Acesso WEB com Proxy HTTP e Filtro DNS**. 2020. 65 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas/TO, 2020[1].

O presente trabalho apresenta duas abordagens de controle de acesso WEB e, através do estudo de ferramentas com essas abordagens, realiza um comparativo entre essas ferramentas, auxiliando na escolha de uma delas de acordo com as necessidades específicas de aplicação ou, até mesmo, a possibilidade da utilização das ferramentas em conjunto. Inicialmente foi realizado o referencial teórico, para que houvesse a compreensão do funcionamento da filtragem WEB utilizando o *proxy* HTTP e o Filtro DNS. Posteriormente, a metodologia foi construída, definindo os parâmetros para o desenvolvimento dos cenários de rede, análises de rede e testes de controle de acesso. Entre as ferramentas selecionadas estão o *proxy* HTTP Squid e o filtro DNS NxFilter, utilizadas para a filtragem WEB. Com os parâmetros metodológicos definidos, os cenários de rede foram implantados, as ferramentas de filtragem instaladas, as análises de rede realizadas e os testes de controle de acesso concluídos. Em posse de todas as informações obtidas com o estudo, foi desenvolvido um paralelo, considerando os aspectos de cada ferramenta de filtragem WEB, para que se auxilie na escolha de uma delas para implantação.

Palavras-chave: Controle. Proxy. Filtragem. DNS.

[1] Elemento incluído com a finalidade de posterior publicação do resumo na internet. Sua formatação segue a norma ABNT NBR 6023, por isto o alinhamento e o espaçamento diferem do padrão do texto.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

DHCP - Dynamic Host Configuration Protocol

DNS – Domain Name System

GNU GPL - General Public License

HTTP - Hypertext Transfer Protocol

HTTPS - Hypertext Transfer Protocol Secure

IEC - International Electrotechnical Commission

IP – Internet Protocol

ISO - International Organization of Standardization

ISP - Internet Service Provider

MTR – My Traceroute

RFC - Request for Comments

SSL - Secure Sockets Layer

UDP - User Datagram Protocol

VM – Virtual Machine

SUMÁRIO

1 INTRODUÇÃO

Em um ambiente de rede pode haver a necessidade do controle de acesso à WEB por várias razões, inclusive, por questões de segurança neste ambiente. Um meio comumente utilizado para realizar a proteção de uma rede de computadores a acessos não autorizados na WEB é através de um serviço de *proxy*, que dá a possibilidade de se impor limites no acesso podendo, inclusive, definir tipos diferentes de permissões para determinados tipos de usuários.

Um serviço comumente utilizado para realizar esse controle de acesso é o *proxy*, serviço de rede em que o propósito inicial era separar redes corporativas da internet e, com o tempo, adquiriu outros recursos, inclusive para realizar a filtragem de conteúdo WEB (OLIVEIRA, 2009). O *proxy* se vale do protocolo de transferência mais utilizado na internet, o HTTP, para desempenhar suas funções.

Atualmente, uma das ferramentas mais utilizadas para implantar o serviço de *proxy* é o Squid, *software* licenciado pela GNU GLP, que permite realizar filtragem WEB (Squid-cache, 2020, online). Além da possibilidade de filtragem, o Squid também pode ser utilizado para o cache WEB, auxiliando na economia de banda e redução de tráfego WEB, além de *proxy* reverso, que auxilia no controle acesso em servidores WEB, interceptando as requisições enviadas por clientes WEB.

Outra abordagem que pode ser utilizada para realizar o processo de filtragem é a aplicação de regras de controle de acesso através do sistema de nomes de domínio, o DNS, essencial para acesso à internet. O DNS traduz os sites em endereços IP, para que se realize a conexão entre cliente e servidor. O NxFilter é um *software* que realiza a filtragem DNS, ou seja, através do sistema de resolução de nomes ele desempenha o papel de controle de acesso WEB, autorizando ou revogando este acesso (SHIGEZUMI, 2006). Possui versões paga e gratuita e, além do filtro DNS, oferece o WEB *cache,* filtragem remota e outros diversos recursos.

Dada a possibilidade de abordagens distintas para o controle de acesso à internet, é possível surgir dúvidas sobre qual método de filtragem WEB escolher ou, até mesmo, se o uso dos dois em conjunto é uma alternativa viável, considerando a diferença no método de cada ferramenta para efetivar o serviço de filtragem. Sendo assim, torna-se interessante

entender o funcionamento do processo de controle de acesso em cada tipo de abordagem, a fim de se realizar a melhor escolha para alcançar o resultado desejado.

O questionamento sobre qual método de filtragem utilizar ou se seria melhor usar os dois métodos em conjunto surgiu da necessidade do autor em aplicar a filtragem WEB no seu ambiente de trabalho, para controlar o acesso dos usuários da rede de computadores à internet. Percebe-se, assim, que a motivação do estudo se dá de um problema real e que pode estar presente em outros ambientes de trabalho em outras situações.

Este trabalho tem como intuito apresentar o serviço de *proxy* HTTP, utilizando a ferramenta Squid, e o filtro DNS, utilizando a ferramenta NxFilter, aplicando-os em cenários de rede, demonstrando o funcionamento. A partir dos resultados de testes e análises de rede nos diferentes cenários, realizar um paralelo entre esses dois serviços, o que pode auxiliar na escolha de um serviço de filtragem WEB.

2 REFERENCIAL TEÓRICO

Na vida pessoal e nas organizações, os dispositivos e informações presentes nas redes de computadores podem ser algo valioso no contexto em que estão inseridos. Partindo desse princípio, é importante garantir a segurança desses recursos. Para que redes de computadores possam ser utilizadas de forma segura, é necessário recorrer a métodos que garantam a proteção da rede.

De acordo com Santiago e Lisboa (2001, p. 2), "uma importante medida a ser tomada buscando a segurança da informação é a criação de uma política de segurança" que assegure o controle de acesso, seja de dispositivos internos da rede para uma rede externa, como a internet, ou de um acesso externo à dispositivos internos. A norma ISO/IEC 27000 (2016) define o controle de acesso como "meios para assegurar que o acesso a bens é autorizado e restrito com base nos requisitos de segurança e de negócios".

Um dos métodos utilizados para o controle de acesso numa rede de computadores é a implantação de um *firewall* de rede. O documento técnico RFC 2979 (2000, *online*) afirma que "*Firewalls* agem como um ponto final de protocolo e retransmissão (...), como um filtro de pacote ou alguma combinação de ambos". Com um *firewall* é possível controlar a entrada e saída de pacotes de rede, permitindo ou bloqueando os mesmos de acordo com a política de segurança estabelecida. Kurose e James (2013, p. 520) endossam essa afirmação dizendo que "um firewall localiza-se entre a rede da organização e a rede pública, controlando os acessos de pacote de e para a rede".

É interessante observar que o *firewall* funciona como um intermediário de pacotes de rede. Tanembaum (2003, p. 825) afirma que utilizar um *firewall* é como "cavar um fosso profundo em torno do castelo. Esse recurso forçava todos aqueles que quisessem entrar ou sair do castelo a passar por uma única ponte levadiça". O castelo seria a rede interna e a parte externa ao castelo seria qualquer rede externa, inclusive a internet. Dessa forma, é possível controlar os pacotes quanto a origem, destino, entrada e saída da rede interna.

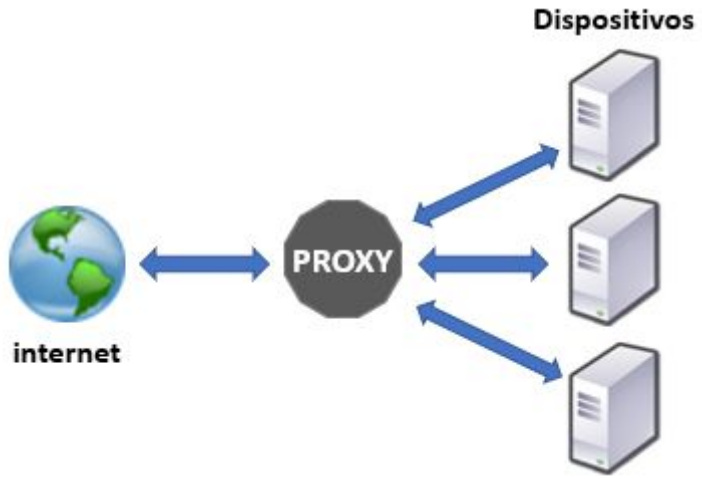
Utilizando a analogia realizada no parágrafo anterior, apesar do *firewall* de rede ser a "ponte levadiça do castelo", limitando o trânsito e possibilitando a filtragem de pacotes de rede quanto a origem e destino, ele não faz a filtragem baseada em conteúdo. Nesse caso, é preciso ter "guardas na ponte levadiça analisando tudo o que está sendo importado ou exportado no castelo".

Para que o controle de acesso numa rede de computadores seja realizado considerando o conteúdo solicitado na rede interna ou importado da internet, pode-se utilizar um *proxy* HTTP ou um filtro DNS, que serão abordados nas subseções a seguir.

2.1 *Proxy* de Rede

O serviço de *proxy* de rede foi utilizado inicialmente para separar uma rede de computadores da internet, por questões de segurança corporativa. De acordo com Oliveira (2009, p. 12), o *proxy* "funciona como um procurador ou representante, porque todas as requisições WEB da rede local são feitas através dele, que se encarrega de buscar as informações no mundo externo (internet)". Em vez de cada dispositivo da rede interna realizar as requisições diretamente pela internet, o *proxy* realiza essas requisições para todos os dispositivos de rede. A figura 1 exemplifica o funcionamento do *proxy* como intermediador de requisições WEB.

Figura 1: Funcionamento do *proxy* com requisições WEB.



Com o *proxy* sendo utilizado como intermediário de requisições WEB, surge a necessidade e a possibilidade de filtrar o conteúdo dessas requisições. A partir daí, o serviço de *proxy* ganha listas de controle de acesso, relatórios e outras funcionalidades que permitem gerenciar o acesso da WEB pelos dispositivos da rede (OLIVEIRA, 2009).

Além de intermediar as requisições WEB e filtrar o conteúdo requisitado, o *proxy* também pode funcionar como um WEB *cache*, armazenando localmente conteúdo web e disponibilizando na rede interna de acordo com as requisições. Segundo Yeu e Fedel (2014, p. 15), "isso permite agilizar o processo de acesso às páginas da Internet, reduzindo assim o tempo médio de espera para acessar um determinado conteúdo, através do método conhecido como cache de conteúdos Web".

Para que o servidor *proxy* exerça a função de WEB *cache*, ele armazena os principais conteúdos requisitados pelos dispositivos da rede interna na WEB e, caso ocorra uma nova

requisição do mesmo conteúdo, o *proxy* em vez de importar diretamente esse conteúdo na WEB, apenas repassa o conteúdo armazenado em *cache* para o dispositivo requisitante. Essa função pode auxiliar na economia de banda da internet, reduzindo a quantidade de requisições externas à rede.

O *proxy* utiliza uma porta de rede específica para receber e enviar as requisições WEB e tratá-las conforme sua configuração. Para isso, é necessário que os dispositivos e sistemas sejam configurados para direcionar as requisições WEB para a porta de rede utilizada pelo *proxy*. Esse método onde é necessário configurar os dispositivos para utilização do *proxy* é chamado de *proxy* não transparente.

Em determinadas situações, como na utilização de dispositivos pessoais em redes onde o uso do *proxy* é obrigatório, ter que configurar o dispositivo ao *proxy* da rede e, posteriormente, retirar essa configuração para utilizar o dispositivo em outra rede, não se mostra algo eficiente. Para que o *proxy* fosse utilizado na rede automaticamente, sem configuração adicional, foi desenvolvido um novo método. De acordo com Oliveira (2009, p. 13) "a este método damos o nome de *Proxy* transparente, pois toda requisição feita à porta 80 do *default gateway* é ao *Proxy*, que se encarrega de fazer a solicitação externa e posterior entrega". As requisições HTTP para o *gateway* de rede que são, por padrão, na porta 80, são redirecionadas para a porta do *proxy*, a fim de que ele gerencie as requisições WEB mesmo sem configuração adicional nos dispositivos da rede.

Para que o *proxy* possa realizar todas as funções citadas, ele necessita utilizar algum recurso de rede. Um recurso que pode ser explorado pelo *proxy* para seu devido funcionamento é o protocolo de transferência HTTP, que será relatado na subseção a seguir.

2.1.1 *Proxy* HTTP

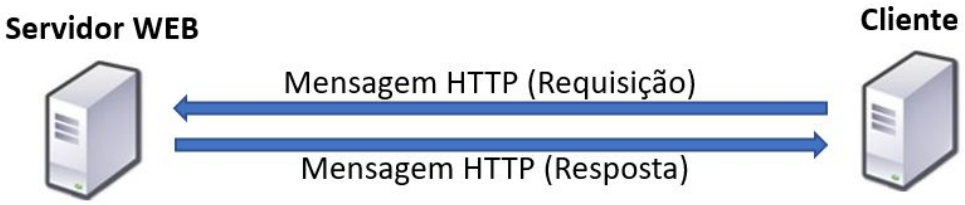
Conforme explicitado anteriormente, o *proxy* funciona como um filtro de requisições WEB, intermediando essas requisições entre a rede interna e a internet. Um meio comum do *proxy* cumprir com sua função de filtragem WEB é através do protocolo HTTP.

De acordo com o documento técnico RFC 2616 (1999, online), o *Hipertext Transfer Protocol* (HTTP) é um "protocolo do nível de aplicativo para informações distribuídas, colaborativas e sistemas hipermídia". Trata-se de um protocolo de transferência amplamente utilizado na internet, que permite a comunicação entre servidores WEB e dispositivos que necessitam utilizar os recursos disponíveis nesses servidores.

O funcionamento do protocolo HTTP é relativamente simples: o cliente envia uma mensagem ao servidor solicitando determinado recurso, e o servidor responde à mensagem do

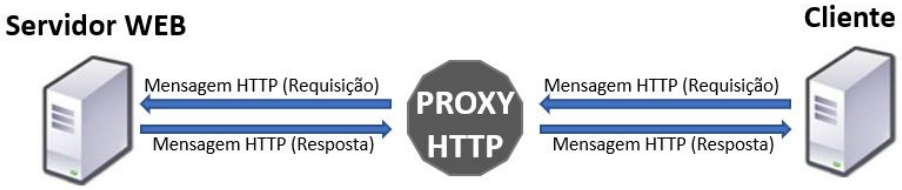
cliente. Kurose e James (2013, p. 72) afirmam que “o HTTP define a estrutura dessas mensagens e o modo como o cliente e o servidor as trocam”. As mensagens HTTP contém, basicamente, informação textual sobre origem e destino, recursos a serem transferidos pela WEB, e dados relacionados ao próprio recurso que se deseja utilizar como, por exemplo, *login* e senha. A Figura 2 demonstra o processo básico de troca de mensagens através do protocolo HTTP.

Figura 2: Processo básico de troca de mensagens HTTP.



Assim como pode ser observado na Figura 2, o cliente envia uma mensagem HTTP para o servidor, chamada de “requisição”. O servidor responde o cliente enviando o pedido contido na requisição, chamado de “resposta”. É importante observar que a requisição é sempre realizada pelo cliente de rede, o solicitante do recurso. O *proxy* HTTP intercepta as mensagens do cliente para o servidor, ou seja, as requisições para realizar a filtragem. A seguir, a Figura 3 demonstra o *proxy* HTTP trabalhando como intermediário das mensagens HTTP entre cliente e servidor.

Figura 3: *Proxy* HTTP como intermediário das mensagens HTTP.



Como pode ser observado na Figura 3, o cliente envia uma requisição com destino ao servidor WEB. A requisição passa pelo *proxy* HTTP, que cumpre a função de filtro WEB. Caso a requisição esteja autorizada a dar continuidade, baseada na política de segurança adotada, o *proxy* HTTP destina a requisição ao servidor WEB. Na resposta do servidor WEB, ele direciona a mensagem para o *proxy* HTTP, que redireciona para o cliente.

Apesar da forma simples de utilização do HTTP tornar a comunicação entre dispositivos na internet descomplicada, existe um risco de segurança no tráfego de requisições e respostas pelo protocolo. Da forma como ele funciona, teoricamente favorece interceptações de conteúdo das mensagens.

Baseado na necessidade de proteger informações relevantes que trafegam na rede pelo HTTP como dados bancários, senhas e informações confidenciais, foi criado o HTTPS. De acordo com Silva et al. (2009, p. 87) "HTTPS (Hypertext Transfer Protocol Secure) é a variação do protocolo HTTP com o protocolo SSL (Secure Sockets Layer) [...] que tem por finalidade garantir a segurança na transmissão de dados, em aplicações que envolvam dados sigilosos".

No processo de troca de mensagens HTTPS, a mensagem é criptografada para garantir a segurança das informações. Além disso, para garantir que a mensagem está sendo direcionada para o servidor correto, são utilizados certificados digitais, que possuem as chaves de descriptografia dos dados.

Enquanto o HTTP utiliza a porta 80 para a transmissão das mensagens, o HTTPS utiliza por padrão a porta 443. Sendo assim, o *proxy* HTTP, para interceptar a comunicação realizada por HTTPS, também deve se valer da porta 443 para realizar a filtragem WEB.

Para que a requisição possa chegar ao servidor WEB, o cliente precisa identificar o endereço de rede deste servidor na internet. Para isso, é utilizado o sistema de nomes de domínio, que será discorrido na seção a seguir.

2.2 Sistema de Nomes de Domínio (DNS)

Nas redes de computadores, inclusive na internet, os dispositivos são identificados pelo endereçamento IP, que consiste em quatro sequências de até três números separadas por pontos, no caso do IP versão 4, ou oito sequências de quatro caracteres hexadecimais separados por dois pontos, no caso do IP versão 6.

O endereço IP é utilizado para o endereçamento de servidores WEB na internet. Isso tornaria a identificação do endereço do servidor WEB difícil para o usuário, que teria que memorizar essas sequências para, por exemplo, digitá-las na barra de endereços do navegador e ter acesso ao serviço WEB.

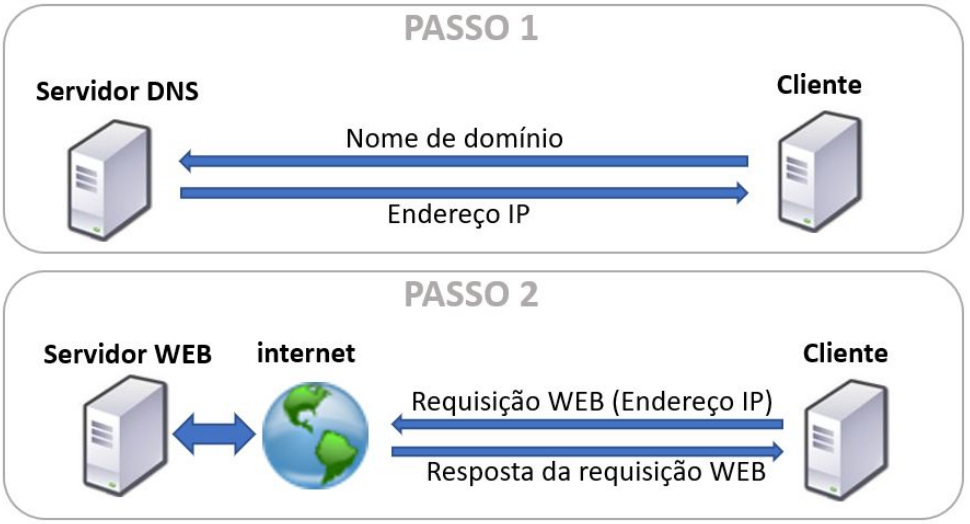
De acordo com Neves (2015, p. 43), "máquinas podem funcionar bem com números, mas o homem melhor se habitua a palavras, siglas ou denominações em geral". Para facilitar esse processo de memorização foram criados os nomes de domínio. O documento técnico RFC 1035 (1987, online) diz que "O objetivo dos nomes de domínio é fornecer um

mecanismo para nomear recursos de forma que os nomes possam ser usados em diferentes hosts, redes, famílias de protocolo, internets e organizações administrativas".

Os nomes de domínio são uma definição de nomes para identificar endereços IP na internet. Para efetivar a associação de nomes de domínio a endereços IP, foi criada a resolução de nomes de domínio, realizada através do sistema de resolução de nomes, ou DNS (NEVES, 2015).

O documento técnico RFC 8499 (2019, *online*) afirma que "o Sistema de Nomes de Domínio (DNS) é um protocolo de consulta-resposta simples cujas mensagens em ambas as direções têm o mesmo formato". O cliente faz uma consulta utilizando o nome de domínio do servidor WEB e o servidor DNS responde com endereço IP para que o cliente encontre o servidor WEB na internet. A figura 4 exemplifica o funcionamento do DNS.

Figura 4: Demonstração do funcionamento do DNS



Assim como pode ser observado na Figura 4, no passo um o cliente, que deseja alcançar um servidor WEB, realiza uma consulta enviando a requisição DNS com o nome de domínio para o servidor DNS. O servidor, por sua vez, traduz o nome de domínio em endereço IP e envia uma resposta para o cliente, com o endereço IP. No passo dois, o cliente realiza uma requisição para o servidor WEB na internet utilizando o endereço IP informado pelo servidor DNS. O servidor DNS pode estar interno à rede do cliente ou externo a ela, sendo fundamental que o cliente consiga alcançá-lo. Sendo assim, pode-se afirmar que o DNS é um sistema importante no processo de comunicação entre cliente e servidor WEB.

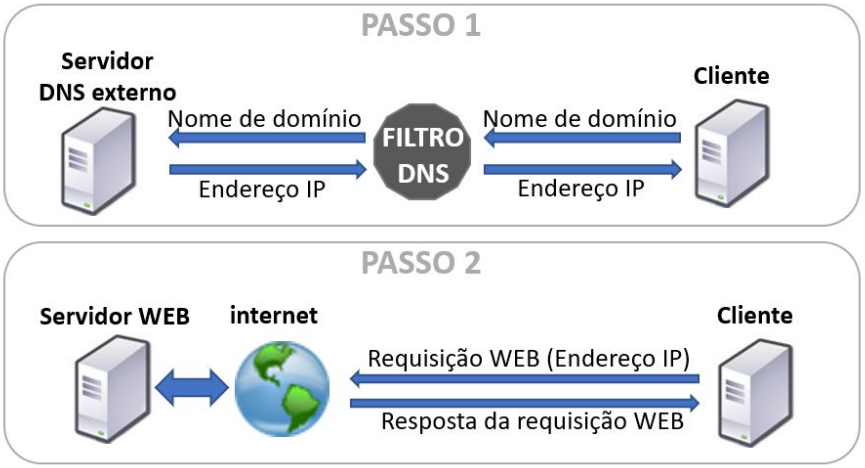
O fato de o DNS ser amplamente utilizado para a conexão entre cliente e servidor WEB faz com que surjam outras possibilidades de uso como, por exemplo, a filtragem WEB. A subseção a seguir trata do uso do DNS como um filtro WEB.

2.2.1 Filtro DNS

Assim como o HTTP, o DNS é parte importante para que uma requisição WEB de um recurso na internet seja alcançável. Essa dependência possibilita a utilização do DNS como ferramenta para filtragem da WEB.

Um filtro DNS consiste na interceptação da requisição WEB no momento em que ela solicita o endereçamento IP do servidor WEB de destino ao servidor DNS. Nesse momento é possível realizar a aprovação ou revogação desta requisição (SHIGEZUMI, 2006). Com a requisição sendo filtrada pelo servidor DNS, a necessidade de um intermediário entre as mensagens HTTP é eliminada. A Figura 5 demonstra o funcionamento de um filtro DNS implantado numa rede local.

Figura 5: Funcionamento de um filtro DNS implantado numa rede local



Como demonstrado na Figura 5, no passo um o cliente solicita a tradução de um nome de domínio em endereço IP para um servidor DNS que está exercendo a função de filtragem WEB. No caso em questão o filtro DNS está implantado na rede local e, caso o mesmo não possua o endereço IP do servidor WEB solicitado, ele irá consultar em um servidor DNS externo à rede, para repassar o endereço IP ao cliente. No passo dois o cliente, em posse do endereço IP do servidor WEB, realiza a requisição direcionada.

Naturalmente, toda requisição WEB feita utilizando um nome de domínio dependerá do servidor DNS. Sendo assim, pode-se considerar que o tempo utilizado na consulta e resposta a um servidor DNS é intrínseco a qualquer requisição, independente de haver um filtro WEB ou não. Levando em consideração que, na filtragem WEB realizada pelo *proxy* HTTP é necessário direcionar todas as requisições da rede interna para um ponto, funcionando como intermediário das mensagens entre cliente e servidor, teoricamente o filtro DNS não alongaria o tempo da troca de mensagens, pois ele utiliza um gargalo necessário e já existente no processo de troca de mensagens, não necessitando criar outro gargalo como no *proxy* HTTP.

Ainda com relação a questão do tempo nas requisições WEB, as mensagens DNS tendem a ser mais rápidas, pois utilizam o protocolo UDP para transporte de mensagens, além de serem menores e mais simples que as mensagens HTTP.

No filtro DNS, a troca de mensagens entre cliente e servidor WEB não possui intermediário, pois a filtragem é realizada antes mesmo da requisição do cliente ser direcionada ao servidor.

A partir dos conceitos que foram abordados no referencial teórico, obteve-se uma base de conhecimento para estabelecer os procedimentos e materiais a serem adotados para o desenvolvimento do trabalho. A seguir, na seção de materiais e métodos, os recursos envolvidos e processos adotados são devidamente detalhados.

3 MATERIAIS E MÉTODOS

Esta seção apresenta os processos e recursos que foram utilizados para o desenvolvimento do projeto. O material se refere ao hardware utilizado para a criação do ambiente de rede e dos softwares utilizados para o desenvolvimento do trabalho. A metodologia apresenta os procedimentos que foram adotados para a aplicação do controle de acesso à WEB, além dos testes realizados para obter os resultados para a análise dos cenários. O desenho do estudo é abordado na subseção a seguir.

3.1 Desenho do Estudo

O presente trabalho consiste em apresentar e comparar dois métodos de controle de acesso à WEB, a partir de um *proxy* HTTP e de um filtro DNS. Para tanto, os controles de acesso foram aplicados em quatro cenários de rede: o primeiro cenário sem aplicação de controle de acesso à WEB; o segundo cenário com a aplicação de um *proxy* HTTP para controle de acesso à WEB; o terceiro cenário com a aplicação de um filtro DNS para controle de acesso à WEB; o quarto cenário com a aplicação conjunta de um *proxy* HTTP e um filtro DNS para controle de acesso à WEB.

A utilização dos conceitos citados no referencial teórico se dá a partir da aplicação do controle de acesso à WEB por um *proxy* HTTP e por um filtro DNS em cenários distintos de rede, além da análise dos resultados obtidos com os testes em cada um dos cenários.

O trabalho se trata de uma pesquisa aplicada, ou seja, utiliza os conhecimentos de determinados conceitos para buscar solucionar um problema. Sendo assim, foi apresentada a utilização dos conceitos de *proxy* HTTP e Filtro DNS para realizar o controle de acesso à WEB em cenários de rede. A figura 6 traz uma representação do fluxo do trabalho.

Figura 6: Representação do fluxo de trabalho.

Inicialmente, foi elaborado o referencial teórico, que disponibilizou a base conceitual necessária para a realização do trabalho. Em seguida, os cenários de rede foram desenvolvidos a partir dos conhecimentos obtidos no referencial e dos materiais e procedimentos definidos

na metodologia. Depois, os testes e análises de rede foram realizados nos quatro cenários de rede que foram criados, a fim de levantar as informações necessárias sobre o controle de acesso à WEB. Por fim, a partir das observações dos resultados dos dados, foram levantadas as informações para o paralelo entre o *proxy* HTTP e o filtro DNS, que foi gerado visando demonstrar a efetividade do uso de cada conceito em cada situação testada e analisada.

Na próxima seção, os materiais que foram utilizados para o desenvolvimento do trabalho são detalhados.

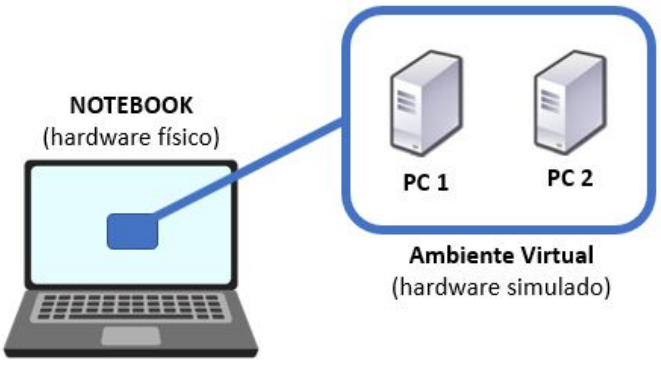
3.2 Materiais

O referencial teórico do trabalho foi elaborado utilizando livros, artigos e monografias, além de normas e documentos técnicos provenientes de institutos normativos internacionais e, de modo secundário, sites de conteúdo técnico sobre a utilização das ferramentas que integraram o projeto. Para o desenvolvimento do trabalho foram utilizados os recursos de hardware e software apresentados nas próximas seções.

3.2.1 Hardware e Padrão de Cenários

O *hardware* para o desenvolvimento do trabalho foi definido de forma a utilizar o mínimo de dispositivos e recursos necessários para a implantação dos *softwares* utilizados. Os cenários de rede foram criados utilizando o recurso de máquina virtual em um *notebook*, ou seja, o *hardware* dos computadores utilizados no projeto são criados via *software*. A seguir, na figura 7, a utilização do hardware é representada.

Figura 7: Demonstração do hardware utilizado no trabalho.



Para a implantação dos cenários de rede propostos no trabalho, foram utilizados dois computadores no ambiente virtual, sendo um servidor e um cliente. O servidor foi o

responsável pela intermediação da conexão à internet para o cliente de rede gerando, assim, um padrão para a criação dos quatro cenários de rede propostos no trabalho. Na Figura 8 é possível compreender a configuração padrão dos cenários de rede.

Figura 8: Configuração padrão dos cenários de rede.

Em cada cenário de rede, o padrão demonstrado na Figura 8 foi seguido e os computadores utilizados foram configurados da seguinte forma:

- o computador utilizado como servidor de rede continha duas placas de rede, sendo uma ligada à rede externa em modo *bridge* com a placa de rede do computador hospedeiro da virtualização, e a outra ligada na rede interna virtualizada. O *software* utilizado neste computador foi o pfSense. Ele foi configurado de forma a prover o serviço de *firewall de rede*, *gateway* de rede e servidor DHCP na rede interna, fazendo com que os demais dispositivos na rede interna dependam do servidor de rede para qualquer acesso externo à rede. Essa dependência se faz necessária considerando que o objetivo do trabalho perpassa por analisar o controle de acesso à WEB e, para realizar esse controle, é necessário limitar o acesso por uma única via;
- o computador utilizado como cliente de rede continha apenas uma placa de rede, conectada à rede interna virtualizada. O sistema operacional utilizado foi o Ubuntu Desktop, versão para computadores pessoais e de trabalho. Para que o cliente de rede pudesse acessar a internet, era necessário que as configurações de rede estivessem ajustadas para a utilização do servidor de rede como *gateway* de rede.

A partir das definições do *hardware* utilizado e da configuração padrão dos cenários de rede, os *softwares* puderam ser instalados, configurados e utilizados conforme a necessidade de cada cenário. Os recursos de *software* utilizados são descritos na próxima seção.

3.2.2 Software

Para o desenvolvimento do trabalho foram utilizadas as seguintes ferramentas: Sistema Operacional Ubuntu, pfSense, Oracle VM VirtualBox, *proxy* HTTP Squid, Filtro DNS NxFilter, Namebench, Firewall e MTR. Todas as ferramentas são descritas nas subseções a seguir.

3.2.2.1 Sistema Operacional Ubuntu

O Ubuntu é um sistema operacional para servidores, baseado na plataforma UNIX. Possui a versão *desktop*, para estações de trabalho, e a versão *server*, para servidores. A versão *desktop* é gratuita, possui todo um aparato de segurança além de fácil instalação (Canonical, 2020, online).

A utilização do Ubuntu versão *desktop* foi justificável tendo em vista que sua performance e estabilidade numa máquina virtual foram satisfatórias, além de permitir a instalação e utilização dos demais softwares utilizados no trabalho de forma simplificada.

3.2.2.2 pfSense

De acordo com Neves et. al (2014, p.24) "O pfSense é um software livre customizado da distribuição do FreeBSD, sendo adaptado para uso como firewall e roteador, que é inteiramente gerenciado via interface WEB". O pfSense pode ser utilizado como um servidor de rede, tendo em vista que, além das funções já disponíveis no próprio *software*, outras funções podem ser agregadas através do *download* e instalação de pacotes, inclusive, *softwares* para filtragem WEB.

O pfSense foi escolhido para o trabalho tendo em vista que as funções intrínsecas de um servidor de rede (gateway e firewall) já estão incluídas no *software*, otimizando o tempo que seria utilizado configurando essas funcionalidades em um sistema operacional. Além disso, o PfSense proveu o suporte para os demais softwares que foram utilizados no servidor de rede.

3.2.2.2 Oracle VM VirtualBox

A ferramenta VirtualBox possui como sua função a virtualização de um ambiente de hardware nas arquiteturas x86 e x64, propiciando a implantação e utilização de sistemas operacionais e ferramentas diversas. De acordo com Oracle (2020, online) "O VirtualBox não é apenas um produto de alto desempenho e extremamente rico em recursos para clientes corporativos, mas

também é a única solução profissional que está disponível gratuitamente como Software de código-fonte aberto".

A ampla compatibilidade com vários sistemas e a configuração simples fizeram com que VirtualBox fosse a ferramenta de máquina virtual escolhida para ser utilizada neste trabalho.

3.2.2.3 Proxy HTTP Squid

O Squid é uma ferramenta cuja finalidade é disponibilizar o serviço de *proxy* HTTP. O Squid oferece controle de acesso, autorização, ambiente de registro, aplicativos de serviço de conteúdo e opções de otimização de tráfego, que já vêm configuradas por padrão, para simplificar a implantação e o uso (Squid-cache, 2020, online).

A ferramenta Squid foi selecionada para compor este trabalho pois, de acordo com Curi e Filho (2019, p. 3), o "*Proxy* Squid é um software muito popular no ambiente computacional, de código aberto e gratuito para uso comercial ou pessoal". Essa afirmação permite induzir sua estabilidade na entrega do serviço, além do fato de possuir ampla documentação.

3.2.2.4 Filtro DNS NxFilter

O NxFilter é uma ferramenta de filtro WEB, que utiliza o DNS para atingir seu objetivo. Ele funciona como um servidor DNS e todas as requisições de resolução de nomes de domínio passam por ele. A partir dessas requisições é realizada a filtragem de acordo com a política de restrições estabelecida.

Como ele funciona como o próprio servidor DNS, possivelmente não causa aumento na latência de rede. Além disso, ele possui cache de respostas DNS, autenticação, integração com Active Directory, filtragem remota e outros recursos a mais (NxFilter, 2020, online).

O Nxfilter possui toda a sua configuração em uma interface WEB limpa e intuitiva, razões para sua escolha neste trabalho.

3.2.2.5 Namebench

O namebench nasceu de um projeto do Google para oferecer uma ferramenta de *benchmark* de DNS, apoiando na escolha do servidor DNS mais eficiente a partir do computador que esteja executando a ferramenta. O namebench utiliza testes de conexão com vários serviços DNS, além de informações disponíveis no próprio computador, como o histórico de navegação WEB e saída do tcpdump (Google, 2020, online).

Pelo namebench, também é possível testar a eficiência de um determinado DNS. O usuário digita o endereço do servidor e a ferramenta analisa o desempenho do DNS informado. A partir da utilização do namebench, foi possível realizar os testes de DNS, necessários para este trabalho.

3.2.2.6 MTR

O MTR é uma ferramenta que realiza a combinação das funcionalidades dos utilitários PING e TRACEROUTE para facilitar o diagnóstico de rede. O MTR investiga a conexão realizada entre o host solicitante e o host de destino, coletando o endereço de cada salto entre as máquinas, avaliando a qualidade da conexão, e disponibilizando as informações e estatísticas sobre a conexão (BitWizard, 2020, online).

Para realizar os testes de latência de conexão, o MTR se mostrou uma ferramenta adequada, sendo então utilizada neste trabalho.

3.3 Metodologia do Desenvolvimento

Considerando a necessidade de avaliar a efetividade da implantação de dois métodos diferentes de filtragem WEB num ambiente de rede, diferentes cenários de rede precisaram ser utilizados para que os dados fossem devidamente coletados para posterior comparação.

A partir dessa configuração padrão, foram criados quatro cenários de rede, sendo eles: sem filtragem WEB; filtragem WEB com *proxy* HTTP Squid; filtragem WEB com filtro DNS NxFilter; e filtragem WEB com *proxy* HTTP Squid e Filtro DNS NxFilter.

Com o intuito de observar o impacto real causado pelo controle de acesso à WEB no desempenho e uso da rede e suas conexões, foram realizadas análises de rede utilizando as ferramentas *namebench*, para coletar informações sobre o serviço DNS e MTR, para coletar dados sobre a latência da rede.

As ferramentas de coleta de dados foram instaladas no cliente de rede, dispositivo utilizado para realizar as requisições WEB. As coletas de dados de latência ocorreram em duas situações nos cenários de rede: com e sem fluxo contínuo de dados entre a rede interna e a internet. As situações em questão foram definidas com o intuito de avaliar se a carga de rede pode influenciar no desempenho dos serviços de filtragem de rede.

Após a realização da análise da latência da rede, para observar a efetividade da filtragem WEB nos cenários de rede desenvolvidos, foram realizados testes de controle de acesso. Estes testes foram necessários para, a partir de situações reais, verificar se o controle

de acesso se comportou conforme o desejado. Os testes utilizados para verificar o controle de acesso realizados pelos filtros foram:

- teste de requisição com nome de domínio: as requisições realizadas foram feitas para um nome de domínio válido (www.google.com, por exemplo). Neste teste foi verificado se a requisição chegou ao seu destino ou se foi interceptada por um sistema de filtragem WEB;
- teste de requisição com endereço IP: as requisições foram realizadas utilizando um endereço IP de um servidor WEB na internet (191.31.170.115, por exemplo). Neste teste foi verificado se a requisição chegou ao seu destino ou se foi interceptada por um sistema de filtragem WEB.
- teste de requisição HTTP: as requisições foram realizadas utilizando o protocolo HTTP para o acesso às páginas *web* (utiliza-se o termo "http://" no início do endereço a ser requisitado). Neste teste foi verificado se a requisição chegou ao seu destino ou se foi interceptada por um sistema de filtragem WEB;
- teste de requisição HTTPS: as requisições são realizadas utilizando o protocolo HTTPS para o acesso às páginas *web* (utiliza-se o termo "https://" no início do endereço a ser requisitado). Neste teste foi verificado se a requisição chegou ao seu destino ou se foi interceptada por um sistema de filtragem WEB;
- teste de requisição em portas de rede distintas: as requisições foram realizadas utilizando endereços com portas de rede diferentes da porta 80, padrão do HTTP, e da porta 443, padrão do HTTPS (www.teste.com:1234, por exemplo). Neste teste foi verificado se a requisição chegou ao seu destino ou se foi interceptada por um sistema de filtragem WEB.

Após a coleta das informações necessárias foi realizado o paralelo dos resultados obtidos durante o desenvolvimento do trabalho. Neste paralelo, foi considerado o processo de implantação da filtragem WEB nos cenários de rede, os resultados das análises de latência da rede e de resposta do DNS e, por fim, os resultados dos testes de controle de acesso. No paralelo realizado, as informações coletadas foram comparadas, considerando cada ferramenta de filtragem WEB, com o objetivo de apontar qual a melhor alternativa em cada aspecto observado.

RESULTADOS E DISCUSSÃO

Com a definição dos materiais a serem executados e os métodos a serem seguidos, foi possível realizar a aplicação prática do trabalho, onde os resultados da filtragem WEB são demonstrados e discutidos.

Nesta seção é apresentada a implantação dos cenários de rede, a análise de latência da rede, a análise de resposta do DNS, os testes de controle de acesso e, por fim, um paralelo das ferramentas de filtragem WEB, a partir das informações obtidas com os resultados encontrados. Na seção a seguir os cenários são desenvolvidos.

4.1 Cenários de Rede

Nos cenários de rede com filtragem WEB, as ferramentas foram instaladas da seguinte forma:

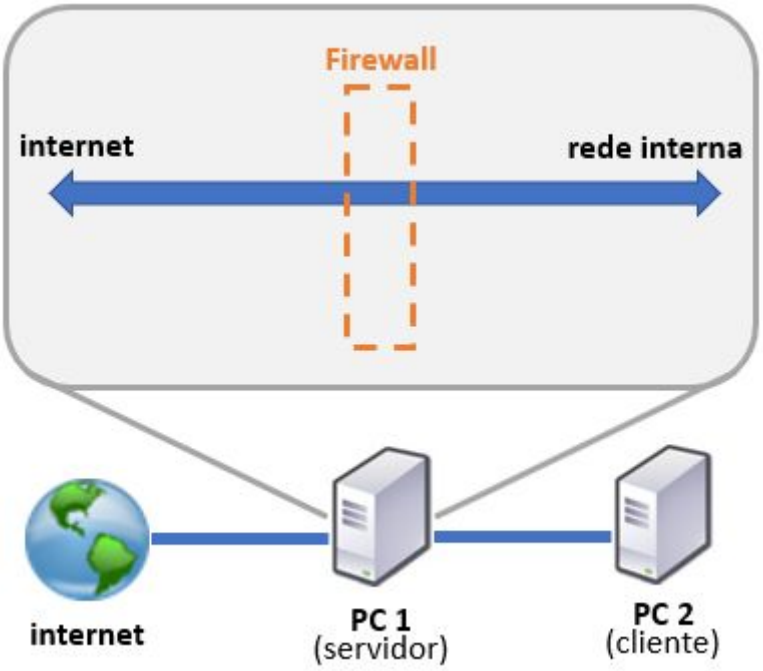
- *proxy* HTTP Squid: instalado como um pacote adicional do PfSense, com as configurações básicas para filtragem WEB em HTTP e HTTPS, considerando a utilização de certificado SSL, confirmando a interceptação através do *log* da ferramenta e sem utilização de cache;
- filtro DNS NxFilter: instalado por meio de um script disponibilizado no site da ferramenta, com as configurações básicas para filtragem WEB em HTTP e HTTPS, considerando a utilização de certificado SSL, confirmando a interceptação através do *log* da ferramenta e sem utilização de cache.

Nas seções a seguir, são apresentadas as características de cada cenário implantado.

4.1.1 Cenário 1 - sem controle de acesso à WEB

No primeiro cenário implantado não houve a instalação de ferramenta para realizar o controle de acesso à WEB. O servidor de rede funcionou exclusivamente como um gateway de rede, apenas disponibilizando o acesso à internet para a rede interna. O cliente da rede pôde acessar a internet de maneira livre, sem quaisquer bloqueios aplicados. O cenário 1 é demonstrado na Figura 9, a seguir.

Figura 9: Demonstração do cenário 1 - sem controle de acesso à WEB.



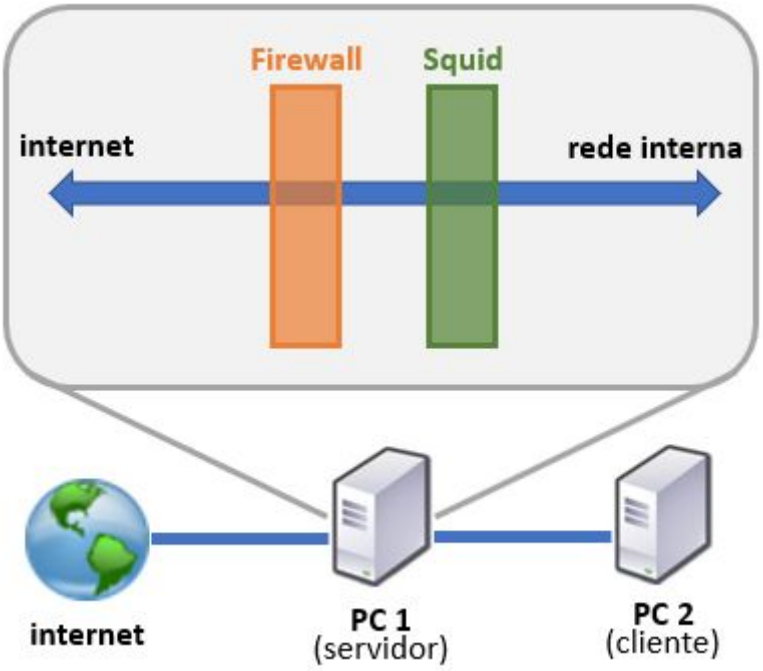
No cenário 1, o *firewall* foi configurado apenas para disponibilizar o serviço de *gateway* de rede, que tem por objetivo criar uma conexão entre a rede interna e externa. O *firewall* foi configurado sem quaisquer restrições ou redirecionamentos. As requisições WEB puderam transitar livremente entre o cliente e o servidor WEB de destino. O cliente utilizou o serviço de DNS do roteador, sem quaisquer restrições.

Este cenário de rede foi importante para o trabalho pois, na aplicação dos testes, foi possível coletar os dados de controle de acesso e desempenho da rede sem nenhum tipo de interferência na conexão. Todos os demais cenários, apresentados nas próximas seções, possuíam meios de controle de acesso.

4.1.2 Cenário 2 - controle de acesso com *proxy* HTTP Squid

No segundo cenário de rede implantado foi utilizada a ferramenta Squid para filtragem WEB. O Squid foi devidamente instalado e configurado no servidor de rede, para que pudesse executar a função de controle de acesso. A Figura 10 demonstra o cenário 2, a seguir.

Figura 10: Demonstração do cenário 2 - controle de acesso com *proxy* HTTP Squid.



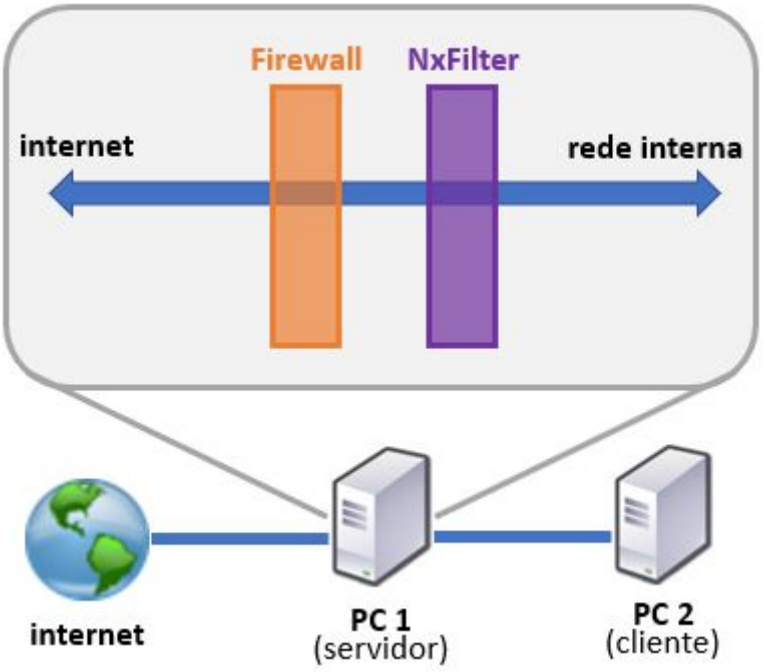
No servidor de rede, o Squid foi configurado para uso como *proxy* transparente. Para que isso fosse possível, foi realizada a configuração do *firewall* para que o tráfego do protocolo HTTP do servidor de rede na porta 80 fosse redirecionado para a porta padrão de uso do Squid, a 3128. O serviço de DNS utilizado foi o do roteador, sem quaisquer funções de filtragem.

Apesar do Squid também possuir a possibilidade de ser usado como *proxy* não transparente, ele não foi utilizado dessa forma, pois o NxFilter não possui a função de filtro não transparente, o que inviabilizou um comparativo das duas ferramentas nessa modalidade.

4.1.3 Cenário 3 - controle de acesso com filtro DNS NxFilter

O cenário 3 foi implantado utilizando o NxFilter para realizar a filtragem WEB. A ferramenta em questão foi instalada no servidor de rede, intermediário no acesso à internet. Assim como o Squid, o NxFilter também necessitou de auxílio do *firewall* para que a filtragem fosse realizada. A Figura 11 facilita o entendimento do funcionamento do cenário 3.

Figura 11: Demonstração do cenário 3 - controle de acesso com filtro DNS NxFilter.



Como o NxFilter funciona como um servidor DNS, o cliente de rede precisou utilizar o serviço de DNS do NxFilter para que a filtragem WEB fosse realizada. Para isso duas configurações foram necessárias:

- no servidor de rede, o *firewall* foi configurado de forma a bloquear a porta de rede padrão do serviço de DNS, a porta 53, de qualquer tráfego advindo da rede interna com destino externo à rede. Dessa forma, o cliente de rede não conseguia utilizar nenhum serviço de DNS que não estivesse na própria rede, sendo obrigado a usar o DNS do NxFilter;
- o endereço de rede do serviço de DNS do NxFilter foi atribuído para o cliente de rede através do serviço de DHCP do PfSense.

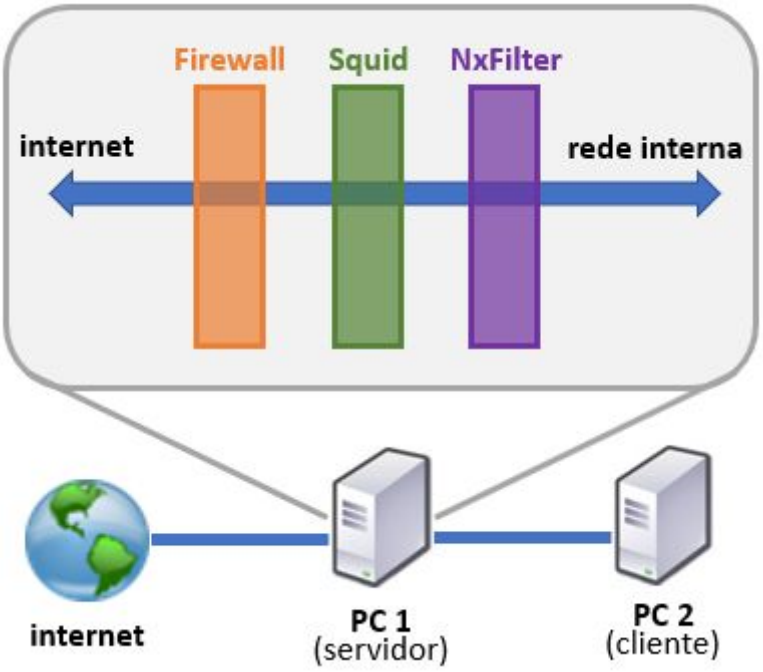
O fato de o NxFilter funcionar como um servidor DNS local, o faz depender de um outro servidor DNS disponível na WEB para que ele redirecione as solicitações, após a realização da filtragem WEB. Para atender a essa configuração, foi utilizado o mesmo servidor que o serviço de DNS do modem utiliza para redirecionar as consultas.

4.1.4 Cenário 4 - controle de acesso com *proxy* HTTP Squid e Filtro DNS NxFilter

O quarto cenário de rede utilizado para compor este trabalho foi implantado com a utilização do *proxy* HTTP Squid e do NxFilter em conjunto. Assim como nos cenários 2 e 3,

as ferramentas de filtragem WEB foram instaladas no servidor de rede, com as configurações básicas necessárias. A Figura 12, a seguir, apresenta o cenário 4.

Figura 12: Demonstração do cenário 4 - controle de acesso com *proxy* HTTP Squid e filtro DNS NxFilter.



O Squid foi utilizado como *proxy* transparente, de forma que o *firewall* realizou o redirecionamento da porta 80. O NxFilter foi utilizado como o serviço DNS da rede interna e o *firewall* realizou o bloqueio da porta 53 para forçar o uso do servidor DNS local.

4.2 Latência da Rede com MTR

Nos quatro cenários, a ferramenta MTR enviou cem mensagens simples para um destino específico, o qual respondeu essas mensagens. Durante esse processo, foi contabilizado o tempo entre o envio das mensagens e a chegada das respostas, que é a latência da rede.

Uma mensagem enviada de um dispositivo da rede interna com destino a um servidor na internet percorre um caminho repleto de conexões, em que cada ponto de conexão é chamado de *gateway* de rede. O MTR disponibilizou, junto com a latência da rede, o caminho pelo qual a mensagem percorreu até chegar ao seu destino e o caminho da resposta até o dispositivo solicitante, ou seja, exibiu os *gateways* pelos quais a mensagem e a resposta

passaram. Essa informação foi importante no processo de análise pois, se em algum teste, por algum motivo o caminho for diferente, naturalmente a latência também sofrerá algum tipo de alteração.

Duas situações foram definidas para analisar o desempenho do tráfego rede, sendo a primeira sem fluxo contínuo de rede e a segunda com fluxo contínuo de rede. As duas situações são descritas a seguir:

- Na primeira situação, o cliente de rede utiliza tráfego mínimo com a internet, sem qualquer aplicação ou serviço específico demandando o uso da rede para tráfego contínuo. Nessa situação não há grande exigência de tráfego pelo servidor de rede;
- Na segunda situação, o cliente de rede está com o navegador WEB em execução, com duas páginas abertas, reproduzindo uma transmissão ao vivo de vídeo na plataforma de compartilhamento de vídeos YouTube. Os dois vídeos possuem alta resolução (1920x1080 pixels). Nessa situação, há um fluxo contínuo de tráfego sendo solicitado pelo cliente de rede e transmitido através do servidor de rede.

Para a coleta de dados de latência da rede com o MTR, os seguintes destinos foram definidos:

- Servidor do Google, com o endereço WEB www.google.com;
- Serviço de *gateway* de rede imediatamente após o *gateway* do servidor de rede, ou seja, o modem de conexão à internet utilizado pelo notebook em que os cenários virtualizados estavam, com o endereço IP 192.168.100.1.

A escolha do servidor do Google para a coleta de dados se deu para que, através da análise, se avaliasse a latência em um destino distante, que percorria vários gateways para a entrega da solicitação de mensagem e o recebimento da resposta. Já a escolha do *gateway* imediatamente após o servidor de rede se deu para coletar dados de latência no percurso mais curto possível, em que o único *gateway* percorrido entre a origem e o destino seja o do próprio servidor de rede.

A coleta de dados foi realizada em cada cenário de rede, considerando as situações definidas e os destinos anteriormente citados. A tabela 1 demonstra a coleta dos dados de latência da rede.

Tabela 1: Definição da coleta de dados de latência da rede com a ferramenta MTR.

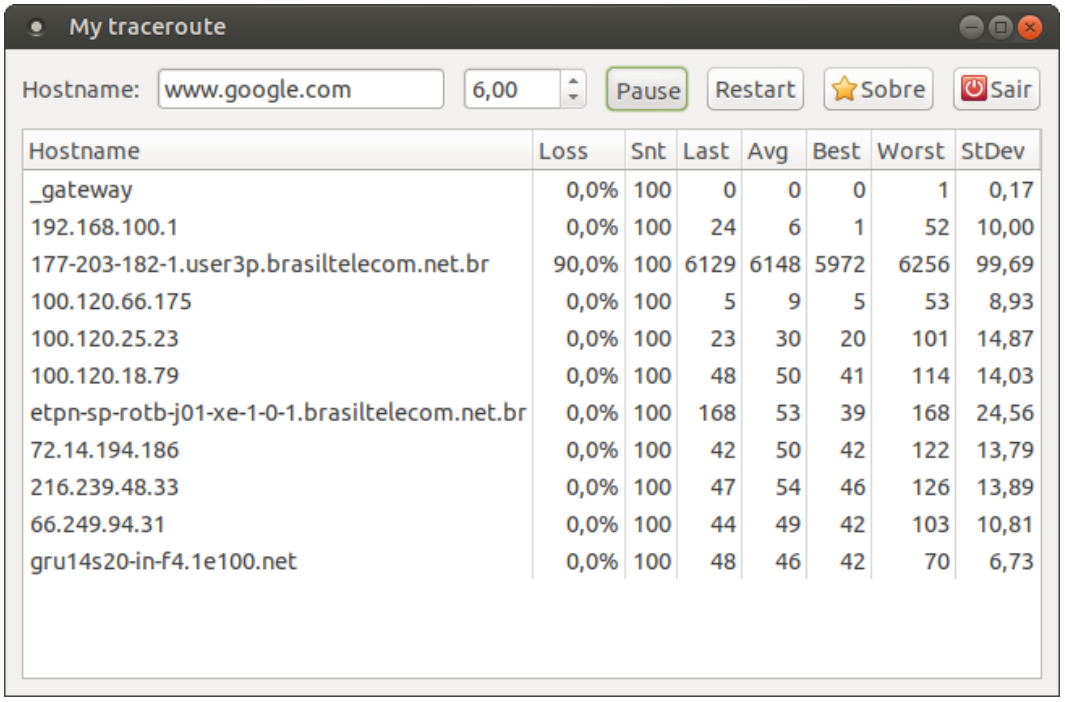
Nº	CENÁRIO	SITUAÇÃO	DESTINO
01	1 – Sem filtragem WEB	Sem fluxo contínuo	www.google.com
02	1 – Sem filtragem WEB	Com fluxo contínuo	www.google.com
03	1 – Sem filtragem WEB	Sem fluxo contínuo	192.168.100.1
04	1 – Sem filtragem WEB	Com fluxo contínuo	192.168.100.1
05	2 – Filtragem WEB com proxy HTTP Squid	Sem fluxo contínuo	www.google.com
06	2 – Filtragem WEB com proxy HTTP Squid	Com fluxo contínuo	www.google.com
07	2 – Filtragem WEB com proxy HTTP Squid	Sem fluxo contínuo	192.168.100.1
08	2 – Filtragem WEB com proxy HTTP Squid	Com fluxo contínuo	192.168.100.1
09	3 – Filtragem WEB com filtro DNS NxFilter	Sem fluxo contínuo	www.google.com
10	3 – Filtragem WEB com filtro DNS NxFilter	Com fluxo contínuo	www.google.com
11	3 – Filtragem WEB com filtro DNS NxFilter	Sem fluxo contínuo	192.168.100.1
12	3 – Filtragem WEB com filtro DNS NxFilter	Com fluxo contínuo	192.168.100.1
13	4 – Filtragem WEB com proxy HTTP Squid e Filtro DNS NxFilter	Sem fluxo contínuo	www.google.com
14	4 – Filtragem WEB com proxy HTTP Squid e Filtro DNS NxFilter	Com fluxo contínuo	www.google.com
15	4 – Filtragem WEB com proxy HTTP Squid e Filtro DNS NxFilter	Sem fluxo contínuo	192.168.100.1
16	4 – Filtragem WEB com proxy HTTP Squid e Filtro DNS NxFilter	Com fluxo contínuo	192.168.100.1

Após a coleta dos dados, obteve-se material suficiente para comparação de desempenho de latência e verificação da possibilidade de influência da filtragem WEB nesse desempenho. A seção a seguir apresenta a análise de latência da rede.

4.2.1 Análise de Latência da Rede

A partir da coleta de dados com o MTR, foi possível levantar as informações de latência em cada cenário, considerando as situações colocadas. A seguir, na Figura 13, é possível observar o resultado da coleta de dados de latência utilizando o servidor do Google como destino, no cenário 1, sem filtragem WEB e sem fluxo de rede contínuo.

Figura 13: Resultado da coleta de dados de latência utilizando a ferramenta MTR no cenário 1.



Hostname	Loss	Snt	Last	Avg	Best	Worst	StDev
_gateway	0,0%	100	0	0	0	1	0,17
192.168.100.1	0,0%	100	24	6	1	52	10,00
177-203-182-1.user3p.brasiltelecom.net.br	90,0%	100	6129	6148	5972	6256	99,69
100.120.66.175	0,0%	100	5	9	5	53	8,93
100.120.25.23	0,0%	100	23	30	20	101	14,87
100.120.18.79	0,0%	100	48	50	41	114	14,03
etpn-sp-rotb-j01-xe-1-0-1.brasiltelecom.net.br	0,0%	100	168	53	39	168	24,56
72.14.194.186	0,0%	100	42	50	42	122	13,79
216.239.48.33	0,0%	100	47	54	46	126	13,89
66.249.94.31	0,0%	100	44	49	42	103	10,81
gru14s20-in-f4.1e100.net	0,0%	100	48	46	42	70	6,73

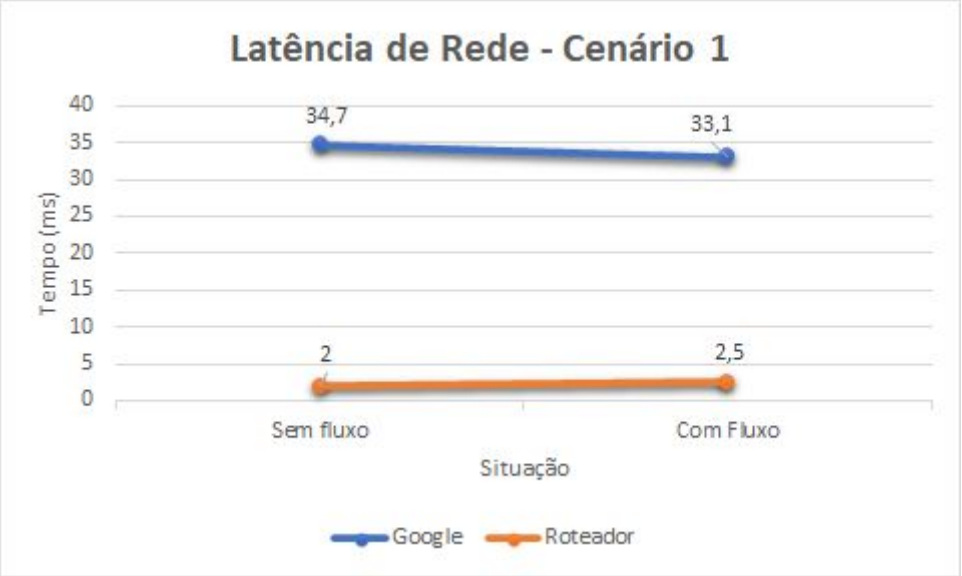
Como mostra a Figura 13, os pacotes transitaram entre origem e destino sem perdas de pacotes, o que garante a estabilidade da conexão. A perda de pacotes pode ser observada na coluna com o título "*Loss*", que exibe a porcentagem de pacotes perdidos. A exceção encontrada na coleta se refere ao terceiro gateway (177-203-182-1.user3p.brasiltelecom.net.br), porém, era um resultado esperado, tendo em vista que se trata da representação do *Internet Service Provider*, (ISP), o provedor de serviços de internet. Esse resultado é esperado no ISP pois nele, possivelmente, é realizada a limitação de largura de banda, o que faz com que o MTR traduza esse mecanismo como uma perda de pacote. O ISP é apresentado no teste com perda de pacote, mas não influencia no resultado final.

A coleta de dados com o MTR demonstrada na Figura 13 foi realizada da mesma forma em todos os cenários de rede, obtendo o mesmo tipo de retorno, apenas com valores diferentes, o que dispensa a necessidade de inserir no projeto todas as capturas de tela. Em vez disso, gráficos foram gerados com os resultados obtidos nas coletas de dados e são disponibilizados no decorrer da seção.

Nas análises realizadas a partir do retorno do o MTR, para observar a latência da rede, foram utilizadas as informações constantes na coluna com o título "Avg", que se trata da média aritmética de tempo entre os saltos em milissegundos. Para se chegar a um resultado único para comparação, foi realizada a média aritmética de todos os valores constantes na coluna "Avg", com exceção do resultado referente ao ISP. Com essa informação foi possível

criar os gráficos para análise e, ao final, a discussão sobre os resultados. Na figura 14 é apresentado o gráfico para análise de latência da rede no cenário 1, sem filtragem WEB.

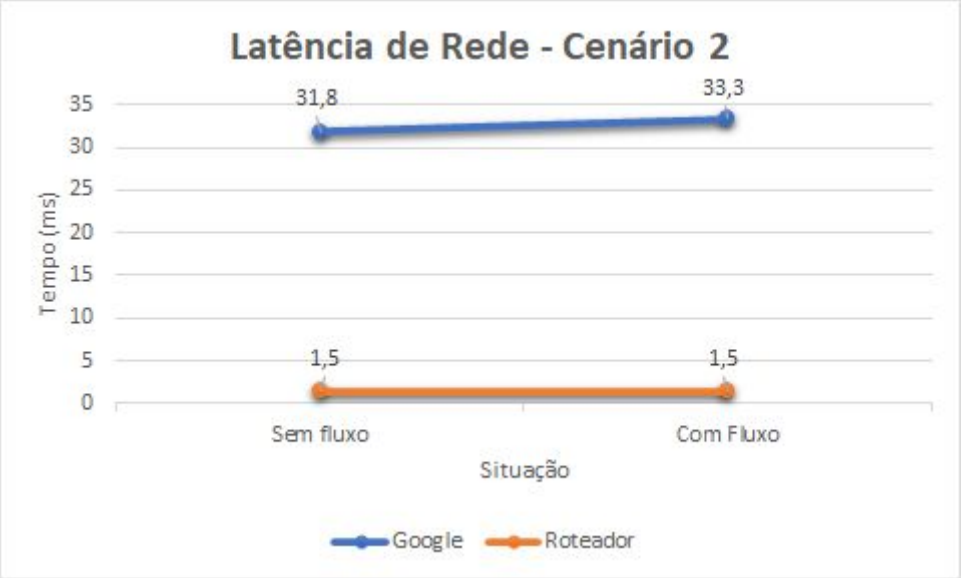
Figura 14: Gráfico de latência da rede no cenário 1.



Curiosamente, como pode ser observado na Figura 14, a latência no teste com o servidor do Google enquanto havia fluxo de rede é menor do que enquanto não havia fluxo. Porém, a situação se inverte quando a latência é analisada entre cliente de rede e o roteador, onde o caminho percorrido é menor. Esse resultado de latência da rede em questão não possui nenhum tipo de influência de filtro WEB, devido à configuração do cenário 1.

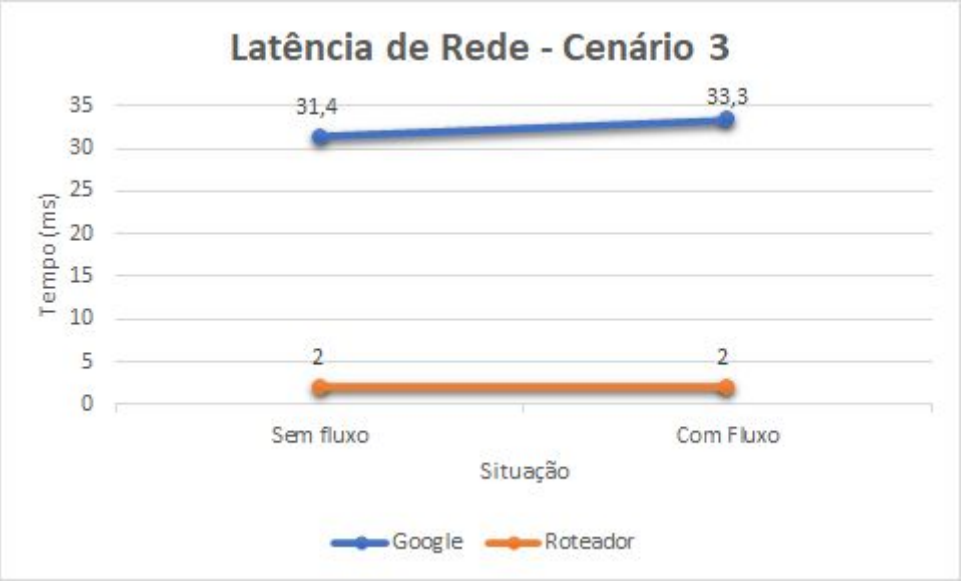
A seguir, na Figura 15, são exibidas as informações para a análise de latência do cenário 2, com filtragem WEB utilizando o Squid.

Figura 15: Gráfico de latência da rede no cenário 2.



Considerando as informações contidas no gráfico da Figura 15, observa-se um ligeiro aumento na latência da rede quando o MTR visou alcançar o servidor do Google quando havia fluxo contínuo de rede. Já a latência com o roteador como alvo permaneceu a mesma. Em seguida, na Figura 16, podemos observar os resultados de latência no cenário 3, onde a filtragem WEB foi realizada com o NxFilter.

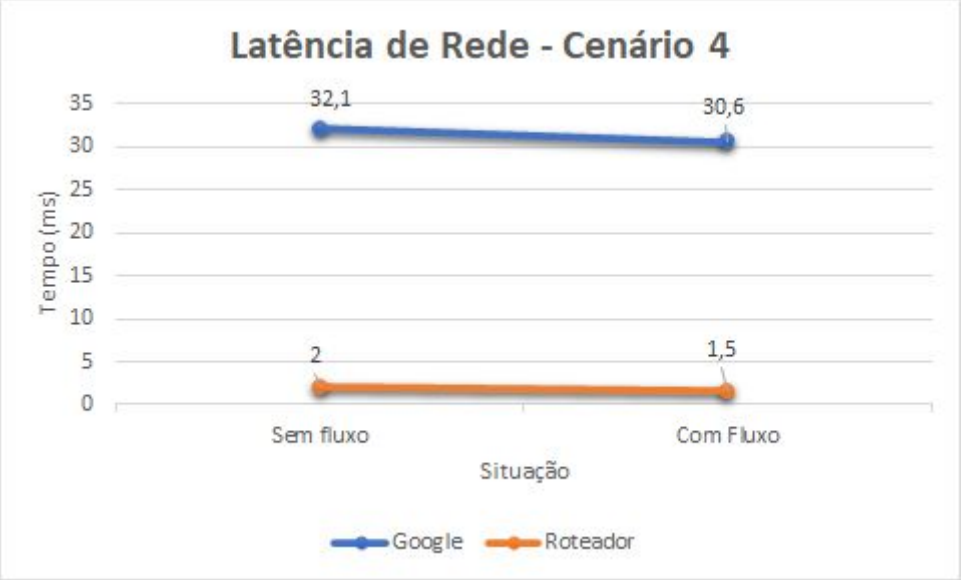
Figura 16: Gráfico de latência da rede no cenário 3.



Assim como no cenário 2, onde a filtragem WEB foi realizada utilizando o Squid, no cenário 3, com a filtragem WEB pelo NxFilter, também houve um ligeiro aumento na latência da rede quando o destino a ser alcançado pelo MTR foi o servidor do Google, enquanto havia

fluxo contínuo de rede. Na coleta realizada com o roteador como destino, também não houve alteração na latência, com ou sem fluxo de rede. Na Figura 17 é apresentado o gráfico para análise da latência da rede do cenário 4, onde a filtragem WEB foi realizada com o Squid e com o NxFilter em conjunto.

Figura 17: Gráfico de latência da rede no cenário 4.



Como é possível observar na Figura 17, assim como na Figura 14, houve uma ligeira queda na latência da rede quando o alvo a alcançar foi o servidor do Google na situação de fluxo contínuo, em relação à situação de não haver fluxo. No cenário 4 em específico, a redução de latência também pôde ser observada quando o destino a ser alcançado foi o roteador, entre as situações sem e com fluxo de rede.

Em posse de todas as informações de análise de latência da rede nos quatro cenários do trabalho, é possível chegar às seguintes conclusões:

- a latência da rede não sofreu alterações significativas se comparadas situações de análise com fluxo e sem fluxo de rede;
- a fato de a latência sofrer uma ligeira queda nos cenários 1 e 4, na situação de fluxo de rede contínuo quando o alvo a ser alcançado foi o servidor do Google, demonstra que a filtragem WEB com *proxy* HTTP e com filtro DNS pode não influenciar significativamente na latência quando há um volume de dados relativamente alto e constante na rede.

- A alteração na latência da rede observada de maneira geral nos 4 cenários, provavelmente possui influência na própria oscilação natural da rede em si, do que propriamente na filtragem WEB.

É importante observar que fluxo contínuo de rede não possui relação direta com quantidade de requisições. A partir do momento em que uma requisição é realizada e obtém uma resposta do servidor, pode ser criada uma conexão com fluxo contínuo de dados onde este fluxo não necessita demandar requisições constantes.

Além da latência da rede, outro aspecto importante que foi analisado é a resposta do serviço DNS utilizado na rede.

4.3 Consulta DNS com namebench

O DNS é primordial para a conexão com a internet e a utilização dos serviços disponíveis via WEB. Como já citado no referencial teórico, este serviço indica o endereço IP do servidor a partir de um domínio de rede. Como o NxFilter utiliza o serviço DNS para realizar a filtragem WEB, foi importante observar se, após a aplicação da filtragem, houve alguma redução de desempenho na consulta ao servidor DNS.

Aproveitando o teste de DNS em função do NxFilter, tornou-se interessante analisar se o Squid realizando filtragem WEB também impactou de alguma forma o desempenho da consulta do DNS, para fins de comparação.

Para coletar os resultados, o namebench fez 250 consultas em cada servidor DNS. A partir da rede interna e, baseado na região e conexões, o *software* procurou na internet o DNS que ofereceu o menor tempo de consulta entre os servidores utilizados no trabalho e os principais servidores públicos disponíveis na internet. Baseadas nas questões de desempenho e na possibilidade de encontrar o DNS mais eficiente para a rede, as coletas de dados foram realizadas da seguinte forma:

- nos cenários onde a filtragem WEB não estava sendo realizada pelo NxFilter, as consultas foram realizadas no servidor DNS fornecido pelo modem (192.168.100.1), que redirecionava as consultas para o DNS da operadora de rede que disponibilizava a internet (201.10.128.3) e, também, pela busca da melhor alternativa de DNS público na internet pelo namebench;
- nos cenários onde a filtragem WEB estava sendo realizada pelo NxFilter, as consultas foram realizadas no próprio DNS do NxFilter (192.168.1.1), configurado para redirecionar a consultas para o servidor DNS da operadora de rede que disponibilizava

a internet (201.10.128.3). As consultas também foram realizadas no DNS do modem (192.168.100.1), que redirecionava as consultas para a operadora de rede que disponibilizava a internet (201.10.128.3) e pela busca da melhor alternativa de DNS pelo namebench.

Seguindo as definições acima citadas, os dados coletados pelo namebench foram utilizados para analisar o tempo de consulta dos servidores DNS nos quatro cenários de rede. Na seção a seguir a análise é apresentada.

4.3.1 Resultados da Análise de consulta DNS

Após cada execução, a ferramenta namebench gera uma página WEB com todas as informações coletadas. Entre estas informações, foram escolhidas para a análise:

- melhor servidor DNS para a rede, com o status “Fastest”;
- parte de tabela de resultados, com serviços DNS que ocuparam as três primeiras posições, e as seguintes colunas: “IP”, que identificava o endereço IP do servidor DNS; “Descr”, que continha uma breve descrição do DNS; “*Hostname”,* que exibia o nome do servidor; com o título “AVG (ms)”, que informava o tempo médio de consulta ao DNS; “Diff”, que apresentava a porcentagem de tempo a mais da consulta ao DNS em relação ao tempo médio da consulta ao DNS com menor tempo de consulta;
- gráfico comparativo da média aritmética do tempo de consulta de todos os servidores DNS testados.

O namebench gera mais informações além das que foram selecionadas, porém para a análise proposta por este trabalho, as informações selecionadas já foram suficientes. A seguir, na Figura 18, é possível observar os resultados da análise de DNS no cenário 1, onde não há filtragem WEB.

Figura 18: Resultados da coleta de dados de consulta aos servidores DNS no cenário 1.

Observando os resultados demonstrados na Figura 18, o servidor DNS do modem, que redireciona para o servidor DNS da operadora, obteve o menor tempo de consulta pelo namebench. Em seguida, o DNS público do Google ocupa o segundo e o terceiro lugar, com aumento de 50% e 54,7% no tempo de consulta em relação ao primeiro lugar, respectivamente. Sendo assim, pode-se considerar, obviamente, que o DNS do roteador é o ideal para a rede em questão.

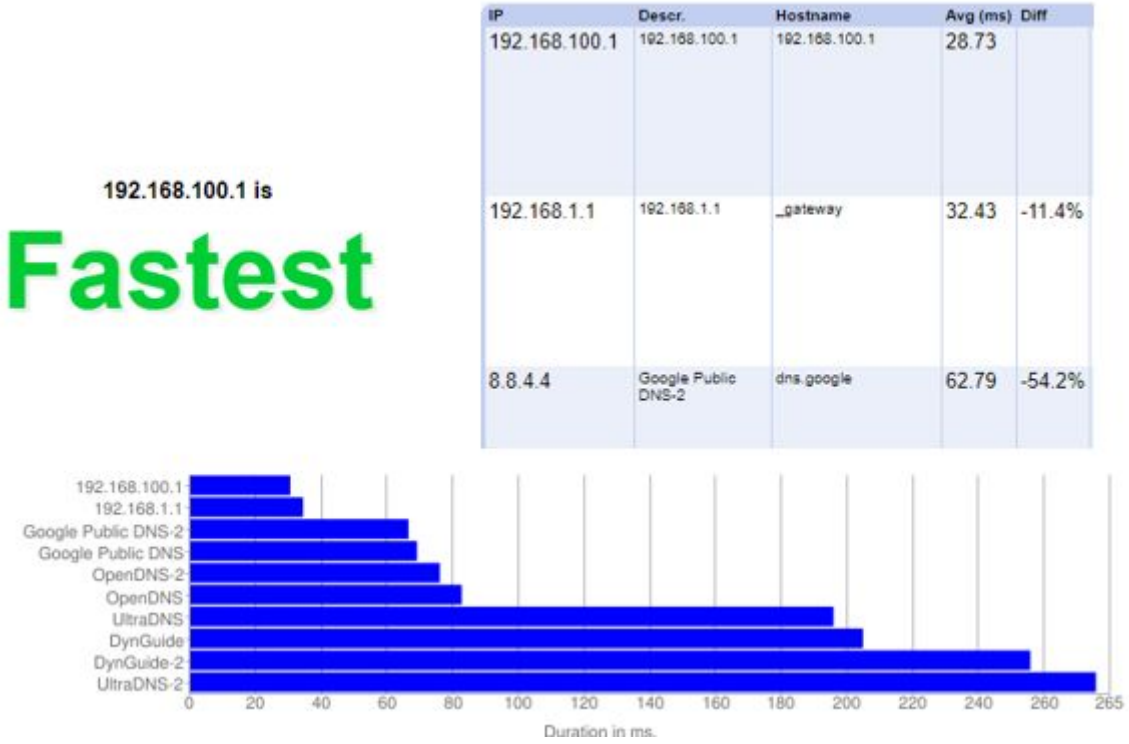
O cenário de rede analisado não possui filtragem WEB, assim como não possui servidor DNS local, ou seja, não há nenhum tipo de recurso que possa influenciar no resultado em si. Já no cenário 2, onde há filtragem WEB utilizando o Squid, é interessante observar se o intermediário das requisições WEB realizava alguma influência no tempo de consulta do DNS. A Figura 19 demonstra os resultados da coleta de dados no cenário 2.

Figura 19: Resultados da coleta de dados de consultas aos servidores DNS no cenário 2.

Analisando as informações contidas na Figura 19, percebe-se que, no cenário 2, onde há filtragem WEB com Squid, o servidor DNS do modem foi a melhor alternativa de escolha de DNS, devido seu tempo de consulta ser menor. Já o segundo e o terceiro lugares foram ocupados pelos servidores DNS do Google, com o tempo de consulta aumentado em 52,3 % e 55,3% em relação ao servidor DNS do modem, respectivamente.

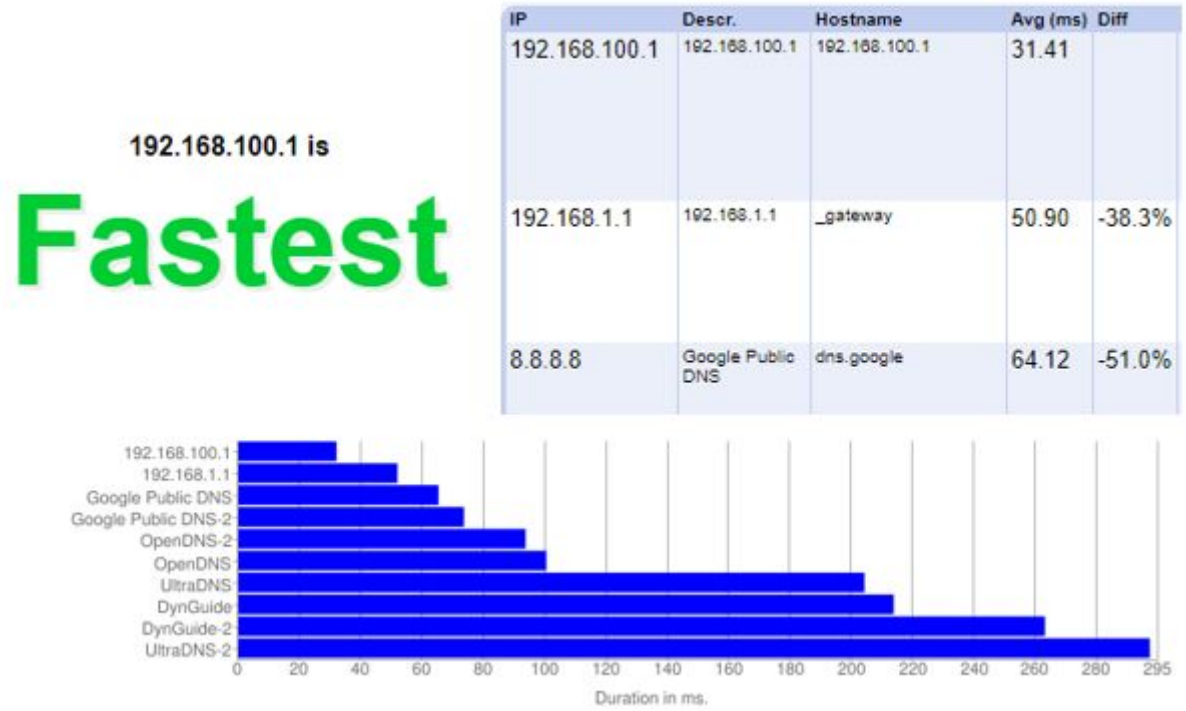
Os cenários analisados até então não continham um servidor de DNS local em uso. As próximas coletas de dados demonstradas foram realizadas em cenários onde existia um servidor DNS local, sendo este o filtro DNS. A Figura 20 apresenta os resultados obtidos na coleta de dados do cenário 3, onde a filtragem WEB foi realizada utilizando o NxFilter.

Figura 20: Resultados da coleta de dados de consultas aos servidores DNS no cenário 3.



A partir dos resultados demonstrados na Figura 20, observa-se que o DNS com melhor resultado no tempo de resposta foi o DNS do modem. O servidor DNS do NxFilter ficou em segundo, sendo 11,4% menos rápido que o DNS do modem. O servidor DNS público do Google ficou em terceiro lugar, com aumento de 54,2% no tempo de consulta em relação ao primeiro lugar. Em seguida, a Figura 21 traz os resultados obtidos na coleta de dados do cenário 4, onde havia filtragem WEB com o Squid e com o NxFilter.

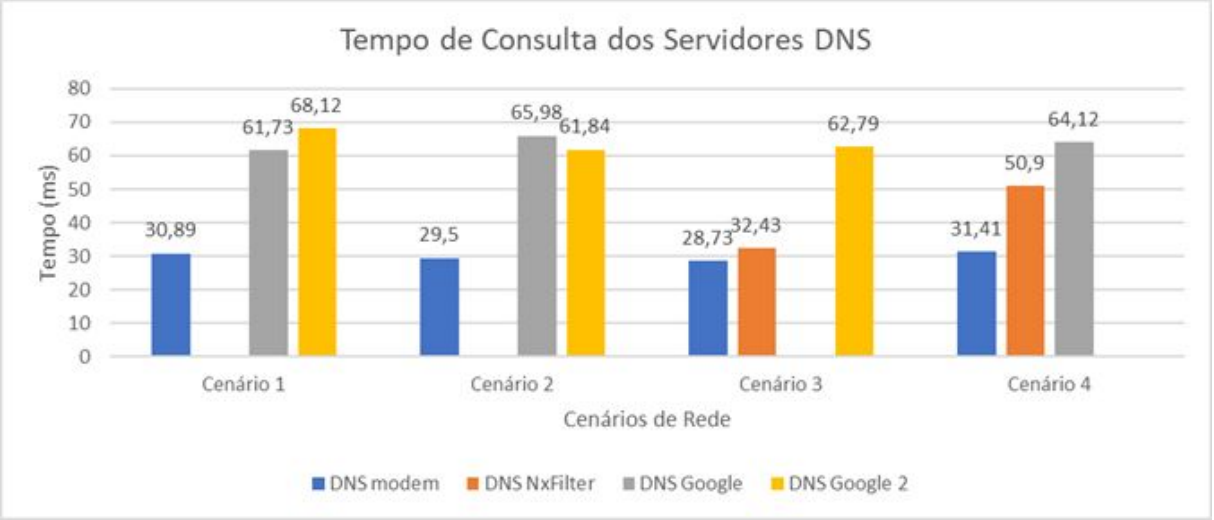
Figura 21: Resultados da coleta de dados de consultas aos servidores DNS no cenário 4.



Observando os resultados obtidos na coleta de dados realizada no cenário 4, nota-se que em primeiro lugar estava o servidor DNS do modem. Em segundo lugar se posicionou o DNS do NxFilter, com o tempo de consulta 38,3% maior em relação ao primeiro lugar e, em terceiro na posição, o DNS público do Google, que foi 51% menos eficiente no tempo de consulta que o primeiro colocado.

Baseado no resultado da coleta de dados das consultas de DNS nos quatro cenários de rede do trabalho, foi gerado um gráfico, para auxiliar melhor na comparação dos valores de tempo de resposta dos três primeiros servidores DNS classificados. A Figura 22 a seguir apresenta o gráfico em questão.

Figura 22: Gráfico de tempo de resposta dos três servidores DNS mais bem colocados no tempo de consulta DNS nos cenários de rede.



A partir das informações coletadas e do gráfico da Figura 22 com os tempos de consulta dos servidores DNS em cada cenário de rede, é possível chegar às seguintes conclusões:

- qualquer alteração substancial nos tempos de consulta dos servidores DNS ocorreram exclusivamente nos servidores DNS presentes na rede interna. Na rede externa (modem e internet), não houve alteração significativa;
- em todos os cenários de rede, o servidor DNS do modem obteve os menores tempo de consulta. Sendo assim, pode-se afirmar que o DNS do modem é a melhor alternativa de DNS para a rede, considerando sua eficiência em detrimento dos demais servidores DNS analisados;
- nos cenários 3 e 4, onde o servidor DNS do NxFilter foi analisado, ele obteve o segundo lugar, atrás do servidor DNS do modem. Com isso, pode-se inferir que o serviço de filtragem WEB NxFilter aumenta o tempo de consulta do DNS, considerando os serviços DNS externos à rede. É importante levar em consideração que tanto o NxFilter quanto o modem redirecionavam suas requisições para o mesmo serviço DNS;
- considerando que o modem é o meio pelo qual o servidor de rede tem acesso à internet, faz sentido o servidor DNS do modem possuir tempo de consulta menor que o servidor DNS do NxFilter. Enquanto o servidor DNS do modem é o próprio *gateway* de acesso à internet e o cliente já recorre diretamente a ele para a requisição, o servidor DNS do NxFilter é mais um ponto adicionado ao caminho do serviço, que necessita utilizar o *gateway* do modem para ter acesso ao servidor DNS externo;

- a filtragem WEB realizada pelo Squid não influenciou no tempo de consulta dos servidores DNS do modem e do Google. Porém, no cenário 4, pode-se observar um aumento substancial no tempo de resposta do servidor DNS do NxFilter e, neste cenário em questão, os dois modos de filtragem WEB estão operando em conjunto. Assim, é possível afirmar que a utilização do *proxy* HTTP Squid em conjunto com o NxFilter pode fazer com que o processo de filtragem WEB por filtro DNS perca desempenho.

É possível perceber, através dessa análise, que o tempo de resposta do DNS pode sofrer algum tipo de alteração em função da aplicação de filtragem WEB, especificamente no caso da utilização conjunta do Squid e do NxFilter. Na seção a seguir, são demonstrados os resultados dos testes de controle de acesso nos cenários de rede e, ao final, é realizado um paralelo entre as ferramentas de filtragem utilizando as informações coletadas durante o desenvolvimento do trabalho.

4.4 Testes de Controle de Acesso

Para observar o funcionamento do controle de acesso proporcionado pela filtragem WEB, os testes de controle de acesso foram aplicados nos quatro cenários de rede do trabalho. Foram escolhidos cinco nomes de domínio para que a ferramenta de filtragem os bloqueasse. Os nomes de domínio escolhidos foram:

- uol.com.br;
- terra.com.br;
- to.gov.br;
- globo.com;
- nxfilter.org.

Estes nomes de domínio foram incluídos na lista de bloqueio das ferramentas de filtragem WEB constantes nos cenários testados. Logo após, foi aberto um navegador WEB e uma série de requisições foram realizadas, com nomes de domínio incluídos na lista de bloqueio e nomes de domínio e endereços IP não incluídos nessa lista. Na Tabela 2 os nomes de domínio e endereços IP utilizados nas requisições são apresentados.

Tabela 2: Nomes de domínio e endereços IP utilizados nos testes de controle de acesso.

TESTE	ENDEREÇO DA REQUISIÇÃO
Com Nome de Dominio	google.com
	ufrgs.br
	uol.com.br
	terra.com.br
Com Endereço IP	216.58.195.78 (google.com)
	143.54.2.20 (ufrgs.br)
	200.147.3.157 (uol.com.br)
	186.192.90.12 (globo.com)
Com Requisição HTTP	ufrgs.br
	sipros.pa.gov.br
	terra.com.br
	esgepen.cidadaniaejustica.to.gov.br
Com Requisição HTTPS	facebook.com
	wikipedia.org
	to.gov.br
	g1.globo.com
Com Portas de Rede Distintas	esgepen.ddns.net:8080/share
	demo.nxfilter.org:4080/admin

Como pode ser visto na Tabela 2, para cada teste, metade das requisições foram realizadas com endereços não bloqueados e a outra metade com endereços bloqueados. Os resultados foram coletados e são demonstrados na seção a seguir.

4.4.1 Resultados dos Testes de Controle de Acesso

A partir do envio das requisições realizadas nos testes, elas percorriam o servidor de rede livremente, no caso do cenário 1, e eram interceptadas e bloqueadas ou liberadas nos cenários 2, 3 e 4. Na Tabela 3, a seguir, os resultados dos testes são apresentados.

Tabela 3: Resultados dos testes de controle de acesso nos quatro cenários de rede.

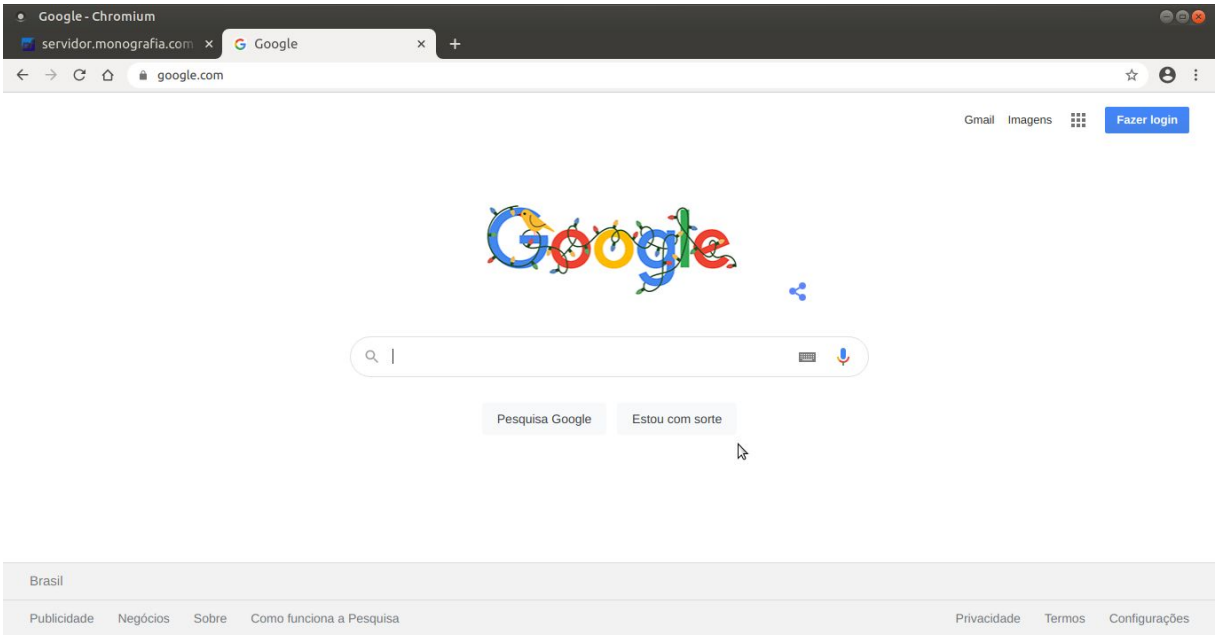
		RESULTADO POR CENÁRIO			
TESTE	ENDEREÇO DA REQUISIÇÃO	CENÁRIO 1	CENÁRIO 2	CENÁRIO 3	CENÁRIO 4
Com Nome de Dominio	google.com	Sem Filtro	Liberada	Liberada	Liberada
	ufrgs.br	Sem Filtro	Liberada	Liberada	Liberada
	uol.com.br	Sem Filtro	Bloqueada	Bloqueada	Bloqueada
	terra.com.br	Sem Filtro	Bloqueada	Bloqueada	Bloqueada
Com Endereço IP	216.58.195.78 (google.com)	Sem Filtro	Liberada	Liberada	Liberada
	143.54.2.20 (ufrgs.br)	Sem Filtro	Liberada	Liberada	Liberada
	200.147.3.157 (uol.com.br)	Sem Filtro	Bloqueada	Bloqueada	Bloqueada
	186.192.90.12 (globo.com)	Sem Filtro	Bloqueada	Bloqueada	Bloqueada
Com Requisição HTTP	ufrgs.br	Sem Filtro	Liberada	Liberada	Liberada
	sipros.pa.gov.br	Sem Filtro	Liberada	Liberada	Liberada
	terra.com.br	Sem Filtro	Bloqueada	Bloqueada	Bloqueada
	esgepen.cidadaniaejustica.to.gov.br	Sem Filtro	Bloqueada	Bloqueada	Bloqueada
Com Requisição HTTPS	facebook.com	Sem Filtro	Liberada	Liberada	Liberada
	wikipedia.org	Sem Filtro	Liberada	Liberada	Liberada
	to.gov.br	Sem Filtro	Bloqueada	Bloqueada	Bloqueada
	g1.globo.com	Sem Filtro	Bloqueada	Bloqueada	Bloqueada
Com Portas de Rede Distintas	esgepen.ddns.net:8080/share	Sem Filtro	Liberada	Liberada	Liberada
	demo.nxfilter.org:4080/admin	Sem Filtro	Liberada	Erro	Erro

Considerando as informações obtidas através da Tabela 3, pode-se observar que existem quatro tipos de resultados das requisições utilizadas nos testes, sendo eles:

- Sem Filtro: a requisição não foi interceptada por nenhum tipo de filtragem WEB, ou seja, a requisição alcançou o servidor WEB e a página WEB foi devidamente carregada;
- Liberada: a requisição foi interceptada por um filtro WEB e foi devidamente encaminhada ao seu destino, baseado na política de controle de acesso. A resposta da requisição foi o devido carregamento da página WEB solicitada;
- Bloqueada: a requisição foi interceptada por um filtro WEB e não foi encaminhada ao seu destino, baseado na política de controle de acesso. A resposta da requisição foi uma página gerada pelo próprio filtro WEB informando que o acesso foi negado;
- Erro: a requisição foi interceptada por um filtro WEB e, em vez de dar algum retorno esperado, como o carregamento da página solicitada, ou a exibição da página informando a negação de acesso, foi exibida uma página informando um erro na resposta à requisição.

Durante a realização dos testes, foi possível identificar cada tipo de resultado através do retorno visual do carregamento de páginas WEB no navegador, sejam do alvo da requisição, de negação de acesso ou de informação sobre algum erro. Na Figura 23, a seguir, é apresentada a resposta do retorno de uma requisição realizada no cenário 1.

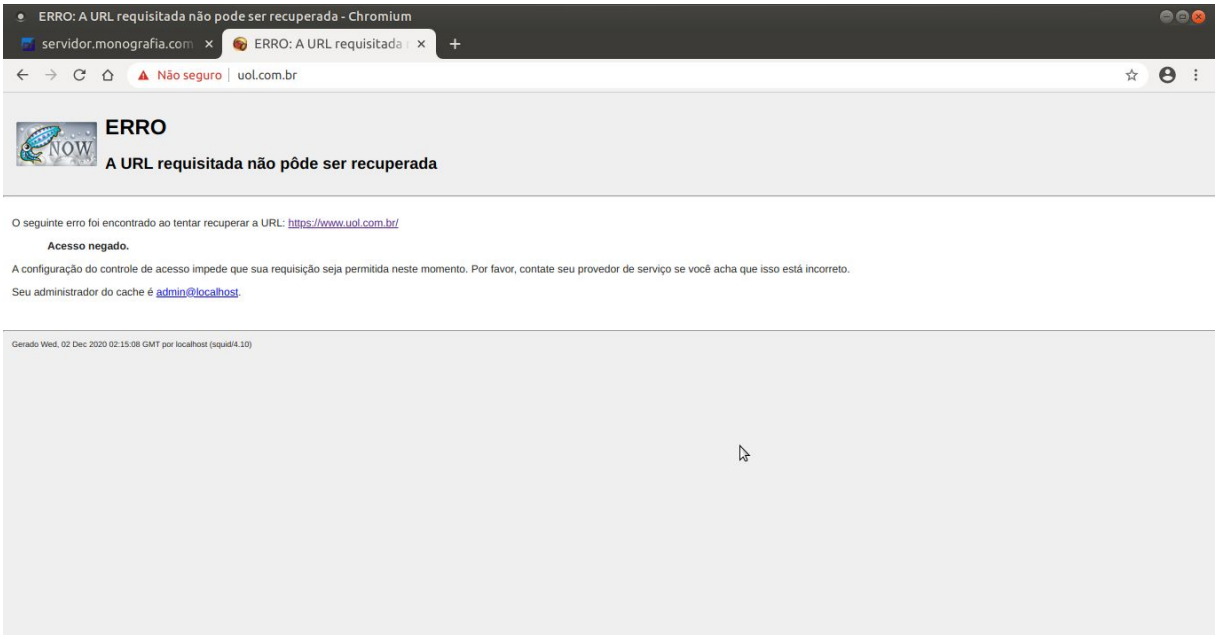
Figura 23: Resposta de requisição ao endereço google.com no cenário 1.



Na Figura 23 foi realizada uma requisição WEB com o endereço “google.com” e a página WEB foi devidamente carregada. O processo em que a requisição chegou ao servidor WEB e mesmo respondeu, carregando a página WEB, foi observado em todas as requisições cujos resultados foram “Sem Filtro” e “Liberada”, nos testes em todos os cenários de rede. Por este motivo não foram incluídas no trabalho as demais capturas de tela com este mesmo resultado durante os testes e utilizando a Figura 23 como exemplo para todos os resultados semelhantes.

Nos cenários onde estava sendo realizada a filtragem WEB utilizando o Squid, as requisições WEB que foram interceptadas e bloqueadas apresentaram uma página de aviso, informando que o acesso foi negado, assim como pode ser observado na Figura 24, a seguir.

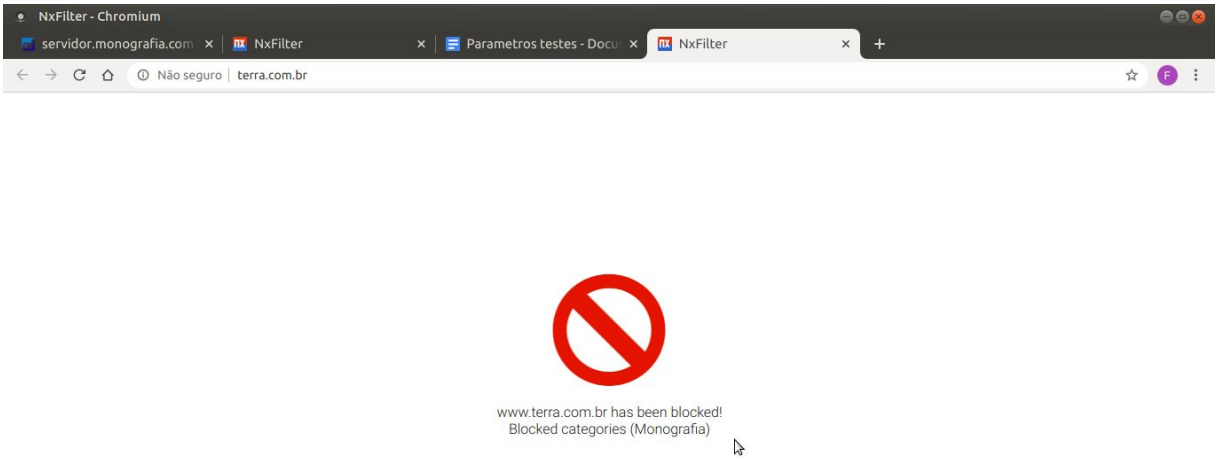
Figura 24: Resposta de requisição WEB bloqueada pelo *proxy* HTTP Squid no cenário 2.



Na figura 24, foi realizada uma requisição no cenário 2 com o endereço “uol.com.br” pelo navegador WEB, e o acesso foi negado, já que o nome de domínio em questão estava incluso na lista de bloqueio do Squid. Cada requisição com esse resultado está listada na Tabela 3 como “Bloqueada”. A Figura 24 serve como exemplo para todas as requisições que obtiveram esse retorno, não necessitando incluir no trabalho todas as capturas de tela com o resultado semelhante.

Assim como no bloqueio de requisições pelo Squid, o NxFilter também exibia uma página de negação de acesso caso a requisição contivesse um nome de domínio constante na lista de bloqueios. A seguir, na Figura 25, a resposta de negação de acesso do NxFilter é apresentada.

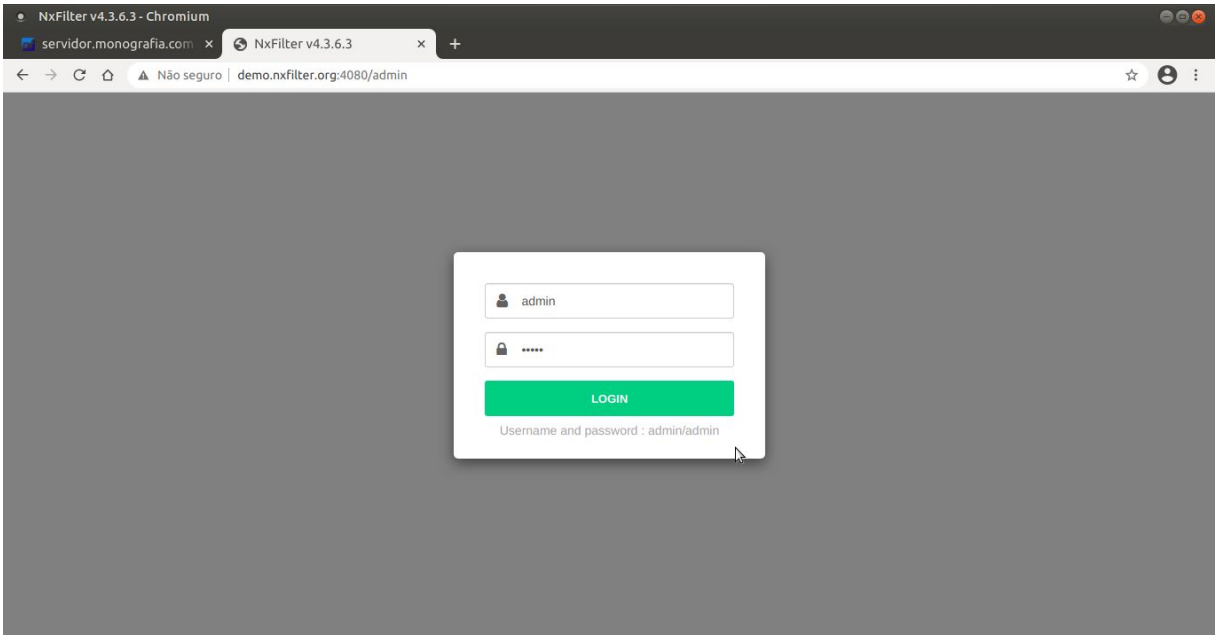
Figura 25: Resposta de requisição WEB bloqueada pelo filtro DNS NxFilter no cenário 3.



Como pode ser observado na Figura 25, a resposta de bloqueio foi apresentada após a requisição com o endereço “terra.com.br” ser realizada no cenário 3. O nome de domínio em questão estava contido na lista de bloqueio do filtro DNS. As requisições com este resultado estão listadas na Tabela 3 como “Bloqueada”, e a Figura 25 serve como parâmetro para exemplificar todos os testes com resultado semelhante.

No teste com portas de rede distintas, no cenário 2, com a requisição utilizando o endereço “demo.nxfilter.org:4080/admin”, a requisição foi “Liberada” conforme pode ser observado na Figura 26 a seguir.

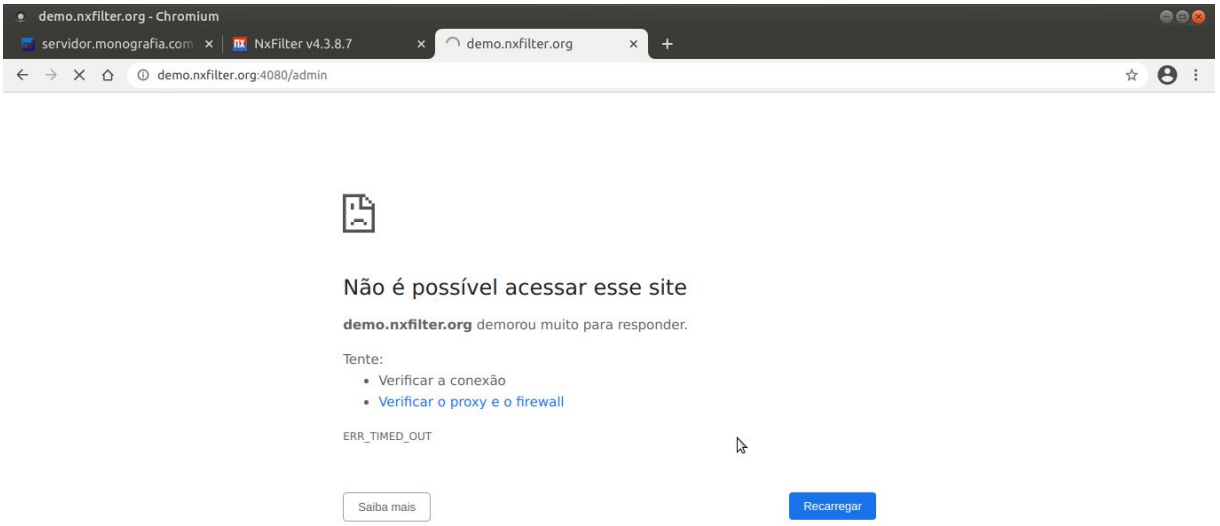
Figura 26: Resultado da requisição no teste com portas de rede distintas no cenário 2.



A requisição alcançou o servidor WEB de destino e carregou a página corretamente, porém o nome de domínio em questão estava adicionado na lista de domínios bloqueados. A filtragem WEB utilizando o Squid permitiu que a requisição alcançasse o destino, o que em tese não deveria ter acontecido. O Squid também não bloqueou a requisição WEB no cenário 4, no teste com portas de rede distintas, utilizando o mesmo endereço do caso anteriormente mencionado.

Entre todas as respostas padrão apresentadas no bloqueio de acesso do Squid e NxFilter, houve duas situações em que o retorno foi diferente do esperado. A Figura 27 a seguir apresenta o resultado ocorrido no cenário 3, no teste com portas de rede distintas, com a requisição utilizando o endereço “demo.nxfilter.org:4080/admin”.

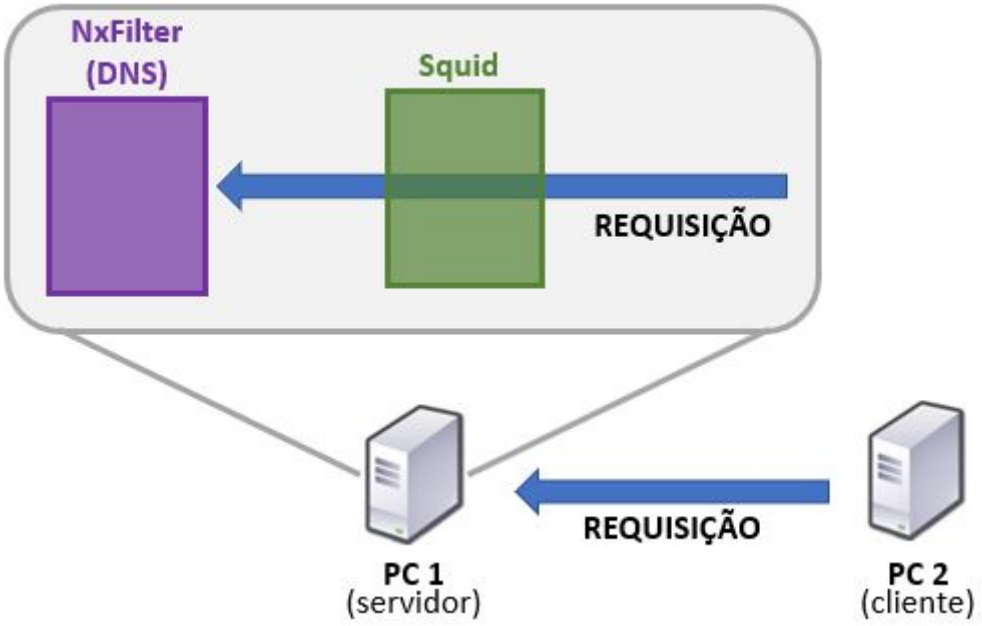
Figura 27: Resultado da requisição no teste com porta de rede distinta, no cenário 3.



Como pode ser visto na Figura 27, o retorno da requisição foi uma página de erro, informando que o servidor de destino demorou muito para responder. Esse erro geralmente ocorre quando o servidor não é alcançado e não há nenhuma resposta até o tempo limite da requisição. O resultado esperado neste teste era que a requisição fosse bloqueada pelo NxFilter, porém em vez da página de bloqueio, a requisição não obteve resposta até atingir o tempo limite de espera. O mesmo erro apresentado na Figura 27 também ocorreu no teste com portas de rede distintas, no cenário 4, com o mesmo endereço de requisição.

Durante a realização dos testes no cenário 4, foi observado que todas as requisições realizadas foram interceptadas primeiramente pelo Squid e, posteriormente, pelo NxFilter. Esta sequência é justificada pelo fato de o Squid ser um *proxy*, ou seja, todas as requisições WEB realizadas pelo cliente de rede, ao chegarem na placa de rede do servidor, foram interceptadas pelo *proxy*, já que o mesmo era o intermediário das requisições. A seguir a Figura 28 demonstra esse processo.

Figura 28: Intermediação das requisições pelo *proxy* no cenário 4.



Uma consulta a um servidor DNS faz parte de uma requisição WEB. Sendo assim, a intermediação das requisições pelo *proxy* intercepta a requisição de consulta a um servidor DNS. No caso do cenário 4, obrigatoriamente a consulta ao servidor DNS passou pelo *proxy*, já que o servidor DNS em questão se encontrava no servidor de rede.

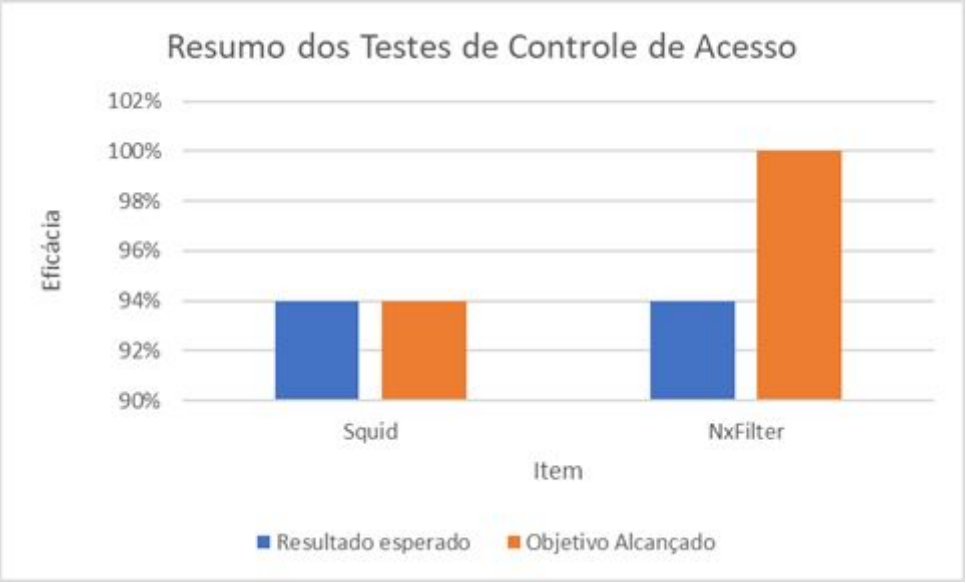
Considerando todos os resultados mencionados acerca dos testes de controle de acesso, é possível chegar às seguintes conclusões:

- no cenário 1, os resultados se comportaram como o esperado (todas as requisições chegaram ao seu destino e as páginas WEB foram devidamente carregadas);
- no cenário 2, no teste com portas de rede distintas, o filtro permitiu que a requisição com o endereço “demo.nxfilter.org:4080/admin” alcançasse o servidor WEB e a página foi carregada. A partir dessa constatação pode-se afirmar que, na configuração básica, o Squid pode não filtrar adequadamente requisições WEB que utilizam portas diferentes da 80 (HTTP) e 443 (HTTPS), e permitir que o servidor WEB seja alcançado;
- no cenário 3, no teste com portas de rede distintas, foi carregada uma página de erro na requisição com o endereço “demo.nxfilter.org:4080/admin”. É possível constatar que, ainda que o retorno não tenha sido o esperado, o filtro não permitiu que a requisição fosse atendida, ou seja, o objetivo foi alcançado;
- no cenário 4 ocorrem os mesmos resultados não esperados observados nos cenários 2 e 3;

- no cenário 4, o Squid se sobrepôs ao NxFilter, realizando a filtragem primeiro. Isso fez com que o NxFilter funcionasse como uma espécie de segunda camada de controle de acesso, realizando o processo de filtragem no momento em que o Squid falhou no processo de filtragem. Sendo assim, o Squid cumpriu um papel primário e o NxFilter um papel secundário no cenário;
- apesar de o Squid realizar a filtragem WEB primeiro, o NxFilter estava realizando o processo de filtragem assim que uma requisição fosse permitida pelo Squid e a consulta ao servidor DNS fosse realizada;

Para resumir os resultados dos testes de controle de acesso, observando a efetividade de cada ferramenta, a Figura 29 demonstra através de um gráfico os números gerais alcançados.

Figura 29: Resumo dos números dos testes de controle de acesso.



Analisando a Figura 29, percebe-se que, enquanto o Squid empata com o NxFilter com relação ao resultado esperado, ficando em 94%, o NxFilter alcançou em 100% o objetivo do controle de acesso, enquanto o Squid falhou no controle de acesso em 6% dos testes.

Em posse de todas as informações derivadas do desenvolvimento do trabalho, foi possível então realizar o paralelo entre as duas ferramentas, demonstrando as diferenças entre cada abordagem de filtragem. Na seção a seguir o paralelo entre as ferramentas de filtragem WEB é demonstrado.

4.5 Paralelo entre Squid e NxFilter

Foi realizado um paralelo entre as ferramentas de filtragem WEB Squid e NxFilter, considerando os resultados obtidos na implantação dos cenários de rede, nas análises de latência da rede e consulta DNS e nos testes de controle de acesso. A Tabela 4 apresenta o paralelo em questão.

Tabela 4: Paralelo entre as ferramentas de filtragem WEB Squid e NxFilter.

		FERRAMENTA DE FILTRAGEM WEB	
MÉTODO	**ASPECTO**	**Squid**	**NxFilter**
Implantação dos Cenários de rede	Abordagem da Ferramenta	Realiza filtragem intermediando as requisições entre a rede interna e externa.	Realiza a filtragem na consulta ao servidor DNS.
	Método de Filtragem	Transparente e não transparente.	Apenas transparente.
	Interface de Configuração	Não possui, porém torna-se configurável pela interface do PfSense.	Possui interface própria, além uma série de configurações avançadas.
	Configuração Básica	Funciona bem com HTTP, necessita de configurações mais complexas para HTTPS.	Funciona bem com HTTP e HTTPS, e disponibiliza extensão de navegador para facilitar a utilização de certificado SSL
	Configuração de lista de Bloqueios	Simples pela interface do PfSense	Simples pela interface da própria ferramenta, e possui várias configurações avançadas de simples configuração
	Informações de Filtragem	Arquivo de log da ferramenta	Além do arquivo de log, possui dashboard com lista de bloqueios, números totais de requisições, gráficos, filtro por requisição, horário, usuário, politica de bloqueio, etc.
Análise de Latência de Rede	Impacto na Latência sem Fluxo Contínuo de Rede	Sem alteração significativa	Sem alteração significativa
	Impacto na Latência com Fluxo Contínuo de rede	Sem alteração significativa	Sem alteração significativa
Análise de Resposta DNS	Impacto da Filtragem no Tempo de Resposta DNS	Sem impacto no tempo de resposta	Impacto considerável no tempo de resposta
Testes de Controle de Acesso	Resultado Esperado	Em 94% das tentativas o resultado foi o esprado	Em 94% das tentativas o resultado foi o esprado
	Objetivo Alcançado	Em 94% das tentativas o objetivo foi alcançado	Em 100% das tentativas o objetivo foi alcançado
	Implantação Conjunta (Squid e NxFilter)	Interceptou as requisições antes do NxFilter. Foi o filtro primário.	Interceptou as requisições após serem interceptadas pelo Squid. Foi o Filtro Secundário.

Considerando as informações constantes no paralelo apresentado pela Tabela 4 e ponderando a relevância dessas informações para a decisão da escolha de uma ferramenta de filtragem WEB, é possível chegar às seguintes conclusões:

- com relação a abordagem das ferramentas, o Squid se torna a ferramenta indicada caso se deseja ter total controle sobre as requisições que trafegam pelas portas de rede na conexão entre a rede interna e o servidor. Já o Nxfilter pode se tornar a melhor alternativa caso o interesse seja exclusivamente com o controle de acesso, não necessitando ter controle sobre o tráfego em si nas portas de rede;
- no aspecto do método de filtragem, se for necessário disponibilizar a filtragem WEB apenas para alguns dispositivos ou usuários na rede ou, ainda, limitar o acesso a internet apenas para os dispositivos configurados manualmente para utilização do filtro, o Squid é a melhor alternativa, considerando a possibilidade de uso como *proxy* não transparente;
- analisando sobre a instalação das ferramentas no PfSense, as duas são de fácil instalação. A única ressalva nesse sentido é que a ferramenta NxFilter exige um tempo maior de instalação. Neste sentido, não há grande relevância considerar este parâmetro para direcionar a escolha de uma ferramenta;
- sobre a interface de configuração, após a instalação do Squid, é criada uma área específica para o serviço de *proxy* na interface de configuração do próprio sistema, facilitando a realização da configuração. Já o Nxfilter possui sua própria interface de navegação, com todas as configurações disponíveis de forma fácil para acesso e alterações, sendo mais completo que o Squid nesse aspecto. É mais simples configurar o NxFilter;
- considerando a configuração básica das ferramentas para utilização no trabalho, o Squid não vem configurado por padrão para interceptação de requisições HTTPS, necessitando de ajustes adicionais nesse sentido. Já o NxFilter já vem, na sua configuração básica, habilitado para filtrar requisições em HTTPS, facilitando o trabalho nesse tipo de protocolo. Além disso, o NxFilter disponibiliza extensão de navegador para problemas de certificação SSL no processo de filtragem HTTPS, o que pode isentar configurações extensas nesse sentido;
- com relação a lista de bloqueios, a interface de configuração do Squid no PfSense apesar de simples se mostra limitada, já que disponibiliza apenas um espaço para inserir os domínios de rede que serão bloqueados. Já a interface do NxFilter, apesar de simples, disponibiliza listas categorizadas pré-definidas, com uma série de nomes de domínio já incluídos e permite a criação de categorias personalizadas. O NxFilter se mostra uma ferramenta mais completa para manipular os bloqueios;

- no aspecto das informações de filtragem das ferramentas, o NxFilter possui uma visão muito bem elaborada da interceptação das requisições. A ferramenta disponibiliza um *Dashboard* com gráficos, lista de bloqueios e possibilidade de filtrar as requisições de várias formas, desde o período de interceptação até o endereço IP requisitante. Já o Squid disponibiliza apenas um arquivo de texto sem tratamento de dados;
- analisando o impacto que as ferramentas podem gerar na latência da rede, com ou sem fluxo contínuo de dados, os resultados não mostraram situações que pudessem significar interferência da filtragem WEB. Sob esse aspecto não há uma ferramenta que traga uma vantagem em detrimento da outra;
- considerando a análise de tempo de resposta do DNS, o Squid mostrou não ter impacto significativo nesse quesito. Já o NxFilter aumentou o tempo de resposta do DNS, o que pode ser uma informação importante na escolha da ferramenta;
- com relação aos testes de controle de acesso, o NxFilter obteve o melhor resultado considerando que, mesmo não dando o *feedback* correto de todas as respostas das requisições, bloqueou todas as requisições cujos domínios estavam na lista de bloqueios. Já o Squid permitiu que algumas requisições conseguissem alcançar o servidor WEB de destino mesmo com o domínio da mesma estando na lista de bloqueio;
- ainda sobre os testes de controle de acesso, a utilização conjunta do Squid e do NxFilter para a filtragem WEB viabilizou a possibilidade da filtragem WEB em duas camadas onde, caso o Squid falhe no bloqueio de uma requisição, o NxFilter realiza esse bloqueio. Além da filtragem em duas camadas, não houve nenhum benefício adicional encontrado nos testes realizados.

Com a análise do paralelo entre as duas ferramentas percebe-se que a escolha da ferramenta da filtragem WEB depende exclusivamente dos objetivos específicos identificados para o ambiente de rede. Através das informações coletadas e analisadas conclui-se que, caso o controle sobre as requisições nas portas de rede, o uso do filtro como não transparente e o tempo mínimo possível para consultas DNS sejam consideradas pontos imprescindíveis para escolha de um sistema de filtragem WEB, o *proxy* HTTP Squid se torna a alternativa viável. Porém, se as necessidades observadas para a escolha da filtragem WEB sejam o foco específico no processo de filtragem, não considerando o tráfego de requisições nas portas de rede, o uso da filtragem exclusivamente no modo transparente, a configuração padrão já incluir a interceptação HTTPS, as configurações de bloqueio sejam variadas e categorizadas,

informações de interceptação elaboradas, com gráficos e filtros, O filtro DNS NxFilter se torna a melhor opção.

Com relação ao uso conjunto das duas ferramentas de filtragem WEB, conclui-se que a alternativa é válida caso haja necessidade de possuir duas camadas de filtro na rede, de forma que, se o Squid, por um motivo qualquer, falhar na interceptação, o NxFilter intercepta posteriormente.

4 CONSIDERAÇÕES FINAIS

O presente estudo procurou esclarecer o mecanismo de funcionamento de duas ferramentas utilizadas para filtragem WEB, Squid e NxFilter, que possuem abordagens distintas para alcançar este objetivo. Com o entendimento sobre a atuação do *proxy* HTTP e do Filtro DNS no processo de filtragem, quatro cenários de rede foram desenvolvidos, as ferramentas em questão foram implantadas e análises e testes foram realizadas nestes cenários, com o intuito de coletar informações que pudessem auxiliar na criação de um paralelo comparando as abordagens de filtragem em cada cenário, de forma a auxiliar em uma possível escolha de uma ferramenta com essa finalidade.

Com a realização do trabalho, foi possível concluir que a escolha de uma abordagem específica ou a utilização das duas abordagens estudadas em conjunto deve, prioritariamente, observar questões particulares do ambiente de rede que será submetido ao processo de filtragem. Cada abordagem atende a determinados requisitos que devem ser observados nesse processo de escolha. Com o paralelo entre as ferramentas dessas abordagens, é possível levantar cada um desses requisitos para realizar a escolha que, possivelmente, será a mais eficiente.

Apesar do estudo utilizar filtragem HTTP e HTTPS, análise de latência, análise de consulta DNS e testes de controle de acesso, é possível explorar em um trabalho futuro outros aspectos disponíveis para complementar o arcabouço de conhecimento sobre a filtragem WEB e reforçar a escolha da ferramenta ideal para determinadas situações. Entre os aspectos que podem ser estudados em um novo trabalho estão as configurações avançadas das ferramentas, o tempo médio de vida da requisição até ela alcançar seu objetivo ou ser bloqueada, além do impacto do volume de requisições na filtragem.

O fato do estudo surgir de uma necessidade real encontrada pelo autor, reforça a relevância que este estudo pode ter para auxiliar casos reais de aplicação de filtragem WEB em redes de computadores.

REFERÊNCIAS

BITWIZARD. **MTR**. 2020. Página da WEB. Disponível em: http://www.bitwizard.nl/mtr. Acesso em: Setembro de 2020.

CANONICAL. **Sistema Operacional Ubuntu PC | Ubuntu**. 2020. Página da WEB. Disponível em: https://ubuntu.com/desktop. Acesso em: Setembro de 2020.

CURI, M.; RIBEIRO FILHO, C. F. PROXY SQUID: Os impactos em produtividade e segurança com o uso de controladores de conteúdo nas microempresas. **Interação Revista de Ensino, Pesquisa e Extensão**, Varginha, v. 17, n. 17, p. 94 – 110, Fevereiro 2019. Disponível em: https://periodicos.unis.edu.br/index.php/interacao/article/view/79. Acesso em: Setembro de 2020.

GOOGLE. **Google Code Archive - Long-term storage for Google Code Project Hosting**. 2020. Página da WEB. Disponível em: https://code.google.com/archive/p/namebench/. Acesso em: Setembro de 2020.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION / INTERNATIONAL ELECTROTECHNICAL COMMISSION. **ISO/IEC 27000:** Information technology — Security techniques — Information security management systems — Overview and vocabulary. [S.l.],
2016. Disponível em: https://www.iso.org/obp/ui/#iso.std.iso-iec.27000.ed-4.v1.en. Acesso em: Agosto de 2020.

INTERNET ENGINEERING TASK FORCE. **RFC 1035:** Domain Names - Implementation and Specification. [S.l.], 1987. Disponível em: https://tools.ietf.org/html/rfc1035. Acesso em: Setembro de 2020.

INTERNET ENGINEERING TASK FORCE. **RFC 2979**: Behavior of and Requirements for Internet Firewalls. [S.l.], 2000. Disponível em: https://tools.ietf.org/html/rfc2979. Acesso em: Agosto de 2020.

INTERNET ENGINEERING TASK FORCE. **RFC 8499:** DNS Terminology. [S.l.], 2019. Disponível em: https://tools.ietf.org/html/rfc8499. Acesso em: Setembro de 2020.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet:** uma abordagem top-down. 6. ed. São Paulo. Pearson Education do Brasil Ltda, 2013. 634 p.

USPTO. Shigezumi Makiba. **DNS Server Filter**. US20010052007A1, 2 jan 2001. Disponível em: https://patents.google.com/patent/US20010052007A1/en. Acesso em: Setembro de 2020.

NEVES, F. C. das; MACHADO, L. A.; CENTENARO, R. da F. **Implantação de Firewall PfSense**. 2014. 66 p. Monografia (Tecnologia em Sistemas de Telecomunicações) — Universidade Tecnológica Federal do Paraná. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/3968/1/CT_COTEL_2014_2_02.pdf. Acesso em: Setembro de 2020.

NEVES, K. A. **Nomes de Domínio na Internet:** Aplicação do Sistema de Solução de conflitos. 1. ed. São Paulo. Novatec, 2015. 176 p.

NXFILTER. **NxFilter, Your free DNS filter! | DNS based webfilter for free**. 2020. Página da WEB. Disponível em: https://nxfilter.org/p3/why-nxfilter/. Acesso em: Setembro de 2020.

OLIVEIRA, V. S. M. **EVOLUÇÃO DA TECNOLOGIA DE PROXY: Análise de Características**. 2010. 73 p. Monografia (Especialização em Gerência de Redes e Tecnologia Internet) — Universidade Federal do Rio de Janeiro. Disponível em: https://pantheon.ufrj.br/bitstream/11422/3166/3/VSMOliveira.pdf. Acesso em: Agosto de 2020.

ORACLE. **Oracle VM VirtualBox**. 2020. Página da WEB. Disponível em: https: //www.virtualbox.org. Acesso em: Setembro de 2020.

SANTIAGO, H. L. P.; LISBOA, G. dos S. Segurança de Sistemas de Informação – O Contexto da Segurança dos Sistemas de Informação. **Núcleo de Iniciação Científica**, Paracatu, 2011. Disponível em: http://www.atenas.edu.br/Faculdade/arquivos/ NucleoIniciacaoCiencia/REVISTAS/REVIST2011/6.pdf. Acesso em: Agosto de 2016.

SQUID-CACHE. **squid. Optimising Web Delivery**. 2020. Página da WEB. Disponível em: http://www.squid-cache.org/Intro/. Acesso em: Setembro de 2020.

TANENBAUM, A. S. **Redes de computadores**. 4. ed. Rio de Janeiro. Elsevier, 2013. 946 p.

YEU, Y. C.; FEDEL, G. de S. Aceleração no acesso à Internet: estudo sobre o servidor proxy/cache Squid. **Revista Tecnológica da Fatec Americana**, Americana, v. 2, n. 1, p. 12 – 34, Março 2014. Disponível em: http://www.fatec.edu.br/revista_ojs/index.php/ RTecFatecAM/article/view/9/14. Acesso em: Agosto de 2020.