



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

BRUNO HENRIQUE SIQUEIRA MASCHIO

SEGURANÇA DE INFRAESTRUTURA COM PFSENSE

Assis/SP

2017



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

BRUNO HENRIQUE SIQUEIRA MASCHIO

SEGURANÇA DE INFRAESTRUTURA COM PFSENSE

Projeto de pesquisa apresentado ao curso de ciência da computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito à obtenção do Certificado de Conclusão.

**Orientando: Bruno Henrique Siqueira Maschio
Orientador: Prof. Douglas Sanches Da Cunha**

Assis/SP

2017

RESUMO

Ter uma infraestrutura segura, é essencial para qualquer ambiente de rede de computadores, assim como ter uma boa medida de segurança adotada, e um dos melhores sistemas de segurança na atualidade é a utilização de um *firewall*. O *pfsense* trata-se de uma ferramenta gratuito e com todos recursos essenciais, podendo apresentar medidas de segurança, mais do que suficientes. Para testar isto, será posto à prova, as funcionalidades da ferramenta, através de um *pentest*, onde será auditado, com o objetivo de encontrar supostas falhas, seja de má configuração, ou qualquer outra. Os testes são baseados em um ambiente configurado com alguns serviços sendo executados, com a ferramenta fazendo sua frente de defesa, não apenas as tentando as proteger, mas também registrar qualquer ocorrência dentro da rede, para que assim sejam analisados os dados e proposto melhorias, além de tentar identificar a origem do ataque, seu tipo e as consequências causadas. Em geral a segurança da infraestrutura montada será baseada em protocolos adotados dentro na mesma sempre buscando utilizar uma compatibilidade com o suporte oferecido pelo *pfsense*.

Palavras-chave: Segurança, PFSense, Pentest.

ABSTRACT

Containing secure infrastructure is essential for any computer network environment and a good security measure, and one of the best security systems today is the use of a firewall. Pfsense is a free tool with all the essential features and can present more than enough security measures. To test, be tested, as functionality of the tool, through a pentestino, where they will be checked, to find the alleged defects in any way or in any other way. The test is based on an environment configured with some services running, with the defense tool, which is not suspended as a protection attempt, but also registers contingencies within the network, so that, as analyzed data and proposed improvements, Besides trying to identify the origin of the attack, its type and its consequences. In general, the security of the mounted infrastructure will be informed in the protocols adopted within it always trying to use compatibility with the support provided by PFSense.

Key Words: Security, PFSense, Pentest.

FICHA CATALOGRÁFICA

MASCHIO, Bruno Henrique Siqueira

Segurança de infraestrutura com PFSense. Fundação Educacional do Município de Assis – FEMA – Assis, 2017.
50p.

Orientador: Prof. Douglas Sanches Da Cunha

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA

1. PFSense 2. Pentest 3. Ambiente 4. Testes

CDD: 001.6
Biblioteca da FEMA

SUMÁRIO

1. INTRODUÇÃO.....	7
1.1. OBJETIVO.....	8
1.2. PUBLICO ALVO.....	8
1.3. JUSTIFICATIVA.....	8
1.4. MOTIVAÇÕES.....	9
1.5. PERSPECTIVA DA CONTRIBUIÇÃO.....	9
1.6. METODOLOGIA DE DESENVOLVIMENTO.....	9
1.6.1 RECURSOS.....	10
2. PFSENSE.....	11
2.1. NETWORK ADDRESS TRANSLATION.....	12
2.2. GATEWAY.....	13
2.3. DOMAIN NAME SYSTEM.....	13
2.4. PROXY.....	14
2.5. DYNAMIC HOST CONFIGURATION PROTOCOL.....	16
2.6. INSTALAÇÃO DO PFSENSE.....	17
2.7. CONFIGURAÇÃO BÁSICA.....	17
2.7.1. WIDE AREA NETWORK.....	18
2.7.2. DHCP.....	19
2.7.3. DNS.....	21
2.8. OUTRAS CONFIGURAÇÕES.....	22
2.8.1. ROTEAMENTO / REDIRECIONAMENTO.....	22
2.8.2. FIREWALL.....	23
2.9. DEMAIS RECURSOS.....	24
3. PENTEST.....	25
4. AMBIENTE DE TESTES.....	28
5. TESTES DE DISPONIBILIDADE DO AMBIENTE.....	30
6. TESTES COM KALI LINUX BASEADAS NO AMBIENTE.....	34
6.1. TESTE SERVIDOR 0 - RDP.....	34
6.2. TESTE SERVIDOR 1 - HTTP.....	36
7. OUTRAS REGRAS DE SEGURANÇA.....	40
7.1. SUB-REDE.....	41
7.2. VLAN.....	41
7.3. FAILOVER E LOADBALANCER.....	43
7.4. ÁREAS DMZ.....	44
7.5. PFSENSE COM AUXILIO DE SERVIDORES SECUNDÁRIOS.....	45
8. CONCLUSÃO.....	47
8.1. TRABALHOS FUTUROS.....	48
9. REFERENCIAS.....	49

1. INTRODUÇÃO

Atualmente, é cada vez mais comum, empresas utilizarem sistemas de gerenciamento ERP (Enterprise Resource Planning, ou Planejamento de Recursos Empresariais), e servidores de compartilhamento de arquivos, deixando todas as informações dentro de uma única rede. Portanto, utilizar uma estratégia de segurança é de extrema importância, pois são os dados da organização que estão em evidência, assim é necessário buscar uma alternativa que dê segurança, e que também proporcione agilidade para o acesso.

A segurança é a base para dar às empresas a possibilidade e a liberdade necessária para a criação de novas oportunidades de negócio. É evidente que os negócios estão cada vez mais dependentes das tecnologias e estas precisam estar de tal forma a proporcionar confidencialidade, integridade e disponibilidade. (CAETANO; SOUZA; COSTA, 2011, p.4)

A utilização de um *firewall* é uma ação simples a ser adotada, implicando toda a política de rede, onde o principal objetivo é prevenir e monitorar os acessos não autorizados. Em geral o *firewall* é um “muro” entre a rede local e a externa, onde se filtra todos os pacotes de dados, aplicações, *proxy* de rede, e tráfego das informações.

Não é suficiente utilizar apenas uma medida de segurança, e como a necessidade de atuar na área de segurança é cada vez maior, adotar o sistema *pfSense* como firewall se torna uma boa escolha, pois, trata de uma alternativa totalmente gratuita. Por se basear no sistema operacional *FreeBSD*, acaba herdando as estratégias de segurança do mesmo, e adicionando funções de filtro e roteamento, as funções do sistema, ficam com a mesma qualidade que geralmente só se encontram em programas pagos. A ferramenta por utilizar um hardware exclusivo para ser executado, não apenas um software instalado no computador do usuário, ganha vantagem, por ter processamento dedicado para executar todas suas funções, não deixando uma comunicação lenta, e dando suporte a todas as funções necessárias para a rede.

A integridade de um parque computacional, é promovida através da segurança montada pelos profissionais da área, assim, este trabalho tem como objetivo apresentar uma configuração mínima a ser mantida ao *pfSense*, para poder apresentar uma rede mais segura, contra um possível ataque, e crimes virtuais.

1.1. OBJETIVO

Pesquisar e investigar o uso do *Pfsense*, como ferramenta de produção, tem como objetivo somar nas táticas de segurança, criando mais uma barreira, além de acompanhar com outros métodos já utilizados, pois não existe uma estrutura totalmente segura, a menos que não esteja conectado a nada, mas sim existem redes melhores preparadas para suportar imprevistos.

Segundo ALECIO e PEREIRA, 2014, Um *firewall* segue as regras, diretrizes previamente configuradas pelo administrador de rede. Assim para testar como seria uma melhor configuração será feito uma série de testes de invasão denominadas *pentest*, onde o objetivo é simular um ataque de forma maliciosa, onde é feito uma análise das vulnerabilidades do sistema, resultados de uma falha ou má configuração, e com base nos resultados, realizar uma invasão por meios das brechas encontradas.

1.2. PUBLICO ALVO

O trabalho busca atingir os responsáveis pela área de infraestrutura em tecnologia da informação e segurança, pois há uma grande responsabilidade sobre os quais, independentemente do tamanho da estrutura, pois, infelizmente alguns só começa a ter algum tipo de preocupação após sofrer um primeiro ataque, e conseqüentemente perdendo dados.

Grandes são os desafios para o gerenciamento de uma rede de dados e seus ativos, tendo como premissa a manutenção de sua disponibilidade (BENINI e DAIBERT, 2011). Com isso o Trabalho busca atingir principalmente aos que possuem uma maior preocupação por lidar com este problema de segurança todos os dias, como os responsáveis por acessos a bancos, data centers, e servidores voltados para nuvem, onde ocorrem inúmeros acessos diariamente, assim tendo que manter um fácil acesso para o usuário, e a segurança de seus dados, além de tentar acrescentar ao conhecimento dos interessados com o assunto.

1.3. JUSTIFICATIVA

Espera-se que sempre quando efetuado uma configuração de um novo *firewall*, sejam adotados os protocolos a serem apresentados, como parte de uma configuração essencial, assim criando uma política de rede mais refinada. Com isso, espera-se proporcionar novos estudos e metodologias de segurança, e encontrar soluções para problemas já existentes, pois

ataques estão se tornando cada vez mais comuns, principalmente pelo crescimento da necessidade de se informatizar todos os processos. Do mesmo modo que se cria novos métodos de ataque, é necessário adotar novas maneiras de defesa, assim intensificar e descrever conceitos importantes de um *firewall*, através das simulações.

1.4. MOTIVAÇÕES

Tem como motivação a realização do trabalho de conclusão de curso em ciência da computação e a exploração da ferramenta, buscando o aprendizado, e aplicações para a área de estudo.

1.5. PERSPECTIVA DA CONTRIBUIÇÃO

Com este trabalho tem a perceptiva compartilhar um novo estudo da plataforma, a qual é muito abrangente, mas explorando algumas funções específicas. Com base nas ações a serem tomadas comprovando que a ferramenta é confiável e eficaz, assim estando preparado para um real comprometimento com a defesa da rede.

Outra contribuição é deixar o conteúdo, como objeto de estudo, para qualquer um que deseje agregar conhecimento, ou até mesmo continuar o trabalho.

1.6. METODOLOGIA DE DESENVOLVIMENTO

Para a elaboração deste trabalho de conclusão de curso serão consultados livros, sites e tutoriais que forneçam informações referentes ao sistema usado para fornecer segurança a uma rede.

Na primeira etapa será apresentado e estudado a ferramenta, buscando a melhor configuração para se defender de um ataque, após isto, será estudado como funciona de um ataque, assim podendo pensar da mesma forma que um invasor.

Na segunda etapa será montado e configurado o ambiente para os testes, com um servidor que será atacado, o *firewall* que efetuara a defesa, e um computador executando os ataques. Após isto será recolhido dados dos testes, para uma conclusão, se a ferramenta realmente oferece total segurança, ou não, com base nos resultados do ocorrido.

1.6.1 RECURSOS

Para desenvolver a pesquisa serão necessários os recursos de hardware e software citados a seguir:

- 01 (um) firewall com 02 (duas) placas de rede;
- 01 (um) ou mais servidores;
- 01 (um) computador;
- Sistemas Operacionais FreeBSD, Kali Linux, Debian, entre outros sistemas operacionais;
- Acervo bibliográfico para consulta;
- Sites e fórum de discussões relacionadas a área;

2. PFSENSE

Neste capítulo, é abordado do que trata-se o *pfSense*, os protocolos de rede necessário para configurá-lo, além de explicar como utilizá-los dentro da ferramenta para uma configuração básica, e ao final como criar as primeiras regras voltadas para a parte de segurança, além citar seu potencial de expansão.

O *pfSense* é uma ferramenta poderosa, assim se torna eficiente e com uma interface amigável, já vindo preparado com uma pré-configuração básica, e pode ser incrementado com excelentes ferramentas de segurança e roteamento, entre outras soluções, que podem ser adquiridas por um sistema de pacotes, que vem a receber constantes atualizações e expansões. O projeto *pfSense* foi iniciado em setembro de 2004, por Chris Buechler e Scott Ullrich, baseado no sistema operacional *FreeBSD*, o qual é suportado por inúmeras plataformas, mostrando ser um sistema seguro, estável, e conhecido por ter alta performance em servidores, e aplicações para redes. Desta maneira o *pfSense* pode estar presente em diversas vertentes de uma infraestrutura de computadores.

Por se tratar de um *firewall*, a ideia da ferramenta é fornecer um acesso à internet controlado, onde filtra os dados destinados a rede interna, bloqueia tudo, e só libera o necessário. Quando chega um novo pacote, o analisa através de seu cabeçalho de IP (Internet Protocol, ou Protocolo de Internet), e os compara com suas tabelas de regras, assim podendo chegar a uma conclusão se o pacote prossegue ou não. Se permitido utiliza sua função de roteamento, o direcionado ao seu destino, ao contrário, se recusar o recebimento, descarta o pacote. Seu funcionamento é baseado em regras de aceitar ou rejeitar (accept or deny).

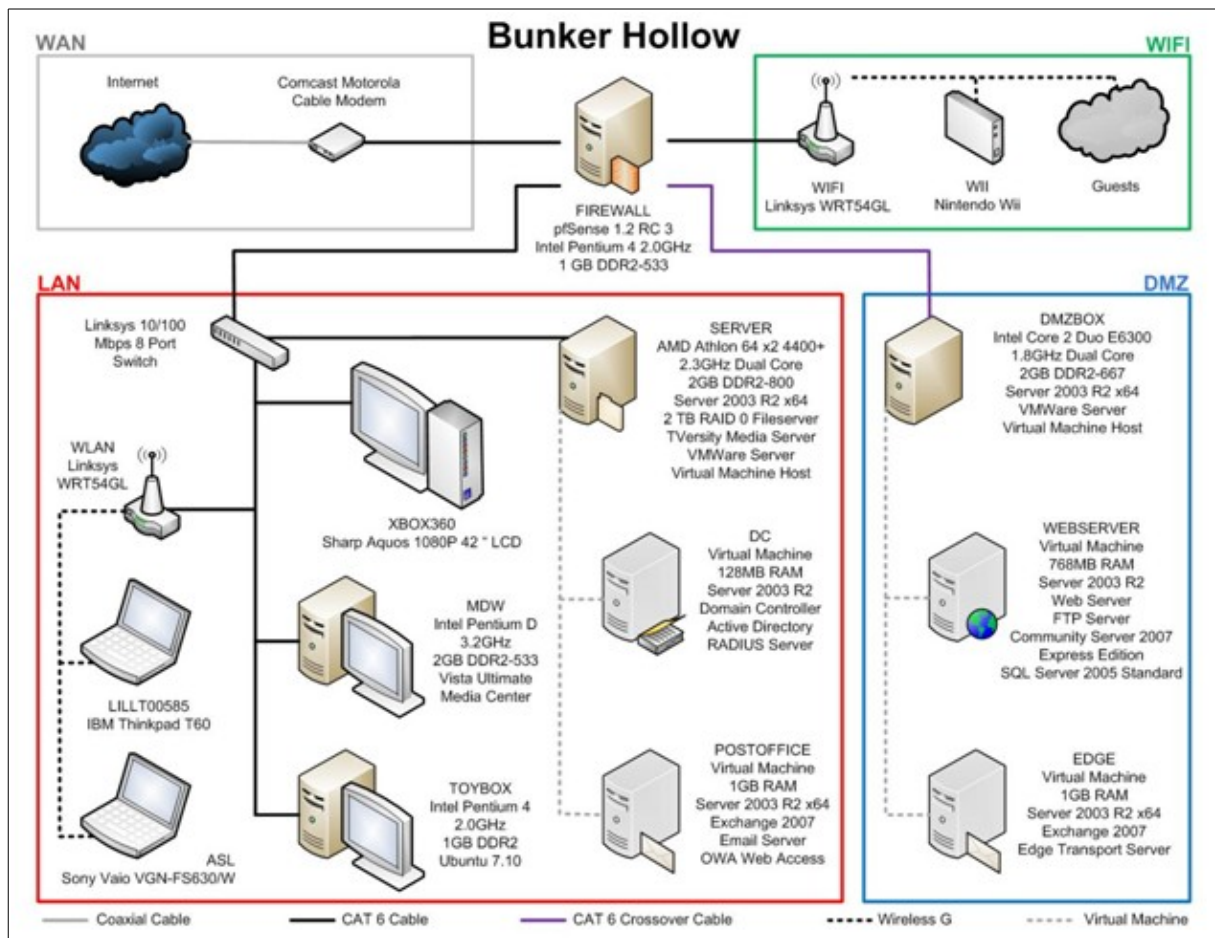


Figura 1 - Exemplo cenário de implantação

Fonte: https://www.safaribooksonline.com/library/view/pfsense-2-cookbook/9781849514866/graphics/4866_AppendixB_01.jpg

Na figura acima, é importante observar o posicionamento do equipamento com o pfsense instalado, pois o fato de estar entre a rede externa, e as demais redes, é que permite que ele trabalhe do melhor modo, onde filtra todo o tráfego conforme as regras criadas.

Para se montar uma rede com o *pfsense* implantado, é necessário ter em mente alguns conceitos básicos de redes, os quais devem ser configurado dentro do *firewall* para poder operar corretamente, e posteriormente facilitar uma análise, e um monitoramento do que ocorre, além de facilitar outras incrementações.

2.1. NETWORK ADDRESS TRANSLATION

A utilização de um NAT (Network Address Translation, ou Tradução do Endereço de Rede) é essencial, pois, utilizando uma tabela de dispersão, reescreve endereços de IP, e pelos

protocolos de roteamento distribuir a informação a seu destino específico para a rede interna. Por exemplo o IP interno 192.168.1.25, faz uma requisição para um endereço externo, assim um pacote sai da estação para um intermediador entre o ambiente interno e externo, mandando a requisição para o real destino com o uso de um *gateway* (onde utiliza o endereço real da rede), e o pacote é entregue ao destino solicitado pela estação, e ao retorno é recebido novamente pelo intermediador e entregue ao IP da estação requisitante.

De acordo com Battisti (2013), com o uso do NAT, é possível fornecer acesso à internet para um grande número de computadores da rede interna, usando um número bem menor de endereços IP, válidos na internet. O Serviço NAT precisa de três componentes para realizar sua função, o de **tradução de endereços**, onde fornece a conexão de internet, o componente de **endereçamento**, onde atua como um servidor de DHCP (Dynamic Host Configuration Protocol, ou Protocolo de Configuração de Host Dinâmico) simplificado, e componente de **resolução de nomes**, onde atua também como um servidor de DNS (Domain Name System, ou Sistema de Nomes de Domínio), assim quando ocorrer uma consulta para resolução de nomes, é repassado para um servidor DNS da Internet, retornando a resposta obtida para o cliente.

2.2. GATEWAY

Outro conceito importante que deve ser entendido para a implantação do *pfsense*, é do uso de um *gateway*, onde pode ser interpretado como uma porta de entrada, sendo definido nas propriedades da rede, ficando encarregado de acessar uma rede externa, e entregar um resultado a uma estação, conforme requisitado. Ao conectar-se à internet através de um provedor, é recebido um endereço de IP válido, para se obter o acesso a web, com isso apenas um computador teria acesso à internet, mas isto é facilmente resolvido com a utilização de um NAT, assim, disponibilizando conexão a outros computadores.

2.3. DOMAIN NAME SYSTEM

Toda configuração para conexão com a internet, é preciso ter um servidor de DNS, o qual é responsável por localizar e traduzir os endereços de IP, onde torna-se uma camada de abstração em transformar uma numeração IP em um endereço específico, o mesmo serve ao DNS reverso, buscando o domínio correspondente a um IP, e o autenticando. Para não ser

necessário todo acesso ocorrer a uma tradução, se utiliza o DNS cache, onde se ganha tempo, redirecionando uma requisição com base nos resultados de traduções anteriores. Com a importância do uso do DNS, também é importante a segurança, sendo feita pelo DNSSEC (DNS Security Extension, ou Extensão de Segurança da DNS), onde existe um esquema de criptografia, que utiliza chaves públicas e privadas, para a autenticação dos endereços consultados, garantindo o redirecionamento para o IP correto, evitando desvios de rotas e fraudes. O DNS pode ser utilizado também, para a detecção de sites falsos ou infectados, até mesmo melhorar o desempenho de navegação, e influenciando em um controle de permissões.

É importante ter o conhecimento do servidor de DNS escolhido, pois pode ocorrer roubo de informações através dela, como um ataque de DNS *poisoning*, onde é adicionada uma DNS falsa, direcionando a uma página de *phishing* (pescaria) onde acontece o roubo de informações. Outro método que pode ocorrer é o DNS *tunneling* (túnel de DNS), onde coletam informações através de requisições, assim utiliza um método que tem fácil progresso, pois pode ter bloqueio de uma requisição FTP (File Transfer Protocol), ou HTTP (Hyper Text Transfer Protocol), mas uma DNS estará sempre liberada. Para proteger o ambiente deve sempre utilizar servidores DNS confiáveis, já que uma vez infectado, é bem complicado perceber o ataque.

2.4. PROXY

Em uma rede, o *proxy* (procurador) é um intermediário para as requisições de um cliente a um outro servidor, onde ao buscar um arquivo, informações ou páginas web, suas solicitações passam por ele, e ao chegar a seu destino, tem um registro do IP da rede. O *proxy* é a porta de ligação com o *gateway*, e por conta deste fator, sua segurança é importante, onde protege o IP legítimo da rede com uma máscara, criando uma maior dificuldade para um possível invasor.

Uma das principais utilizações de um *proxy*, é para a filtragem de conteúdo, podendo controlar o tráfego, garantindo que a navegação fique dentro de uma política de rede, além de criar um registro dos acessos para um monitoramento, ou até mesmo para estatísticas de uso de rede.

Existe um tipo de *proxy*, o chamado transparente, onde tem uma arquitetura que permite que os clientes não saibam da existência do *proxy*, acreditando que as solicitações são diretas com o servidor, assim o *proxy* transparente precisa capturar e processar a solicitação.

Conforme Leite (2016), a principal vantagem nesta arquitetura é que não é necessária a configuração de *proxy* nos navegadores cliente. Outra (incorretamente) alegada vantagem é que o *proxy* não transparente não impede a conexão direta à Internet.

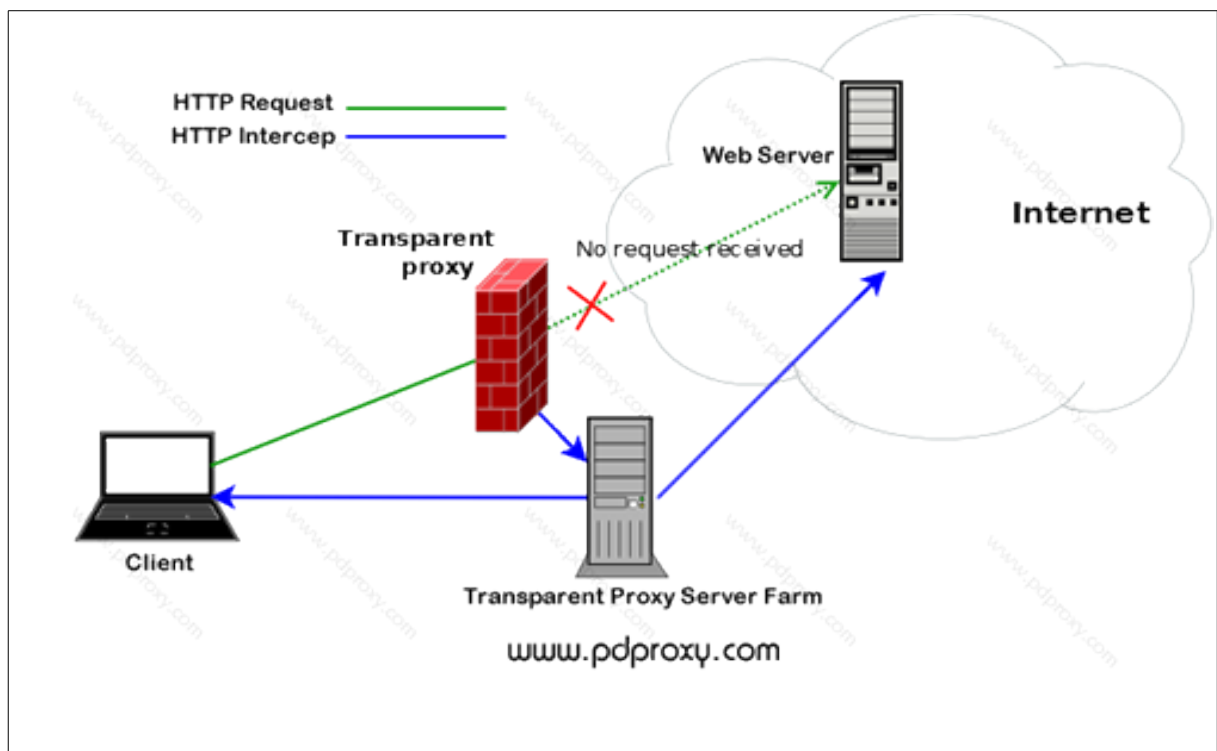


Figura 2 - Rede Com Proxy Transparente

Fonte: <http://www.it9.com.br/arquivos/2016/02/proxy-transparente-x-nao-transparente-desvendando-os-mitos.png>

Caso tenha a utilização de um *proxy* transparente é importante ter um servidor interno de DNS ligado a cache de um servidor *squid* (servidor proxy), assim não deixando uma requisição ser enviada a diante, sem antes haver uma verificação. Caso ocorra a utilização de um *proxy* não transparente, não ocorre nenhuma resolução DNS, mas uma solicitação dos recursos do servidor *proxy*.

De acordo com um tópico da comunidade hardware, onde tem uma discussão referente ao uso de *proxy*, disponível em <<http://www.hardware.com.br/comunidade/v-t/316639/>> são apresentados alguns pontos para não se utilizar em modo transparente, como por não funciona com protocolo HTTPS, FTP, ou qualquer site que utilize a porta diferente da 80, assim não

entrando dentro dos filtros, então existe um acesso permitido, além de poder ser alterado, configurando outro servidor de *proxy* manualmente.

2.5. DYNAMIC HOST CONFIGURATION PROTOCOL

O DHCP é um protocolo que é o responsável por distribuir conexão aos terminais, onde entrega as informações de rede, para que os clientes obtenham os dados de forma automática, recebendo um endereço de IP para a rede interna. Para isto ocorrer, quando um novo computador se conecta à rede envia um pacote solicitando a configuração da rede ao DHCP, o chamado *discover* (descobrir), o qual retorna o pedido com uma oferta de conexão, denominado *offer* (ofertar), assim o cliente confirma a solicitação com o *request* (pedido), e, por fim, ocorre o *acknowledge* (reconhecimento), onde o servidor DHCP, que está responsável por todos computadores conectados à rede, e com uma faixa de IP já definida, disponibiliza ao requisitante um IP válido, junto aos parâmetros de *gateway*, nome de domínio, e DNS.

De acordo com Pereira (2009), como o DHCP possui suporte para diversas plataformas, ele traz uma solução eficiente e fornece uma grande ajuda para os administradores de rede.

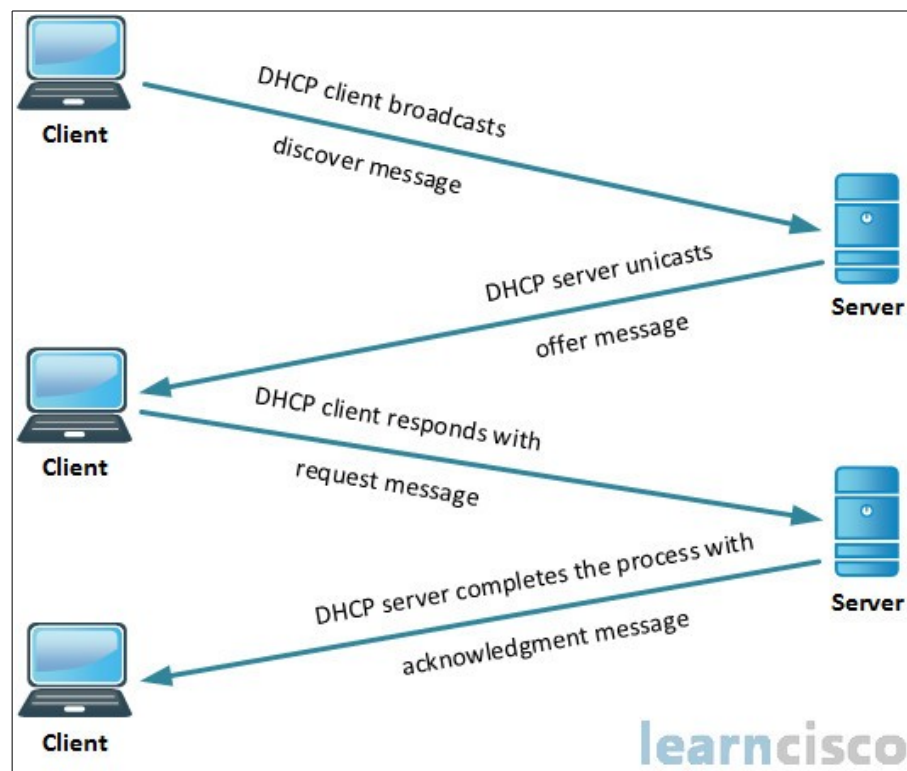


Figura 3 - Conexão DHCP

Fonte: <http://www.learncisco.net/assets/images/icnd1/80-dhcp-process.jpg>

2.6. INSTALAÇÃO DO PFSENSE

O *pfsense* pode ser instalado em qualquer computador, mas o recomendado seria a utilização de um *appliance firewall*. Seu reconhecimento de *drivers de hardware* se dá pela biblioteca do *FreeBSD*, e é necessário se utilizar no mínimo duas placas de rede, sendo uma para conectar a rede externa e a outra a interna, assim o *firewall* realizando seu papel.

O arquivo de instalação, ou ISO, sendo a imagem do arquivo de instalação do SO (Sistema Operacional) pode ser adquirida gratuitamente através do site da ferramenta.

2.7. CONFIGURAÇÃO BÁSICA

Uma vez com o sistema operacional instalado, é necessário conhecê-lo e configurado, e o primeiro passo a tomar, é conhecer as placas de rede do dispositivo, pois trata-se de manipular uma ferramenta de rede. Para apenas tomar conhecimento das placas será utilizado a opção 1 (um) no menu console, apenas para saber o endereço de MAC (Media Access Control, ou Controle de Acesso de Mídia) de cada uma das placas, e as configurar

posteriormente via interface gráfica, no navegador de um computador cliente conectado a rede local. Ao saber o endereço MAC, será cancelado a operação pois não será posto em prática a opção acessada, onde é utilizada para ativar ou desativar as placas, sendo que tal opção pode ser feita com maior facilidade via interface gráfica. Será utilizado a opção 2 (dois), do menu também para definir ou mudar os IP's da rede, como o endereço de IP padrão da rede LAN (Local Area Network, ou Rede de Areal Local), definindo um alternativo, caso não deseje utilizar o padrão. Após saber os endereços físico, através de uma máquina secundária da rede será acessado o sistema via cliente WEB, e será conectado através de um usuário e senha, e então acessado pelo menu principal as opções **Interfaces | (assign) | Interface assignment**, onde com base no endereço MAC pode ser trocado qual placa acessara determinado parâmetro de rede, como interfaces WAN, LAN, ou outra interface adicional.

2.7.1. WIDE AREA NETWORK

O primeiro item a ser configurada é o acesso WAN (Wide Area Network, ou rede de longa distância), é a que permite que nos conectemos a outros servidores, principalmente os que nos disponibiliza as páginas WEB, e outros tipos, o que é conhecido popularmente como Internet. Ao conhecer a numeração das placas de rede, e a interface que cada uma está destinada, deve ser acessado **Interfaces | WAN**, e parametrizar conforme as informações reais da rede, a qual pode variar conforme cada tipo de rede, mas sempre recebendo um *gateway* e a definindo para a rede WAN, com um endereço válido, assim distribuindo conexão aos computadores pertencentes ao domínio, exemplo com conexão dedicada, utilizando IPV4 (Internet Protocol Version 4, ou Protocolo de Internet Versão 4), onde tem o *gateway* 192.168.0.1, e seu primeiro o IP válido 192.168.0.103 com máscara de rede /24, será ativado a interface, setado um nome, e definido suas informações, e bloqueando as conexões de endereços privados, os quais não reconhecidos pelo IANA (Internet Assigned Numbers Authority, ou Autoridade para Atribuição de Números da Internet) a qual é uma organização mundial que supervisiona a atribuição global dos endereços de IP.

Static IPv4 Configuration

IPv4 Address /

IPv4 Upstream gateway + Add a new gateway

If this interface is an Internet connection, select an existing Gateway from the list or add a new one using the "Add" button.
On local area network interfaces the upstream gateway should be "none". Gateways can be managed by [clicking here](#).

Reserved Networks

Block private networks and loopback addresses Blocks traffic from IP addresses that are reserved for private networks per RFC 1918 (10/8, 172.16/12, 192.168/16) and unique local addresses per RFC 4193 (fc00::/7) as well as loopback addresses (127/8). This option should generally be turned on, unless this network interface resides in such a private address space, too.

Block bogon networks Blocks traffic from reserved IP addresses (but not RFC 1918) or not yet assigned by IANA. Bogons are prefixes that should never appear in the Internet routing table, and so should not appear as the source address in any packets received.
Note: The update frequency can be changed under System->Advanced Firewall/NAT settings.

Figura 4 - Configuração Gateway Interface WAN

Aplicando tal configuração, será criada uma regra automaticamente de roteamento, onde qualquer requisição vinda da interface LAN será redirecionada para o *gateway* padrão, assim, liberando conexão com internet a qualquer computador conectado a rede local, a regra pode ser observada acessando ao menu principal **Firewall | Rules | LAN**, onde é visto as configurações da interface recém-criada, é notável a regra onde se bloqueia requisição de solicitações as redes privadas, caso adicionada ao momento da configuração de conexão.

2.7.2. DHCP

Como o objetivo o *firewall* é ficar ao meio da conexão externa e interna, para ter a comunicação com a LAN, é necessário ter os parâmetros setados nos terminais, os quais pode ser definidos manualmente, ou através do DHCP, o qual facilita o trabalho de configurar várias estações de trabalho. Para ativar a função de DHCP no *pfsense*, é muito simples basta acessar no menu principal **Services | DHCP Server | LAN**, e então definir o *range* (alcance

de disponibilidade para a interface), como por exemplo 192.168.2.50 – 192.168.2.99, e então salvar e aplicar as configurações, assim, enviando endereços disponíveis aos clientes, e evitando problemas de duplicação.

A configuração padrão de DHCP, disponibiliza um IP disponível, assim, um mesmo computador desconectar da rede, e depois se conectar novamente, fazendo uma nova solicitação de conexão, pode-se receber um endereço diferente, com isso o ideal é ter um endereço estático para as máquinas, assim ao criar uma regra de redirecionamento para um computador, ou de bloqueio, não sendo necessário se preocupar com o endereço que está usando, pois assim sempre utilizará o mesmo, assim ativamos tal função acessado **Status | DHCP leases**, e vemos todos os computadores que fizeram requisições de conexão, e então selecionar o “+” encontrado na frente da requisição, assim criando uma reserva de IP para o endereço de MAC do solicitante.

Com o DHCP disponibilizando as configurações da rede, deve ter noção de quais informações são enviadas, como o DNS, que pode ser informado de duas formas, uma o **DNS forwarder**, onde se habilitado, transforma o próprio servidor do *pfSense* como também o servidor de DNS, e o outro modo é quando a opção anterior não está habilitada, assim tornando necessário especificar os servidores de DNS em **General Setup**, o mesmo ocorrendo para o *gateway*, nome de domínio, ou até mesmo utilizando um servidor específico para o DHCP, onde é necessário criar um DHCP *Relay*, onde todas as requisições são direcionadas ao servidor responsável por tal configuração, e desativando a função da interface.

Services / DHCP Relay

DHCP Relay Configuration

Enable Enable DHCP relay on interface

Interface(s) LAN

Interfaces without an IP address will not be shown.

Append circuit ID and agent ID to requests
If this is checked, the DHCP relay will append the circuit ID (pfSense interface number) and the agent ID to the DHCP request.

Destination server 192.168.1.2 Delete Add

This is the IP address of the server to which DHCP requests are relayed.

Figura 5 - Redirecionamento Servidor DHCP

2.7.3. DNS

Na maioria dos casos o DNS é fornecido pelo próprio provedor da rede WAN, o que acaba não tornando recomendado a ser utilizada, pois saber a procedência do servidor de DNS a utilizar, pode estar se evitando modificações de dados, redirecionamentos, e ataques a navegação dos serviços. Para fazer tal mudança dentro do *PFsense*, acessar **System | General Setup** e especificar o endereço do servidor DNS, seja ele publico, ou de um servidor interno específico para executar a função, e então desmarcar a opção **Allow DNS server list to be overridden by DHCP/PPP on WAN**, pois se estiver habilitado pegara o servidor oferecido pela rede WAN. Uma excelente alternativa é a utilização do *DNS Forwarder* da própria ferramenta, onde permite resolver os pedidos de conexão utilizando o *hostname* obtido pelo serviço de DHCP, ou manualmente quando inseridas, ou ate mesmo encaminhar todas as solicitações para um domínio em específico.

2.8. OUTRAS CONFIGURAÇÕES

Com o *pfSense* e as configurações básicas, é possível utilizar suas outras funcionalidades, e umas dos principais para facilitar as configurações é a criação de um alias, onde oferece uma separação entre os dados das regras, como grupos de IP's e portas, recomendando sua utilização para facilitar uso das regras de *firewall* e roteamento. Para criar um alias basta acessar **Firewall | Aliases**, e definindo um IP, ou *host*, rede, URL ou até mesmo um grupo deles, assim, facilitando ao lugar de criar várias regras iguais para vários destinos, é necessário criar apenas uma, tendo um ganho na quantidade de regras a serem criadas, e ainda adicionar uma descrição a cada um, assim criando maior flexibilidade para necessidades de mudanças futuras, além de dar a possibilidade de adicionar um alias dentro de um alias, ou até mesmo importar os endereços de uma lista.

2.8.1. ROTEAMENTO / REDIRECIONAMENTO

Uma regra de roteamento, pode variar entre muito simples a muito complexa, mas a ideia básica é pegar uma entrada e encaminhá-la a um destino, e tal função é feita através de um NAT, assim acessando **Firewall | NAT**, e adicionando uma nova regra. É importante saber que ao ter uma regra criada, onde oferece passagem direta para um destino através da porta, e uma brecha esta criada, por isso é importante conhecer o quanto é seguro o serviço que está deixando em aberto, principalmente se não reservado as origens de acesso, e também é essencial conhecer o protocolo de comunicação que está usando, por exemplo deixar um acesso de TS (Terminal Service) em aberto com porta padrão, para qualquer origem, e extremamente perigoso pois se trata de um serviço nada seguro.

Para criar uma simples regra é extremamente simples, basta habilitar a regra, escolhendo a origem, LAN, WAN, ou qualquer outra interface presente, o protocolo de uso, podendo ser TCP, UDP, ICMP, ESP, AH, GRE, IPV6, IGMP, PIM, ou OSPF, então definir fonte da informação (a qual é opcional, mas recomendada para refinar a regra), as portas de entrada, destino, portas de destino, assim disponibilizando entradas de serviços como FTP, HTTP, HTTPS, IMAP, PPTP, entre outras, além de ser possível vincular a nova regra com uma já existente. É importantes atentar-se com a ordem que se encontra as regras, pois a ferramenta as lê as NATs de cima para baixo, assim uma regra pode sobrescrever outra, e acabando interferindo em sua funcionalidade.











Rules										
	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
<input type="checkbox"/>	 	INTERNET	TCP	*	*	INTERNET address	3389 (MS RDP)	*	Bloqueia TS	  
<input type="checkbox"/>	 	INTERNET	TCP	*	*	WebService	443 (HTTPS)	172.254.1.78 443 (HTTPS)	Roteamento Servidor WEB	  

Figura 6 - Regras Genéricas de Roteamento

Por padrão para cada NAT criado, também é criada uma regra de *firewall*, mas ambas são distintas, onde pode ocorrer de um NAT estar desabilitado, mas a regra *firewall* habilitada, assim o caminho ainda em aberto permitindo o tráfego, ou pode se ter uma NAT habilitado e o *firewall* o bloqueando. Seu funcionamento deve obedecer aos seguintes critérios, Interface, Protocolo, Origem e seus intervalos de porta, e destino, e seus intervalos de portas, assim se tem uma regra criada para criar um roteamento.

2.8.2. FIREWALL

Criar uma regra de *firewall*, seria como criar um buraco em um muro, onde tem uma pequena passagem para o outro lado, assim quando criado essa abertura é importante estar sempre a monitorando, para que assim não tenha a passagem por ela de algo que não deveria. Para criar a regra assim como criar um redirecionamento é simples, sendo necessário dizer a ferramenta os parâmetros necessários, indo até **Firewall | Rule | WAN** e adicionar uma nova regra, especificando a WAN a ser utilizada, o Protocolo TCP, a origem e sua porta, destino e porta, e a regra está criada. Todo tráfego é checado dentro da lista de regras, e quando algum pacote corresponder a uma regra, a mesma é executada, podendo ser permitida a passagem, bloqueada, ou devolvida ao remetente.

Alguns dos recursos avançados do *pfSense* pode verificar a fonte de dados de acordo com o sistema operacional, através do **Source OS**, priorizar tráfego com base em valores com o **Diffserv Code Point**, tratamento de pacote, mecanismo de rastreamento, entre outros serviços. Cada regra criada, não segue nenhum tipo de padrão, cabe a quem o configura saber da necessidade da regra e de como será tratada, podendo variar seu estado e a forma que será utilizada.

2.9. DEMAIS RECURSOS

O *pfsense*, é uma ferramenta repleta de funcionalidades, e como o objetivo desse trabalho não é os explorar, mas sim mostrar seus aspectos básicos, e testar sua segurança, mas pode utilizar outros recursos como:

- Redes Privada (VPN);
- IP Virtual;
- Rotas Estáticas;
- QoS;
- Ponte Bridge;
- LAN Virtual;
- Captive Portal;
- Redundância e Balanceamento de Trafego;
- Monitoramento e Registros da Ferramenta;

3. PENTEST

Neste capítulo, é falado do que trata-se o *pentest*, as etapas para sua execução, e a importância de executar o mesmo, principalmente pelo impacto que pode impactar se passado por uma infraestrutura, ocorrendo danos a mesma.

O *pentest* é um dos meios utilizados para encontrar as vulnerabilidades de uma infraestrutura, simulando ataques, encontrando falhas e as explorando. Também utilizado para validar as táticas de segurança adotadas como mecanismo de defesa, e entender as consequências das falhas, e com seus resultados adotar medidas de correção.

Para os mecanismos de defesa VIEGAS (2016) diz que “segurança total e irrestrita contra crimes virtuais (pelo menos hoje) não passa de um sonho. Contudo, é possível testar a solidez do sistema, detectar falhas e criar barreiras que desencorajem e minimizem o impacto desse tipo de ação.” Assim, o intuito é diminuir o número de brechas e entender o quanto exposto esta a rede para o mundo fora do domínio local.

A realização de um *pentest*, necessita de persistência, principalmente quando deparam com uma estrutura com critérios de segurança, onde é necessário identificar erros, não para sabotar dados, mas documentar os processos e testes, com a finalidade de realizar modificações para correção de erros. É muito importante conhecer como funciona um ataque, para que assim possa buscar a melhor forma de fechar o ambiente de infraestrutura, mas deixar acessível para as necessidades, tornando um desafio a disponibilidade dos recurso e a alta segurança para a informação. Para realizar um *pentest*, é necessário o dividir em etapas, assim cada etapa, pode ser definido como um nível de penetração, e assim se modelando o teste a ser feito conforme a situação a se deparar, resultante a uma organização para os esforços.

A primeira fase dos passos da invasão, se dá pelas primeiras interações, onde conscientiza os envolvidos, sobre as expectativas e as possibilidades dentro do escopo do projeto. De início se tenta levantar o máximo de informações do alvo, utilizando um método chamado de engenharia social, tentando conseguir dados e informações uteis, como e-mails, informações de rede, links e documentos. Com as primeiras informações, avança mais uma camada no processo de invasão, e a fase de reconhecimento inicia, tentando usar a transferência da zona DNS, para a replicação de dados, obtendo mais informações do local, e quebrando uma das maiores vulnerabilidades que pode se encontrar em uma rede.

A Segunda fase, é iniciada pelo rastreamento de portas, identificando quais estão abertas, e quais serviços estão sendo executados nas mesmas. Cada *socket* se tratando de um serviço, onde obtêm-se uma melhor visão do alvo, sempre buscando atacar as portas mais conhecidas, pois pode acarretar a avisar uma aplicação específica sobre a suposta invasão, e assim pode ter como efeito o fechamento da porta.

Cada fase que passa, mais informação tem recolhido, assim podendo buscar mais vulnerabilidades e falhas de um programa ou serviços, fragilizando os servidores, assim realizando uma bateria de ataques, buscando cada vez mais falhas. O processo de escaneamento de erros realiza-se de forma totalmente automatizada, com ferramentas próprias para esse tipo de serviço, onde Segundo (Kennedy et al., 2011), vários sistemas operacionais tendem a responder diferentemente quando é feita a sondagem da rede por causa das diferentes implementações de rede em uso.

A fase final é dada pela exploração, onde se tenta o acesso não autorizado a sistema do alvo, tomando a forma do administrador do sistema, usando métodos de *exploit* e força bruta através das falhas encontradas nas análises, quebrando assim todo o esquema de segurança. O resultado dos ataque pode ser categorizado em 4 tipos, sendo eles interceptação, modificação, interrupção e a personificação, cada um deixando uma consequência a seu alvo.

Tendo em vista os aspectos percebe a importância de *pentesting* para uma empresa ou corporação, e mesmo com leis e divisões de crimes praticados através de computadores os proprietários preferem pagar para manter seus dados seguros. Os *pentesting* fazem uso das mesmas ferramentas que um cracker utiliza para fazer invasão não autorizada, porém, visam trabalhar para proteger empresas ou corporações de possíveis invasões que só levariam as mesmas a grandes perdas das mais variadas formas, inclusive de credibilidade na área em que trabalha e com seus clientes. (SILVA; PEREIRA, 2013, 5 p.)

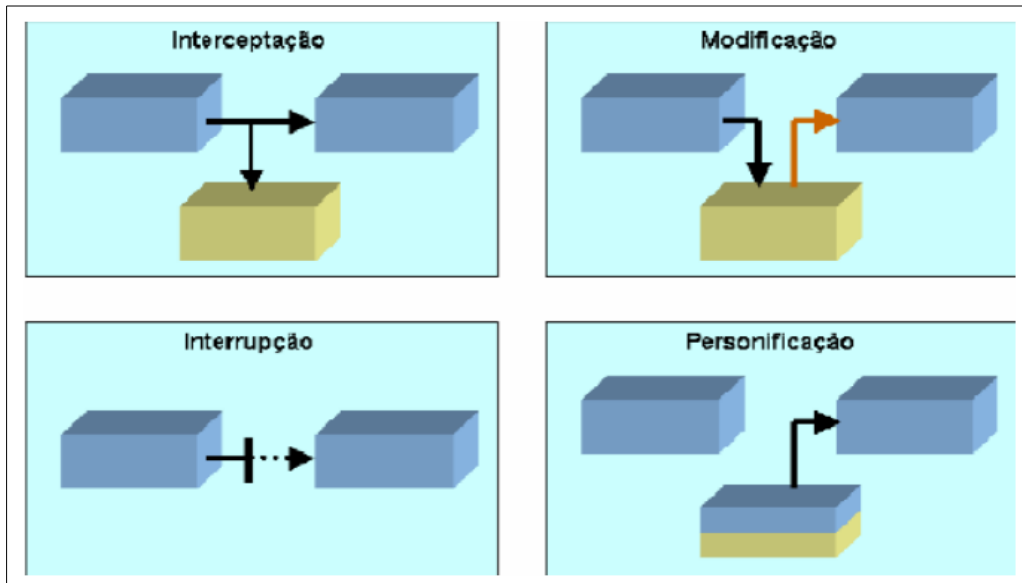


Figura 7 - Modificação da Informação

Fonte: <http://web.unipar.br/~seinpar/2013/artigos/Raquel%20Fonseca%20da%20Silva.pdf>

4. AMBIENTE DE TESTES

Esta etapa do trabalho será mostrado como foi estruturado o ambiente de teste, para que assim coloque a prova a ferramenta, e a eficiência da mesma, especificando o material usado para sua montagem, e a estrutura lógica com o posicionamento da máquina de segurança.

Para a realização deste trabalho foi montado um ambiente de teste, assim, podendo pôr a prova as idéias apresentadas. O laboratório de testes ira ter dois servidores, cada um distinto do outro, executando serviços diferentes, assim podendo pôr a prova os protocolos de cada um. Como ponto de entrada se localiza o *PFSense*, assim podendo realizar sua função, como *firewall*, e gerenciador de rede, dividindo o ambiente dentre LAN e WAN, disponibilizando assim conexão aos servidores. Como a ideia é usar apenas o *log* da própria ferramenta para registro de informação, será feito alguns ajustes para tentar diminuir a quantidade de informação registrada, e para se facilitar a análise de informações, mantendo o que foi passado pela regra de portas e descartando o que for bloqueado (dentro do registro), onde deduz, o que não obteve êxito de entrada, logo foi vetada pela aplicação, onde ao ocorrer alguma falha de segurança, encontra-se por onde foi a brecha, já que só haverá registro de regras de passagem.

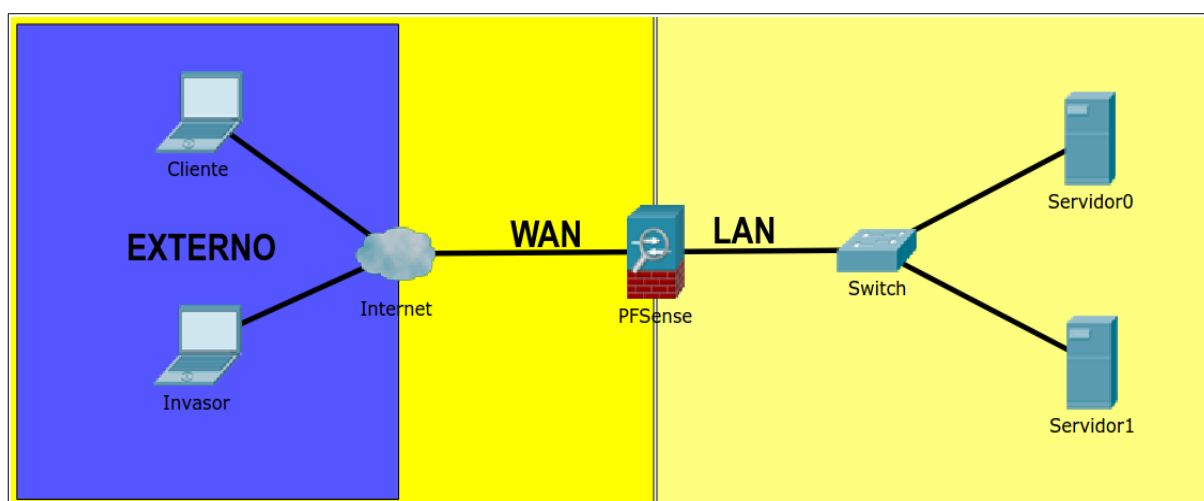


Figura 8 - Laboratório

Na figura acima, encontra-se o ambiente para o desenvolvimento do trabalho, onde encontra-se o posicionamento estratégico do *pfsense*, e as 2 principais redes, a LAN com os servidores

locais, e a WAN, onde se tem o cliente que deseja acessar o local, mas também o invasor, sendo necessário colocar a testes de disponibilidade e segurança em sua fase de execução.

Para se ter o registro das passagens, será editado cada regra de *firewall*, habilitando a opção de registro de *log* das regras de redirecionamento e abertura de portas, o qual vem por padrão desabilitado, e será executado os primeiros testes de acesso simples para verificar a autenticidade dos registros, e comparados com as ações executadas, pois registros de acesso, necessitam de paciência aos analisar, onde geram grande quantidade de informação, as quais nem sempre são todas necessárias, tendo uma vez qualquer ocorrência a ferramenta, se cria um registro da ação, também torna os arquivos muito grandes, e mesmo assim necessários para a auditoria de ações tomadas.

PFSense:

Sistema Operacional: FreeBSD 10 + PFSense 2.3.

Hardware: Processador Intel Celeron 2.66GHz, Memória RAM 1,5 Gb, Swap 4 Gb, Partição de disco “/” arquivos de configuração, e arquivos de registro, “/var/run” carregamento do sistema em execução.

Serviços: NAT + Firewall, com possibilidades de mudança conforme testes, Gateway dedicado + DNS publico da google, monitoramento de trafego, e registro de log.

Servidor 0

Sistema Operacional: Windows XP SP3.

Hardware: Processador Intel Celeron 2.66GHz, Memória RAM 1 Gb.

Serviço: Microsoft Terminal Service.

Servidor 1

Sistema Operacional: GNU/LINUX, Xubuntu.

Hardware: Processador Intel Celeron 2.66GHz, Memória RAM 1 Gb.

Serviço: Apache2 + PHP5.

O modo de acesso é feito via internet, com endereço de IP fixo, e com o serviço de DNS da própria instituição de ensino, o qual fornecerá a conexão, como também as máquinas para a realização do trabalho, assim transformando o cenário o mais real possível, como em uma situação real de funcionamento de alguns órgão de execução de serviços computacionais, que necessite de uma infraestrutura de disponibilidade de acesso externo.

5. TESTES DE DISPONIBILIDADE DO AMBIENTE

Com o ambiente pronto, neste capítulo será validado o ambiente verificando a disponibilidade de acesso externo do mesmo, além de tentar encontrar suas vulnerabilidades e as explorar, e ao final propor melhorias, para o ambiente se tornar mais seguro.

Ao montar toda a estrutura, se inicia os primeiros testes, colocando a prova a autenticidade do que foi montado, e observar os primeiros resultados que podem ser coletados, em um funcionamento convencional, sem ter a interferência de nenhum teste de sobrecarga de sistema.

Dentro da ferramenta são criadas duas regras iniciais de *firewall*, onde é liberado as portas 80 (HTTP), e a porta 7001, para a interface WAN para qualquer origem utilizando conexão TCP, e duas regras de NAT, onde faz um redirecionamento da porta HTTP para o servidor 1, e um redirecionamento da porta 7001 para a 3389 (MS RDP) para o servidor 0, as demais portas são bloqueadas por padrão pela ferramenta, seguindo o princípio, o que não está liberado é rejeitado.

Nos primeiros testes, foi feito três verificações, onde pelos princípios de configuração os resultados são ser dois como sucesso, onde se testa serviços que estão liberados, e um obtido como sem êxito, visto que será um protocolo de rede onde não se tem a liberação de usabilidade. Os acessos desta primeira, busca acessar a uma página WEB posta ao servidor 0, e o acesso remoto liberado ao servidor 1, e por final será feito um *ping* ao endereço destino, o qual espera que não se tenha uma resposta.

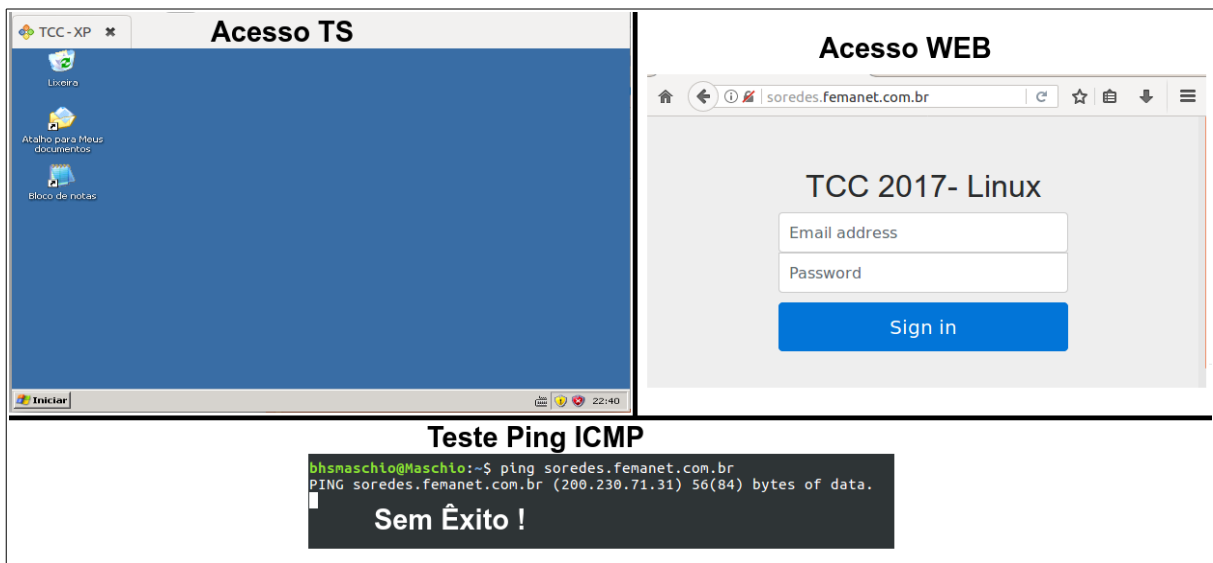


Figura 9 - Testes de Disponibilidade

De acordo com o esperado dos resultados, são gerados, onde teve acesso do que estava liberado, e teve um acesso vetado diante o que não se devia, assim espera que os registros, sejam coerentes com os resultados, colocando a prova uma função básica e essencial da ferramenta, pondo a prova a autenticidade dos registros. Mesmo a ideia inicial de monitoramento de *log* é apenas ver as regras de *pass* será habilitado por um instante o *log* de portas bloqueadas, para ver mais uma vez a autenticidade dos registro, logo que o acesso de ICMP não obteve êxito, após isso será voltado a configuração inicial e dará continuidade aos testes.

Action	Time	Interface	Rule	Source	Destination	Protocol
✘	Jul 12 15:39:24	WAN	Default deny rule IPv4 (1000000103)	177.55.62.253	200.230.71.31	ICMP
✘	Jul 12 15:39:25	WAN	Default deny rule IPv4 (1000000103)	177.55.62.253	200.230.71.31	ICMP
✘	Jul 12 15:39:26	WAN	Default deny rule IPv4 (1000000103)	177.55.62.253	200.230.71.31	ICMP
✔	Jul 12 21:29:35	WAN	NAT Apache Xubuntu (1494016575)	189.51.132.19:36218	192.168.1.101:80	TCP:S
✔	Jul 12 21:29:35	LAN	let out anything IPv4 from firewall host itself (1000002665)	189.51.132.19:36218	192.168.1.101:80	TCP:S
✔	Jul 12 21:32:20	WAN	let out anything from firewall host itself (1000002761)	200.230.71.31:123	200.160.7.193:123	UDP
✔	Jul 12 21:36:36	WAN	let out anything from firewall host itself (1000002761)	200.230.71.31:123	200.160.7.193:123	UDP
✔	Jul 12 21:40:10	WAN	NAT Apache Xubuntu (1494016575)	89.109.42.242:2188	192.168.1.101:80	TCP:S
✔	Jul 12 22:08:08	WAN	NAT TS Servidor Windows TCC (1494009188)	189.51.132.19:46622	192.168.1.100:3389	TCP:S

■ ICMP
 ■ MS RDP
 ■ HTTP

Figura 10 - Resultado Em Log

Na figura acima, é observado que foi registrado as ações, com o seu resultado de permitir ou liberar, distinguindo cada uma pelo nome da regra, porta e protocolo utilizado, tornando-se de fácil entendimento para quem as analisa.

Para realizar uma validação final do ambiente, será verificado via *terminal linux* os resultados obtidos através do comando *nslookup*, onde o comando é usado tanto por administradores de rede, para realização de testes de conexão, podendo ter o retorno de informações da rede, e conferência de informação, mas também acaba sendo usado para analisar a infraestrutura com intuítos de ataque, pois a função facilita o levantamento de informações, e toma-se alguns conhecimentos do destino, ao visualizar os resultados do comando, obtendo informação do endereço IP do destino, como controladores do domínio DNS, assim tendo melhor extração de disponibilidade.

Com o resultado do *nslookup*, pode-se realizar uma varredura de portas vendo as que estão disponíveis, tanto como validação, como para uma possível invasão. Com o resultado das portas abertas realizadas pelo *scan* pode-se concluir que demais portas permanecem fechadas, dando confiabilidade no que está sendo feito, e que nenhuma porta será liberada independentemente.

```
bhsmaschio@Maschio:~$ nslookup soresdes.femanet.com.br
Non-authoritative answer:
Name:   soresdes.femanet.com.br
Address: 200.230.71.31

bhsmaschio@Maschio:~$ nslookup
> set q=ns
> soresdes.femanet.com.br
Authoritative answers can be found from:
femanet.com.br
    origin = monica.femanet.com.br
    mail addr = root.femanet.com.br

bhsmaschio@Maschio:~$ nmap 200.230.71.31
Host is up (0.084s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
7001/tcp  open  afs3-callback
```

Figura 11 - Informações de Rede

Na imagem acima, foi realizado testes de tradução de nomes de *DNS* para endereços IP, podendo chegar a um contato mais certo ao alvo, onde não teria algum desvio de rota, além de buscar a qual grupo de zona de endereços *DNS* ele pertence, e as portas que estão liberadas, estando abertas para acesso.

Apesar dos acessos estarem funcionando, é importante saber que as regras criam acessos diretos para os servidores destino, pelas portas especificadas, sendo importante saber que tipo de serviço está sendo liberado, e como é a segurança do mesmo, pois por utilizar um *firewall*, não significa nenhum sinal de segurança completa, pois a falta de ciência das configurações que estão setadas, não estiverem visando sempre como prioridade a segurança, pode se ter um resultado inesperado. Em muitos casos, são necessários criar algumas camadas a mais de defesa, ou até mesmo isolar áreas para manter um ambiente mais seguro. Os próximos testes têm o intuito de mostrar que apenas as configurações que foram realizadas não são o suficiente, tendo em vista principalmente que está liberado um serviço de acesso remoto, que ao ser penetrado, pode acessar com mais facilidade terminais da rede interna.

6. TESTES COM KALI LINUX BASEADAS NO AMBIENTE

O *Kali linux*, trata-se de um sistema operacional equipado com ferramentas para testes de invasão, facilitando a realização das ações, assim veremos os resultados obtidos com alguns utilitários disponíveis no mesmo. É importante ressaltar que as conclusões do trabalho, são com base testes realizados, assim, podendo encontrar pontos fortes ou fracos com base na utilização, mas também pode acabar tendo resultados diferentes se realizados outros tipos de testes.

6.1. TESTE SERVIDOR 0 - RDP

O primeiro teste dessa etapa, foi utilizando a ferramenta *thc-hydra*, onde através de tentativas e erros tenta obter o login e senha de seu alvo, assim buscando principalmente autenticar ao serviço de acesso remoto, e permite deixar as tentativas de forma anônima, sendo um desafio ao *PFSense*, registrar as ações do mesmo. O foco do teste não é a invasão do servidor, mas sim o registro, com isso partindo do pressuposto que o invasor conhece a porta que está em aberto, mesmo por não tratar-se de uma porta padrão, e acaba tendo a informação que se trata de um RDP. Independente se a autenticação foi feita ou não, busca como o resultado capturar a ação e sua origem.

Existes outras formas de invadir um serviço de acesso remoto, com ferramentas mais poderosas, e mais eficientes, mas a ideia delas, são baseadas em um mesmo princípio, onde para a questão proposta com o *thc-hydra* se encaixa bem para os testes a ser realizado, pois utiliza um método muito comum, para este tipo de ação. A execução do comando é de forma simples, onde se chama o nome da aplicação “hydra”, seguido por um “-l” e o possível nome de login, ou “-L” para uma lista com nomes de login a ser testada, buscando sempre nomes padrões como administrador, root, ou nome do usuário, o comando continua com os parâmetros “-p” passando a senha, ou “-P” uma lista de senhas, que pode ser extraída com engenharia social, e “-s” com a porta a ser atacada, e por final o protocolo e o endereço de IP.

```

root@kali:~# hydra -l administrador -p senhateste -s 7001 rdp://200.230.71.31
Hydra v8.2 (c) 2016 by van Hauser/THC - Please do not use in military or secret service organiza
tions, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-07-17 21:35:19
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number o
f parallel connections and -W 1 or -W 3 to wait between connection to allow the server to recove
r
[DATA] max 1 task per 1 server, overall 64 tasks, 1 login try (l:l/p:1), ~0 tries per task
[DATA] attacking service rdp on port 7001
[ERROR] Child with pid 1778 terminating, can not connect
[ERROR] Child with pid 1783 terminating, can not connect
[ERROR] Child with pid 1784 terminating, can not connect
[STATUS] attack finished for 200.230.71.31 (waiting for children to finish) ...
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-07-17 21:36:00

```

Figura 12 - Teste com THC-Hydra

A quebra de senha não obteve sucesso, pois foi feita de forma muito superficial, e para se obter êxito seria necessário trabalhar em cima da invasão com mais eficácia, mas visando o resultado para com o trabalho, e voltando o olhar para o objetivo, onde se pega os registros de *log*, temos o resultado conforme a figura abaixo.

✓	Jul 17 22:35:21	WAN	NAT TS Servidor Windows TCC (1494009188)	187.85.199.195:58678	192.168.1.100:3389	TCP:S
✓	Jul 17 22:35:21	LAN	let out anything IPv4 from firewall host itself (1000003715)	187.85.199.195:58678	192.168.1.100:3389	TCP:S
✓	Jul 17 22:35:51	WAN	NAT Apache Xubuntu (1494016575)	116.10.203.153:56773	192.168.1.101:80	TCP:S
✓	Jul 17 22:35:51	LAN	let out anything IPv4 from firewall host itself (1000003715)	116.10.203.153:56773	192.168.1.101:80	TCP:S
✓	Jul 17 22:35:55	WAN	NAT TS Servidor Windows TCC (1494009188)	187.85.199.195:58680	192.168.1.100:3389	TCP:S
✓	Jul 17 22:35:55	LAN	let out anything IPv4 from firewall host itself (1000003715)	187.85.199.195:58680	192.168.1.100:3389	TCP:S
✓	Jul 17 22:35:55	WAN	NAT TS Servidor Windows TCC (1494009188)	187.85.199.195:58682	192.168.1.100:3389	TCP:S
✓	Jul 17 22:35:55	LAN	let out anything IPv4 from firewall host itself (1000003715)	187.85.199.195:58682	192.168.1.100:3389	TCP:S

Figura 13 - Registro das Tentativas de Acesso Remoto

Para cada teste de autenticação feito, temos um registro de passagem, sendo observado o horário da tentativa de acesso totalmente coerente com o horário do *log*, e se sabendo exatamente a qual serviço e servidor foi designado, o suposto IP de origem da tentativa de conexão, e quantas tentativas foram realizadas. Como existe a regra de passagem, a ferramenta não realiza nenhum tipo de validação por padrão para verificar a conexão, assim, a segunda camada de segurança acaba ficando sob responsabilidade do próprio servidor de destino, diante de medidas de segurança tomadas, mas a modo mais simples para identificar um possível ataque seria observar a quantidade exagerada de requisições, pela utilização da

mesma regra no *log* do *PFSense*, assim poderia se tomar alguma medida de segurança mais ríspida para a regra da ocorrência, e buscando mais sobre a origem da ação.

Algumas medidas de segurança que podem ser tomadas neste caso, são como já realizado, utilizando uma porta sem ser a padrão dentro do *firewall*, e também alterando a porta de acesso dentro do próprio sistema operacional, assim dificultando ainda mais a descoberta de qual serviço se trata. Outra função adicional que se pode utilizar, é um segundo serviço de autenticação, necessitando de algum tipo de PIN gerado individualmente para o acesso, assim obrigando o invasor a forçar mais uma aplicação para descoberta de credenciais.

Existem medidas que podem ser tomadas dentro da própria configuração da regra de *firewall* no *PFSense*, onde pode adicionar alguns critérios para a tentativa de conexão, sendo eles, limitar por entradas de estado máximo, limitar por número máximo de *hosts* de origem, limite por número máximo de conexões estabelecidas por *host*, limite de entradas máximas de estado por *host*, máximo de novas conexões por *host*, e tempo limite do estado, e até mesmo definir horários que poder ser acessados, assim podendo criar agendamentos, e faixas de tempo para a permissão.

Uma alternativa ainda mais segura é a utilização de uma VPN, onde se cria um túnel de comunicação entre o cliente e o servidor de modo remoto, deixando a utilização da porta fechada, e travando uma acesso criptografado entre os terminais, assim mantendo credenciais de autenticação através de certificados para a troca de informação. Segundo Martins (2015), os processos para envio de dados em uma VPN segue seis etapas, onde o primeiro os dados são criptografados e encapsulados, então no segundo passo são adicionados dados extras na mensagem, como o número de IP da máquina remetente, podendo assim o receptor identificar quem mandou o pacote, e então no terceiro passo os dados são enviados pelo túnel de conexão, para que no quarto passo a máquina receptora inicie a verificação dos dados que recebeu e então no quinto passo os dados são realmente recebidos e desencapsulados, e finalmente no sexto passo os dados são descriptografados e armazenados no computador.

6.2. TESTE SERVIDOR 1 - HTTP

Para esta segunda etapa do teste, existe um desafio ainda maior, pois uma página web, não pode simplesmente criar um acesso exclusivo para uma determinada origem, pois ele precisa estar sempre estar disponível em qualquer hora e lugar, mesmo que seu acesso tenha métodos

de autenticação via código sobre ela, a mesma encontra-se rodando em um *web service*, que possui também suas rotinas de segurança, mas isto não basta para a manter sua estrutura ativa. A dificuldade para esse tipo de protocolo é com a identificação de um ataque, pois a cada atualização da página tem uma passagem, assim um registro, sendo difícil diferenciar o que se trata de um acesso comum de uma tentativa de invasão.

Para a realização dos testes para esse protocolo, foi utilizado a ferramenta *Nikto*, que exerce a função de *scanner* de vulnerabilidades, buscando arquivos, configurações e programas que estão sendo executados, assim tornando um auxiliar para outras ferramentas destinados a derrubar a página. Para entender melhor os resultados trazidos pela ferramenta, é necessário acessar um OSVDB, onde encontra-se o explicativo de cada falha encontrada, baseado no *report* gerado. A utilização da ferramenta torna-se ainda mais completa quando utilizada em conjunto com um serviço de *proxy* anônimo, a fim da origem do ataque não serem descobertos.

Para utilizar o *Nikto*, basta utilizar o comando “*nikto -h*” e passar o host a ser analisado, também podendo ser adicionado o comando “*-p*” com uma ou mais portas a serem averiguadas, tentando encontrar alguns serviços ocultos sendo executados, a ter uma possibilidade também de analisar URLs completas. Com base nos resultados serão entendidas as falhas e com base nisso, poderão ser utilizadas ferramentas específicas para explorar o erro, além de buscar informações sobre o serviço encontrado e como o forçar a um erro.

Apos a análise os resultados que serão obtidos de volta, existem outros serviços sendo executados na página, como de *smtp* para e-mail ou *ftp* para arquivos, onde ocorre conseqüentemente mais interações do usuário com o servidor, assim podendo ser encontrado ainda mais vulnerabilidades, principalmente se existe alguns um banco de dados para com a estrutura, assim existe uma maior importância a quem quer preservar informação. A página criada para este trabalho é extremamente simples, contendo apenas uma interface inicial de login, um validador, uma página de sucesso, e um validador de seção, assim por ser de pequeno porte, quase não se encontrara muitas brechas, onde se tentara explorar os recursos oferecidos pelo *apache* (web service), e pelo PHP que mantém.

Ao realizar o comando, é possível salvar seus resultados em um “.txt”, assim ao analisar o ambiente deste trabalho, se obteve o seguinte resultado:

```

- Nikto v2.1.6
-----
+ Target IP: 200.230.71.31
+ Target Hostname: soredes.femanet.com.br
+ Target Port: 80
+ Start Time: 2017-07-21 11:12:48 (GMT-4)
-----
+ Server: Apache/2.4.18 (Ubuntu)
+ Retrieved via header: 1.1 Firewall (squid/3.5.26)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to
protect against some forms of XSS
+ Uncommon header 'x-cache-lookup' found, with contents: MISS from Firewall:3128
+ Uncommon header 'x-cache' found, with contents: MISS from Firewall
+ The X-Content-Type-Options header is not set. This could allow the user agent to render
the content of the site in a different fashion to the MIME type
+ Cookie PHPSESSID created without the httponly flag
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server banner has changed from 'Apache/2.4.18 (Ubuntu)' to 'squid/3.5.26' which may
suggest a WAF, load balancer or proxy is in place
+ Uncommon header 'x-squid-error' found, with contents: ERR_INVALID_URL 0
+ Web Server returns a valid response with junk HTTP methods, this may cause false
positives.
+ Server leaks inodes via ETags, header found with file /icons/README, fields: 0x13f4
0x438c034968a80
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7553 requests: 0 error(s) and 11 item(s) reported on remote host
+ End Time: 2017-07-21 11:18:37 (GMT-4) (349 seconds)
-----
+ 1 host(s) tested

```

Figura 14 – Resultado Nikto

Na imagem acima, temos o relatório de erros gerados na página, onde com base em uma tabela de informações, é tomado consciência de todos os pontos falhos, tornando possível uma invasão.

Dependendo do resultado que tem de retorno, consegue-se acesso via URL a arquivos de autenticação, e páginas não validadas, assim existindo uma brecha na programação, e consequentemente onde tem uma manipulação do invasor, e a alteração de dados. Este é apenas um passo para o *pentest* dentro de um sistema web, assim com os resultados, sabe-se mais ainda do alvo, e busca suas vulnerabilidades para correção, mas mesmo podendo ir a fundo nas falhas, podendo tomar algumas outras medidas básicas de segurança para esse tipo de acesso, onde reforça a estrutura.

Uma medida básica de segurança para páginas HTTP, é as utilizar junto com um certificado digital, implementado sob uma camada SSL ou TLS, tendo o HTTPS, onde os dados são criptografados e transmitidos de uma forma mais segura, e a partir disto então a página web é

executada na porta 443, assim evitando que as informações possam ser observadas por terceiros.

De acordo com a OWASP Foundation, dados de 2007, essas são as 10 vulnerabilidades de segurança mais críticas em aplicações WEB.

A1 – Cross Site Scripting (XSS)	Os furos XSS ocorrem sempre que uma aplicação obtém as informações fornecidas pelo usuário e as envia de volta ao navegador sem realizar validação ou codificação daquele conteúdo. O XSS permite aos atacantes executarem scripts no navegador da vítima, o qual pode roubar sessões de usuário, pichar sites Web, introduzir worms, etc.
A2 – Falhas de Injeção	As falhas de injeção, em especial SQL Injection, são comuns em aplicações Web. A injeção ocorre quando os dados fornecidos pelo usuário são enviados a um interpretador com parte do comando ou consulta. A informação maliciosa fornecida pelo atacante engana o interpretador que irá executar comandos mal intencionados ou manipular informações.
A3 – Execução maliciosa de arquivos	Os códigos vulneráveis à inclusão remota de arquivos (RFI) permite ao atacante incluir código e dados maliciosos, resultando em ataques devastadores, como o comprometimento total do servidor. Os ataques de execução de arquivos maliciosos afeta PHP, XML e todos os frameworks que aceitem nomes de arquivo ou arquivos dos usuários.
A4 – Referência Insegura Direta a objetos	Uma referência direta à objeto ocorre quando um desenvolvedor expõe a referência a um objeto implementado internamente, como é o caso de arquivos, diretórios, registros da base de dados ou chaves, na forma de uma URL ou parâmetro de formulário. Os atacantes podem manipular estas referências para acessar outros objetos sem autorização.
A5 – Cross Site Request Forgery (CSRF)	Um ataque CSRF força o navegador da vítima, que esteja autenticado em uma aplicação, a enviar uma requisição pré-autenticada à um servidor Web vulnerável, que por sua vez força o navegador da vítima a executar uma ação maliciosa em prol do atacante. O CSRF pode ser tão poderoso quanto a aplicação Web que ele ataca.
A6 – Vazamento de Informações e Tratamento de Erros Inapropriado	As aplicações podem divulgar informações sobre suas configurações, processos internos ou violar a privacidade por meio de uma série de problemas na aplicação, sem haver qualquer intenção. Os atacantes podem usar esta fragilidade para roubar informações consideradas sensíveis ou conduzir ataques mais estruturados.
A7 – Autenticação falha e Gerenciamento de Sessão	As credenciais de acesso e token de sessão não são protegidos apropriadamente com bastante frequência. Atacantes comprometem senhas, chaves ou tokens de autenticação de forma a assumir a identidade de outros usuários.
A8 –	As aplicações Web raramente utilizam funções criptográficas de forma

Armazenamento Criptográfico Inseguro	adequada para proteção de informações e credenciais. Os atacantes se aproveitam de informações mal protegidas para realizar roubo de identidade e outros crimes, como fraudes de cartões de crédito.
A9 – Comunicações inseguras	As aplicações frequentemente falham em criptografar tráfego de rede quando se faz necessário proteger comunicações críticas/confidenciais.
A10 – Falha de Restrição de Acesso à URL	Frequentemente, uma aplicação protege suas funcionalidades críticas somente pela supressão de informações como links ou URLs para usuários não autorizados. Os atacantes podem fazer uso desta fragilidade para acessar e realizar operações não autorizadas por meio do acesso direto às URLs.

Tabela 1 – Possíveis Vulnerabilidades de Uma Pagina WEB

Fonte: https://www.owasp.org/images/4/42/OWASP_TOP_10_2007_PT-BR.pdf

Em geral para manter uma página sempre disponível, é necessário manter uma rotina de infraestrutura muito regular, realizando contentemente *scans* de *malware*, análise de servidor, rotinas de *pentest*, e um rigoroso monitoramento de tráfego de rede, sendo assim não existe uma fórmula certa para manter a página intocável, mas sim um acompanhamento de seu funcionamento.

7. OUTRAS REGRAS DE SEGURANÇA

Além do que já foi desenvolvido, neste capítulo será abordado outros aspectos importantes, que são necessários ter dentro de um ambiente, para que assim o mesmo seja mais seguro, utilizando ferramentas que são suportadas nativamente pelo *pfsense*.

Como o *PFSense*, é apenas um adicional nas regras de segurança, existem outras medidas muito simples que podem ser adicionadas a rede para ter ainda uma maior precaução para caso ocorra uma invasão. Algumas dessas medidas comportam nativamente dentro da ferramenta, o que de certa forma economiza recursos, com a utilização da mesma, e outras funções, são aplicadas de forma lógica não necessitando de um hardware.

É claro que para tomar alguma medida dentro da infraestrutura é necessário conhecer a necessidade de implantação, sendo uma medida inicial, onde deve ter uma análise previa da disponibilidade de recursos a ser implementada, assim vindo a necessidade de adicionar recursos ao projeto, como um exemplo, adicionar mais interfaces de rede ao *pfsense*, para

ocorrer uma divisão de estruturas, não limitando apenas a áreas de WAN e LAN, mas tendo redundâncias para conexão, para se evitar quedas, e as subdivisões da rede interna, para promovendo melhor gerenciamento, e maior segurança.

7.1. SUB-REDE

Um sub-rede é utilizada para dividir uma rede computacional, resultando em tráfego reduzido, melhor performance, aplicando tal ação apenas modificando a máscara de rede. Ao se aplicar tal ação existe um grande ganho parte de segurança, onde se ocorrer alguma falha, esta fica isolada do restante das áreas, pois não há comunicação de uma área com a outra, assim um potencial invasor se conseguir um acesso a um terminal, acaba ficando preso aquela sub-rede, assim cria uma maior dificuldade a chegar ao servidor. Aplicar uma sub-rede é de forma simples, pois é necessário apenas fazer o cálculo proporcional ao ambiente, e reconfigurar o DHCP, onde o recurso já é oferecido pelo próprio *PFSense*.

7.2. VLAN

Apesar de utilizar uma sub-rede tem suas vantagens, e sua aplicação é bem simples, a melhor forma de isolar áreas é através das VLANs (Rede Local Virtual), mas a sub-rede ainda se torna necessário, por causa do potencial de crescimento da rede, assim podendo adicionar mais terminais, mas ao subdividir com VLANs pode-se agrupar máquinas de forma lógica, criando critérios, como tipo de tráfego, grupos, ou departamentos, e a comunicação entre elas é possível através de um roteamento, mas sendo possível também filtrar o conteúdo, assim criando uma segurança, pois pode se limitar o acesso aos servidores, podendo controlar por VLANs, ou por máquinas específicas todo o tráfego, sendo gerenciável, onde controlar pacotes e protocolos torna mais fácil, e o mais importante, monitorando as ocorrências.

De acordo com Varreira todas essas redes virtuais seriam distintas umas das outras mesmo pertencendo a mesma rede interna, a conexão entre elas deverá ser feita através de um *firewall* que aplicaria as regras de acesso e as políticas de segurança de uma rede à outra. Essa subdivisão é importante para a segurança, pelo mesmo motivo que uma sub-rede, o isolamento de uma área, já é algo essencial, onde ver a infraestrutura de maneira segmentada, e poder tratar as regras por partes, e ter áreas exclusivas para os servidores, pode-se resultar em um tratamento mais aguçada para setores mais críticos, sem interferir nas demais. Para

entender melhor a estrutura de uma VLAN, observado a figura 14, vemos que temos 5 separações físicas (andares do prédio), mas temos máquinas mesmo fisicamente distantes, pertencentes a mesma estrutura lógica (VLAN), assim reorganizando a rede, e fazendo com que cada máquina pertencente ao mesmo grupo seja classificado como pertencente a mesma localização.

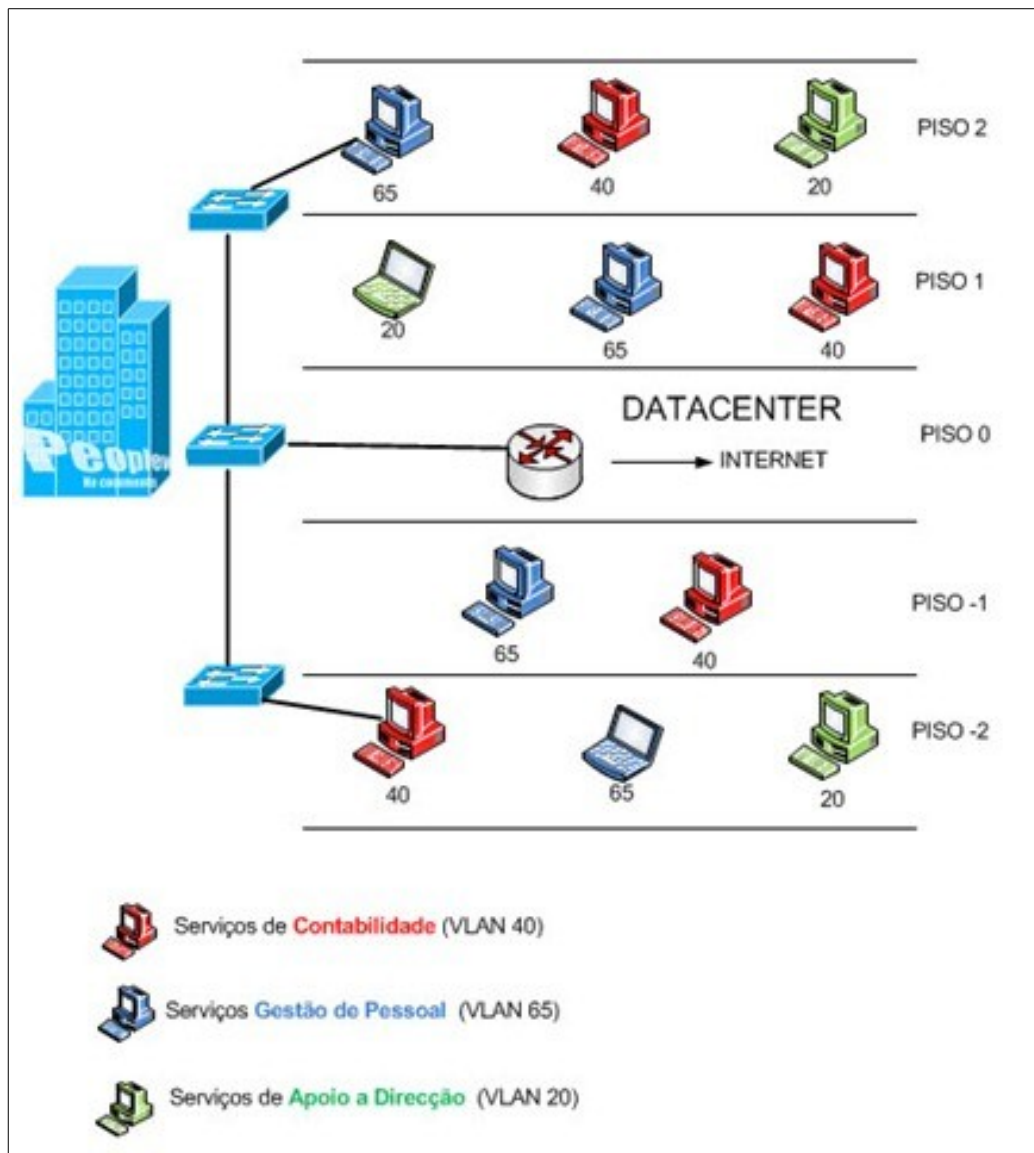


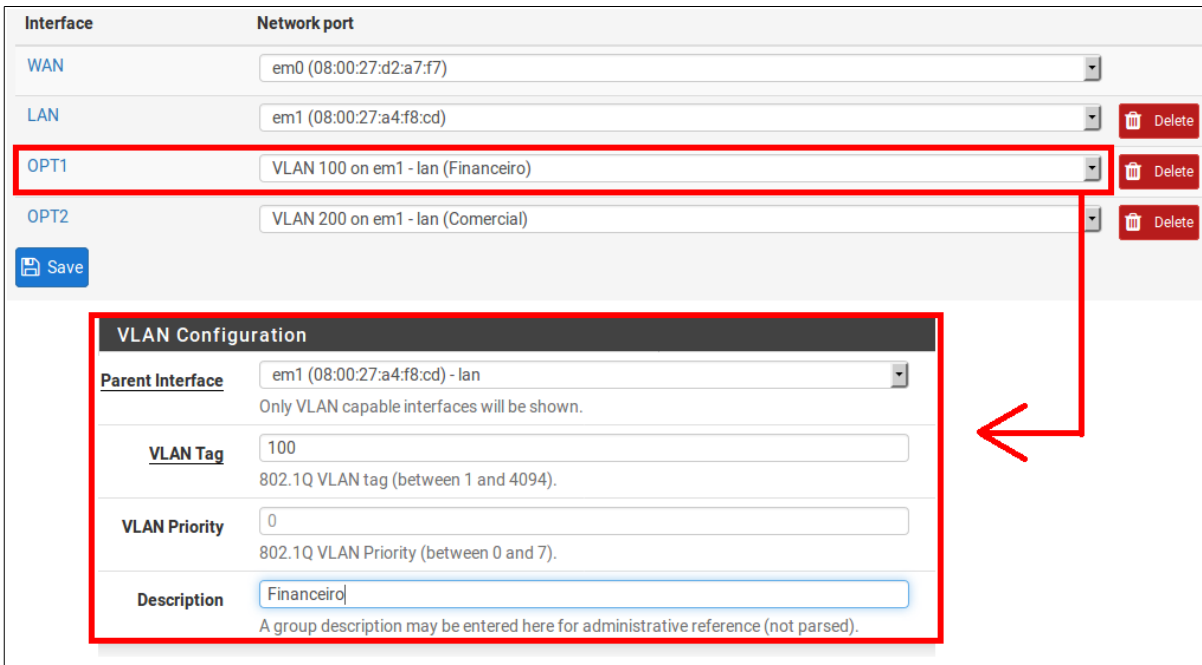
Figura 15 - Exemplo de Rede Subdividida por VLANs

Fonte: https://pplware.sapo.pt/wp-content/uploads/2010/12/vlans_00.jpg

O *PFSense* contém o recurso de VLAN, assim ele pode distribuir dados em uma interface física, mas fragmentado em duas ou mais interfaces virtuais, assim, não fica limitado as

interfaces de rede, onde é atribuído a configuração individualmente, onde com um recuso de segurança básico, pode ser incrementado quando optando a utilizar a ferramenta como parte da infraestrutura, transformando uma interface em várias outras.

Para criar uma VLAN no *pfSense*, basta acessar **Interfaces | Assign | VLANs**, criando uma tag (número), qual interface física ira utilizar, sua prioridade perante as outras, e uma descrição, assim, será criando uma simbologia para a interface física, a qual acessar **Interfaces | Assign | Interface assignments**, poderá ativar a mesma, e criar regras conforme a necessidade, tratando a mesma como uma estrutura adicional.



The screenshot displays the pfSense interface for configuring VLANs. The main table lists interfaces: WAN (em0), LAN (em1), OPT1 (VLAN 100 on em1 - lan (Financeiro)), and OPT2 (VLAN 200 on em1 - lan (Comercial)). A red box highlights the OPT1 row. Below the table, a 'VLAN Configuration' dialog box is open, also highlighted with a red box. A red arrow points from the OPT1 row to the dialog box. The dialog box contains the following fields:

VLAN Configuration	
Parent interface	em1 (08:00:27:a4:f8:cd) - lan <small>Only VLAN capable interfaces will be shown.</small>
VLAN Tag	100 <small>802.1Q VLAN tag (between 1 and 4094).</small>
VLAN Priority	0 <small>802.1Q VLAN Priority (between 0 and 7).</small>
Description	Financeiro <small>A group description may be entered here for administrative reference (not parsed).</small>

Figura 16 - Interface VLAN PFSense

7.3. FAILOVER E LOADBALANCER

É essencial uma infraestrutura estar sempre disponível, e não dependente de manutenções manuais, assim sempre mantêm o equipamento ativo, utilizando redundâncias de fontes de alimentação e *nobreaks*, mas quando esta dependendo de terceiros, existem coisas que não temos controle, como a disponibilidade de links de internet, não é recomendado a utilização de apenas uma interface para a WAN, mas sempre um reserva, e utilizando em conjunto um *failover*.

O *failover* é a capacidade de migrar um serviço para outro automaticamente, ou seja, se um link de internet cair, automaticamente é redirecionado para outro, sem nenhuma intervenção, assim mantendo o serviço com base em um recurso redundante, e a função sempre ativa.

O *failover* contém um problema, onde existe um link parado, e usado apenas quando existir uma falha, assim desperdiçando o recurso, para ter uma melhor utilização do mesmo, e mantendo o serviço contra falhas, podemos utilizar o *loadbalancer* (balanceamento de carga), resolvendo não só esse problema, mas também o de sobrecarga de hardware, onde cada pacote de rede, pode passar pelo link 1 ou pelo link 2, distribuindo o serviço, e em caso de falhas, irá apenas utilizar o link ativo, sem interferir na usabilidade.

Ambos os recursos são oferecidos pelo *pfSense*, sendo o *loadbalancer* para distribuição de conexão, e o *failover* voltado para servidores redundantes, onde a ferramenta monitora os serviços para saber de sua atividade e cria *pools* de mudança, e seu funcionamento é feito de um túnel virtual, para que se mantenha a estrutura padrão.

7.4. ÁREAS DMZ

Um recurso muito importante em uma infraestrutura voltada à segurança, são as áreas de DMZ, em outras palavras uma zona desmilitarizada, onde traz uma ideia de isolar servidores da rede local, principalmente os que oferecem um acesso externo, onde acrescenta uma camada de segurança, onde se ocorrer um ataque, diminui o prejuízo causado na área, mas sempre controlados por ALGs (Application Layer Gateway) filtrando todas as entradas e saídas. Segundo Mâcedo (2012) uma DMZ pode ser implementada com filtros de rede configurados nas suas bordas, estes filtros são responsáveis por realizar o controle de acesso do que entra e do que sai da DMZ e podem ser do tipo packet filtering, stateful packet filtering e de cache como servidores de proxy.

Além de conter um possível ataque concentrado em uma área, pode-se criar uma área DMZ de análise, onde qualquer pacote não concedido para passagem a uma rede, em vez de ser rejeitado pode ser movido a essa zona de testes onde será inspecionado, assim servindo como ponto de referência para criação de novas regras, e observar comportamentos de possíveis ameaças em descoberta, assim mantendo um banco de dados de ameaças sempre atualizado, e conhecendo o potencial de aplicação mal-intencionadas. A criação de uma zona de DMZ pode ser feita através de qualquer ferramenta de firewall com suporte a NAT.

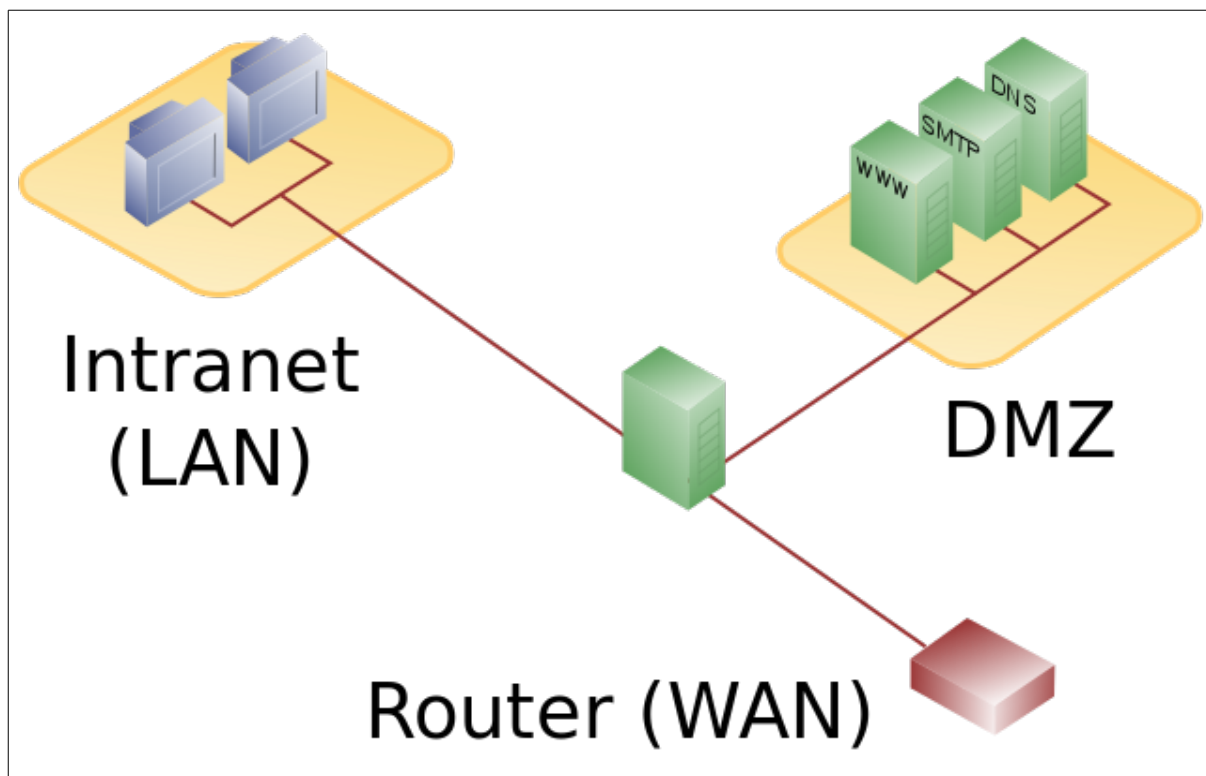


Figura 17 - Zona de DMZ

Fonte: https://www.gta.ufrj.br/grad/15_1/firewall/media/dmz2.png

7.5. PFSENSE COM AUXILIO DE SERVIDORES SECUNDÁRIOS

Apesar de o *pfsense* oferecer inúmeros recursos, é interessante não o deixar responsável por muitos serviços, por uma questão natural de desempenho, mas não o descartando como ferramenta de filtragem mas, utilizando sua convergência para outros servidores, assim o deixando mais livre em recursos para ser o principal filtro, além de poder criar uma escalabilidade para a segurança, não tendo uma centralização de recursos.

Um dos principais recursos a ser utilizado em um servidor secundário, seria para o envio de *log*, onde a ferramenta oferece suporte para até três servidores externos, podendo escolher quais serviços serão enviados a eles, assim diminuindo a taxa de uso de disco, além de poder manter as informações em uma área diferente, assim uma hipótese não perder os registros de ocorrências, e dados importantes.

Outro recurso interessante a se manter como secundário, seria um monitor de rede e desempenho, podendo assim ter um parâmetro comparativo para as ocorrências da rede, e mais um ponto de análise para resultados, como por exemplo o uso de um servidor **zabbix**, onde oferece recursos de notificação, e um monitoramento convencional. Apesar de o *pfsense* oferecer tais recursos em seu *dashboard*, é importante nunca se limitar em uma fonte de informação para relatar situações em uma infraestrutura, principalmente se estiver sob suspeita de um ataque e em ocorrência de informações alterada.

8. CONCLUSÃO

Com as etapas do trabalho executadas, desde o estudo da ferramenta, tentando a compreender e como a configurar da melhor forma, principalmente tendo o conhecimento de como funciona uma invasão, criando regras para os possíveis ataques, foi possível montar um ambiente de testes para verificar toda a capacidade do *pfSense*, e seu potencial de crescimento, vendo a disponibilidade que é entregue por ele do ambiente, e os recursos de segurança oferecidos, principalmente por registrar ocorrências, facilitando uma melhor análise de melhorias, e a expansão oferecida.

O *pfSense* trata-se de uma ferramenta de borda, onde gere todo o tráfego entre a rede local, externa, e demais interfaces quando existem, assim, filtrando e gerenciando o destino de cada informação, principalmente quando é necessário realizar bloqueios evitando itens maliciosos, tornando-se a primeira camada de uma infraestrutura visada a segurança. A ferramenta é de certa forma robusta, trazendo inúmeros recursos adicionais, assim, utilizando ele mesmo pode acrescentar recursos adicionais de confiabilidade de acesso. A aplicação apresenta um custo extremamente baixo para implantação, dependendo apenas de um hardware comum para instalação, e como passa por constantes atualizações sempre está apresentando melhorias, principalmente por ser executado em cima da estrutura do sistema operacional FreeBSD, além de buscar encaixar em várias camadas de segurança, onde quanto mais camadas existem, mais difícil é um invasor chegar a seu destino final e causar danos.

Seu gerenciamento tenta atingir desde a porta de entrada com o uso de *firewall* e redirecionamento NAT, passando pela rede em seu monitoramento, até atingir o reconhecimento dos dispositivos e os usuários com filtros de acesso. De acordo com os testes realizados, onde há uma etapa de um *pentest* tentando encontrar falhas, ele se saiu muito bem, onde conseguiu identificar as ações realizadas, e bloqueios, dando credibilidade às regras criadas, além de ser flexível a melhorias, mostrando que é uma ferramenta pronta para ser implantada em infraestruturas, pelo seu potencial. É claro que apenas os procedimentos realizados não prova total segurança, onde mais testes são necessários ser realizados, mas a medida que mais força bruta é posta no cenário, mais regras também são criadas, tentando criar um balanceamento entre ataque e defesa.

8.1. TRABALHOS FUTUROS

Mesmo o *pfsense* sendo uma ferramenta robusta, e feito alguns testes sob o mesmo, ainda existe mais coisas a serem exploradas, para validar recursos diferentes, ver seu comportamento diante a outras aplicações, e desenvolvendo ambientes mais robustos, assim existindo um potencial desenvolvimento futuro como continuação deste trabalho.

Como o mesmo é executado em um sistema operacional, pode-se existir trabalhos desenvolvendo aplicações novas para o mesmo, assim existindo programas que complemente a aplicação, a deixando mais poderosa, e alcançando áreas ainda não suportadas pela mesma.

9. REFERENCIAS

ALECIO, Willian Santos; PEREIRA, Júlio Cesar. IMPLANTAÇÃO DE FIREWALL: Segurança em Redes de Computadores. Universidade Paranaense (Unipar), Paranavaí – PR, 2014. 6p.

ALENCAR, Felipe. O que é DNS. Disponível em <<http://www.techtudo.com.br/noticias/noticia/2014/07/o-que-e-dns.html>> Acessado em: 21 de dezembro de 2016.

BARWINSKI, Luísa. O que é proxy?. Disponível em <<https://www.tecmundo.com.br/navegador/972-o-que-e-proxy-.htm>> Acessado em: 21 e dezembro de 2016.

BATISTA, Julio. TCP/IP, NAT. Disponível em <http://juliobattisti.com.br/artigos/windows/tcpip_p20.asp> Acessado em: 19 de dezembro de 2016.

BENINI, Renata Aparecida; DAIBERT Marcelo Santos. Monitoramento de Redes de Computadores – Artigo Revista Infra Magazine 1, abril de 2011.

CAETANO, Aparecida Caetano; SOUZA, Marta Alves; COSTA, Helder Rodrigues. SEGURANÇA DA INFORMAÇÃO: Um Estudo a Partir dos Crimes Virtuais.

CUNHA, Juan Carlos. Http e https – segurança com https – diferença entra http e https. Disponível em <<https://juancarloscunha.wordpress.com/2009/06/02/http-e-https-seguranca-com-https-diferenca-entre-http-e-https/>> Acessando em 20 de julho de 2017.

DOCUMENTAÇÃO PFSENSE. Disponível em <doc.pfsense.org> Último acesso em: 20 de julho de 2017.

HAUSER Van. Documentation THC-Hydra. Disponível em <thc.org/thc-hydra> Acessado em 17 de julho de 2017.

KENNEDY; O’GORMAN; KEARNS; AHARONI. METASPLOIT. The Penetration Tester’s Guide. São Francisco: No Starch Press, 2011.

LEITE, Danilo. Proxy transparente x não transparente: Desvendando os mitos. Disponível em <<http://www.it9.com.br/proxy-transparente-x-nao-transparente-desvendando-os-mitos/>> Acessado em: 21 e dezembro de 2016.

MÂCEDO, Diego. Conceito de DMZ. Disponível em <<http://www.diegomacedo.com.br/conceito-de-dmz/>> acessado em 24 de julho de 2016.

MARIMOTO, Carlos. Gateway. Disponível em <<http://www.hardware.com.br/termos/gateway>> Acessado em: 20 de dezembro de 2016.

MARTINS ELAINE, O que é VPN?. Disponível em <<https://www.tecmundo.com.br/1427-o-que-e-vpn-.htm>> Acessado em 19 de julho de 2017.

NOVAZ, Rafael. DNS: Entenda como se proteger com o analista de segurança da Psafe. Disponível em <<http://www.psafe.com/blog/dns-alterado-rouba-dados-usuarios/>> Acessado em: 21 e dezembro de 2016.

OWASAP Foundation. As 10 vulnerabilidades de segurança mais críticas em aplicações WEB. Disponível em <https://www.owasp.org/images/4/42/OWASP_TOP_10_2007_PT-BR.pdf> Acessado em 21 de dezembro de 2016.

PARMAR, Primesh; FELEON, Alex. Aplicação de Pentest em Sistemas Computacionais para Análise de Vulnerabilidades: Um Estudo de Caso. Fundação Centro de Análise, Pesquisa e Inovação Tecnológica (FUCAPI), Manaus – AM, 2013. 10 p.

PEREIRA, Ana Paula. O que é DHCP?. Disponível em <<https://www.tecmundo.com.br/2079-o-que-e-dhcp-htm>> Acessado em: 22 e dezembro de 2016.

SILVA, Raquel Fonseca; PEREIRA, Julio Cesar. IDENTIFICANDO VULNERABILIDADES DE SEGURANÇA COMPUTACIONAL. Universidade Paranaense (UNIPAR), Paranavai – PR, 2013. 5 p.

TANENBAUM, Andrew Stuart. Redes de Computadores, Tradução de Vandenberg de Souza 4ª Edição, Editora Campos, 2011.

TORRES, Gabriel. Redes de Computadores, Rio de Janeiro, Axcel Books do Brasil, 2001.

VARREIRA, Jéssica. Como VLANs podem ajudar na segurança. Disponível em <<http://micreiros.com/como-as-vlans-podem-ajudar-na-seguranca/>> Acessado em 23 de julho de 2017.

VIEGAS, Julio. O QUE É (E PARA QUE SERVE) O PENTEST. Disponível em <<http://blog.onedaytesting.com.br/o-que-e-pentest-e-para-que-serve>> Acessado em: 13 de dezembro de 2016.

VIVA O LINUX. Artigo: Nikto – tutorial basico e e avançado. Disponível em <<https://www.vivaolinux.com.br/artigo/Nikto-Tutorial-basico-e-avancado?pagina=1>> Acessado em 20 de julho de 2017.

WILLIAMSON, Matt. Pfsense 2 Cookbook, Editora Packt Publishing, 2011