URIAS ESTEVES CHAVES

ESTUDO DA VIABILIDADE DE IMPLEMENTAÇÃO DE UM SERVIDOR LINUX DE AUTENTICAÇÃO COMO FERRAMENTA DE APOIO NO GERENCIAMENTO DA LAN – REDE LOCAL DA PREFEITURA MUNICIPAL DE NOVO CRUZEIRO.

TEÓFILO OTONI – MG
FACULDADES UNIFICADAS DE TEOFILO OTONI
2016

URIAS ESTEVES CHAVES

ESTUDO DA VIABILIDADE DE IMPLEMENTAÇÃO DE UM SERVIDOR LINUX DE AUTENTICAÇÃO COMO FERRAMENTA DE APOIO NO GERENCIAMENTO DA LAN – REDE LOCAL DA PREFEITURA MUNICIPAL DE NOVO CRUZEIRO.

Monografia apresentada ao curso de Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni, como requisito parcial à obtenção do título de bacharel em Sistemas de Informação.
Área de concentração: Redes de Computadores.
Orientador: Msc. Salim Ziad Pereira Aouar.

TEÓFILO OTONI – MG
FACULDADES UNIFICADAS DE TEOFILO OTONI
2016

rede de ensino
DOCTUM

FACULDADES UNIFICADAS DE TEÓFILO OTONI
NÚCLEO DE TCC / SISTEMAS DE INFORMAÇÃO
Autorizado pela Portaria 4.012 de 06/123/2004 – MEC

FOLHA DE APROVAÇÃO

A monografia intitulada: *Estudo da viabilidade de implementação de um servidor Linux de autenticação como ferramenta de apoio no gerenciamento da LAN – Rede Local da Prefeitura Municipal de Novo Cruzeiro – MG,*

elaborada pelo aluno Urias Esteves Chaves,

foi aprovada por todos os membros da Banca Examinadora e aceita pelo curso de Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni, como requisito parcial da obtenção do título de

BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Teófilo Otoni, 21 de novembro de 2016

Professor Orientador: Salim Ziad Pereira Aouar

Professor Examinador: Amaury Gonçalves Costa

Professor Examinador: Luiz Fernando Alves

Dedico esse trabalho aos meus pais
Silvio Alves Chaves (In Memoriam) e Maria das Graças Esteves Chaves, amados pais!

AGRADECIMENTOS

Primeiramente a Deus, por me conceder saúde física e mental para que eu pudesse concluir o curso, me confortando nos momentos de dúvida. A minha mãe, que em nenhum momento me desamparou, esteve sempre presente nos momentos de dificuldades vividos ao longo do curso.

Ao meu amigo Walter Augusto, que contribuiu de forma significativa disponibilizando um computador desktop para que eu pudesse desenvolver esse projeto.

Ao meu orientador e professor Msc. Salim Ziad Pereira Aouar, pela dedicação, sempre me orientando qual melhor caminho seguir para que eu pudesse obter êxito na conclusão desse projeto.

A minha namorada Uilcleides Braga da Silva, pelo apoio e incentivo.

Também aos demais mestres Arnon Roberto Rihs, Ahmine Handeri de Oliveira Lima Froede, Wilbert Viana, Fabiano Souza Santos, Jonilson Batista Campos, Mauro Gomes da Fonseca, Oseas Teixeira, Amaury Gonçalves Costa, Iara Pereira da Encarnação Alcântara, Yvssa Carneiro Desmonts, Sérgio Soares Macedo, Luciano Campos Lavall, que tive a oportunidade de conviver em sala durante o curso.

Tudo aquilo que sou, ou pretendo ser, devo a um anjo, minha mãe.

Abraham Lincoln

LISTAS DE FIGURAS

LISTA DE SIGLAS

ACL
Access Control List, 18

DHCP
Protocolo de Configuração Dinâmica de Host, 26

DNS
Domain Name System, 26

FTP
Protocolo de Transferência de Arquivos, 14

HTTP
Hypertext Transfer Protocol, 19

HTTPS
Hyper Text Transfer Protocol Secure, 20

IP
Internet Protocol, 15

IPsec
IP Security, 21

KVM
Kernel-based Virtual Machine, 24

OWNCLOUD
Servidor de Arquivos em Nuvem, 26

SAMBA
Servidor de Arquivos, 26

SMB
Server Message Block, 18

SMB/CIFS
Server Message Block/ Common Internet File System, 59

SSL
Secure Sockets Layer, 19

TCP
Transmission Control Protocol. *Consulte*

TLS
Transport Layer Security, 19

VPNs
Virtual Private Networks, 22

WEP
Wired Equivalent Privacy, 22

RESUMO

Monografia desenvolvida tendo como título “Estudo da Viabilidade de Implementação de um Servidor Linux de Autenticação como Ferramenta de Apoio no Gerenciamento da Lan – Rede Local da Prefeitura Municipal de Novo Cruzeiro”, usando software livre para redução de custos ao órgão. Moldando sobre a plataforma de virtualização KVM e sobre as ferramentas de redes usadas em máquinas virtuais separadas com DNS, DHCP, SAMBA, SQUID, IPTABLES, OWNCLOUD. Gerando um domínio com autenticação e controle de permissões para os usuários conectados na rede.

Palavras-Chave: Autenticação; Samba4-pdc; Gnu/Linux; Owncloud; Squid.

SUMÁRIO

INTRODUÇÃO

No cenário atual há um grande número de procrastinação no cumprimento do serviço prestado pelos órgãos públicos, a demora na prestação de serviços pode estar relacionada à diversos fatores, como formalismo exagerado com procedimentos demasiadamente burocráticos, com a inadequada utilização das ferramentas de tecnologia da informação.

O uso inadequado dos equipamentos de informática pode prejudicar o desempenho do serviço, como exemplo, tem-se o uso da internet onde um número x de servidores a usa para acessar redes sociais, realizar downloads dos mais diversos tipos de arquivos dos milhares de sites existentes, podendo causar uma lentidão na rede, uma vez que o link de internet disponível é na sua maioria consumida por essas ações.

Por outro lado, computadores muitas vezes são tomados como bens próprios, servindo para armazenar arquivos pessoais, instalando programas supérfluos que ferem os interesses do órgão ao qual exercem suas atividades profissionais.

Com o passar do tempo podem surgir problemas para realizar tarefas triviais como instalar um programa necessário para a realização de um serviço, surgir um vírus que afete todos os computadores da rede podendo danificar ou corromper arquivos, etc.

É possível que não exista uma fiscalização ou um controle dos equipamentos de informática para que sejam usados de forma produtiva no cumprimento do serviço público prestado na *Prefeitura Municipal de Novo Cruzeiro.* Como cada gestor assume por um tempo pré-determinado pela lei não atentam ao comportamento dos seus subordinados, pelas características institucionais é possível que ocorra a contratação de funcionários com algum tipo de relação pessoal com a administração pública, uma vez que a contratação, com exceção dos servidores concursados, ocorre de maneira subjetiva e ocupam cargos de chefia que desempenham funções ou atividades como chefes na hierarquia administrativa.

É possível que estes funcionários não estejam preocupados com a eficiência máxima ao cumprir com as atribuições que o cargo impõe e isso na sua essência pode gerar uma lentidão no cumprimento do serviço público prestado. Para quem observa a gestão pública de

fora, entende que os que a compõem não querem cumprir com suas obrigações, no entanto o seu funcionamento é como um relógio, se uma peça não fizer a sua tarefa todas as outras ficarão prejudicas ou afetadas.

As pessoas e máquinas fazem parte de um sistema que produz informações e produtos na qualidade de serviços, ambos desempenham atividades que são cruciais para os objetivos da administração pública. O manuseio adequado dos equipamentos de informática pode proporcionar maior eficiência na realização das tarefas.

Existem mecanismos que podem contribuir para melhorar o desenvolvimento das atividades e comunicação em uma organização, dentre eles um preciso controle e acompanhamento das ferramentas tecnológicas usadas.

Com um controle preciso na rede local de internet é possível resolver quase que em 100% (cem por cento) essas e outras falhas na realização dos trabalhos. No uso da internet é possível bloquear o acesso a qualquer site, evitando que o funcionário a use de forma imprópria; no uso do computador é possível limitar o seu manuseio indevido para que o mesmo não se torne uma propriedade de quem o usar. No que diz respeito aos dados que trafegam na rede, também existem mecanismos de segurança para protegê-los, evitando que sejam alterados, apagados ou modificados sem que o seu destinatário tenha tido acesso a eles.

Segundo a Fundação GNU "o GNU/Linux é um sistema operacional que é um software livre". Por ter em sua essência o seu uso livre não gera quaisquer custos a seus usuários quanto a licença e/ou instalação de novas versões etc. No GNU/Linux raramente necessita de reinicialização, a mesma poderá ocorrer se por ventura surgir uma falha em um dos hardwares (parte física) do computador no qual foi instalado ou na instalação de um ou mais periféricos. No modo texto, não exige muitos recursos do computador para funcionar de maneira eficiente. Outra vantagem no seu uso é que *não é vulnerável a vírus*, consegue acessar e/ou reconhecer discos (HD) formatados pelo *DOS, Windows, Novell, OS/2, NTFS, SunOS, Atari, Mac dentre outros.* Na parte de segurança é possível criar o seu *firewall* com regras específicas para o uso da rede, com roteamento estático e dinâmico de pacotes, *proxy* tradicional e transparente, ponte entre redes, *proxy arp.* Na parte de controle interno dos usuários é possível através do *SAMBA* – "software servidor" auxiliar no compartilhamento e gerenciamento de arquivos, impressoras e acesso a determinadas pastas do servidor para obter um total controle sobre cada computador que irá compor a rede. Também existe uma

quantidade enorme de softwares para monitoramento como, por exemplo, o *zabbix*, que foi desenvolvido para monitorar a disponibilidade dos serviços e ativos da rede.

Diante dos desafios apresentados, o presente estudo apresenta como problema a seguinte questão: Como o uso do *Servidor Linux* de autenticação poderia melhorar o desempenho dos serviços realizados na *Prefeitura Municipal de Novo Cruzeiro?* Com o intuito de responder tal problema de pesquisa foram estabelecidas as seguintes hipóteses, a saber:

H0 – Não será possível implementar o *Servidor Linux* com suas respectivas funcionalidades no ambiente de trabalho, em consequência das constantes mudanças dos gestores responsáveis pela órgão público.

H1 – O controle do uso da *internet* feito pelo *proxy* e pelo *squid* poderia melhorar a segurança da rede, gerenciaria o acesso aos *sites* e/ou pesquisas feitas nos diversos buscadores a exemplo o *google*, também melhoraria a navegação, uma vez que armazenaria um cópia das páginas *WEB* acessadas com maior frequência em (*cache*), sem precisar carregar todo o conteúdo do site novamente apenas atualizando o que for necessário.

H2 – O gerenciamento e compartilhamento dos arquivos administrados pelo *samba* permitiria aos computadores usando sistemas operacionais *Linux e/ou Windows* disfrutarem de serviços como: controle de acesso em nível de usuário, domínios, compartilhamento de arquivos/impressoras/diretórios etc.

H3 – O *firewall* configurado pelo *Iptables* atuaria no controle dos pacotes de origem e destino verificando através das regras predefinidas se os mesmos têm ou não permissões para transitar na rede.

H4 – O *backup feito pelo Owncloud* auxiliaria na recuperação/arquivamento dos dados, seja por necessidade para uma consulta futura, falha no computador do usuário ou se o mesmo excluir arquivos por engano que venha precisar desses dados no futuro. Verificar se os serviços instalados no *Servidor Linux* irão contribuir de forma produtiva para o ambiente organizacional da *Prefeitura Municipal de Novo Cruzeiro.* Construir um *Servidor Linux* de teste com serviços *backup, dhcp, dns, proxy, iptables, samba, Owncloud* usando o *KVM* e verificar em um ambiente de teste sua eficácia no controle de uma rede fictícia. Detectar caso haja, possíveis falhas na configuração dos serviços supracitados. Avaliar os resultados dos testes e posteriormente analisar a viabilidade da implementação do servidor no ambiente da

Prefeitura Municipal de Novo Cruzeiro. Elaborar a estrutura organizacional da *Prefeitura Municipal de Novo Cruzeiro* para criar níveis de acesso à rede, senhas e usuários. Os gestores das empresas públicas assim como os de empresas privadas tomam decisões e têm necessidades informacionais. O presente trabalho apresenta um ganho cientifico por tratar de aspectos que visam o aperfeiçoamento da comunicação de uma entidade sem fins lucrativos e associados a uma revisão de literatura, buscando relacionar conceitos trazidos na comunidade acadêmica e a realidade prática de uma entidade.

No cenário atual, a proteção ou recuperação de dados/informação são essenciais para usuários considerados leigos na área de informática. Cabe ao administrador de redes proporcionar meios para que isso aconteça, a exemplo de dados deletados por acidente, uma falha em um HD local, etc.Com as configurações e serviços instalados corretamente no servidor é possível proporcionar tranquilidade aos usuários caso ocorra esse e demais problemas.

A conscientização do uso da internet de forma restrita aos interesses do seu local de trabalho pode evitar que o usuário acesse sites inapropriados aos interesses e ambiente de trabalho, uma vez que não importa o nível de conhecimento que um usuário tenha, se o mesmo estiver de posse de um computador que tenha acesso a internet, tende a usar de forma que bem o convier, ou seja, para seus interesses próprios, visto que o ser humano de forma natural tende a maximizar seus interesses particulares.

Limitação do acesso aos dados que transitarão no *servidor* com o uso de hierarquias aos usuários, uma vez que, pessoas de diferentes repartições executando serviços diversificados não precisam ter contato com dados de setores alheios aos seus.

O desenvolvimento de competências necessárias para trabalhar com administração de servidores *Linux* com estudos que levarão ao domínio de serviços como *dhcp, dns, proxy, Iptables, samba, backup usando o Owncloud* dentre outros, conteúdos esses que são essenciais para um administrador de redes, irá contribuir para a formação acadêmica e profissional pessoal, a qual está sendo desenvolvida o objeto de estudo em questão.

O acesso e domínio de uma tecnologia que não se aprende em sala de aula como o uso do *KVM* que segundo o *site <https://wiki.debian.org/KVM>* "é uma solução de virtualização completa para Linux em x86" incluindo 64 bits. Na qual será instalado os serviços separados em micros servidores e interligados por endereçamentos *IP* que trabalharão de forma homogênea formando um servidor central, sendo realizada uma simulação e testes de viabilidade por meio de um ambiente virtual.

1 REVISÃO LITERÁRIA

1.1 SOFTWARE LIVRE

1.1.1 Conceitos

Para ser considerado um software livre é necessário que o programa forneça aos usuários quatro liberdades essenciais (Fundação Software Livre).

Conforme afirma a Fundação Software Livre:

> A liberdade de executar o programa como você desejar, para qualquer proposito (liberdade 0).
> A liberdade de estudar como o programa funciona, e adaptá-lo às suas necessidades (liberdade 1). Para tanto, acesso ao código-fonte é um pré-requisito.
> A liberdade de redistribuir copias de modo que você possa ajudar ao próximo (liberdade 2).
> A liberdade de distribuir copias de suas versões modificadas a outros (liberdade 3). Para tanto, acesso ao código-fonte é um pré-requisito. (Disponível em <http://www.gnu.org/>).

Seguindo essas definições das quatro liberdades, pode-se entender que o termo "software livre" está vinculado ao modo de uso, seja para estuda-lo, melhora-lo, distribui-lo sem que isso gere quaisquer custos ao usuário, uma vez que as quatro liberdades supracitadas amparam a quem fizer uso do mesmo.

1.1.2 Fundação GNU

Fundada em 1985 por *Richard Matthew Stallman*, com sede em Boston, MA, EUA, a Fundação do Software Livre (GNU) têm como pilares manter, criar, promover e firmar o direito de uso do software livre por usuários de computadores.

A fundação parte do princípio de que todo usuário tem o direito de controlar como o seu software se comporta, conhecendo o seu kernel (núcleo) para promover o seu uso e desenvolvimento e/ou aperfeiçoamento atendendo as necessidades de cada um, desde que essas modificações juntamente com código fonte seja disponibilizado para que todos possam ter acesso.

1.1.3 Licença

Na versão atual versão 3 (três), a GNU Public License (GPLv3) foi redigida sobre as quatro liberdades supracitadas, para que os usuários e, ou desenvolvedores de software livre usufruírem de maneira que achar correta do software podendo alterar, copiar, etc sem serem julgados criminalmente por essas ações. A versão atual da licença protege os usuários de três ameaças:

Nesse assunto, afirma Brett Smith:

> Tivoização: Algumas empresas têm criado vários tipos diferentes de dispositivos que executam o software GPL, e depois manipulados o hardware para que possam mudar o software que está sendo executado, mas você não pode. Se um dispositivo pode executar software arbitrário, é um computador de uso geral, e seu proprietário deve controlar o que ele faz. Quando um dispositivo impede você de fazer isso, nós chamamos isso de tivoização.
>
> As leis que proíbem o software livre: Legislação como o Digital Millennium Copyright Act e da Directiva Direito de Autor da União Europeia torna-lo um crime para escrever ou software de compartilhamento que pode quebrar DRm (Gestão Digital de Restrições). Estas leis não devem interferir com os direitos do GPL concede-lhe.
>
> Acordos de patentes discriminatórios: Microsoft recentemente começou a dizer às pessoas que não vai processar usuários de software livre por violação de patente, como desde que você obter o software de um fornecedor que está pagando Microsoft para o privilégio. Em última análise, a Microsoft está a tentar recolher royalties para o uso de software livre, o que interfere com a liberdade dos usuários. Nenhuma empresa deve ser capaz de fazer isso. (Disponível em <http://www.gnu.org/licenses/quick-guide-gplv3.html>).

A GPL[1] usa o Copyleft que "é um método legal de tornar um programa em software livre e exigir que todas as versões modificadas e estendidas do programa também sejam

[1] http://www.gnu.org/licenses/licenses.pt-br.html

software livre". Método esse que é um acrônimo do Copyright que se baseia na posse do direito intelectual unicamente do seu criador, não permitindo a disseminação seja por qual meio for.

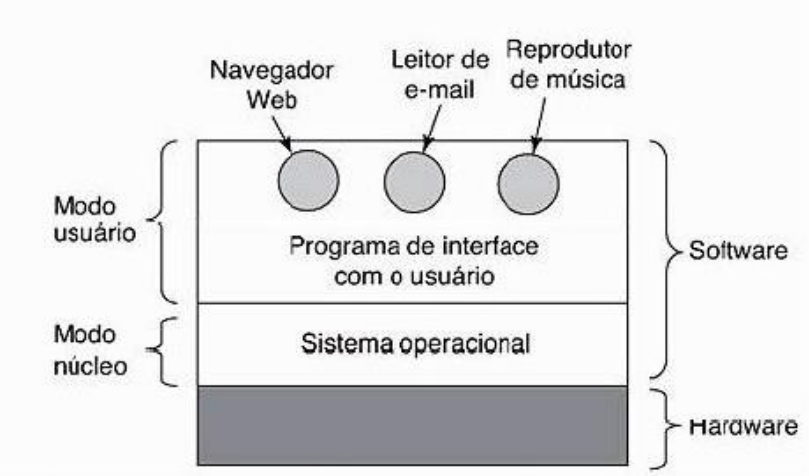
1.4 SISTEMA OPERACIONAL

1.4.1 Conceitos

Conforme Tanenbaum (2009), um sistema operacional pode ser definido como um conjunto de programas para executar as tarefas solicitadas pelos usuários, ou serviços que irão realizar no decorrer do seu funcionamento. Em contato com o usuário comum, permite por exemplo, que o mesmo consiga realizar o seu manuseio realizando tarefas corriqueiras como ouvir uma música, ler um texto, salvar e/ou gravar um arquivo em uma mídia.

Para Tanenbaum (2009, p.2) "os computadores têm um dispositivo de software denominado sistema operacional, cujo trabalho é fornecer aos programas do usuário um modelo de computador melhor, mais simples e mais limpo."

Figura 1: Encaixe do Sistema Operacional



Fonte: TANENBAUM, 2009, p.2.

De acordo com a Figura 1, é possível compreender melhor o que Tanenbaum (2009) afirma, analisando a mesma é possível afirmar o que foi mencionado no início, ou seja, para que o usuário acesse a internet será necessário um software como por exemplo o *navegador web* que estará sobre um interface gráfica denominada modo usuário, que está sobre o sistema operacional denominado modo núcleo ou kernel que por sua vez está diretamente sobre o hardware do computador controlando os processos de entradas e saídas.

O modo usuário ou interface gráfica no GNU/Linux é gerenciado pelo software *XFree (Servidor de Janelas)*, é ele o responsável por controlar a aparência, a forma, a posição e as funções desempenhadas pelos ícones que aparecem para o usuário. Segundo Fábio Berbert de Paula "já existem cerca de 20 gerenciadores de janelas para o Linux. (Disponível em <https://www.vivaolinux.com.br/artigo/Interfaces-Graficas-no-Linux>). Dentre os mais populares pode-se citar: *Kde, Gnome, AfterStep, Enlightnment, WindowMaker, IceW, BlackBox, Fvwm90",* e o modo núcleo é popularmente chamado de Kernel, onde está escrito todas as diretivas de funcionamento do sistema operacional.

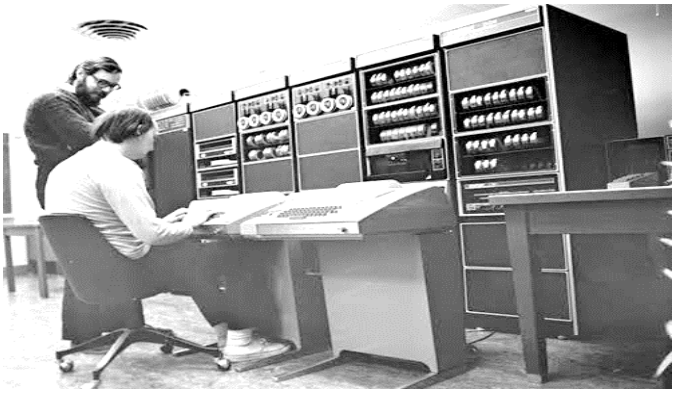
1.4.2 O Sistema GNU/Linux

Segundo o site[2] o Linux surgiu a partir de um sistema chamado Unix multitarefa e multiusuário que funciona em muitos computadores. Ele funciona de duas formas, a primeira é focalizada no kernel que tem a incumbência de interligar-se com o hardware e a outra é o funcionamento de tudo que irá ser executado no sistema depende do kernel para isso.

Para exemplificar a história do GNU/Linux é preciso recorrer a datas longínquas a começar pelo ano de 1965 onde a união de três interesses mútuos da Bell Telephone Labs da AT&T, General Electric com o projeto MAC do MIT (Massachusetts Institute of Technology) criam um sistema operacional de nome Multics. Em 1969 Ken Thompson e Dennis Richie (criador da linguagem C) criam uma versão do sistema até então chamado de Unics que ficou sendo chamado de Unix, e em 1971 esse sistema é codificado para um computador PDP-11 exibido na Figura 2 e em 1973 o Unix é reescrito em linguagem C por Dennis Ritchie.

[2] https://www.vivaolinux.com.br/artigo/Historia-do-GNU-Linux-1965-assim-tudo-comecou/

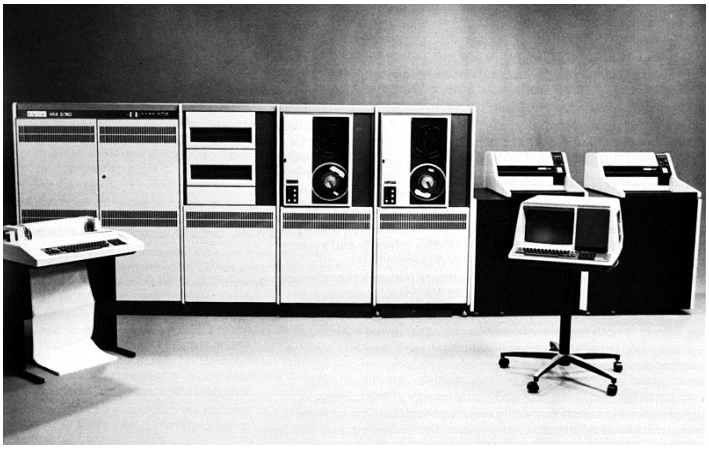
Figura 2: Computador PDP 11



Fonte: http://fatecads2011.blogspot.com.br/

Nos anos seguintes surgiram versões do Unix criadas para atender interesses comerciais da AT&T que criou o Unix System III e a versão da Universidade de Berkeley na Califórnia que criou a BSD (Berkeley Systems Division) para computadores VAX, como mostra a Figura 3.

Figura 3: Computador VAX



Fonte: http://computadorso.blogspot.com.br/2009/03/vax.html

No ano de 1984 um cientista do MIT (Massachusetts Institute of Technology) de nome Richard Stallman cria um projeto de nome GNU com intenção de recriar um sistema operacional com as mesmas funcionalidades do Unix de modo que funcionasse gratuitamente. No ano seguinte ele funda a Free Software Foundation (Fundação Software Livre) para apoiar no desenvolvimento e disseminação do projeto seguindo a GPL (GNU General Public License) que conseguiu desenvolver funcionalidades necessários de um sistema operacional, porém, faltava um kernel, o núcleo do sistema responsável por integrar todos os componentes,

conquista feita por Linus Torvalds em cinco de outubro de 1991, que um ano depois integra o seu kernel de nome Linux ao projeto GNU gerando então o sistema GNU/Linux.

1.4.3 Distribuições

Uma distribuição pode ser definida como um conjunto de aplicativos junto com o núcleo (kernel) e manuais para funcionar de forma mais amigável[3]. Dentre as muitas existentes, cita-se; Red Hat Linux, Slackware Linux, Debian GNU/Linux, Mandrake, SuSE Linux, Conectiva Linux e PUx[4].

[3] http://www.vivaolinux.com.br
[4] https://www.vivaolinux.com.br/artigo/Linux-Breve-introducao-bom-para-iniciantes

1.5 REDES DE COMPUTADORES

1.5.1 Conceitos

A ideia primordial em redes de computadores é o compartilhamento de recursos (Tanenbaum, 2003), uma vez que dois ou mais computadores estejam conectados com acesso ou não a internet é possível usar recursos mútuos. Exemplo, em uma pequena empresa desprovida de recursos financeiros para investir em novos equipamentos para uso interno, as topologias de redes existentes ajudam a disseminar o que somente um dos hosts (computadores) tem acesso, por exemplo a uma impressora. O modelo de referência TCP/IP explicado mais adiante, exemplifica um protocolo usado em larga escala pelos inúmeros hosts (computadores) existentes para transferir arquivos, o *FTP*. Tanenbaum (2003, p.47) diz que "o protocolo de transferência de arquivos permite mover dados com eficiência de uma máquina para outra."

1.5.2 Tipos e Classificações de Redes

Usando o conceito que Tanenbaum (2009) aplicou a definição de redes de computadores, e com o avanço da tecnologia onde o compartilhamento da informação não mais se restringiu a dois ou três computadores, mas, sim, a umas dezenas, umas centenas, etc. surgiu a necessidade de distinguir a abrangência de alcance das mesmas.

Segundo Soares, Lemos e Colcher (1995, p.10) "O sistema de comunicação vai se constituir de um arranjo topológico interligando os vários módulos processadores através de

enlaces físicos (*meios de transmissão)* e de um conjunto de regras com o fim de organizar a comunicação(*protocolos*)".

Redes Locais (Local Área Networks – LANs): usadas para conectar computadores locais, a exemplo um campus de uma faculdade e possui restrição no alcance de disseminação do sinal podendo ser de 10 (dez) metros a alguns quilômetros.

Redes Metropolitanas (Metropolitan Área Networks – MANs): possui um alcance maior sendo possível atender uma cidade.

Redes Geograficamente Distribuídas (Wide Área Networks – WANs): interliga a âmbito geograficamente, ou seja, de um continente ao outro.

Assim afirma (Soares, Lemos e Colcher 1995, p.12):

> Por terem um custo de comunicação bastante elevado (circuitos para satélites e enlaces de microondas), tais redes são em geral públicas, isto é, o sistema de comunicação, chamado *sub-rede de comunicação*, é mantido, gerenciado e de propriedade de grandes operadoras (públicas ou privadas), e seu acesso é público.

1.5.3 Arquitetura TCP/IP

O modelo de referência TCP/IP se tornou o mais utilizado em razão dos seus dois protocolos, o TCP (*Transmission Control Protocol – Protocolo de controle de transmissão*) e o IP (*Internet Protocol*).

Tanenbaum (2003) define que o modelo *TCP/IP* surgiu com o crescimento da ARPANET que era uma rede de pesquisa patrocinada pelo Departamento de Defesa dos Estados Unidos (DoD). Na época as conexões eram feitas por via de linhas telefônicas dedicadas, com o surgimento das tecnologias de redes de rádio e satélites surgiram problemas para realizar a conexão de redes distintas de maneira uniforme, viu-se então a necessidade de criar um novo modelo de referência. O modelo foi projetado para ser flexível e preciso no seu funcionamento, de forma que fosse capaz de realizar transferência de arquivos e transmissão de voz em tempo real. Sobre essas premissas o modelo foi subdivido em 4(quatro) camadas; camada de host/rede, camada de inter-redes, camada de transporte e camada de aplicação. Tanenbaum (2003, p.49) assim apresenta:

- **Camada de Host-Rede:** "O modelo de referência TCP/IP não especifica muito bem o que acontece ali, exceto o fato de que o host tem de se conectar à rede utilizando algum protocolo para que seja possível enviar pacotes IP."
- **Camada de Inter/Redes:** Define o protocolo IP (Internet Protocol) usado para realizar a comunicação entre os host e/ou enviar e receber dados.
- **Camada de Transporte:** é definida pelos protocolos *TCP(Transmission Control Protocol – protocolo de controle de transmissão),* é um protocolo orientado a conexões confiável que permite a entrega sem erros de um fluxo de bytes originário de uma determinada máquina em qualquer computador da inter-rede e o protocolo UDP *(User Datagram Protocol – protocolo de datagrama do usuário),* é um protocolo sem conexão e não-confiável destinado a aplicações que não querem controle de fluxo nem manutenção da sequência das mensagens enviadas, e desejam fornecer seus próprios recursos para isso.
- **Camada de Aplicação:** Aqui ficou concentrado os protocolos de nível mais alto responsáveis pela comunicação, envio, recebimento, dentre outros. A exemplo o HTTP usado para busca e navegação na internet, FTP usado para transferência de arquivos entre hosts, SNMP usado para gerenciar mensagens tramitadas via e-mail, DNS usado para converter nomes de domínios (sites) em números IP, dentre outros.

1.5.4 Protocolo IPv4

Segundo os autores Kurose (2010) e Tanembaum (2003), o protocolo IPv4 tem sua estrutura definida através de um datagrama onde são definidos os seus limites de aplicabilidade. Trata-se de um protocolo responsável pelo controle e gerenciamento das informações transmitidas pela rede, o mesmo tem em seu datagrama campos responsáveis por controlar sua versão, tamanho do cabeçalho do pacote que será transmitido, qual tipo de serviço que será provido, tempo de vida desse pacote, o protocolo que será usado dentre outros.

Conforme explica Kurouse (2010, p.252) sobre o endereçamento do protocolo IPv4:

> Cada endereço IP tem cumprimento de 32 bits (equivalente a 4 bytes). Portanto, há um total de 2^{32} endereços IP possíveis. Fazendo uma aproximação de 2^{10} por 10^3, é fácil ver que há cerca de 4 bilhões de endereços IP possíveis. Esses endereços são

> escritos em notação decimal separa por pontos (dotted-decimal notation), na qual cada byte do endereço é escrito em sua forma decimal e separado dos outros bytes do endereço por um ponto. Por exemplo, considere o endereço IP 193.32.216.9. o 193 é o número decimal equivalente aos primeiros 8 bits do endereço; o 32 é o decimal equivalente ao segundo conjunto de 8 bits do endereço e assim por diante. Por conseguinte, o endereço 193.32.216.9, em notação binária, é:
>
> 11000001 00100000 11011000 00001001.

Entende-se que protocolo IPv4 é o responsável por gerenciar as diferentes classes de redes, que após definidas seja A, B, C, ou D; o mesmo fará as especificações do IPs juntamente com suas respectivas mascaras de rede e os protocolos que tramitarão no envio e recebimento pelos hosts.

1.6 SERVIÇOS DE REDES DE COMPUTADORES

1.6.1 Servidores de Rede

Ferreira (2008) apresenta duas formas para que um servidor de rede possa ser inicializado, sendo por um superservidor (super daemon) usado para iniciar servidores como o Talk que possibilita a comunicação entre usuários, o Telnet usado para realizar conexões remotas via terminal nos hosts (computadores), o RSH semelhante ao telnet sendo possível realizar a cópia de arquivos de forma remota e o rsync usado para realizar a sincronização de dados entre computadores ou de maneira independente como o SSH que se usado como usuário root da total controle sobre o host.

São taxados assim, pois é deles a responsabilidade de iniciar os demais servidores que compõem a rede. No Linux e demais sistemas Unix tem o inetd, mais antigo e o xinetd, mais recente e detém maior quantidade de recursos.

Para gerenciar o uso dos serviços disponibilizados pelos servidores Linux controlando os usuários de rede – hosts se os mesmos tem ou não permissão para acessar tais serviços, aconselha o uso do *tcp wrappers*.

1.6.2 Servidores GNU/Linux

Hunt (2004) menciona alguns servidores usados para prover os mais diversos serviços de rede, dentre os quais citados temos o *Servidor de Internet, Servidor DNS, Servidor de*

Correio, Servidor Web Apache, Servidor Gateway de Rede, Servidor Departamental, Servidor SAMBA.

1.6.3 Serviço de Autenticação e Controlador de Domínio – Samba

Stato Filho (2004, p.313) afirma que "O samba é um conjunto de aplicativos rodando sobre a plataforma **Linux**, que utiliza um protocolo chamado SMB - Server Message Block nativo do Windows. Ele é utilizado em redes Windows para compartilhar recursos, tais como impressoras, discos."

O SAMBA auxilia a rede podendo atuar como servidor de arquivos completo; Controlador de Domínio atuando no gerenciamento das contas de usuários vinculadas a cada host da rede, inserindo esses hosts no domínio para que possam fazer uso dos recursos disponibilizados.

1.6.4 Serviço de Proxy – Squid

Proxy possui um serviço de *cache* que após ser configurado mantém uma lista dos sites mais acessados pelos hosts da rede e quando forem acessá-los novamente não será necessário carregar todo o site, possui métodos de limitar o uso de acesso à internet criando ACL - *Access Control List*. Para Stato Filho (2004, p. 220), "As ACLs permitem especificar endereços de origem ou destino, domínios, horários, portas ou métodos de conexão ao proxy, que serão utilizadas para permitir ou negar acessos", dentre outras funcionalidades.

Segundo o site <www.squid-cache.org> o squid é:

> Um proxy para a Web de suporte HTTP, HTTPS, FTP, e muito mais. Ele reduz a largura de banda e melhora os tempos de resposta de cache e reutilização de páginas web frequentemente solicitadas. Squid tem amplos controles de acesso e faz um grande acelerador de servidor. Ele é executado na maioria dos sistemas operacionais disponíveis, incluindo Windows e está licenciado sob a GNU GPL.

1.6.5 Serviço de Arquivos – OwnCloud

Conforme afirma o site < https://owncloud.org/features/> “ownCloud é uma sincronização de arquivos auto-hospedado e servidor de compartilhamento”. Trabalha com o conceito de arquivos em nuvem, ou seja, dispensa o armazenamento local dos dados seja ele qual for o formato, tamanho, etc.

1.7 SEGURANÇA EM REDES DE COMPUTADORES

1.7.1 Conceitos

Tanenbaum (2003) afirma que para existir segurança no envio e recebimento dos bits contidos nos pacotes trafegados na rede, tanto a criptografia quanto as verificações de integridades devem ser feitas de modo fim a fim, ou seja, na camada de aplicação. “ o processo de origem criptografa e/ou protege a integridade dos dados e os envia ao processo e destino, onde eles são descriptografados e/ou verificados.” (TANENBAUM, 2003, p. 821).

1.7.2 Conexões Seguras – Criptografadas

Strebe e Perkins (2002) afirmam que a criptografia dentre suas mais diversas formas de uso pode ser usada em redes de computadores para “esconder”, ou seja, proteger os seguintes tipos de dados: “Comunicações particulares, Armazenamento de arquivos que precisam estar protegidos, Autenticação de usuários ou de computadores, Intercâmbio seguro de senhas”. (STREBE; PERKINS, 2002, p. 104).

Para que a navegação na internet se torne um local seguro para trafegar dados e os mesmos sejam trafegados de forma segura até seus destinatários, utiliza-se dois protocolos, o HTTP que transporta as informações em texto claro sem o uso de criptografia e o HTTPS que transporta as informações criptografadas por meio de certificados digitais e com o auxílio dos protocolos SSL (*Secure Sockets Layer*), TLS (*Transport Layer Security*) para que as informações sejam entregues de forma segura aos seus destinatários. (<http://cartilha.cert.br/uso-seguro/>).

Dentre os tipos de conexões seguras o site <cartilha.cert.br/uso-seguro/> detalha o protocolo HTTP - Hypertext Transfer Protocol, ou seja, Protocolo de Transferência de Hipertexto usado na maior parte dos acessos aos sites, o mesmo não fornece quaisquer meios de segurança.

Figura 4: Exemplo do uso do protocolo http

Fonte: http://cartilha.cert.br/uso-seguro/

Também o protocolo HTTPS - Hyper Text Transfer Protocol Secure, ou seja, Protocolo de Transferência de Hipertexto Seguro que possibilita a proteção dos dados trafegados através da rede.

Figura 5: Exemplo do uso do protocolo https

Fonte: http://cartilha.cert.br/uso-seguro/

Na conexão segura com EV SSL especifica os mesmos recursos do *https* com um incremento da possibilidade de gerar o certificado do site que é exibido na barra do navegador seguido pelo protocolo usado e em seguida o endereço do site.

Figura 6: Exemplo do uso de certificado digital com protocolo https

Fonte: http://cartilha.cert.br/uso-seguro/

Também há o uso de certificados auto assinado, ou seja, gerou o certificado, porém o navegador não o reconheceu, o mesmo pode ser caracterizado como nível intermediário de segurança por possui a falha na autenticação, oscila entre seguro e inseguro.

Figura 7: Exemplo do uso de certificado auto assinado

Fonte: http://cartilha.cert.br/uso-seguro/

Já determinados sites optam por usar uma forma mista do protocolos *http e https* como forma de proteção.

Figura 8: Exemplo de uso dos protocolos http e https

Fonte: http://cartilha.cert.br/uso-seguro/

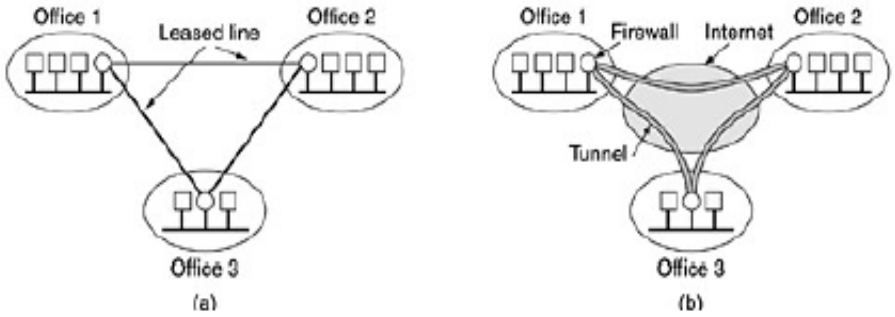
De forma mais abrangente Tanenbaum *(2003)* aborda o tema de segurança em redes de computadores passando por conceitos como o IPsec - IP security um projeto criado para definir a abrangência da criptografia de dados para usuários, em um contexto ao qual usuários entendiam que a mesma era opcional por não acrescentar benefícios ao uso computacional. “Os principais serviços são sigilo, integridade de dados e proteção contra ataques de reprodução (o intruso reproduz uma conversação).” (TANENBAUM, 2003, p. 821).

Firewall

Dentre as infinitas formas de uso de um firewall, Tanenbaum (2003) especifica que a ideia básica por trás de um firewall é impedir a entrada de intrusos e a saída de dados secretos. Onde pode ser controlado tudo ou quase tudo que os usuários internos e/ou externos podem ou não fazer na rede, por exemplo o uso especifico de uma porta na qual irá ser acessado um serviço fornecido pela rede.

Quanto a camuflagem de redes surgiu o conceito de VPNs - Virtual Private Networks, exibido na Figura 4, forma de manter uma rede sobre as redes públicas usando recursos virtuais, como circuitos e memorias. Tanenbaum (2003).

Figura 9 (a) Uma rede privada de linha dedicada (b) Uma rede privada virtual



Fonte: (TANENBAUM, 2003, p. 829)

Redes sem fio

“O padrão 802.11 prescreve um protocolo de segurança do nível de dados, chamado WEP - Wired Equivalent Privacy, projetado para tornar a segurança de uma LAN sem fio tão boa quanto a de uma LAN fisicamente conectada.”. (TANENBAUM, 2003, p. 830).

Protocolos de autenticação

Processo ao qual diferencia autorização de autenticação, onde segundo Tanenbaum (2003, p. 834) “A autenticação lida com a questão de determinar se você está ou não se comunicando com um processo específico. A autorização se preocupa com o que esse processo tem permissão para fazer.” Ele usa como exemplo a exclusão de um determinado arquivo que está em um servidor, onde o processo do usuário solicita a exclusão do arquivo e o servidor tem a responsabilidade de fazer a verificação quanto a autenticação do usuário, se ele é mesmo quem diz ser e depois verificar a autorização do usuário para ver quais são suas permissões de uso desse arquivo.

2. MÉTODOS E MATERIAIS

2.1 CLASSIFICAÇÃO METODOLÓGICA

2.1.1 Quanto aos Fins

Quanto à natureza da pesquisa, classifica-se como pesquisa *Experimental* pois necessita que seja determinado um objeto de estudo, deste surgir um problema que gere hipóteses a serem testadas e provadas seja como verdadeiras ou falsas. A mesma dá a oportunidade de ser criada em ambientes de testes usando informações do ambiente real ao qual será implementada.

"O método experimental consiste essencialmente em submeter os objetos de estudo à influência de certas variáveis, em condições controladas e conhecidas pelo investigador, para observar os resultados que a variável produz no objeto." (GIL, 1999, p.33)

A presente pesquisa apresenta-se como *experimental* por contar com técnicas de simulação, com a criação de um servidor por meio de um ambiente virtual. E posteriormente teste de viabilidade de implementação em uma entidade pública.

Classifica-se também como Pesquisa *Aplicada* uma vez que essa pratica é focalizada na solução de problemas específicos do meio ao qual será introduzida, ou seja, uma pesquisa voltada para uma área específica que é a de tecnologia de informação.

2.1.2. Quanto aos Meios

Classifica como Pesquisa descritiva para a qual será necessário o estudo bibliográfico para dominar os diversos temas que serão abordados para fundamentar a criação dos serviços que serão inseridos no ambiente usado como base para a pesquisa.

O foco da pesquisa são os problemas de comunicação e acessibilidade de dados da prefeitura municipal de Novo Cruzeiro, que será a beneficiária do servidor produzido na pesquisa.

Neste sentido, Andrade (2001), afirma que:

> O método monográfico consiste no estudo de determinados indivíduos, profissões, condições, instituições, grupos ou comunidades, com a finalidade de obter generalizações[...] pode, também, abranger o conjunto das atividades de um grupo social particular, como no exemplo das cooperativas e do grupo indígena. (ANDRADE, 2001, p.135)

2.1.3 Tratamento de Dados

Quanto à abordagem, a pesquisa, é classificada por *pesquisa qualitativa*. Para o alcance dos objetivos e resultados, os dados receberão um tratamento de cunho *qualitativo*, ou seja, sem a utilização de ferramental estatístico e quantitativo para evidenciá-los, pois, os resultados aos quais o projeto se concentra não são numéricos, e sim de compreensão e aceitação do meio no qual será inserido.

Para alcançar os objetivos e testar a veracidade das hipóteses propostas como solução do problema mencionado, o trabalho terá os seguintes passos, descrito no cronograma a seguir.

2.2 MATERIAIS

Tendo como base física um computador Desktop com processador Pentium(R) Dual-Core CPU E5800 @ 3.20GHz e 16 GB de Memória RAM, foi instalado a distribuição Debian na versão 8 64 bits sobre o kernel 3.16.0-4, um switch de 16 portas não gerenciável, um roteador e um smartphone Samsung Galaxy A3 2016. Como forma de integração entre os

serviços que funcionara sobre o computador foi necessário instalar o software com o comando *apt-get install bridge-uitls* para modificar o comportamento padrão das placas de rede transformando-as em "pontes" para que o KVM – Kernel-based Virtual Machine fizesse uso das mesmas, manipulei o arquivo */etc/network/interfaces* onde são armazenadas informações das placas de redes que irá transmitir os pacotes da internet para a rede interna e vice-versa.

Uma vez acessado o arquivo com o comando *nano /etc/network/interfaces* foi criado duas Bridge: uma para comunicação com a internet e outra para a rede interna a fim de que as placas físicas se comuniquem com as placas das máquinas virtuais do KVM e posteriormente tornar possível que os clientes (hosts) usufruam dos serviços disponibilizados por cada máquina.

Figura 10: Criação das Bridges nas placas Físicas

```
# Bridges da eth0 e eth1
auto brwan0 brlan0

# Bridge eth0
iface brwan0 inet static
        address 192.168.0.105
        netmask 255.255.255.0
        gateway 192.168.0.1
        bridge_ports eth0
        bridge_maxwait 0

# Bridge eth1
iface brlan0 inet static
        address 192.168.2.1
        netmask 255.255.255.0
        bridge_ports eth1
        bridge_maxwait 0
```

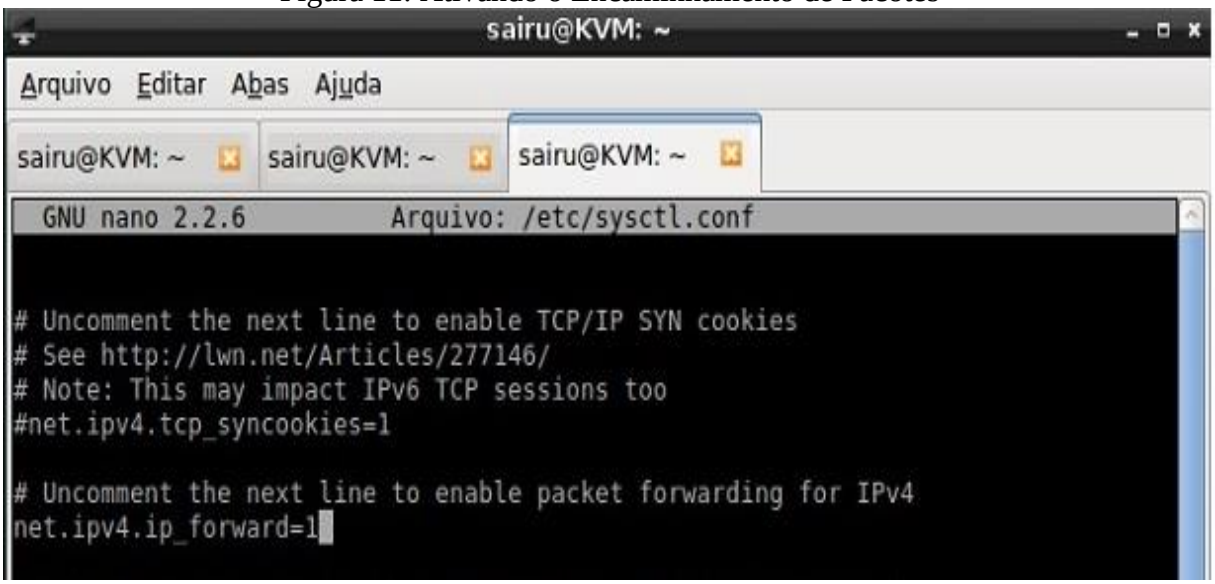
Fonte: Do Próprio Pesquisador

Na placa eth0 foi criado a Bridge[5] brwan0, a qual foi incumbida a responsabilidade de fornecer internet para as máquinas virtuais do KVM e na eth1 a brlan0 para fazer a comunicação da rede interna com os hosts clientes (Figura 10). Nas configurações das placas foram inseridos o número do IP, a máscara da rede, o gateway, seguido de qual interface que essa configuração deve ser aplicada *bridge_ports eth0* ou *eth1*, e por último qual tempo de espera que a placa aguardará pelo pacotes da rede *bridge_maxwait 0*[6], zero significa que não haverá espera alguma.

[5] https://wiki.debian.org/BridgeNetworkConnections
[6] http://manpages.ubuntu.com/manpages/trusty/man5/bridge-utils-interfaces.5.html

Feito isto, foi ativado o encaminhamento de pacotes modificando o arquivo *systcl.conf* de 0 para 1 para encaminhar os dados que viriam das máquinas virtuais para a internet e/ou para a rede interna, conforme apresentado na Figura 11.

Figura 11: Ativando o Encaminhamento de Pacotes



Fonte: Do Próprio Pesquisador

E sobre o mesmo computador físico foi instalado a plataforma de virtualização KVM[7] responsável por administrar e gerenciar os serviços de rede propostos para o trabalho. Para cada serviço foi criado um meio de armazena-lo de forma independente, com o auxílio da ferramenta virt-manager[8], na qual foram criadas máquinas virtuais com os devidos serviços detalhados a seguir.

A máquina DHCP – Protocolo de Configuração Dinâmica de Host foi configurada de forma a prover a faixa de IPs que serão disponibilizados para os hosts da rede, direcionando qual domínio usar e qual rota, ou seja, qual gateway da rede.

A máquina DNS – Domain Name System foi implementada para a resolução de nomes de forma recursiva seja via IP ou string, onde a mesma consegue traduzir pesquisas internas seja usando o IP do host ou o seu hostname.

A máquina SAMBA – Servidor de Arquivos foi desenvolvida para funcionar como um PDC ou seja um controlador de domínio, a mesma é dada a responsabilidade de supervisionar todas as ações que cada usuários fará uma vez logado na rede. Para cada

[7] https://wiki.debian.org/KVM
[8] https://virt-manager.org/

departamento foram criados setores, onde somente usuários pertencentes ao grupo vinculados ao setor terá acesso para controlar os arquivos do mesmo.

A máquina OWNCLOUD – Servidor de Arquivos em Nuvem foi criada para gerar os meios de Backup dos arquivos dos usuários como forma de prevenção a possíveis falhar internas.

Em cada que necessita de acesso continuo a internet foi criado um Firewall como forma de prevenção e proteção aos dados tramitados.

3 SOLUÇÃO

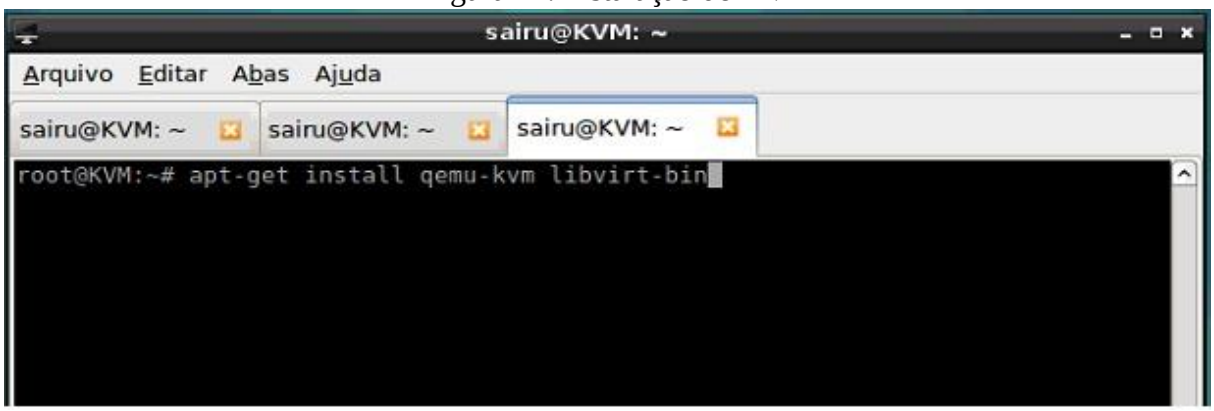
3.1 CONFIGURAÇÃO DOS SERVIDORES

Foi utilizado o KVM para desenvolvimento do servidor com uma instalação usando os repositórios oficiais da distribuição Debian na versão 8.

Para que não ocorressem falhas durante a instalação dos pacotes que seriam baixados, foram utilizados os comandos *apt-get update e apt-get upgrade*, apresentado na Figura 12, o primeiro para atualizar os repositórios da distribuição e o segundo para uma verificação de possíveis atualizações da distribuição, seja ela para o próprio sistema ou para os softwares pré-instalados.

Em seguida com o comando *apt-get install* seguido do emulador e da biblioteca responsáveis por gerenciar o KVM.

Figura 12: Instalação do KVM



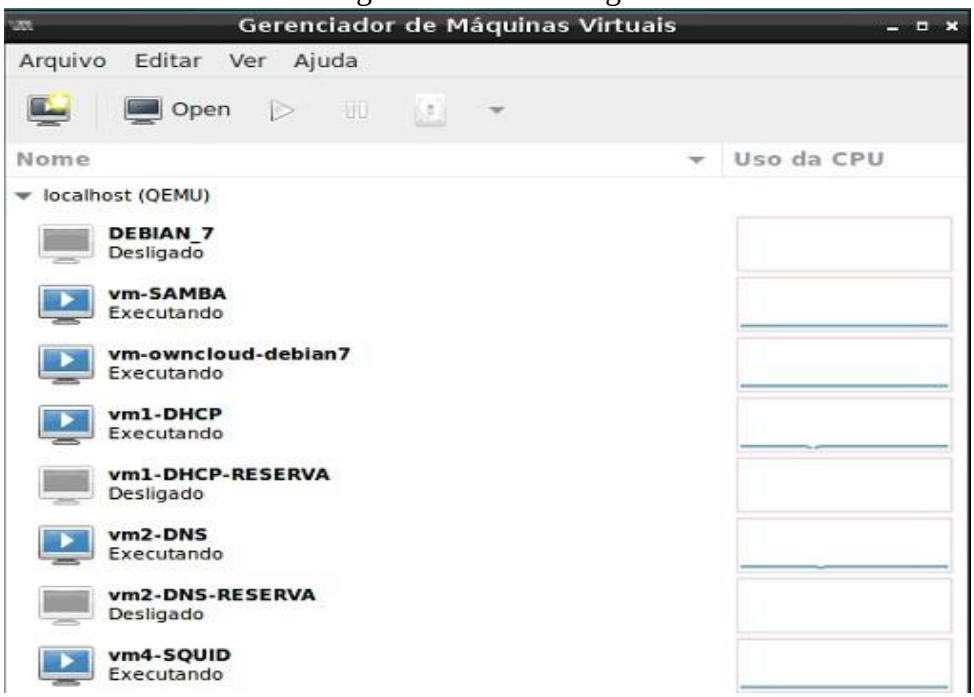
Fonte: Do próprio Pesquisador

Terminado a instalação foi inserido um usuário para administra-lo usando o comando: *adduser nomedousuario kvm, adduser nomedousuario libvirt.*

O mesmo ofertou a possibilidade de configurar os serviços em máquinas separadas para que depois das devidas configurações fossem integrados, funcionando de forma homogênea. Também foi instalado a ferramenta Virt-Manager *apt-get install virt-manager*

para auxiliar de forma gráfica no desenvolvimento das máquinas conforme demostrado na Figura 13.

Figura 13: Virt-Manager



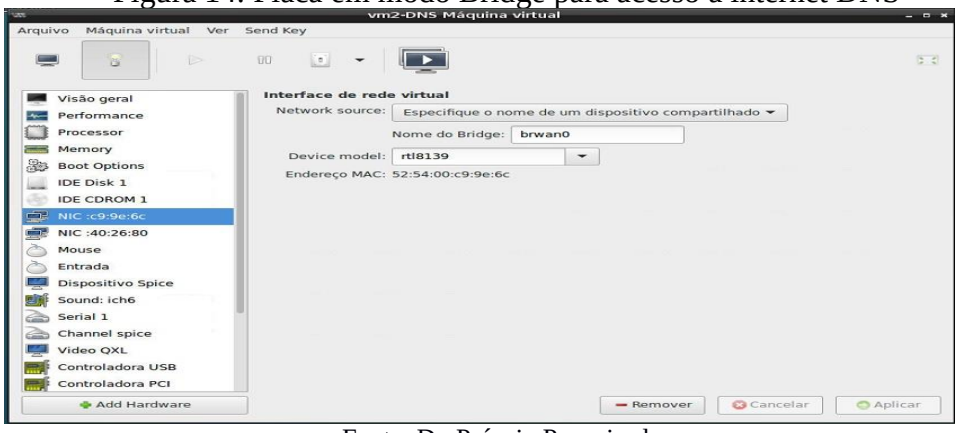
Fonte: Do Próprio Pesquisador

O servidor foi montando usando a seguinte estrutura: Servidor DNS; Servidor DHCP, Servidor Squid; Servidor Samba e Servidor Owncloud. Cada servidor será detalhado em tópico específico, a seguir.

3.1.1 Servidor Dns

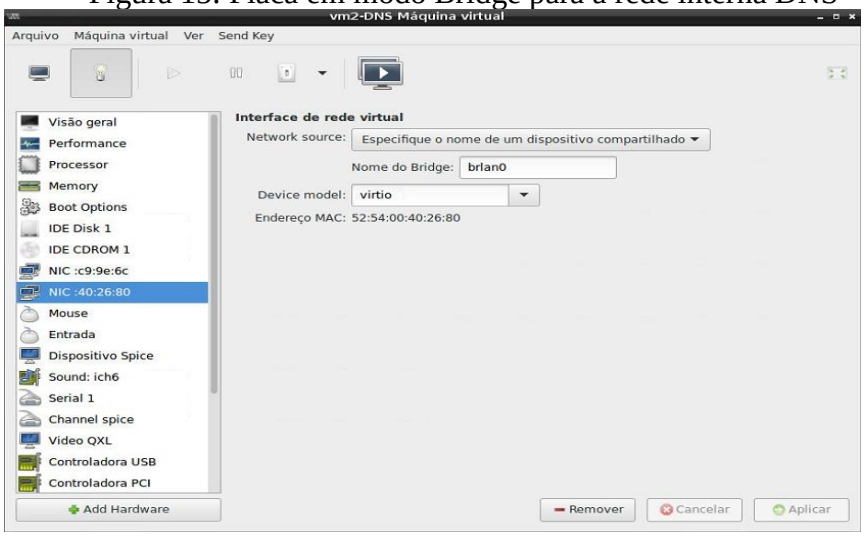
Antes de iniciar a instalação do sistema operacional, foram configuradas duas placas em modo bridge, uma para acesso à internet fazendo ponte com a brwan0 do servidor físico e outra com a brlan0 para a rede interna, conforme apresentado na Figura 14 e 15.

Figura 14: Placa em modo Bridge para acesso à internet DNS



Fonte: Do Próprio Pesquisador

Figura 15: Placa em modo Bridge para a rede interna DNS

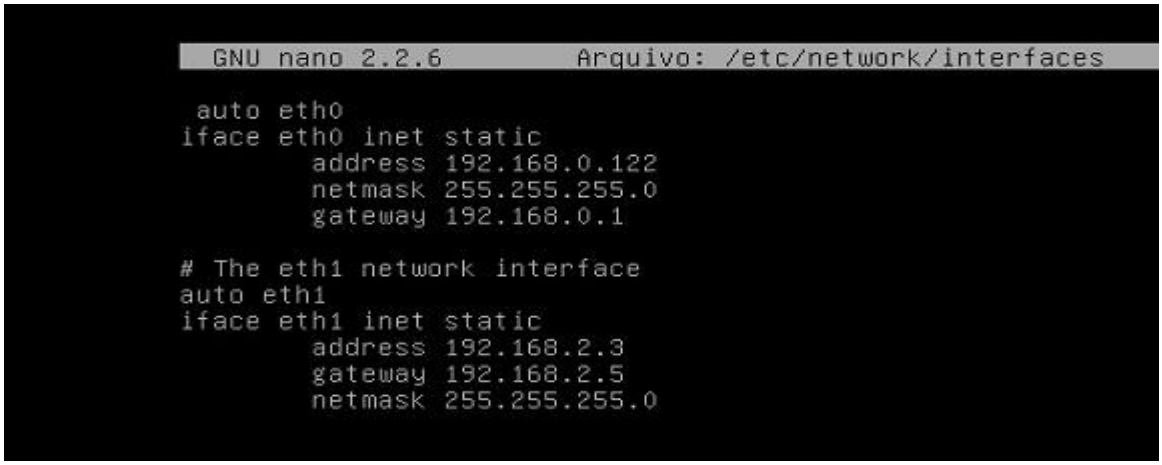


Fonte: Do Próprio Pesquisador

Feito isso, foi instalado a distribuição Debian 8 em modo gráfico e em seguida usando o comando *systemctl set-default multi-user.target* em modo root para remover o modo gráfico, facilitando as operações que viriam a serem feitas.

Uma vez instalado o sistema operacional, foi acessado o arquivo */etc/network/interfaces* e fixado os IPs referentes a cada placa criada anteriormente. Os mesmos poderiam ser usados via DHCP, mas por se tratar de um servidor, optou-se por coloca-los fixos.

Figura 16: Configuração das Placas do Servidor DNS

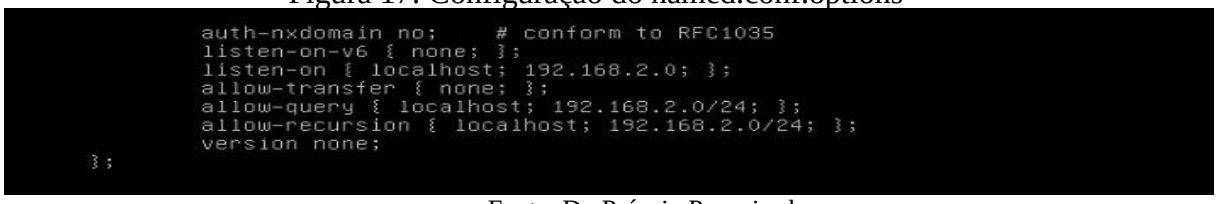


Fonte: Do Próprio Pesquisador

Para a instalação do DNS, foi usado o comando *apt-get install bind9 dns-utils,* trata-se de um software onde são realizadas as configurações e um conjunto de ferramentas para testes da rede, sendo possível saber se o mesmo está resolvendo nomes. Assim como já mencionando anteriormente, foi ativado o encaminhamento de pacotes no arquivo */etc/systcl.conf* mudando o valor de 0 para 1.

Feito o download das dependências, dá-se início as configurações internas do bind9[9], para a resolução de nomes internos. De início foi acessado o arquivo /etc/bind/named.conf.options e especificado as regras de uso globais que abrangem aspectos de segurança e dimensão de uso do mesmo, como em qual rede o mesmo atuara, se atuara como DNS secundário, etc.

Figura 17: Configuração do named.conf.options



Fonte: Do Próprio Pesquisador

Seguindo uma sequência cronológica o arquivo foi configurado com as seguintes regras:

✓ Listen-on-v6 – Se o servidor irá resolver nomes IP na versão6, determinado none, ou seja, não.

[9] https://wiki.debian.org/Bind9

✓ Listen-on – controla em quais interfaces o DNS atuará, como foi configurado para funcionar em uma máquina separada, o mesmo resolverá nomes localmente e em toda a rede 2.0.

✓ Allow-transfer – se o servidor irá responder a um servidor secundário.

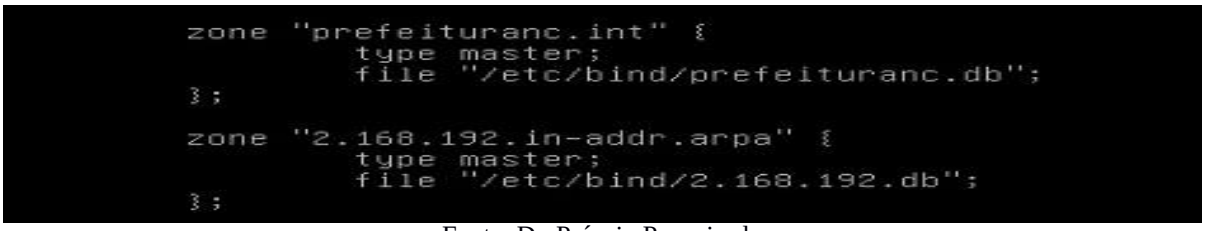
✓ Allow-query – determina de quais locais poderão ser feitas consultas no servidor.

✓ Allow-recursion – especifica de onde poderão ser feitas pesquisas recursivas direcionadas ao servidor, seja via IP ou usando o hostname (nome) da máquina.

✓ Version – se o servidor irá informa sua versão.

Implementadas essas configurações, é necessário configurar o arquivo /etc/bind/named.conf.local para que sejam especificados os diretórios e nomes dos arquivos de configuração das zonas que irão controlar as buscas por IP ou por nome.

Figura 18: Configuração do named.conf.local



Fonte: Do Próprio Pesquisador

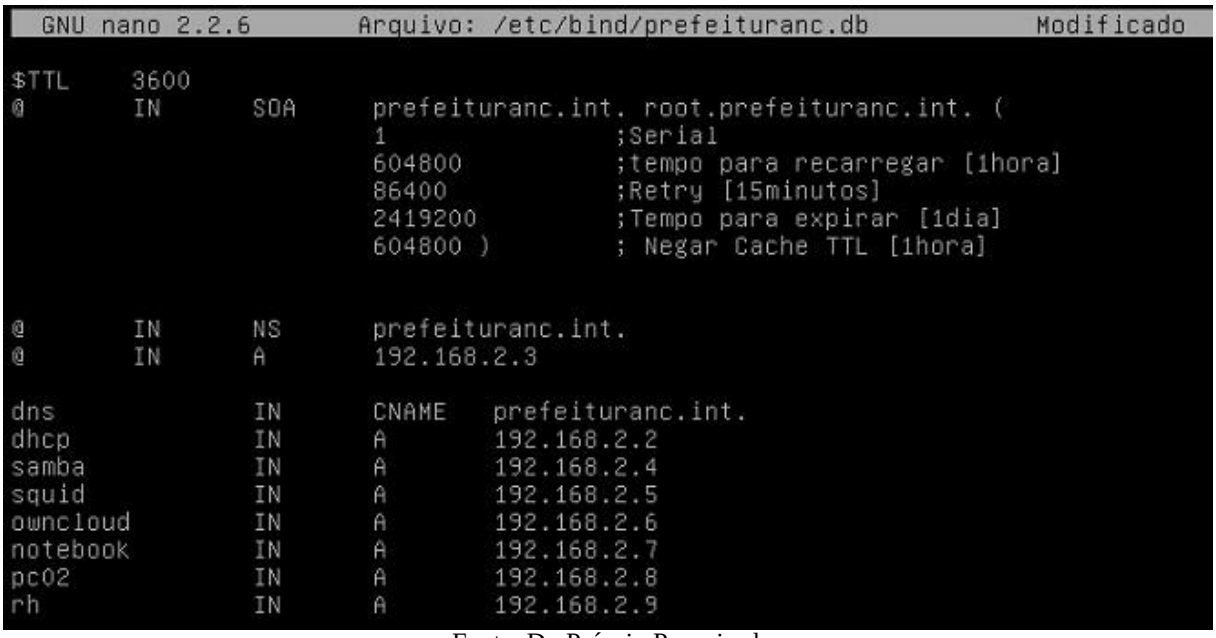
✓ Zone – nome do arquivo para busca via hostname

✓ Type master – a hierarquia da zona a ser criado, se será principal ou secundaria.

✓ File – o caminho do diretório no qual ficará armazenado o arquivo contendo suas configurações.

Após a especificação de localização dos arquivos, tem-se então a necessidade de cria-los para que sejam definidas as diretivas de funcionamento do domínio que será configurado.

Figura 19: Arquivo de Configuração da Zona e Domínio no Bind9



Fonte: Do Próprio Pesquisador

Conforme Especificado na Figura 19:

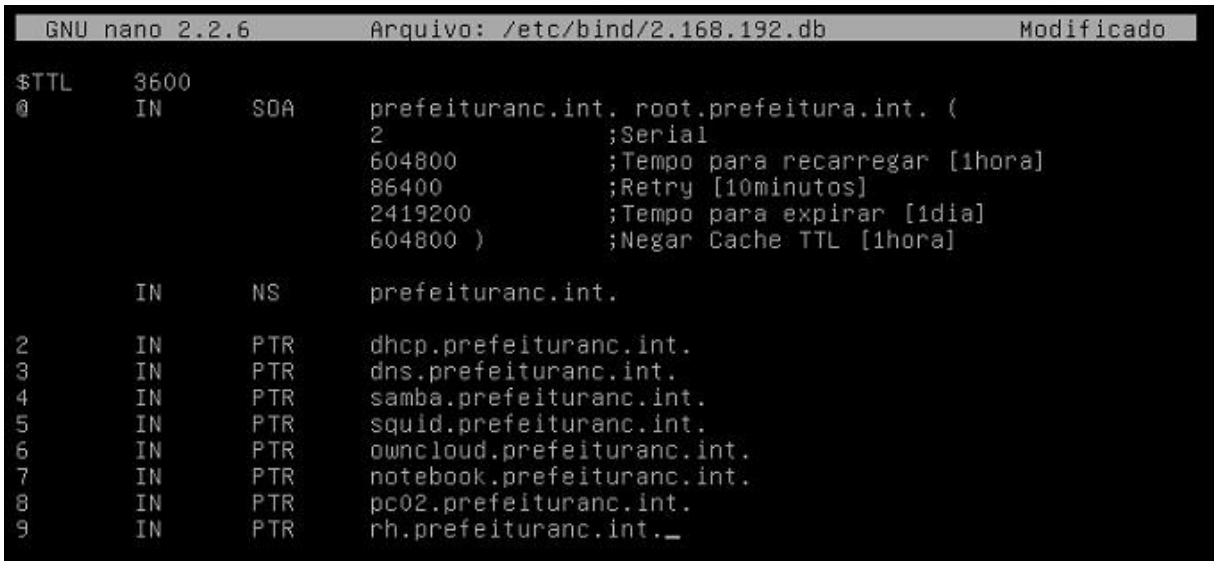
- Primeira linha: especifica o valor padrão da zona, Time to Live (TTL) que cada recurso da zona ficará disponível para serem armazenados em cache por servidores secundários/clientes.
- Segunda linha: @ dá início a criação da zona tendo como servidor de nomes primário o prefeituranc.int e usando o usuário root como raiz para validação da zona.
- Terceira linha: Controla as modificações feitas nesse arquivo.
- Quarta linha: Tempo usado para verificar se houve alguma alteração na zona.
- Quinta linha: Tempo de espera usado para solicitar uma atualização se o servidor parar de resolver nomes.
- Sexta linha: Tempo decorrido em que o servidor de nomes ficar inoperante, não irá resolver nomes na rede.
- Sétima linha: Tempo para armazenamento de informações da rede
- Oitava linha: aponta onde ficará localizado o servidor de nomes NS.
- Nova linha: faz uso do atributo A para apontar em qual IP o servidor de nomes será localizado.
- Decima linha: faz o mapeamento do hostname do servidor usando o atributo CNAME, criando um “alias”, ou seja, uma espécie de link para o nome do mesmo, uma vez que ele está configurado na máquina a qual está sendo referenciado.

Nas próximas linhas do arquivo são inseridos os clientes que farão uso desse servidor, usando o atributo A para referenciar o seu hostname a um endereço e IP para que seja possível posteriormente, realizar buscas na rede pelos mesmos.

Na Figura 20 é demostrada a configuração da zona reversa, que segue o mesmo padrão da configuração do início da zona, com uma variação na configuração dos hosts clientes que irão usufruir da rede local. Para que sejam feitas pesquisas reversas usando string ou seja, o hostname (nome) dos clientes, é necessário usar o parâmetro PTR que funciona como um ponteiro para localizar o host.

Conforme demonstrado abaixo na Figura 20, para cada host foi usado o seu número IP que o identifica na rede com o ponteiro indicado seu hostname atrelado ao domínio, possibilitando a realização de buscas pelo nome do host.

Figura 20: Arquivo de Configuração da Zona Reversa no Bind9



Fonte: Do Próprio Pesquisador

Figura 21: Configuração do Resolv.conf

Fonte: Do Próprio Pesquisador

De posse dessas configurações, para que o servidor de nomes trabalhe com a tradução de nomes na rede interna, foi acessado o arquivo */etc/resolv.conf* onde são armazenadas informações do servidor usado para fazer essa função e alterado com as

especificações que foram feitas anteriormente como nome do domínio, local de busca e o IP do servidor que fará esse serviço, conforme demostrado na Figura 21. Como o serviço foi configurado na própria máquina, foi inserido o IP 127.0.0.1 para que a mesma realize essa tarefa.

Para que as configurações entrassem em funcionamento fez-se uso do comando /etc/init.d/bind9 restart, para atualizar as alterações feitas. E por fim foi criado o seu firewall, a fim de manter as conexões estáveis na rede interna.

- ✓ INPUT – analisa tudo que entra no servidor
- ✓ OUTPUT – analisa tudo que sai do servidor
- ✓ FORWARD – analisa tudo que passa pelo servidor
- ✓ PREROUTING – redireciona pacotes para portas especificas
- ✓ POSTROUTING – mascara os pacotes ocultando e identificando o IP
- ✓ ACCEPT – deixa a requisição passar
- ✓ DROP – bloqueia a requisição e não emite aviso ao solicitante
- ✓ REDIRECT – transfere a requisição para uma porta especificada
- ✓ MASQUERADE – faz o mascaramento dos pacotes
- ✓ -F: apaga todas as regras do firewall
- ✓ -X: apaga uma regra específica
- ✓ -t: qual tabela a ser usada
- ✓ -A: insere uma nova regra
- ✓ -s: destino de origem
- ✓ -d: destino final
- ✓ -j: solicita uma ação a ser aplicada
- ✓ -to-port: porta que será usada
- ✓ -i interface de rede a ser usada
- ✓ -p: solicita um protocolo
- ✓ -j – define a ação a ser tomada
- ✓ --dport – porta de destino
- ✓ Limit – limita a quantidade de tentativas para execução da regra
- ✓ State – define um estado para a conexão
- ✓ Icmp – Internet Control Message Protocol
- ✓ -syn – permite o uso dos bits ACK E FIN
- ✓ NEW – pacote que usa uma nova conexão

- ✓ ESTABLISHED – pacotes que tem conexão existente
- ✓ RELATED – pacote que não está na conexão existente
- ✓ INVALID – pacote desconhecido ou não identificado

```
#! /bin/bash
###LIMPANDO AS REGRAS
iptables -F
iptables -x
iptables -t nat -F
iptables -t nat –X
####ALTERANDO AS POLITICAS DAS CADEIAS
iptables –P INPUT ACCEPT
iptables –P OUTPUT ACCEPT
iptables –P FORWARD ACCEPT
####TRAFEGO DNS BIND9
iptables –A INPUT –s 192.168.2.0/24 –p tcp –dport 53 –j ACCEPT
iptables –A INPUT –s 192.168.2.0/24 –p udp –dport 53 –j ACCEPT
iptables –A FORWARD –s 192.168.2.0/24 –p tcp –dport 53 –j ACCEPT
iptables –A FORWARD –s 192.168.2.0/24 –p udp –dport 53 –j ACCEPT
######BLOQUEANDO PING Of Death ou DoS
iptables –A INPUT –p icmp –icmp-type echo-request –m limit –limit 1/s –j ACCEPT
iptables –A FORWARD –p icmp –icmp-type echo-request –j DROP
iptables –A FORWARD –p icmp –icmp-type echo-request –m limit –limit 1/s –j ACCEPT
####EXCLUINDO PACOTES SUSPEITOS
iptables –A INPUT –m state –state INVALID –j DROP
iptables –A INPUT –p tcp ! –syn –m state –state NEW –j DROP
iptables –A FORWARD –m state –state INVALID –j DROP
iptables –A FORWARD –p tcp ! –syn –m state –state NEW –j DROP
###CONFIRMA CONEXOES JA ESTABELECIDAS
iptables –A FORWARD –m state –state RELATED,ESTABISHED –j ACCEPT
```

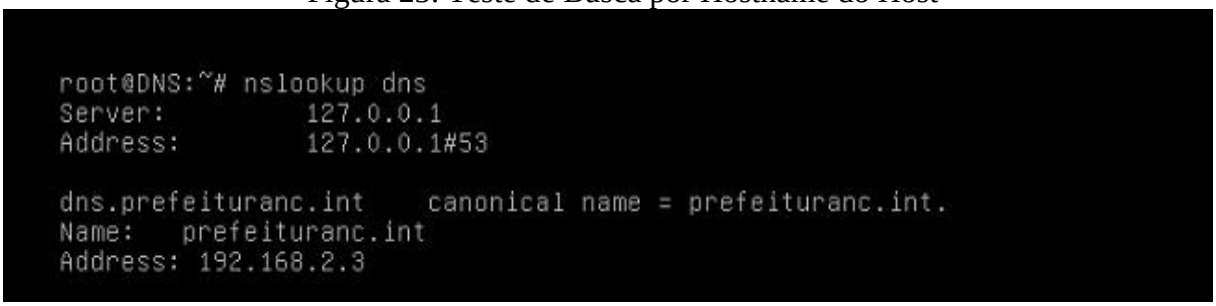
Para verificar a validação das configurações, fiz uso do conjunto de ferramentas que foram instaladas junto do Bind9, o *dns-utils*, o mesmo fornece meios para que seja verificado a precisão das implementações.

Figura 22: Teste de Resolução de Nomes

Fonte: Do Próprio Pesquisador

Como visto acima na Figura 22, o servidor está resolvendo nomes localmente na porta 53 à qual é usada pelo dns.

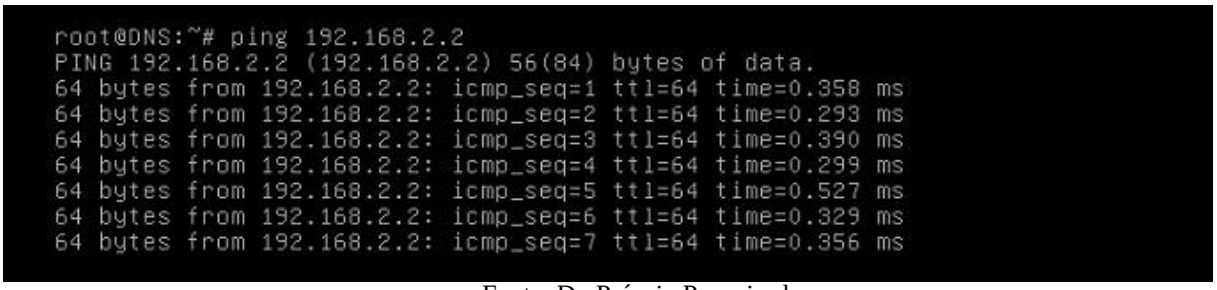
Figura 23: Teste de Busca por Hostname do Host



Fonte: Do Próprio Pesquisador

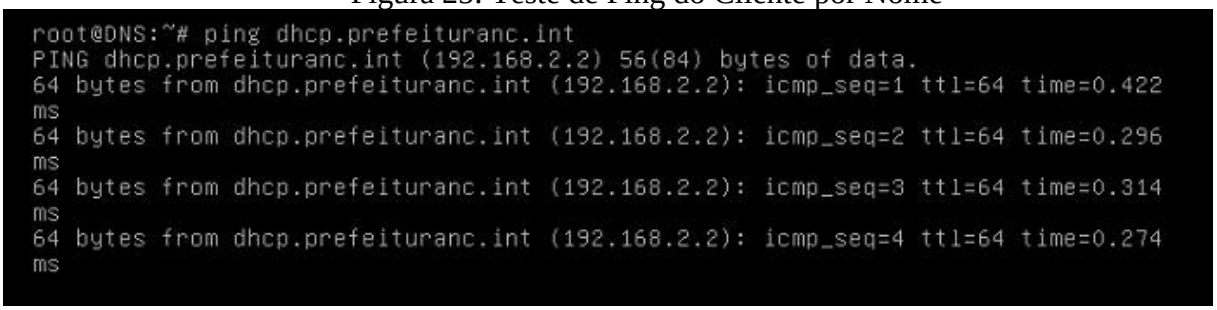
Fazendo uma busca pelo hostname da máquina, tem-se como resposta sua localização e o número do seu IP como é visto na Figura 23. A seguir foram feitos testes com os clientes servidores cadastrados no servidor DNS para analisar o funcionamento da zona, se a mesma está realizando buscas na rede seja via IP ou via string, usando o nome da máquina, visto nas Figuras 24,25 e 26.

Figura 24: Teste de Ping do Cliente por IP



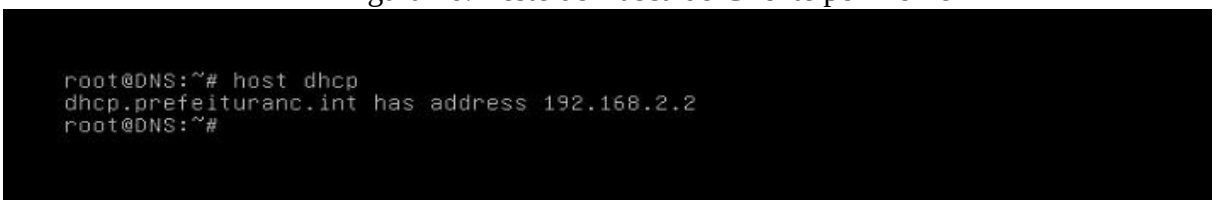
Fonte: Do Próprio Pesquisador

Figura 25: Teste de Ping do Cliente por Nome



Fonte: Do Próprio Pesquisador

Figura 26: Teste de Busca do Cliente por Nome



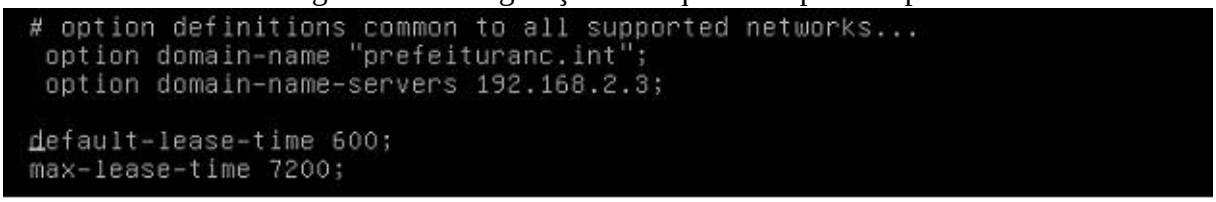
Fonte: Do Próprio Pesquisador

3.1.2 Servidor Dhcp

Por fim o servidor DHCP responsável por entregar os serviços disponibilizados pela rede usando um endereço IP.

Para que não surja redundância nas informações de criação da máquina virtual no KVM usando o Virt-Manager, não será demostrada a criação da mesma, uma vez que os passos são idênticos aos já mencionados na criação da máquina virtual DNS, atentando-se apenas aos detalhes de configuração do serviço.

Figura 27: Configuração do arquivo dhcpd.conf parte 1



Fonte: Do Próprio Pesquisador

Conforme visto na Figura 27, foi definido o nome do domínio para “prefeituranc.int” criado no servidor dns, logo baixo é especificado em qual endereço IP ele escutará na rede.

Em default-lease-time é definido o tempo máximo que um host cliente poderá fazer uso do IP fornecido a ele quando o cliente não solicitar um período maior, foi definido para 600 segundos que corresponde a 10 minutos.

Em max-lease-time é definido o tempo que o servidor poderá fornecer um endereço IP, mesmo que o cliente solicite por um período maior, 7200 segundos que corresponde a 2 horas.

Figura 28: Configuração do arquivo dhcpd.conf parte 2

```
subnet 192.168.2.0 netmask 255.255.255.0 {
        range 192.168.2.8 192.168.2.14;
        option routers 192.168.2.5;
        option domain-name-servers 192.168.2.3;
        option broadcast-address 192.168.2.255;
#       deny unknown-clients;//negar clientes que não estejam cadastrados por M$

        host DNS{
                hardware ethernet 52:54:00:40:26:80;
                fixed-address 192.168.2.3;
        }
        host SAMBA{
                hardware ethernet 52:54:00:80:4f:0f;
                fixed-address 192.168.2.4;
        }
```

Fonte: Do Próprio Pesquisador

No parâmetro subnet são definidas as informações que serão repassadas aos hosts como qual será a classe da rede e qual faixa de atuação juntamente com sua máscara de sub rede. Neste servidor foi usada uma rede classe C, pois a mesma é limitado ao intervalo de IP 192.168.0.0 a 223.255.255.255 ofertando a implementação de 254 hosts na rede, pois o primeiro IP é reservado para a rede e o último conforme será explicado é o endereço de broadcast.

Em range foi limitado de qual até qual IP a rede irá disponibilizar para seus clientes. Em seguida foi introduzido a rota padrão ou seja o gateway da rede ao qual o clientes irão usar para acessar a internet.

Em domain-name-servers é informado o IP do servidor DNS, em broadcast-address, como o próprio nome sugere, é o número do IP final da rede responsável por enviar os pacotes a todos os hosts pertencentes a faixa que foi usada. Para finalizar as regras, optou-se por cadastrar cada máquina servidor para receber o mesmo IP, de um sequencia fora dos IPs que serão ofertados na rede. No final das Figuras 20 e 21 é possível observar a implementação feita, usando o mac adress e o IP que cada host usará.

Figura 29: Configuração do arquivo dhcpd.conf parte 3

```
        host SQUID{
                hardware ethernet 52:54:00:74:51:23;
                fixed-address 192.168.2.5;
        }
        host OWNCLOUD{
                hardware ethernet 52:54:00:36:42:9f;
                fixed-address 192.168.2.6;
        }
        host NOTEBOOK{
                hardware ethernet dc:0e:a1:c5:55:b3;
                fixed-address 192.168.2.7;
        }
```

Fonte: Do Próprio Pesquisador

3.1.3 Servidor Squid

Trata-se do gateway da rede, o Squid tem o papel gerenciar as requisições dos hosts clientes seja para envio ou recebimento de dados na internet, armazenando cache dos sites com a finalidade de diminuir o tempo de carregamento das páginas.

Como de praxe, antes da instalação do software foi feita uma verificação de atualização nos repositórios e outra de atualização do sistema operacional com os comandos *apt-get update && apt-get upgrade*.

O software Squid foi baixado usando o comando *apt-get install Squid3*, posteriormente foi acessado o seu arquivo de configuração padrão que se localiza em /etc/squid3/squid.conf.

Antes de quaisquer regras de controles para uso da internet, a rede deve estar inserida em uma ACL – Acces Control List principal onde são criadas as políticas de uso da internet. Pesquisando pelo parâmetro *acl localnet src* no arquivo do Squid, o mesmo direciona o local apropriado para inserção da ACL.

Figura 30: Criação da Acl Src – principal da rede interna

```
  GNU nano 2.2.6        Arquivo: /etc/squid3/squid.conf

#Default:
# ACLs all, manager, localhost, and to_localhost are predefined.
#
#
# Recommended minimum configuration:
#

# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
#acl localnet src 10.0.0.0/8    # RFC1918 possible internal network
#acl localnet src 172.16.0.0/12 # RFC1918 possible internal network
#acl localnet src 192.168.0.0/16        # RFC1918 possible internal network
#acl localnet src fc00::/7       # RFC 4193 local private network range
#acl localnet src fe80::/10      # RFC 4291 link-local (directly plugged) machi$

acl rede_interna-prefeitura src 192.168.2.0/24
```

Fonte: Do Próprio Pesquisador

Posteriormente sua criação, deve ser autorizada o uso da mesma, pesquisando pelo parâmetro *http_access allow localnet* onde ficará liberado o trafego da máquina local e a rede interna.

Figura 31: Liberando Acl principal da rede interna

```
## TRAFEGO LIBERADO PARA A REDE INTERNA
http_access allow localhost
http_access allow rede_interna-prefeitura
```

Fonte: Do Próprio Pesquisador

Para que o servidor seja identificado pelos seus hosts clientes também foi inserido o seu hostname pesquisando pelo parâmetro *visible_hostname localhost* e alterado para *visible_hostname SQUID*. Um dos papeis de extrema importância que o Squid faz na rede é diminuir o tempo de consulta na internet, tarefa essa feita pelo armazenamento de cache do sites acessados, realizando uma coisa e mantendo disponível para uma futura solicitação atualizando apenas o necessário, através do parâmetro *cache_dir ufs*, onde será especificado o diretório de armazenamento dos cache.

Figura 32: Ativando o Cache no squid

Fonte: Do Próprio Pesquisador

- ufs – sistema de arquivos
- 1024 – tamanho em MB reservado para armazenamento de cache
- 10 – número de diretórios
- 100 – número de subdiretórios gerados por cada diretório.

Também é preciso dizer ao Squid qual será o tamanho máximo do arquivo cache que será armazenado.

Figura 33: Ativando Tamanho do Cache no Squid

Fonte: Do Próprio Pesquisador

Configurado para que funcione de forma transparente, ou seja, sem a necessidade que seja inserido um proxy no navegador “IP do servidor Squid” para acessar a internet, foi inserido o parâmetro *intercept* à frente da porta padrão de uso do Squid.

Figura 34: Ativando o Proxy Transparente no Squid

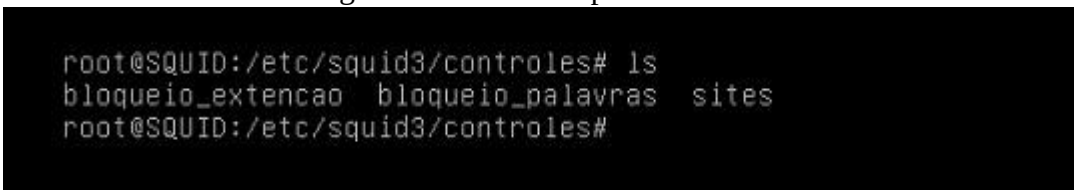
Fonte: Do Próprio Pesquisador

Feito isto, é preciso criar as regras de acesso para a rede interna através de ACLs como feito no início, porem no lugar de cadastrar a rede, será criado regras de uso para a

mesma. Nesta implementação o Squid conseguira bloquear apenas requisições *http*, sendo preciso posteriormente a implementação de um *Firewall* como forma de auxílio.

Para que seja possível a criação das ACLs, deve ser criado um ou mais diretórios para armazenamento dos conteúdos que serão proibidos na rede. Fazendo uso do comando *mkdir* seguido do caminho */etc/squid3* e o nome, conforme demonstrado na Figura 35, foi gerado o diretório controles e adicionado os arquivos om os conteúdos.

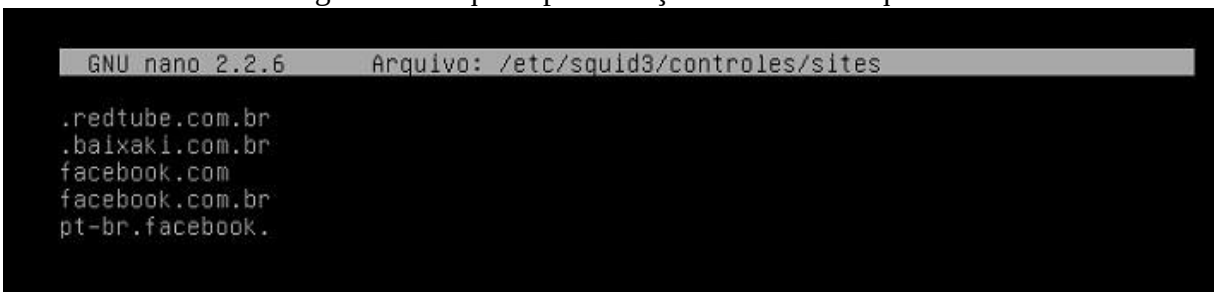
Figura 35: Diretórios para as ACLs



Fonte: Do Próprio Pesquisador

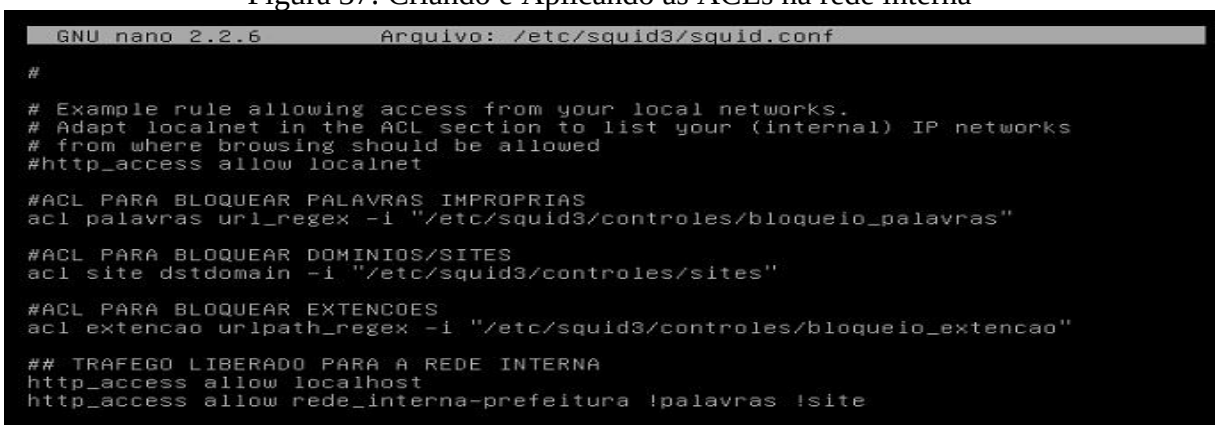
Como os três arquivos seguem a mesma lógica, sendo necessário apenas trocar os dados pelos que desejar, através do editor *nano* demonstrado abaixo o conteúdo dos sites.

Figura 36: Arquivo para criação de ACL no Squid



Fonte: Do Próprio Pesquisador

Figura 37: Criando e Aplicando as ACLs na rede interna



Fonte: Do Próprio Pesquisador

Na Figura 37 foram geradas três ACLs referentes aos arquivos criados anteriormente, a primeira usa o parâmetro *url_regex -i* que irá restringir a navegação a quaisquer sites que contenha palavras contidas no arquivo. A segunda tem o parâmetro *dstdomain –i* irá limitar o acesso aos sites contidos no arquivo. A terceira tem o parâmetro *urlpath_regex –i* que controla o uso das extensões contidas no arquivo.

E para validar todos os processos feitos foi usado o comando */etc/init.d/squid restart.* Conforme já mencionando o Squid configurado aqui realizará o bloqueio apenas do protocolo http, por isso surgiu a necessidade do firewall *NETFILTER*[10] do Linux usando a ferramenta *IPTABLES* que irá analisar as requisições. O Iptables é divido em tabelas, as usualmente são a *FILTER* que especifica as regras de entrada "INPUT", saída "OUTPUT" e passagem pelo servidor "FORWARD", a *NAT* usada para mascarar pacotes, a *MANGLE* para manipulação de IPs e a *RAW* que lida com dados do sistema operacional. Incorporado as tabelas existe as cadeias ou "chains", local onde é aplicado as regras criadas. Por padrão as cadeias possuem suas políticas de uso como ACCEPT, ou seja, tudo que não está bloqueado é permitido.

O Iptables funciona de forma estrutural, ou seja, irá executar em uma sequência cronológica suas regras, portanto é necessário repensar a lógica do conteúdo bloqueado, para que não surja transtornos na rede.

- ✓ INPUT – analisa tudo que entra no servidor
- ✓ OUTPUT – analisa tudo que sai do servidor
- ✓ FORWARD – analisa tudo que passa pelo servidor
- ✓ PREROUTING – redireciona pacotes para portas especificas
- ✓ POSTROUTING – mascara os pacotes ocultando e identificando o IP
- ✓ ACCEPT – deixa a requisição passar
- ✓ DROP – bloqueia a requisição e não emite aviso ao solicitante
- ✓ REDIRECT –transfere a requisição para uma porta especificada
- ✓ MASQUERADE – faz o mascaramento dos pacotes
- ✓ -F: apaga todas as regras do firewall
- ✓ -X: apaga uma regra específica
- ✓ -t: qual tabela a ser usada
- ✓ -A: insere uma nova regra
- ✓ -s: destino de origem

[10] http://www.planetaunix.com.br/2014/12/firewall-iptables-parte-01.html

- ✓ -d: destino final
- ✓ -j: solicita uma ação a ser aplicada
- ✓ -to-port: porta que será usada
- ✓ -i interface de rede a ser usada
- ✓ -p: solicita um protocolo
- ✓ --dport – porta de destino
- ✓ Limit – limita a quantidade de tentativas para execução da regra
- ✓ State – define um estado para a conexão
- ✓ Icmp – Internet Control Message Protocol
- ✓ -syn – permite o uso dos bits ACK E FIN
- ✓ --tcp-flags – analisa flags do protocolo TCP
- ✓ NEW – pacote que usa uma nova conexão
- ✓ ESTABLISHED – pacotes que tem conexão existente
- ✓ RELATED – pacote que não está na conexão existente
- ✓ INVALID – pacote desconhecido ou não identificado

Abaixo arquivo do script do firewall criando no diretório /etc/init.d de nome meu_firewall.sh usando o editor nano: nano /etc/init.d/meu_firewall.sh

```
#! /bin/bash
###LIMPANDO AS REGRAS
iptables -F
iptables -x
iptables -t nat -F
iptables -t nat –X
####ALTERANDO AS POLITICAS DAS CADEIAS
iptables –P INPUT ACCEPT
iptables –P OUTPUT ACCEPT
iptables –P FORWARD ACCEPT
####LIBERANDO A PORTA HTTPS
iptables -A FORWARD -s 192.168.2.0/24 -p tcp --dport 443 -j ACCEPT
iptables -A FORWARD -s 192.168.2.0/24 -p udp --dport 443 -j ACCEPT
###REDIRECIONANDO O TRAFEGO PARA O SQUID
iptables -t nat -A PREROUTING -s 192.168.2.0/24 -p tcp --dport 80 -j REDIRECT --to-port
3128
iptables -t nat -A PREROUTING -s 192.168.2.0/24 -p udp --dport 80 -j REDIRECT --to-port
3128
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
#####CONTROLANDO A QUALIDADE DAS CONEXOES PARA BLOQUEAR
SCANNERS
```

```
iptables –A INPUT –p tcp –tcp-flags SYN,ACK,FIN,RST RST –m limit –limit 1/s –j
ACCEPT
iptables –A FORWARD –p tcp –tcp-flags SYN,ACK,FIN,RST RST –m limit –limit 1/s –j
ACCEPT
####EXCLUINDO PACOTES SUSPEITOS
iptables –A INPUT –m state –state INVALID –j DROP
iptables –A INPUT –p tcp ! –syn –m state –state NEW –j DROP
iptables –A FORWARD –m state –state INVALID –j DROP
iptables –A FORWARD –p tcp ! –syn –m state –state NEW –j DROP
#####SYN-FLOOD
iptables –A FORWARD –p tcp –syn –m limit –limit 1/s –j ACCEPT
###CONFIRMA CONEXOES JA ESTABELECIDAS
iptables –A FORWARD –m state –state RELATED,ESTABISHED –j ACCEPT
#####################################################################
##########BLOQUEADO/LIBERANDO O FACEBOOK
######IP DA REDE INTERNA LIBERADO#################################
iptables -A FORWARD -i eth1 -s 192.168.2.7 -m string --algo bm --string "facebook.com" -j
ACCEPT
iptables -A FORWARD -i eth0 -d 192.168.2.7 -m string --algo bm --string "facebook.com" -j
ACCEPT
######DEMAIS IPS DA REDE INTERNA BLOQUEADOS#####################
iptables -A FORWARD -i eth1 -m string --algo bm --string "facebook.com" -j DROP
iptables -A FORWARD -i eth0 -d 192.168.2.7 -m string --algo bm --string "facebook.com" -j
DROP
```

Após sua criação foi necessário dar permissão de execução ao arquivo com o comando *chmod +x /etc/init.d/meu_firewall.sh* para que as regras fossem aplicadas na rede e */etc/init.d/meu_firewall.sh* para iniciar o script.

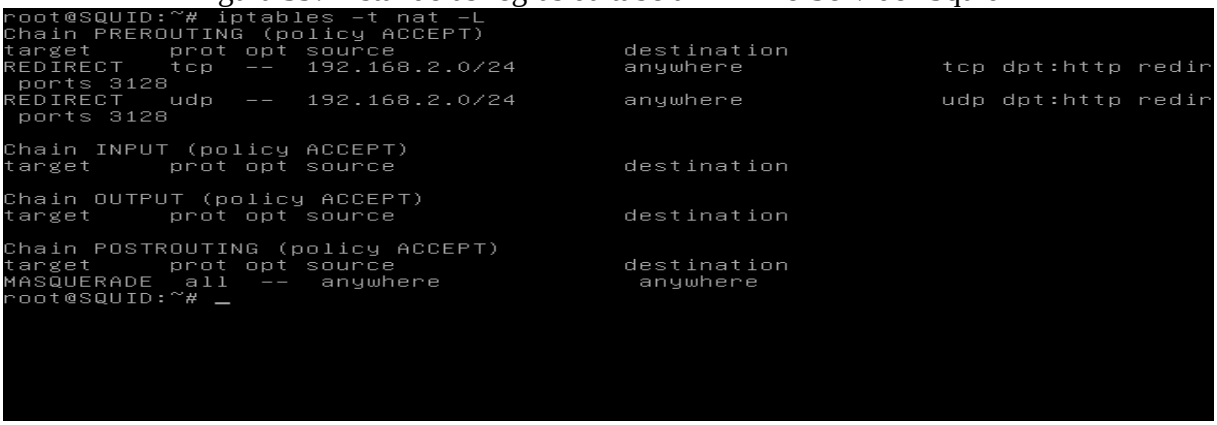
Listando as regras com o comando *iptables –L* e *iptables –t nat –L* é possível a execução das mesmas em cada cadeia, conforme demonstrado nas Figuras 38 e 39.

Figura 38: Listando as regras Iptables do Servidor Squid

```
DROP       all  --  anywhere             anywhere             state INVALID
DROP       tcp  --  anywhere             anywhere             tcp flags:!FIN,SYN
,RST,ACK/SYN state NEW
DROP       tcp  --  anywhere             anywhere             tcp flags:!FIN,SYN
,RST,ACK/SYN state NEW
ACCEPT     tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,
RST,ACK/SYN limit: avg 1/sec burst 5
ACCEPT     tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,
RST,ACK/SYN limit: avg 1/sec burst 5
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTA
BLISHED
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTA
BLISHED
ACCEPT     all  --  192.168.2.7          anywhere             STRING match  "fac
ebook.com" ALGO name bm TO 65535
ACCEPT     all  --  anywhere             192.168.2.7          STRING match  "fac
ebook.com" ALGO name bm TO 65535
DROP       all  --  anywhere             anywhere             STRING match  "fac
ebook.com" ALGO name bm TO 65535
DROP       all  --  anywhere             192.168.2.0/24       STRING match  "fac
ebook.com" ALGO name bm TO 65535

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@SQUID:~#
```

Fonte: Do Próprio Pesquisador

Figura 39: Listando as regras da tabela NAT no Servidor Squid



Fonte: Do Próprio Pesquisador

3.1.4 Servidor Samba

Criado para funcionar como o controlador do domínio, cabe a ele cadastrar cada computador com um usuário e senha para realizar as políticas de uso local de cada máquina. Para que exista o compartilhamento de arquivos no domínio, foi inserido dois HDs no servidor, o primeiro particionado para instalação do sistema operacional com uma partição primária usando o formato *swap* para armazenamento de memória virtual e uma segunda partição primária com formato de arquivo *XFS*[11] para o sistema operacional, o segundo HD será explicado o seu uso posteriormente uma vez que o mesmo será usado após a instalação do Samba.

Feitas as devidas configurações nas interfaces de redes, conforme já demonstrado anteriormente, foi acessado o diretório *cd /root* e com o comando *wget* usado para realizar download de arquivos da internet, foi baixado o arquivo Samba4 na extensão *.tar.gz* através do site https://download.samba.org/pub/samba/stable. Terminado foi necessário descompacta-lo usando o comando *tar –zxvf nomedoarquivo* para que pudesse ser compilador e instalado.

Para que o Samba funcione como um controlador de domínio é necessário que haja um amontoado de bibliotecas instaladas que permitirá a sua configuração, as mesmas citadas abaixo:

apt-get install build-essential libacl1-dev libattr1-dev libblkid-dev libgnutls28-dev libreadline-dev python-dev python-dnspython gdb pkg-config libpopt-dev libldap2-dev

[11] https://www.vivaolinux.com.br/artigo/Sistemas-de-arquivos-para-GNU-Linux?pagina=5

dnsutils libbsd-dev attr krb5-user docbook-xsl libcups2-dev libkrb5-dev libssl-dev python-software-properties acl quota

Com todas as dependências instaladas foi acessado a pasta *cd /root/sama-4* e executa o comando *./configure* para que fosse compilado o arquivo e posteriormente *make e make install* para a instalação do mesmo.

Fugindo do método de instalação "padrão", onde o Samba é instalado no diretório */etc* optou-se por instala-lo no diretório /usr/local[12] uma vez que ele foi compilado e instalado manualmente.

Para que o mesmo funcione como um pdc, ou seja, um controlador de domínio e compartilhe arquivos na rede, foi necessário usar o comando */usr/local/samba/bin/samba-tool domain provision –use-rfc2307 –use-xattrs=yes –interactive*, onde foi solicitado as diretivas do domínio a ser criado.

Realm: PREFEITURANC.INT – nome do domínio
Domain [PREFEITURANC]: - nome NetBios
Server Role (dc, member, standalone) [dc]: - definindo como controlador de domínio
DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE)
[SAMBA_INTERNAL]: - servidor para resolver nomes
DNS forwarder IP address (write 'none' to disable forwarding) [192.168.0.1]:
192.168.2.5//ip do servidor SQUID – servidor externo para resolver nomes
Administrator password: senha root do administrador
Retype password:

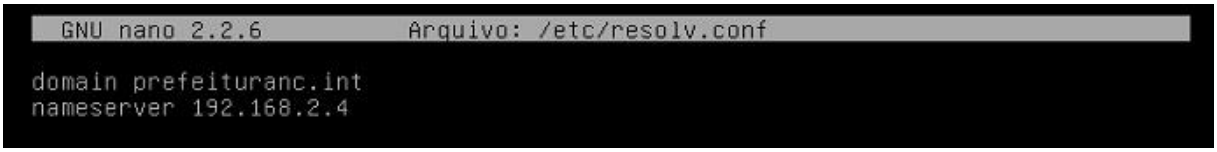
Ao final do processamento das informações prestadas, o samba gerou o domínio propriamente dito com os dados que serão usados pelas máquinas que fizer uso da rede e forem adicionadas a ele.

Figura 40: Domínio Gerado no Samba

Fonte: Do Próprio Pesquisador

Para testar a criação do domínio, foi alterado o arquivo *resolv.conf* da seguinte maneira:

Figura 41: Resolv.conf do Samba



Fonte: Do Próprio Pesquisador

[12] http://www.hardware.com.br/dicas/diretorios-linux.html

Inserindo o domínio criado tanto para prover o serviço quanto para ser localizado pelos hosts da rede, e denominando-o como o servidor de nomes principal, uma vez que ele está usando o seu próprio dns local para tradução de nomes.

Em seguida ativado o daemon Samba usando o comando */usr/local/sbin/samba*, e posteriormente para saber se se as configurações foram efetivas, foi usado o comando */usr/local/bin/smbclient –version.* Feito isto, foi realizada uma verificação nos diretórios padrões criados durante a instalação do Samba, com os parâmetros */usr/local/samba/bin/smbclient –L localhost –U%* conforme demostrado na Figura 42 para a pasta netlogn conforme demonstrado na Figura 43, apenas como precaução e ou confirmação das alterações feitas.

Figura 42: Verificação das pastas no Samba

```
Domain=[PREFEITURANC] OS=[Unix] Server=[Samba 4.1.16]

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
```

Fonte: Do Próprio Pesquisador

Figura 43: Verificando a pasta Netlogon do Samba

```
root@SAMBA:~# /usr/local/samba/bin/smbclient //localhost/netlogon -Uadministrato
r%'ch@v8s_pnc2016' -c 'ls'
Domain=[PREFEITURANC] OS=[Unix] Server=[Samba 4.1.16]
  .                                   D        0  Sun Oct 16 17:39:46 2016
  ..                                  D        0  Sat Sep 24 21:09:33 2016
  rh-01.bat                           A      249  Sun Oct 16 17:39:13 2016
  rh-02.bat                           A      249  Sun Oct 16 17:39:59 2016

                35974 blocks of size 4194304. 35508 blocks available
```

Fonte: Do Próprio Pesquisador

E posteriormente testado a resolução de nomes via protocolos, usando o tcp, udp e o próprio domínio, conforme demonstrado nas Figuras 44,45 e 46.

Figura 44: Verificando o Funcionamento do Protocolo TCP

```
root@SAMBA:~# host -t SRV _ldap._tcp.prefeituranc.int
_ldap._tcp.prefeituranc.int has SRV record 0 100 389 samba.prefeituranc.int.
root@SAMBA:~# _
```

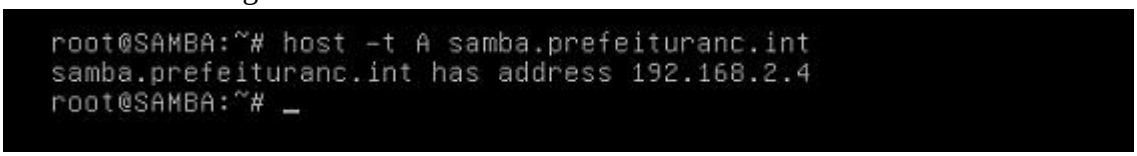
Fonte: Do Próprio Pesquisador

Figura 45: Verificando o Funcionamento do Protocolo UDP

```
root@SAMBA:~# host -t SRV _kerberos._udp.prefeituranc.int
_kerberos._udp.prefeituranc.int has SRV record 0 100 88 samba.prefeituranc.int.
root@SAMBA:~#
```

Fonte: Do Próprio Pesquisador

Figura 46: Verificando o Funcionamento do Domínio



Fonte: Do Próprio Pesquisador

Terminando isto, foi verificado o arquivo de configuração do protocolo Kerberos[13], que irá supervisionar as senhas dos usuários conectados ao domínio, o mesmo deve conter o realm "domínio" gerado para que os usuários sejam autenticados na rede.

Figura 47: Arquivo de configuração do Kerberos

Fonte: Do Próprio Pesquisador

O samba em sua instalação armazena a senha do administrador por um período irrisório como medida de segurança, forçando-o a renová-la de tempos em tempos. Usado apenas para testar esse recurso, o comando */usr/local/samba/bin/samba-tool user setpassword administrator* permite a criação de uma nova senha. Caso deseje verificar a duração da mesma, o comando kinit acompanhado do usuário administrador e domínio, oferta a exibição em dias da duração da mesma, foram inseridos os seguintes parâmetros: *kinit administrator@prefeituranc.int*, o mesmo retornou o tempo de vida da senha correspondentes a 41 dias.

Antes de iniciar o gerenciamento do samba, foi necessário que o segundo *HD* instalado estivesse em uso, para que fosse criado os diretórios que cada departamento irá usar, com limitação a apenas o usuários pertencentes ele. Fazendo uso do comando *fdisk /dev/sdb1* que é a partição do segundo HD que ainda não está em uso, o software *fdisk* proporcionou um menu com as opções para que fosse gerada a tabela de partições. Uma vez dentro do mesmo foi escolhida a opção n conforme descrito a seguir:

Comando (m para ajuda): n

Partition type:

p primary (0 primary, 0 extended, 4 free) e estendido

select (default p): p

[13] http://www.gta.ufrj.br/grad/99_2/marcos/kerberos.htm

número da partição (1-4, padrão 1): 1

primeiro setor (2048-10485759, padrão 2048): 2048

Last setor, +setores or +size{K,M,G} (2048-10485759, padrão 10485759):

Usando valor padrão 10485759

Foi necessário gravar a tabela no disco, escolhendo a opção w, em seguida formatado com o comando mkfs.ext4 /dev/sdb1, e montado na inicialização inserido os parâmetros *user_xattr,ac,barrier* para que o samba administre os fluxo de usuários e senhas que serão cadastrados, conforme citado a baixo:

nano /etc/fstab

/dev/sdb1 /mnt/samba ext4 user_xattr,acl,barrier=1 1 1

Para finalizar, foi criado o diretório principal em *cd /mnt* com o comando *mkdir* de nome *samba* e dentro do mesmo, os diretórios que cada departamento irá usar na rede. Em cada sub diretório foi dado permissão de uso *chmod –R 777*, ou seja, controle total ao proprietário, grupo e usuários que pertencem ao departamento.

Figura 48 Diretórios para uso dos departamentos

```
root@SAMBA:/mnt# cd samba/
root@SAMBA:/mnt/samba# ls
COMPRAS    FINANCAS  JURIDICO   lost+found  RECEPCAO  TRIBUTOS
CONVENIOS  GABINETE  LICITACAO  Owncloud    RH
root@SAMBA:/mnt/samba#
```

Fonte: Do Próprio Pesquisador

Como forma de auxilio gráfico e facilitando as inúmeras tarefas necessárias no gerenciamento do samba, foi usado o software da *Microsoft RSAT*[14] que proverá a recursos administrativos para inserção dos computadores ao samba, como facilidade no seu gerenciamento para criação dos usuários, grupos, políticas de uso no domínio, etc. Feita a instalação dos softwares na máquina cliente, foi necessário ativar os recursos administrativos que os mesmos dispõem, acessando o *painel de controle, programas, ativar recursos do Windows*, como medida de precaução foram marcadas todas as opções do menu *ferramentas de Administração de Servidor Remoto* e seus sub menus. Conforme já visto o servidor *DHCP* foi criado de forma a prover IPs de forma dinâmica, ou seja, automática sem a necessidade de quaisquer alterações por parte do usuário ao se conectar. Com o usuário administrador, foi alterado na máquina apenas o endereço *DNS* primário para o IP do servidor samba e o DNS secundário para o IP do próprio servidor DNS da rede interna, de forma que o host cliente terá

[14] https://www.microsoft.com/pt-br/download/details.aspx?id=7887

duas opções de resolução de nomes na rede, se uma falhar automaticamente a outra assume o seu lugar, evitando a negação de serviço na rede.

Devendo os hosts clientes serem adicionados no domínio, foram seguidos os seguintes passos a saber:

- Menu iniciar, meu computador
- Clicado com o botão direito em propriedades
- Alterar configurações
- Id de rede e selecionando a opção “este computador faz parte de uma rede corporativa...”
- Minha empresa usa uma rede com um domínio

Seguinte essas etapas será solicitado que informa o usuário administrador do sistema juntamente com sua senha e o nome do domínio criado e por fim a definição da conta a ser criado, seja para o próprio administrador ou para um usuário da rede, por fim será solicitado que reinicie o host para validar as configurações. Feitas essas alterações o Windows mudará sua forma logon sendo preciso usar a combinação das teclas *ctrl+alt+delete* conforme visto na Figura 48 e 49, e inserir o usuário cadastrado e senha cadastrada.

Figura 49: Tela de Login do Windows 7

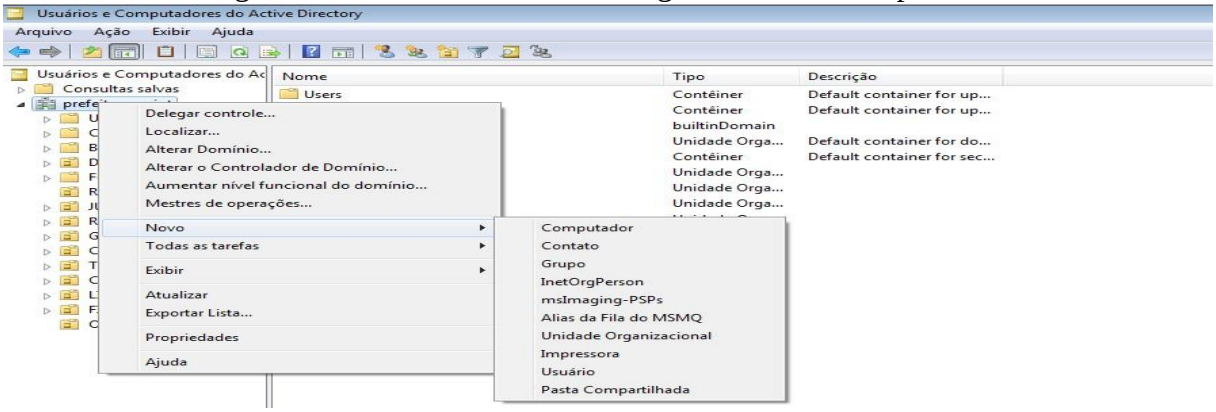
Fonte: Do Próprio Pesquisador

Figura 50: Acessando o Domínio com usuário administrador

Fonte: Do Próprio Pesquisador

De forma a organizar o domínio, foi criado uma unidade organizacional para cada setor, clicando sobre sua raiz “domínio” em cada uma um grupo. Para realizar essas tarefas foi usado o Windows com usuário administrativo, posteriormente acessado o painel de controle e o menu *ferramentas administrativas* dentro do mesmo o tópico *usuários e computadores do active directory*.

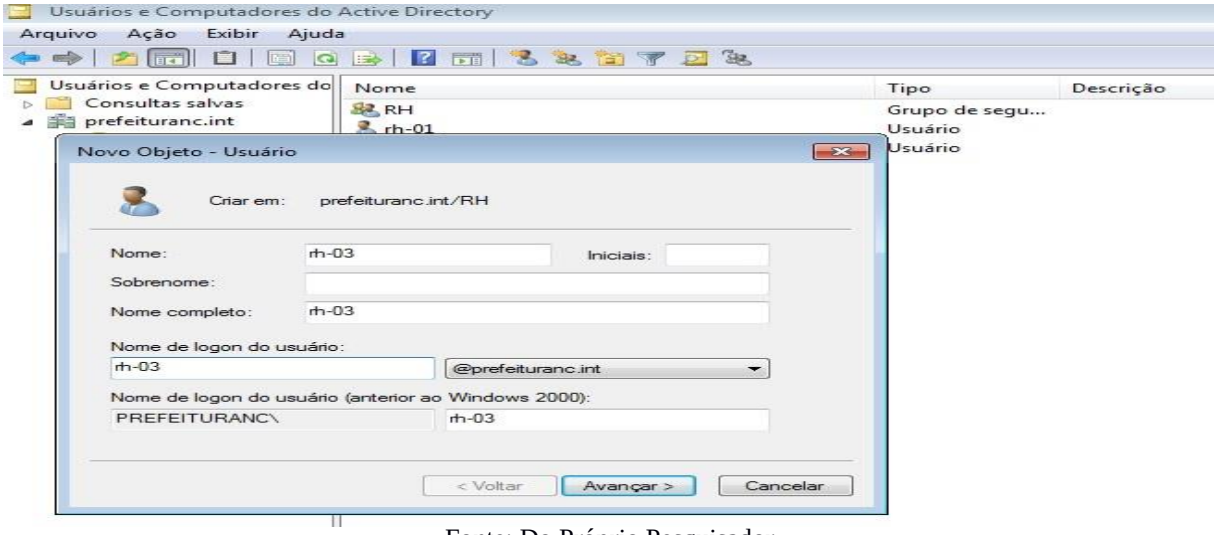
Figura 51: Criando as Unidades Organizacionais, Grupos e Usuários



Fonte: Do Próprio Pesquisador

A criação dos usuários nos grupos requer uma ordem cronológica, botão direto, novo, usuário, devido a necessidade dos mesmos serem inseridos nos seus grupos para que possam ser autenticados no domínio quando realizarem logon, após o preenchimento dos dados pessoais o assistente propicia algumas medidas de segurança como a troca de senha ou não, a mesma requerendo um grau de segurança para proteção dos dados

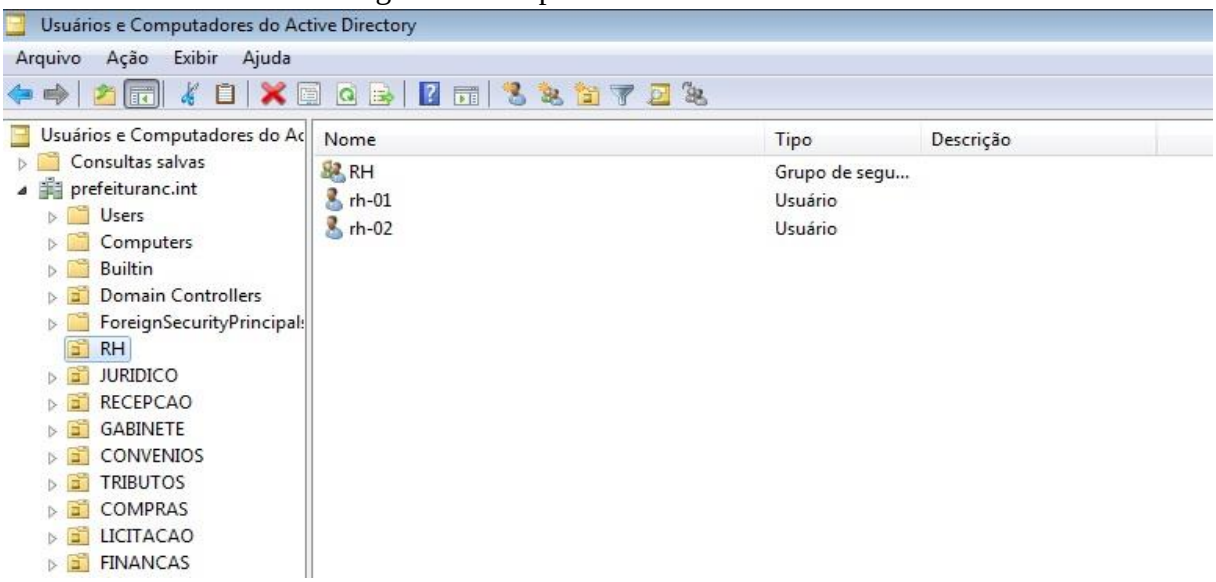
Figura 52: Criando um Usuário no Domínio



Fonte: Do Próprio Pesquisador

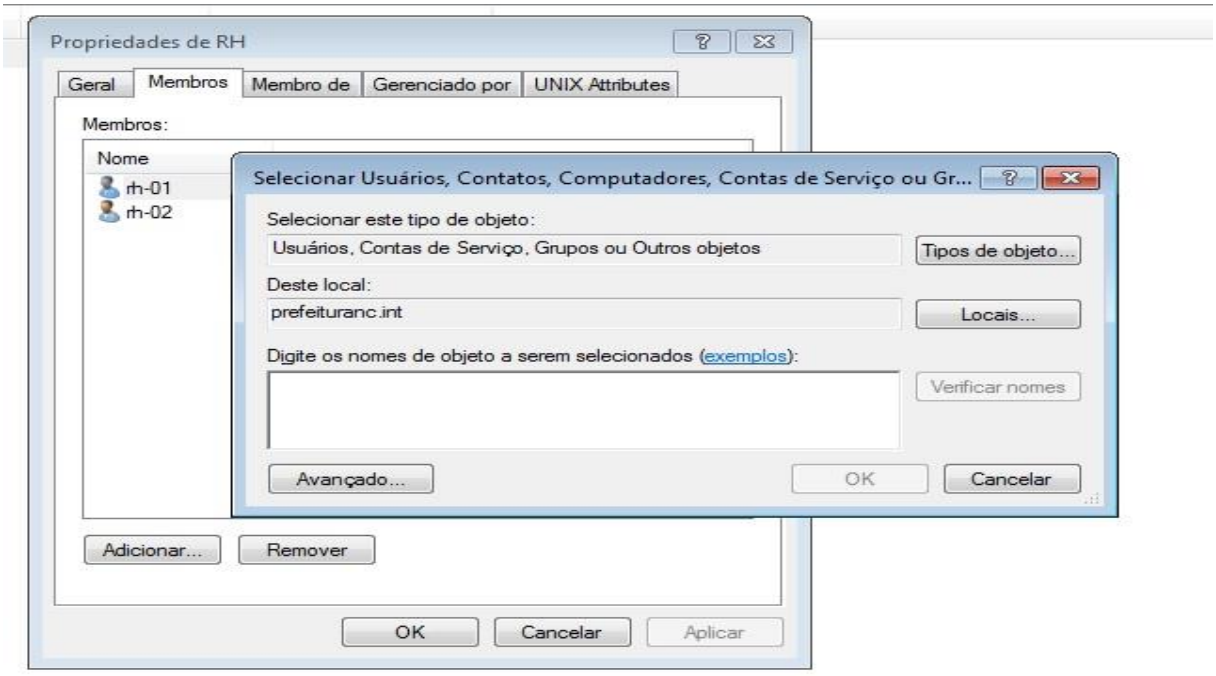
Após concluir as informações e criar uma senha de grau aceitável o assistente irá concluir a inserção do usuário ao grupo, sendo preciso inseri-lo ao grupo e inserir o grupo ao mesmo.

Figura 53: Grupo com seus Usuários Inseridos



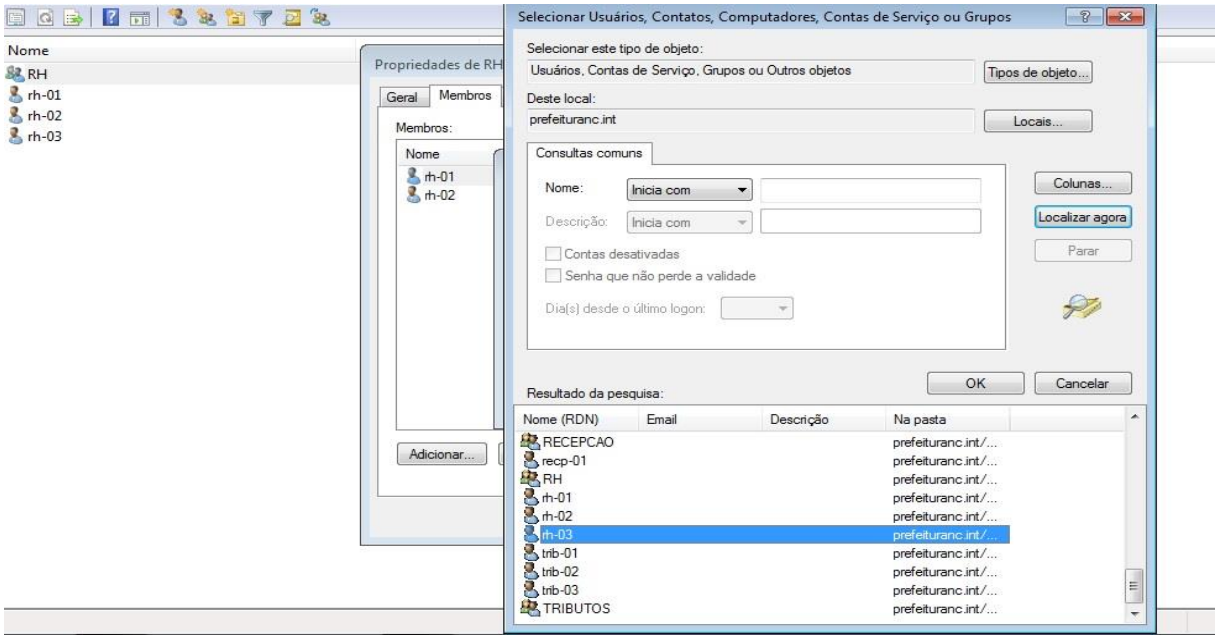
Fonte: Do Próprio Pesquisador

Figura 54: Buscando Usuário do Grupo



Fonte: Do Próprio Pesquisador

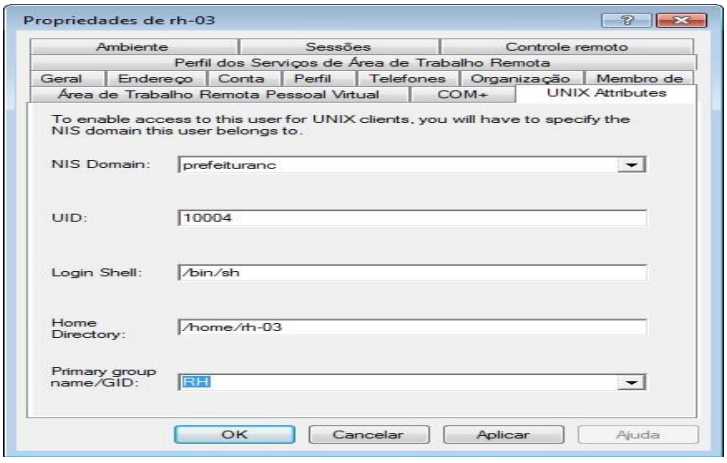
Figura 55: Adicionando Usuário ao Grupo



Fonte: Do Próprio Pesquisador

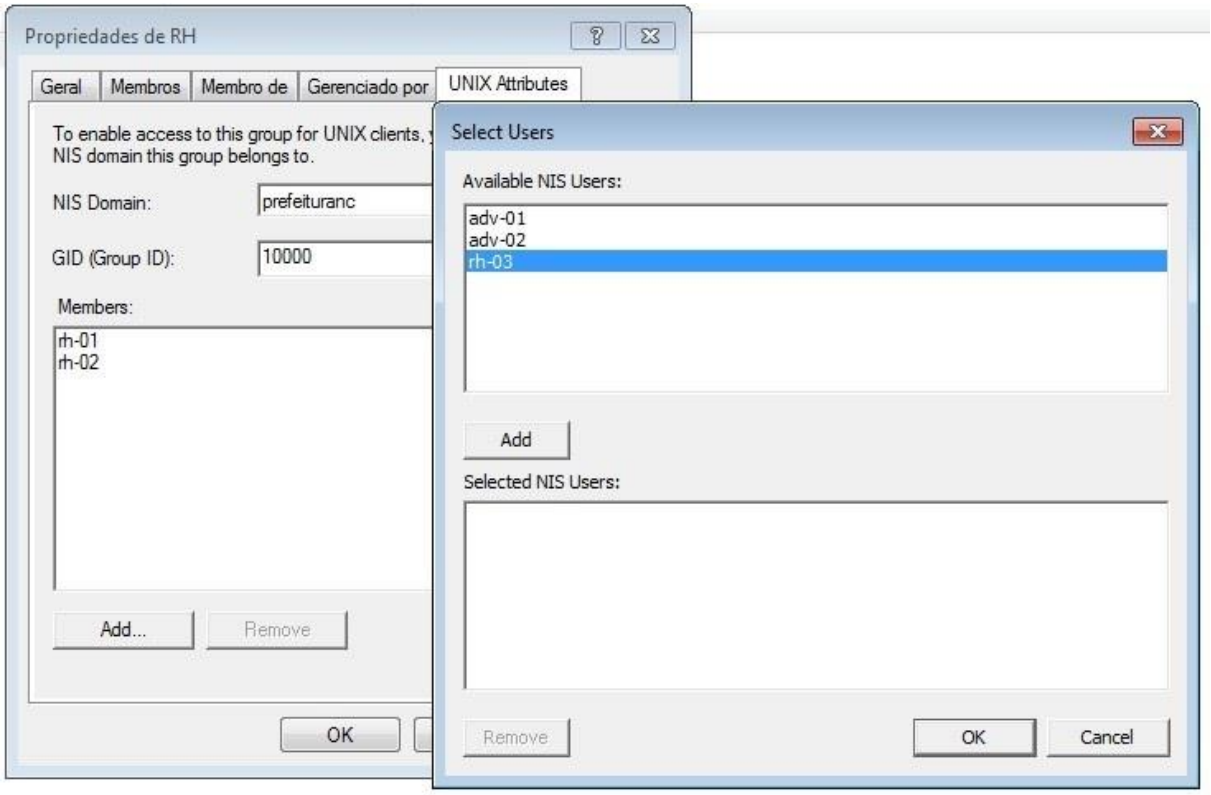
Conforme as Figuras 54 e 55, após a criação o usuário o mesmo foi transformado em um membro do grupo ao qual está inserido, seguindo os passos, a saber: propriedades do grupo, membros, avançado, em seguida localizar agora e feito uma busca pelo nome do mesmo. Foi necessário também vincula-lo ao grupo usando o atributo *UNIX Attibutes*, selecionando as informações do domínio conforme demonstrado abaixo, para que ele ingressasse no grupo, posteriormente repetir o mesmo passo para adiciona-lo como um novo usuário do grupo.

Figura 56: Vinculando o Grupo ao Usuário



Fonte: Do Próprio Pesquisador

Figura 57: Vinculando o Usuário ao Grupo

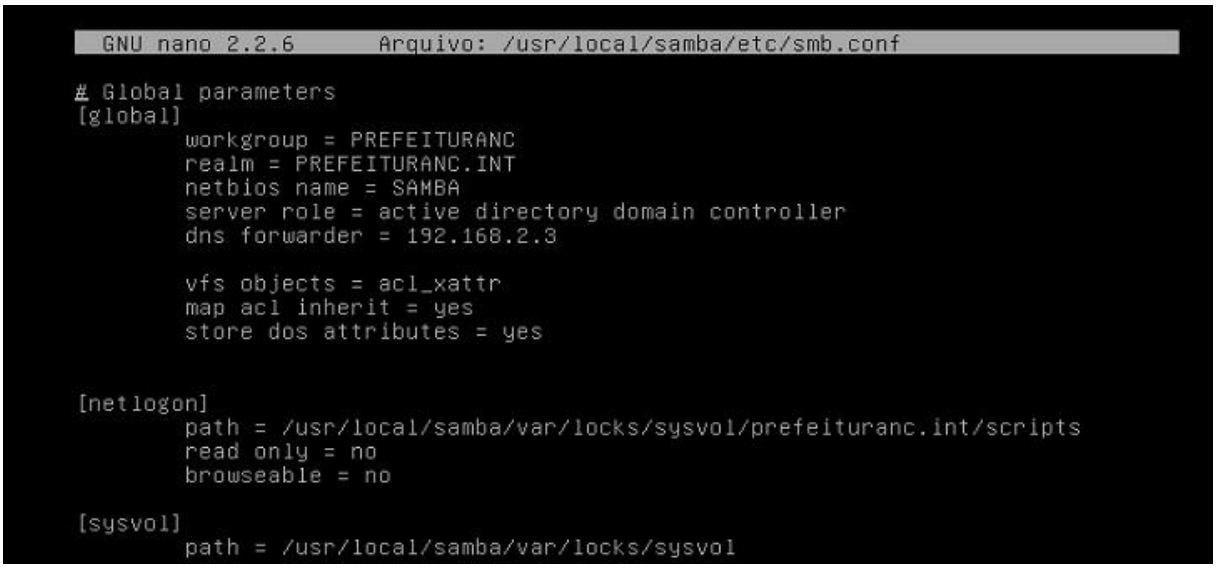


Fonte: Do Próprio Pesquisador

Todo esse e demais processos usando o *RSAT* pode ser feito de maneira menos intuitiva pelo terminal, fazendo uso da sintaxe */usr/local/samba/bin/samba-tool group addmembers nomedogrupo nomedousuário***,** ação essa ofertada pelo samba-tool, um conjunto de ferramentas do samba para execução de comando via texto para administra-lo de maneira mais direta. Concluídas as operações mencionadas, foi necessário ativar o compartilhamento dos discos rígidos do servidor, os quais estão instalados o daemon samba e o segundo onde foi criado os diretórios contendo os grupos e usuários, fazendo uso do parâmetro */usr/local/samba/bin/ /net rpc rights grant 'PREFEITURANC.INT\Domain Admins' SetDiskOperatorPrivilege –U'PREFEITURANC\administrator',* em seguida a configuração do *smb.conf* para que o samba disponibilizasse os conteúdos na rede. por padrão na sua instalação o samba gera no arquivo smb.conf o parâmetro global, netlogon e sysvol, no primeiro é setado dados como nome do dominio, nome do servidor, tipo e dns para consulta externo, o segundo entre outras funções é usado para guarda scripts de perfis do usuários, o terceiro para armazenar as regras *GPO*[15] dos usuários.

[15] https://wiki.samba.org/index.php/GSOC_GPO

Figura 58: Smb.conf



Fonte: Do Próprio Pesquisador

Devido à importância e necessidade do arquivo smb.conf para o funcionamento do samba foram inseridas diretivas de usos nos diretórios compartilhados usando os seguintes parâmetros a saber:

[global]

workgroup = PREFEITURANC
realm = PREFEITURANC.INT
netbios name = SAMBA
server role = active directory domain controller
dns forwarder = 192.168.2.3

vfs objects = acl_xattr
map acl inherit = yes
store dos attributes = yes

[netlogon]

path = /usr/local/samba/var/locks/sysvol/prefeituranc.int/scripts
read only = No
browseable = no

[sysvol]

path = /usr/local/samba/var/locks/sysvol
read only = No
browseable = no

[RH]

path=/mnt/samba/RH
read only = no
valid users = @RH

[JURIDICO]

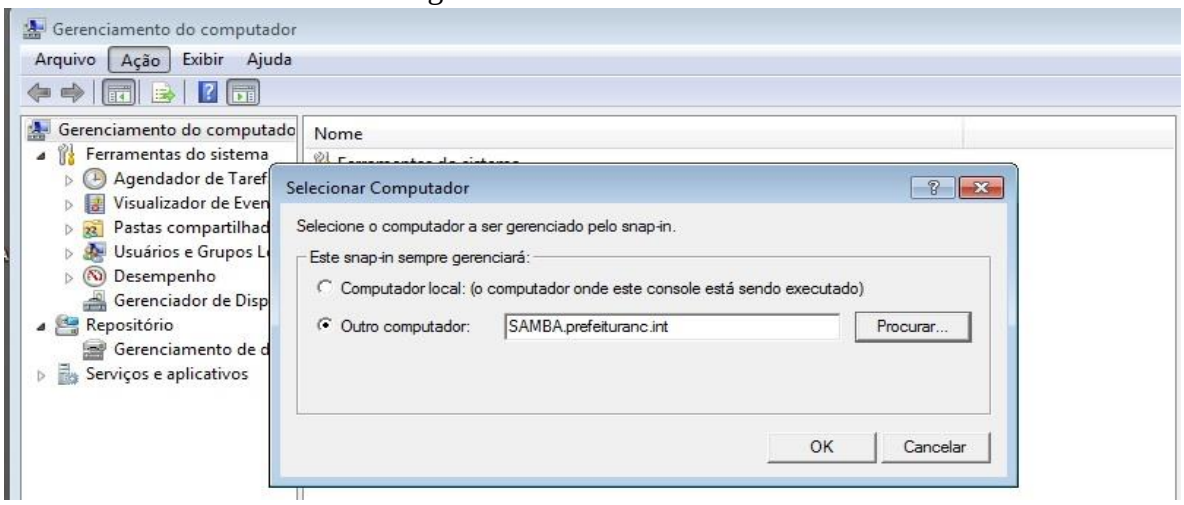
```
        path=/mnt/samba/JURIDICO
        read only = no
        valid users = @JURIDICO

[RECEPCAO]

        path=/mnt/samba/RECEPCAO
        read only = no
        valid users = @RECEPCAO

[GABINETE]

        path=/mnt/samba/GABINETE
        read only = no
        valid users = @GABINETE

[CONVENIOS]

        path=/mnt/samba/CONVENIOS
        read only = no
        valid users = @CONVENIOS

[TRIBUTOS]

        path=/mnt/samba/TRIBUTOS
        read only = no
        valid users = @TRIBUTOS

[COMPRAS]

        path=/mnt/samba/COMPRAS
        read only = no
        valid users = @COMPRAS

[LICITACAO]

        path=/mnt/samba/LICITACAO
        read only = no
        valid users = @LICITACAO
```

- ✓ Workgroup – determina o nome do domínio
- ✓ Realm – do domínio criado
- ✓ Netbios – nome do host
- ✓ Server role – tipo de domínio criado
- ✓ Path – caminho usado para compartilhar o conteúdo
- ✓ Browseable – se definido como “no” o conteúdo não será visível na rede se “yes” fica exposto a todos.
- ✓ Read only no – somente leitura, não.
- ✓ Valid users @- determina que somente o grupo terá permissão de uso.
- ✓ Vfs objects = acl_xattr – usado para mapear o Windows na rede
- ✓ Map acl inherit = yes – permite que o Windows acesse o samba via Windows explorer

✓ Store dos atributes = yes – para carregamento do compartilhamento de modo automático.

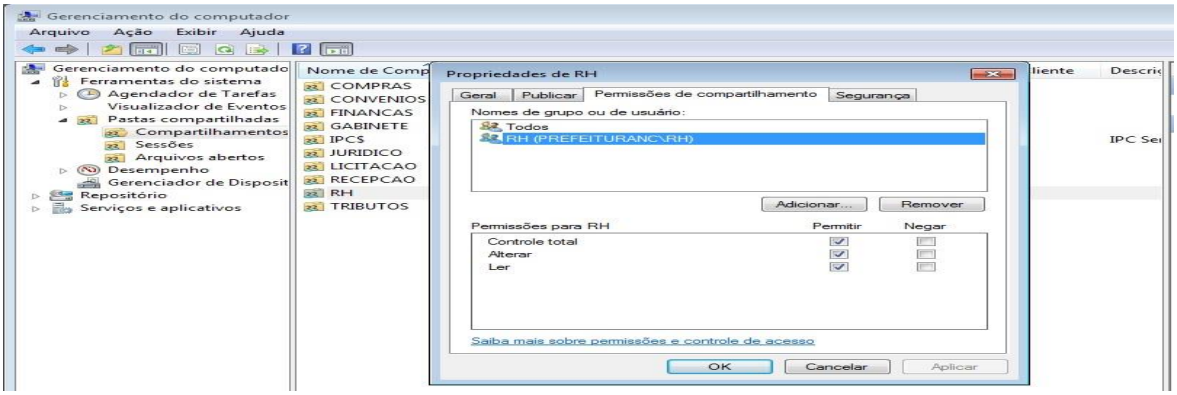
Para que os usuários consigam usar a pasta atribuída ao grupo que pertencem é preciso adiciona-los com privilégios, ou seja, permissão de uso da mesma, para que possam executar quaisquer tarefas como salvar, excluir, editar, ler dos dados contidos na mesma. Para que isso foi feito o gerenciamento das pastas contidas no disco rígido do samba clicando em *menu iniciar*, *meu computador botão direito e gerenciar* em seguida *ação*, autorizando nas suas permissões de compartilhamento os respectivos usuários que teriam privilégios para usa-la.

Figura 59 Conectando ao Samba



Fonte: Do Próprio Pesquisador

Figura 60 Dando Permissão de acesso na pasta ao grupo



Fonte: Do Próprio Pesquisador

Não foi administrado o uso de GPO[16], uma vez que a forma de adição dos computadores ao domínio exige que tenha uma conta administrativa oculta nos computadores, evitando que os usuários padrões consigam alterar as configurações dos mesmos. Nos testes feitos para implementação de GPO com o sistema operacional Windows 7, não foi obtido êxito algum.

3.1.5 Servidor Owncloud

Máquina servidor que fará o Backup dos dados de cada usuário, o Owncloud tem seu funcionamento de forma sincronizada, ou seja, os dados enviados de um computador desktop via rede poderá ser acessado de qualquer ponto da mesma usando diferentes equipamentos como um celular, um notebook, uma vez que ele funciona via navegador. O mesmo oferece recursos como edição, compartilhamento, backup dentre outros.

Para que essas tarefas sejam cumpridas com precisão, o owncloud faz uso de uma base de dados, que aqui foi usado o Mysql.

Figura 61: Inserido o Repositório para download do Owncloud

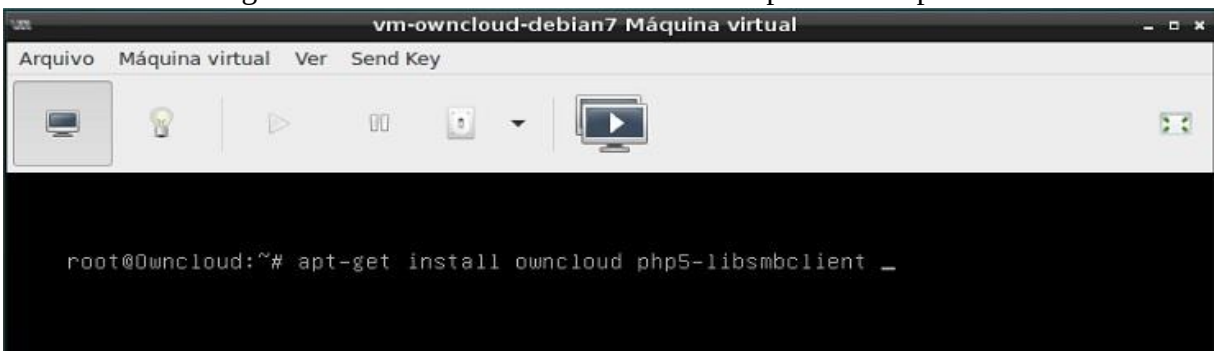
Fonte: Do Próprio Pesquisador

Fugindo do padrão da versão Debian usada em todas máquinas virtuais, o owncloud foi instalado no Debian 7, pois em testes feitos na versão atual da distribuição foram obtidos erros, impossibilitando o êxito nas suas configurações.

Na Figura 61, é demostrado a o uso do comando *echo* usado para fazer uma cópia do diretório opensuse para dentro do arquivo */etc/apt/sources.list.d/owncloud.list* possibilitando o download do software, uma vez que o Debian não possui em seus repositórios oficiais, nenhum do opensuse.

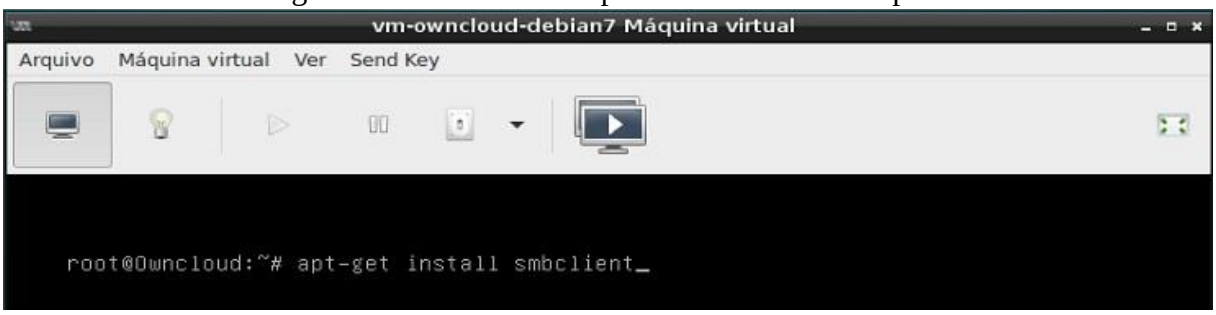
[16] https://www.microsoft.com/en-us/download/details.aspx?id=6243

Figura 62: Baixando o Owncloud e suas dependências parte 1



Fonte: Do Próprio Pesquisador

Figura 63: Baixando as dependências Owncloud parte 2



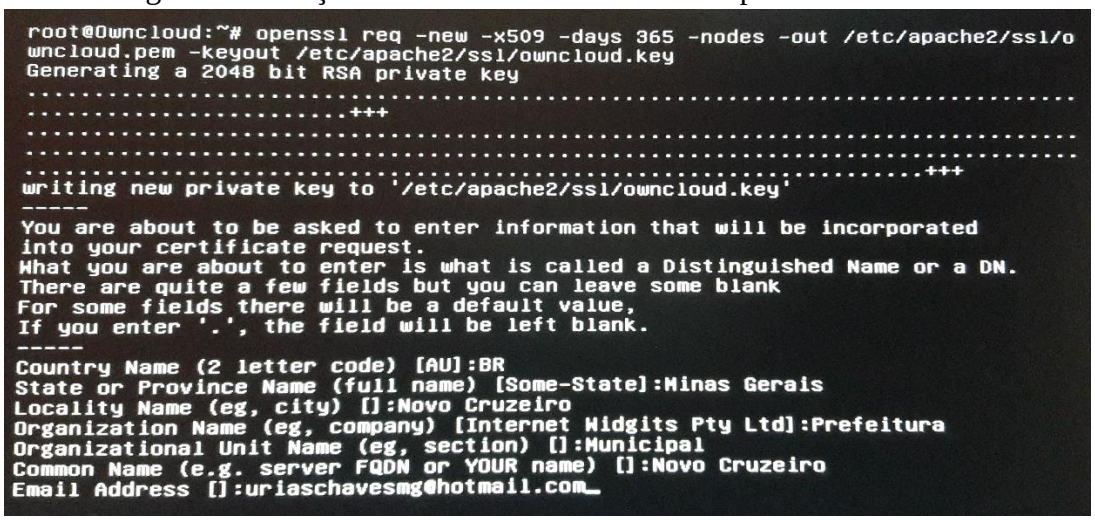
Fonte: Do Próprio Pesquisador

Nas Figuras 62 e 63, são detalhadas as dependências necessárias para a integração do owncloud com o Samba4 proposto para esse trabalho. Juntamente com o software Owncloud foi baixada a biblioteca *php5-libsmbclient* na qual está contida o servidor web *apache* que fará a propagação de acesso via navegador ou usando o software cliente do owncloud na rede. Em seguida conforme é demonstrado na Figura 63 foi baixado o *smbclient*, para realizar a comunicação entre o Samba e o Owncloud.

Por padrão o acesso ao Owncloud é feito na porta 80 usando o protocolo *http*, como forma de aumento da segurança foi ativado o acesso a porta 443 para usar o protocolo *https* por meio da criação de um certificado auto assinado que o navegador irá ter acesso quando fizer uma solicitação de conexão com o servidor.

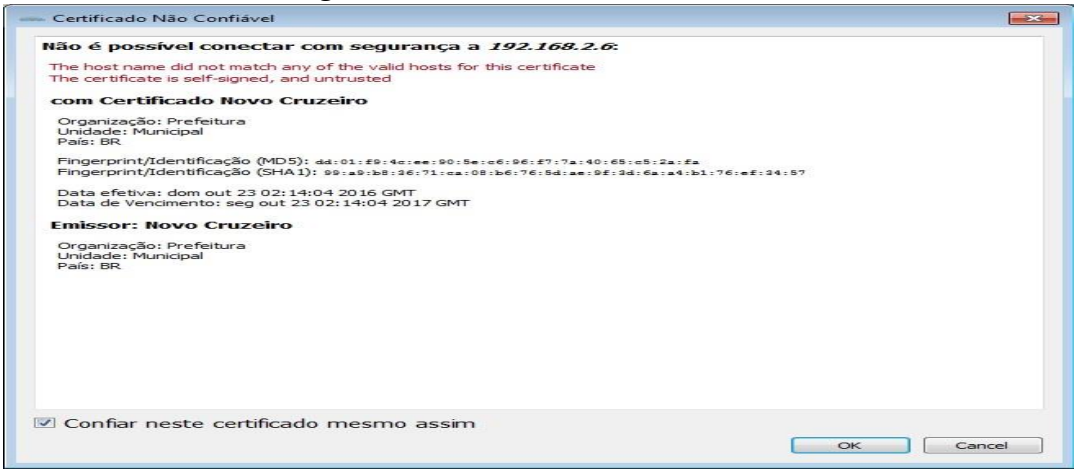
Na Figura 64 são demonstrados os requisitos usados na sua criação, antes desse processo foi baixado o software openssl com o comando *apt-get install opensll*, responsável pela geração do certificado.

Figura 64: Criação do Certificado auto assinado para Owncloud



Fonte: Do Próprio Pesquisador

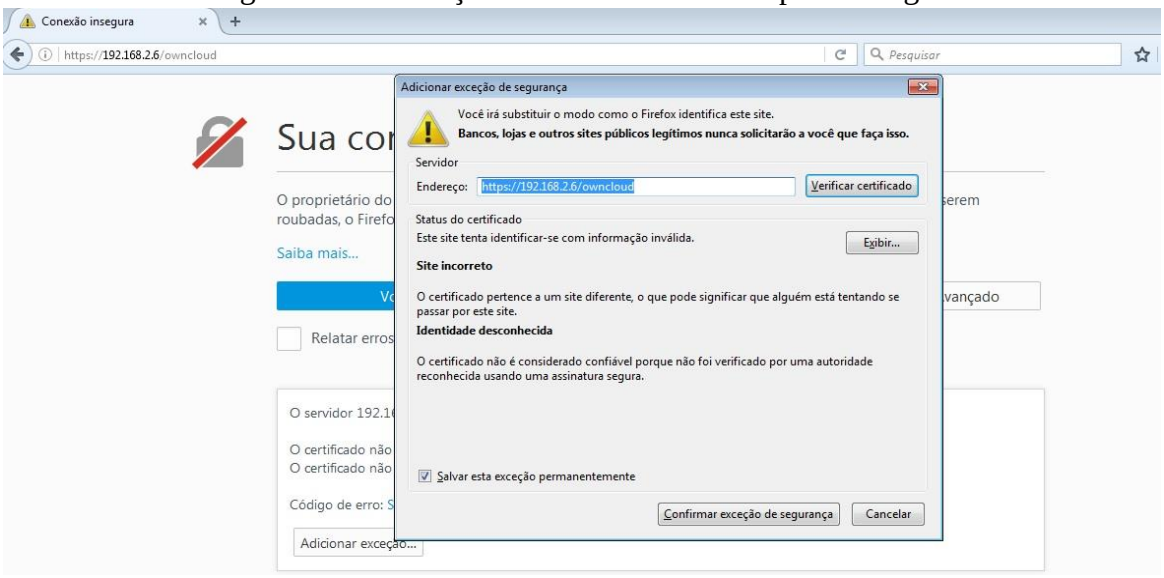
Figura 65: Certificado Auto assinado



Fonte: Do Próprio Pesquisador

Na Figura 65, é impresso de forma gráfico o certificado, solicitando ao usuário que confie na entidade para que o Owncloud trabalhe sobre as regras impostas a ele. Em seguida na Figura 66, é feito a tentativa de acesso onde é informada a existência de um certificado vinculado ao servidor, e solicitando que seja adicionado para o que o servidor redirecione para a porta 443 usando *https.*

Figura 66: Solicitação de uso do certificado pelo navegador



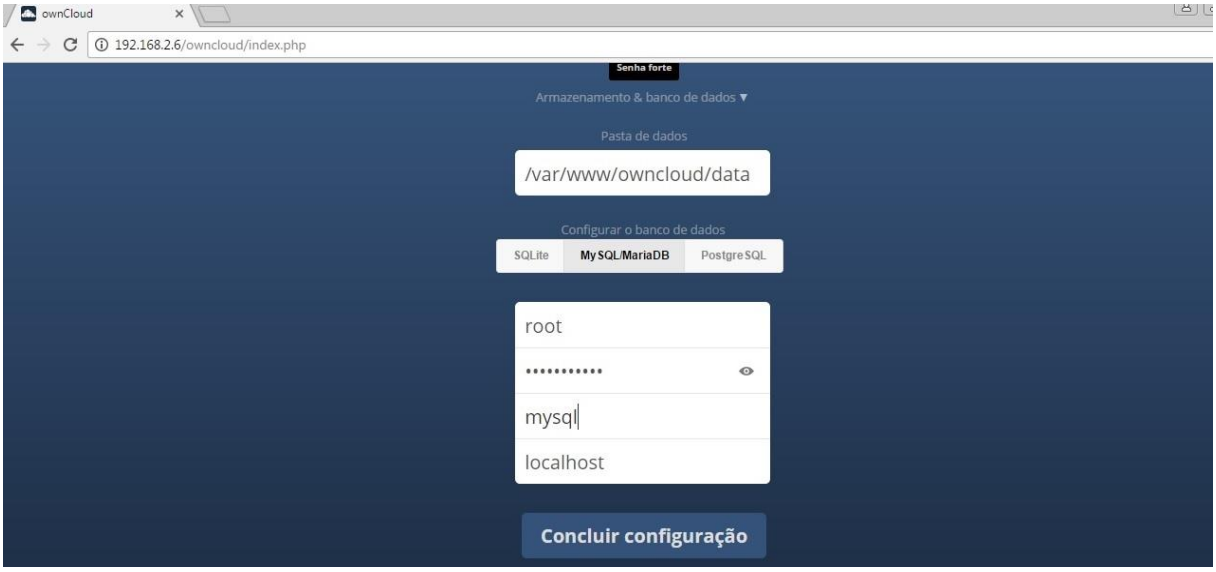
Fonte: Do Próprio Pesquisador

Figura 67: Configurando o Owncloud via Navegador parte 1

Fonte: Do Próprio Pesquisador

No primeiro acesso via navegador usando o IP do servidor e a pasta do Owncloud, é solicitado que sejas definidas as diretivas do seu funcionamento como o usuário e senha que irá administrar o servidor realizando as operações como cadastrar, atualizar, excluir usuários ou configurações necessárias para o seu funcionamento.

Figura 68: Configurando o Owncloud via Navegador parte 2



Fonte: Do Próprio Pesquisador

Em seguida é solicitado que informe qual banco de dados será usado, por padrão o Owncloud usa o SQLite, porem aqui optou-se por usar o Mysql, logo abaixo foram inseridas as informações de autenticação na máquina virtual, como o usuário root e a senha usada para acessar a máquina e por fim o nome do banco de dados. Como auxiliar na proteção dos dados foi criado o firewall demostrado abaixo.

- ✓ INPUT – analisa tudo que entra no servidor
- ✓ OUTPUT – analisa tudo que sai do servidor
- ✓ FORWARD – analisa tudo que passa pelo servidor
- ✓ PREROUTING – redireciona pacotes para portas especificas
- ✓ POSTROUTING – mascara os pacotes ocultando e identificando o IP
- ✓ ACCEPT – deixa a requisição passar
- ✓ DROP – bloqueia a requisição e não emite aviso ao solicitante
- ✓ REDIRECT –transfere a requisição para uma porta especificada
- ✓ MASQUERADE – faz o mascaramento dos pacotes
- ✓ -F: apaga todas as regras do firewall
- ✓ -X: apaga uma regra específica
- ✓ -t: qual tabela a ser usada
- ✓ -A: insere uma nova regra
- ✓ -s: destino de origem
- ✓ -d: destino final
- ✓ -j: solicita uma ação a ser aplicada

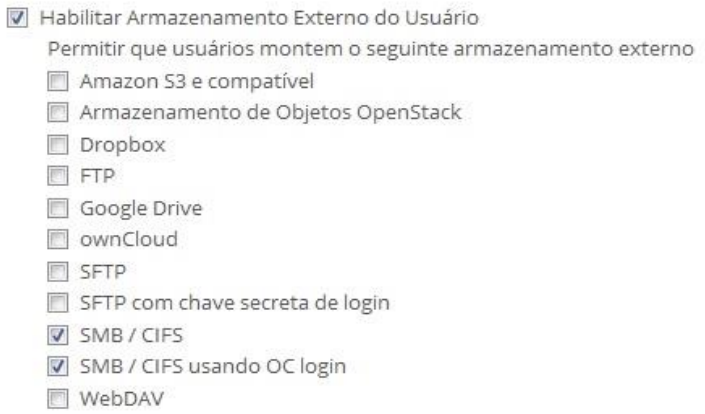
- ✓ -to-port: porta que será usada
- ✓ -i interface de rede a ser usada
- ✓ -p: solicita um protocolo
- ✓ --dport – porta de destino
- ✓ Limit – limita a quantidade de tentativas para execução da regra
- ✓ State – define um estado para a conexão
- ✓ Icmp – Internet Control Message Protocol
- ✓ -syn – permite o uso dos bits ACK E FIN
- ✓ --tcp-flags – analisa flags do protocolo TCP
- ✓ NEW – pacote que usa uma nova conexão
- ✓ ESTABLISHED – pacotes que tem conexão existente
- ✓ RELATED – pacote que não está na conexão existente
- ✓ INVALID – pacote desconhecido ou não identificado

```
#! /bin/bash
####PACOTES SUSPEITOS
iptables –A INPUT –p icmp –icmp-type echo-request –m limit –limit 1/s –j ACCEPT
iptables –A FORWARD –p icmp –icmp-type echo-request –j DROP
iptables –A FORWARD –p icmp –icmp-type echo-request –m limit –limit 1/s –j ACCEPT
###SCANNERS
iptables –A INPUT –p tcp –tcp-flags SYN,ACK,FIN, RST RST –m limit –limit 1/s –j ACCEPT
iptables –A FORWARD –p tcp –tcp-flags SYN,ACK,FIN, RST RST –m limit –limit 1/s –j ACCEPT
####PACOTES SUSPEITOS
iptables –A INPUT –m state –state INVALID –j DROP
iptables –A INPUT –p tcp ! –syn –m state –state NEW –j DROP
iptables –A FORWARD –m state –state INVALID –j DROP
iptables –A FORWARD –p tcp ! –syn –m state –state NEW –j DROP
####MASCARANDO OS PACOTES
iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE
```

Acessado o servidor, foi necessário ativar o modulo de armazenamento de arquivos externos para providenciar a integração ao servidor Samba por meio do protocolo *SMB/CIFS*

– *Server Message Block/ Common Internet File System* usado para compartilhamento de arquivos via rede.

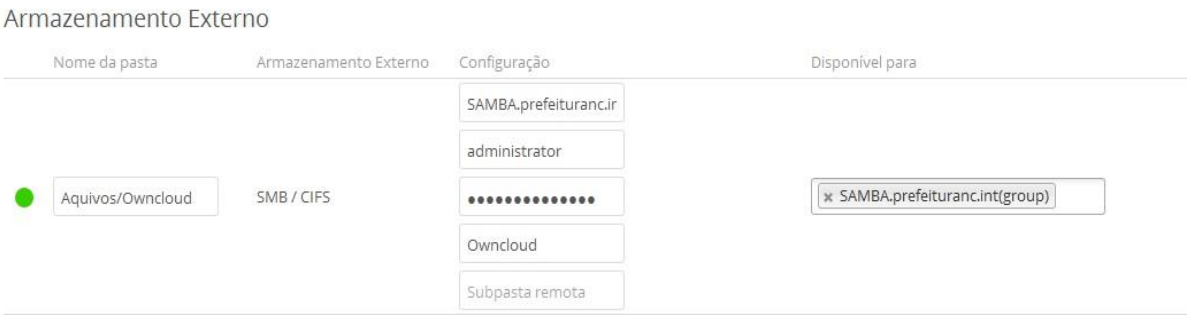
Figura 69: Habilitando Armazenamento Externo do Usuário



Fonte: Do Próprio Pesquisador

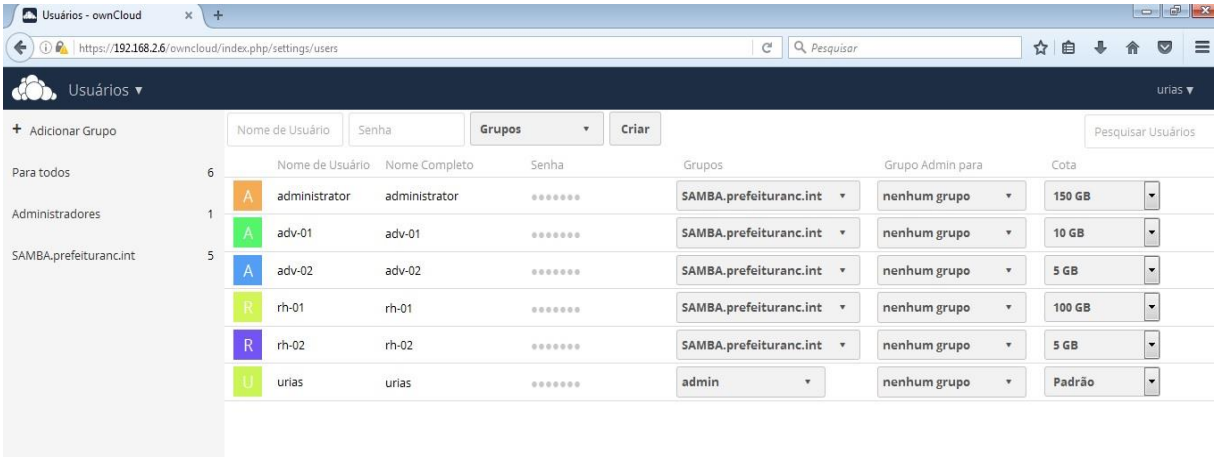
Para que houvesse a confirmação da integração dos servidores, foram usadas credenciais para autenticação do uso do serviço, conforme demonstra a Figura 70, com o usuário administrator do samba e sua senha, juntamente com o grupo que estará vinculado aos dois servidores.

Figura 70: Integração do Owncloud com Samba



Fonte: Do Próprio Pesquisador

Figura 71: Painel de Usuários do Owncloud



Fonte: Do Próprio Pesquisador

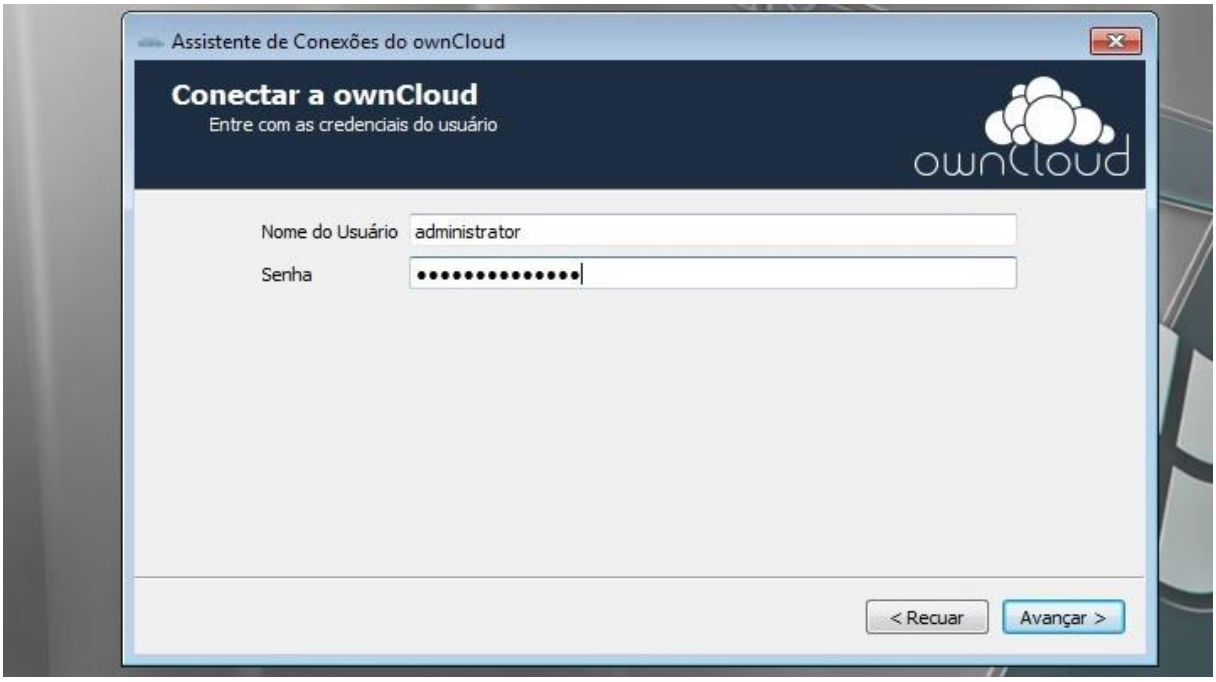
Como o painel do Owncloud é bem intuitivo, a criação dos seus usuários não proporciona dificuldade ao administrador. No canto direito superior aparece o nome do administrador do sistema, clicando sobre ele é exibido um sub menu e no mesmo foi clicado sobre o sub menu usuários, e direcionado para a tela que é exigida acima. Nesta tela foram criados os usuários com suas respectivas senhas e vinculados ao grupo do domínio samba e para cada um criado uma cota de upload de arquivos.

Após o cadastramento dos usuários, foi instalado o software Owncloud cliente para que ocorresse a sincronização dos arquivos. No processo, é solicitado que informe o endereço do servidor, para que seja verificado as credenciais do usuário, como vista na Figura 72 e posteriormente na 73.

Figura 72: Conectando o cliente ao Servidor Owncloud

Fonte: Do Próprio Pesquisador

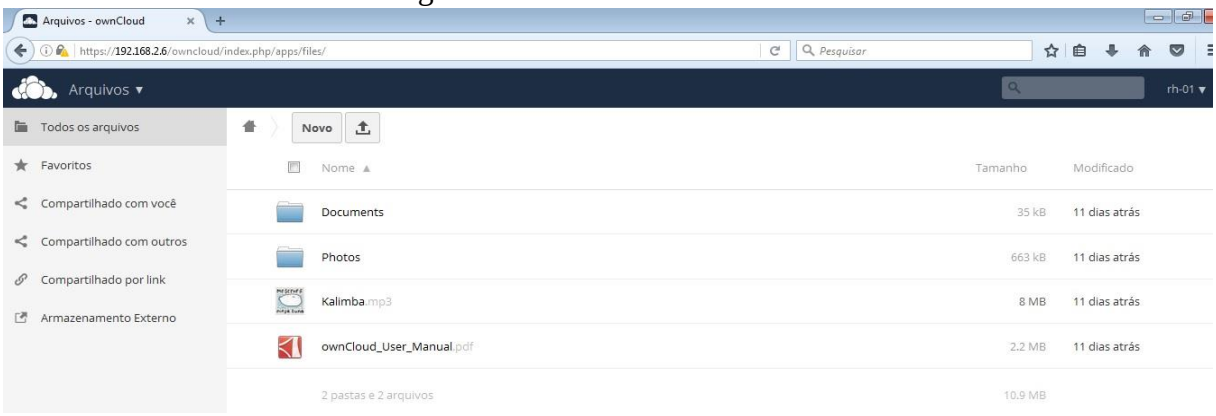
Figura 73: Autenticando o cliente no Servidor Owncloud



Fonte: Do Próprio Pesquisador

Finalizando a configuração o assistente, solicita que conecte no servidor, posteriormente será questionado o modo de visualização dos arquivos, seja via pasta local ou via navegador. Acessando via navegador basta informar https://ipdoservidor/owncloud que será direcionado para a pasta pessoal, onde é possível fazer o upload e/ou download, compartilhar, editar, dentre outras funções com os arquivos.

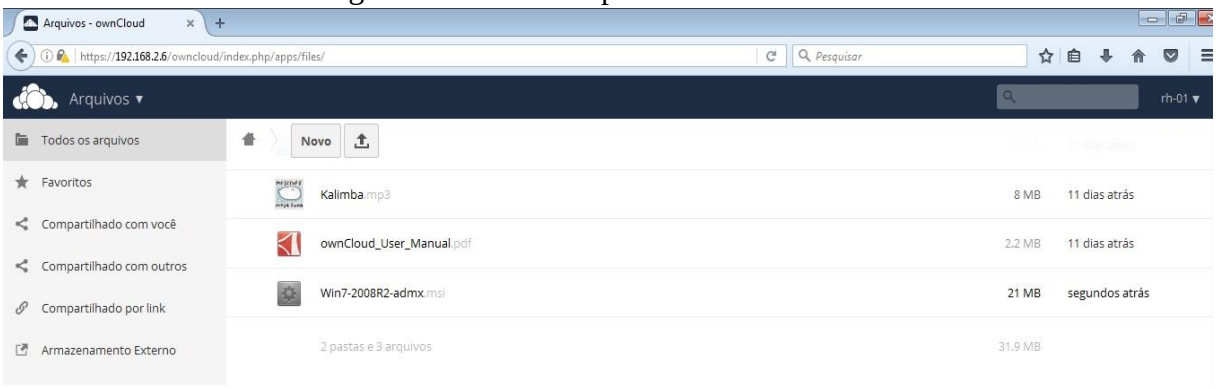
Figura 74: Painel do Usuário Owncloud



Fonte: Do Próprio Pesquisador

Caso deseje realizar as tarefas mencionadas, como realizar o backup dos dados da máquina local, basta apenas clicar na seta próximo ao botão novo conforme visto na Figura 74 e selecionar o arquivo que o Owncloud concluirá a tarefa.

Figura 75: Teste de Upload no Servidor Owncloud



Fonte: Do Próprio Pesquisador

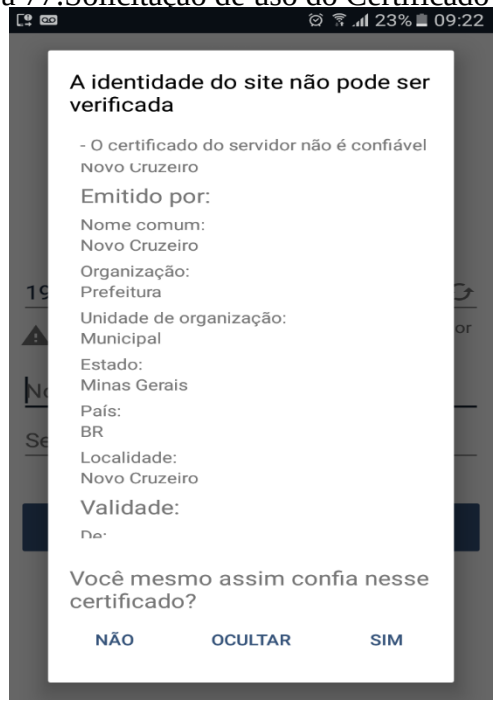
Para fazer uso da mobilidade proposta pelo servidor na rede, foi criada uma sub rede fazendo uso de um roteador afim de ofertar acesso mobile, ou seja, via celular aos usuários a seus dados armazenados no servidor. Foi feito o download e instalação do software cliente para mobile via *play store*, com uma interface de simples acesso para que o usuário insira o caminho do servidor, ou seja, o seu IP e seu nome seguido da senha, conforme exposto na Figura 76,

Figura 76:Tela de Acesso via Mobile ao Servidor Owncloud

Fonte: Do Próprio Pesquisador

Como já mencionando foi criado um certificado auto assinado para que o servidor funcione de forma segura via protocolo *https*, no primeiro acesso o mesmo solicita ao usuário permissão de uso com um aviso, alertando sobre sua entidade, sendo possível conferir se os dados dispostos são realmente os que foram informados na criado do mesmo, conforme constado na Figura 77.

Figura 77:Solicitação de uso do Certificado Owncloud



Fonte: Do Próprio Pesquisador

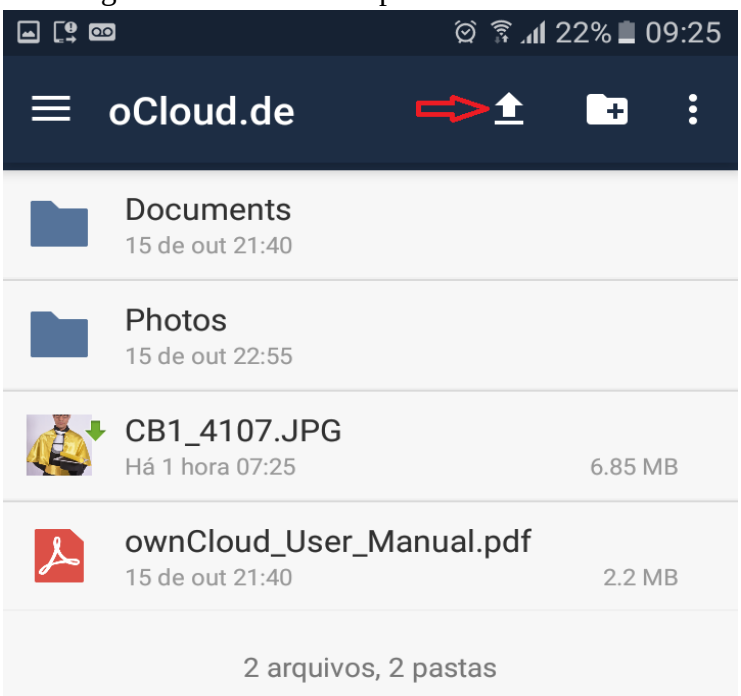
Uma vez acessado o servidor via celular, todas as operações já descritas anteriormente se tornam disponíveis ao usuário com seus respectivos arquivos enviados ao servidor da sua estação de trabalho, na Figura 78, é exibido o painel de configuração, onde o mesmo pode ter acesso a todos os seus dados que estão no servidor.

Figura 78: Painel de Configuração do software mobile

Fonte: Do Próprio Pesquisador

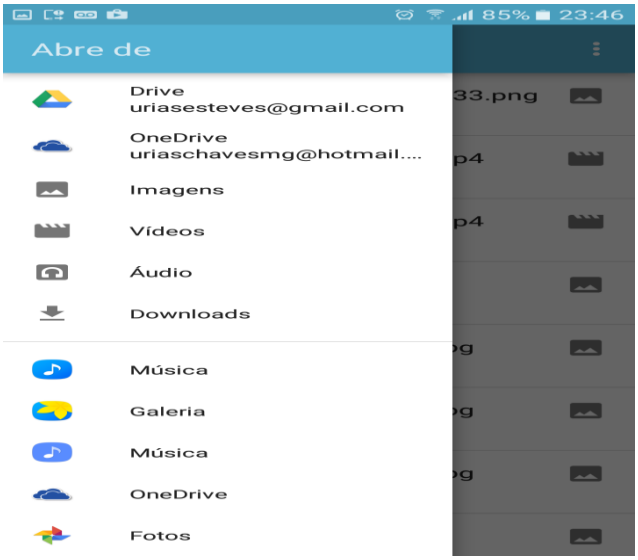
Para realizar um upload, ou seja, enviar um arquivo armazenando localmente no celular ou outro diretório de acesso externo para o servidor é necessário apenas um click no botão demarcado com um seta vermelha, como detalhado na Figura 79 abaixo e selecionar o arquivo do diretório desejado como é exibido na Figura 80, que automaticamente o mesmo será sincronizado com os demais dados já contidos no servidor junto a conta do usuário.

Figura 79: Painel Principal do software mobile



Fonte: Do Próprio Pesquisador

Figura 80: Acessando Diretórios para Upload de arquivo



Fonte: Do Próprio Pesquisador

4 RESULTADOS E DISCUSSÕES

No cenário montado para realizar os testes a fim de verificar a precisão dos serviços ofertados pelo servidor, foi constatado que os mesmos desempenharam suas funções as quais foram designados. Criado sobre o KVM foi possível gerar uma econômica de custo quanto à aquisição dos equipamentos, uma vez que o mesmo oferta a possibilidade de usar as configurações de hardware da máquina física gerando máquinas virtuais para armazenar os serviços proposto.

O uso do software livre possibilitou economia quanto ao uso de licenças e atualizações, pois exonerou quaisquer gastos nas mudanças de novas versões.

Para o usuário final os serviços dispostos geraram um nível a mais de segurança da ofertada pelo próprio sistema operacional local que agregado do firewall limitou a tramitação dos dados oriundos de acessos externos como a internet, bloqueando conteúdos impróprios ou considerados dispensáveis para o ambiente de trabalho como execução de vídeos online, acesso a mídias sociais.

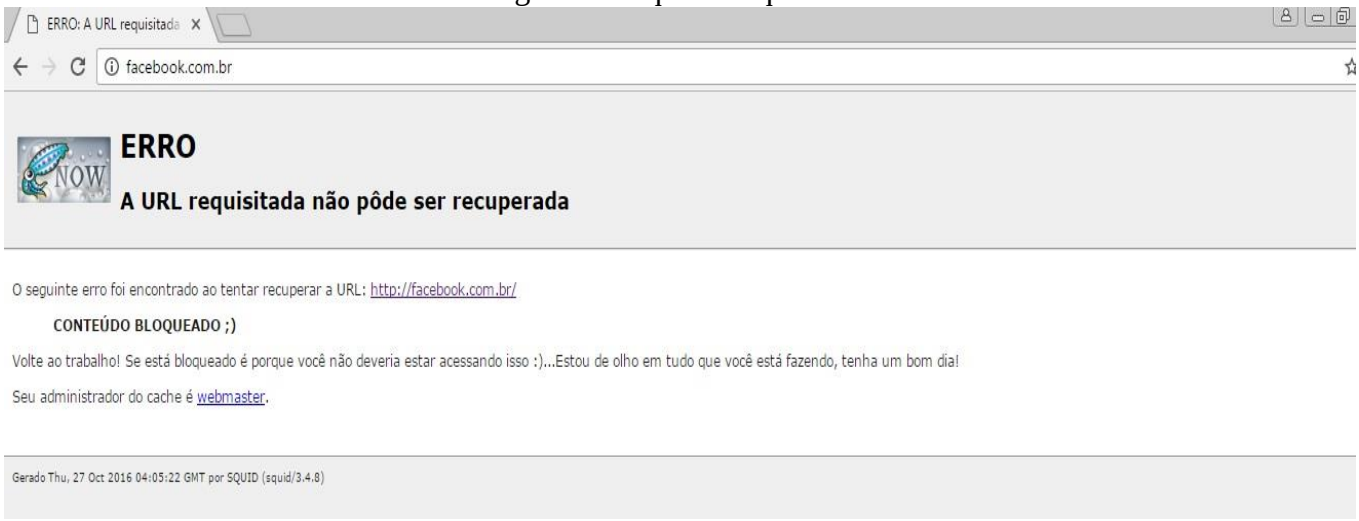
Com o armazenamento de cache feito pelo Squid foi gerado um aumento significativo no uso da internet, pois o mesmo trabalhou mantendo copias das páginas para um carregamento mais rápido pelos clientes.

O compartilhamento dos dados passou a ser de forma centralizada a cada departamento e seus usuários com o uso do Samba, inibindo o acesso as informações de usuários curiosos da rede.

CONSIDERAÇÕES FINAIS

Foi constatado a veracidade da afirmação feita na hipótese 1 e 3 quanto ao uso do squid e firewall, através de testes feitos em clientes usando o VirtualBox[17] o Squid cumpriu com o seu papel, bloqueando os conteúdos especificados como impróprios para uso na rede e as requisições que usam protocolo https foram tratadas no firewall.

Figura 81: Squid Bloqueando Sites

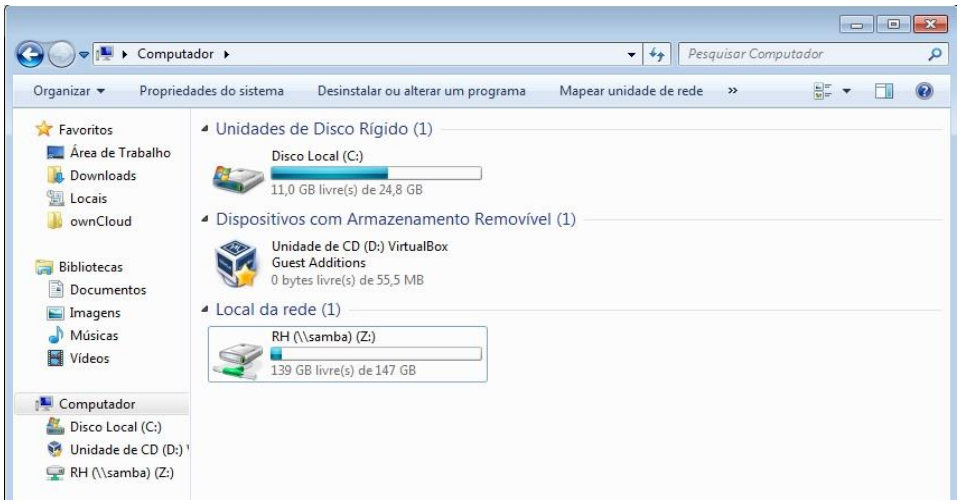


Fonte: Do Próprio Pesquisador

A hipótese 2, também foi validada com a comprovação do acesso a rede pelos usuários, os mesmos tendo acesso somente aos seus diretórios para compartilhamento mapeado como unidade de rede.

[17] https://www.virtualbox.org/

Figura 82: Mapeamento de Diretório do Samba



Fonte: Do Próprio Pesquisador

Hipótese 4 é válida pois já foi explicada e exemplificada nas Figuras 74 e 75, com o envio de um arquivo para o servidor, ou seja, mantendo o arquivo fora do armazenamento local como forma de backup, sendo possível acessa-lo de qualquer lugar da rede interna e realizar as operações citadas.

A hipótese 0 considerada nula, foi constatada como valida devido a transição de gestores no órgão. Optou-se por não arcar com o custo de um computador para a criação do ambiente de rede.

Contudo, o desenvolvimento do servidor apresentado forneceu um meio de redução nos gastos relacionados ao uso de equipamentos tecnológicos em uma rede de pequeno porte, visto que o uso do KVM gera a possibilidade de ofertar inúmeros serviços de rede através de máquinas virtuais, máquinas estas que desempenham o papel fielmente ao feito pelas máquinas físicas.

REFERÊNCIAS

ANDRADE, Maria Margarida de. *Introdução à metodologia do trabalho cientifico*. 5. Ed. São Paulo: Atlas, 2001.

BRETT SMITH, *Um guia rápido para GPLv3*. Disponível em < http://www.gnu.org/licenses/quick-guide-gplv3.html>. Acessado em 14 mai.2016.

BERBERT DE PAULA Fábio. Interfaces Gráficas. Disponível em <https://www.vivaolinux.com.br/artigo/Interfaces-Graficas-no-Linux>. Acessado em 07 mai.2016.

ESTANISLAU DA SILVA Jefferson, *História do GNU/Linux: 1965 assim tudo começou!*. Disponível em < https://www.vivaolinux.com.br/artigo/Historia-do-GNU-Linux-1965-assim-tudo-comecou/>. Acessado em 29 mai.2016

FERREIRA, Rubem E. *Linux: guia do administrador do sistema.* 2. ed. rev. e ampl. São Paulo: Novatec Editora, 2008.

FUNDAÇÃO DO SOFTWARE LIVRE. Disponível em < http://www.gnu.org/>. Acessado em 25 abr.2016.

GIL, Antônio Carlos. *Métodos e técnicas de pesquisas sociais.* 5.ed. São Paulo: Atlas, 1999.

GUSTAVO ARAÚJO RESENDE Geraldo, *Linux – Breve introdução, bom para iniciantes.* Disponível em <https://www.vivaolinux.com.br/artigo/Linux-Breve-introducao-bom-para-iniciantes>. Acessado em 28 mai.2016.

HUNT, Craig. *Linux: servidores de rede.* Rio de Janeiro: Editora Ciência Moderna Ltda, 2004.

SOARES, Luiz Fernando Gomes; LEMOS, Guido; COLCHER, Sérgio. *Redes de Computadores: das LANs, MANs e WANS às redes ATM.* 2º.ed. Rio de Janeiro: Elsevier, 1995.

STATO FILHO, André. *Domínio Linux: do básico aos servidores.* 2ª ed. Florianópolis: VisualBooks, 2004.

STREBE, Matthew; PERKINS, Charles: *Firewalls.* São Paulo: MARKRON Books, 2002.

TANENBAUM, Andrew S. *Redes de computadores.* 17ª reimpressão. Rio de Janeiro: Elsevier, 2003.

TANENBAUM, Andrew S. *Sistemas operacionais modernos*. 3. ed. São Paulo: Pearson Prentice Hall, 2009.

KVM. Disponível em < https://wiki.debian.org/KVM >. Acessado em 27 de abril de 2016.

Configurar um Active Directory Samba Domain Controller. Disponível em < https://wiki.samba.org/index.php/Main_Page>. Acessado em 15 mai.2016.

Licenças. Disponível em < http://www.gnu.org/licenses/licenses.pt-br.html>. Acessado em 14 mai.2016.

Squid: Entrega Web Optimização. Disponível em < http://www.squid-cache.org/>. Acessado em 14 mai.2016.

Uso seguro da Internet. Disponível em < http://cartilha.cert.br/uso-seguro/>. Acessado em 20 mai.2016

APENDICE: Modelo Operacional da Pesquisa