Solução em *Software* Livre para Controle de Acesso a Conteúdo WEB

Jones Bunilha Radtke

[1]Faculdade de Tecnologia SENAC
Rua Gonçalves Chaves, 602 – Pelotas – RS – Brasil
Caixa Postal – 96015560

jones.radtke@gmail.com

Resumo. *Este artigo tem por objetivo apresentar o estudo de uma solução em software livre para a realização de controle de acesso a conteúdo WEB por meio de proxy com autenticação de forma automática no Active Directory. Neste será apresentado testes efetuados para determinar o desempenho com a utilização deste recurso.*

Abstract. *This article aims to present the study of a free software solution for performing access control to web content via proxy with authentication automatically in Active Directory. This will be presented tests performed to determine the performance using this feature.*

1. Introdução

Toda rede de computadores onde haja administração, se faz necessário um mecanismo para controlar e gerenciar o conteúdo WEB acessado pelos usuários da rede. Atualmente, na rede mundial de computadores, pode-se facilmente encontrar conteúdos ilícitos potencialmente prejudiciais, que tornam-se uma ameaçam a segurança da rede como um todo. Outros fatores relevantes e que devem ser levados em conta em uma rede de computadores é garantir a disponibilidade de serviços com qualidade, visto que, os recursos de rede poderão estar sendo desperdiçados em acessos que não condizem com as políticas da empresa, tornando-se um prejuízo e diminuindo a produtividade de seus colaboradores.

Um indício sempre presente é o *download* e a propagação de *software* não autorizados e não licenciados, caminhando contra a ética e moral de uma corporação, pois o combate a pirataria encontra-se cada vez mais em evidência, desta forma, políticas de acesso são impostas através de liberações e bloqueios a conteúdo hospedado na *Internet* afim de coibir tais práticas. O recurso fortemente utilizado pelos administradores de rede é a utilização de um *proxy* WEB, onde este será o responsável por realizar as requisições das páginas solicitadas, podendo assim, realizar um controle sobre o conteúdo que será acessado.

2. Fundamentação teórica

Nesta seção serão abordados alguns conceitos e definições que serão necessários para uma melhor compreensão do artigo.

2.1. Kerberos

O Kerberos é um protocolo de rede que permite comunicações individuais seguras e identificadas. Desenvolvido pelo *Massachusetts Institute of Technology*, disponibiliza um pacote de aplicativos que implementam o protocolo no intuito de garantir a integridade dos dados. Inicialmente foi projetado na arquitetura cliente-servidor sendo possível a autenticação mutua [MTI 2014]. A Microsoft utiliza uma variante do Kerberos, como seu método de autenticação padrão. As modificações realizadas no conjunto de protocolos do Kerberos são documentadas na RFC 3244 chamada de **Microsoft Windows 2000 Kerberos change Password and Set Password Protocols** [RFC3244 2014].

2.2. NetBios

O NetBIOS é um protocolo desenvolvido para prover a compatibilidade de serviços de rede entre sistemas Microsoft Windows. O protocolo é o mecanismo padrão de resolução de nomes, tradução de endereço IP para o *hostname*, em uma rede IP sem a necessidade de um serviço de DNS [Tecmundo 2014]. Este consiste em um endereço de 16 *bytes* usado para identificar um recurso NetBIOS na rede. Quando um processo do protocolo está comunicando-se com um processo específico em um computador, é usado um nome exclusivo para tal. Quando um processo de NetBIOS está comunicando-se com vários processos em vários computadores, é usado um nome de grupo [Tecmundo 2014]

2.3. *Proxy*

O *proxy* é um servidor responsável por encaminhar as requisições solicitadas pelos clientes que utilizam os protocolos HTTP, HTTPS e FTP. Este serviço é capaz de analisar o conteúdo solicitado, registrando-o e verificando em listas, ACLs (*Acces Control List*), se tal solicitação é permitida ou bloqueada de acordo com as políticas impostas [Ricci 2006]. Tais permissões, em geral, são classificadas por grupos de acesso, sendo estes compostos por computadores ou usuários.

A Figura 1, ilustra o mecanismo de requisição de páginas a um servidor WEB por intermédio de um servidor *proxy*. O cliente solicita ao *proxy* a página na qual deseja carregar, e este, é o responsável por solicitar ao servidor WEB e entregar para o computador cliente.

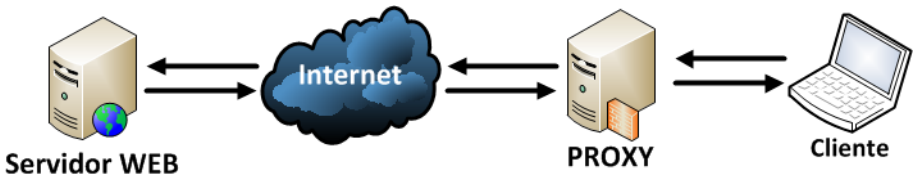


Figura 1. Requisição de página a um servidor WEB por intermédio de um servidor *proxy*.

O servidor também dispõe de recursos como limitar a largura de banda de acessos a um domínio em específico e a possibilidade de realizar *cache*, armazenamento dos conteúdos mais acessado. Ambos recursos tornam-se interessantes quando há a necessidade de se garantir a qualidade de serviço em acessos indispensáveis para os processos de trabalho de uma organização, evitando desperdício de largura de banda com conteúdo de menor importância.

No mercado atual existem algumas alternativas de *proxies* proprietários e soluções em *software* livre, bem como *appliances* intitulados de UTM (*Unified Threat Management*) que caracterizam-se como uma solução abrangente, criada para o setor de segurança de redes, onde em um único dispositivo, com *software* e *hardware* dedicado, centraliza os recursos de segurança e garante um alto desempenho através das ferramentas de *firewall*, *proxy*, prevenção de intrusões de rede, antivirus, VPN, balanceamento de carga e geração de relatórios de acesso [Ricci 2006].

Tais dispositivos são, em grande parte, soluções proprietárias com um custo elevado, tornando-se inviável a sua aplicabilidade em determinados casos, desta forma, a utilização de *software* livre ainda encontra-se em evidência para tal finalidade.

2.4. WinBind

O Winbind é um componente disponível para o Samba destinado a disponibilizar recursos de integração entre sistemas UNIX e Microsoft Windows. Utiliza uma implementação em UNIX para as chamadas Microsoft RPC, *Pluggable Authentication Modules* (PAMs), e o interruptor de serviço de nome (NSS), afim de permitir que usuários do domínio Windows sejam utilizados como usuários do UNIX. Desta forma, o serviço permite a unificação UNIX e Windows com relação a gestão de contas de usuários, fazendo com que um sistema UNIX se torne um membro com pleno direito de um domínio Windows, sendo possível a utilização de usuários do Windows e grupos, como se fossem nativos usuários e grupos UNIX [Samba 2014].

O resultado é que sempre que há a necessidade de um processo, em uma máquina UNIX, solicitar ao sistema operacional a busca de um usuário ou grupo em específico, a consulta será resolvida por meio do controlador de domínio Windows, pois o Winbind utilizará recursos em baixo nível do sistema operacional, como os módulos NSS de resolução de nomes na biblioteca C, e este redirecionamento para o controlador de domínio será completamente transparente para o usuário que está operando o sistema [Samba 2014].

A única indicação de que o Winbind estará sendo utilizado é que os nomes de usuário e grupo irão assumir a configuração **DOMÍNIO\usuário** e **DOMÍNIO\grupo**, isto se faz necessário, porque permite o Winbind determinar quando é necessário o redirecionamento para um controlador de domínio [Samba 2014].

Além disso, é possível fornecer um serviço de autenticação que conecta no sistema PAM para fornecer autenticação através de um domínio para todos os aplicativos habilitados para o PAM. Esse recurso resolve o problema de sincronização de senhas entre sistemas, uma vez que todas as senhas são armazenadas em um único local, no controlador de domínio [Samba 2014].

3. Solução de Proxy automaticamente Autenticado

Por meio da integração das ferramentas, Squid, Active Directory da Microsoft, Samba na versão 4 e SARG é possível a implementação de uma solução em *software* livre de controle e monitoramento dos acessos na utilização de recursos da *Internet*, de forma autenticada e automática, com a identificação do usuário solicitante.

3.1. Squid

O Squid é uma solução em *software* livre para a implementação de um servidor *proxy* para requisições WEB com suporte ao protocolos HTTP, HTTPS, FTP. Os benefícios de sua utilização estão em reduzir a largura de banda e melhorar os tempos de resposta com a utilização do *cache*.

A reutilização de páginas WEB, frequentemente solicitado, fazem este recurso um importante aliado. O serviço dispõe, ainda, de amplos controles de acesso, permitindo a autenticação de usuários por variados métodos e compatível com a maioria dos sistemas operacionais disponíveis como Microsoft Windows e sistemas UNIX, onde é altamente aplicado e licenciado sob a GNU/GPL [Cache 2014].

3.2. *Active Directory* Microsoft

O Active Directory (AD) é uma implementação de serviço de diretórios da Microsoft. Um serviço de diretórios é uma forma organizada de armazenar informações sobre os recursos e os utilizadores de uma rede. Geralmente este segue o organograma hierárquico da instituição e permite aos administradores de rede gerenciarem o acesso de usuários a recursos e sistemas por meio de políticas de grupos e unidades organizacionais aplicadas em um domínio em específico.

Uma rede poderá conter vários domínios, e este será o limite administrativo e de segurança. O administrador de domínio possui permissões somente em seu domínio e as políticas de segurança também se aplicaram somente ao domínio, neste caso, diferentes domínios podem ter diferentes administradores e diferentes políticas de segurança [Microsoft 2014].

Pode-se definir que o serviço de diretório atua como uma camada abstrata entre o usuário e estes sistemas, recurso que surgiu da necessidade de centralizar o gerenciamento de acessos aos mais variados serviços de rede, como *e-mail*, serviços FTP e WEB. Os serviços tornam-se disponíveis na rede, quando o usuário efetua *logon* no sistema, neste instante é realizada uma verificação no AD e se as informações fornecidas pelos usuários, como *login* e senha, são válidas e sua autenticação é efetuada disponibilizando o serviço para o uso [Microsoft 2014].

Nos domínios baseados no AD, há dois tipos de servidores o **Controlador de Domínio** (*Domain Controller*) e **Servidor Membro** (*Member Server*). O **Controlador de Domínio** é o servidor que contém a base de dados principal das configurações do AD, e é automaticamente replicada para os **Servidores Membros** com o intuito de criar redundância dos dados [Microsoft 2014].

3.3. Samba

O Samba é um serviço e um conjunto de recursos que permite o compartilhamento de arquivos, impressoras e implementação de domínios, serviço de diretórios, recurso este disponível partir da versão 4 do Samba, de sistemas Microsoft Windows na plataforma UNIX.

O projeto Samba surgiu da necessidade pessoal de seu desenvolvedor, o Australiano Andrew Tridgell, que inicialmente necessitava montar um espaço no disco do seu

computador em um servidor UNIX, desta forma, foi preciso que houvesse suporte ao NetBIOS, protocolo de resolução de nome em redes Microsoft, e como utilizava o sistema de arquivos NFS (*Network File System*), que não oferece suporte ao NetBIOS, Tridgell desenvolveu um *sniffer*, *script* utilizado para captura de tráfego de dados em rede, e assim, permitiu efetuar uma análise e a interpretação dos dados capturados gerado pelo NetBIOS. Com esta ferramenta foi possível a realização de engenharia reversa e o protocolo SMB, Samba, e foi implementado no sistema UNIX, tornando possível a comunicação entre as duas plataformas.

Em 1992 foi disponibilizado o código publicamente, mas somente em 1994 a Microsoft disponibilizou as especificações do SMB e do NetBIOS, possibilitando um avanço no desenvolvimento do Samba [Samba.org 2014].

Na versão 4 do Samba torna-se possível a implementação do *Active Directory*, tornando um sistema UNIX um Controlador de Domínio ou um Servidor Membro, o que permite gerenciar todos os recursos disponível na versão do AD da Microsoft através de um UNIX, aumentando os recursos presente nesta ferramenta [Foca 2014].

3.4. SARG

SARG (*Squid Analysis Report Generator*) é um utilitário em UNIX, *open source*, que tem por finalidade a geração de relatórios, semanais e mensais, em HTML, baseado nos arquivos de *logs* do Squid. Com esta ferramenta é viável o monitoramento dos *sites* e recursos de *Internet* utilizado pelos usuários da rede, sendo possível verificar a lista dos endereços mais acessados, sendo classificada por endereço IP ou usuário, os endereços no qual foram bloqueados o acesso, a quantidade de dados trafegados, uso total de largura de banda e os *downloads* efetuados por cada usuário [Tecmint 2014].

4. Implementação

Em grande parte das corporações, onde é aplicada políticas de controle de acesso, faz-se necessário a implantação de um servidor *proxy*. Uma das soluções mais utilizadas é o Squid, serviço que realiza a verificação do conteúdo requisitado e avalia se encontra-se dentro das políticas estipuladas.

Em geral, o Squid, realiza um confronto entre dois dados no ato da solicitação de uma página, é verificado o domínio solicitado e o endereço de origem do solicitante. A partir de então, aplica-se as regras estabelecidas, liberando ou bloqueando o acesso ao conteúdo para o computador cliente. No sistema, são criadas listas de domínios, bem como, listas de grupos de clientes onde são constituídas de endereços físicos ou endereços de rede IP e a combinação de ambas listas são utilizadas juntamente com regras de liberação e bloqueio, *allow* e *deny*, para criar as políticas de acesso. Este tipo de controle torna a utilização da rede de forma estática, por parte dos usuários, e sem a possibilidade de identificação dos acessos, já que, as listas são constituídas por endereços fixos, consequentemente computadores, e não por usuários.

No instante em que a opção é utilizar um *proxy*-Squid, com autenticação, torna-se possível a identificação e a aplicação de políticas por grupos de usuários. Para tornar o mecanismo de autenticação no *proxy*, automática, utiliza-se recursos disponíveis no Squid, efetuando a sua integração com o *Active Directory*, e assim, possibilitando a centralização

do gerenciamento do sistema, desta forma, as políticas de acesso passam a ser organizadas de acordo com os grupos existentes no AD, aumentando o controle e facilitando as configurações do *proxy*.

Na Figura 2 é apresentado um diagrama dos servidores implementados para o estudo da integração dos serviços.

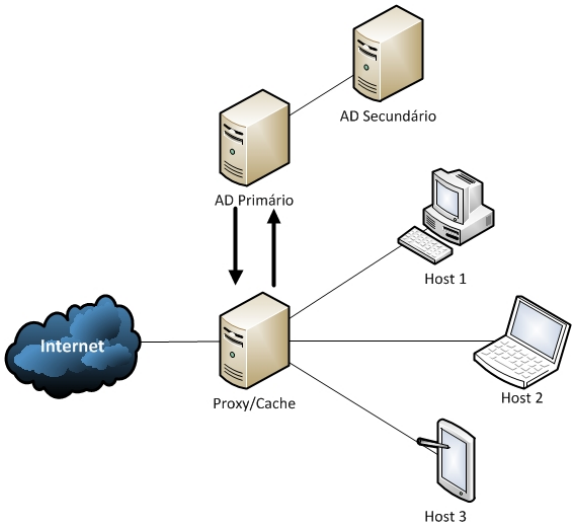


Figura 2. Diagrama dos Servidores Implementados.

Para o estudo da solução a implementação de todos os servidores e clientes foi efetuada em ambiente virtualizado com o uso da ferramenta *VMWare Player* [VMWare 2014].

4.1. Servidor *Proxy/Cache*

O servidor *Proxy/Cache* é composto por um Debian 7.4 (*Wheezy*), onde foi instalado o Squid 3. Para efetuar a sua integração com o *Active Directory*, necessita-se que o servidor tenha instalado os pacotes de suporte ao protocolo Kerberos, responsável pelo processo de autenticação dos sistemas Windows, Winbind e Samba, ambos responsáveis em realizar a resoluções de nomes de *hosts* Windows e integração do sistema Unix com o AD.

Com a configuração do Squid através do arquivo */etc/squid3/squid.conf*, permitiu que o serviço se comunique com o servidor AD e realize a verificação do usuário que efetuará a autenticação.

Na Figura 3 são listados os módulos de mecanismos de autenticação necessário para tal tarefa.

```
auth_param ntlm program /usr/bin/ntlm_auth –helper-protocol=squid-2.5-ntlmssp
auth_param ntlm children 10
auth_param ntlm keep_alive on
auth_param basic program /usr/bin/ntlm_auth –helper-protocol=squid-2.5-basic
external_acl_type grupo_ad %LOGIN /usr/lib/squid3/wbinfo_group.pl
```

Figura 3. Mecanismos de autenticação.

Ainda no arquivo *squid.conf* ocorre a configuração das ACLs de integração dos grupos de usuários dispostos no AD, com as políticas adotadas no *proxy* conforme apresentado na Figura 4 :

```
acl grp-presidencia external grupo_ad Presidencia
acl grp-comercial external grupo_ad Comercial
acl grp-treinamento external grupo_ad Treinamento
```

Figura 4. ACLs de integração com AD.

No exemplo a ACL **grp-presidencia** faz referência ao grupo de usuário **Presidencia** existente no AD e a ACL **grp-comercial** faz ao grupo **Comercial** e segue esta a lógica.

Com os parâmetros e ACLs da Figura 5, a criação de políticas passa a ser efetuada de acordo com os grupos do AD.

```
http_access allow grp-presidencia "lista de domínios"
http_access deny grp-treinamento "lista de domínios"
```

Figura 5. Criação de políticas por grupos do AD.

Para a organização dos domínios que serão liberados ou bloqueados, as listas de domínios podem ser classificadas de acordo com categoria de *sites* semelhantes, desta forma, as políticas podem ser aplicadas de forma individual para cada grupo de usuários.

Um exemplo seria a criação de uma lista denominada **multimídia** onde, seria disposta de *sites* que possuem por prática comum a publicação de conteúdo de audio e video. Em geral o acesso a este tipo de conteúdo consome uma alta largura de banda do *link* de *Internet*, tornado-se nocivo ao desempenho da rede, desta forma, pode-se efetuar as políticas da seguinte forma, onde a lista **multimídia** pode conter domínios como *youtube.com* e similares.

```
http_access allow grp-presidencia multimidia
http_access deny grp-treinamento multimidia
```

Figura 6. Exemplo de criação de políticas por grupos do AD.

Com o exemplo da Figura 6, os usuários pertencentes ao grupo **Presidencia** terão acesso aos *sites* da lista **multimidia** e os usuários do grupo **Treinamento** serão bloqueados ao acessar tais *sites*.

Seguindo está lógica, a administração do *proxy* é efetuada pelos grupos do AD, adicionando ou removendo algum usuário de um grupo para disponibilizar ou bloquear o acesso a um domínio em específico, o que torna mais prática e organizada a administração das políticas de acesso dos clientes.

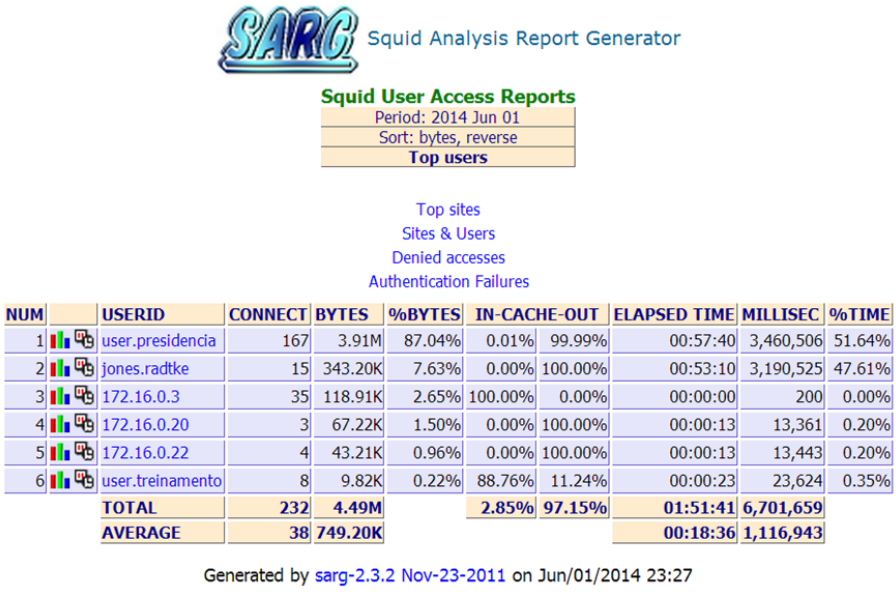
Outro recurso para reduzir a largura de banda é a utilização de *cache* no Squid. Com os parâmetros listados na Figura 7, pode-se ativar e configurar este recurso.

```
cache_mem ”tamanho do cache em MB”
cache_dir ufs /var/spool/squid3 500 16 256
```

Figura 7. Configuração de cache.

Com o objetivo de manter um monitoramento e a geração de relatórios periódicos sobre os acessos efetuados por cada usuário a *Internet*, a ferramenta SARG foi instalada no servidor.

Por uma interface WEB é possível a visualização das informações conforme observa-se na Figura 8. Para o correto funcionamento do recurso torne-se necessário a instalação de um serviço WEB, através dos módulos Apache [Apache 2014].



NUM		USERID	CONNECT	BYTES	%BYTES	IN-CACHE-OUT		ELAPSED TIME	MILLISEC	%TIME
1		user.presidencia	167	3.91M	87.04%	0.01%	99.99%	00:57:40	3,460,506	51.64%
2		jones.radtke	15	343.20K	7.63%	0.00%	100.00%	00:53:10	3,190,525	47.61%
3		172.16.0.3	35	118.91K	2.65%	100.00%	0.00%	00:00:00	200	0.00%
4		172.16.0.20	3	67.22K	1.50%	0.00%	100.00%	00:00:13	13,361	0.20%
5		172.16.0.22	4	43.21K	0.96%	0.00%	100.00%	00:00:13	13,443	0.20%
6		user.treinamento	8	9.82K	0.22%	88.76%	11.24%	00:00:23	23,624	0.35%
		TOTAL	232	4.49M		2.85%	97.15%	01:51:41	6,701,659	
		AVERAGE	38	749.20K				00:18:36	1,116,943	

Figura 8. Interface do SARG.

Na tela inicial do SARG, ainda pode-se observar as informações classificadas por usuário. Na Figura 9 tem-se as informações de um usuário em específico, *user.treinamento*, onde foi registrado um bloqueio de acesso ao domínio *youtube.com*.

4.2. Servidor AD Primário

Utilizando um Windows Server 2008, foi possível promovê-lo a um **Controlador de Domínio**, instalando o serviço *Active Directory* através do comando ”*dcpromo*”, e assim iniciar a criação das **Unidades Organizacionais** (OUs) e grupos de usuários, que são utilizados para a aplicação das políticas de acesso a conteúdos implementadas no *proxy*.

A figura 10, ilustra a estrutura de grupos implementadas no AD, e os usuários pertencentes a cada grupo.

4.3. Servidor AD Secundário

Com os novos recursos dispostos no Samba 4 [Vaz 2013], também tornou-se viável a implementação de uma redundância do AD Microsoft, utilizando um servidor UNIX com

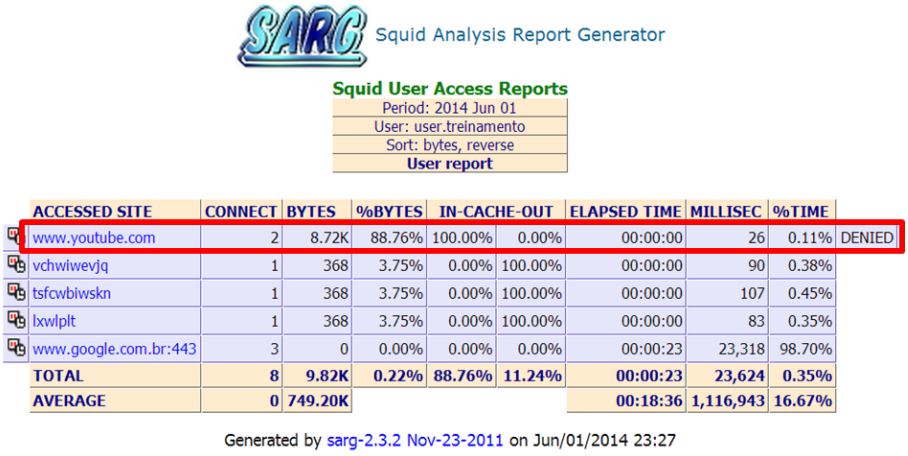


Figura 9. Informações de acesso por usuário.

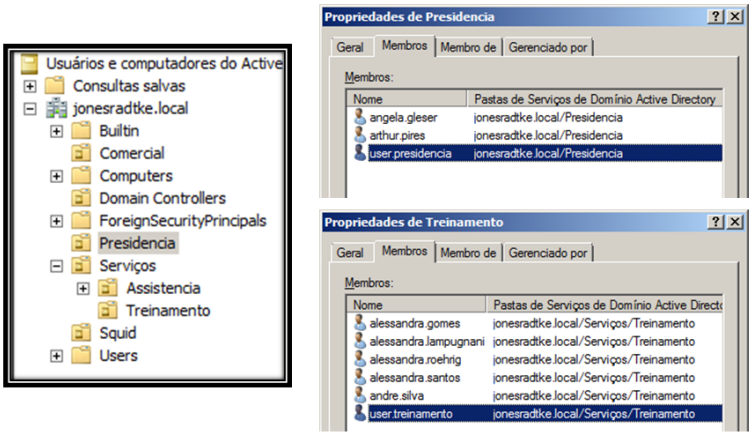


Figura 10. Hierarquia de OUs e Grupos de Usuários criados.

o sistema operacional CentOS 6.3 instalado. Este servidor atuará como um **Servidor Membro** do domínio e todo recurso criado no servidor AD Primário será automaticamente replicado para o Servidor Membro, criando uma cópia da base primária do AD.

5. Testes Realizados

Nesta seção será apresentado os testes efetuados na solução de *proxy*, afim de mensurar quais os prejuízos em termos de performance com a utilização do recurso de autenticação, e assim, quantificar o atraso nas requisições dos computadores clientes, por meio de múltiplas conexões simultâneas.

5.1. Metodologia de Testes

Os testes foram efetuados em um horário de baixa utilização da *Internet*, para que os resultados tivessem a menor influência possível de fatores externos a rede local.

Por meio de *script*, desenvolvido em linguagem Python pelo MSc. Eduardo Maroñas Monks, professor e coordenador do curso de Redes de Computadores na Faculdade de Tecnologia SENAC-RS, foram efetuadas múltiplas conexões através do servidor *proxy* e verificado o tempo de acesso para cada conexão. Com o aumento gradativo

do número de conexões, foi possível registrar a variação dos tempos em meio de três cenários distintos, primeiramente sem a utilização de *proxy*, com a utilização *proxy* e com a utilização de *proxy*, mas com uma lista de domínios 100 vezes maior.

Em cada teste foi efetuado a autenticação por meio de um usuário, ”*user.presidencia*”, que pertence a um grupo do AD, que no qual, não há uma restrição de acesso ao domínio *youtube.com*, e este domínio está presente na lista **multimidia**, conforme as políticas de acessos da apresentada na Figura 11.

```
http_access allow grp-presidencia multimidia
http_access deny grp-treinamento multimidia
```

Figura 11. Políticas de acesso.

A utilização do *script* consiste na passagem de três parâmetros, o endereço IP do servidor *proxy*, a porta de conexão e o número de conexões que serão geradas para cada domínio especificado em uma lista de *sites*. A variação do número de conexões, durante os testes, foi de forma linear, iniciando com 100 conexões e aumentando com um intervalo de 25 conexões por cada teste, até o último ser efetuado com 150 conexões simultâneas. Nas rotinas foi verificado os tempos de acesso para os domínios *pelotas.com.br*, *youtube.com* e realização de *download* de um arquivo hospedado no domínio *ucpel.tche.br*.

6. Resultados

Na Figura 12 pode-se verificar o gráfico de números de conexões pelo tempo médio de cada conexão, em segundos, de acesso para o *download* do arquivo no domínio *ucpel.tche.br*.

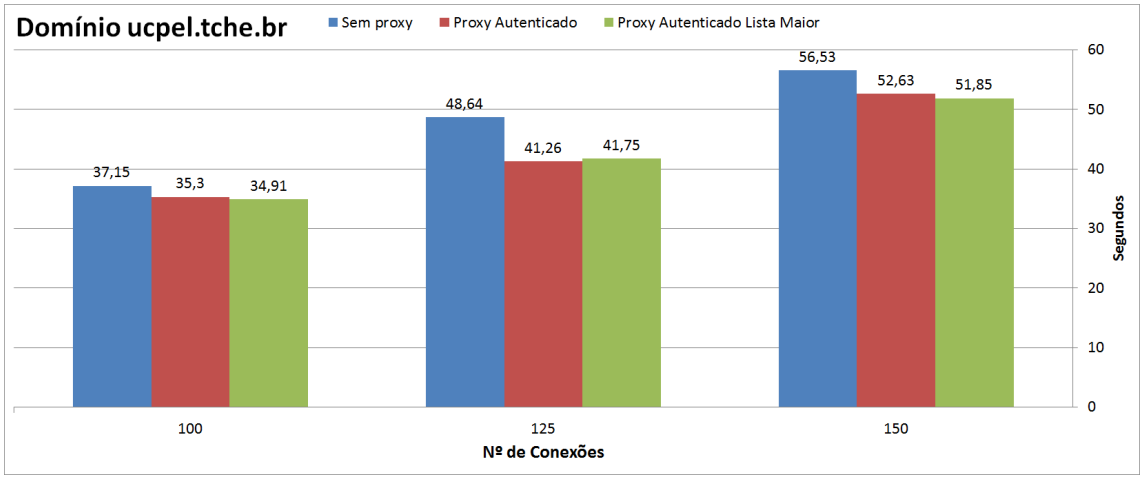


Figura 12. Gráfico para o domínio ucpel.tche.br.

Já na Figura 13, verifica-se o tempo médio para o domínio *pelotas.com.br*.

Para ambos domínios, os três cenários de testes, apresentaram resultados semelhantes, mostrando que a utilização do *proxy* não impôs atraso significativo para os acessos.

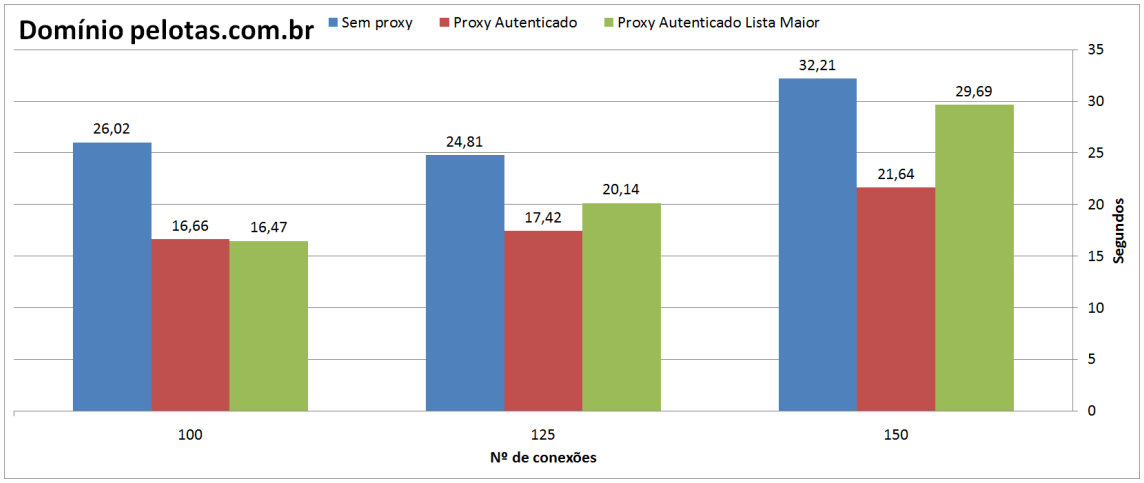


Figura 13. Gráfico para o domínio pelotas.com.br.

Na Figura 14, tem-se o gráfico para o domínio *youtube.com*, onde pode ser observado que houve uma diferença significativa nos tempos na utilização do *proxy*, com um aumento de tempo, resultando em atraso nas conexões.

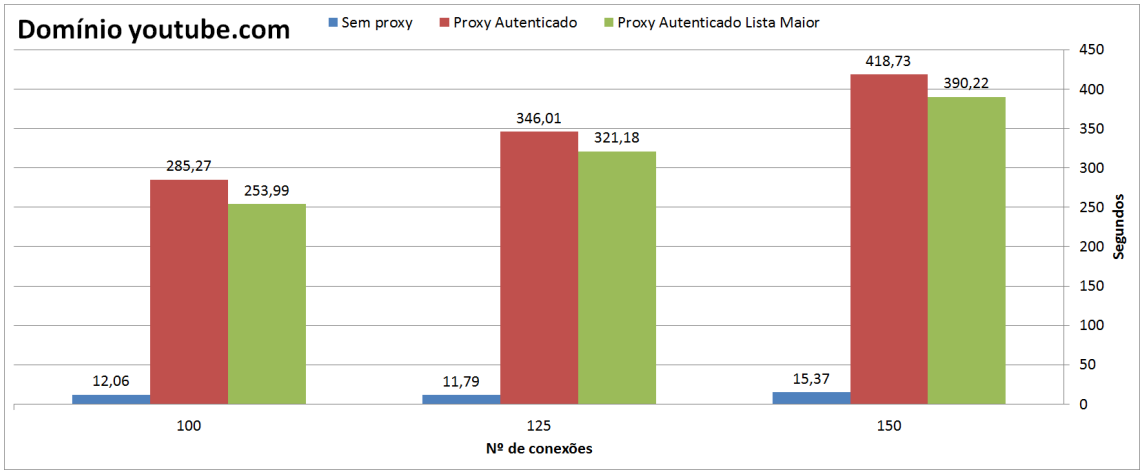


Figura 14. Gráfico para o domínio youtbe.com

Os números de conexões foram acrescidos de forma linear, da mesma forma observa-se que os tempos de acessos também se comportaram similarmente, dando uma idéia de proporcionalidade nos resultados, desta forma, pode-se estimar os tempos para um número maior de conexões do que os testados.

7. Conclusões

Pode-se concluir, com o estudo realizado, que a solução apresentada atendem de forma satisfatória a principal proposta, que é a realização do controle a conteúdo não autorizado pelas políticas estipuladas, com métodos de autenticação de usuário no *Active Directory* da Microsoft de forma automática, o que possibilita a identificação dos acessos efetuados por cada usuário da rede e disponibilizados uma flexibilidade de uso por parte dos usuário.

De qualquer forma deve-ser ter ciência de que tais recursos apresentam uma adição ao tempo de acesso as páginas WEB, com os testes efetuados e deverá ser avaliado

a sua implantação em um cenário real, para que a utilização do mecanismo de autenticação não se torne um prejuízo, em termos de desempenho a acessos de páginas na *Internet*, ao invés de benefícios.

Referências

Apache (2014). Apache foundation. http://www.apache.org/>. Acesso em: junho 2014.

Cache, S. . (2014). Squid: Optimising web delivery. Disponível em: <http://www.squid-cache.org/>. Acesso em: junho 2014.

Foca, G. (2014). Samba. Disponível em: <http://www.guiafoca.org/cgs/guia/avancado/chsamba.html>. Acesso em: junho 2014.

Microsoft (2014). Active directory. Disponível em: <http://technet.microsoft.com/pt-br/library/cc668412.aspx>. Acesso em: junho 2014.

MTI (2014). Mti - kerberos documentation. Disponível em: <http://web.mit.edu/kerberos/krb5-devel/doc/index.html>. Acesso em: junho 2014.

RFC3244 (2014). Rfc 3244 - microsoft windows 2000 kerberos change password and set password protocols. Disponível em: <http://tools.ietf.org/html/rfc3244>. Acesso em: junho 2014.

Ricci, B. (2006). *Squid Solução Definitiva*. Ciência Moderna.

Samba (2014). winbindd. Disponível em: <http://www.samba.org/samba/docs/man/manpages/winbindd.8.html>. Acesso em: junho 2014.

Samba.org (2014). Samba. Disponível em: <http://www.samba.org/>. Acesso em: junho 2014.

Tecmint (2014). Sarg : Squid analysis report generator and internet bandwidth monitoring tool. Disponível em: <http://www.tecmint.com/sarg-squid-analysis-report-generator-and-internet-bandwidth-monitoring-tool/>. Acesso em: junho 2014.

Tecmundo (2014). Dns. Disponível em: <http://www.tecmundo.com.br/o-que-e/829-o-que-e-dns-.htm>. Acesso em: junho 2014.

Vaz, D. (2013). Samba 4 e microsoft active directory. *Faculdades Senac - Redes de Computadores*.

VMWare (2014). Virtualização da vmware. Disponível em: <http://www.vmware.com/br/>. Acesso em: junho 2014.